SIEMENS

SINUMERIK ONE

Commissioning from Virtual to Real Documentation

SINUMERIK ONE

Edition 2025.07
Training Manual

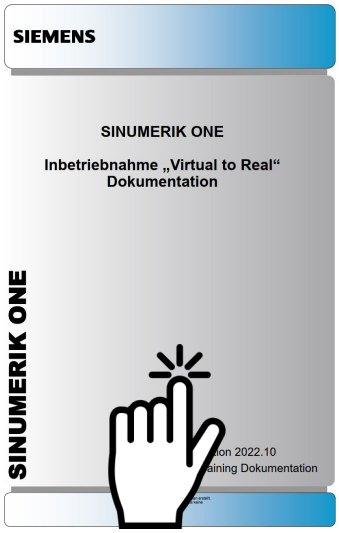
This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

Manual Structure

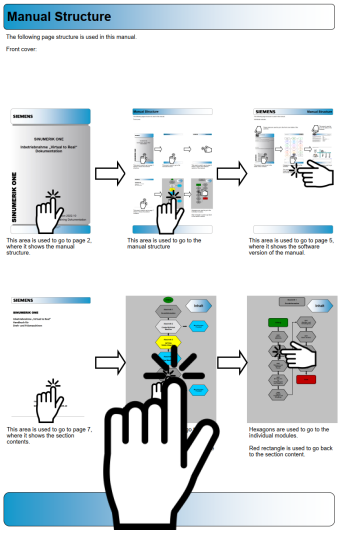
SIEMENS

The following page structure is used in this manual.

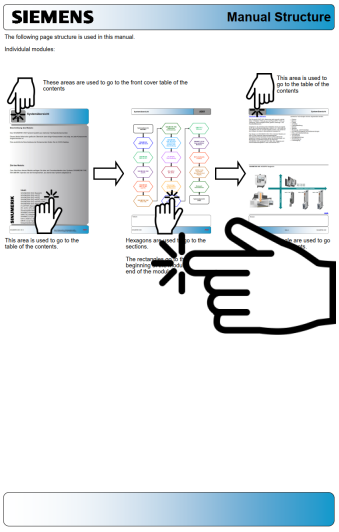
Front cover:



This area is used to go to page 2, where it shows the manual structure.



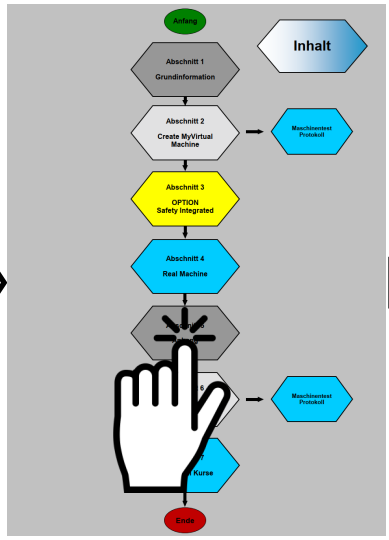
This area is used to go to the manual structure



This area is used to go to page 5, where it shows the software version of the manual.

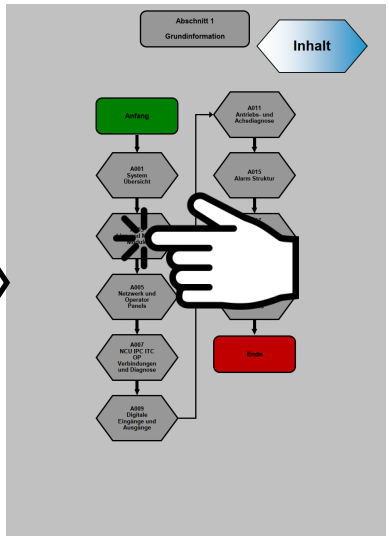


This area is used to go to page 7, where it shows the section contents.



Hexagons are used to go to the module contents.

Red oval is used to go back to the front cover.

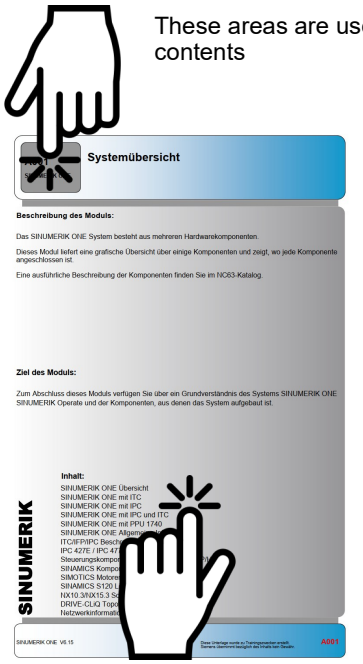


Hexagons are used to go to the individual modules.

Red rectangle is used to go back to the section content.

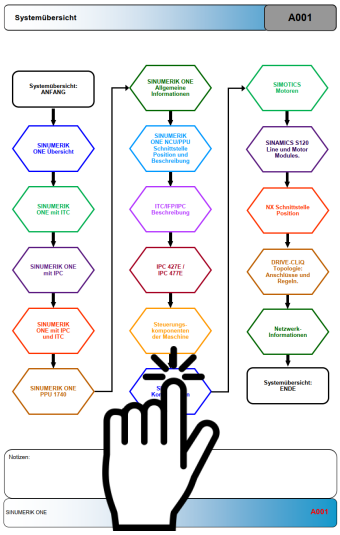
The following page structure is used in this manual.

Individual modules:



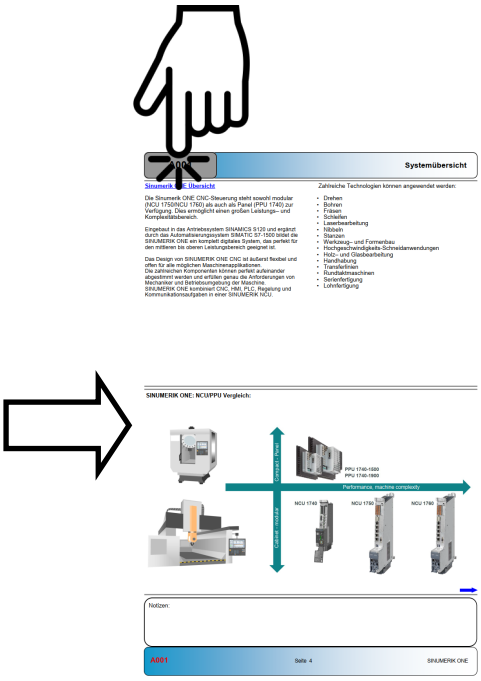
These areas are used to go to the front cover table of the contents

This area is used to go to the table of the contents.



Hexagons are used to go to the sections.

The rectangles go to the beginning of the module, and the end of the module.



The top rectangle is used to go back to the table of contents.

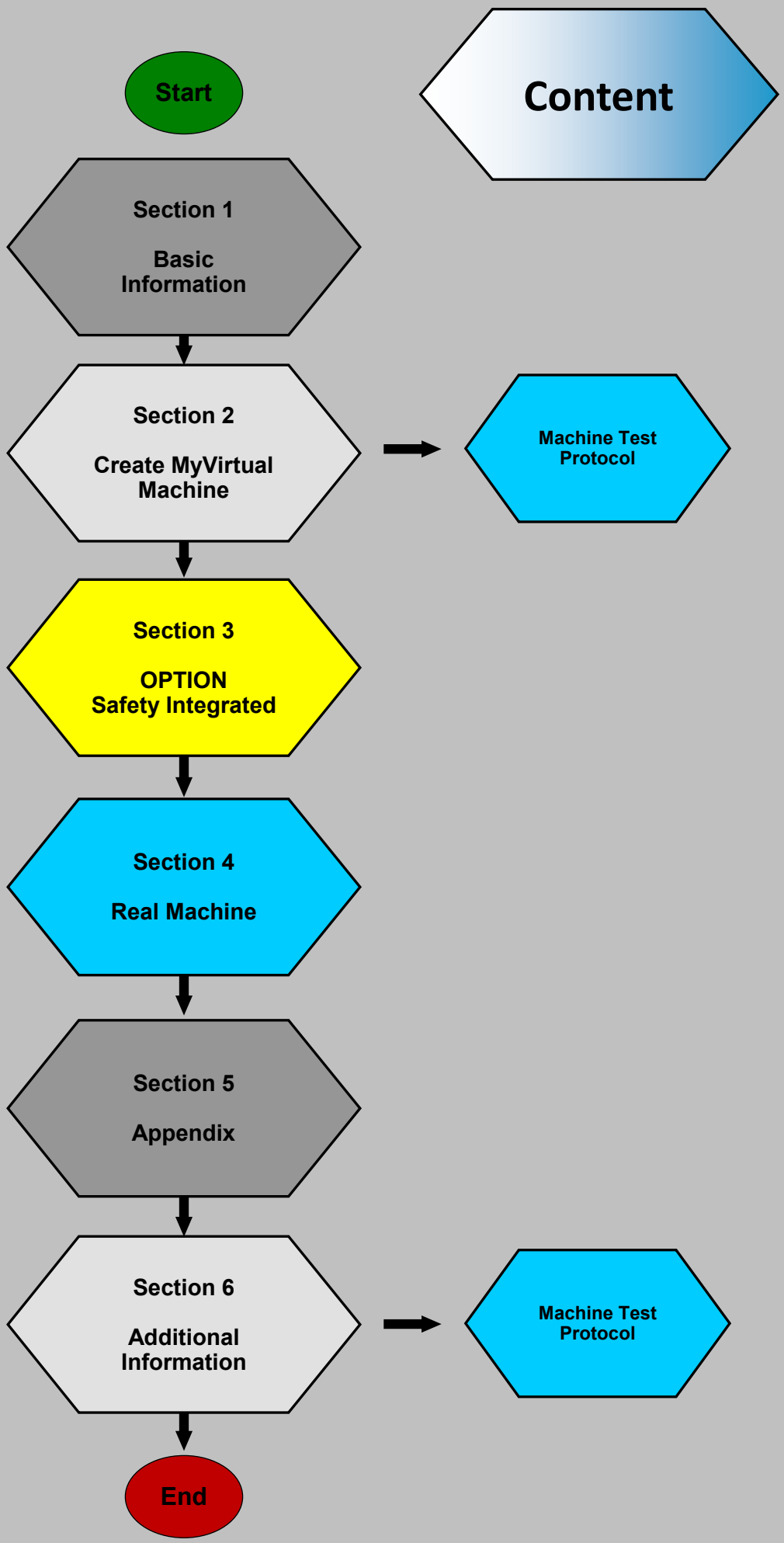
SIEMENS

SINUMERIK ONE

Commissioning from Virtual to Real
Manual for
Turning and Milling machines

Valid for:

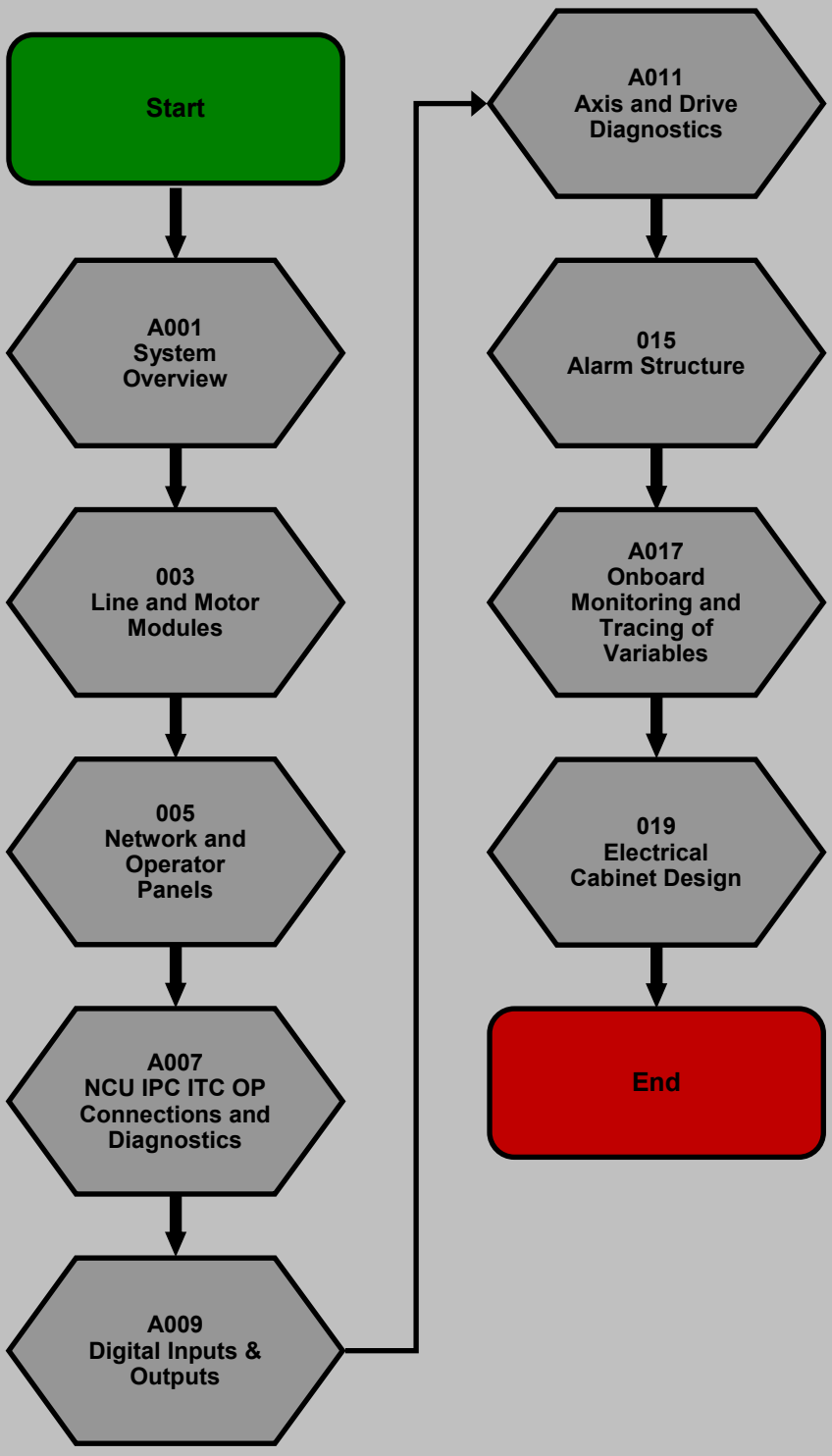
SINUMERIK ONE	**SW6.25**

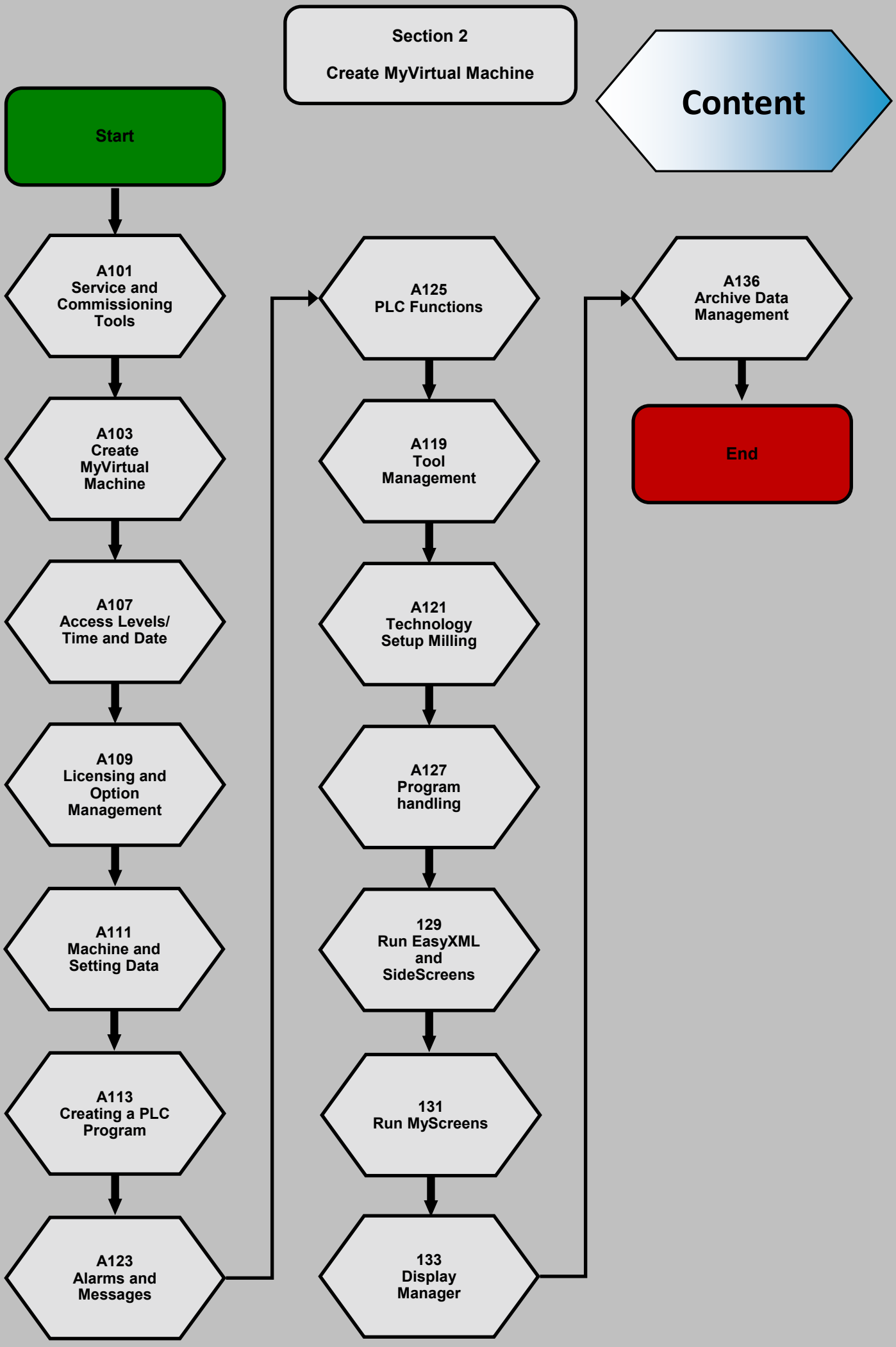

Content
Start
Section 1
Basic Information
Section 2
Create MyVirtual Machine
Machine Test Protocol
Section 3
OPTION
Safety Integrated
Section 4
Real Machine
Section 5
Appendix
Section 6
Additional Information
Machine Test Protocol
End

Section 1

Basic Information

Content

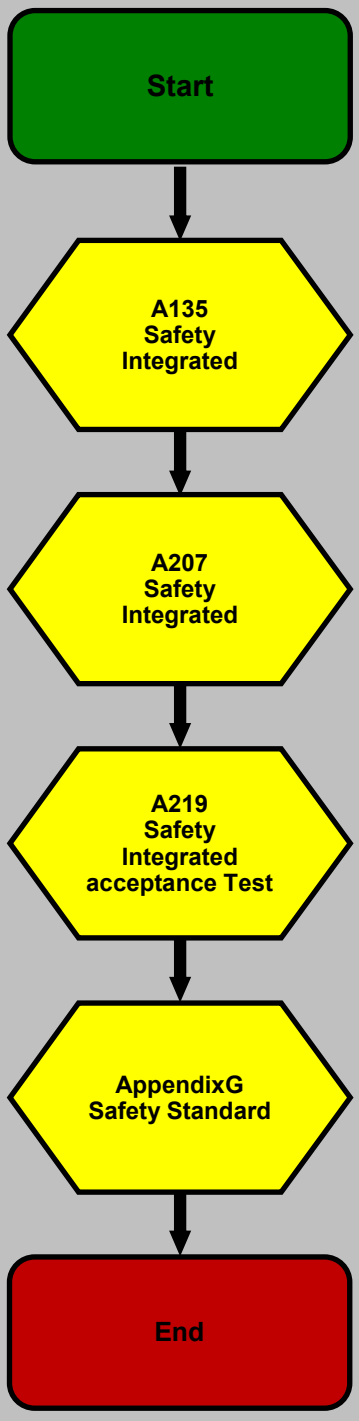



Section 2
Create MyVirtual Machine
Content
Start
A101
Service and
Commissioning
Tools
A103
Create
MyVirtual
Machine
A107
Access Levels/
Time and Date
A109
Licensing and
Option
Management
A111
Machine and
Setting Data
A113
Creating a PLC
Program
A123
Alarms and
Messages
A125
PLC Functions
A119
Tool
Management
A121
Technology
Setup Milling
A127
Program
handling
129
Run EasyXML
and
SideScreens
131
Run MyScreens
133
Display
Manager
A136
Archive Data
Management
End

Section 3

OPTION safety Integrated

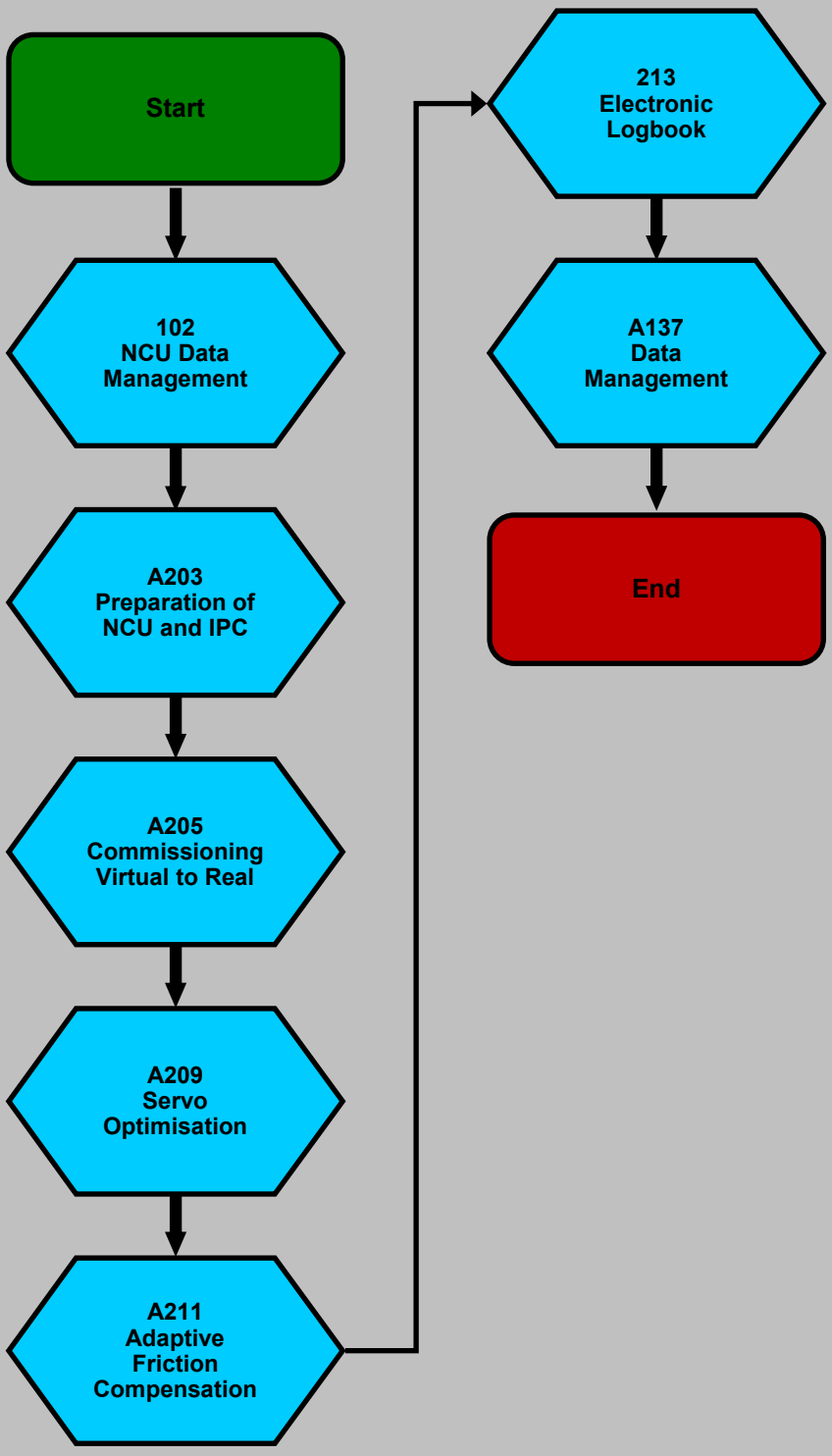
Content



Section 4

Real Machine

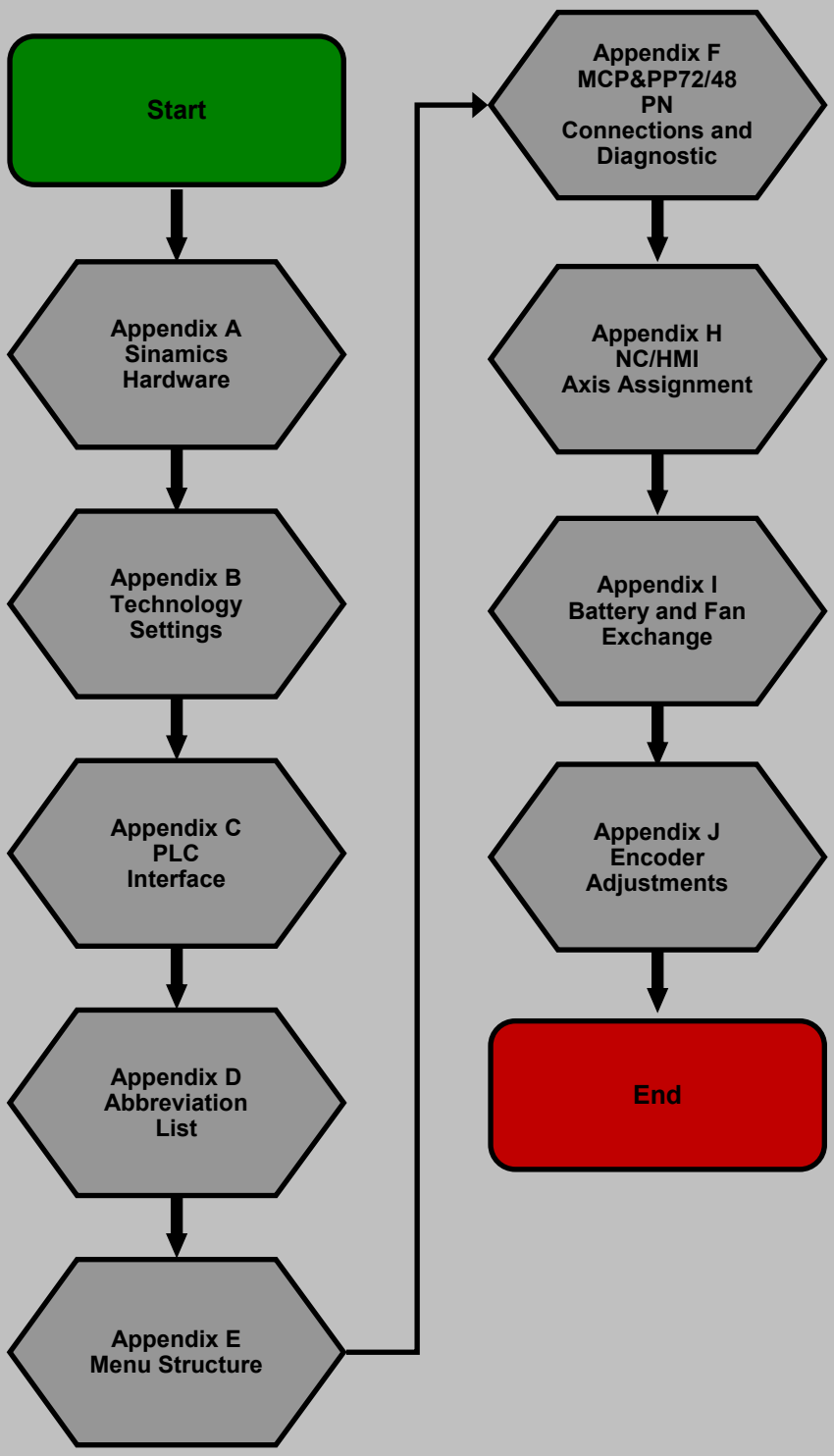
Content



Section 5

Appendix

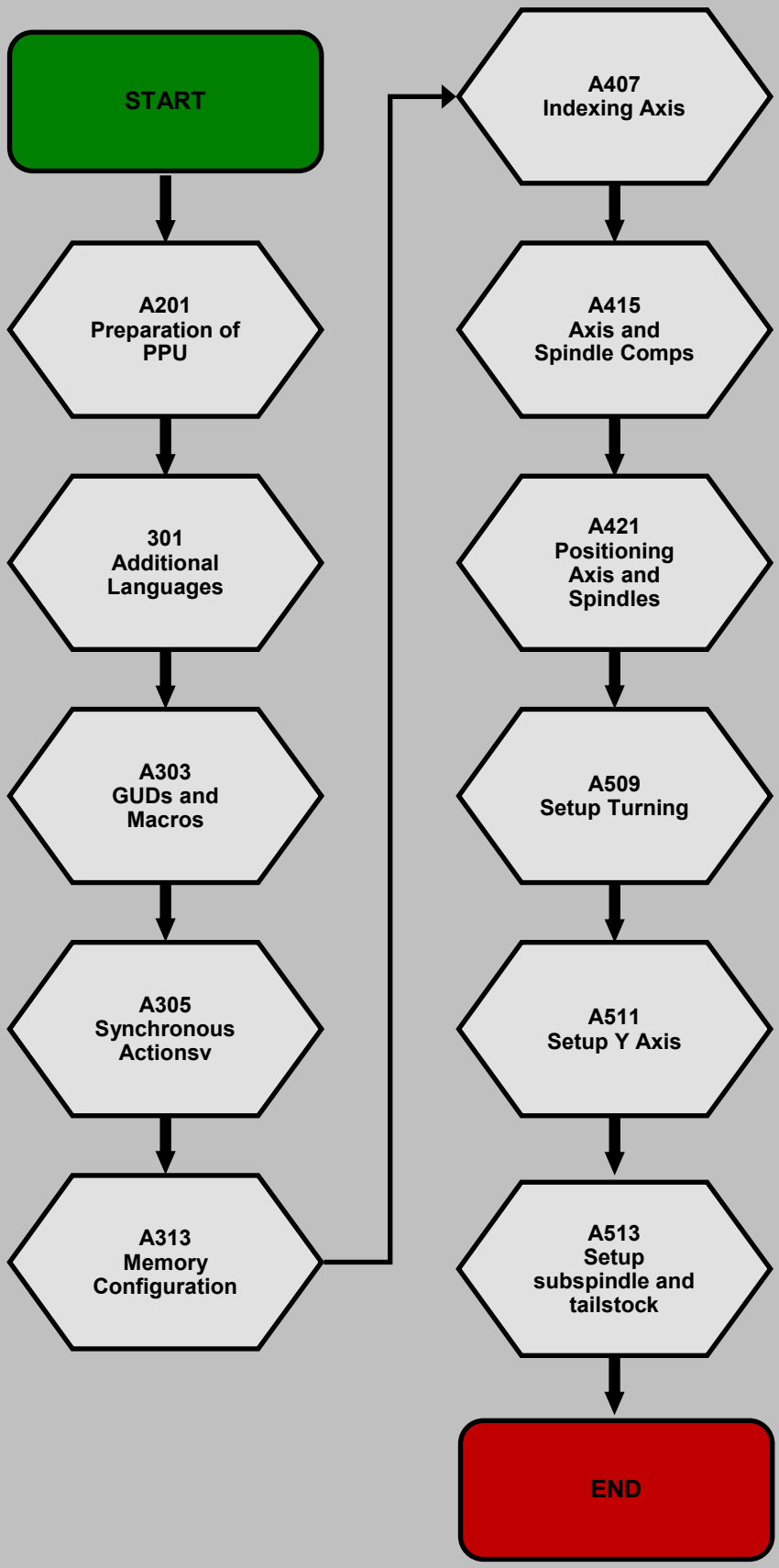
Content



Section 6

Additional Information

Content



A001

SINUMERIK ONE

System Overview

Module Description:

The SINUMERIK ONE system is made up of a variety of hardware components.

This module gives a graphical overview and description of some of the components and shows where each component is connected.

A detailed description is available in the NC63 catalogue.

motion-control-NC63-complete-english-2024-08.pdf

SINUMERIK ONE online Product Catalogue

Module objective:

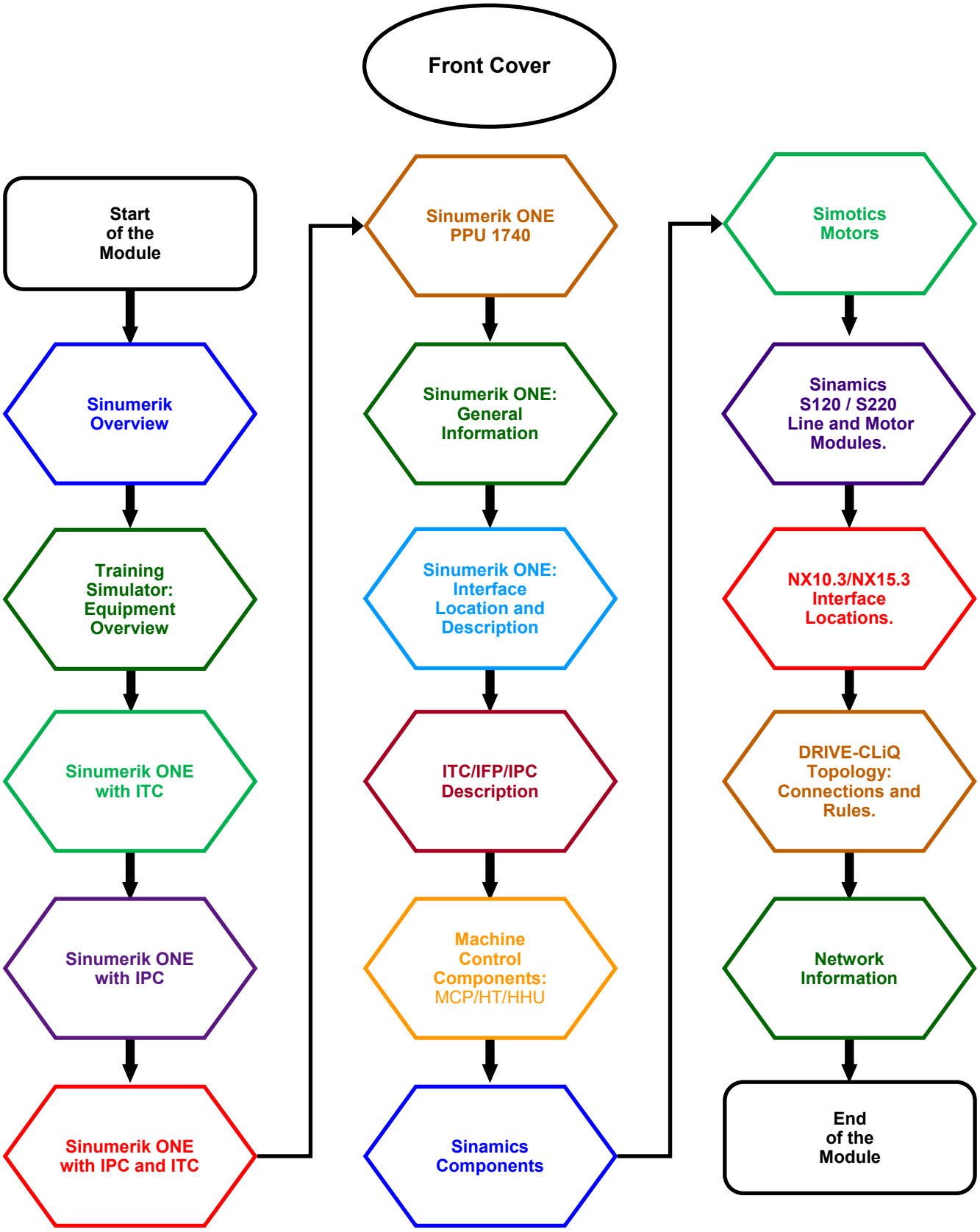
Upon completion of this module you will have a basic understanding of the SINUMERIK ONE system and the components which make up the system.

Content:

Sinumerik ONE Overview
Training Simulator: Equipment Overview
Sinumerik ONE with ITC
Sinumerik ONE with IPC
Sinumerik ONE with IPC and ITC
Sinumerik ONE with PPU 1740
Sinumerik ONE General Information
ITC/IFP/IPC Description
Machine Control Components - MCP/HT/HHU
Sinamics Components: MCP/HT/HHU
Simotics Motors
Sinamics S120 Line and Motor Modules
NX10.3/NX15.3 Interface Locations
DRIVE-CLiQ Topology: Connections and Rules
Network Information

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Sinumerik Overview
Training Simulator: Equipment Overview
Sinumerik ONE with ITC
Sinumerik ONE with IPC
Sinumerik ONE with IPC and ITC
Sinumerik ONE PPU 1740
Sinumerik ONE: General Information
Sinumerik ONE: Interface Location and Description
ITC/IFP/IPC Description
Machine Control Components: MCP/HT/HHU
Sinamics Components
Simotics Motors
Sinamics S120 / S220 Line and Motor Modules.
NX10.3/NX15.3 Interface Locations.
DRIVE-CLiQ Topology: Connections and Rules.
Network Information
End of the Module

SINUMERIK Overview :Sinumerik CNC Portfolio - 808D to ONE

Entry level machines

SINUMERIK 808D





- Panel based Compact CNC
- Turning and Milling
- Up to 6 axes/spindles
- 1 channel
- 7.5” / 8.4” - display
- SIMATIC S7-200 based PLC

Compact machines

SINUMERIK 828D



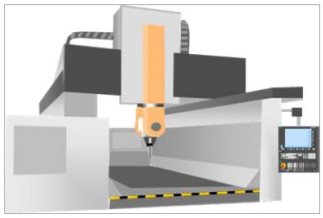


- Panel based Compact CNC
- Turning, Milling and grinding
- Up to 10 axes/spindles
- 2 channels
- 10.4” / 15.5” - multi-touch display
- SIMATIC S7-200 based PLC

Modular machines

SINUMERIK 840D sl



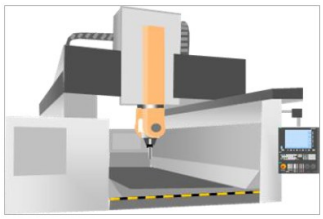


- Drive based modular CNC
- Multi technology
- Up to 31 axes/spindles
- Up to 10 channels
- 15‘‘ to 24“- multi touch display modular
- SIMATIC S7-300 based PLC

Modular machines

SINUMERIK ONE - Real and Virtual





- Drive based Compact CNC & panel based Compact CNC
- Multi technology
- Up to 31 axes/spindles
- Up to 10 channels
- 15‘‘ to 24“- multi touch display modular
- SIMATIC S7-1500 based PLC

Virtual SINUMERIK:

- Digital twin of product (machine tool)
- Digital twin of production

Notes:

Sinumerik ONE Overview

The Sinumerik ONE CNC controller is available in modular (NCU 1740/NCU 1750/NCU 1760) and panel (PPU 1740) formats.

These offer a wide performance and complexity range.

Integrated into the SINAMICS S120 drive system and complemented by the SIMATIC S7-1500 automation system, the SINUMERIK ONE forms a complete digital system that is ideally suited for the mid to upper performance range

The SINUMERIK ONE CNC design offers outstanding flexibility and openness for any machine application.

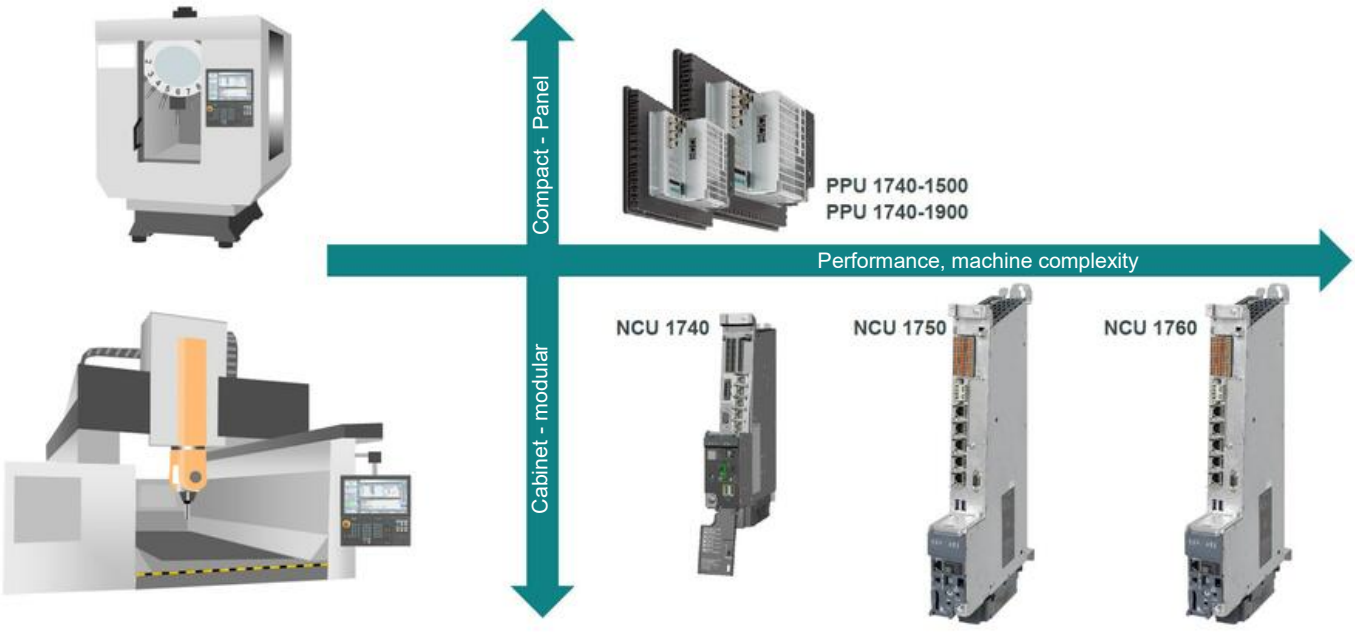
The numerous components can be perfectly matched to each other - tailored precisely to the requirements of the mechanical engineer and the subsequent operating environment of the machine.

The SINUMERIK ONE combines CNC, HMI, PLC, closed loop control and communication tasks on one SINUMERIK NCU.

Many technologies are possible:

- Turning
- Drilling
- Milling
- Grinding
- Laser machining
- Nibbling
- Punching
- Tool and mould making
- High-speed cutting applications
- Woodworking and glass processing
- Handling
- Transfer lines
- Rotary indexing machines
- Mass production
- JobShop production

SINUMERIK ONE: NCU/PPU Comparison:



Notes:

SINUMERIK ONE – NCUs

Main features in comparison with SINUMERIK 840D sl;

SINUMERIK 840D sl		SINUMERIK ONE	
	NCU710: 8 axes/spindles NCU720/730: 31 axes/spindles	NCU1740*: 12 axes/spindles NCU1750/60: 31 axes/spindles	
	PLC S7-300	PLC S7-1500F	
	1x PROFINET (2 ports NRT/RT/IRT)	1x PROFINET (2 ports, NRT/RT/IRT) 1x PROFINET (1 port, NRT/RT)	
	2x PROFIBUS DP	1x PROFIBUS DP	
	2x USB 2.0	2x USB 3.0	
	PLC rotary switch	PLC toggle switch	
	System CF card slot	System SD card slot	
	Buffer battery	No buffer battery	

Notes:

SINUMERIK ONE PPU1740: Combining the best from Sinumerik 840D sl NCU 710 and SINUMERIK 828D

SINUMERIK 840D sl NCU 710—cabinet based	
	• All high end technologies • Machine network (X120) • Standard PROFINET periphery
SINUMERIK 828D / PPU 290 - panel based	
	• Mult-touch panel • No buffer battery • Onboard periphery: 2 hand wheels 32 digital I/O 1 analogue output

Additional advantages SINUMERIK ONE PPU1740

SINUMERIK ONE PPU 1740 - Panel based	
	PLC S7-1500F
	2nd PROFINET interface
	SD Card slot
	12 axes / spindles + 8 positioning axes

SINUMERIK ONE PPU 1740: Main features in comparison with SINUMERIK 840D sl NCU710 / 828D:

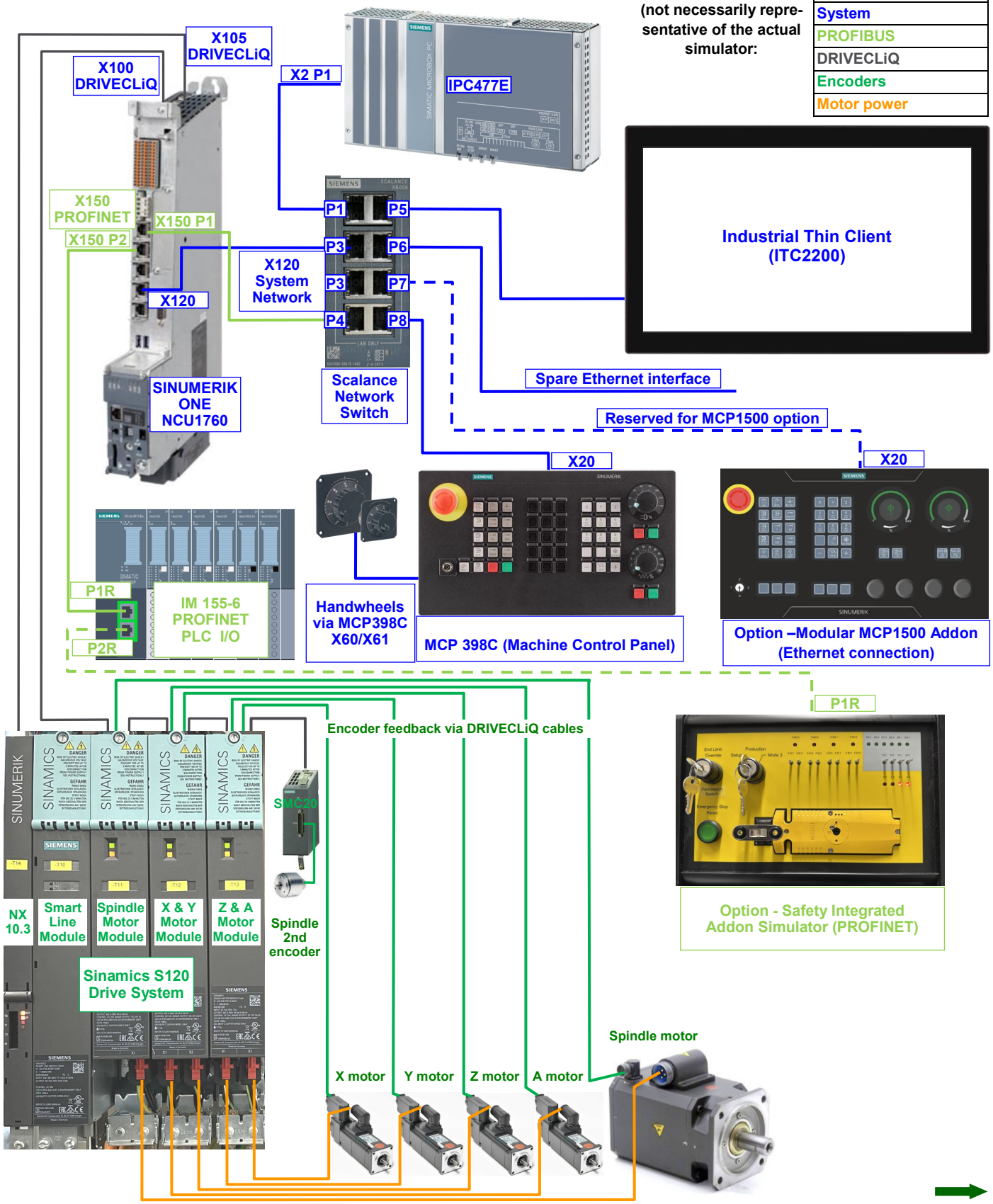
SINUMERIK 840D sl NCU 710—cabinet based		SINUMERIK ONE PPU 1740	
	8 axes / spindles, + 7 position axes	12 axes/spindles + 8 position axes	
	-	2 x onboard hand-wheels	
	24 digital I/O onboard -	32 onboard digital I/O 1 onboard analogue output (spindle)	
	1 x PROFINET 2x PROFIBUS DP	2 x PROFINET -	
	2x USB 2.0	3 x USB 3.0	
	Fan and battery	Without fan and battery	
SINUMERIK 828D / PPU 290 - panel based			
	PLC S7-200	PLC S7-1500F	
	Proprietary periphery	Standard PROFINET periphery	
	-	System network (X120)	
	M, T, G technologies	All high end technologies	

Notes:

Training Simulator: Equipment Overview

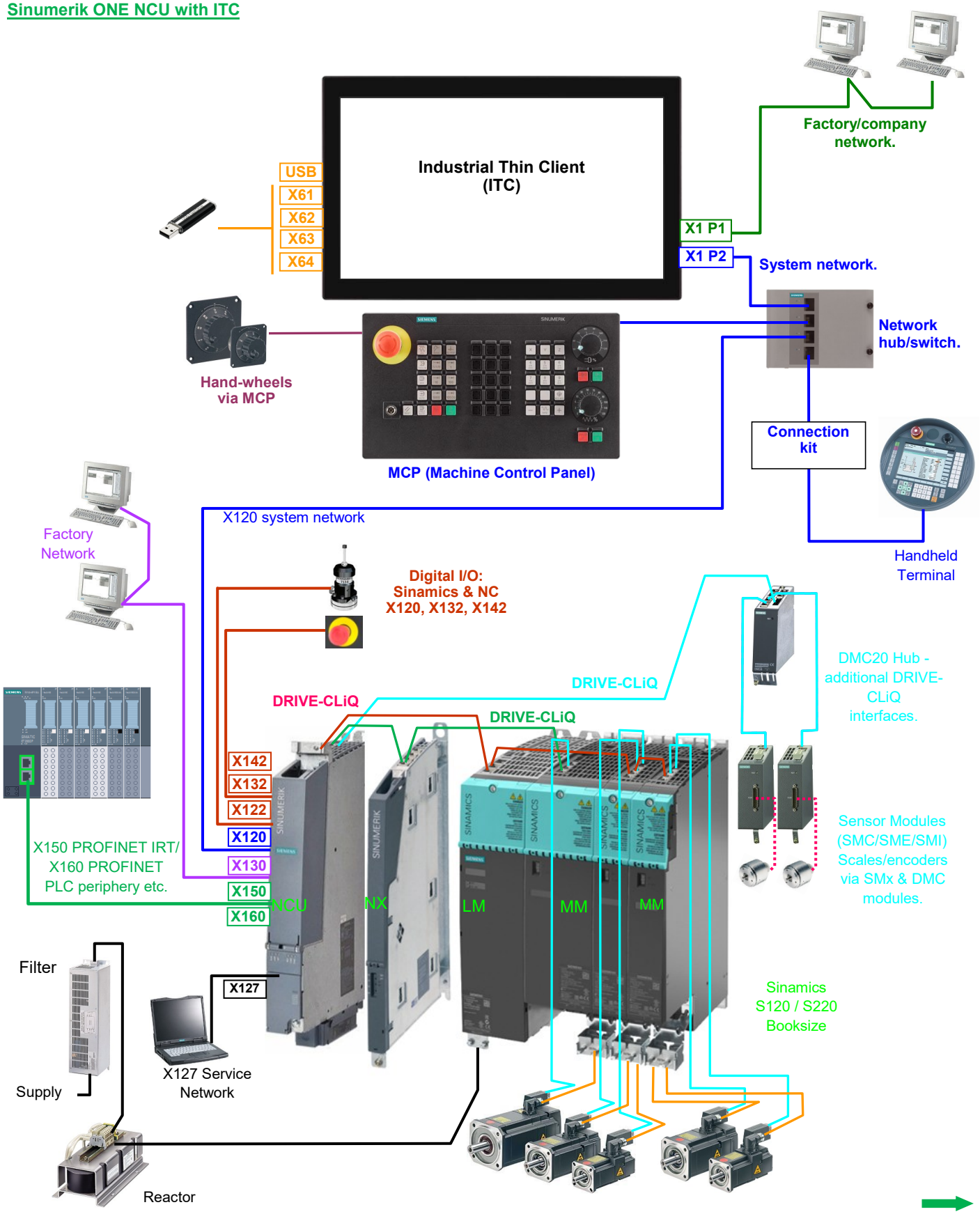
Right - colour code: (not necessarily representative of the actual simulator:

Network/function
System
PROFIBUS
DRIVECLiQ
Encoders
Motor power



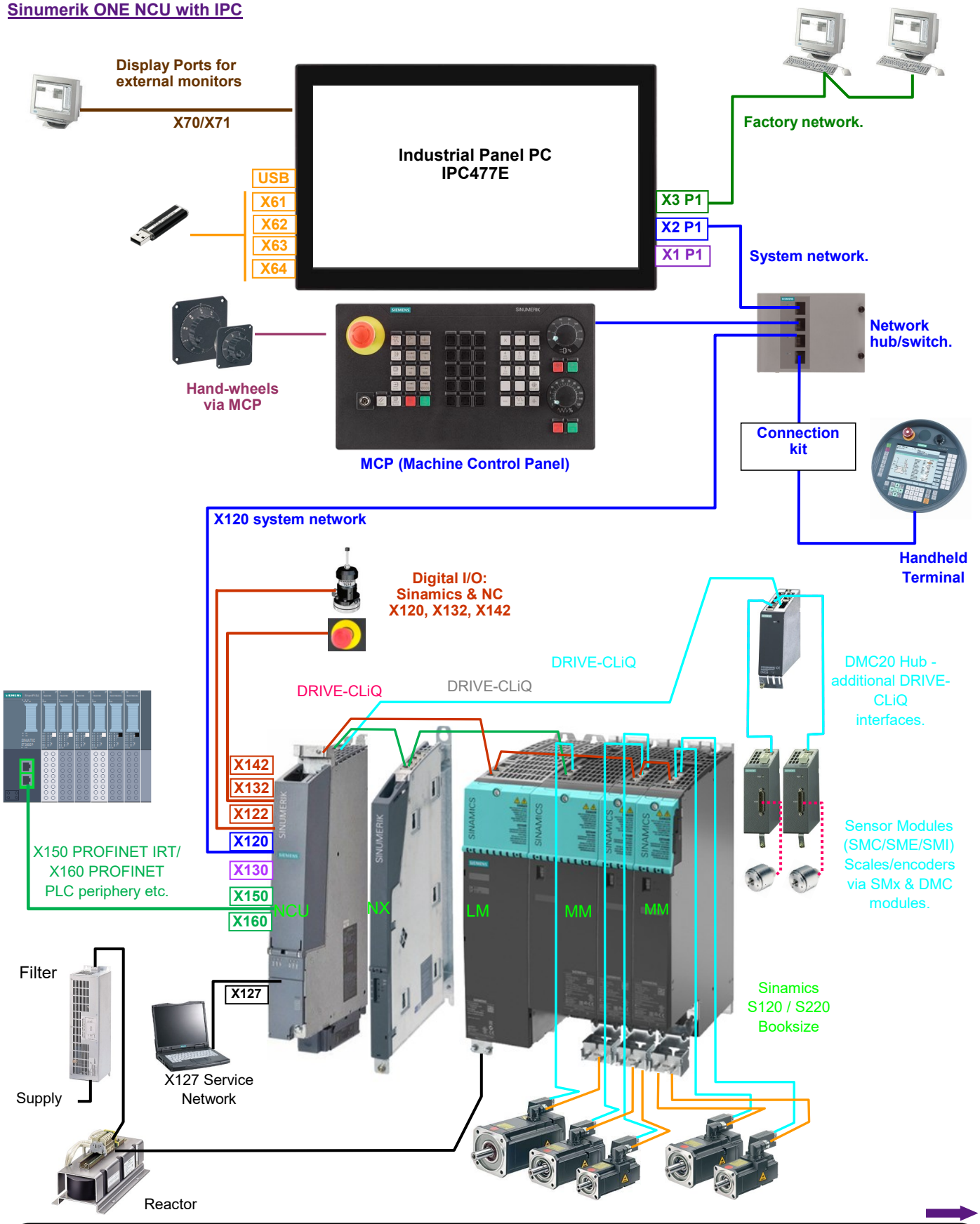
Notes:

Sinumerik ONE NCU with ITC



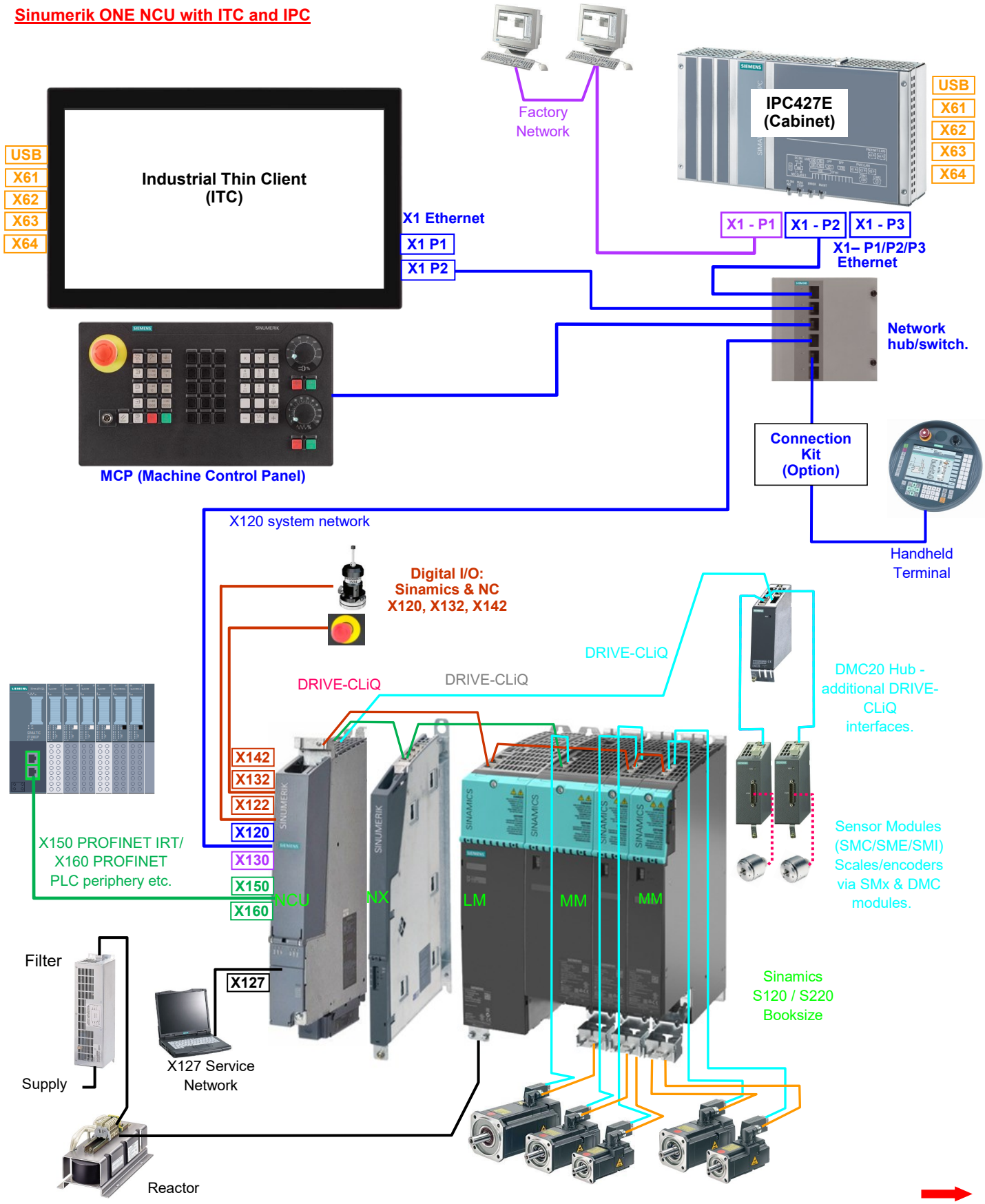
Notes:

Sinumerik ONE NCU with IPC



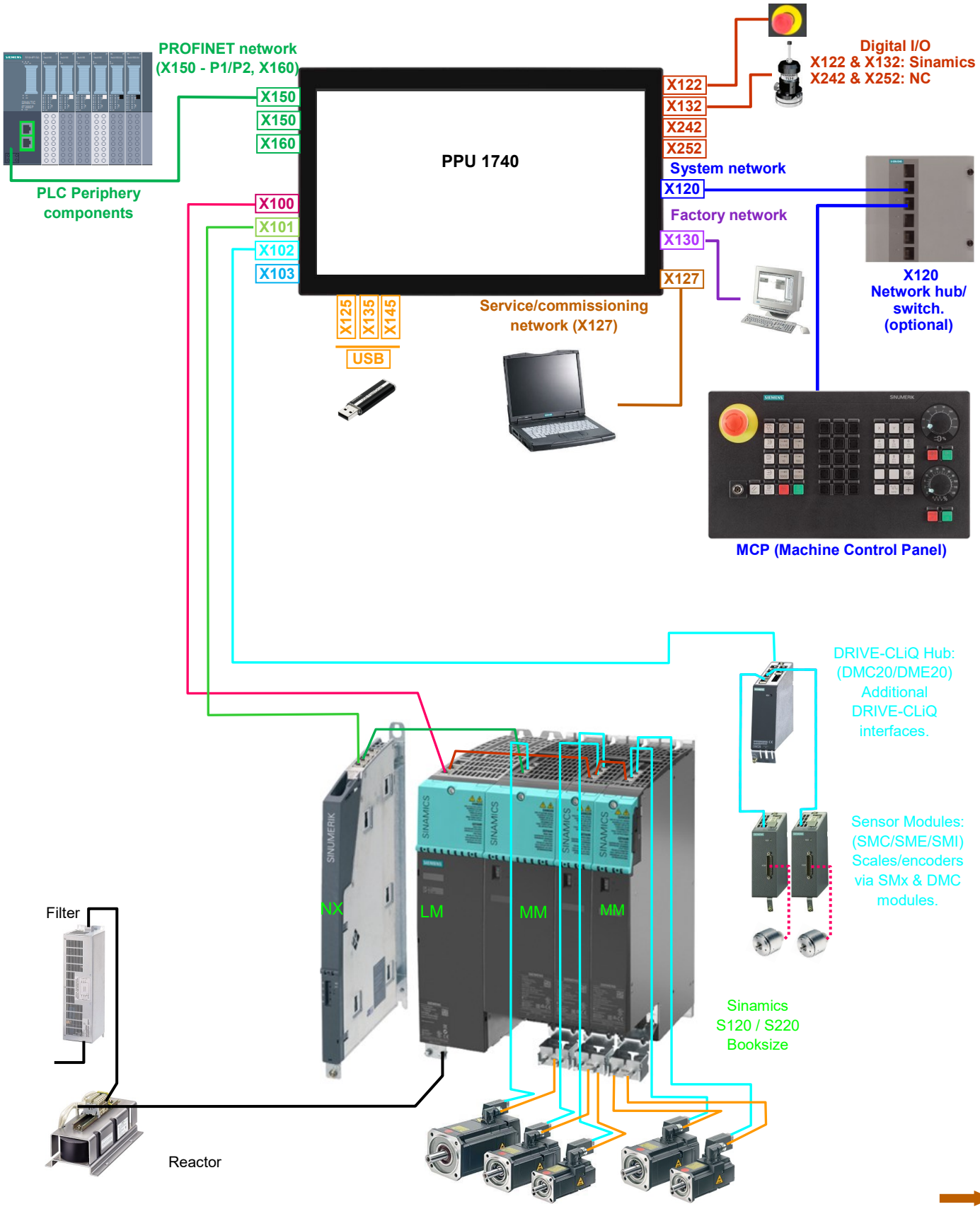
Notes:

Sinumerik ONE NCU with ITC and IPC



Notes:

Sinumerik ONE PPU 1740



Notes:

Sinumerik ONE: General Information

Type	Part number	Format	PLC	System Software
PPU 1740-1500 15” Blackline Plus	6FC5317-4AA00-1CA0	Panel (Multitouch)	S7-1500F	SD Card 8GB
PPU 1740-1900 19” Blackline Plus	6FC5317-4AA00-1DA0	Panel (Multitouch)	S7-1500F	SD Card 8GB
NCU 1740	6FC5317-4AA00-0AA0	Modular	S7-1500F	SD Card 8GB
NCU 1750	6FC5317-5AA00-0AA1	Modular	S7-1500F	SD Card 8GB
NCU 1760	6FC5317-6AA00-0AA0	Modular	S7-1500F	SD Card 8GB



PPU 1740
Front



PPU 1740
Rear



NCU 1740



NCU 1750/NCU 1760

	PPU 1740	NCU 1740	NCU 1750	NCU 1760	NX10/15
SIMATIC PLC System	PLC 1500F	PLC 1500F	PLC 1500F	PLC 1500F	-
Max. no. NC Axes	12	12	31	31	3/6
Max. no. PLC Axes (* minus active NC axes)	20*	20*	40*	50*	
Max. no. Channels	4	4	10	10	-
Max. no. Mode Groups	4	4	10	10	-
Max. no. NX modules	2	2	5	5	-
DRIVE-CLiQ Interfaces	4	4	6	6	4
Drive Current Controller Cycle Time (kHz)	8	8	8	8	8
Drive Speed Controller Cyce Time (kHz)	8	8	8	8	8
I/RF active line in kHz	8	8	8	8	-
Digital I/O X122 (Drive)	6DI, 4DI/DO	6DI, 4DI/DO	6DI, 4DI/DO	6DI, 4DI/DO	6DI, 4DI/DO
Digital I/O X132 (Drive)	6DI, 4DI/DO	6DI, 4DI/DO	6DI, 4DI/DO	6DI, 4DI/DO	-
Digital I/O X142 (NCK)	-	4DI, 4DI/DO	4DI, 4DI/DO	4DI, 4DO	-
Digital I/O X242 (NCK)	4DI, 4DI/DO	-	-	-	-
Digital I/O X252 (NCK)	4DI, 4DI/DO	-	-	-	-
Analogue Output X252 (NCK)	1	-	-	-	-
PROFINET Interfaces	X150 - P1 & P2 X160 - P1	X150 - P1 & P2 X160 - P1	X150 - P1 & P2 X160 - P1	X150 - P1 & P2 X160 - P1	-
PROFIBUS Interface	-	X126	X126	X126	-
USB Interfaces	X125, X135, X145	X125, X135, X145	X125, X135	X125, X135	-
Max. no. ITCs	-	4	4	4	-
SInumerik Operate maximum resolution	1920 x 1080 px	1920 x 1080 px	1920 x 1080 px	1920 x 1080 px	-

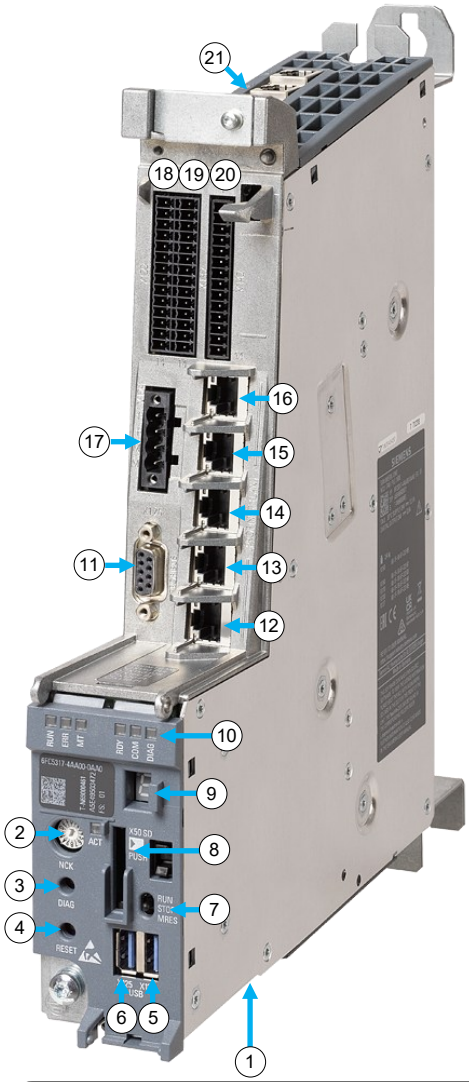
Notes:

CNC Software Information (NCU/PPU/IPC)

CNC Software Type	Version	SD Card Part Number	Delivery medium	Comments
Export - single licence	Current	**6FC5852-1YG00-0YA0**	On SD card.	None
Export - single licence	V6.24	**6FC5852-1YG24-0YA0**	On SD card.	None
Export - without licence	V6.24	**6FC5852-1YG24-0YA8**	On SD card.	None
Export - single licence	Current	**6FC5852-1YF00-0YB0**	Licence only	None
Full - single licence	Current	**6FC5852-1XG00-0YA0**	On SD card.	Export approval required
Full - single licence	V6.24	**6FC5852-1XG24-0YA0**	On SD card.	Export approval required
Full - without licence	V6.24	**6FC5852-1XG24-0YA8**	On SD card.	Export approval required
Full - single licence	Current	**6FC5852-1XF00-0YB0**	Licence only	Export approval required
Export - without licence	V6.24	**6FC5852-1YC24-0YA8**	DVD-ROM	None
Full - without licence	V6.24	**6FC5852-1XC24-0YA8**	DVD-ROM	Export approval required

Sinumerik ONE: Interface Location and Description

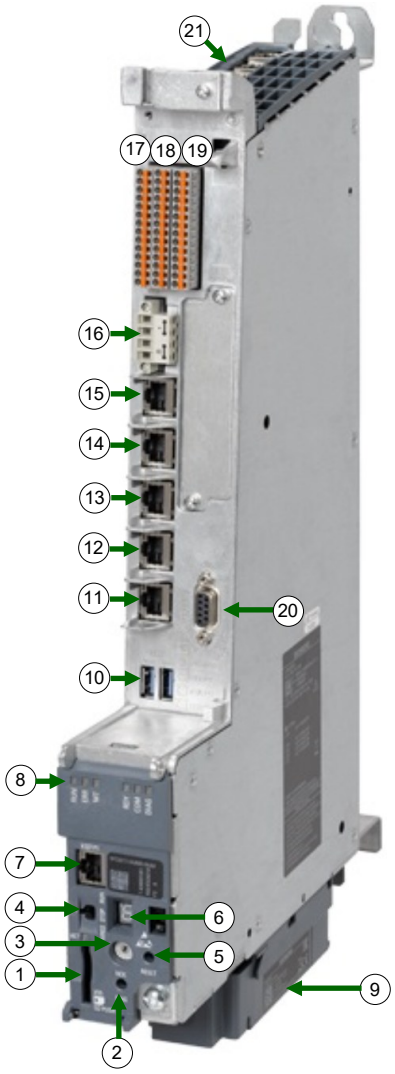
NCU 1740 6FC5317-4AA00-0AA0



No.	Interface	Description
1	X127	Servic/commissioning Ethernet (lower side)
2	-	Diagnostic rotary switch
3	X125	Diagnostic button
4	X135	Reset button
5	X125	USB interface
6	X135	USB interface
7	-	PLC mode selector switch (toggle switch)
8	X55	System SD (Secure Digital) card slot
9	-	7-Segment display
10	-	Diagnostic LEDs
11	X126	Profibus DP interface
12	X130	PN/IE - Factory Network (X130)
13	X120	PN/IE - System Network (X120)
14	X160 - P1	1 x PROFINET interface
15	X150 - P2	PROFINET IRT interface (Isochronous Real Time)
16	X150 - P1	PROFINET IRT interface (Isochronous Real Time)
17	X124	Power supply - 24V DC
18	X122	Drive digital inputs and outputs
19	X132	Dirive digital inputs and outputs
20	X142	NCK Digital inputs and outputs
21	X100 - X105	4 x DRIVE-CLiQ interfaces

Notes:

NCU 1750 6FC5317-5AA00-0AA1 & NCU 1760 6FC5317-6AA00-0AA0

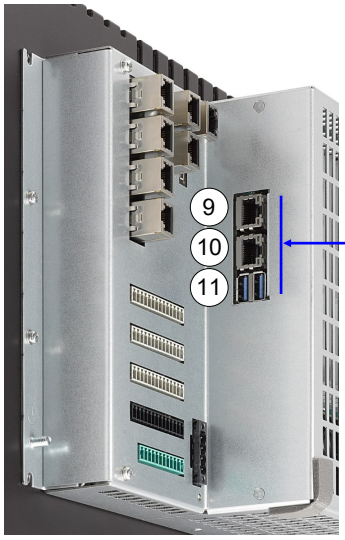


Item	Interface	Description
1	X55	System Secure Digital (SD) Card Slot
2	-	Diagnostics Button
3	-	NCK Commissioning Switch (rotary switch type)
4	-	PLC Run/Stop/Commissioning Toggle Switch
5	-	Reset button
6	-	7 Segment Display
7	X127	PN/IE Service/Commissioning Interface
8	-	Diagnostics LEDs
9	-	Fan housing
10	X125/X135	USB Interfaces
11	X130	PN/IE - Factory Network (X130)
12	X120	PN/IE - System Network (X120)
13	X160 - P1	1 x PROFINET Interface
14	X150 - P2	PROFINET Interfaces IRT (Isochronous Real Time)
15	X150 - P1	PROFINET Interfaces IRT (Isochronous Real Time)
16	X124	+24V DC Supply
17	X122	Drive Digital I/O
18	X132	Drive Digital I/O
19	X142	NCK Digital I/O
20	X126	PROFIBUS Interface
21	X100 - X105	6 x DRIVE-CLiQ Interfaces

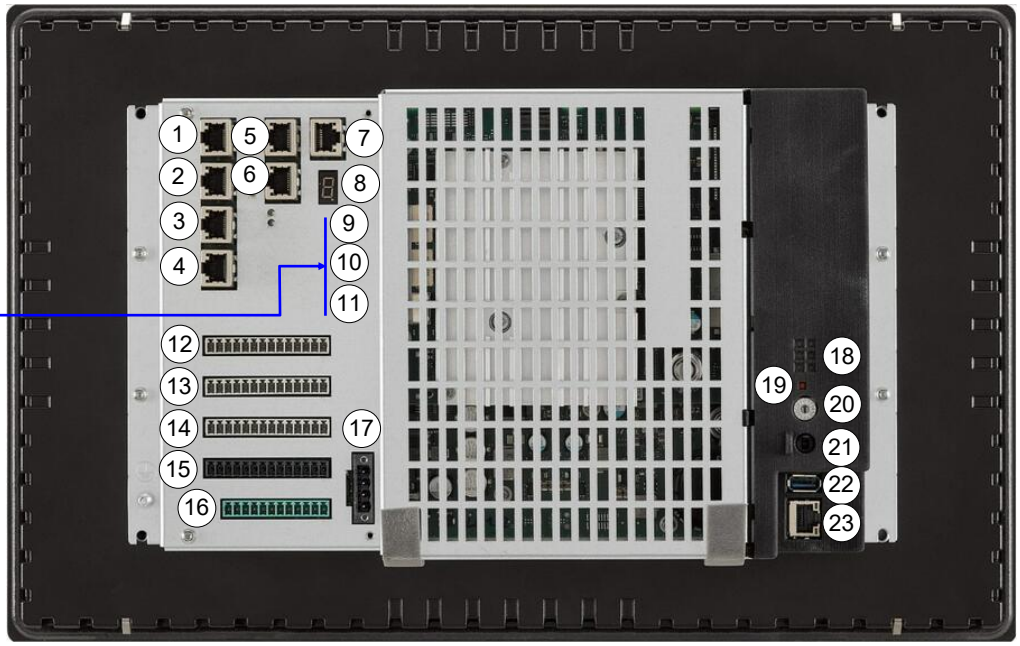
Notes:

PPU 1740 6FC5317-4AA00-1DA0 / 6FC5317-4AA00-1CA0

Sideview of PPU 1740 showing the location of items 9, 10 & 11.



Rear view of PPU 1740

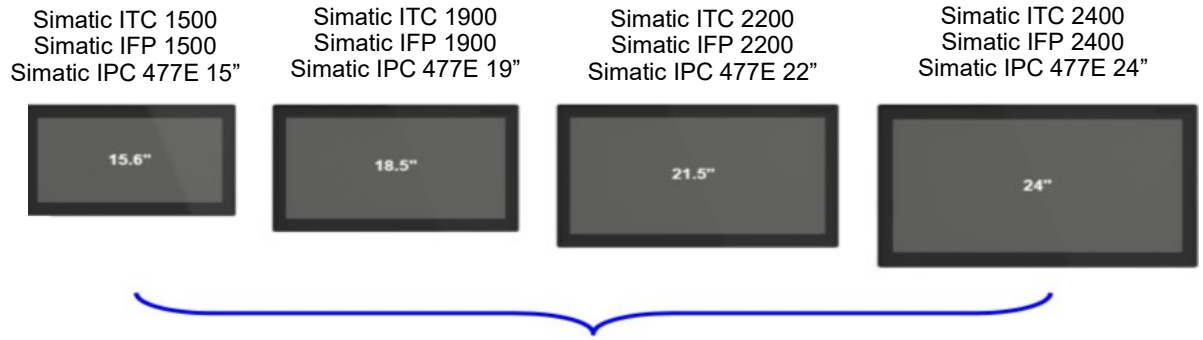


Item	Interface	Description
1	X100	DRIVE-CLiQ Interface
2	X101	DRIVE-CLiQ Interface
3	X102	DRIVE-CLiQ Interface
4	X103	DRIVE-CLiQ Interface
5	X150 P1	PROFINET IRT Port 1
6	X150 P2	PROFINET IRT Port 2
7	X160	PROFINET
8	-	7 Segment Display
9	X120	System Ethernet
10	X130	Factory Ethernet
11	X135/X145	USB
12	X122	Sinamics digital I/O
13	X132	Sinamics digital I/O
14	X242	NC digital I/O
15	X252	NC digital I/O
16	X143	Handwheel interface - two handwheels
17	X1	DC power supply - 24V
18	-	Diagnostic LEDs
19	-	Reset button
20	-	Diagnostic rotary switch
21	-	PLC Run/Stop switch
22	X125	USB
23	X127	Service/commissioning Ethernet

Notes:

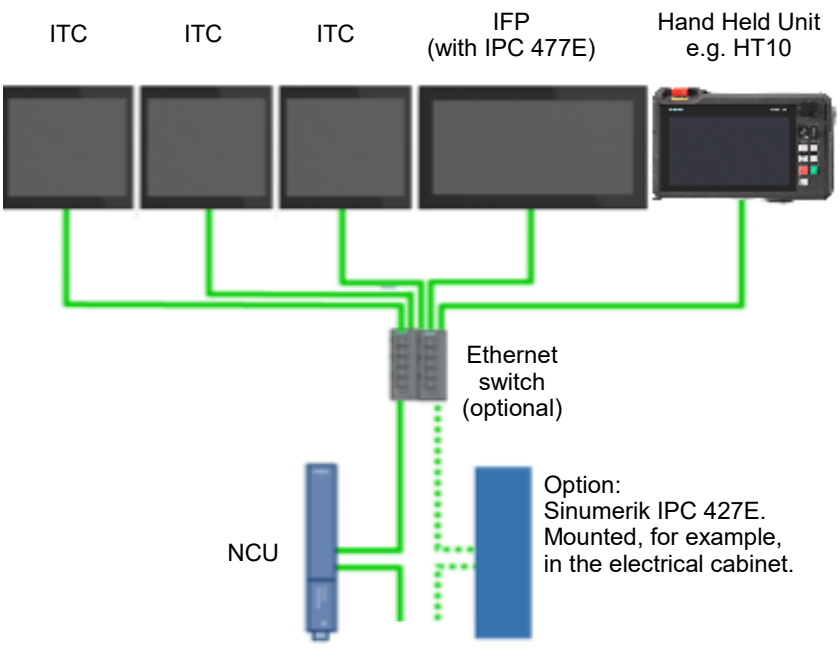
ITC/IFP/IPC Description

A selection of operator panels are available for use with Sinumerik Operate HMI embedded software on the NCU (Linux platform) or on an IPC (Windows platform)





Multiple Operator Panels



User interface on several ITCs.

- NCU 1750/1760: up to 4 ITCs
- Different screen resolutions possible
- 100 m between the switch and device
- Arbitrary number of panels through intelligent displacement with more than 2 ITCs

ITC Variants (Multi-touch) :
- ITC1500 15” (2 variants)
- ITC1900 19” (2 variants)
- ITC2200 22”
- ITC2400 24”

IFP Variants (Multi-touch) :
- IFP1500 15”
- IFP1900 19”
- IFP2200 22”
- IFP2400 24”

ITC: Industrial Thin Client
IFP: Industrial Flat Panel
IPC: Industrial PC

Notes:

Industrial Thin Clients (ITCs)

Industrial Thin Client Unit combined with OP (Multi-touch widescreen TFT display, capacitive touch sensor).
Available in 5 sizes - 12”/15”/19”/22”/24”



Maximum of 4 TCUs per NCU or 2 per PPU1740.

Interfaces:
2 x Industrial Ethernet - 1,000 Mbits
4 x USB
24V DC Supply

General Information

Part numbers:

ITC1200 V3, 12" Multi-touch:	**6AV6646-1BA12-0NA1**	NCU 1740 /1750 /1760 / PPU1740 - all
ITC1500 V3, 15" Multi-touch:	**6AV6646-1BA15-0NA1**	NCU 1740 /1750 /1760 / PPU1740 - all
ITC1500 V3, 15" Multi-touch:	**6AV6646-1BA16-0NA1**	NCU 1740 /1750 /1760
ITC1900 V3, 19" Multi-touch:	**6AV6646-1BA18-0NA1**	NCU 1740 /1750 /1760 / PPU1740 - 1900
ITC1900 V3, 19" Multi-touch:	**6AV6646-1BA20-0NA1**	NCU 1740 /1750 /1760
ITC2200 V3, 22" Multi-touch:	**6AV6646-1BA22-1NA1**	NCU 1740 /1750 /1760
ITC2400 V3, 24" Multi-touch:	**6AV6646-1BA24-0NA1**	NCU 1740 /1750 /1760

Multi-touch screens allow up to 5 finger operation.

Industrial Flat Panels (IFPs)

Industrial LCD monitors are for use with industrial IPCs - all compatible with NCU 1740 /1750 /1760 (not PPU1740)



Compatibility:

The IFPs are compatible with the following IPC427E IPCs

6AG4141-1AA17-0FA0
6AG4141-5AB17-0FA0

General Information

Part numbers:
IFP1500 V2 15” Multi-touch: **6AV7863-5MA10-2NA0**
IFP1900 V2 19” Multi-touch: **6AV7863-6MA10-2NA0**
IFP2200 V2 22” Multi-touch: **6AV7863-4MA10-2NA0**
IFP2400 V2 24” Multi-touch: **6AV7863-7MA10-2NA0**

- Multi-touch screens allow up to 5 finger operation.

Extended V2 host module: **6AV7860-3EH00-0AA0**
HDBaseT V2 transceiver
Display port
USB to HDBaseT
24 V power supply

Click for Installation guide

IFP Features:

- Rugged front
- Brilliant TFT display with a wide reading angle
- 1920 x 1080 pixel resolution
- Up to 16 million colours
- Capacitive multi-touch screen available in sizes 15", 19", 22” and 24"
- Can be placed up to 100 m from the IPC
- DVI-D interface
- DisplayPort V1.2 interface
- 2 x USB 2.0 type A
- Supply: 24V DC

Notes:

SIMATIC Industrial Panel & Box PCs

IPC277G Panel PCs:

All-in-one device with integrated industrial PC (Panel PC)

SIMATIC IPC277G, nanopanel PC, 15" - **6AV7886-0TE21-2AC0**
SIMATIC IPC277G, nanopanel PC, 19" - **6AV7886-0UE22-2AC0**
SIMATIC IPC277G, nanopanel PC, 15" - **6AV7886-0TE24-2AC0 (Slim)**
SIMATIC IPC277G, nanopanel PC, 19" - **6AV7886-0UE24-2AC0 (Slim)**

IPC PX-39A Panel PCs:

SIMATIC IPC PX-39A, 22" - **6AV7242-3NB10-1AA0**
SIMATIC IPC PX-39A, 24" - **6AV7242-3PC10-1AA0**

IPC BX-39A & IPC277G Panel PCs:

IPC277G Standard Performance - **6ES7647-8CE21-2AC3**
BX-39A High Performance - **6AG4142-3HB10-1AA0**
BX-39A High Performance Plus - **6AG4142-7HC10-2AA0**

All compatible with NCU 1740 /1750 /1760 (not PPU1740)

Panel type PC:



Box type PC



Notes:

Machine Control Components

Various operating components can be integrated into the system. These include Machine Control Panels (MCPs), Hand Held Unit (HHU), Hand Held Terminals (HT).



MCP310



MPP483



MCP483



MCP398



Modular MCPs



Mini HHU



HT2



HT8



HT10

Please refer to ***Appendix F*** for detailed information about machine control components.

Sinumerik Blackline Plus MCP (MCP398C PN)

The MCP398C offers additional functionality when used in conjunction with Expansion Modules (EM).

Sinumerik blackline plus MCP							
MCP 398C PN		EM 66		EM 131		EM 187	
							
Width	Height	Width	Height	Width	Height	Width	Height
398	225	66	225	131	225	187	225
379.5+1	206.5+1	47.5+1	206.5+1	112.5+1	206.5+1	Planning phase	

EM 66: When installed next to the MCP 398C, this results in a total width of 464 mm. The EM 66 is therefore suitable for use with the ITC 1900.

- Four 16 mm or three 22 mm slots (depending on the size of the switching elements).
- Space for the integration, for example, of pushbuttons, USB interface or other operator controls.

EM 131: When installed next to the MCP 398C, this results in a total width of 529 mm. The EM 131 is therefore suitable for use with the ITC 2200.

- Six 16 mm or 22 mm slots (depending on the size of the switching elements).
- Space for the integration, for example, of pushbuttons, a USB interface, an Ethernet interface, a handwheel, an additional override switch or other operator controls.

EM 187: When installed next to the MCP 398C, this results in a total width of 585 mm. The EM 187 is therefore suitable for use with the SIMATIC IPC 477E 24".

- Eight 16 mm or 22 mm slots (depending on the size of the switching elements).
- Space for the integration, for example, of pushbuttons, a USB interface, an Ethernet interface, a handwheel, an additional override switch or other operator controls.

Notes:

Sinumerik ONE: Sinamics Components

Line Modules
Smart Line Modules
Active Line Modules
Basic Line Modules



Line-side components
Line chokes
line filter
Active Interface Modules



Motor Modules
Single Motor Modules
Double Motor Modules
Booksize Compact



Chassis Modules
AIM / ALM / MM



SINAMICS S120 / S220
S120M
HLA



SINAMICS S120 Combi



DC Link Components
Braking resistors
Braking Module
Capacitor Module
Control Supply Module



Control Units
NX10.3 / NX15.3
CU320-2
CU310-2



Supplementary Components
DMC20 / DME20
TM31
TM41
TM15
TM120
VSM10



Sensor Modules
SME20 /25/120/125
SMC10 /20/30/40



Sinamics Components: Brief Description

Item	Meaning	Comments
Sinamics	Siemens drive system/components	Hardware and software belonging to Siemens drive family
DRIVE-CLiQ	Sinamics drive communication bus	RJ45 cables - with or without 24V DC cores. (See individual component requirements)
CU	Control Unit - drives	CU310-2 PN/CU32-2PN. Positioning of axes via PROFINET
NX	Numerical Extension	Additional servos. NX10 - 3 servos. NX15 - 6 servos.
LM	Line Module - see below	Supplies DC voltage to Motor Modules
ALM	Active Line Module	Regenerative, stabilised DC link
SLM	Basic Line Module	Regenerative, non stabilised DC link
BLM	Smart Line Module	Non regenerative, non stabilised DC link
MM	Motor Module	Control of AC servo motors
AIM	Active Interface Module	Combined filter/reactor
SM	Sensor Module - see below	Interfacing of additional encoders/scales into the DRIVE-CLiQ bus - see below
SMC	Sensor Module Cabinet	Cabinet mounted device
SME	Sensor Module External	Machine mounted device
SMI	Sensor Module Internal	Internally mounted (DRIVECLiQ compatible servo motors/scales/encoders)
DMC20	DRIVE-CLiQ Hub	Cabinet mounted device for additional DRIVE-CLiQ interfaces
DME20	DRIVE-CLiQ Hub	Machine mounted device for additional DRIVE-CLiQ interfaces
OP	Operator Panel - VNC viewer platform	HMI hardware comprising of a display and optional keyboard.
TM	Terminal Modules (TM15, 31, 41, 20)	Sinamics I/O (digital/analogue)/encoder/temperature sensors)
HLA	Hydraulic Linear Actuator	Control of hydraulic axes.
VSM	Voltage Sensing Module	Only for S120 Chassis integrated format

Notes:

Sinumerik ONE: Simotics Motors

SIMOTICS S servo motors
SIMOTICS S-1FT7
SIMOTICS S-1FK7
SIMOTICS S-1FK2 (2 cable connections)
SIMOTICS S-1FG1



0.18 – 1,650 Nm

SIMOTICS M main motors
SIMOTICS M-1PH8
SIMOTICS M-1FE1
SIMOTICS M-1FE2
SIMOTICS M-1PH2
SIMOTICS M-1PH3



13.0 – 12,435 Nm

SIMOTICS L linear motors
SIMOTICS L-1FN3



Up to 20,700 Nm

SIMOTICS T torque motors
SIMOTICS T-1FW6
SIMOTICS T-1FW6 HS

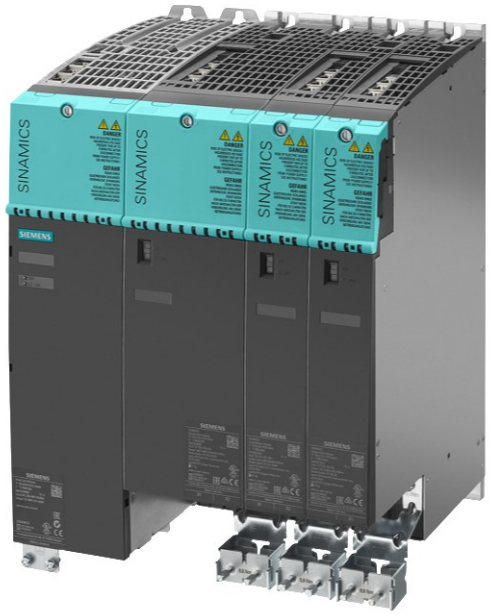


34.4 – 10,900 Nm

Notes:

Sinamics S120/S220 Line and Motor Modules.

The Sinamics S120 / S220 drive components are the most commonly used with the Sunumerik ONE.
The S120 versions are now being superseded by the S220 versions.



S120 Line Module (LM)

S120 Motor Modules (MM)



S220 Line Module (LM)

S220 Motor Modules (MM)

Overview: SINAMICS S220 Booksize Smart Line Modules

The SINAMICS S120 Booksize range set the standards for performance and quality in drive technology for machine tools, production machines, and general motion control applications.

With Sinamics SW V5.2, the SINAMICS S120 Booksize system has reached full maturity and is now set to make an evolutionary leap.

The proven servo drive system, SINAMICS S120, continues under the umbrella of the innovative SINAMICS S220 with new motor modules for the Booksize variants.

The new motor modules represent space savings, increased overload capacity, stability, and efficiency.

New innovations have been developed on the SINAMICS S220 platform, which offers a unique architecture for future requirements as well as more powerful features compared to the proven SINAMICS S120.

General Sinamics S120 / S220 Information

- Line and motor modules supply power to the AC servomotors.
- All Motor Modules have to be connected to the Drive-CLiQ bus system.
- All Line Modules, with the exception of 5 & 10Kw SLM modules, have to be connected to the Drive-CLiQ bus system.
- Line modules require a 3 phase AC supply. This is rectified into a DC supply which is then supplied to the motor modules via the DC link bus bars.
- Motor modules convert the DC link into a controllable AC supply which is used to drive the servo motors.
- By regulating the frequency and voltage of the AC supply to the servo motors it is possible to precisely control the motor
- Line and motor modules are available in different variants depending on the machine requirements.
- Each module requires a user supplied +24V DC control voltage. This can be supplied to the line module and then distributed to the motor modules via a bus system.

Sinamics S120 booksize format - Industry Mall

Hinweis: Weiterführende Informationen finden Sie in Modul *Anhang A*.

Notes:

S220 Smart Line Modules (SLM).

- Smart Line Modules in the power ranges 16 kW / 24 kW / 36kW / 44kW are available.
- Available in two versions (C type and D type)
- C type: Optimized for continuous load with up to 200% overload (continuous motion)
- D type: Optimized for highly dynamic, intermittent duty cycles with up to 300% overload (discontinuous motion)
- Mixed operation of SINAMICS S220 SLM and S120 Motor Modules with the Sinumerik ONE Control Units (SINAMICS Firmware V5.2 SP3 HF10 and higher) is possible. (16 kW / 24kW / 36kW / 55kW D type, 24 kW C type)

WARNING

Danger to life due to unsuitable firmware version

Operating SINAMICS S220 booksize components with an unsuitable firmware version can result in unsafe operating states.

For example, it can lead to converter components overheating, a motor unexpectedly accelerating or safety functions not functioning.

- When using a SINAMICS S220 Smart Line Module, use a Control Unit with firmware V5.2 SP3 HF10 or later.
- When using a SINAMICS S220 Motor Module, use a Control Unit with firmware V5.2 SP3 HF9 or later.

Additional S220 Information

Valuable space in the control cabinet is saved due to a reduction in module width of up to 50% and reduced module height due to the integrated fan.

The motor modules are capable of overload capacities of up to 300%.

The motor modules are now even more stable.
This means even easier handling thanks to interference-resistant connections and faster, simpler wiring in an even smaller space.

Fan replacement and maintenance remains straightforward.

Note: Detailed Sinamics component information can be found in module **A003** and **Appendix A.**

Notes:

NX10.3/NX15.3 Interface Locations.

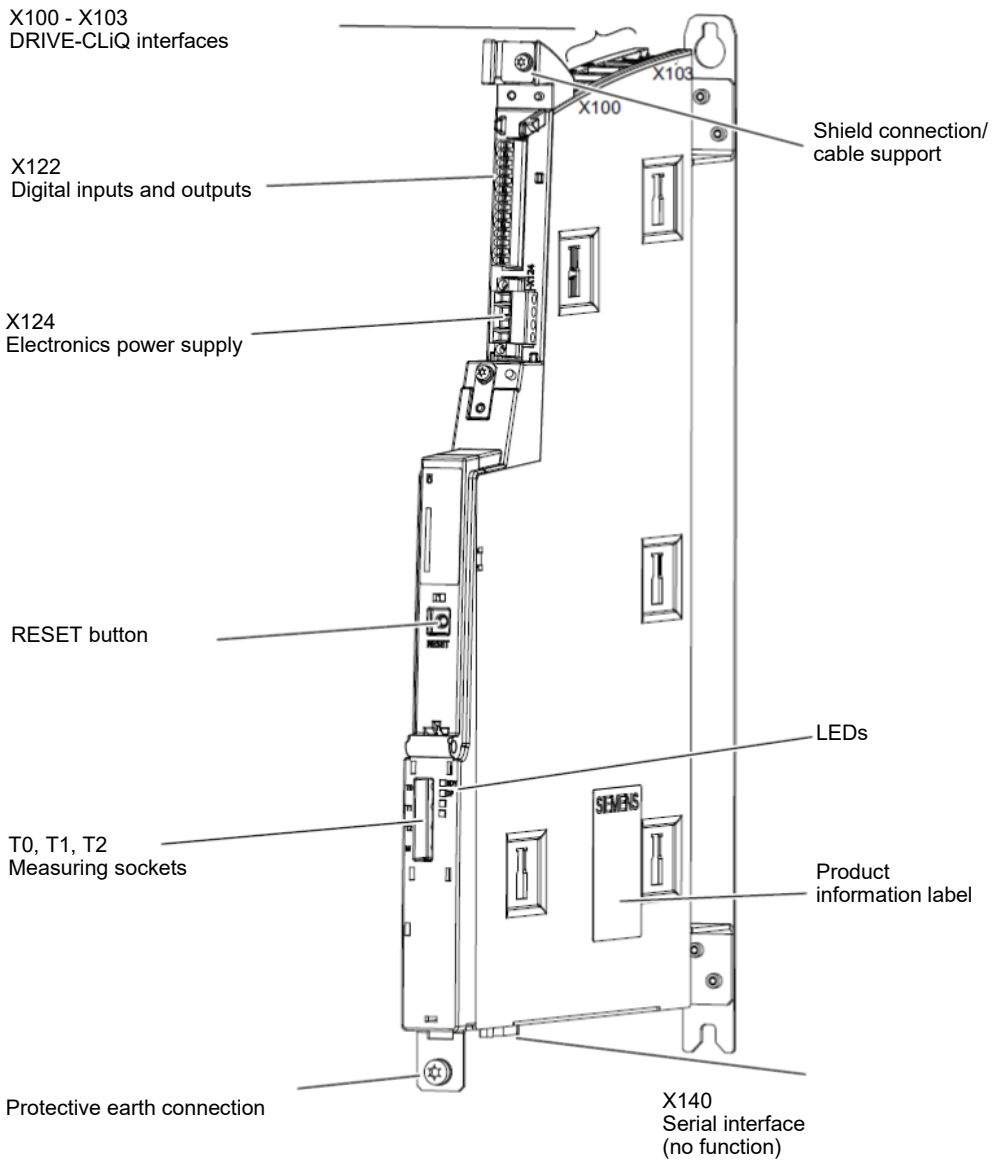
SINAMICS Numeric Control Extension NX15.3 Drive control extension for SINUMERIK ONE

NX10.3 6SL3040-1NC00-0AA0 Drive control extension for up to 3 axes

NX15.3 6SL3040-1NB00-0AA0 Drive control extension for up to 6 axes

If more than 6 axes are required it will be necessary to use NX modules.

NX modules can also be used to facilitate high speed spindles, for example, where the drive cycle time of the NX can be reduced to improve performance.



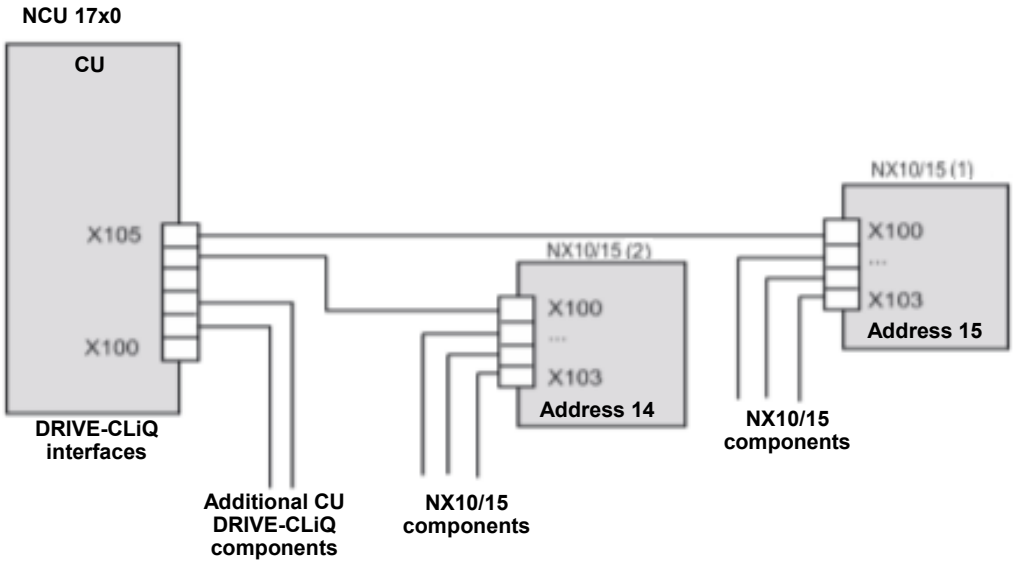
Notes:

DRIVE CLiQ Topology: Connections and Rules.

DRIVE-CLiQ topology: Drive components are connected to the Control Unit via DRIVE-CLiQ.

The following rules apply for the wiring:

- Only one star topology is permitted between the NX and the Control Unit. This means that only one NX can be operated per DRIVE-CLiQ port on a Control Unit.
- DRIVE-CLiQ ports not assigned to NX can be wired to other DRIVE-CLiQ components.



- Once an NX has been connected and configured (TIA PLC Hardware Configuration) you cannot simply insert it into a different DRIVE-CLiQ port, as the addresses of the integrated drives are set permanently from the point of view of the PLC.

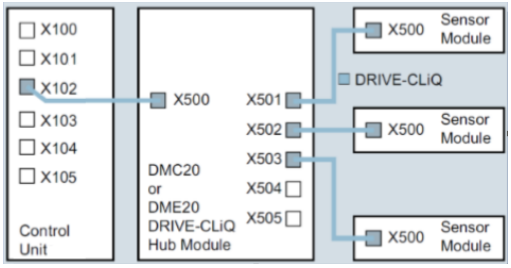
The following table illustrates this relationship:

DRIVE-CLiQ Port	Drive PROFIBUS addresses
X105	15
X104	14
X103	13
X102	12
X101	11

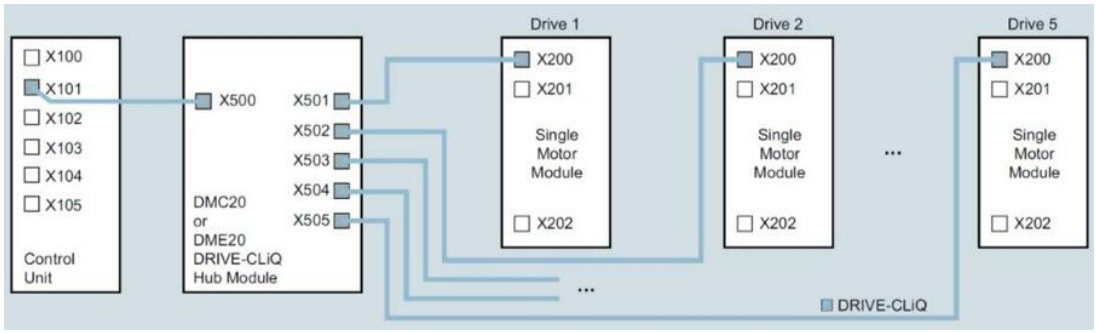
Notes:

DRVECLiQ Hub Connection Examples:

Expanding upon the number of available DRIVE-CLiQ interfaces, for example, when there are multiple additional drive components such as second measuring circuits etc.



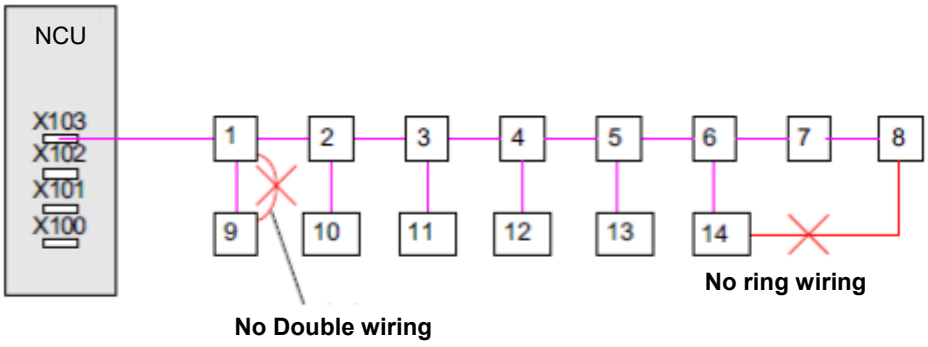
Facilitating the easy removal/parking of optional drive objects and components without disturbing the DIVECLiQ connections to remaining axes and/or reducing the cabling requirements between the electrical cabinet and machine mounted drive components.



Drive CLiQ Rules

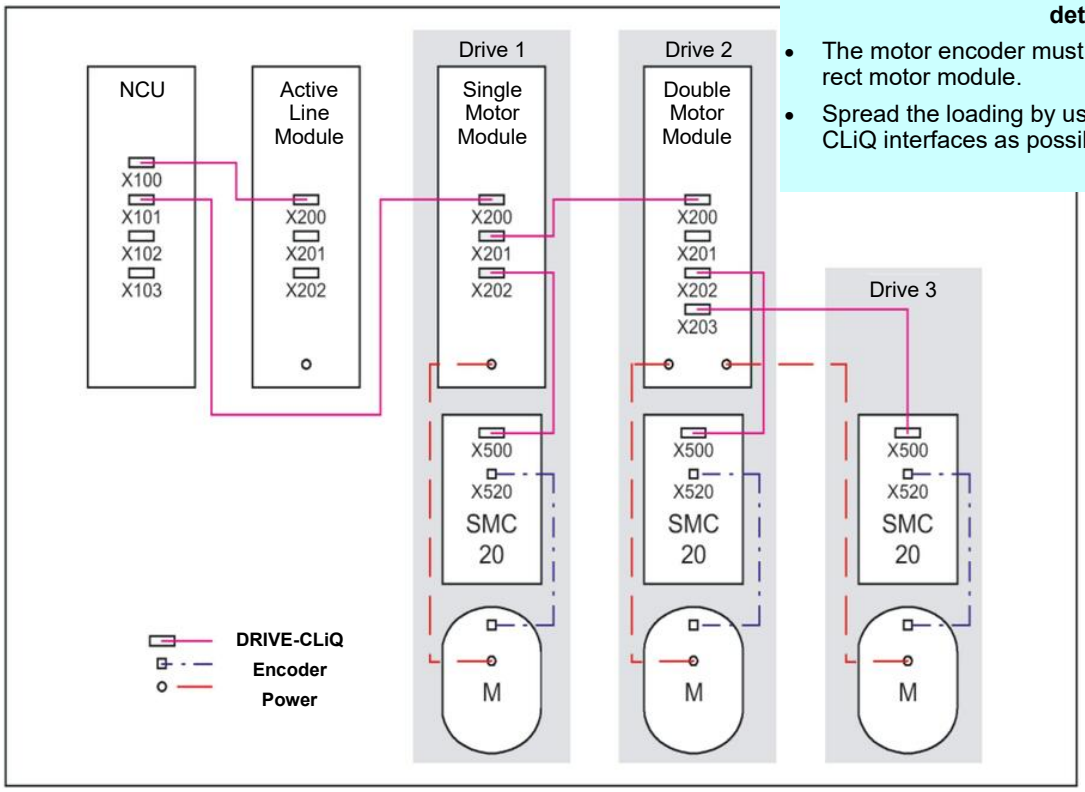
Obligatory Rules for DRIVE-CLiQ:

- **Max. 198 DRIVE-CLiQ participants per NCU**
- **Max. 14 participants on one connector**
- **Max 8 participants in a row**
- **No ring wiring**
- **No Double wiring**



Notes:

Optional Rules for DRIVE-CLiQ



Adherence to these rules allows automatic topology detection.

- The motor encoder must be connected to the correct motor module.
- Spread the loading by using as many of the DRIVE-CLiQ interfaces as possible.

Network Information

It is important to avoid using IP addresses which are used by the system (see the next page for detailed information)

X120 - System network (DHCP server)
Default IP address - 192.168.214.1
Note: Can be changed by the machine builder!

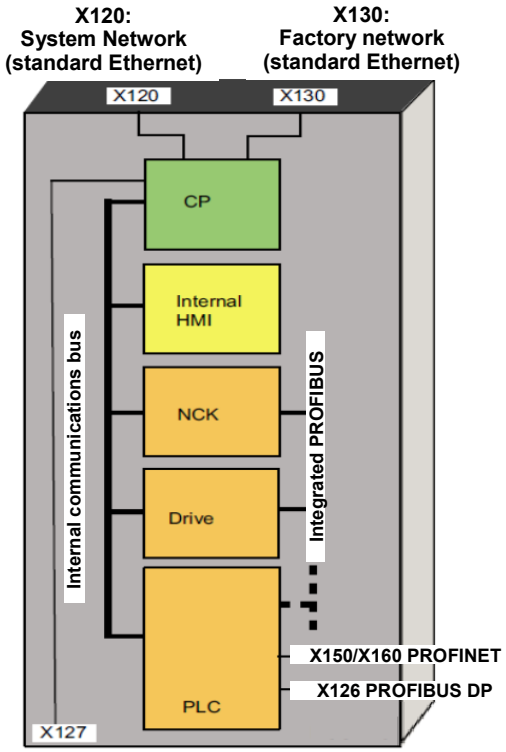
X127 - Service and commissioning network
Default IP address - 192.168.215.1
Subnet mask - 255.255.255.224
The subnet mask reserves IP addresses up to and including 192.168.215.31.
IP addresses above this range are available for use by the user.

X130 - Factory/Customer network (DHCP client or manual IP address)
Usually assigned at the end user and for accessing remote PCs/networks. (avoid IP addresses in use by the system)

X150/X160 - PROFINET interface. Connection of PLC/drive periphery components etc.

X126 - PROFIBUS interface (backwards compatibility). Connection of PLC/drive periphery components etc.

Below - the internal network structure of the SINUMERIK ONE



Notes:

The Sinumerik ONE uses IPv4 IP addresses for the System network (X120), PROFINET network (X150/X160) and the Service/ commissioning (X127) networks.
Any device connected to these networks requires a unique IP address.
Some Ethernet devices are issued IP addresses from a DHCP (Dynamic Host Configuration Protocol) server. These are known as DHCP clients.
It is also possible to connect devices using a fixed/manual IP address.

System Network (X120)

The following table shows the default IP addresses for the system network.
While it is common practise to use the default settings it is also possible to change them.
An IP address must only be used once on the system.

X120 System Network - Default IP: 192.168.214.1 Subnet mask: 255.255.255.0

IP address	Network Station	Remarks
192.168.214.1	NCU on X120	Default DHCP server
192.168.214.2 - 9	Additional NCUs with fixed IP address	Unassigned
192.168.214.10 - 239	Additional ITCs, subsequently for additional IPCs, NCUs, MCPs & MPPs	DHCP clients
192.168.214.241	Fixed IP address for IPC (Connected via ETH2 of IPC)	Default
192.168.214.242 - 249	Additional IPCs with a fixed IP address	Unassigned
192.168.214.250 - 254	For PCs with a fixed IP address (Service connection)	Unassigned

Factory Network (X130) - Can be set as a DHCP client or assigned with a manual IP address.

X130 Factory Network - Default: DHCP client

IP address	Network Device	Remarks
Assigned by DHCP server	Internet, factory network etc.	Default DHCP client
Manual assignment	PCs, factory network	Avoid IP addresses already in use - X120/X150/X160/X127

PROFINET Network (X150/X160)

PROFINET interfaces are used for communicating with periphery devices such as PLC I/O components.
The PROFINET protocol is industry wide and open for the connection of many types of components from manufacturers other than Siemens. This has to be facilitated by the machine manufacturer in the PLC hardware configuration.

The default IP addresses are shown below. These are assigned in the TiA Portal device configuration.
Note: The IP addresses can also be specified by the machine manufacturer.

X150 PROFINET Network - (IRT Isochronous Real Time) Default IP: 192.168.0.1 Subnet mask: 255.255.255.0

IP address	Network Device	Remarks
192.168.0.1	X150 P1/P2 PROFINET interface	Default DHCP server
192.168.0.2 - 255	Additional PROFINET devices	DHCP clients

X160 PROFINET Network - Default IP: 192.168.1.1 Subnet mask: 255.255.255.0

IP address	Network Device	Remarks
192.168.1.1	X160 PROFINET interface	Default DHCP server
192.168.1.2 - 255	Additional PROFINET devices	DHCP clients

Service/Commissioning Network (X127)
X127 is available for the connection of a PC (DHCP client) in order to carry out service/commissioning tasks. Additionally, it is possible to connect devices requiring a fixed IP address.

X127 Service/Commissioning Network - Default 192.168.215.1 Subnet mask: 255.255.255.224

IP address	Network Device	Remarks
192.168.215.1	NCU on X127	Default DHCP server
192.168.215.2 - 23	PCs for service/commissioning tasks	DHCP clients
192.168.215.24 - 31	Devices with fixed IP addresses e.g. modems	Manual setting
192.168.215.32 - 255	Available - can be used for X130 manual IP address	Manual setting

A001: END

Notes:

A001: END

Notes:

003

SINUMERIK

Sinamics Line and Motor Modules

Module Description:

The SINUMERIK ONE and 828D CNC controller product ranges predominantly use ***Sinamics S120 and S220*** drive components for controlling the axis servo motors.

The Sinamics components are connected to the controller via the DRIVECLiQ drive bus.

Additional Sinamics components are also available to enhance the performance of the machine tool.

This module describes the various types of modules and how they are connected to the DRIVECLiQ bus.

A brief description of the modules is also given as well as the correct enabling/disabling sequence.

Detailed information can be accessed via the following hyperlinks:

Click here for S120 Booksize format online information

Click here for S120 Combi format online information

Click here for S220 Booksize format online information

Module objective:

Upon completion of this module you will know how the drive components of the SINUMERIK ONE and 828D CNC
system are integrated and how the power to these components should be applied and removed.

Content:

Overview: Sinamics S120/S220 Line and Motor Modules

Supply and Connection Information

Smart Line Module (SLM)

Active Line Module (ALM)

Motor Module (MM)

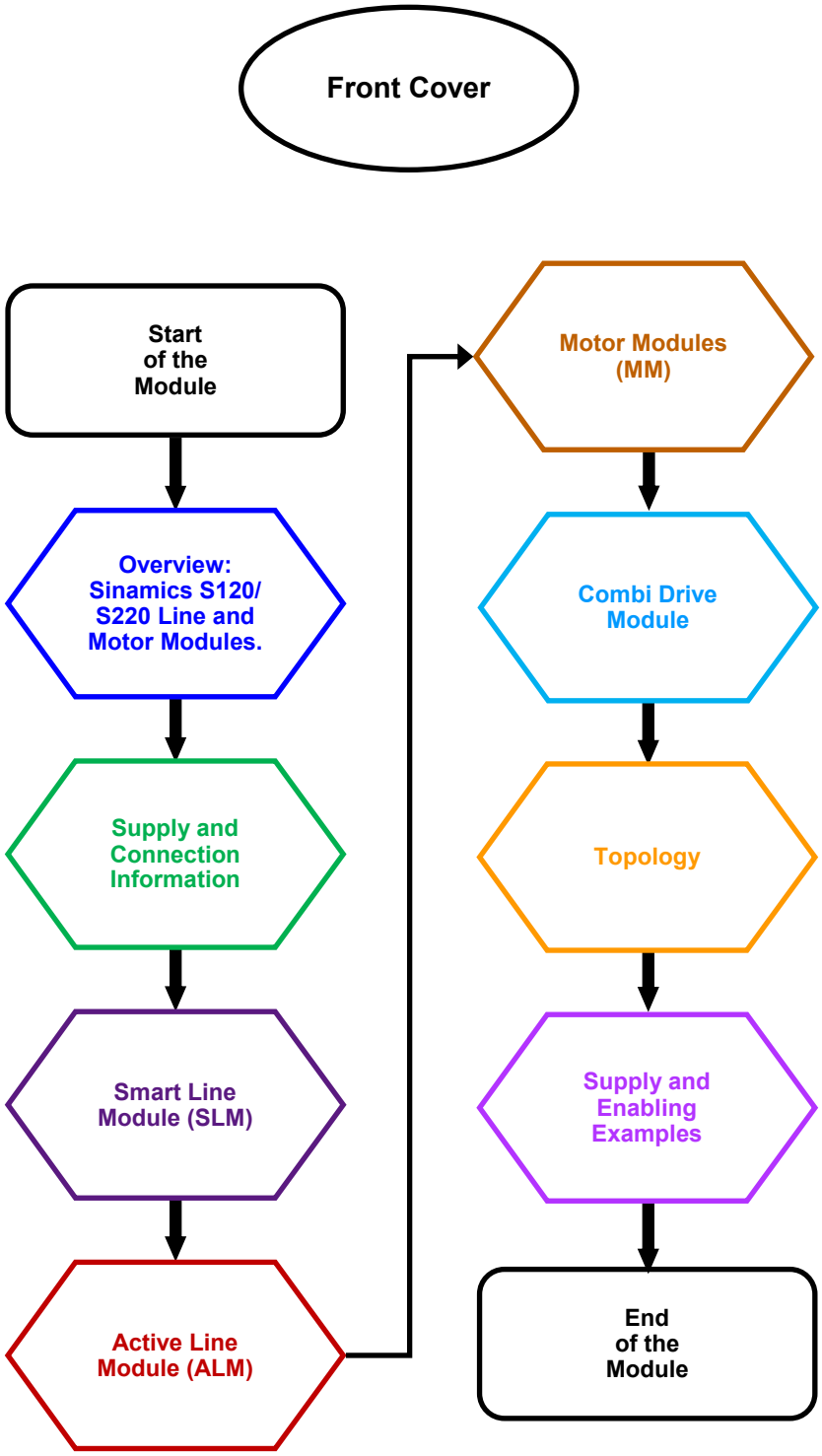
Combi Drive Module

Topology

Supply and Enabling Examples

Sinumerik

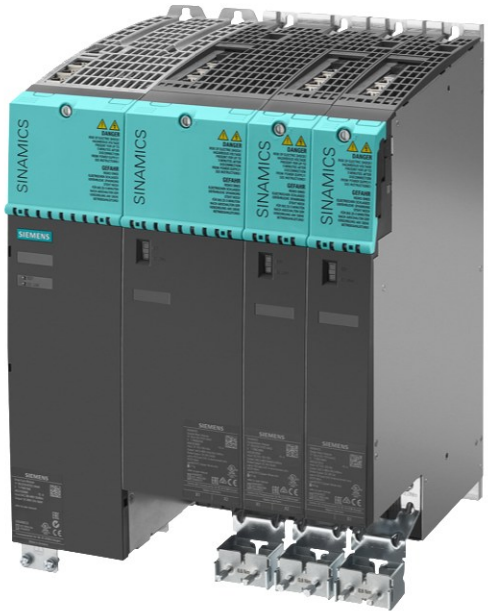
This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Overview: Sinamics S120/ S220 Line and Motor Modules.
Supply and Connection Information
Smart Line Module (SLM)
Active Line Module (ALM)
Motor Modules (MM)
Combi Drive Module
Topology
Supply and Enabling Examples
End of the Module

Overview: Sinamics S120/S220 Line and Motor Modules.

The Sinamics S120 / S220 drive components are the most commonly used with the Sunumerik ONE and 828D
The S120 versions are now being superseded by the S220 versions.

Right: Example, S120 Line module with three motor modules:



S120 Line Module (LM)

S120 Motor Modules (MM)

Right: Example, S220 Line module with three motor modules:



S220 Line Module (LM)

S220 Motor Modules (MM)

Overview: SINAMICS S220 Booksize Smart Line Modules

The SINAMICS S120 Booksize range set the standards for performance and quality in drive technology for machine tools, production machines, and general motion control applications.

With Sinamics SW V5.2, the SINAMICS S120 Booksize system has reached full maturity and is now set to make an evolutionary leap.

The proven servo drive system, SINAMICS S120, continues under the umbrella of the innovative SINAMICS S220 with new motor modules for the Booksize variants.

The new motor modules represent space savings, increased overload capacity, stability, and efficiency.

New innovations have been developed on the SINAMICS S220 platform, which offers a unique architecture for future requirements as well as more powerful features compared to the proven SINAMICS S120.

Note: The S120/S220 line and motor modules are physically very similar.
Descriptions shown in this document will only use one of the module types.
Any differences between the variants will be highlighted at the relevant place.

S120 Booksize format online information

S220 Booksize format online information

General Sinamics S120 / S220 Information

- Line and motor modules supply power to the AC servomotors.
- All Motor Modules have to be connected to the Drive-CLiQ bus system.
- All Line Modules, with the exception of 5 & 10Kw SLM modules, have to be connected to the Drive-CLiQ bus system.
- Line modules require a 3 phase AC supply. This is rectified into a DC supply which is then supplied to the motor modules via the DC link bus bars.
- Motor modules convert the DC link into a controllable AC supply which is used to drive the servo motors.
- By regulating the frequency and voltage of the AC supply to the servo motors it is possible to precisely control the motor
- Line and motor modules are available in different variants depending on the machine requirements.
- Each module requires a user supplied +24V DC control voltage. This can be supplied to the line module and then distributed to the motor modules via a bus system.

Note: The S120/S220 line and motor modules are physically very similar.
Descriptions shown in this document will only use one of the module types.

Notes:

S220 Smart Line Modules (SLM).

- Smart Line Modules in the power ranges 16 kW / 24 kW / 36kW / 44kW are available.
- Available in two versions (C type and D type)
- C type: Optimized for continuous load with up to 200% overload (continuous motion)
- D type: Optimized for highly dynamic, intermittent duty cycles with up to 300% overload (discontinuous motion)
- Mixed operation of SINAMICS S220 SLM and S120 Motor Modules with the Sinumerik ONE Control Units (SINAMICS Firmware V5.2 SP3 HF10 and higher) is possible. (16 kW / 24kW / 36kW / 55kW D type, 24 kW C type)

WARNING

Danger to life due to unsuitable firmware version

Operating SINAMICS S220 booksize components with an unsuitable firmware version can result in unsafe operating states.

For example, it can lead to converter components overheating, a motor unexpectedly accelerating or safety functions not functioning.

- When using a SINAMICS S220 Smart Line Module, use a Control Unit with firmware V5.2 SP3 HF10 or later.
- When using a SINAMICS S220 Motor Module, use a Control Unit with firmware V5.2 SP3 HF9 or later.

Additional S220 Information

Valuable space in the control cabinet is saved due to a reduction in module width of up to 50% and reduced module height due to the integrated fan.

The motor modules are capable of overload capacities of up to 300%.

The motor modules are now even more stable.
This means even easier handling thanks to interference-resistant connections and faster, simpler wiring in an even smaller space.

Fan replacement and maintenance remains straightforward.

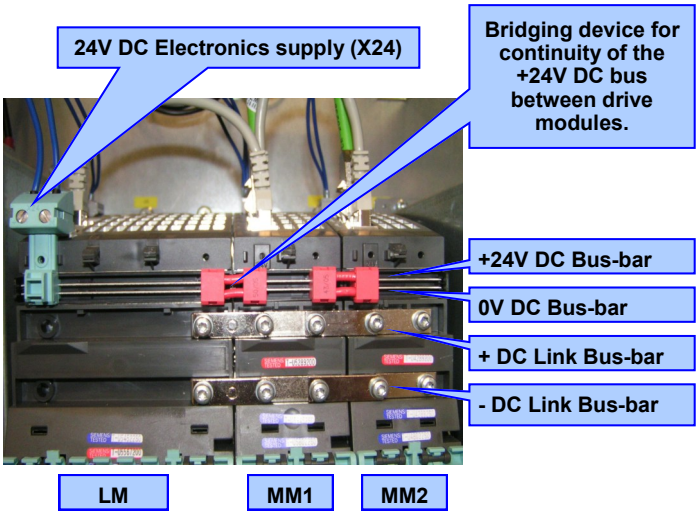
Supply and Connection Information.

The drive modules require two external power supplies in order to operate:

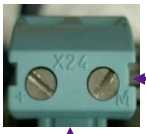
- 3 phase AC main power supply (380V to 480V ±10% (-15% < 1 min) - to the bottom of the line module is used to produce 600V DC link supply.
  The DC link is distributed to the motor modules via a bus bar system on the front of the drive modules beneath a flip down cover.

- An electronics supply (+24V DC) is also required.
  This is supplied to the Line Module via connector X24, and is distributed via a bus bar system to the remaining drive modules.
  The electronics supply is used internally by the drive modules for powering of the electronic circuitry etc.
  Externally it is used for supplying the Drive-CLiQ encoders and fault/ready/enable signals.

Location of the electronics supply and DC link bus bars:



X24– Electronic Supply 24V DC



Notes:

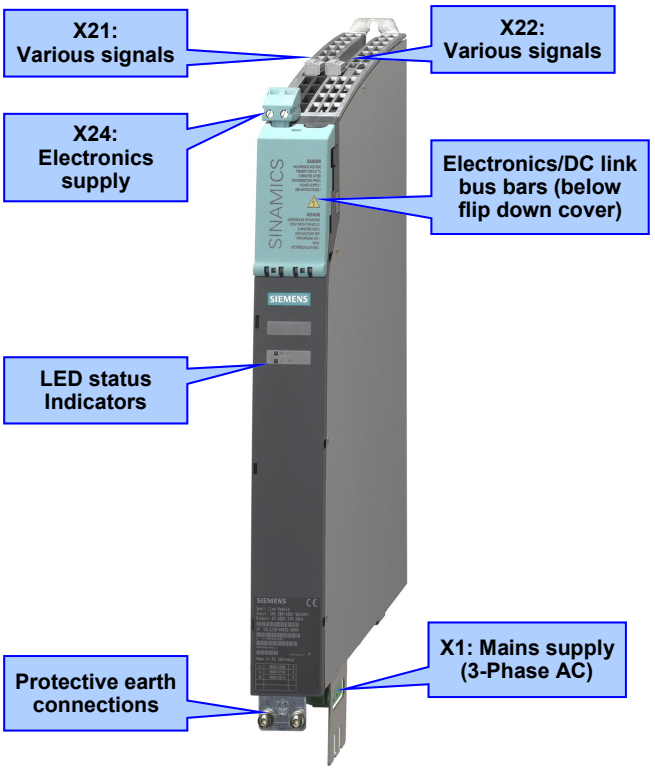
Smart Line Modules (SLM)

- SLMs rectify the AC mains supply into the DC link voltage. This is then utilised by the Motor Modules.
- The DC link voltage is non-stabilised on all SLM types.
- The SLM is capable of regenerating the current caused by decelerating motors.
- The regenerative current is fed back into the mains network. This eliminates the need for braking resistors and other measures for heat dissipation.
- The SLM is available in 5, 10, 16, 36 & 55Kw variants.
- The 5 & 10Kw SLMs are not used on the Drive-CLiQ network. Therefore, there are slight differences when compared to DriveCLiQ type modules.

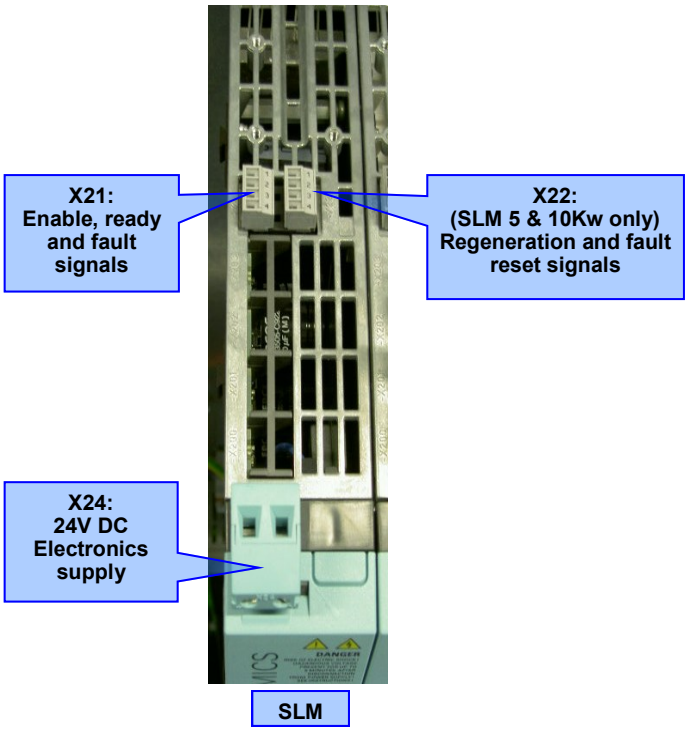


Booksize format - Smart Line Modules

SLM 5 & 10Kw (S120 nloy) Interface locations



S120 SLM 5 & 10Kw Connector locations X21, X22 & X24





S120 LM 5 & 10Kw X22 Terminal description and functionality

Terminal	Function	Description
1	+24V DC (Output)	+24 V DC Supply for use with the functions of X22.2 & 3 described below.
2	Disable Regeneration (Input)	Applying +24V DC will cause the regeneration function to be disabled. In some cases it may be necessary to stop the regeneration of current caused by braking motors. Example:- A poor quality mains Network not able to handle regeneration. Note:-The fitting of resistor modules to dissipate excess energy will be required when this function is used.
3	Fault Reset (Input)	Applying +24V DC will cause the resetting of any drive faults. A fault occurring on the drives system will be latched, this can be reset with this function providing the cause of the original fault is no longer present.
4	0V DC (Ground)	Must be connected to the same 0V of the electronics supply (X24).

Notes:

S120 SLM 5 & 10Kw X21 Terminal description and functionality:



Terminal	Function	Description
1	Ready Signal (Output)	Switches high (+24V DC) when the following conditions are true: • Electronics supply present (X24). • DC Link is pre-charged. • Pulses enabled (X21.3) • No over-temperature • No over-current It is recommended that the signal is connected directly to the 828D controller via connector X122 pin 1. This is the OFF1 enable signal.
2	Pre-warning (Output)	Switches low when any of the following conditions arise: • Over-temperature threshold reached. • I^2t threshold reached. • Mains fault making regenerative feedback impossible. If feedback is intentionally deactivated (X22.2 is high) this warning is disabled. This signal is for use by the manufacturer to set a user alarm and alert the operator to a problem.
3	Enable Pulses (Input)	Applying +24V DC will cause the pulses to be enabled. (Current consumption:10mA) This signal must be removed at least 10ms prior to the 3-phase mains supply being removed. Failure to do this can lead to damaged drive modules. This is especially important when the axes/spindles are decelerating and causing regenerative currents. if the mains circuit is opened these regenerative currents remain within the drive system causing damage to the modules. It is recommended that this terminal is connected via an early break contact on the machine isolator switch and the line contactor (If used). If the terminal is used by the manufacturer it must be ensured that it is controlled in the suggested manner. When used in this way the transistors will be switched off and mains regeneration halted, the motors will then coast to a halt. This is not suitable for an emergency stop situation.
4	Enable Pulses (Ground)	Must be connected to the 0V of the enable pulses supply.

S120 SLM 5 & 10Kw LED Status Display (Non DriveCLiQ line module)

LED	Colour	Status	Explanation	Remedy
READY (Top)	Green	On	Module is healthy and operational.	
	Orange	On	Pre charging of DC link active; Bypass relay dropped out.	
	Red	On	Over temperature/current fault or electronics power supply outside permitted range.	Diagnose fault (Output X21.2) Acknowledge fault (Input X22.3)
DC LINK (Bottom)		Off	Electronics power supply outside permitted range.	
	Orange	On	DC link voltage within tolerance.	
	Red	On	DC link voltage outside tolerance.	Check the mains supply

Notes:

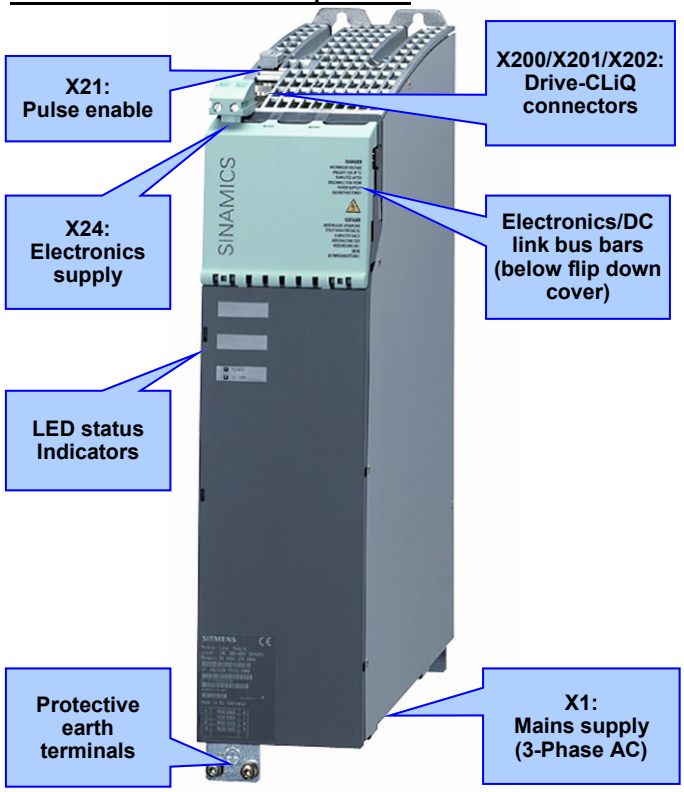
DRIVECLiQ Smart Line Modules (SLM)

S120: 36Kw & 55Kw Booksize & 16Kw Booksize Compact
S220: 16Kw & 24Kw (C & D types)

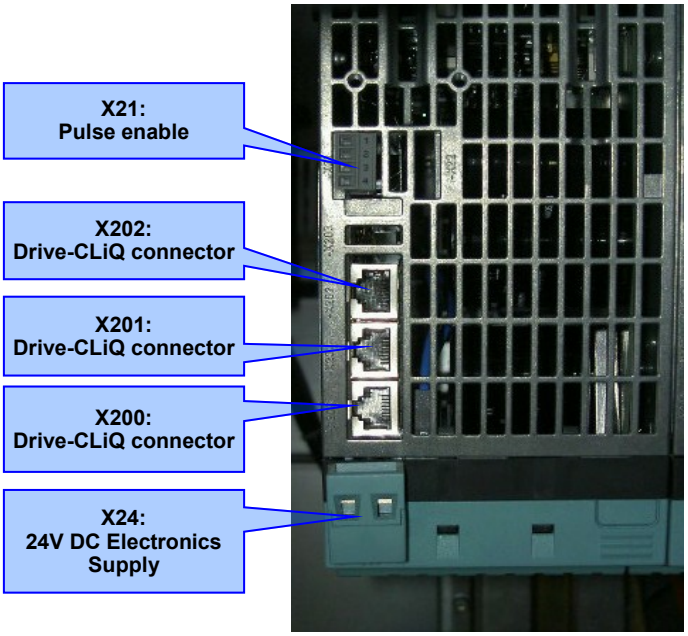
SLMs rectify the AC mains supply into the DC link voltage. This is then utilised by the Motor Modules.

- Available power rating: 16Kw/24Kw/55Kw
- These models are always required to be connected to the Drive-CLiQ drive bus network.
- There are slight differences when compared to non-DriveCLiQ type modules.
- The DC link voltage is non-stabilised.
- The SLM is capable of regenerating the current caused by decelerating motors.
- The regenerative current is fed back into the mains network. This eliminates the need for braking resistors and other measures for heat dissipation.
- Regeneration can be deactivated by parameter P3533-Inhibit regenerative operation.
- Available with internal or external cooling capability.

Interface Locations - example S120



Smart Line Module Top Interfaces - example S120



X21 Terminal description and functionality

S120 S220



Terminal	Function	Description
1	Not used	Not used
2	Not used	Not used
3	Enable Pulses (Input)	Applying +24V DC enables the pulses. (Current consumption:10mA) This signal must be removed at least 10ms prior to the 3-phase mains supply being removed. Failure to do this can lead to damaged drive modules. This is especially important when the axes/spindles are regenerating. If the mains circuit is opened in this state, regenerative currents remain within the drive system causing damage to the modules. It is recommended that this terminal is connected via an early break contact on the machine isolator switch and the line contactor (If used). If the terminal is used by the manufacturer it must be ensured that it is controlled in the suggested manner. When used in this way the transistors will be switched off and mains regeneration halted, the motors will then coast to a halt. This is not suitable for an emergency stop situation.
4	Enable Pulses (Ground)	Must be connected to the 0V of the enable pulses supply.

Notes:

DRIVECLiQ Smart Line Modules LED Status Display

LED	Colour	Status	Explanation	Remedy
READY (Top)		Off	Electronics power supply outside permitted range.	Check 24V DC supply to X24.
	Green	On	Module configured and ready, DriveCLiQ communication is active.	
	Orange	On	DriveCLiQ communication is being established/module not configured.	Check alarms on controller.
	Red	On	At least one fault is present.	Check alarms on controller.
	Red	Flashing @ 0.5Hz	Firmware download procedure is running.	
	Red	Flashing @ 2Hz	Firmware download procedure is complete.	Power off/on.
	Green/ Orange or Red/ Orange	Flashing @ 2Hz	Component recognition is active (p0124 = 1)	
DC LINK (Bottom)		Off	Electronics power supply outside permitted range.	Check 24V DC supply to X24.
	Orange	On	DC link voltage within tolerance-Only if unit is configured and ready.	
	Red	On	DC link voltage outside tolerance-Only if unit is configured and ready.	Check 3-phase supply.

Smart Line Module part number information (with hyperlinks)

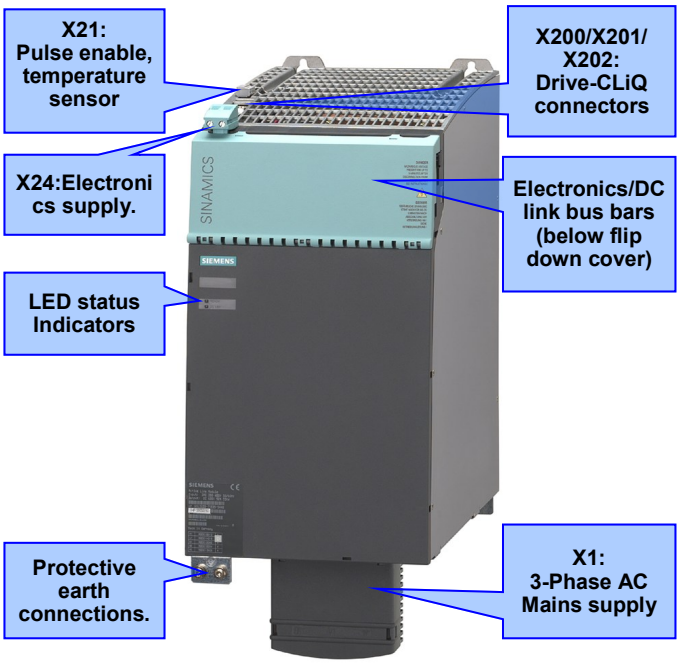
S120/S220 Smart Line Modules			
Rating @ 380v	**Not C or D rated**	**Book sized C Types**	**Book sized D Types**
5Kw	**6SL3130-6AE15-0AB1** (S120 Booksize)	-	-
10Kw	**6SL3130-6AE21-0AB1** (S120 Booksize)	-	-
16Kw	**6SL3430-6TE21-6AA1** (S120 Booksize Compact)	**6SL5130-6UE21-6AC0** (S220 Booksize)	**6SL5130-6UE21-6AD0** (S220 Booksize)
24Kw	-	**6SL5130-6UE22-4AC0** (S220 Booksize)	**6SL5130-6UE22-4AD0** (S220 Booksize)
36Kw	**6SL3130-6TE23-6AA3** (S120 Booksize)	**6SL5130-6UE22-6AC0** (S220 Booksize)	**6SL5130-6UE23-6AD0** (S220 Booksize)
55Kw	**6SL3130-6TE25-5AA3** (S120 Booksize)	**6SL5130-6UE25-5AC0** (S220 Booksize)	**6SL5130-6UE25-5AD0** (S220 Booksize)

Notes:

Active Line Modules (ALM) S120 only

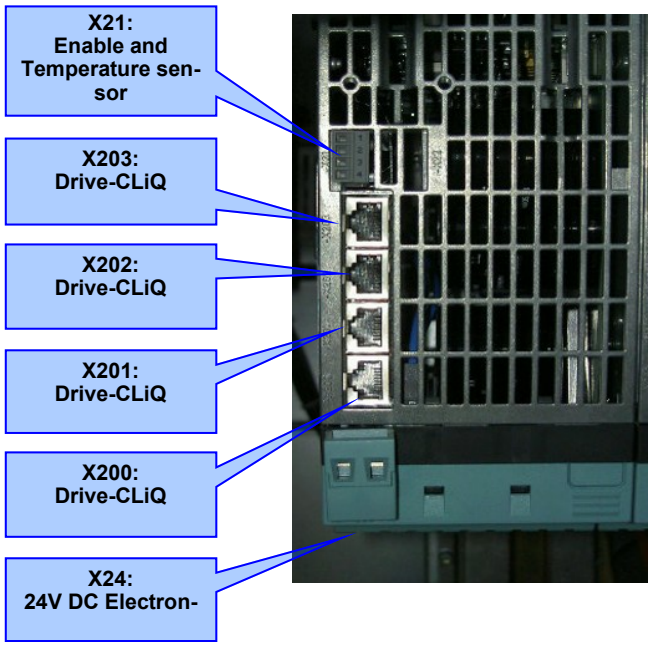
- ALMs rectify the AC mains supply into the DC link voltage which is then utilised by the Motor Modules.
- The DC link voltage is also stabilised, unlike the SLM modules.
- The stabilised DC link makes the ALM suitable for high dynamic machinery where the acceleration and positioning of axes/spindles is critical.
- The ALM is capable of mains regeneration of the current caused by decelerating motors.
- The regenerative current is fed back into the mains network. This eliminates the need for braking resistors and complex measures for heat dissipation.
- Regeneration can be deactivated by parameter P3533- Inhibit regenerative operation.
- The modules are available in 16, 36, 55, 80 & 120Kw variants with internal or external cooling capability.
- The ALM is always connected to the Drive-CLiQ bus and has four Drive-CLiQ connections.

Terminal locations of ALM modules:-



Booksize format - Active Line Modules

Active Line Module Terminal Description:





X21 Terminal description and functionality

Terminal	Function	Description
1	+ Temp (Input)	Temperature sensor input from Active Interface Module (Observe the polarity)
2	- Temp (Input)	Temperature sensor input from Active Interface Module (Observe the polarity)
3	Enable Pulses (Input)	Applying +24V DC enables the pulses. (Current consumption:10mA) This signal must be removed at least 10ms prior to the 3-phase mains supply being removed. Failure to do this can lead to damaged drive modules. This is especially important when the axes/ spindles are regenerating. If the mains circuit is opened in this state, regenerative currents remain within the drive system causing damage to the modules. It is recommended that this terminal is connected via an early break contact on the machine isolator switch and the line contactor (If used). If the terminal is used by the manufacturer it must be ensured that it is controlled in the recommended manner. When used in this way the transistors will be switched off and mains regeneration halted; the motors will then coast to a halt. This is not suitable for an emergency stop situation.
4	Enable Pulses (Ground)	Must be connected to the 0V of the enable pulses supply.

Notes:

S120 Active Line Modules LED Status Display

LED	Colour	Status	Explanation	Remedy
READY (Top)		Off	Electronics power supply outside permitted range.	Check 24V supply to X24.
	Green	On	Module configured and ready, DriveCLiQ communication is active.	
	Orange	On	DriveCLiQ communication is being established/module not configured.	Check alarms on controller.
	Red	On	At least one fault is present.	Check alarms on controller.
	Red	Flashing @ 0.5Hz	Firmware download procedure is running.	
	Red	Flashing @ 2Hz	Firmware download procedure is complete	Power off/on.
	Green/Orange or Red/Orange	Flashing @ 2 Hz	Component recognition is active (p0124 = 1)	
DC LINK (Bottom)		Off	Electronics power supply outside permitted range.	Check 24V supply to X24.
	Orange	On	DC link voltage within tolerance-Only if unit is configured and ready.	
	Red	On	DC link voltage outside tolerance-Only if unit is configured and ready.	Check the mains input.

Active Line Module part number information (with hyperlinks)

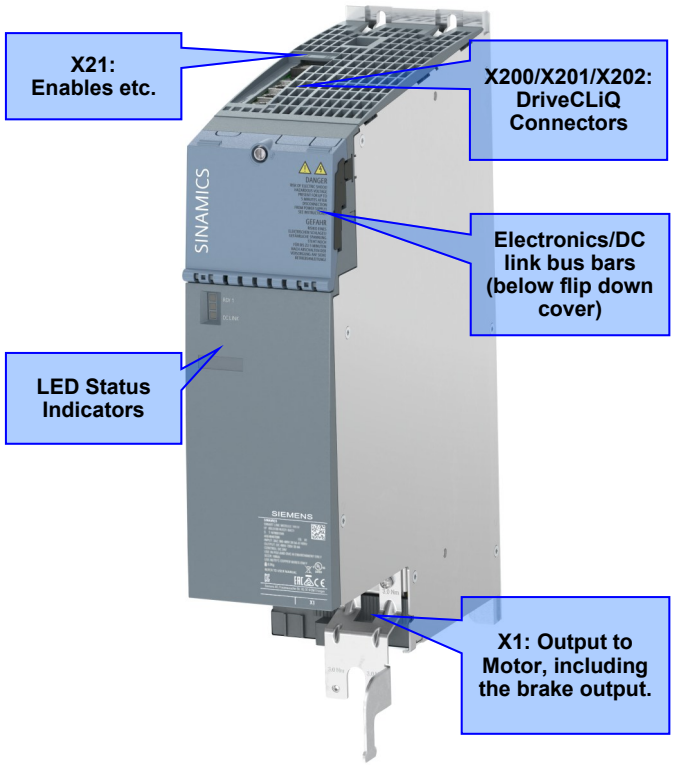
S120 Active Line Modules	
Rating	Part number
16Kw	6SL3130-7TE21-6AA4
36Kw	6SL3130-7TE23-6AA3
55Kw	6SL3130-7TE25-5AA3
80Kw	6SL3130-7TE28-0AA3
200Kw	6SL3130-7TE31-2AA3

Notes:

Motor Modules (MM)

- Motor Modules convert the DC link voltage into a variable AC voltage in order to drive the motors.
- The modules must be connected to the Drive-CLiQ bus.
- Single/double axis and internally/externally cooled versions are available.
- Double axis modules are equipped with four Drive-CLiQ connectors, one more than the single axis modules.

Example:- A single axis ***S220*** MM showing connector locations etc.



Single Axis MM
Available in the following rated current output sizes:
3, 5, 9, 18, 30, 45, 60, 85, 132 & 200A.

S120 Booksize Single Motor Modules

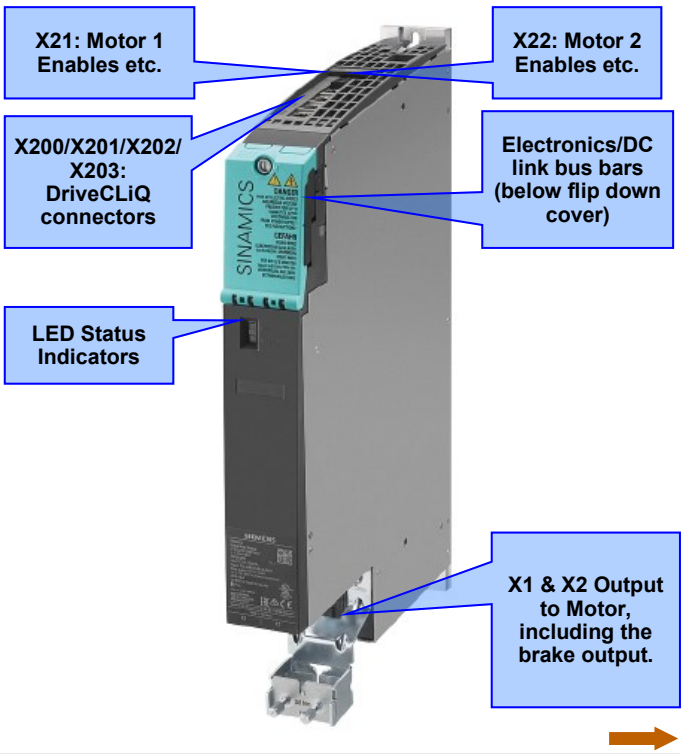
S220 Booksize Single Motor Modules

S120 Booksize Compact Single Motor Modules

Double Axis MM
Available in the following rated current output sizes: **1.7**,**3, 5, 9 & 18A**

These units will have two identical Motor/Brake connectors (X1 & X2) and two identical enable connectors (X21 & X22).

Example:-A double axis ***S120*** MM showing connector locations.



S120 Booksize Double Motor Modules

S220 Booksize Double Motor Modules

S120 Booksize Compact Double Motor Modules

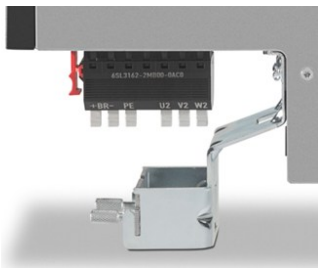
Notes:

Book sized Motor Module: Motor power connectors



Motor plug connectors

- for Double Motor Modules located next to one another
- Shielding integrated in the Motor Module

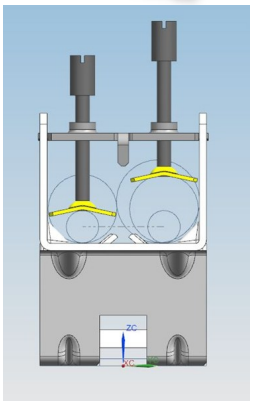


- Hose clip no longer required for self made cabling
- Plug connector with latching mechanism
- PE is integrated

Integrated shield concept



Shield connecting terminal can be removed



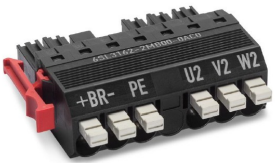
Allows cable sheath diameters from 7mm to 19mm (up to 23mm Single Motor Modules)



- Fast, simple wiring

Motor plug connectors that customers can fabricate/assemble themselves from 1.5mm² (AWG16) to 6mm²
Prefabricated MOTION-CONNET power-cables to 10mm² (AWG 8)

Push-in plug connector



6SL3162-2MB00-0AC0

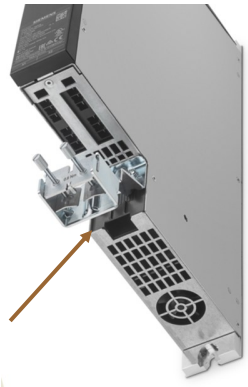
Plug connector screw connection



6SL3162-2MA00-0AC0

Book sized Motor Module: Fan Replacement

- Fan can be replaced in the installed state



Fan with monitoring - supported from SINAMICS firmware V4.8

Can be removed towards the front

Fan module



Fan part numbers:

6SL3162-0AN00-0AA0 (50mm)

6SL3162-0AP00-0AA0 (100mm)

6SL3162-0AH00-0AA1 (200mm)

6SL3162-0AS00-0AA0 (24 A C/D type)

Notes:

Motor Module Ratings and Part Numbers

C type: Optimized for continuous load with up to 200% overload (continuous motion)

D type: Optimized for highly dynamic, intermittent duty cycles with up to 300% overload (discontinuous motion)

S120 Book-sized C and D type Motor Module Information - Internally cooled				Cold plate cooling	
Output	Type	Book sized C Types	Book sized D Types	No C or D rating	Width
3A	Single	-	6SL3120-1TE13-0AD0	6SL3126-1TE13-0AA4	50mm
5A	Single	-	6SL3120-1TE15-0AD0	6SL3126-1TE15-0AA4	50mm
9A	Single	-	6SL3120-1TE21-0AD0	6SL3126-1TE21-0AA4	50mm
18A	Single	6SL3120-1TE21-8AC0	6SL3120-1TE21-8AD0	6SL3126-1TE21-8AA4	100mm
24A	Single	6SL3120-1TE22-4AC0	6SL3120-1TE22-4AD0	-	100mm
30A	Single	6SL3120-1TE23-0AC0	6SL3120-1TE23-0AD0	6SL3126-1TE23-0AA3	100mm
45A	Single	6SL3120-1TE24-5AC0	-	6SL3126-1TE24-5AA3	100mm
60A	Single	6SL3120-1TE26-0AC0	-	6SL3126-1TE26-0AA3	100mm
85A	Single	6SL3120-1TE28-5AA3 (no C/D rating)		6SL3126-1TE28-5AA3	200mm
132A	Single	6SL3120-1TE31-3AA3 (no C/D rating)		6SL3126-1TE31-3AA3	300mm
200A	Single	6SL3120-1TE32-0AA4 (no C/D rating)		6SL3126-1TE32-0AA4	300mm
2 x 3A	Double	-	6SL3120-2TE13-0AD0	6SL3126-2TE13-0AA4	50mm
2 x 5A	Double	-	6SL3120-2TE15-0AD0	6SL3126-2TE15-0AA4	50mm
2 x 9A	Double	-	6SL3120-2TE21-0AD0	6SL3126-2TE21-0AA4	50mm
2 x 18A	Double	6SL3120-2TE21-8AC0	6SL3120-2TE21-8AD0	6SL3126-2TE21-8AA3	100mm

S220 Motor Module Ratings and Part Numbers:

S220 Motor Modules - all internally cooled.				
Output	Type	Book sized C Types	Book sized D Types	Width
24A	Single	-	6SL5120-1UE22-4AD0	50mm
30A	Single	6SL5120-1UE23-0SC0	6SL5120-1UE23-0SD0	50mm
30A	Single	-	6SL5120-1UE23-0AD0	100mm
60A	Single	6SL5120-1UE26-0AC0	-	100mm
85A	Single	6SL5120-1UE28-5AC0	6SL5120-1UE28-5AD0	150mm
132A	Single	6SL5120-1UE31-3AC0	6SL5120-1UE31-3AD0	150mm
200A	Single	6SL5120-1UE32-0AC0	6SL5120-1UE32-0AD0	200mm
2 x 9A	Double	-	6SL5120-2UE21-0AD0	50mm
2 x 18A	Double	6SL5120-2UE21-8SC0	6SL5120-2UE21-8SD0	50mm

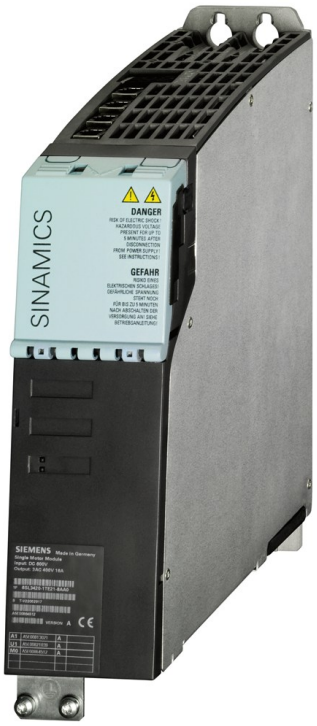
Replacement types for the superceeded booksize Motor Modules:

Output	Type	Previous version	Booksize D-types	Booksize C-types
3A	Single	6SL3120-1TE13-0AA4	6SL3120-1TE13-0AD0	
5A	Single	6SL3120-1TE15-0AA4	6SL3120-1TE15-0AD0	
9A	Single	6SL3120-1TE21-0AA4	6SL3120-1TE21-0AD0	
18A	Single	6SL3120-1TE21-8AA4	6SL3120-1TE21-8AD0	6SL3120-1TE21-8AC0
2 x 3A	Double	6SL3120-2TE13-0AA4	6SL3120-2TE13-0AD0	
2 x 5A	Double	6SL3120-2TE15-0AA4	6SL3120-2TE15-0AD0	
2 x 9A	Double	6SL3120-2TE21-0AA4	6SL3120-2TE21-0AD0	
2 x 18A	Double	6SL3120-2TE21-8AA4	6SL3120-2TE21-8AD0	6SL3120-2TE21-8AC0
30A	Single	6SL3120-1TE23-0AA4	6SL3120-1TE23-0AD0	6SL3120-1TE21-8AC0

Notes:

Motor Modules (MM) S120 Book-sized Compact

The S120 book sized compact motor modules are use to expand the number of drives the Sinamics Combi drive system to the maximum of 6.
The modules are only available with internal cooling.

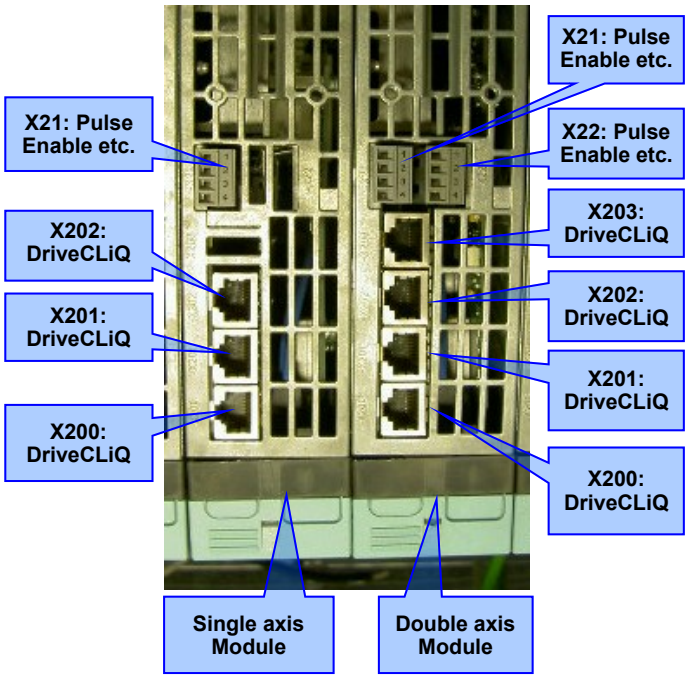


Interfaces:
Single MM
- X1-Output to Motor.
- X21-Enable & temperature sensor input .
- X200, X201, X202-DriveCliq connections.

Double MM
- X1 & X2-Output to Motors.
- X21-Enable & temperature sensor input (Motor 1)
- X22-Enable & temperature sensor input (Motor 2)
- X200, X201, X202, X203-DriveCliq connections.

Single Motor Module	Rated output (I_{rated})
6SL3420-1TE13-0AA1	3A
6SL3420-1TE15-0AA1	5A
6SL3420-1TE21-0AA1	9A
6SL3420-1TE21-8AA1	18A
Double Motor Module	
6SL3420-2TE11-7AA1	2 x 1.7A
6SL3420-2TE13-0AA1	2 x 3A
6SL3420-2TE15-0AA1	2 x 9A

Motor Module Terminal Descriptions:





Connector X21 & X22 (X22 only for double axis modules).

Terminal	Function	Description
1	+ Temp (Input)	Positive temperature sensor input from non-Drive-CLiQ temperature sensor. Type:- KTY84-1C130/PTC
2	- Temp (Input)	Negative temperature sensor input from non Drive-CLiQ temperature sensor. Type:- KTY84-1C130/PTC
3	Enable Pulses (EP) Input	Applying +24V DC will cause the pulses of the individual axis to be enabled. (Current consumption:10mA) When the signal is low, ***Safe pulse suppression*** is activated. Note:-No electrical isolation occurs between the Motor module and the motor. This enable signal is only operational if the Safety Integrated function ***Safe Standstill (SH)*** is enabled. This is a dual channel monitoring system of the Control Unit and Motor Module. The status of the Enable Pulses input is cross checked against the corresponding CU input, an error is reported if they do not match.
4	EP (Ground)	Must be connected to the 0V of the enable pulses supply.

Notes:

S120/S220 Motor Modules LED Status Display

LED	Colour	Status	Explanation	Remedy
READY (Top)		Off	Electronics power supply outside permitted range.	Check 24V DC on X24.
	Green	On	Module configured and ready, DriveCLiQ communication is active.	
	Orange	On	DriveCLiQ communication is being established/module not configured.	Check alarms on controller.
	Red	On	At least one fault is present.	Check alarms on controller.
	Red	Flashing @ 0.5Hz	Firmware download procedure is running.	
	Red	Flashing @ 2Hz	Firmware download procedure is complete.	Power drives OFF/ON.
	Green/ Orange or Red/ Orange	Flashing @ 2Hz	Component recognition is active (p0124 = 1)	
DC LINK (Bottom)		Off	Electronics power supply outside permitted range.	Check 24V DC on X24.
	Orange	On	DC link voltage within tolerance-Only if unit is configured and ready.	
	Red	On	DC link voltage outside tolerance-Only if unit is configured and ready.	Check mains input.

Notes:

S120 Combi Drive Modules (Combined Line and Motor Modules)

The Combi drive unit consists of a Line Module (SLM) and either 3 or 4 Motor Modules, the first of which is dedicated to the Spindle.
They must always be connected to the DriveCLiQ network via controller interface.
The number of axes can be increased by adding Sinamics S120 Book-sized Compact Motor Modules up to the maximum allowed by the system.

Combi Drive - Frame type A:

Separate fan and additional DC link components via busbars.

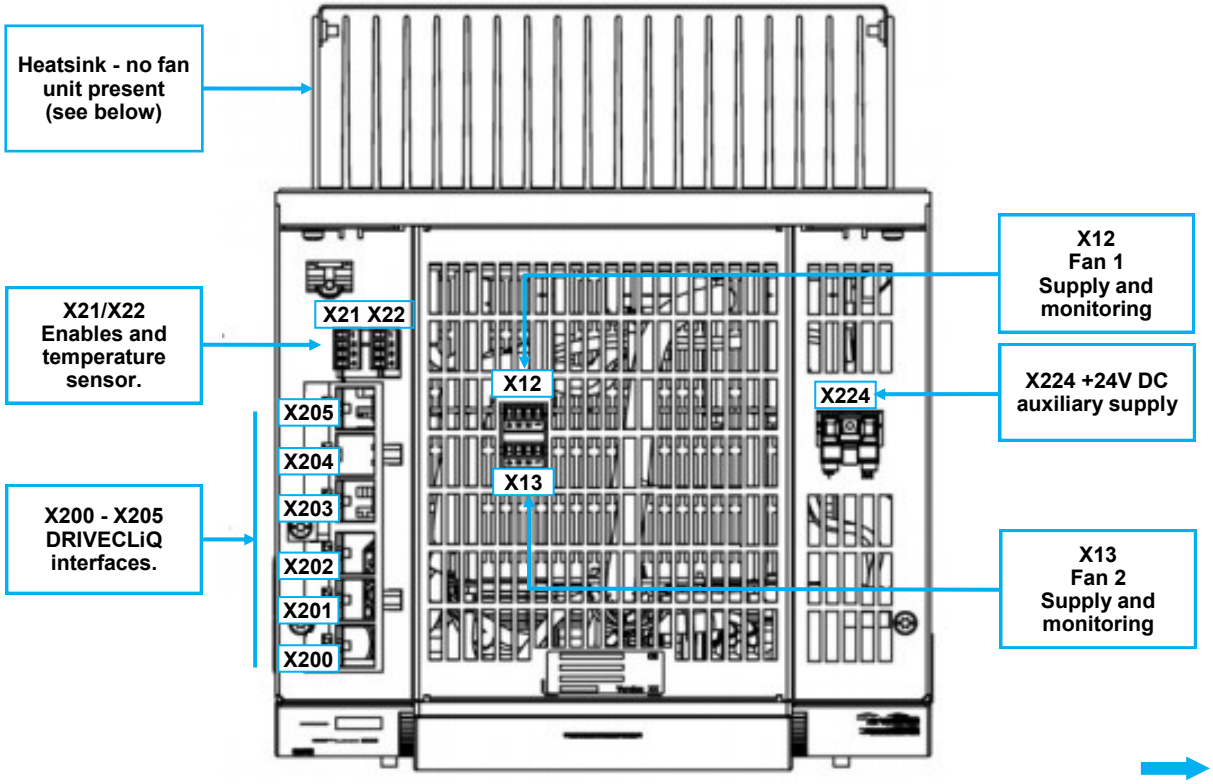


Combi Drive - Frame type B:

Integrated fan (inside the heatsink) and additional DC link components via cable.

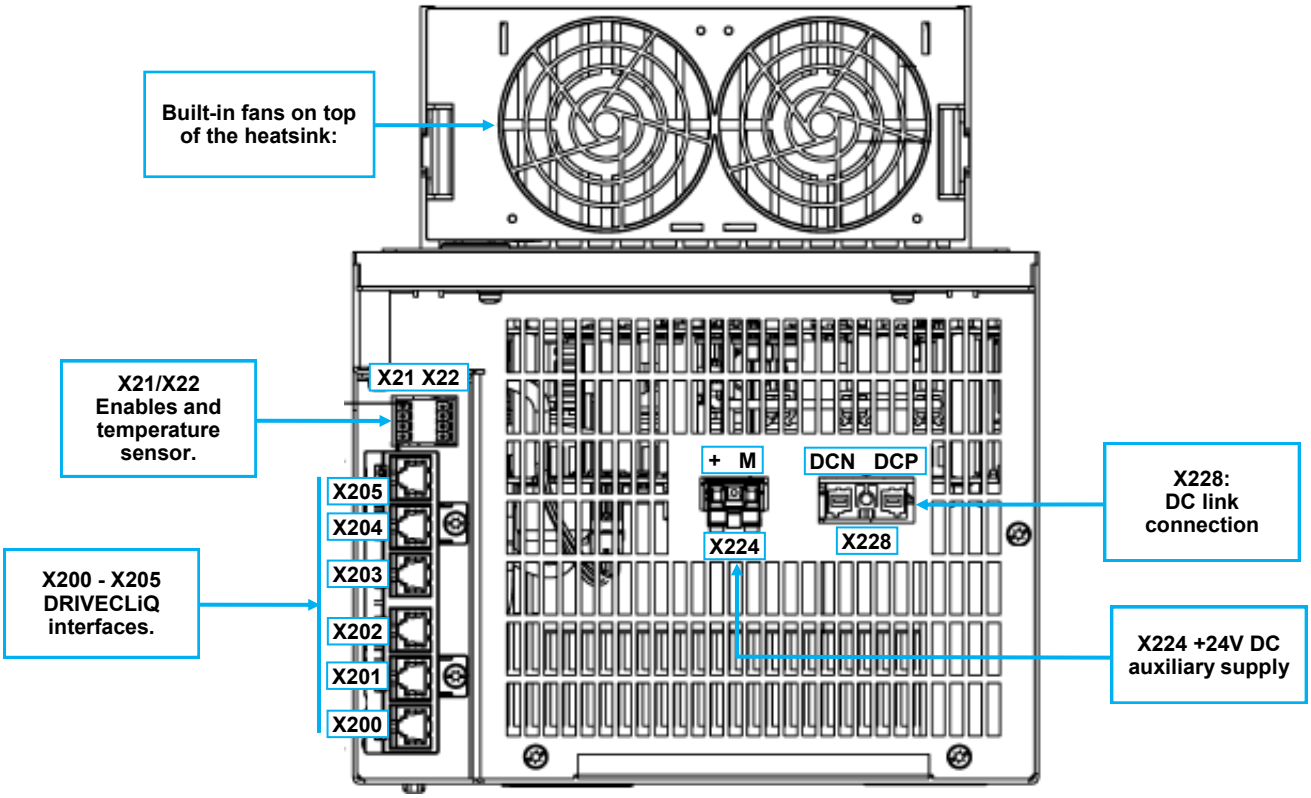


Terminal Description - top view of ***type A*** Combi Drive Unit:



Notes:

Terminal Description - top view of ***type B*** Combi Drive Unit:



X21 Terminal Description - types A and B Combi Drives:

Terminal	Function	Description
1	Not used	Not used
2	Not used	Not used
3	Enable Pulses (Input)	Applying +24V DC enables the pulses. (Current consumption:10mA) This signal must be removed at least 10ms prior to the 3-phase mains supply being removed. Failure to do this can lead to damaged drive modules. This is especially important when the axes/spindles are regenerating. If the mains circuit is opened in this state, regenerative currents remain within the drive system causing damage to the modules. It is recommended that this terminal is connected via an early break contact on the machine isolator switch and the line contactor (If used). If the terminal is used by the manufacturer it must be ensured that it is controlled in the suggested manner. When used in this way the transistors will be switched off and mains regeneration halted, the motors will then coast to a halt. This is not suitable for an emergency stop situation.
4	Enable Pulses (Ground)	Must be connected to the 0V of the enable pulses supply.

X22 Terminal Description - types A and B Combi Drives:

Terminal	Function	Description
1	+ Temp (Input)	Positive temperature sensor input from non-DriveCLiQ temperature sensor. Type:- KTY84-1C130/PTC
2	- Temp (Input)	Negative temperature sensor input from non DriveCLiQ temperature sensor. Type:- KTY84-1C130/PTC
3	Safety Integrated. Enable Pulses EP +24 V (Input)	Applying +24V DC will cause the pulses of the individual axis to be enabled. (Current consumption:10mA) When the signal is low, ***Safe pulse suppression*** is activated. Note:-No electrical isolation occurs between the Motor module and the motor. This enable signal is only operational if the Safety Integrated function ***Safe Standstill (STO)*** is enabled. This is a dual channel monitoring system of the Control Unit and Motor Module. The status of the Enable Pulses input is cross checked against the corresponding CU input, an error is reported if they do not match.
4	Safety Integrated. Enable Pulses EP M1 (Ground)	Must be connected to the 0V of the enable pulses supply.

Notes:

X224 Terminal Description and functionality (types A and B Combi Drives):

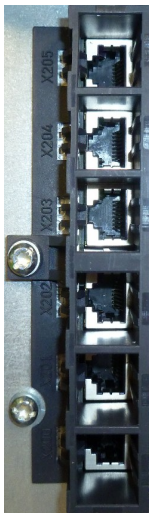


✚ M

Terminal	Function	Description
+	+24 V DC (Input)	+24V DC auxiliary supply to Combi drive module
M	Ground	Ground for fan supply +24V DC auxiliary supply

X200 to X205 Terminal description and functionality (types A and B Combi Drives):

DriveCLiQ Interfaces:



X205 - DMS via SMCxx (If X220 not suitable)

X204 - Z axis motor feedback.

X203 - Y axis motor feedback.

X202 - X axis motor feedback.

X201 - Spindle motor feedback.

X200 - DriveCLiQ drive bus (From CU320-2 PN).

Combi X12/X13 Fan terminal description and functionality (type A Combi only)

Terminal	Function	Description
1	Ground	Ground connection for fan monitoring circuit.
2	Fan Monitoring. (Input)	Fan monitoring signal, input from fan.
3	+24 V DC (Output)	+24V DC supply to fan. Maximum 2 A. (1 x 2 A or 2 x 1A)
4	Ground	Ground for fan supply.

External fan module (fits on top of the heatsink)



Non-Siemens fans may be used without the monitoring. In this case terminals 1& 2 must be linked.

Notes:

X220 Terminal description and functionality - 5V TTL encoders (types A and B Combi Drives):



Pin	Signal	Technical Information
1	+ Temp	Motor temperature sensor positive connection (KTY 84-1C130)
2	Clock	Clock Signal
3	Clock*	Inverse Clock Signal
4	P-Encoder 5V	+5V Encoder Power Supply
5	P-Encoder 5V	
6	P-Sense	+5V Sense Signal
7	M-Encoder (M)	Encoder Power Supply 0V
8	- Temp	Motor temperature sensor negative connection (KTY 84-1C130)
9	M-Sense	0V Sense Signal
10	R	Reference Signal R
11	R*	Inverse Reference Signal R*
12	B*	Inverse Incremental Signal B*
13	B	Incremental Signal B
14	A*	Inverse Incremental Signal A*
15	A	Incremental Signal A

Interfaces X1 to X5



X1 Mains Input-Terminal Description and Functionality:

Terminal	Function	Description
U1	U Phase	Line Phase U
V1	V Phase	Line Phase V
W1	W Phase	Line Phase W
PE	Earth	Protective Earth

X2 to X5 Motor Supply-Terminal Description and Functionality:

Terminal	Function	Description
U2	U Phase	Motor Phase U
V2	V Phase	Motor Phase V
W2	W Phase	Motor Phase W
PE	Earth	Protective Earth

Combi Drive LED Status (types A and B Combi Drives):

LED	Colour	Status	Explanation	Remedy
READY (Top)		Off	Electronics power supply outside permitted range.	Check +24V supply (X24).
	Green	On	Module configured and ready, DriveCLiQ communication is active.	
	Orange	On	DriveCLiQ communication is being established/module not configured.	Check alarms on controller.
	Red	On	At least one fault is present.	Check alarms on controller.
	Red	Flashing @ 0.5Hz	Firmware download procedure is running.	
	Red	Flashing @ 2Hz	Firmware download procedure is complete.	Power drives OFF/ON.
	Green/Orange or Red/Orange	Flashing @ 2Hz	Component recognition is active (p0124 = 1)	
DC LINK (Bottom)		Off	Electronics power supply outside permitted range.	Check +24V supply (X24).
	Orange	On	DC link voltage within tolerance-Only if unit is configured and ready.	
	Red	On	DC link voltage outside tolerance-Only if unit is configured and ready.	Check mains input. (X1)

Notes:

Sinamics S120 Combi - available power modules and rating

SINAMICS S120 Combi Power Module						
Rated power Infeed kW	Rated output current Spindle A	Rated output current Feed 1 A	Rated output current Feed 2 A	Rated output current Feed 3 A	Article No.	Frame type
3-axis Powe4– axis Power Module						
16	18	9	9	–	**6SL3111-3VE21-6FA2**	A01
16	24	9	9	–	**6SL3111-3VE21-6EA1**	A02
20	30	9	9	–	**6SL3111-3VE22-0HA1**	A03
4-axis Powe4– axis Power Module						
10	24 1)	12	12	12	**6SL3111-4VE21-0EA1**	A07
16	18	9	9	9	**6SL3111-4VE21-6FA2**	A04
16	24	9	9	9	**6SL3111-4VE21-6EA1**	A05
16	24	12	9	9	**6SL3111-4VE21-6EC1**	B01
16	24	12	9	9	**6SL3111-4VE21-6ED1**	B02
20	30	12	9	9	**6SL3111-4VE22-0HA1**	A06

Notes:

Topology

General information:

The Control Unit for the drives (**CU**) is contained within the PPU/PU.

The CU has 3 DriveCLiQ interfaces:
X100: Line Modules (Not 5 & 10 Kw types) & Motor Modules
Motors, Encoders, Sensor Modules (SMCxx, SMExx) & Hubs (DMCxx, DMExx) **via** Line Modules/Motor Modules.
X101: Sensor Modules, DriveCLiQ Hubs.
X102: Numerical Extension Module. (NX10)

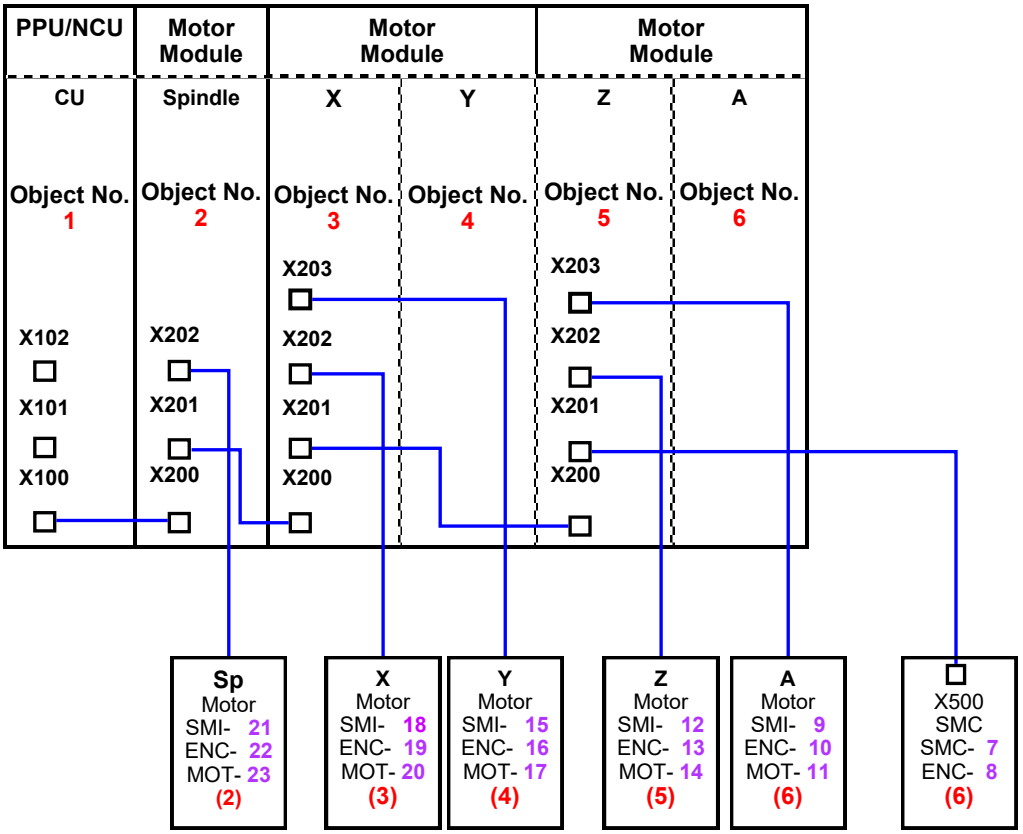
Sinamics drive hardware is classed as Drive Objects and Drive Components:
Drive Objects: CU, NX10, Line modules, Motor Modules.
Drive Components: Motors, Encoders, Sensor Modules, DriveCLiQ Hubs.

During the **automatic topology recognition (fast commissioning)** procedure, each drive object/component is automatically assigned a number for identification purposes.
The CU is always Drive Object No. 1.
Drive Objects connected to the X100 DriveCLiQ bus are the first to be assigned numbers.
The numbers are assigned sequentially, in the order that they are connected on the DriveCLiQ bus.
Drive components are then numbered, sequentially, in reverse order, continuing on from the last drive object.
Hardware connected to X101/X102 is also issued numbers which follow on from the last object/component connected to interface X100.

Example: 5 Axis System (Spindle, X, Y, Z, A)

Objects are numbered in **red**.
Components are numbered in **purple**.

Training equipment drive topology:



Notes:

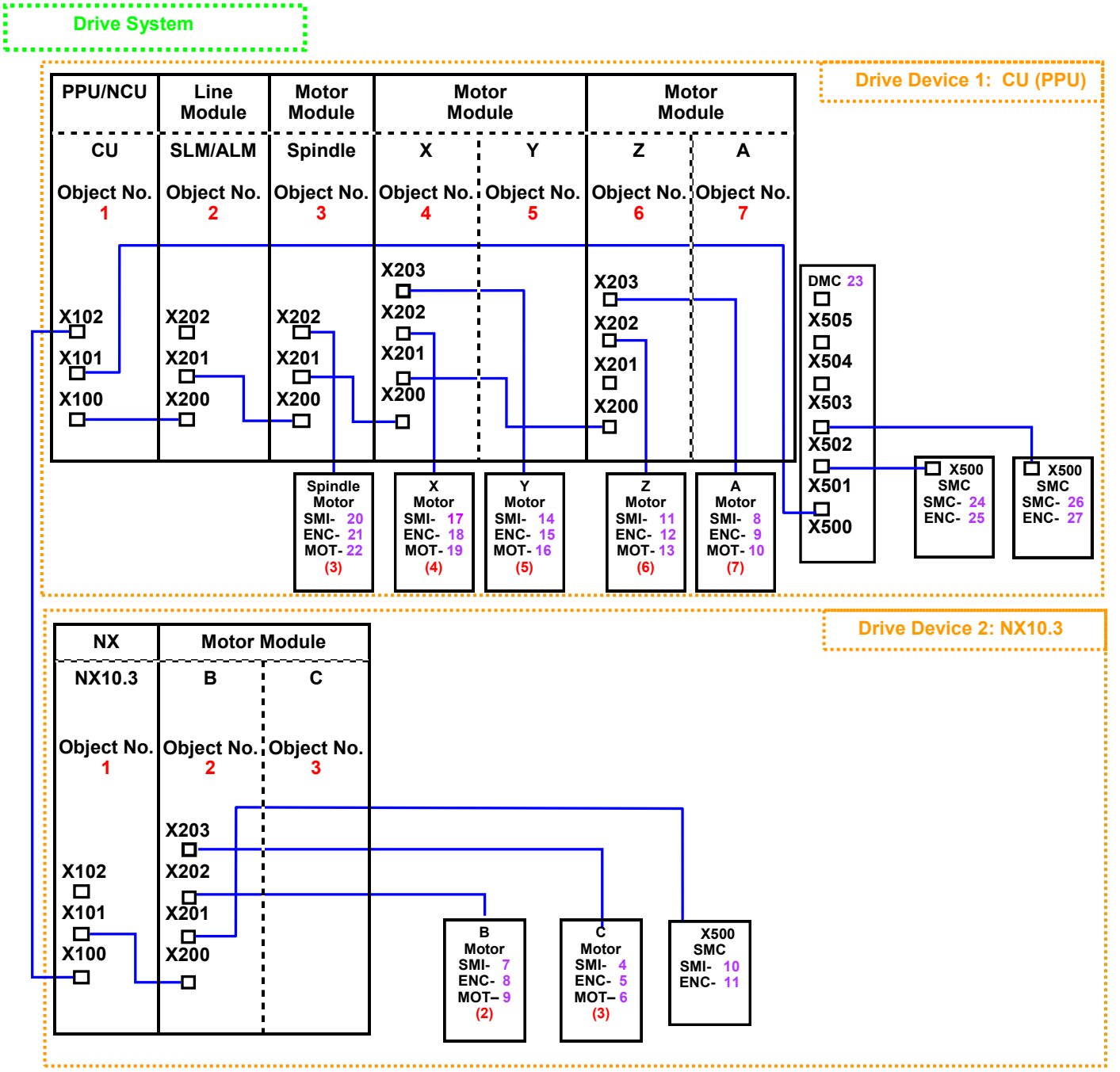
Topology with NX

NX10 modules can be used to increase the number of drives to beyond the six available with the CU.

Equipment connected to X100 & X101 is assigned to the CU and is numbered in the same manner described previously (Page 5)

The NX10 must always be connected to X102.
Like the CU, the NX10 is always Drive Object number 1. This is possible because they are classed as different drive devices.
Drive objects and components connected to the NX10 are issued numbers in the same way as with the CU.

Example- 7 axis system with NX10

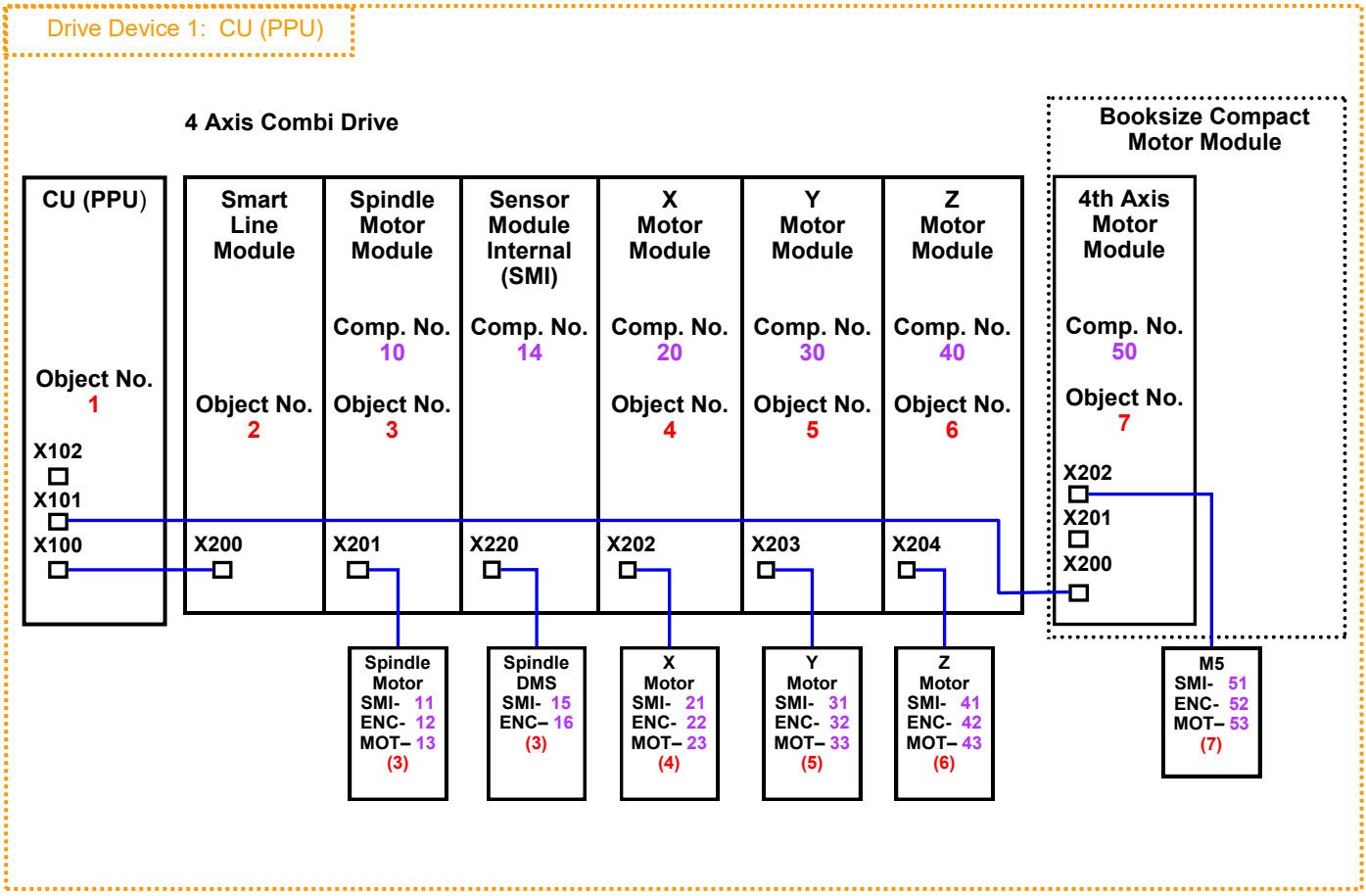


Notes:

Combi Drive Topology

The Combi drive can only be connected to X100.
Additional axes are facilitated by the addition of Book-sized Compact Modules. These must always be connected to X101.
X102 is reserved for the connection of a Terminal Module (TM54) which is used for Safety Integrated functionality.

Example – 5 axis system with Combi Drive:



Notes:

Supply and Enabling Examples

For trouble free operation it is essential that the drives are powered on/off and enabled correctly.

If the mains supply switch/contactor is opened while the drive pulses are enabled and the drive is in regeneration mode, the regenerative current which is normally fed back into the mains will have no where to go, except that is into the motor and line modules, this can result in damage to these modules.

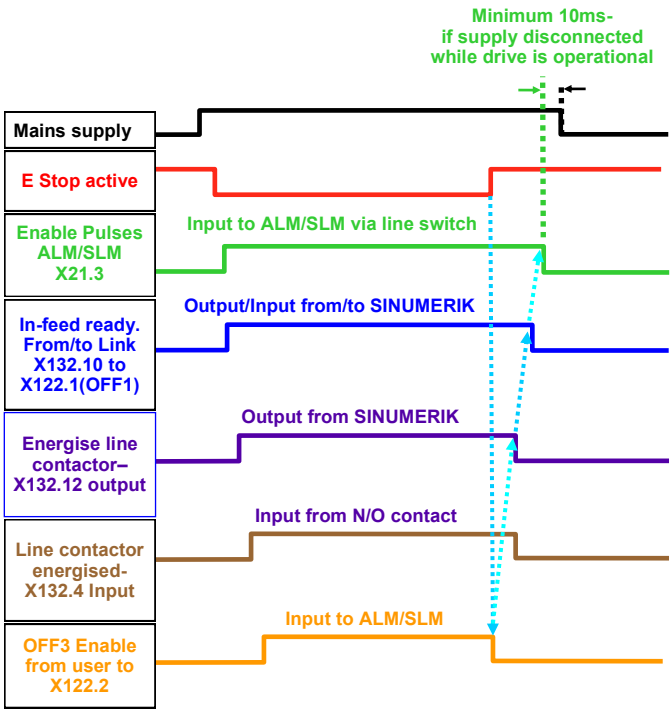
The solution is to use an auxiliary contact fitted to the mains switch and drive line contactor.

This auxiliary contact must open at least 10ms before the main contacts open.

The contact is wired to the Pulse Enable terminal of the SLM/ ALM module.

If the Main switch or contactor is operated while the drive is in operation the pulses will be disabled and the motors will free wheel to a standstill. (Except when the Safety Integrated function ***Safe Stop*** is enabled)

Any potentially harmful current will be unable to enter the drive modules as the transistors are now switched off.



Right: the recommended timing diagram for the enabling of drives via the controller using a line Contactor.

Hyperlinks to the various example connection diagrams

828D PU272.5 Mains Supply and Enabling Examples:

SLM 5 & 10Kw (not DRIVECLiQ) - no line contactor

SLM 5 & 10Kw (not DRIVECLiQ) - user controlled line contactor

ALM and SLM ≥ 36Kw (DRIVECLiQ) - no line contactor

ALM and SLM ≥16Kw (DRIVECLiQ) - user controlled line contactor

ALM & SLM ≥16Kw (DRIVECLiQ) - SINUMERIK controlled line contactor

828D PPU271.4 Mains Supply and Enabling Examples:

SLM 5 & 10Kw(not DRIVECLiQ) - no line contactor

SLM 5 & 10Kw (not DRIVECLiQ) user controlled line contactor

ALM & SLM (≥16Kw) (DRIVECLiQ) - no line contactor

ALM & SLM (≥16Kw) (DRIVECLiQ) - user controlled line contactor

ALM & SLM (≥16Kw) (DRIVECLiQ) - SINUMERIK controlled line contactor

SINUMERIK ONE Mains Supply and Enabling Examples:

SLM 5 & 10Kw (not DRIVECLiQ) - no line contactor

SLM 5 & 10Kw (not DRIVECLiQ) - user controlled line contactor

ALM & SLM (≥16Kw) (DRIVECLiQ) - no line contactor

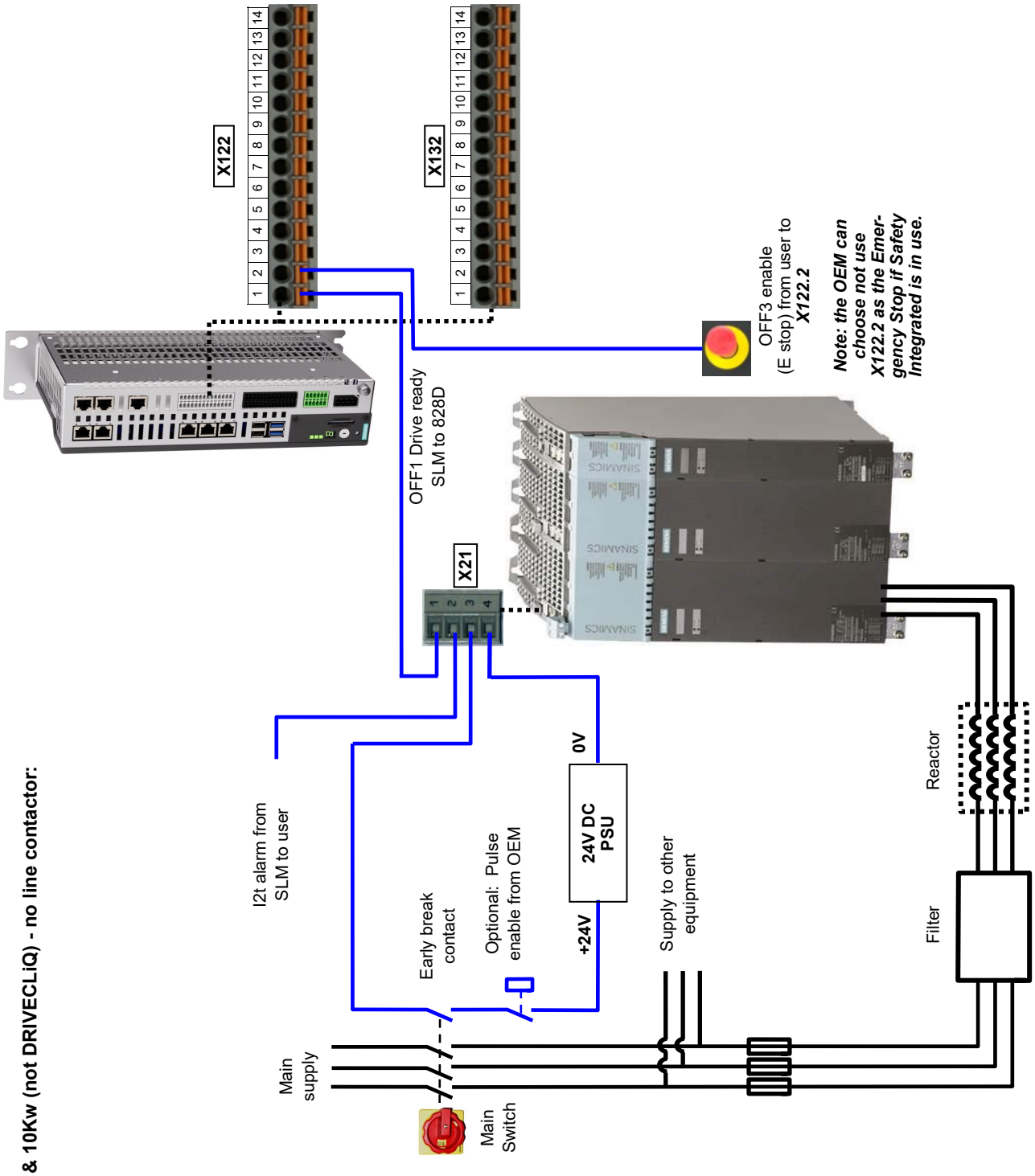
ALM & SLM (≥16Kw) (DRIVECLiQ) - user controlled line contactor

ALM & SLM (≥16Kw) (DRIVECLiQ) - SINUMERIK controlled line contactor

Notes:

828D PU272.5 Supply and Enabling Examples:

SLM 5 & 10Kw (not DRIVECLiQ) - no line contactor:



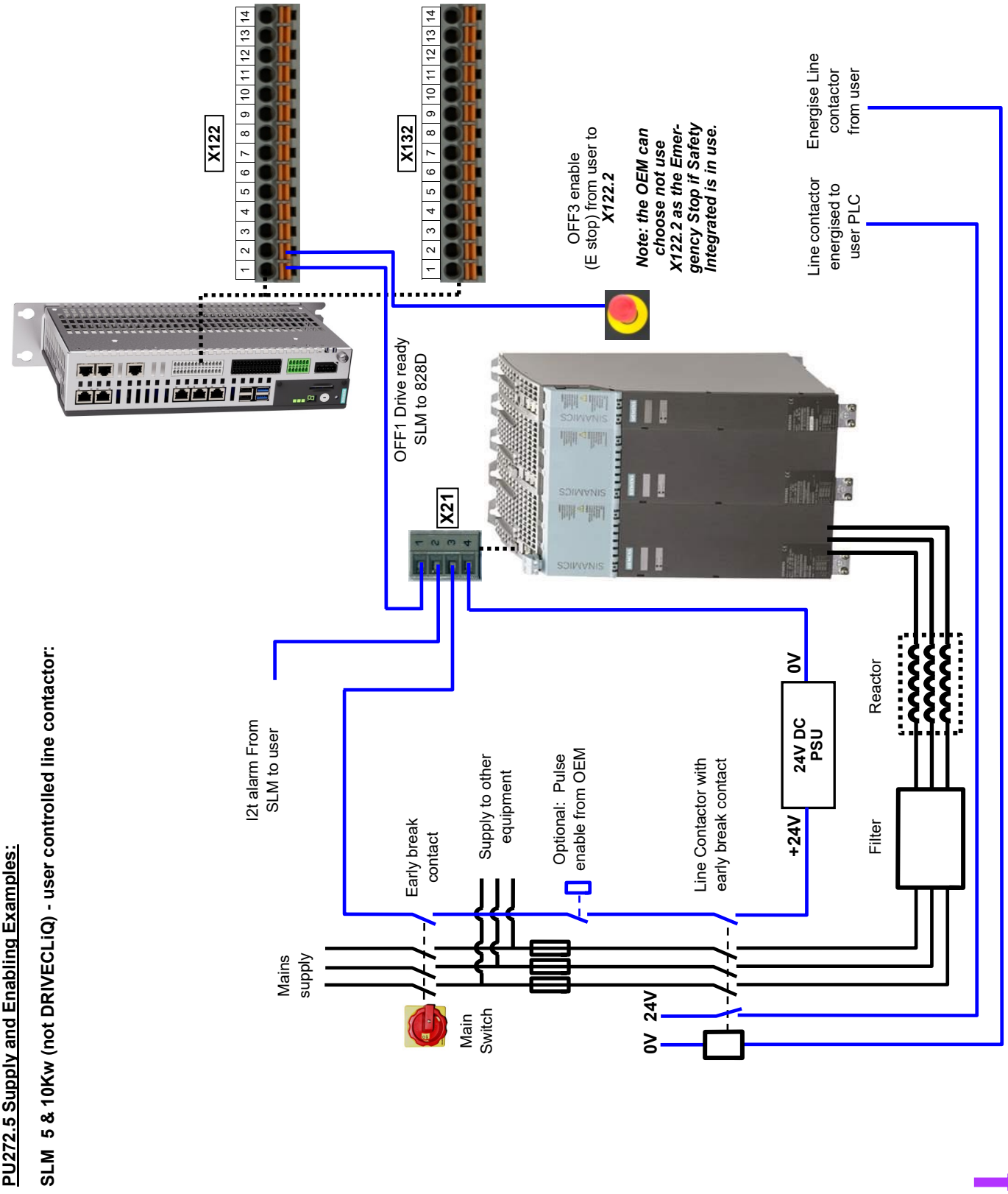
PU272.5 Supply and Enabling Examples

SLM 5 & 10Kw (not DRIVECLiQ) - no line contactor:

Notes:

828D PU272.5 Mains Supply and Enabling Examples:

SLM 5 & 10Kw (not DRIVECLiQ) - user controlled line contactor:



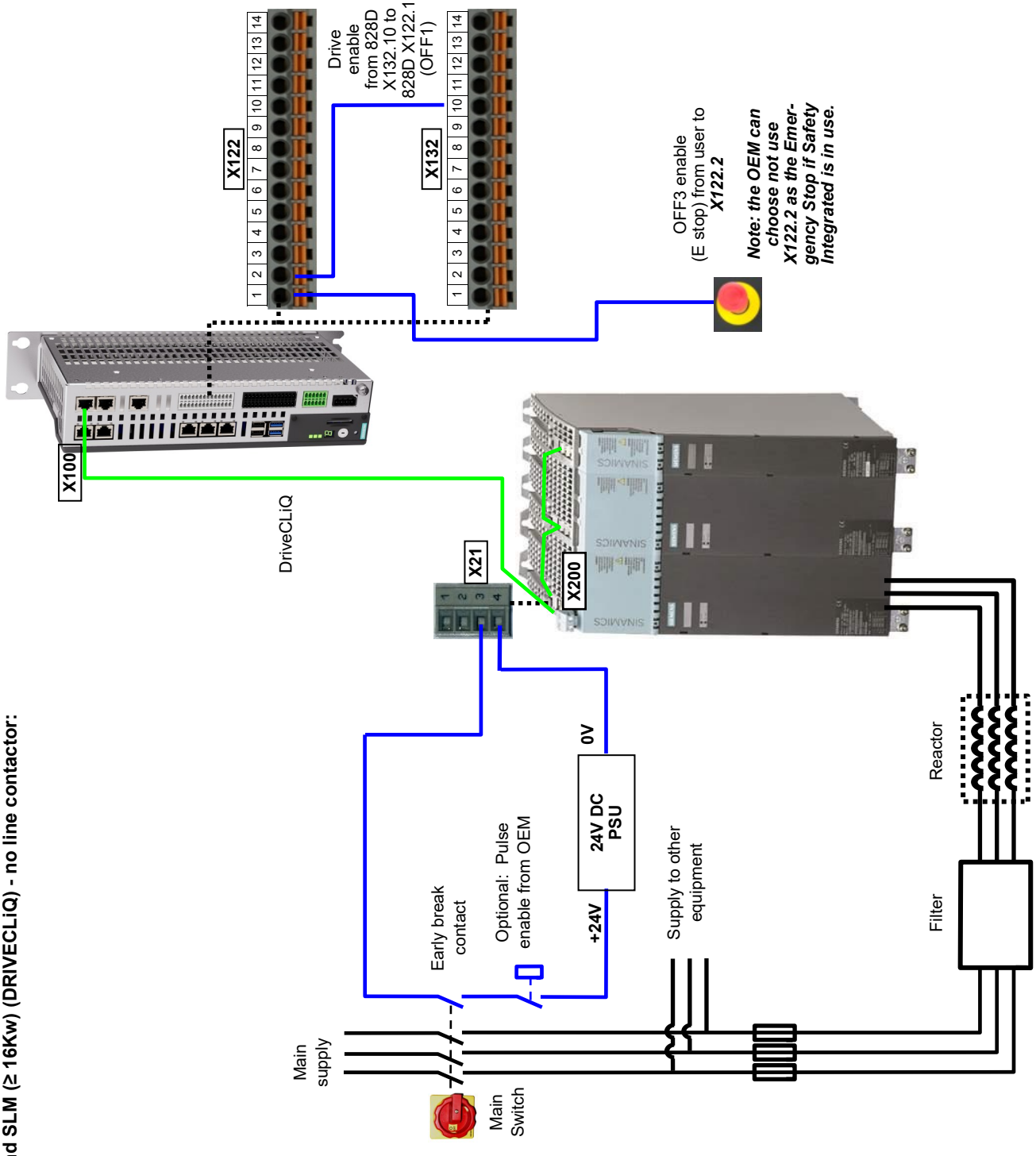
PU272.5 Supply and Enabling Examples:

SLM 5 & 10Kw (not DRIVECLiQ) - user controlled line contactor:

Notes:

828D PU 272.5 Supply and Enabling Examples

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - no line contactor:



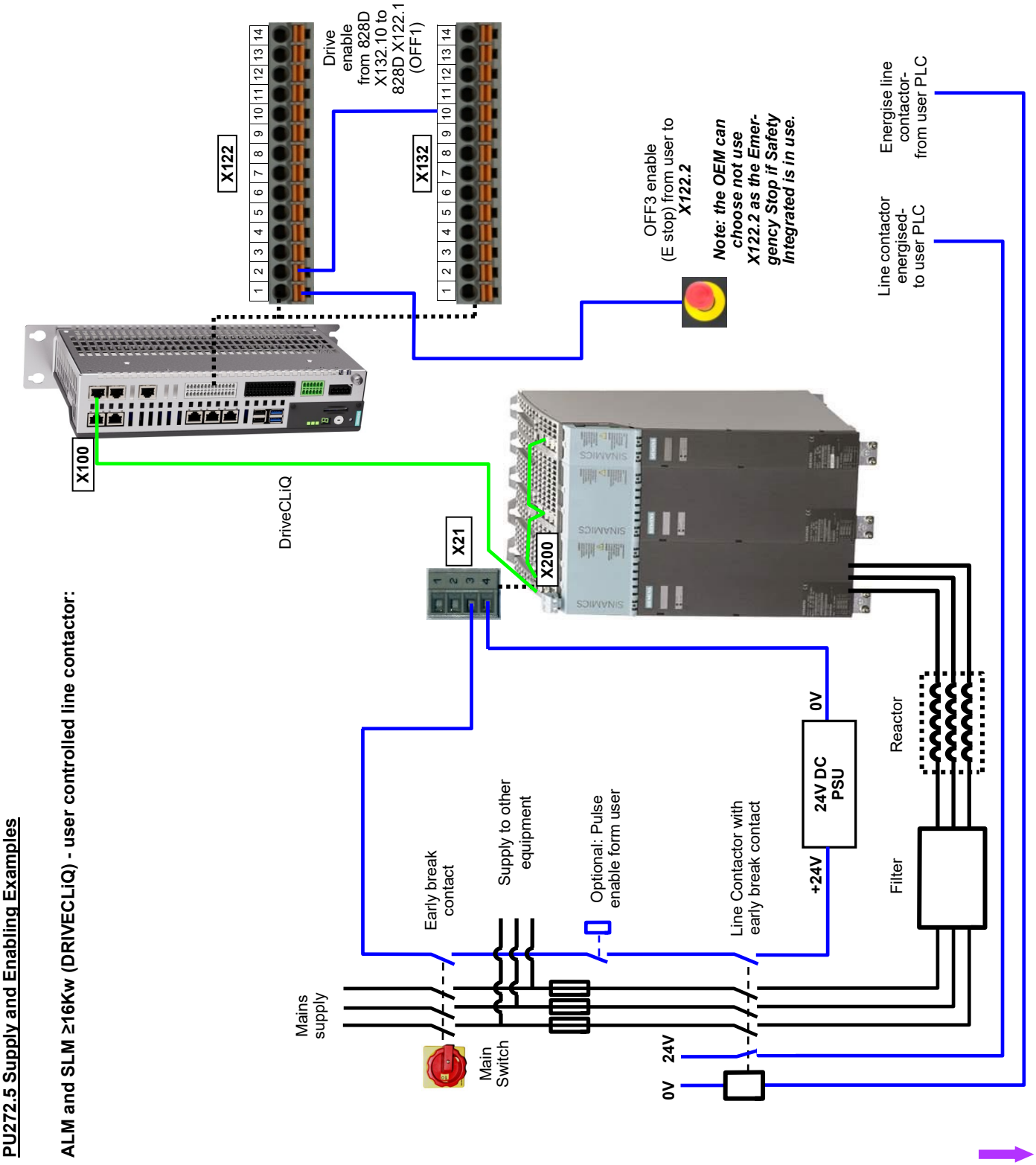
PU 272.5 Supply and Enabling Examples

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - no line contactor:

Notes:

828D PU272.5 Supply and Enabling Examples

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - user controlled line contactor:



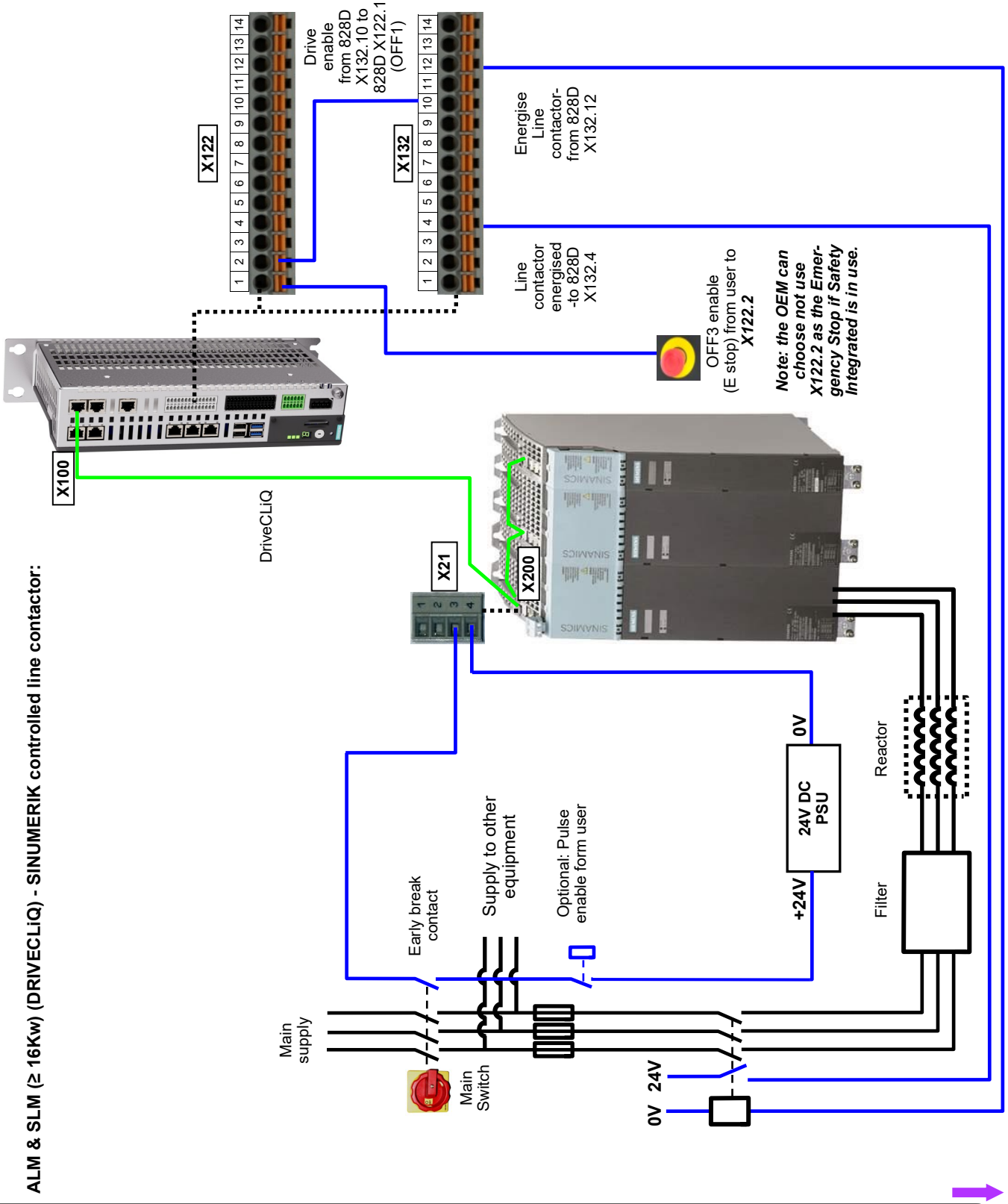
PU272.5 Supply and Enabling Examples

ALM and SLM ≥16Kw (DRIVECLiQ) - user controlled line contactor:

Notes:

828D PU 272.5 Supply and Enabling Examples

ALM & SLM (≥ 16Kw) (DRIVECLiQ) - SINUMERIK controlled line contactor:



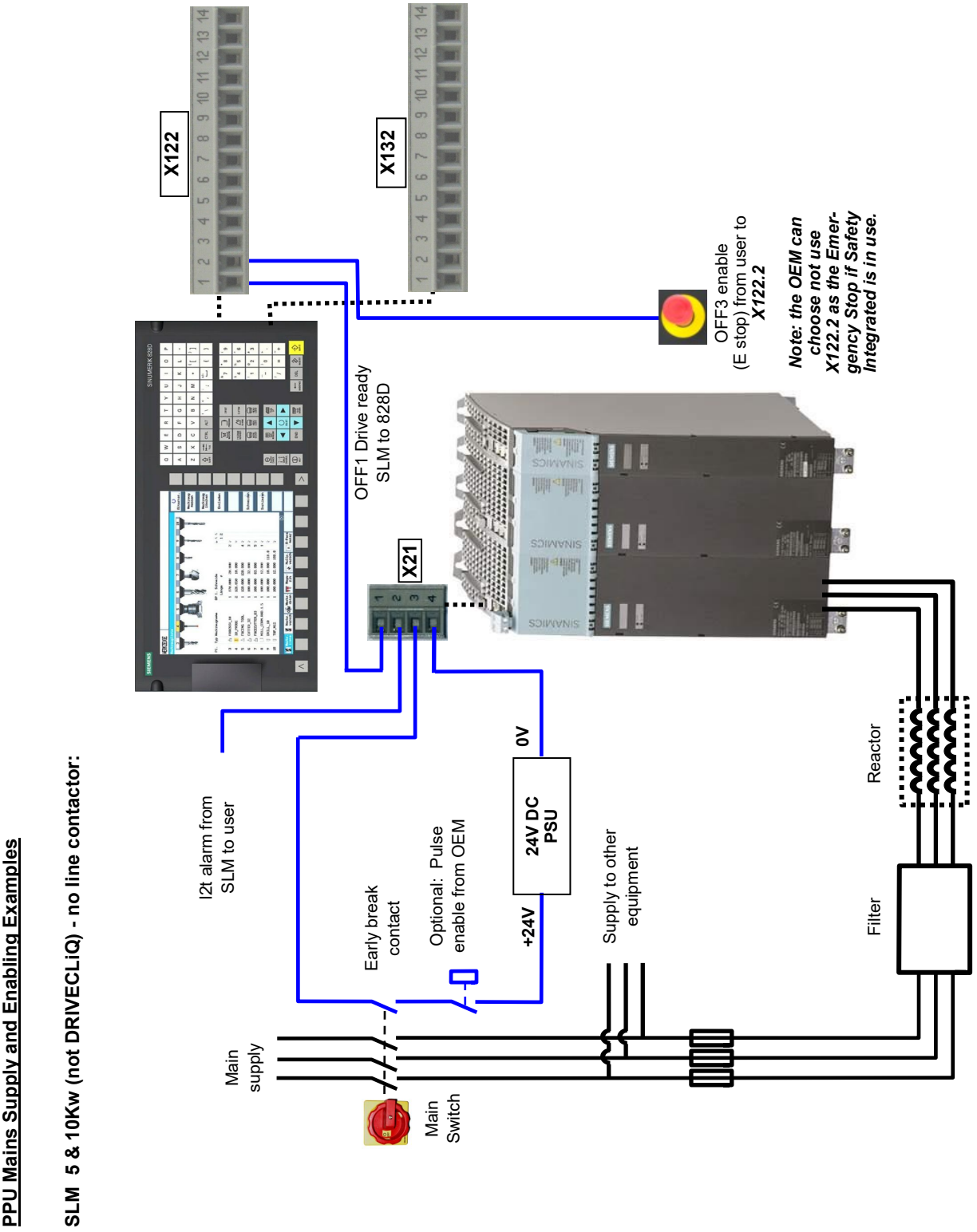
828D PU 272.Supply and Enabling Examples

ALM & SLM (≥ 16Kw) (DRIVECLiQ) - SINUMERIK controlled line contactor:

Notes:

828D PPU Mains Supply and Enabling Examples

SLM 5 & 10Kw (not DRIVECLiQ) - no line contactor:



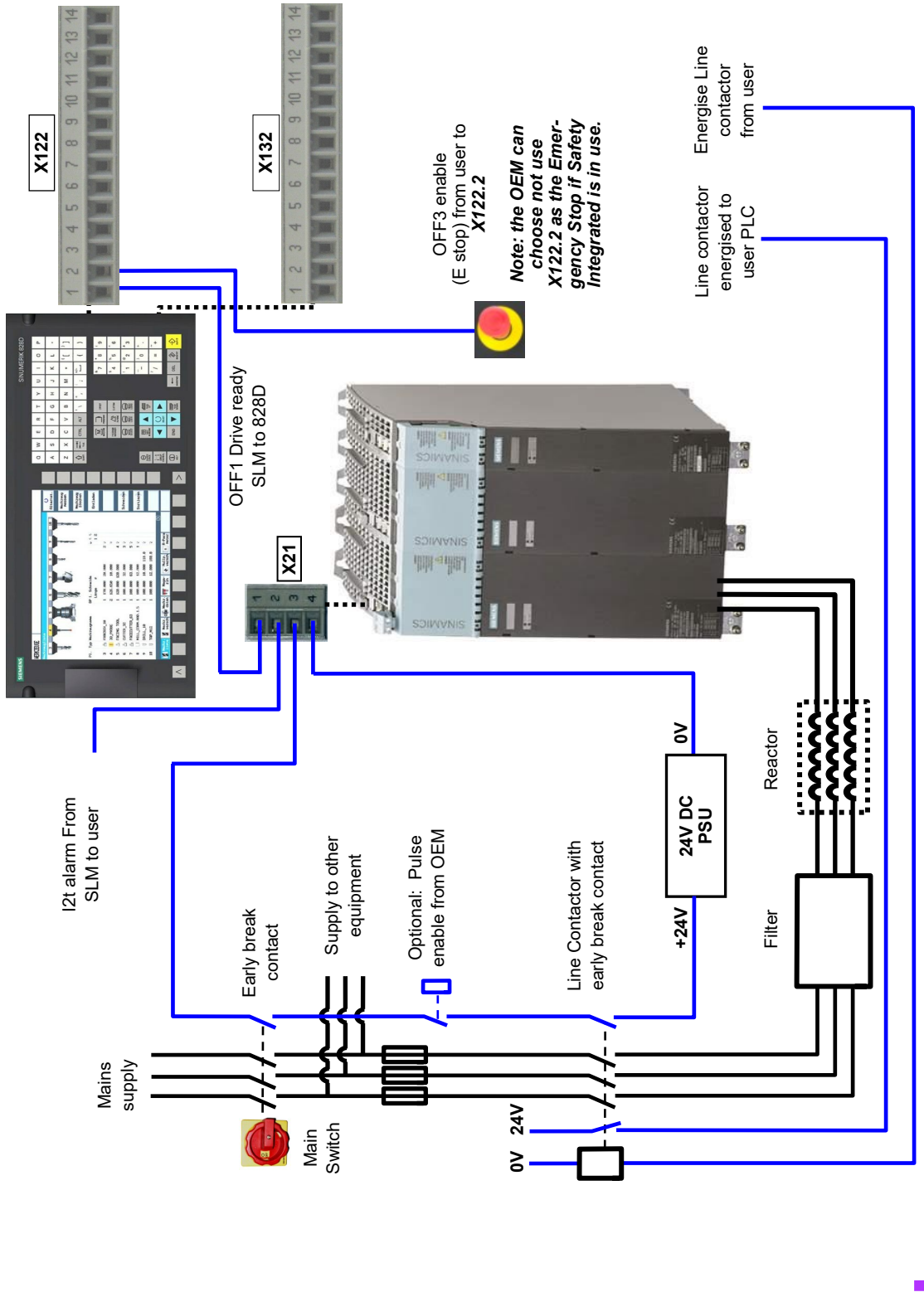
PPU Mains Supply and Enabling Examples

SLM 5 & 10Kw (not DRIVECLiQ) - no line contactor:

Notes:

828D PPU Mains Supply and Enabling Examples

SLM 5 & 10Kw (not DRIVECLiQ) - user controlled line contactor:



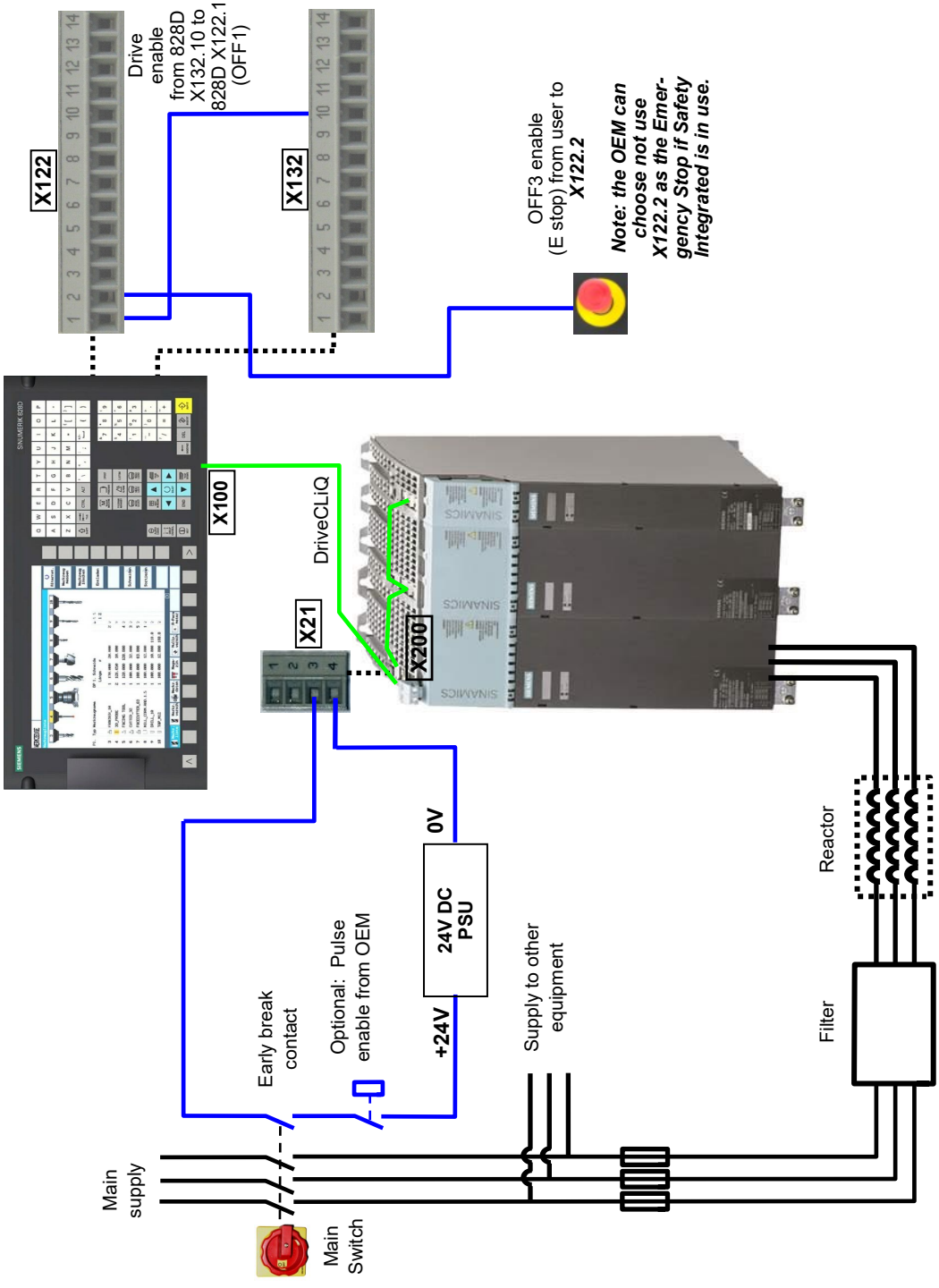
PPU Supply and Enabling Examples

SLM 5 & 10Kw (not DRIVECLiQ) - user controlled line contactor:

Notes:

828D PPU Mains and Enabling Examples

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - no line contactor:



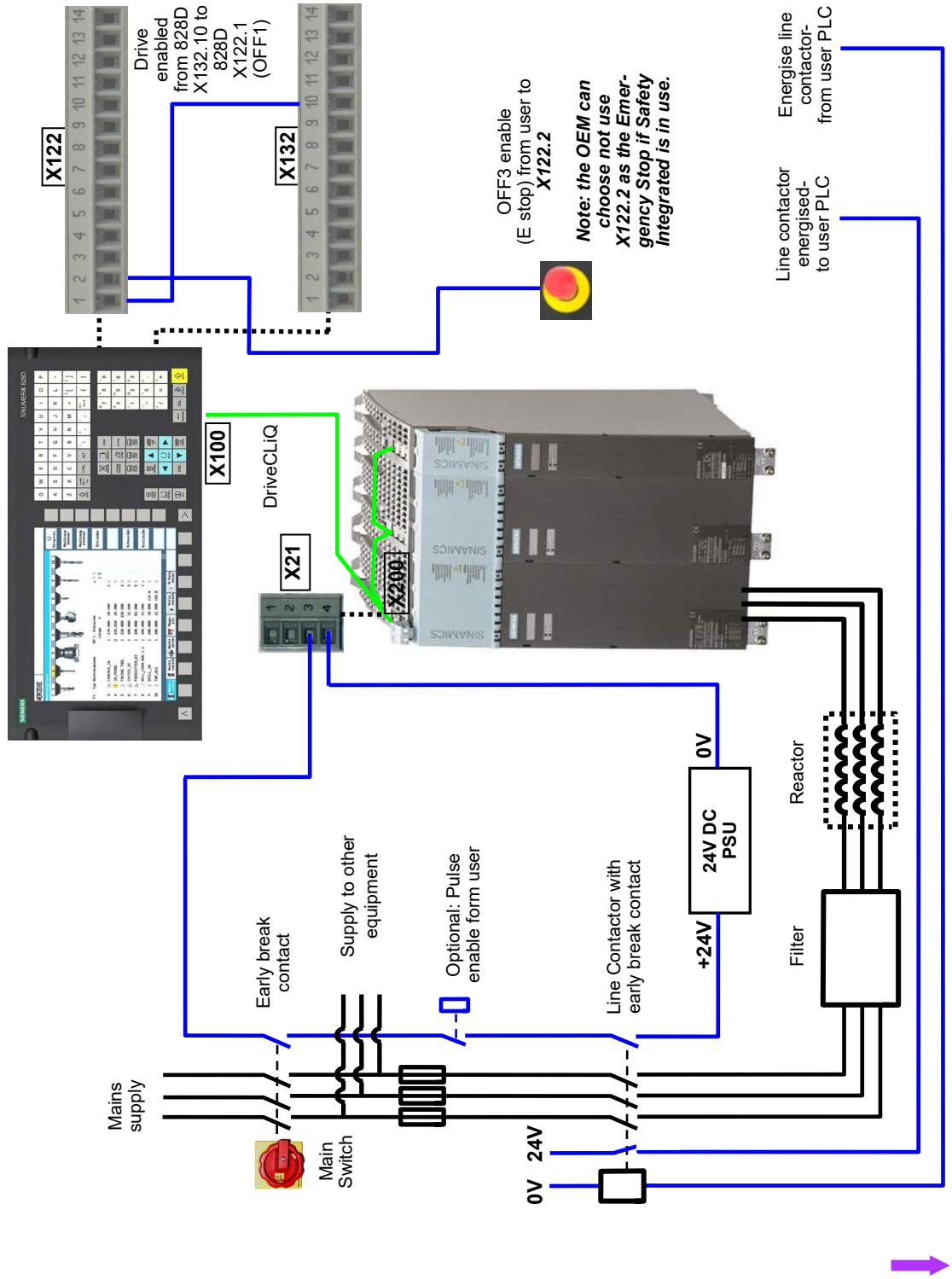
PPU Mains and Enabling Examples

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - no line contactor:

Notes:

828D PPU Supply and Enabling Examples

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - user controlled line contactor:



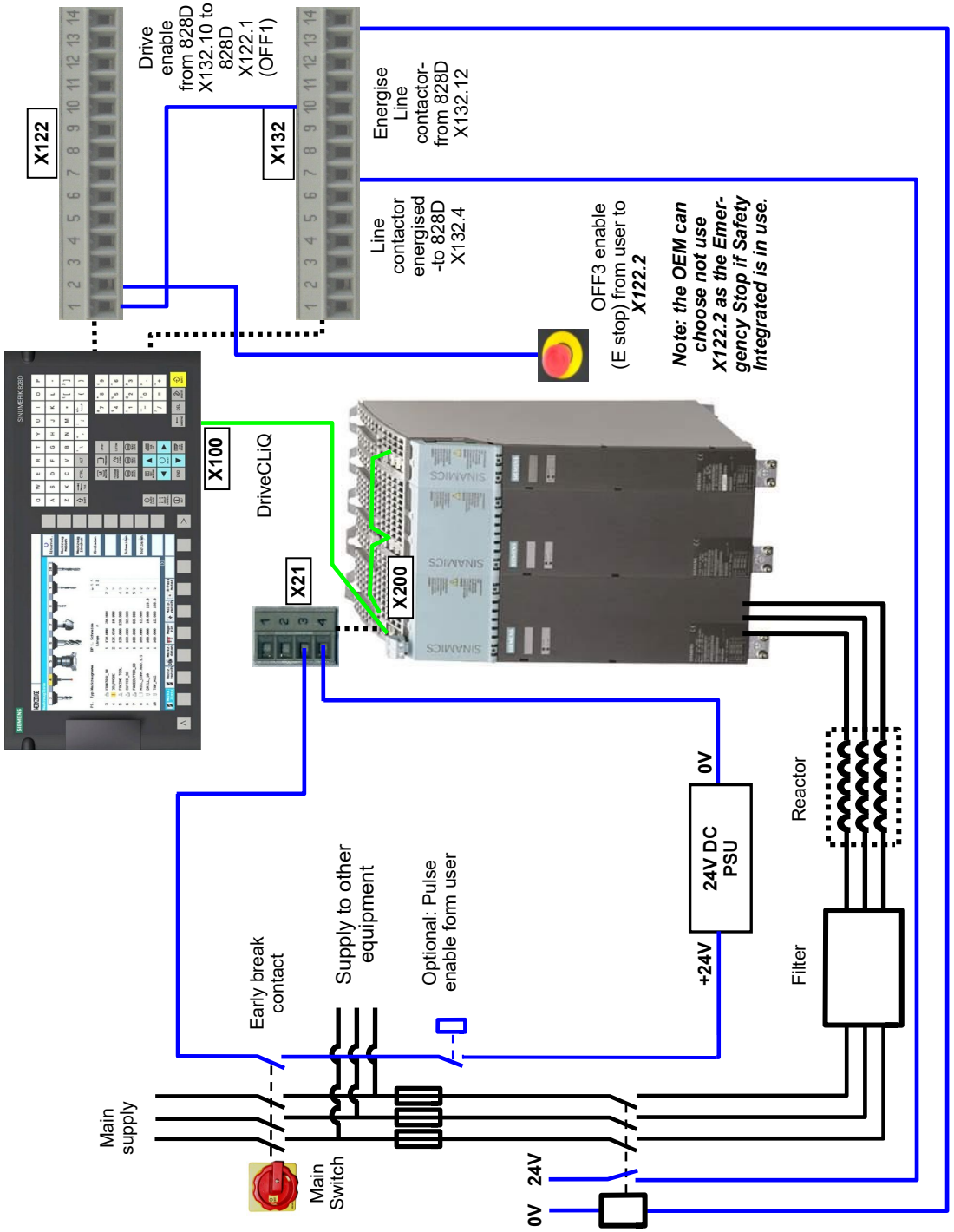
PPU Supply and Enabling Examples

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - user controlled line contactor:

Notes:

828D PPU Supply and Enabling Examples

ALM & SLM (≥ 16Kw) (DRIVECLiQ) - SINUMERIK controlled line contactor:



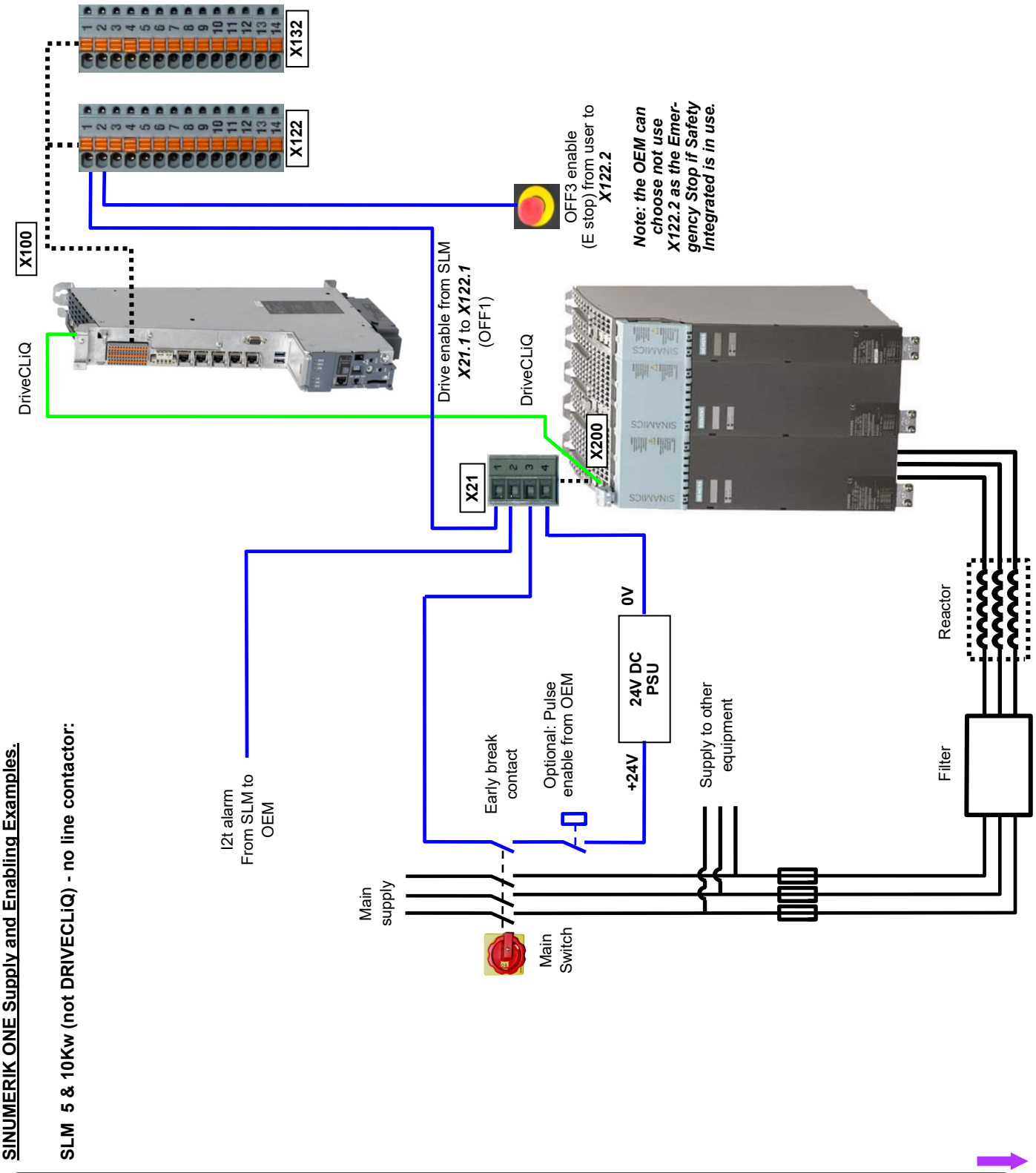
828D PPU Supply and Enabling Examples

ALM & SLM (≥ 16Kw) (DRIVECLiQ) - SINUMERIK controlled line contactor:

Notes:

SINUMERIK ONE Supply and Enabling Examples:

SLM 5 & 10Kw (not DRIVECLiQ) - no line contactor:



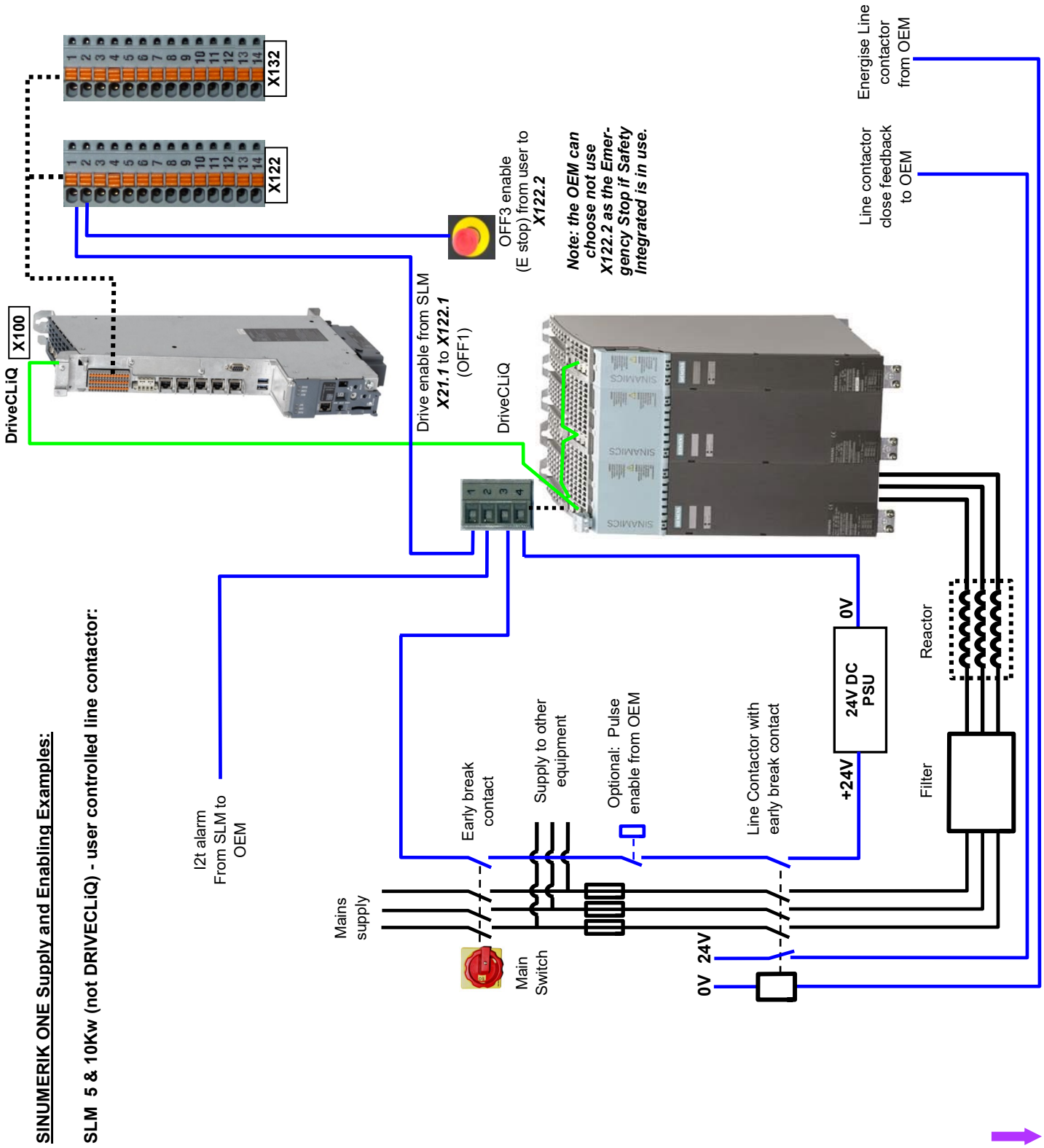
SINUMERIK ONE Supply and Enabling Examples.

SLM 5 & 10Kw (not DRIVECLiQ) - no line contactor:

Notes:

SINUMERIK ONE Supply and Enabling Examples:

SLM 5 & 10Kw (not DRIVECLiQ) - user controlled line contactor:



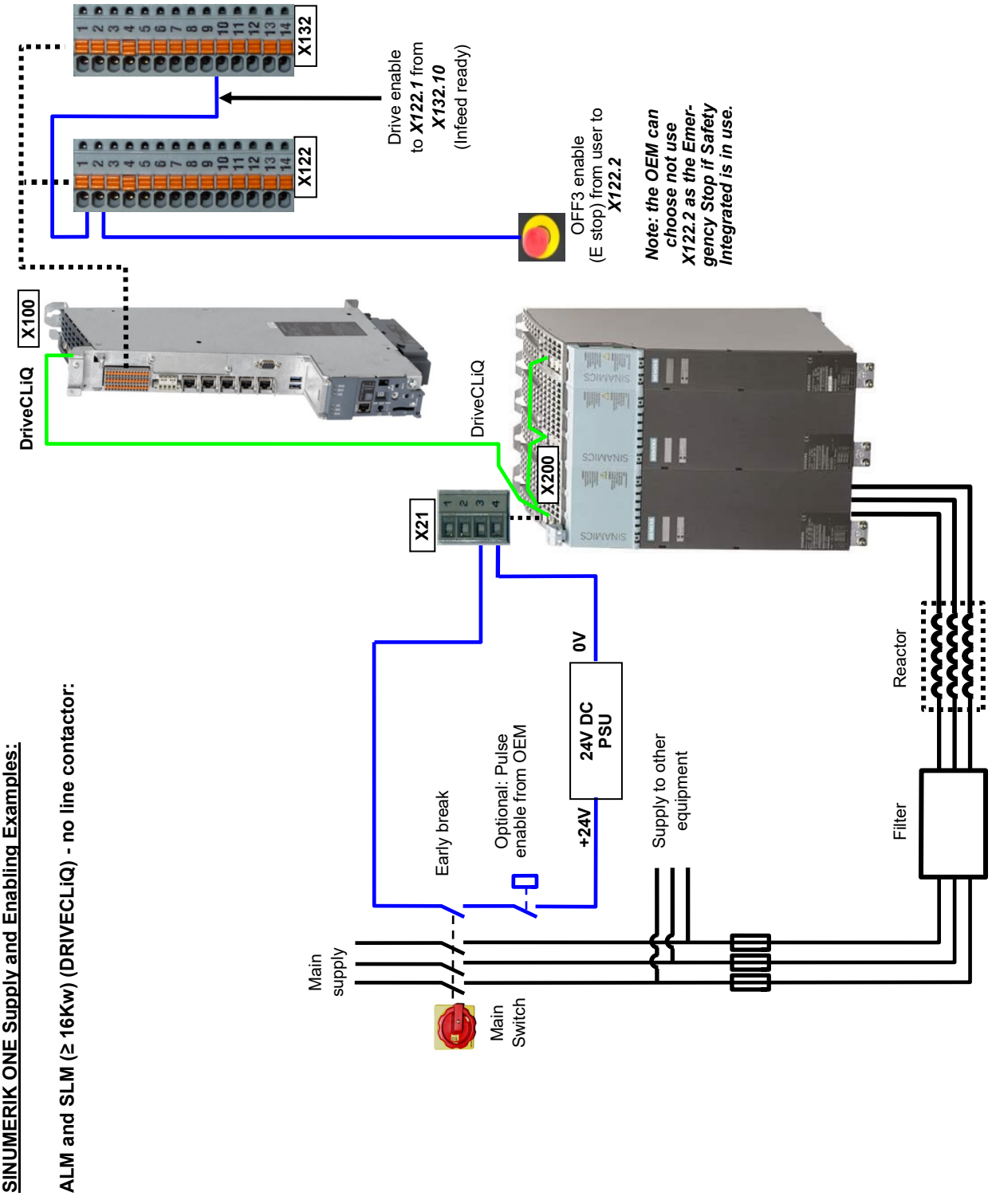
SINUMERIK ONE Supply and Enabling Examples:

SLM 5 & 10Kw (not DRIVECLiQ) - user controlled line contactor:

Notes:

SINUMERIK ONE Supply and Enabling Examples:

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - no line contactor:



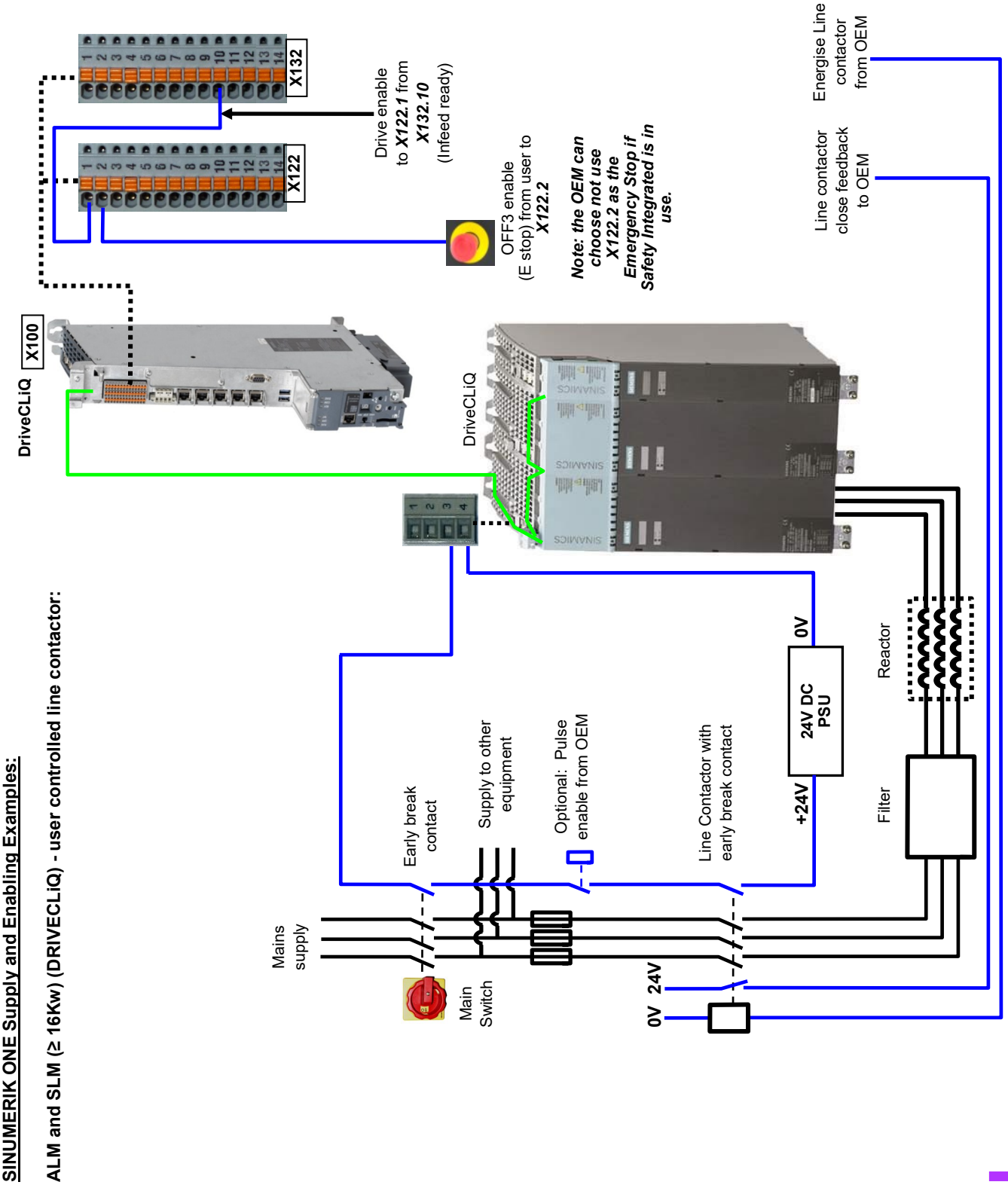
SINUMERIK ONE Supply and Enabling Examples:

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - no line contactor:

Notes:

SINUMERIK ONE Supply and Enabling Examples:

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - user controlled line contactor:



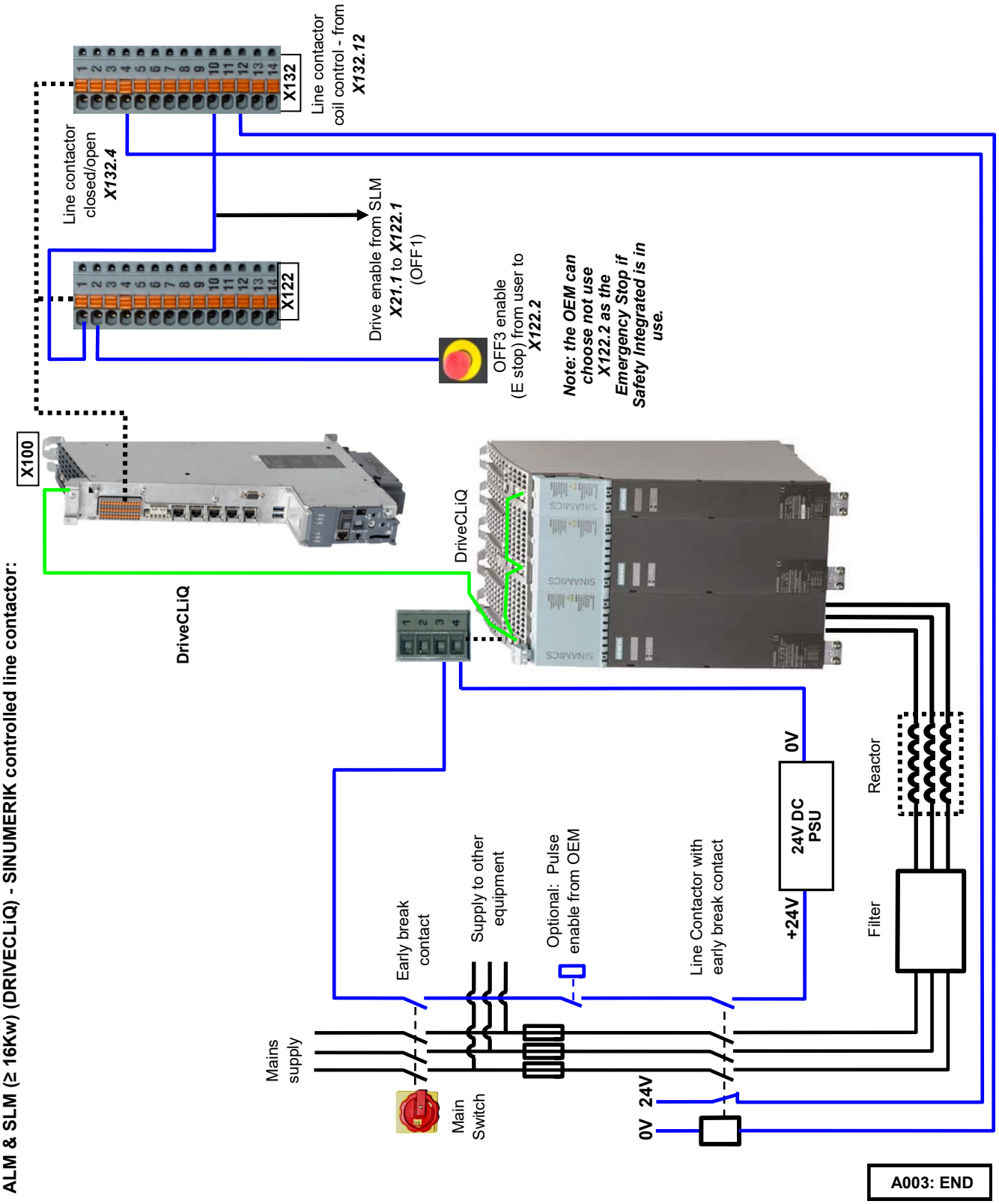
SINUMERIK ONE Supply and Enabling Examples:

ALM and SLM (≥ 16Kw) (DRIVECLiQ) - user controlled line contactor:

Notes:

SINUMERIK ONE Supply and Enabling Examples:

ALM & SLM (≥ 16Kw) (DRIVECLiQ) - SINUMERIK controlled line contactor:



SINUMERIK ONE Supply and Enabling Examples:

ALM & SLM (≥ 16Kw) (DRIVECLiQ) - SINUMERIK controlled line contactor:

Notes:

005

SINUMERIK

Network & Operator Panels

Module Description:

In the event of a problem with the machine it may be necessary to check the status of a network connection or TCU connection.
The information obtained from these status screens can be very useful to technical personnel and help speed up the fault finding process.

Module objective:

Upon completion of this module you will be able to use the network and OP‘s softkey functions.

Content:

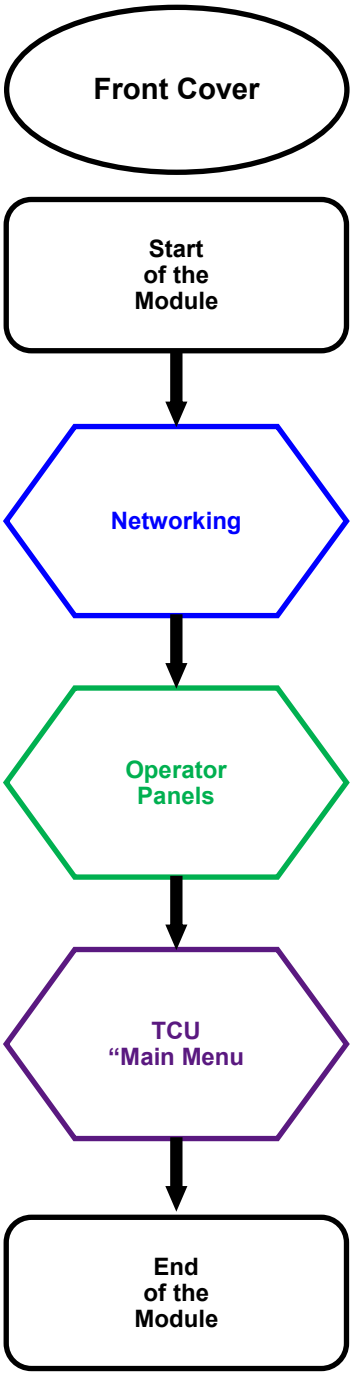
Networking Softkey

Operator Panels Softkey

TCU “Main Menu“

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start
of the
Module
Networking
Operator
Panels
TCU
“Main Menu
End
of the
Module

Network Softkey

The **network softkey** gives an overview of the:



System network (X120)
Company network (X130)
Service network (X127)

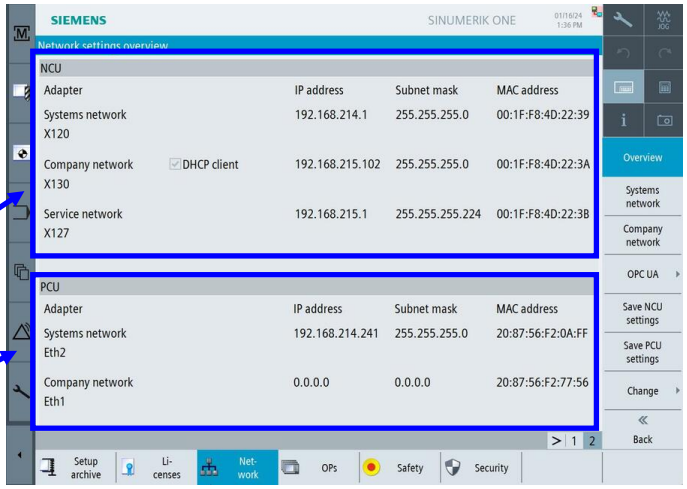
If an IPC is connected an overview of the:

System network (Eth 2)
Company network (Eth 1)

NCU network overview

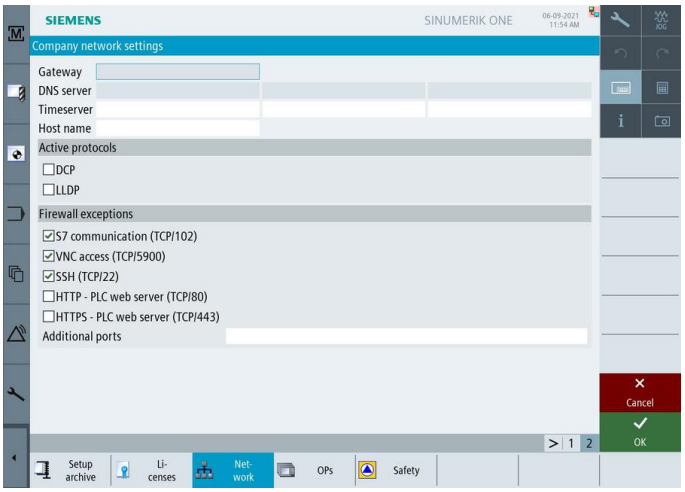
IPC network overview (If connected)

As shown:



Computer Network Settings

When using X130 in a manual set-up, the IP address and subnet mask of the external computer will also have to be manually entered. If the controller is part of an existing network this address must be unique.

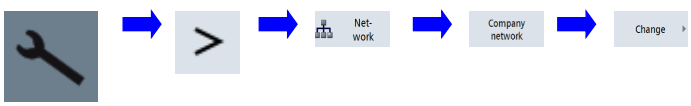


Configuring Interface X130

Ethernet interfaceX130 is configurable, the port can be assigned a fixed IP address or be set to automatically acquire an IP address.
Interface X130 is primarily for connecting to the company network. When configuring for connection to a company network, certain information from the company network administrator is required.

Firewall: X130 has a firewall which can inhibit the online connection of certain software tools. It is possible to open ports to allow connections to be made.

Change the firewall with the following key selection:



Notes:

X130 Controller Set-up

The controller settings for X130 can now be entered.

Any existing X130 settings should be recorded before any changes are made.

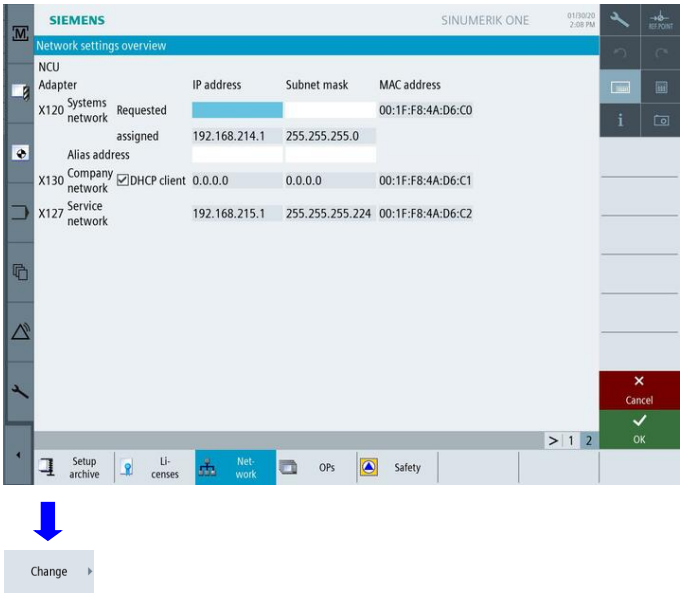
IP address restrictions- IP addresses **192.168.215.xxx** & **192.168.214.xxx**. are reserved for use by the system. Addresses within these ranges should not be used for X130.

The following example shows how to manually assign an IP address to X130.

Push the SK



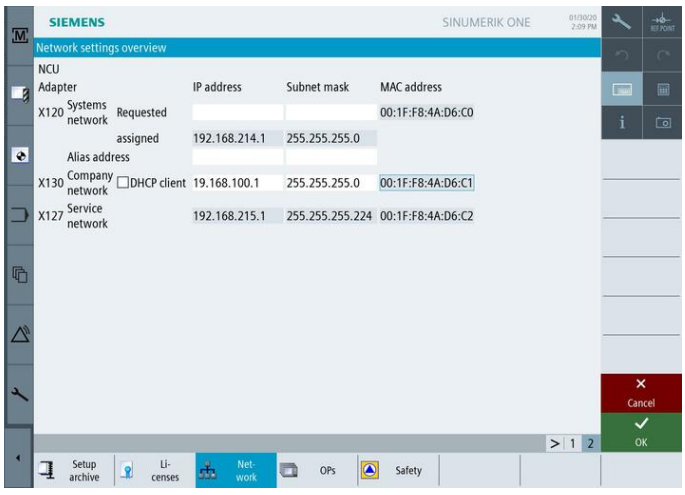
An overview of the X120 and X130 interfaces is displayed.
X120 is the system network to which is connected hardware such as Machine Control Panels (MCPs), Operator Panels (OPs) etc. This interface cannot be configured.
X130 is the company network and is configurable.



☑DHCP-Client To obtain IP address automatically.

☐DHCP-Client To enter a defined IP address.

The following example shows X130 being manually configured with an IP address of 192.168.100.1 and a subnet mask of 255.255.255.0



When the settings have been entered select the “OK” soft-key”

Switch the control off/on to activate the settings.

Notes:

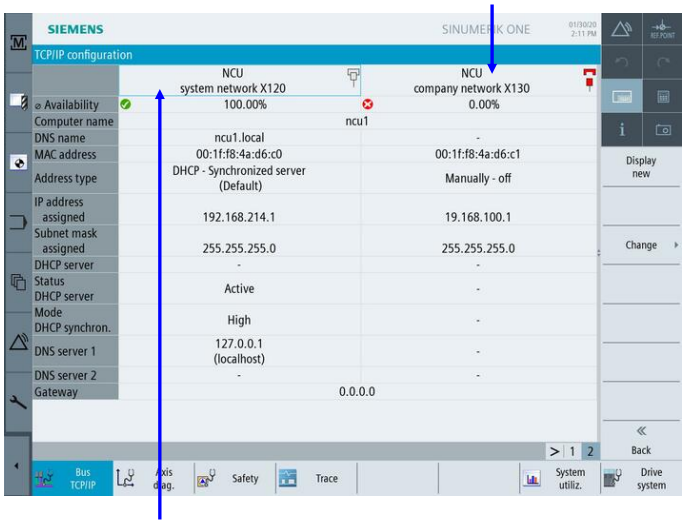
Diagnostic network

Check that the settings for X130 are correct in the “TCP/IP diagnostics” screen with the following key selection:



The “Availability” of the network connection is shown. In the example below X120 has a green/white tick symbol with an availability of 100%, this indicates that the connection has hardware attached and is healthy.

X130: Connection not in use



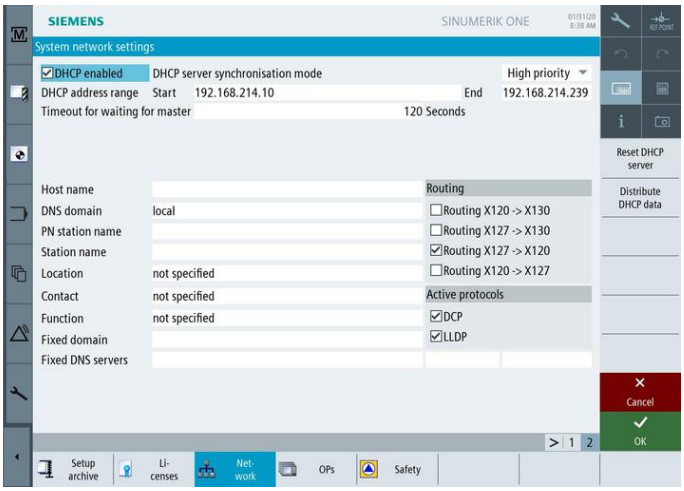
X120: Connection in use

Configuration of port X120 Network

All operator stations in the system network (TCUs, HT8, IPC) are administered on the NCU with "**master priority**", i.e. this contains the config.ini files relevant for the runtime and also the .leases file with all the IP addresses assigned in the system.

Adjusting system network on “**master priority**“





Notes:

Notes:

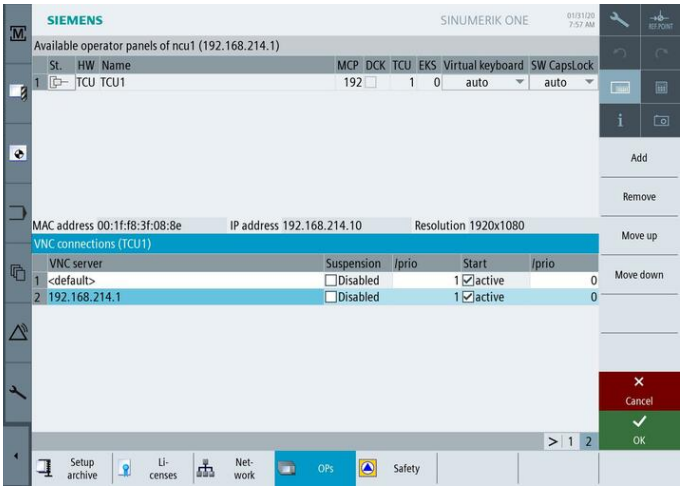
OPs Softkeyc

ONLY available on SINUMERIK ONE

Setting VNC Connection of the OP/TCU

The server IP address is set via the properties of the TCU, e.g. 192.168.214.1 (HMI of the NCU) or 192.168.214.241 (HMI of the IPC with distributed configuration).
For example, for a loading point with OP8/TCU and a distributed IPC with OP15 and TCU in the operator panel. One VNC points to the HMI NCU (OP8), the other to the HMI IPC.

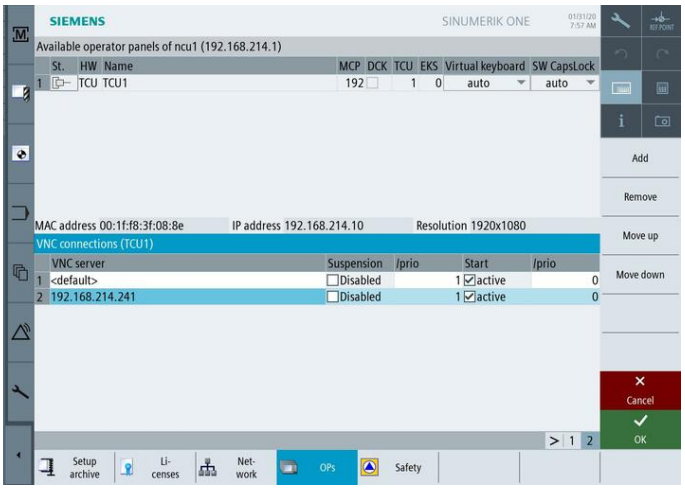




The distribution of the config.ini files to all the other NCUs/ IPCs is performed by the master NCU with the "**Reload**" softkey.

With the "I**nvalidate cache**" softkey and subsequent Power Off/On of the entire system, the NCUs/IPCs are assigned their specified IP addresses and new IP addresses assigned for TCUs and MCPs and then the .leases file distributed to all NCUs/IPCs.

The config.ini files of all operator stations can be edited directly on the master NCU with the page **OPs** , on each HMI.
The distribution of the data to all NCUs/IPCs is performed automatically with each change.

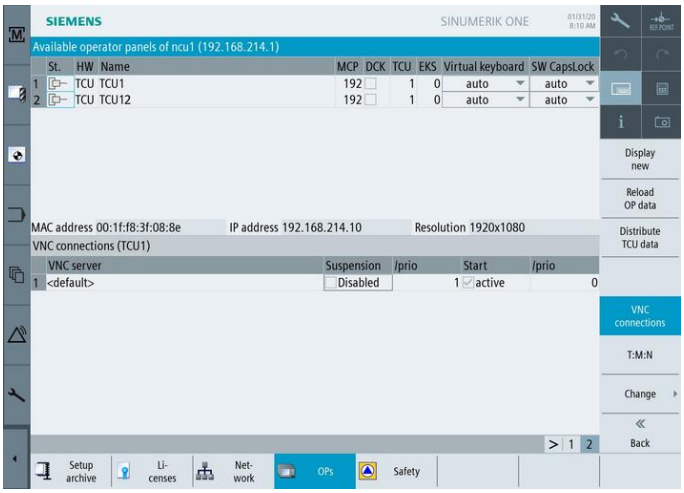


All settings with regard to networking and TCU configuration can be made via "**Network & OP´s**"!

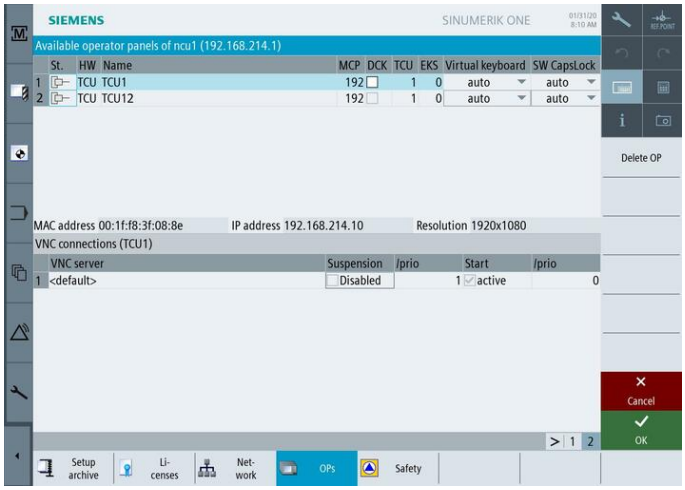
The Config.ini files card/user/common/tcu/<TCU-Name>/common/tcu/ config.ini and the basesys.ini are edited via **Network & OP´s**

Notes:

Removal of TCUs
TCUs that are not active can be removed on the Available operator panels …. .!



Change



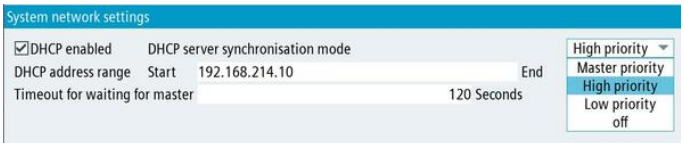
Delete OP

Configuration of one NCU with TCUs and MCPs

In principle, no special settings (basesys.ini) are required for this configuration. DHCP server and internal HMI sl remain switched on.

Configuration of one NCU with one IPC, MCPs and, if required, TCUs

The following settings (basesys.ini) of the NCU are required for this configuration:



As a rule, the internal HMI sl should be switched off, because the operation of two HMIs (HMI Advanced and Operate) on one NCU is only permitted in special cases.

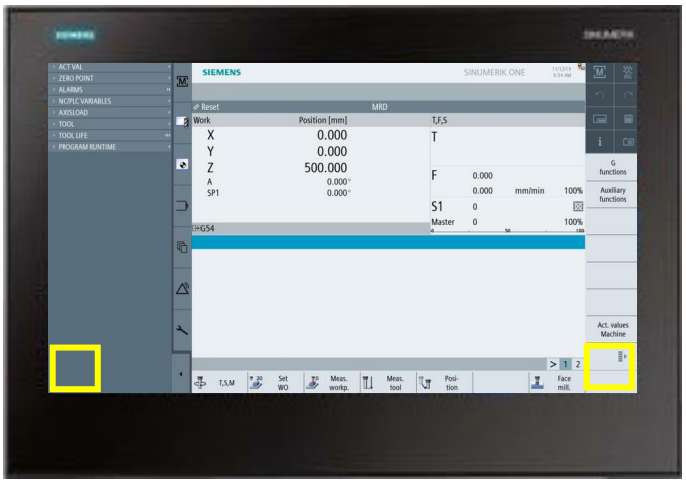
If the system contains a TCU (HT8) that is to be connected to the IPC per default, the IP address of the IPC must be entered at VNC connection (config.ini)!

Notes:

TCU “Main Menu“

TCU for **Sinamic ITCxx00**

Simultaneously touch both, the lower left-hand and right-hand corners of the active display area with one finger each to show the TCU menu.



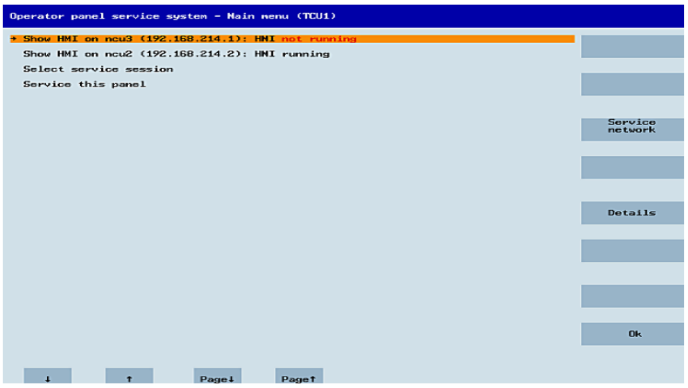
All other TCUs:

The "Main menu (TCU1)" dialog is started by simultaneously pressing  

or with F9+F10 (Keyboard)

The main menu is divided into:
The title "Main menu" followed by the TCU name in parentheses.

- "Show HMI" displays the selected HMI
- "Select service session" triggers a server scan as a first step, to detect all HMI and VNC servers in the local (system) network. This can be used to select the HMI for the TCU!
- "Service this panel" goes to a submenu "Service menu for operator panel". Provides the system settings for the **Operator Panel**



005: END

Notes:

A007

SINUMERIK ONE

NCU IPC ITC Connections and Diagnostics

Module Description:

This module describes the various interfaces and diagnostic features found on the Sinumerik ONE and the most commonly used operator components.

NCU (Numerical Control Unit) - hosts the NC, PLC, CU Drive and HMI control components and is central to all operations. Many interfaces and LEDs are available.

IPC (Industrial PC) - has the various software installed. (Windows, Sinumerik Operate, Shopmill/Shopturn etc.) They are equipped with an SSD (Solid State Disk) drive. Many interfaces are available. Various LEDs are used for diagnostic purposes.

ITC (Industrial Thin Client) - serves solely as a human/machine interface and has no data storage capability or diagnostic displays. Various interfaces are found on the ITC.

IFP (Industrial Flat Panel) - LCD monitors in robust industrial design with widescreen format and featuring multitouch operation. Integration with NCU 1750 and NCU 1760

Module objective:

Upon completion of this module you will be able to locate and understand the functionality of the:

- IPC Interfaces and diagnostic LEDs/displays.
- ITC Interfaces
- NCU/PPU Interfaces and diagnostic LEDs/displays
- IFP Interfaces

Content:

General Information and Commonality

NCU17x0 Interface Locations and Description

PPU1740 Interface Locations and Description

Simatic Industrial PCs IPC427E & IPC477E

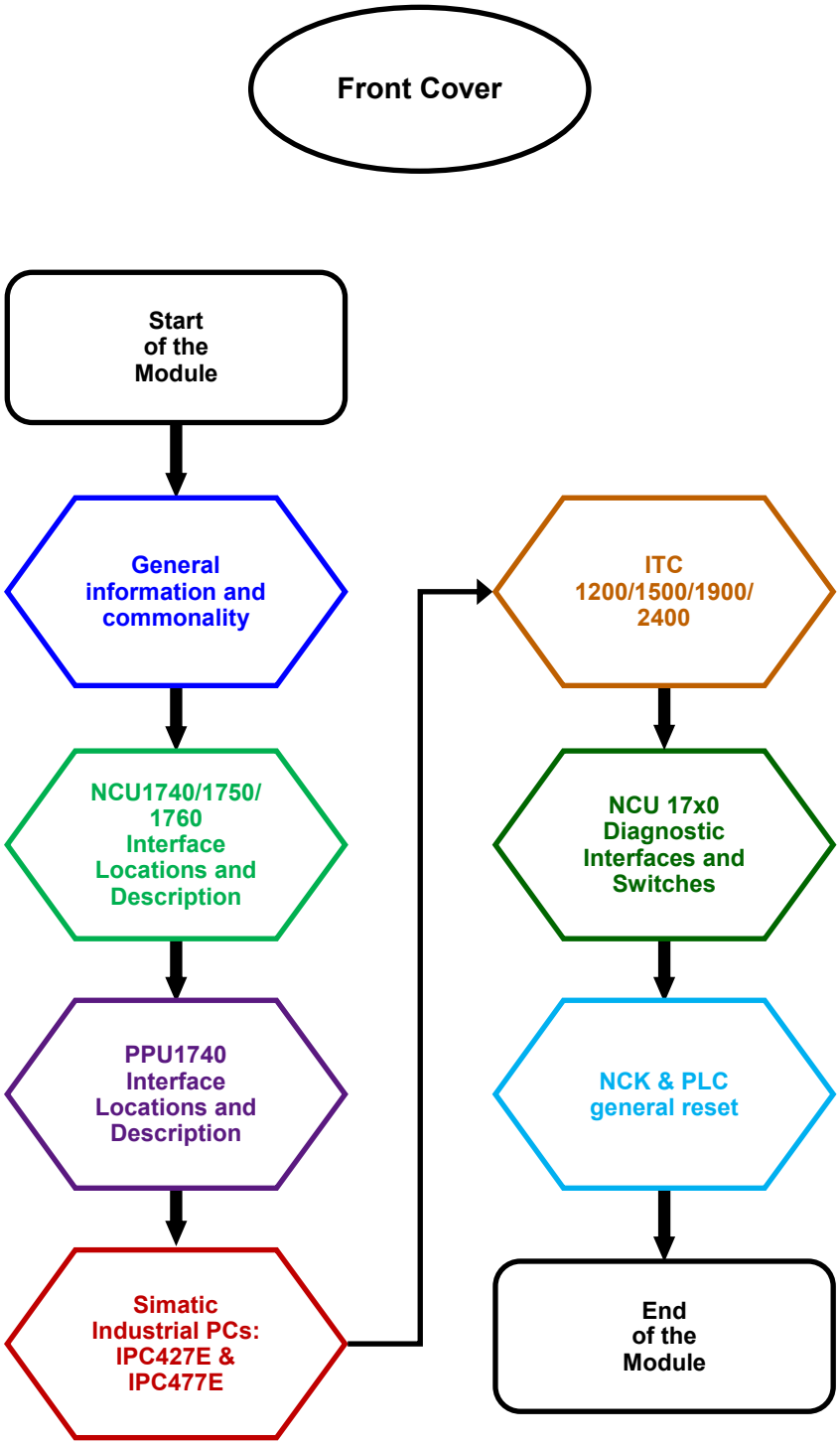
ITC 1200/1500/1900/2400

NCU 17x0 Status LEDs and 7 Segment Display

NCK & PLC General Reset

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
General information and commonality
NCU1740/1750/ 1760 Interface Locations and Description
PPU1740 Interface Locations and Description
Simatic Industrial PCs: IPC427E & IPC477E
ITC 1200/1500/1900/ 2400
NCU 17x0 Diagnostic Interfaces and Switches
NCK & PLC general reset
End of the Module

General information and Commonality

The various equipment described in this module uses common interfaces (PROFINET, Ethernet etc.)
Detailed interface information is found in the following tables.

- X120 provides the link to the system network (e.g. ITC).
- X130 connects the NCU to the company network.
- X127 is used for commissioning and remote diagnostics via teleservice adapter IE.

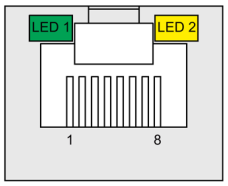
Assignment of the Ethernet interfaces X120, X130 and X127 in 10/100 Mbit/s mode

Connector	Pin	Name	Type	Function
	1	TxD+	O	Transmit data
	2	TxD-		
	3	RxD+	I	Receive data
	4/5	GND	-	(terminated internally with 75 Ω; not required for data transmission)
	6	RD-	I	Receive data
	7/8	GND	-	(terminated internally with 75 Ω; not required for data transmission)
	Shield		-	On connector housing

Assignment of the Ethernet interfaces X120, X130 and X127 in 1 Gbit/s mode

Connector	Pin	Name	Type	Function
	1	DA+	B	Bidirectional pair A+
	2	DA-		Bidirectional pair A-
	3	DB+		Bidirectional pair B+
	4	DC+		Bidirectional pair C+
	5	DC-		Bidirectional pair C-
	6	DB-		Bidirectional pair B-
	7	DD+		Bidirectional pair D+
	8	DD-		Bidirectional pair D-
	Shield	-	-	On connector housing

Ethernet - RJ45 interface LED description.



LED	State	Description
LED 1	Off	10 MBit/s
	Green	100 MBit/s
	Orange	1000 MBit/s
LED 2	Off	no data exchange
	Lit	Active connection
	Flashing	Activity

Notes:

PROFINET:

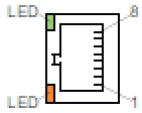
PROFINET IO communications networks between the PLC of the control and the field devices.

- X150 supports RT and IRT
- X160 supports RT

Assignment of PROFINET interfaces X150, ports 1, 2 in the 10/100 MBit/s mode:

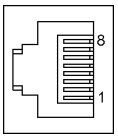
Connector	Pin	Name	Type	Function
	1	TxD+	O	Transmit data
	2	TxD-		
	3	RxD+	I	Receive data
	4/5	GND	-	(terminated internally with 75 Ω; not required for data transmission)
	6	RD-	I	Receive data
	7/8	GND	-	(terminated internally with 75 Ω; not required for data transmission)
	Shield		-	On connector housing

Assignment of PROFINET interface X160 in the 10/100 MBit/s mode

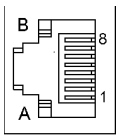
Connector	Pin	Name	Type	Function
	1	DA+	B	Bidirectional pair A+
	2	DA-		Bidirectional pair A-
	3	DB+		Bidirectional pair B+
	4	DC+		Bidirectional pair C+
	5	DC-		Bidirectional pair C-
	6	DB-		Bidirectional pair B-
	7	DD+		Bidirectional pair D+
	8	DD-		Bidirectional pair D-
	Shield	-	-	On connector housing

DRIVE-CLiQ:

Assignment of DRIVE-CLiQ interfaces - without +24V DC supply.

Connector	Pin	Name	Type	Function
	1	TxD+	O	Transmit data
	2	TxD-		
	3	RxD+	I	Receive data
	4/5	GND	-	Not connected
	6	RD-	I	Receive data
	7/8	GND	-	Not connected
	Shield		-	On connector housing

Assignment of DRIVE-CLiQ interfaces - with +24V DC supply.

Connector	Pin	Name	Type	Function
	1	TxD+	O	Transmit data
	2	TxD-		
	3	RxD+	I	Receive data
	4/5	GND	-	Not connected
	6	RD-	I	Receive data
	7/8	GND	-	Not connected
	A	+24V DC	Power	Encoder supply - for connecting between S120 drives/motors/direct measuring systems and sensor modules/direct measuring systems
	B	0V DC	"	
	Shield		-	On connector housing

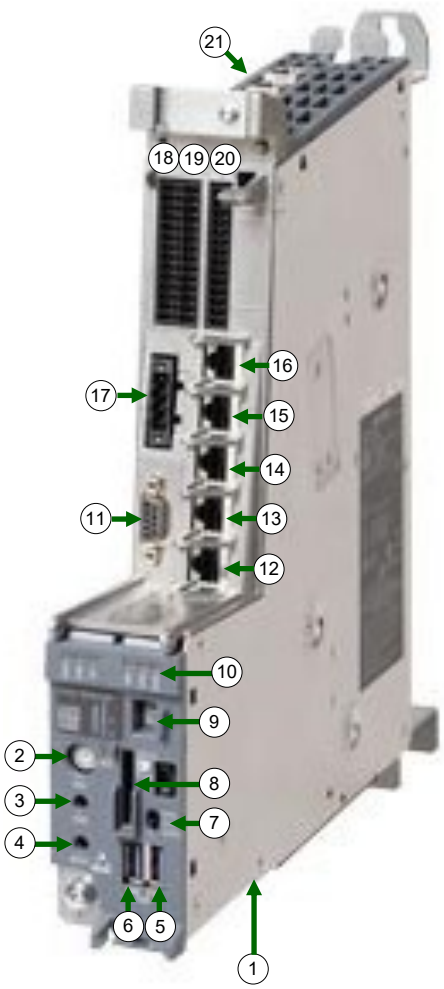


Notes:

NCU1740/750/1760 Interface Locations and Description

The location and a brief description of the various interfaces of the Sinumerik ONE NCU is shown below:

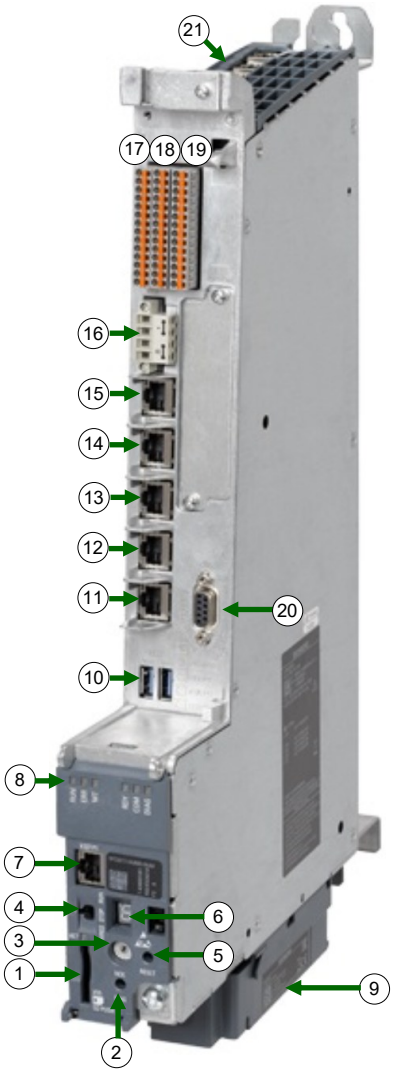
NCU1740:



Item	Interface	Description
1	X127	Service/commissioning Ethernet (lower side)
2	-	Diagnostic rotary switch
3	-	Diagnostic button
4	-	Reset button
5	X125	USB interface
6	X135	USB interface
7	-	PLC mode selector switch (toggle switch)
8	X55	System SD (Secure Digital) card slot
9	-	7-Segment display
10	-	Diagnostic LEDs
11	X126	Profibus DP interface
12	X130	PN/IE - Factory Network (X130)
13	X120	PN/IE - System Network (X120)
14	X160 - P1	1 x PROFINET interface
15	X150 - P2	PROFINET IRT interface (Isochronous Real Time)
16	X150 - P1	PROFINET IRT interface (Isochronous Real Time)
17	X124	Power supply - 24V DC
18	X122	Drive digital inputs and outputs
19	X132	Dirive digital inputs and outputs
20	X142	NCK Digital inputs and outputs
21	X100 - X105	4 x DRIVE-CLiQ interfaces

Notes:

NCU 1750/NCU 1760:

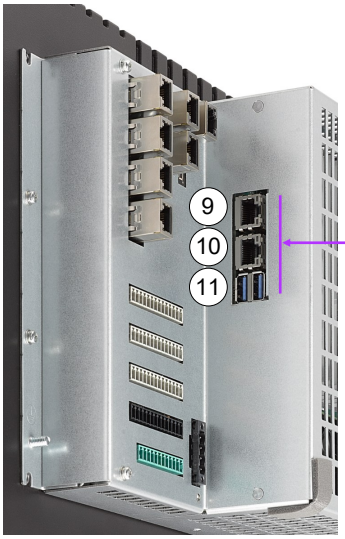


Item	Interface	Description
1	X55	System Secure Digital (SD) Card Slot
2	-	Diagnostics Button
3	-	NCK Commissioning Switch (rotary switch type)
4	-	PLC Run/Stop/Commissioning Toggle Switch
5	-	Reset button
6	-	7 Segment Display
7	X127	PN/IE Service/Commissioning Interface
8	-	Diagnostics LEDs
9	-	Fan housing
10	X125/X135	USB Interfaces
11	X130	PN/IE - Factory Network (X130)
12	X120	PN/IE - System Network (X120)
13	X160 - P1	1 x PROFINET Interface
14	X150 - P2	PROFINET Interfaces IRT (Isochronous Real Time)
15	X150 - P1	PROFINET Interfaces IRT (Isochronous Real Time)
16	X124	+24V DC Supply
17	X122	Drive Digital I/O
18	X132	Drive Digital I/O
19	X142	NCK Digital I/O
20	X126	PROFIBUS Interface
21	X100 - X105	6 x DRIVE-CLiQ Interfaces

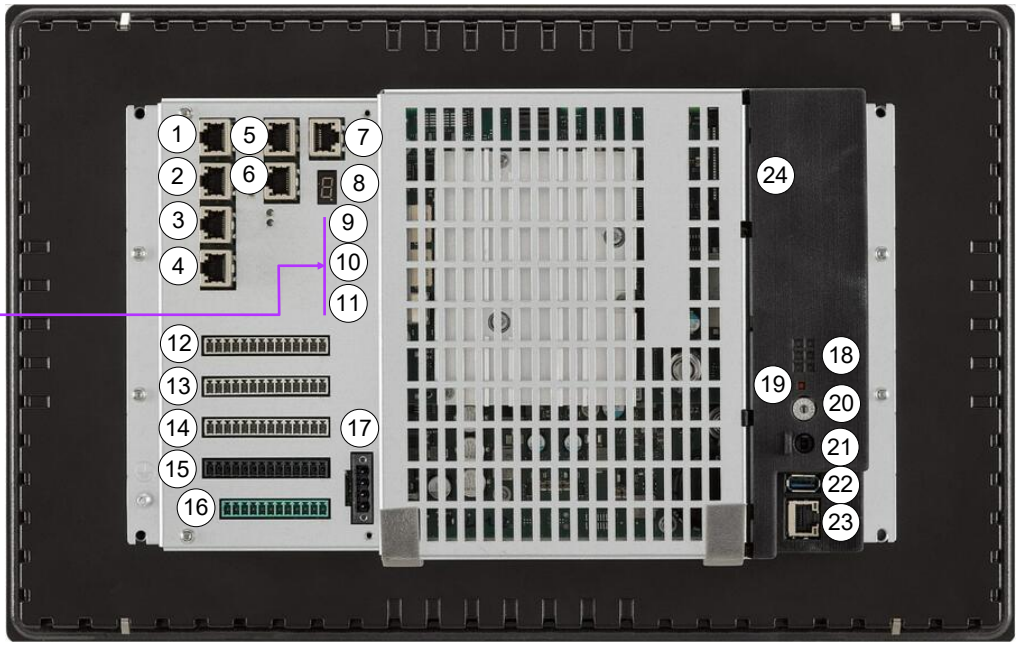
Notes:

PPU1740 Interface Locations and Description

Sideview of PPU 1740 showing the location of items 9, 10 & 11.



Rear view of PPU 1740



Item	Interface	Description
1	X100	DRIVE-CLiQ Interface
2	X101	DRIVE-CLiQ Interface
3	X102	DRIVE-CLiQ Interface
4	X103	DRIVE-CLiQ Interface
5	X150 P1	PROFINET IRT Port 1
6	X150 P2	PROFINET IRT Port 2
7	X160	PROFINET
8	-	7 Segment Display
9	X120	System Ethernet
10	X130	Factory Ethernet
11	X135/X145	USB
12	X122	Sinamics digital I/O
13	X132	Sinamics digital I/O
14	X242	NC digital I/O
15	X252	NC digital I/O
16	X143	Handwheel interface - two handwheels
17	X1	DC power supply - 24V
18	-	Diagnostic LEDs
19	-	Reset button
20	-	Diagnostic rotary switch
21	-	PLC Run/Stop switch
22	X125	USB
23	X127	Service/commissioning Ethernet
24	-	System SD card (accessed by removing the black plastic cover)

Notes:

Simatic Industrial PCs IPC427E & IPC477E

IPC - Industrial PC Overview

IPCs are available in different formats:

- IPC427E Box PC
- IPC477E Microbox PC

Software - Various software can be installed on the SSD of the IPC:

- Windows® operating system
- Sinumerik basic software
- Sinumerik Operate software
- ShopMill/ShopTurn software (option)
- Additional user defined software

Additional Data - System and user files are also stored on the SSD

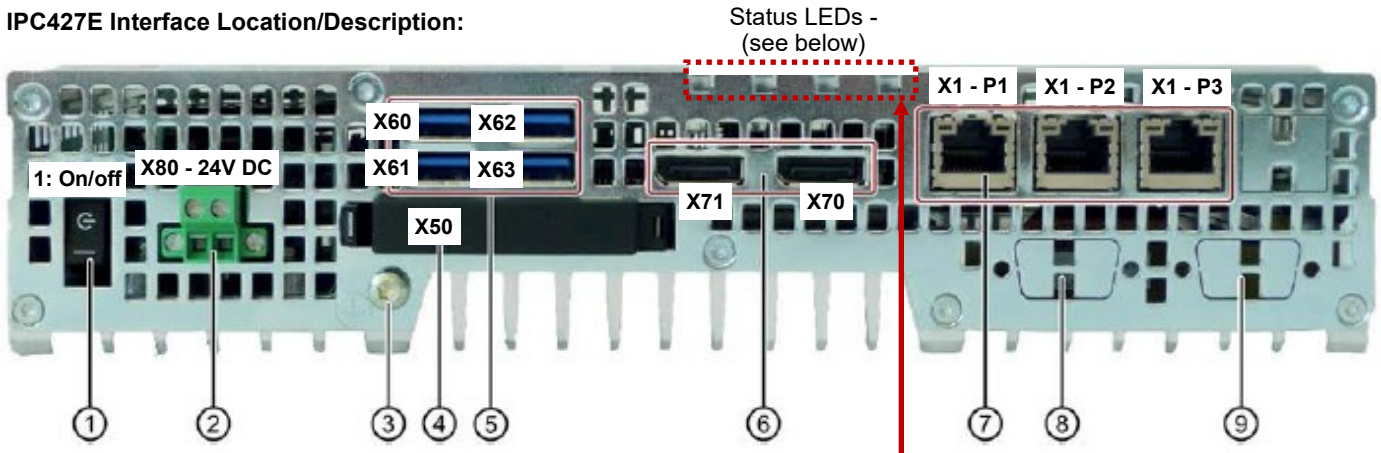
- Archives,
- User files
- Logs
- Configuration files etc.

Functionality - The different IPC versions have similar functionality but differ in several ways.
This is reflected in the varying interfaces/diagnostic displays which are described on the following pages.

Simatic IPC 427E Microbox PC



IPC427E Interface Location/Description:



Item	Interface	Function
①	-	Power off/on
②	X80	24V DC Supply
③	-	Protective earth connection
④	X50	CFast card slot
⑤	X60 - X63	4 x USB 3.0
⑥	X70/X71	2 x DPP Display ports (DVI/VGA optional)
⑦	X1 P1 - P3	3 x PROFINET/Industrial Ethernet
⑧	X30	COM1 - not available
⑨	X31	COM2 - not available



See the next page
for details.

Notes:

IPC 427E Status-LEDs:



LED	Status	Description
PC ON/WD	• Flashing alternately **green**/yellow • Green • Yellow • Flashing **red**	• BIOS Power On Self Test after switch on • PC running • Idle status • Watchdog status display: active
RUN/STOP/L1	• Green • Yellow	• Can be controlled by user program • Can be controlled by controller program e.g. WinAC
ERROR/ L2	• Yellow • Flashing **red**	• - • Can be controlled by user program or controller program e.g. WinAC
MAINT/ L3	• Yellow • Flashing **red**	• - • Can be controlled by controller program e.g. WinAC

Notes:

Simatic IPC 477E - Integrated Multi-touch Panel

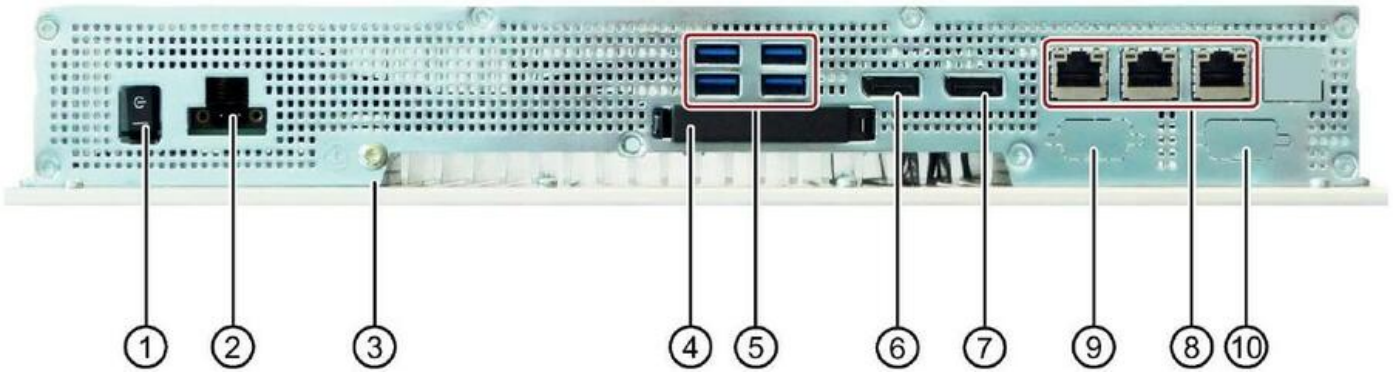
Front view:



Available sizes:
15“, 19“, 22“, 24“

Bottom view:



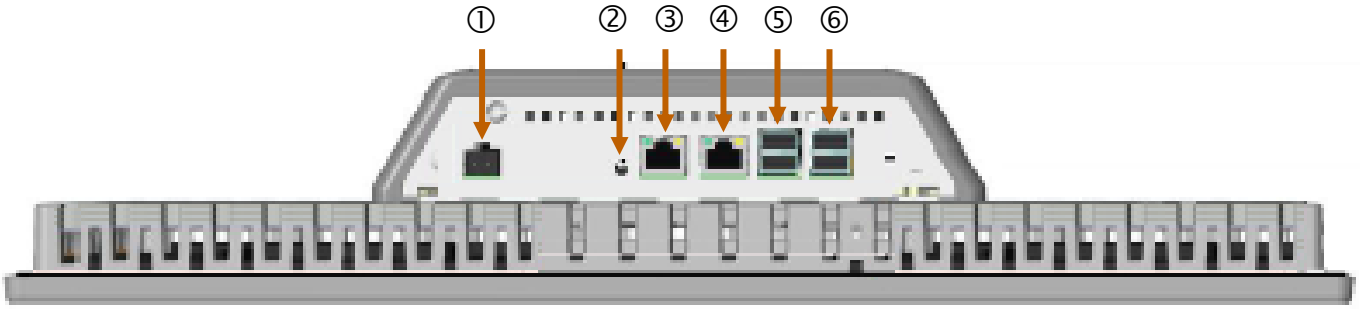


Item	Interface	Function
①	-	Power off/on
②	X80	24V DC Supply
③	-	Protective earth connection
④	X50	CFast card slot
⑤	X60 – X63	4 x USB 3.0
⑥	X70	DPP Display-Port
⑦	X71	DPP Display-Port
⑧	X1 P1 – P3	3 x PROFINET/Industrial Ethernet
⑨	X30	COM1 - not available
⑩	X31	COM2 - not available

Notes:

ITC 1200/1500/1900/2400: Interfaces

Item	Interface
1	X80 power supply connector
2	Factory settings button
3	X1 P1 PROFINET LAN
4	X1 P2 Ethernet LAN
5	X61, X62 USB type A
6	X63, X64 USB type A



Restoring Factory Settings:

- switch the power off
- press and hold the ***Factory Settings*** button
- switch the power.
- hold the ***Factory Settings*** button for 3 seconds.

A message is briefly displayed and the device restarts automatically.

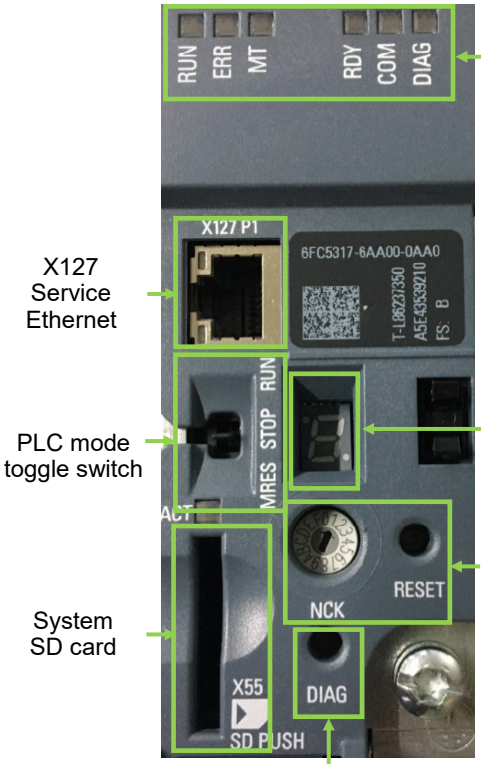
Notes:

NCU 1750/1760 Diagnostic Interfaces and Switches

The NCU diagnostic features are found on the lower front face and consists of a 7-segment display, 6 status-LEDs and reset/commissioning/mode switches.



The 7 segment display/switches etc. are located beneath the flip down cover.



7-Segment display

The 7-Segment display is located behind the lower cover of the control unit.

It serves as a status display during commissioning

During normal operation a **6**. is displayed with the point flashing. The “RDY” LED will be green and constant.

An **8** indicates that the fan is defective or that the Control Unit is operating without a fan.

During the boot phase the display changes to indicate the different stages of the boot process.

7-segment display during power up - Load BIOS

Display	Meaning
0	Before loading the Master Boot Record (MBR) from the internal SSD
1	Before starting the Master Boot Record (MBR) codes
2	Master Boot Record (MBR) code started
3	Second stage of the boot loader started before reading the configuration file
4	Before loading the kernel
5	Before starting the kernel

Booting the basic system

Display	Meaning
1	Basic initializations
2	Cleaning up and preparing the file system
3	The name is set and the loopback interface configured.
4	Starting the system logging. Starting the CFS class 1, provide the network interfaces.
5	Initializing the network interfaces and the firewall
6	Starting the time server (ntpd)
7	Starting the SSH server (generate a host key if one is still not available).
8	Starting the FTP server
9	Starting the TCU services (TFTP, VNC)
P	Partitioned: The system performs a partitioning operation.

Notes:

Error Messages

These error messages are shown on the 7-segment display as continuously running text (***ticker***).

Note
When submitting a query to the SIEMENS Technical Support Service: In addition to the message shown on the 7-segment display, additional information regarding the status LEDs and the operating state (continuous operation, switch-on phase, restart, etc) can provide important information on resolving your problem.

Display	Meaning	Troubleshooting
C	Crash or kernel panic: The Linux system no longer responds	1. Restart the NCU. 2. Execute the DIDAC diagnostics option. This function saves all the diagnostic data to a ZIP file; SINUMERIK data ⇨ user ⇨ sinumerik ⇨ didac. 3. Contact the SIEMENS Technical Support Service and transfer the zipped diagnostics data.
Copy FAIL	The FWH installation was unsuccessful.	
F	Failure	
F.	The point is continuously lit and no longer flashes: NCK crash	
Lxx=yyy	Internal error when starting NCK, ***xx*** is the line number in the mcsystem.ini, ***yyy*** is a unique error code for the code position.	
Merr	Mixer Error: An error has occurred.	• Check the log file under ***/var/log/initncu.log***. • Contact SIEMENS Technical Support service.
NO ARC	The CNC software package is missing or cannot be found on the SD card.	• Boot the NCU using the SINUMERIK service system, and check the partition using the ***Check storage integrity*** diagnostics option. • Replace the SD card if it is shown to be defective. • Contact SIEMENS Technical Support Service if the SD card is not defective.
NO SD	The operating system cannot detect an SD card.	• Insert the SD card. • Check that the SD card is correctly inserted. • Check whether the SD card is defective.
NO SSD	The internal storage medium was not detected.	Contact SIEMENS Technical Support Service.
PART FAIL	Error when repartitioning	Boot the NCU using the SINUMERIK service system, and check the partition using the ***Check storage integrity*** diagnostics option.
SD Error <n>	<n> = single-digit code • SD Error 1: System card reader is not available in HW info.. • SD Error 2: Card reader does not respond; HW problem. • SD Error 3: SD card is missing or does not contain an ext4 partition. • SD Error 4: fsck (file system check) failed with uncorrectable errors. • SD Error 5: Failed to mount the file system. A write or read error has occurred on the SD card.	• It is possible that the SD card is defective and must be replaced. • Check using the diagnostics option: ***Check storage integrity*** of the SD card
SIGErr=<nn>	<nn> = double-digit code. An error has occurred during the signature check. The error can occur if parts of the CNC software package were changed or replaced.	Check whether the original CNC software package is installed and has not been modi□fied. • Install the original CNC software package. • Contact SIEMENS Technical Support Service.
unPAC FAIL	A problem has occurred when reading, unzipping or writing the data.	Boot the NCU using the SINUMERIK service system, and check the partition using the ***Check storage integrity*** diagnostics option. • Contact SIEMENS Technical Support Service if the diagnostics check was unsuccessful.
	The archive with the CNC software is possibly corrupt or unreadable.	Reinstall the CNC software if the SD card is OK.
6	With flashing point: Operating system has crashed where the NCK is no longer clocking	Restart the NCU.
8	7-segment display is 8 and all LEDs are lit yellow: Fan module is not detected.	Check whether the fan module is inserted or is defective.

Notes:

System Errors

System errors are indicated by a combination of the “RDY” LED and the 7-segment display.

“RDY” LED fault description:

RDY LED	Meaning
Flashing red @ 2Hz	Permanent error: System stopped (See right for 7-segment display)
Flashing red/ orange @ 2Hz	Temporary error: System may continue with restricted functions. (See right for 7-segment display)

7_segement display in relation to “RDY” LED:

7 segment display	Meaning
Permanent error	
C	“Crash”: Operating system crash (similar to a “Blue screen” in Windows) Further details can be found in the system log-file.
P	“Partition”: Error when repartitioning
Temporary error	
E	“Error”: A read or write error has occurred on the System CF Card (an additional point is displayed to indicate a “write error). It is possible that the CF card is defective.
F	“Full”: The System CF Card is full, it is possible that not all services will have started. When booting, the error is displayed for 1 minute; the system will then continue to boot, however, it can be assumed that problems will occur.

NCU 17x0 Status LEDs: Functionality/Description

RDY (Ready) LED	
Colour	**Explanation**
Red	**There is at least one fault (e.g. RESET, Watchdog monitoring etc.), or the control unit is in booting.**
Red Flashing (2Hz)	**Not assigned (SINAMICS boot error).**
Orange	**NCU ramp up.**
Orange Flashing (0.5Hz)	**Updating the firmware of the connected DRIVE-CLiQ components**
Orange Flashing (2Hz)	**Firmware update is complete for components. Wait for POWER ON for the components in question.**
Green	**NC running and in cyclic operation.**
Green/ Orange or Red/Orange (1Hz)	**DRIVECLiQ component recognition active: (Parameter p0124[0] = 1).**

COM (Communication) LED	
Colour	**Explanation**
Off	**CP not ramped up/started.**
Green	**Communication via CP possible (RUN operating state)**
Green Flashing	**Communication possible (no configuration)**
Orange	**CP was stopped via PLC/TIA. No communication.**
Orange Flashing	**Configuration received and transition to the RUN operating state.**
Green	**NC running and in cyclic operation.**
Red Flashing	**DRIVECLiQ component recognition active: (Parameter p0124[0] = 1).**

Notes:

PLC Status LEDs: Functionality/Description

RUN LED (run/stop)	ERR LED (error)	MT LED (maintenance)	Explanation
OFF	OFF	OFF	Supply voltage at the CPU not present or too low.
OFF	Red Flashing	OFF	An error has occurred.
Green	OFF	OFF	The CPU is in the RUN operating state.
Green	Red Flashing	OFF	Diagnostic event pending.
Green	OFF	Orange	A plant maintenance request is pending. The affected hardware must be validated/replaced within a short interval.
			Active force job
			PROFIenergy pause
Green	OFF	Orange Flashing	A plant maintenance requirement is pending. The affected hardware must be validated/replaced within a foreseeable interval.
			Configuration faulty
Orange	OFF	Orange Flashing	Firmware update completed successfully.
Orange	OFF	OFF	CPU is in STOP operating state.
Orange	Red Flashing	Orange Flashing	The program on the SIMATIC Memory Card causes a fault
			CPU defective
Orange Flashing	OFF	OFF	CPU performs internal activities during STOP, e.g. ramp-up after STOP.
			Downloading the user program from the SIMATIC Memory Card
Orange /Green Flashing	OFF	OFF	Start-up (transition from RUN → STOP)
Orange /Green Flashing	Red Flashing	Orange Flashing	Start-up (booting the CPU)
			Test the LEDs during start-up
			LED flash test

DIAG LED (Diagnostics)	
Colour	**Explanation**
Green	Everything OK.
	Software-related fault is pending, e.g. at PLC, HMI, Linux. Generate the diagnostic data with the ***Diag*** button and contact the Hotline.
	The data backup operation initiated with the ***Diag*** button is running.
	HW-related tests fail, e.g. memory test during switch on. If no change occurs after multiple switch off/on, replace the device.

Notes:

NCK and PLC general reset

NCK switch functionality:

Switch Position	Function
0	NCK normal run-up, setting in normal operation
1	General reset (NCK run-up with default values)
2	NCK runs up with the data that was saved at the last shutdown.[1: see below]
7	Debug mode (NCK is not started); the SSH port tcp/22 is open at X130
8	IP address for X130 of the NCU appears on the 7-segment display; the NCK is not started; the PLC remains at STOP
9	SSH port tcp/22 is open at X130 (without any entry in basesys.ini file)
All others	Not relevant.

1: System (NCK) previously not switched off, but exited (sc stop nck, sc stop all)

NCK Factory Settings:

When commissioning the NCU for the first time it is necessary to achieve a defined initial state of the whole system, a general NCK and PLC reset must be performed.

1. Turn the NCK commissioning switch to position **1.**
2. Press the ***Reset*** button.
3. Wait for the 7 segment display to show **6**
4. Turn the NCK commissioning switch to position 0
5. NCK Factory settings are now loaded**.**

PLC switch functionality

Switch Position	Function
RUN	Set the PLC to ***RUN*** Mode
STOP	Set the PLC to ***STOP*** Mode
MRES	Memory reset on boot up (MRES)

General reset of the PLC to factory settings using the toggle switch:

While the control system is running up, the PLC can be restored to the initial delivery condition with the toggle switch position ***MRES***. When doing this, the PLC SD card is cleared and the remanent data initialized. The PLC IP address assigned via the TIA Portal is also cleared.

Proceed as follows to restore the PLC to the factory settings (note: steps 5 & 6 must be completed within 3 seconds) :

1. Set the PLC toggle switch to ***STOP*** (middle switch position)
2. Press the ***Reset*** button on the NCU
3. Wait until ***ERR-LED*** flashes ***orange*** and 7-segment display shows **5**
4. Move the toggle switch to position ***MRES*** and hold until the ***ERR-LED*** extinguishes
5. Release the toggle switch
6. Within ***3*** seconds set the toggle switch again to ***MRES*** and release -> ***RUN-LED*** indicates the overall reset by flashing rapidly
7. Set the PLC to ***RUN*** (upper switch position) ***RUN LED*** will change to permanent ***orange.***
8. Finished - the PLC now has factory settings.

Note: if the RUN LED fails to flash quickly at step 6, then the procedure needs to be repeated.

Notes:

DIAG Button

Pressing the DIAG button has various functions depending on when it is pressed while running up.

Requirements: Version (functional state) = FS: J

Diagnostic data of the system bootloader

1. Insert a USB stick with FAT file system and directory "beiboot.log" into the NCU (X125/X135). Press the *Reset* button.
2. A few seconds after switching on and while executing the "System bootloader", press the DIAG button on the NCU.
3. The DIAG-LED flashes yellow slowly.
4. If directory ***beiboot-log*** was found, file ***YYYYMMDD-HHMMSS-CCC.txt*** is created, which contains the content of the usual detailed screen and the syslog from ***System-bootloader***.
5. Once the operation has been completed, the ***DIAG-LED*** flashes ***green*** quickly

Diagnostic data of Operate (Didac)

Once the NCU has run-up, during operation, you can press the DIAG button at any time. This generates file ***out_20xx-xx-xxTxx_xx_xx.7z*** in directory/card/user/sinumerik/didac. This file contains all the relevant NCU diagnostic data.

A007: END

Notes:

A009

SINUMERIK ONE

Digital Inputs and Outputs

Module Description:

The Sinumerik ONE NCU/PPU has interfaces for the connection of drive/NC inputs and outputs (I/O).

Connections used within the drive group components are known as BiCos.

Certain connections can be configured either as an output or an input.

The digital I/O are used to handle various signals, for example:

- Drive Enables
- Probe Inputs
- Automatic Line Contactor Control
- External Reference Mark (Bero)
- Hand wheel connection

This module shows the pin assignment of each connector and shows several examples of how a BiCo is assigned to a function, including probes and line contactors, and how NCK I/O are used in a machining program.

Module objective:

Upon completion of this module you will be able to locate the interface connectors and configure individual connections to perform certain functions.

Content:

NCU: Interface Description: X122/X132/X142

PPU: Interface Description: X122/X132/X242/X252/X143

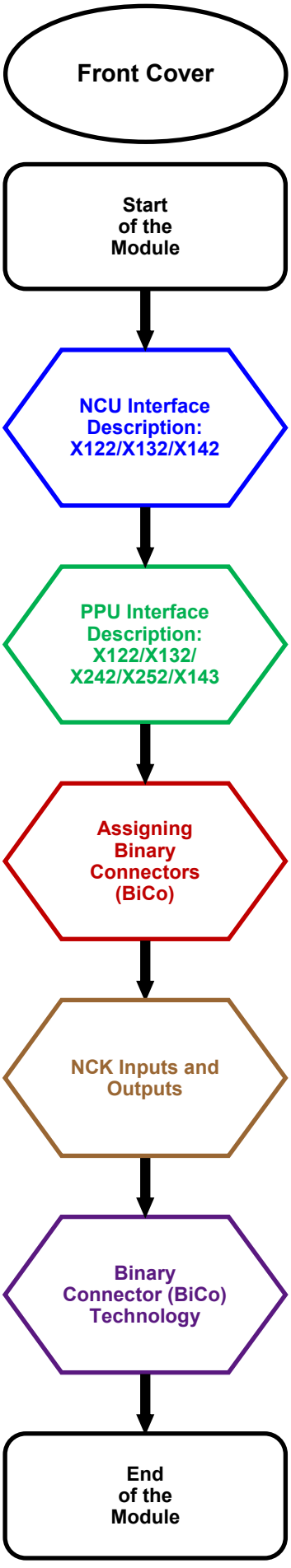
Binary Connector (BiCo) Technology

Assigning Binary Connectors (BiCo)

NCK Inputs and Outputs

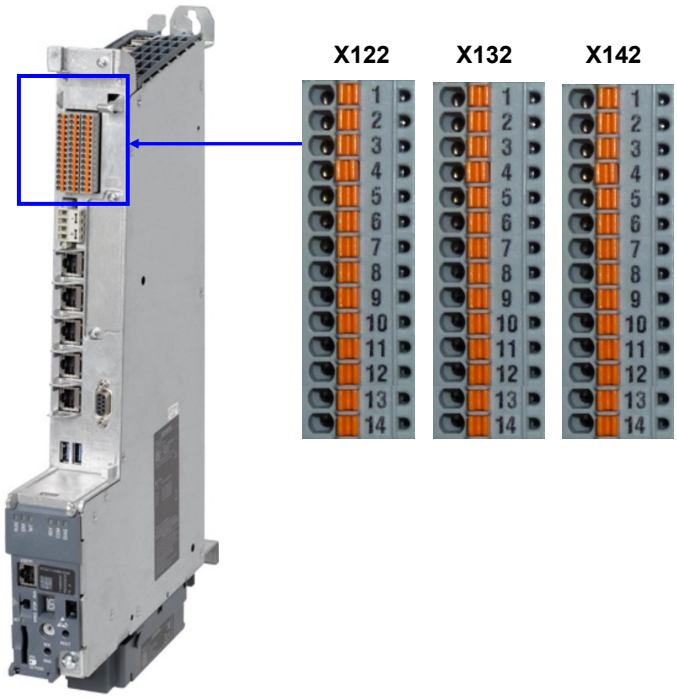
Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
NCU Interface Description: X122/X132/X142
PPU Interface Description: X122/X132/ X242/X252/X143
Assigning Binary Connectors (BiCo)
NCK Inputs and Outputs
Binary Connector (BiCo) Technology
End of the Module

NCU 1740/1750/1760 Drive/NC IO: X122, X132 & X142

X122, X132 & X142 digital I/O interfaces are located beneath the flip-down cover of the NCU.



Sinamics Inputs and Outputs - X122/X132

Sinamics inputs and outputs are also referred to as Binary Connectors (BiCos)
The Binary Connectors (BiCos) of X122 and X132 are used to assign Sinamics inputs/outputs to a specific function.

Examples:

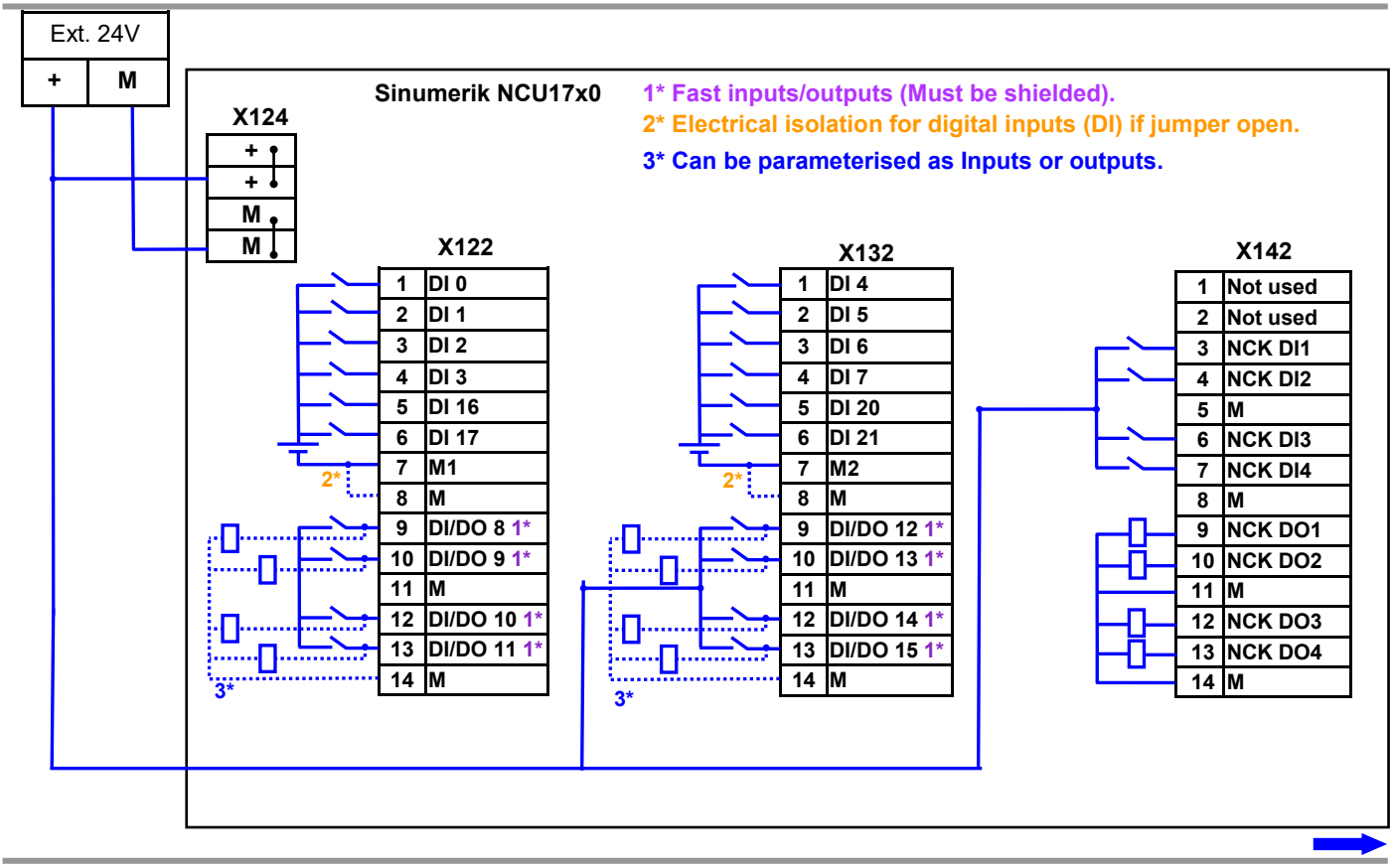
- Sinamics drive enables (drives ready, OFF3 fast stop).
- Measuring system external zero marker
- Automatic line contactor control (3 phase supply to line module)
- Workpiece/tool probes.

BiCo assignment is facilitated via specific CU/drive parameters.
Certain BiCos are automatically assigned to a function by default.
Remaining BiCos can be assigned by the manufacturer.

The status of the terminals is transferred to CU parameter r722. The system monitors r722 and carries out the required action, depending on the status.

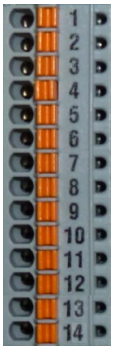
NCK Inputs and Outputs - X142

The NCK inputs and outputs are used within programs and cycles to influence the machining operation.
The input status can be monitored, while the outputs can be monitored and set.



Notes:

X122 - Default BiCo assignments



Colour scheme
Pre-assigned by default
Free/recommended use
Free
Ground
P24V DC

X122 Terminal	Sinamics Signal	CU status/ source parameter	Function/Default Setting	Equipment	BiCo configuration parameters
1	DI 0	r722.0	OFF1 for DriveCLiQ line modules (ALM, SLM >=16Kw)	NCU 1740 X132.10	Infeed p840
			Infeed ready for non DriveCLiQ SLM (5 and10Kw versions)	SLM X21.1	Drive p864
2	DI 1	r722.1	OFF3 drive enable - Fast Stop ***1)**	E Stop circuit	Drive p849
3	DI 2	r722.2	SS1 – Group 1 SINAMICS Safety Integrated ***2)**	User defined	Drive p9620 (all drives in the group)
4	DI 3	r722.3	SS1 – Group 2 SINAMICS Safety Integrated ***2)**	User defined	Drive p9620 (all drives in the group)
5	DI 16	r722.16	Freely available	User defined	-
6	DI 17	r722.17	Freely available	User defined	-
7	G1 (Ground)	-	Ground for terminals 1 to 6	-	-
8	M (Ground)		Ground for 9, 10, 12, 13		-
9	DI/DO 8 (fast)	r722.8/ r747.8	STO – Group 1 (Input) SINAMICS Safety Integrated ***2)**	User defined	CU p738/Drive p9620
10	DI/DO 9 (fast)	r722.9/ r747.9	STO – Group 2 (Input) SINAMICS Safety Integrated ***2)**	User defined	CU p739/Drive p9620
11	M1 (Ground)		Ground for terminals 9, 10, 12, 13	-	-
12	DI/DO 10 (fast)	r722.10/ r747.10	Bero 1 - external zero mark (Input)	User defined	CU p740/Drive p495
13	DI/DO 11 (fast)	r722.11/ r747.11	Measuring probe 1: (Input) ***3)** General MD13210=0/1	Probe 1	CU p741 CU p680/Drive p488
14	M (Ground)		Ground for terminals 9, 10, 12, 13	-	-

Notes:

***1)** X122.2 OFF3 drive enable - Fast Stop. Can be set permanently to 1 if Safety Integrated is enabled.

***2)** Drive based Safety Integrated - refer to additional sources for detailed information.

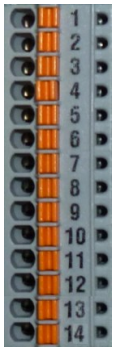
***3)** Centralised measuring: General MD13210=1. set probe input via CU p680 (Default)
Decentralised measuring: General MD13210=0. set probe input via Drive p488

Digital I/O Switching times (µs)

Switching	Inputs	Outputs	Bi-directional	
			Inputs	Outputs
Low to high	50	150 to 400	10	50
High to Low	150	75 to 100	150	300

Notes:

X132- Default BiCo assignments



Colour scheme
Pre-assigned by default
Free/recommended use
Free
Ground
P24V DC

X132 Terminal	Sinamics Signal	CU status/ source parameter	Function/Default Setting	Equipment	BiCo configuration parameter
1	DI 4	r722.4	Freely available	User defined	-
2	DI 5	r722.5	Freely available	User defined	-
3	DI 6	r722.6	Freely available	User defined	-
4	DI 7	r722.7	Line contactor feedback signal ***1)**	Line contactor	LM p860
5	DI 20	r722.20	Freely available	User defined	-
6	DI 21	r722.21	Freely available	User defined	-
7	G2 (Ground)	-	Ground for terminals 1 to 6	-	-
8	M (Ground)	-	Ground for terminals 9, 10, 12, 13		-
9	DI/DO 12	r722.12/ r747.12	Infeed operational - Output ***2)**	User defined	CU p742/LM p863.0
10	DI/DO 13	r722.13/ r747.13	Infeed ready to switch on - Output ***2)**	PPU X122.1	CU p743/LM r899.0
11	Ground (M)		Ground for terminals 9, 10, 12, 13	-	-
12	DI/DO 14	r722.14/ r747.14	Line contactor control - Output ***1)**	Line contactor	CU p744/LM p863.1
			Bero 2 - external zero mark (Input)	User defined	CU p744/Drive p495
			2nd OFF 2 (Input)	User defined	CU p744/Drive p845
13	DI/DO 15	r722.15/ r747.15	**Measuring probe 2: (Input) *3) General MD13210=0/1**	Probe 2	CU p745 CU p680/Drive p489
14	Ground (M)		Ground for terminals 9, 10, 12, 13	-	-

Notes:

***1)** X132.4/X132.12 DRIVECLiQ Line Module only. Freely available if no contactor in use.

***2)** X132.9/X132.10 DRIVECLiQ Line Module only. Freely available if no contactor in use.

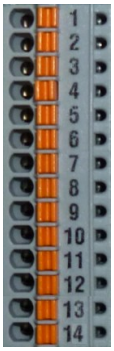
***3)** Centralised measuring: General MD13210=1. set probe input via CU p680 (Default)
Decentralised measuring: General MD13210=0. set probe input via Drive p489

Digital I/O Switching times (µs)

Switching	Inputs	Outputs	Bi-directional	
			Inputs	Outputs
Low to high	50	150 to 400	10	50
High to Low	150	75 to 100	150	300

Notes:

X142-NCK I/O



X142	Signal	Function/Default Setting
1	Reserved	Reserved
2	Reserved	Reserved
3	**NCK DI 0**	Digital Input $A_IN[1]
4	**NCK DI 1**	Digital Input $A_IN[2]
5	GND	Ground
6	**NCK DI 2**	Digital Input $A_IN[3]
7	**NCK DI 3**	Digital Input $A_IN[4]
8	GND	Ground
9	**NCK DO 0**	Digital Output $A_OUT[1]
10	**NCK DO 1**	Digital Output $A_OUT[2]
11	GND	Ground
12	**NCK DO 2**	Digital Output $A_OUT[3]
13	**NCK DO 3**	Digital Output $A_OUT[4]
14	GND	Ground

Digital I/O Switching times (µs)

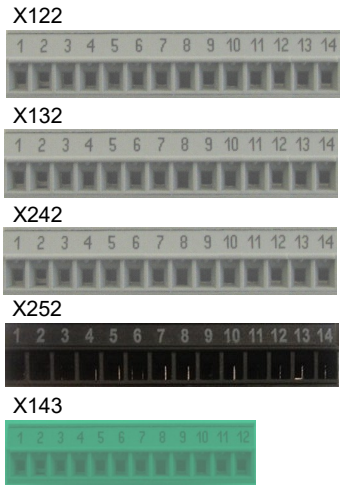
Switching	Inputs	Outputs
Low to high	50	150 to 400
High to Low	150	75 to 100

Notes:

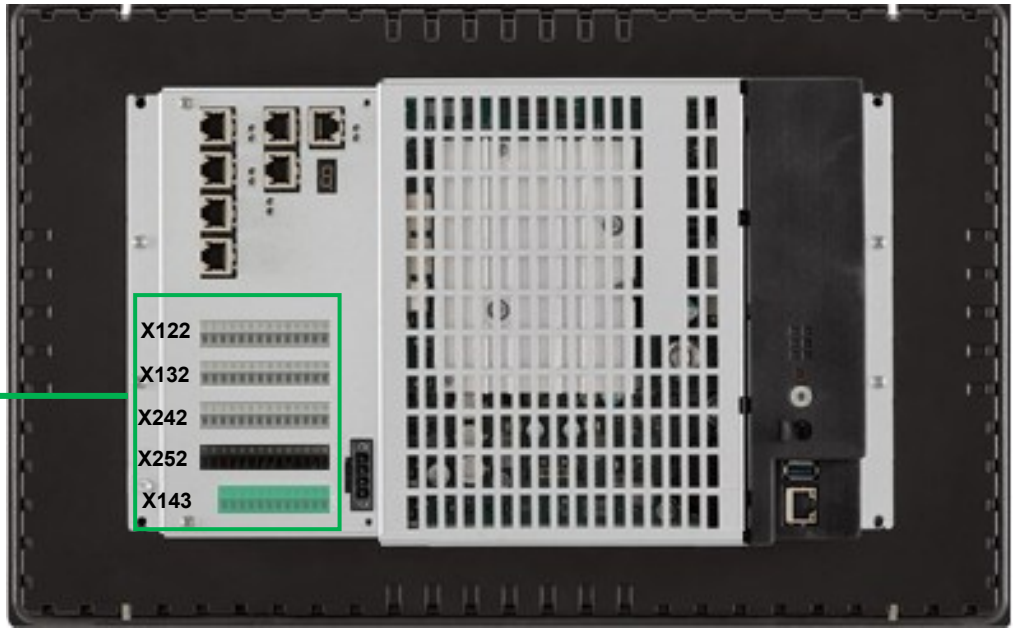
PPU1740 Interface Description: X122/X132/X242/X252/X143

X122/X132 - Sinamics digital I/O
X242 - NC digital I/O
X252 - NC/Analogue digital I/O
X143 - Handwheels (x 2)

Interface locations:



PPU 1740 rear



Sinamics Inputs and Outputs - X122/X132
Sinamics inputs and outputs are also referred to as Binary Connectors (BiCos)
The Binary Connectors (BiCos) of X122 and X132 are used to assign Sinamics inputs/outputs to a specific function.

Examples:
- Sinamics drive enables (drives ready, OFF3 fast stop).
- Measuring system external zero marker
- Automatic line contactor control (3 phase supply to line module)
- Workpiece/tool probes.

There are 6 input terminals and 4 bi-directional terminals.

BiCo assignment is facilitated via specific CU/drive parameters.
Certain BiCos are automatically assigned to a function by default.
Remaining BiCos can be assigned by the manufacturer.

The status of the terminals is transferred to CU parameter r722.

NCK Inputs and Outputs - X242/X252

X242 - 4 digital inputs and 4 digital outputs

X252 - 4 digital inputs and 4 digital outputs ***plus*** analogue output for spindle speed control (+/- 10V DC max)
If the analogue spindle function is active, two digital outputs are reserved for enabling and direction control.

The NCK inputs and outputs can used within programs and cycles to influence the machining operation.
The state of an input can be monitored via the part programs/ cycles (read only).
The output status can monitored and set via the part program. Cycles (read/write).

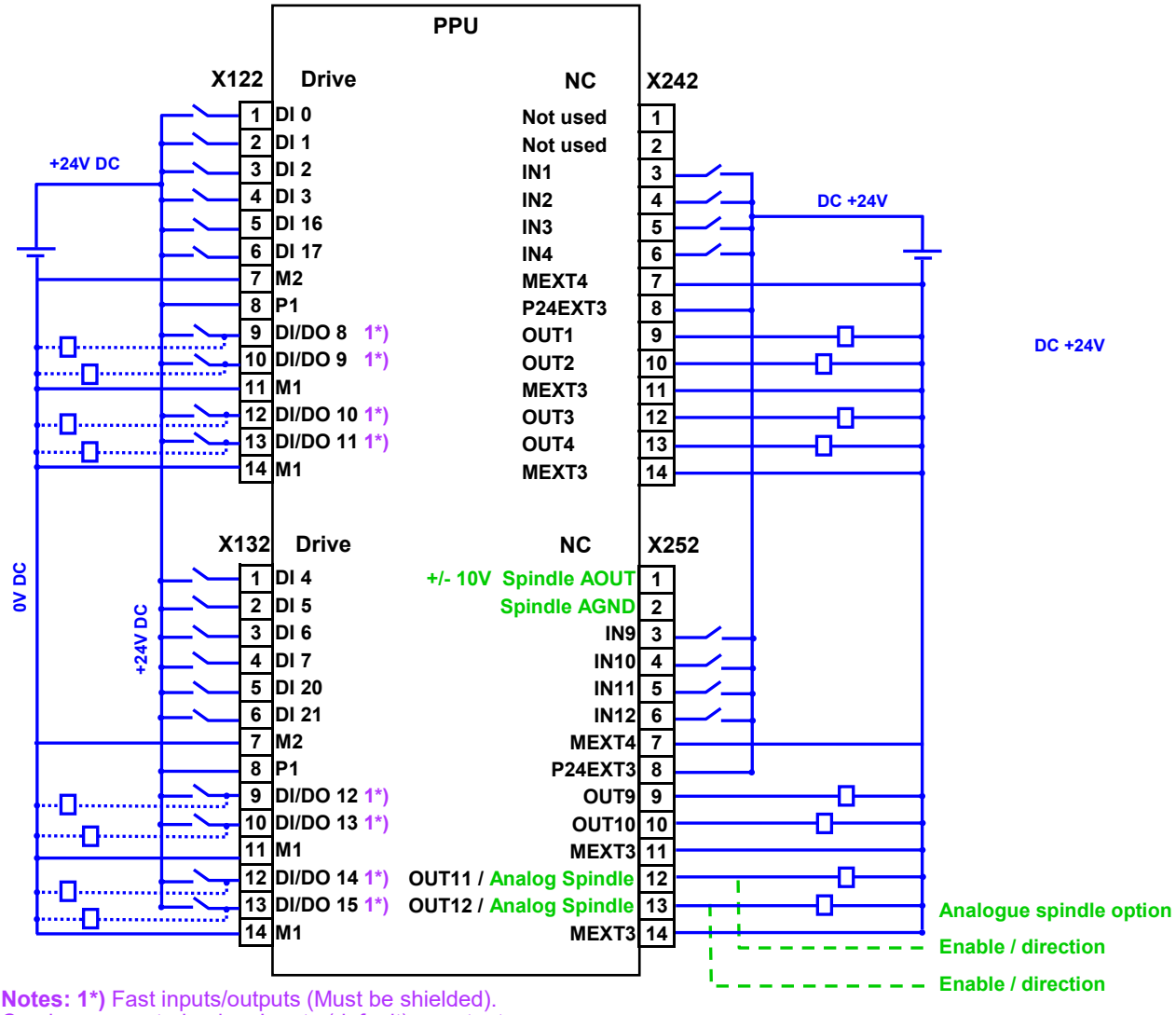
Handwheels - X143
Two handwheels can be connected via interface X143 using differential (RS422) or asymmetrical (5V TTL) signal transfer.

Digital I/O Switching times (µs)

Switching	Inputs	Outputs	Bi-directional	
			Inputs	Outputs
Low to high	50	150 to 400	10	50
High to Low	150	75 to 100	150	300

Notes:

Connection diagram for X122, X132, X242, X252:



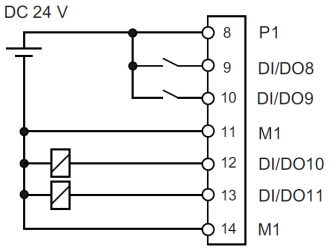
Notes: 1*) Fast inputs/outputs (Must be shielded).
Can be parameterised as Inputs (default) or outputs.

Functionality of X252 pins 12 & 13:
The functionality of X252 pins 12 & 13 changes depending on whether an analogue spindle is declared.

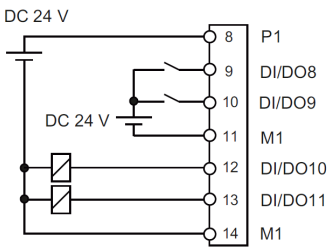
X252	Analogue spindle **is** declared	Analogue spindle **is not** declared
Pin 12	Controller enable/positive direction for analogue spindle option	Digital Output $A_OUT[11]
Pin 13	Controller enable/negative direction for analogue spindle option	Digital Output $A_OUT[12]

X122/X132 24V DC Connection Requirements.

Recommended connection: Inputs and outputs are operated with the same 24 V supply.



Not recommended: Inputs and outputs on different supplies.



Notes:

X122 - Default BiCo assignments



X122 Terminal	Sinamics Signal	CU status/ source parameter	Function/Default Setting	Equipment	BiCo configuration parameters
1	DI 0	r722.0	OFF1 for DriveCLiQ line modules (ALM, SLM >=16Kw)	PPU 1740 X132.10	Infeed p840
			Infeed ready for non DriveCLiQ SLM (5 and10Kw versions)	SLM X21.1	Drive p864
2	DI 1	r722.1	OFF3 drive enable - Fast Stop *1)	E Stop circuit	Drive p849
3	DI 2	r722.2	SS1 – Group 1 SINAMICS Safety Integrated *2)	User defined	Drive p9620 (all drives in the group)
4	DI 3	r722.3	SS1 – Group 2 SINAMICS Safety Integrated *2)	User defined	Drive p9620 (all drives in the group)
5	DI 16	r722.16	Freely available	User defined	-
6	DI 17	r722.17	Freely available	User defined	-
7	M2 (Ground)	-	Ground for terminals 1 to 6	-	-
8	P1(+24V DC)		+ 24V DC Supply		-
9	DI/DO 8 (fast)	r722.8/ r747.8	STO – Group 1 (Input) SINAMICS Safety Integrated *2)	User defined	CU p738/Drive p9620
10	DI/DO 9 (fast)	r722.9/ r747.9	STO – Group 2 (Input) SINAMICS Safety Integrated *2)	User defined	CU p739/Drive p9620
11	M1 (Ground)		Ground for terminals 9, 10, 12, 13	-	-
12	DI/DO 10 (fast)	r722.10/ r747.10	Bero 1 - external zero mark (Input)	User defined	CU p740/Drive p495
13	DI/DO 11 (fast)	r722.11/ r747.11	Measuring probe 1: (Input) *3) General MD13210=0/1	Probe 1	CU p741 CU p680/Drive p488
14	M1 (Ground)		Ground for terminals 9, 10, 12, 13	-	-

Colour scheme
Pre-assigned by default
Free/recommended use
Free
Ground (M)
P24V DC

Notes:

***1)** X122.2 OFF3 drive enable - Fast Stop. Can be set permanently to 1 if Safety Integrated is enabled.

***2)** Drive based Safety Integrated - refer to additional sources for detailed information.

***3)** Centralised measuring: General MD13210=1. set probe input via CU p680 (Default)
Decentralised measuring: General MD13210=0. set probe input via Drive p488

Notes:

X132-Sinamics I/O



X132 Terminal	Sinamics Signal	CU status/ source parameter	Function/Default Setting	Equipment	BiCo configuration parameter
1	DI 4	r722.4	Freely available	User defined	-
2	DI 5	r722.5	Freely available	User defined	-
3	DI 6	r722.6	Freely available	User defined	-
4	DI 7	r722.7	Line contactor feedback signal ***1)**	Line contactor	LM p860
5	DI 20	r722.20	Freely available	User defined	-
6	DI 21	r722.21	Freely available	User defined	-
7	M2 (Ground)	-	Ground for pins 1 to 6	-	-
8	P1 (+24V DC)	-	+ 24V DC Supply		-
9	DI/DO 12	r722.12/ r747.12	Infeed operational - Output ***2)**	User defined	CU p742/LM p863.0
10	DI/DO 13	r722.13/ r747.13	Infeed ready to switch on - Output ***2)**	PPU X122.1	CU p743/LM r899.0
11	Ground (M)		Ground for terminals 9, 10, 12, 13	-	-
12	DI/DO 14	r722.14/ r747.14	Line contactor control - Output ***1)**	Line contactor	CU p744/LM p863.1
			Bero 2 - external zero mark (Input)	User defined	CU p744/Drive p495
			2nd OFF 2 (Input)	User defined	CU p744/Drive p845
13	DI/DO 15	r722.15/ r747.15	**Measuring probe 2: (Input) *3) General MD13210=0/1**	Probe 2	CU p745 CU p680/Drive p489
14	Ground (M)		Ground for terminals 9, 10, 12, 13	-	-

Colour scheme
Pre-assigned by default
Free/recommended use
Free
Ground (M)
+24V DC

Notes:

***1)** X132.4/X132.12 DRIVECLiQ Line Module only. Freely available if no contactor in use.

***2)** X132.9/X132.10 DRIVECLiQ Line Module only. Freely available if no contactor in use.

***3)** Centralised measuring: General MD13210=1. set probe input via CU p680 (Default)
Decentralised measuring: General MD13210=0. set probe input via Drive p489

Notes:

X242-NCK I/O



X242	Signal	Function/Default Setting
1	Reserved	Reserved
2	Reserved	Reserved
3	**NCK DI 0**	Digital Input $A_IN[1]
4	**NCK DI 1**	Digital Input $A_IN[2]
6	**NCK DI 2**	Digital Input $A_IN[3]
7	**NCK DI 3**	Digital Input $A_IN[4]
5	GND	Ground
8	P3	+24V DC common
9	**NCK DO 0**	Digital Output $A_OUT[1]
10	**NCK DO 1**	Digital Output $A_OUT[2]
11	GND	Ground
12	**NCK DO 2**	Digital Output $A_OUT[3]
13	**NCK DO 3**	Digital Output $A_OUT[4]
14	GND	Ground

X252-NCK I/O



X252	Signal	Function/Default Setting
1	AOUT	Output for analogue spindle +/- 10V
2	AGND	Ground for analogue output
3	IN9 NCK DI 9	Digital Input $A_IN[9]
4	IN10 NCK DI 10	Digital Input $A_IN[10]
5	IN11 NCK DI 11	Digital Input $A_IN[11]
6	IN12 NCK DI 12	Digital Input $A_IN[12]
7	MEXT4 Ground (M)	Ground for pins 3 to 6
8	P24EXT3	+24V DC common
9	OUT9 NCK DO 9	Digital Output $A_OUT[9]
10	OUT10 NCK DO 10	Digital Output $A_OUT[10]
11	MEXT3 Ground (M)	Ground for pins 9, 10, 12, 13
12	OUT11 NCK DO 11	Digital Output $A_OUT[11] / Analogue spindle enable/direction
13	OUT12 NCK DO 12	Digital Output $A_OUT[12] / Analogue spindle enable/direction
14	MEXT3 Ground (M)	Ground for pins 9, 10, 12, 13

Notes:

X143-Hand-wheel connections



X143	Signal	Function/Default Setting
Pin 1	5V DC (P5)	+5V DC common
Pin 2	0V DC (M)	Ground
Pin 3	1A	Hand-wheel 1 signal A
Pin 4	–1A	Hand-wheel 1 signal –A
Pin 5	1B	Hand-wheel 1 signal B
Pin 6	–1B	Hand-wheel 1 signal –B
Pin 7	5V DC (P5)	+5V DC common
Pin 8	0V DC (M)	Ground
Pin 9	2A	Hand-wheel 2 signal A
Pin 10	–2A	Hand-wheel 2 signal –A
Pin 11	2B	Hand-wheel 2 signal B
Pin 12	–2B	Hand-wheel 2 signal –B

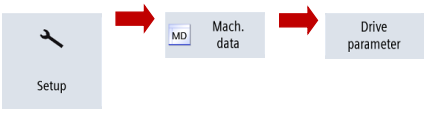
Notes:

Assigning Binary Connectors (BiCo)

Assigning a BiCo to a function is achieved with CU/drive parameters.

OFF3 Bico Example - The default setting for OFF3 can be checked as an example.
The OFF3 signal is input on X122.2, the status of this signal is displayed via CU r0722.1.
In order for the signal to have any effect on the operation of the drives It has to be referenced to each drive.
This is achieved through p0849 for each drive.

Example: How the OFF3 signal is assigned to X axis p0849



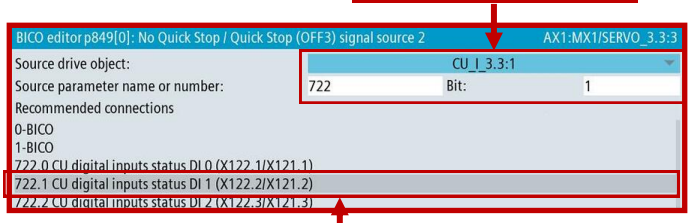
Drive parameters		AX1:MX1/SERVO_3.3:3
p845[0]	No coast-down / coast-down (OFF2) signal source 2	1-BICO
p848[0]	No Quick Stop / Quick Stop (OFF3) signal source 1	SERVO_3.3:3:r2090.2
p849[0]	No Quick Stop / Quick Stop (OFF3) signal source 2	CU_I_3.3:1:r722.1
p852[0]	Enable operation/inhibit operation	SERVO_3.3:3:r2090.3

Use the ***Select*** key to give a more detailed view.



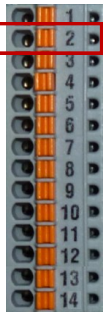
The information regarding the source of the OFF3 signal (X122.2) can be viewed.

Source of OFF3 signal:
CU parameter r722.1



Physical connection of the OFF3 signal:
CU X122 pin 2 = r722.1

X122.2 - OFF 3



Assignment of probe inputs:

Probe 1 is assigned to X122.13 by default.

In the following example, probe 2 will be assigned to BiCo connector X132.13.

The standard probing function of the Sinumerik ONE is ***Centralised measuring (General machine data 13210 = 0)***.



General machine data			
13210	$MN_MEAS_TYPE	0	po

The probes are assigned with CU parameter p680.

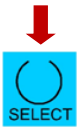


Control Unit parameter		CU_I_3.3:1
p680[0]	Central measuring probe input terminal	[3] DI/DO 11 (X122.1...
p680[1]	Central measuring probe input terminal	[0] No measuring pro...
p680[2]	Central measuring probe input terminal	[0] No measuring pro...
p680[3]	Central measuring probe input terminal	[0] No measuring pro...
p680[4]	Central measuring probe input terminal	[0] No measuring pro...
p680[5]	Central measuring probe input terminal	[0] No measuring pro...
p680[6]	Central measuring probe input terminal	[0] No measuring pro...
p680[7]	Central measuring probe input terminal	[0] No measuring pro...

Probe 1 one already assigned to X122.13 via p680[0]

Example: Assigning probe 2 to X132.13

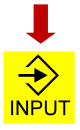
- Cursor to p680[1]
- Use the ***Select*** key to expand the parameter.
- Cursor to **[6] DI/DO 15 (X132.12/X131.5)**
- Accept the selection by pressing the ***Input*** key



Cursor to p680[1] and press ***Select.***

Control Unit parameter		CU_I_3.3:1
p680[0]	Central measuring probe input terminal	[3] DI/DO 11 (X122.1...
p680[1]	Central measuring probe input terminal	[0] No measuring pro...
p680[2]	Central measuring probe input terminal	[0] No measuring pro...
p680[3]	Central measuring probe input terminal	[0] No measuring pro...
p680[4]	Central measuring probe input terminal	[0] No measuring pro...
p680[5]	Central measuring probe input terminal	[0] No measuring pro...
p680[6]	Central measuring probe input terminal	[0] No measuring pro...
p680[7]	Central measuring prot	[0] No measuring pro...
p681	Central measuring prot	[1] DI/DO 9 (X122.10/X121.8)
p682	Central measuring prot	[2] DI/DO 10 (X122.12/X121.10)
p684	Central measuring prot	[3] DI/DO 11 (X122.13/X121.11)
r685	Central measuring prot	[4] DI/DO 13 (X132.10/X131.2)
r686[0]	Central measuring prot	[5] DI/DO 14 (X132.12/X131.4)
r686[1]	Central measuring prot	[6] DI/DO 15 (X132.13/X131.5)
r686[2]	Central measuring prot	[7] DI/DO 8 (X12 9/X121.7)
r686[3]	Central measuring prot	[8] DI/DO 12 (X1 2.9/X131.1)

	CU_I_3.3:1:r2090.0	
	CU_I_3.3:1:r2050.2	
	[0] Measurement wit...	
	0H	
	0	
	0	
	0	
	0	



Cursor to **[6] DI/DO 15 (X132.12/X131.5)** and press ***Input.***

Notes:

Assigning the probe inputs

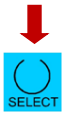
Checking the functionality of the Bico terminals:

X132.9, 10, 12 & 13 can be used as either inputs or outputs. This can be changed by the setting of **CU parameter p728**.

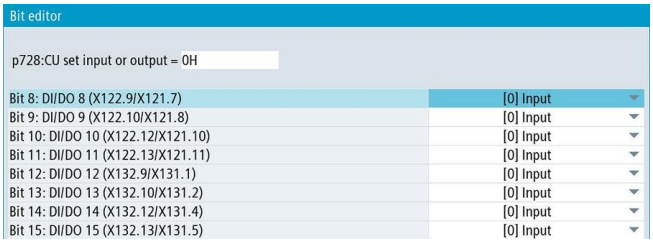
The default setting is for inputs which is correct for probing operations.







The default setting of 0 means that the terminals are set as ***inputs.***



Probe Types:

The probe switch type - normally open or normally closed - has to match the setting in General Machine Data:

- **13200[0]** (Probe 1)
- **13200[1]** (Probe 2)





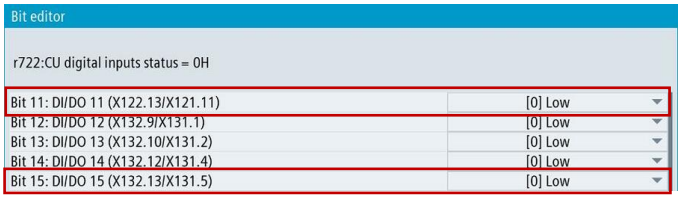
MD13200 = 0 (Default) Normally open contact - the probe signal is **0v** in the non-deflected state.

MD13200 = 1 Normally closed contact - the probe signal is **24v** in the non-deflected state.

Testing the Probe Signals:

The probe signals can be checked via CU parameter r722.





Save the changes:



Probe Status

The status of the probes can also be seen via the PLC interface:
DB10.DBX107.0 = Probe 1
DB10.DBX107.1 = Probe 2

The behaviour of the signal is influenced by MD13200.

The PLC interface signals indicate whether the probe has been deflected, not the presence of 24V dc at the probe terminal

The following table shows the relationship between the setting of machine data 13200 and the PLC interface status.

13200[x]	X122.13/X132.13	DB10.DBX107.x
0	0v	0
0	24v	1
1	0v	1
1	24v	0

Notes:

The probes can be tested via a command in the part program using the following commands:

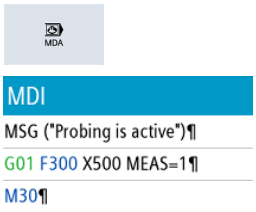
MEAS: Measurement with deletion of distance to go.

MEAS=1 Probe 1, detect rising signal.
MEAS=2 Probe 2, detect rising signal.
MEAS=-1 Probe 1, detect falling signal.
MEAS=-2 Probe 2, detect falling signal.

Example for using probe 1 via part program:
G01 F300 X200 MEAS=1

Example program for testing probe 1:

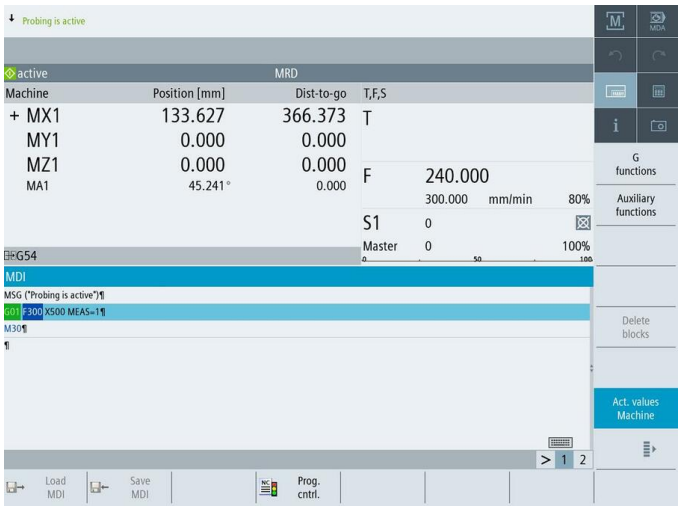
Machine data 13200[0] = 0



After pressing the ***Cycle start*** button, the axis will move in the programmed direction. The switch should be depressed before it reaches the specified position.



X axis moving prior to the probe button being pressed.



The position of the axis when the probe is triggered is read into variables:

$AA_MM[x] - machine coordinate position.
$AA_MW[x] - workpiece coordinate position.

[x] = Axis name.

Assigning an external zero marker:

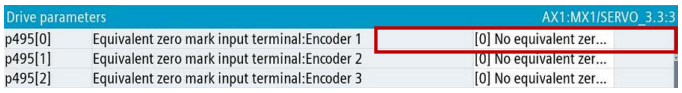
A BERO is a switch which is used instead of the zero marker of an encoder/linear scale.
This is common, for example, with spindles using gear boxes or pullies which don't have 1:1 ratios.

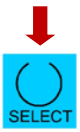
Activating the BERO option:

To make this active the relevant Drive Machine Data (p495) has to be set.

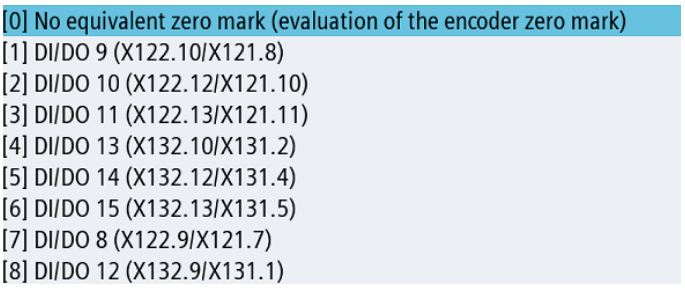
Ensure that the correct drive is selected, i.e. Spindle.





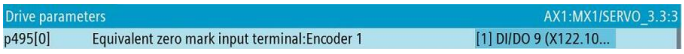


Select the input terminal to which the BERO switch is connected (X122.10).





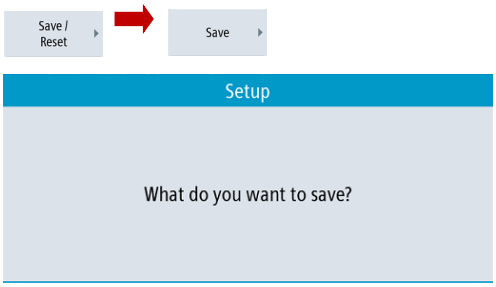
The assignment of the Drive Machine Data p495 and the BERO switch input is now complete.



Notes:

Saving the Changes:

The changed data needs to be saved, use the following soft-key sequence to save the drive data:

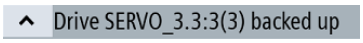


Current DO: saves the parameters of the currently displayed drive object (CU, NX or Servo).

Or

Drive device: saves the parameters of the currently displayed drive device (CU/NX) and all associated objects.

The status of the save process is displayed directly above the horizontal soft-keys:



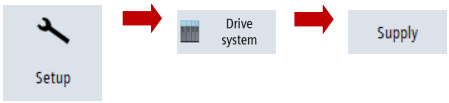
Configuring the automatic line contactor.

The automatic line contactor function is used to ensure correct switching of the line supply to the line modules (DriveCLiQ versions only)

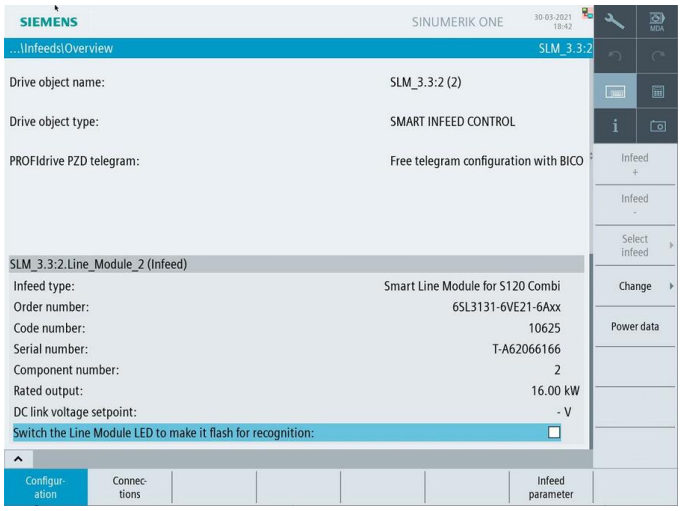
The following example shows how to configure the line contactor coil and line contactor feedback signals.
The feedback (Input) signal will be assigned to X132 pin 4 (Sinamics digital input 7).
The coil signal (Output) will be assigned to X132 pin 12 (Sinamics digital input 14).

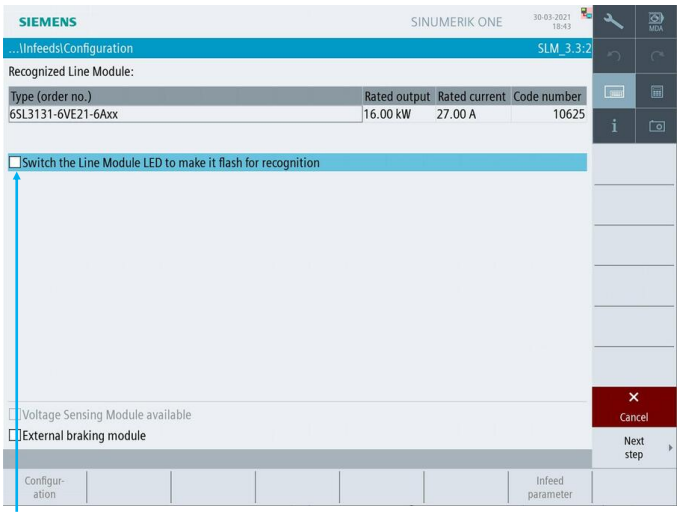
Configuring the feedback signal:

The table of X132 shows information about the source and destination. For the feedback signal, the source is CU parameter r0722.7. The destination of this signal is LM parameter p0860.
Use the following soft-key selection to locate the set-up pages.







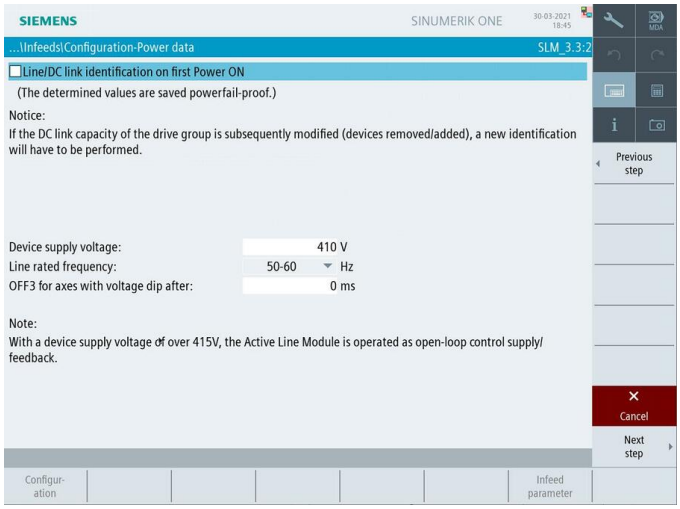


For identification purposes, the LED of the module can be made to flash.

Notes:

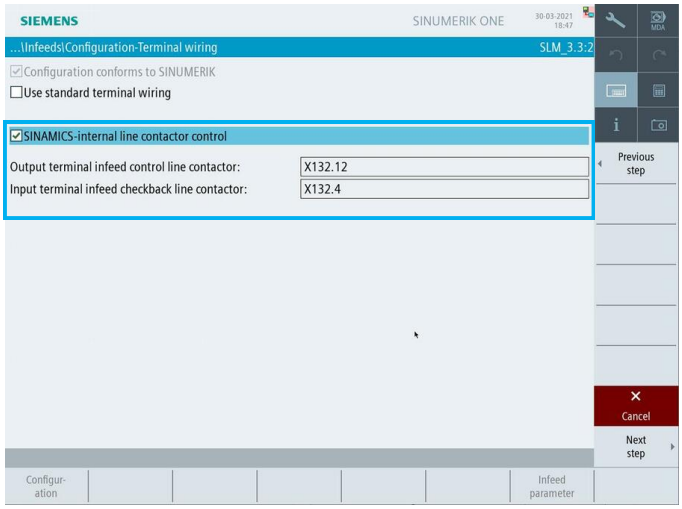
Next step

The supply voltage can be manually entered or detected automatically.



Next step

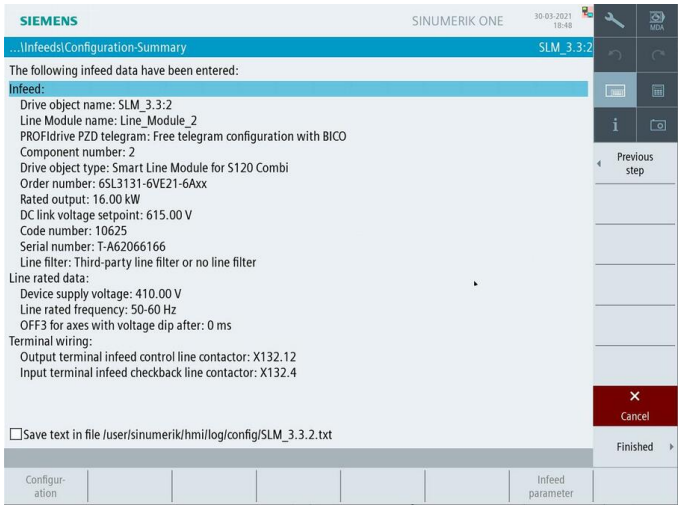
Select the check box for SINAMICS-internal line contactor.



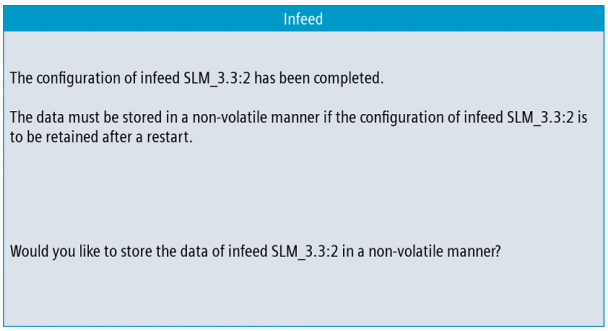
The BiCo connectors for line contactor operation are pre-assigned and cannot be changed.
X132.12 Output to line contactor coil.
X132.4 Input from line contactor.

Next step

A summary of the line supply data is displayed, this can be saved to a text file if required.



Finished

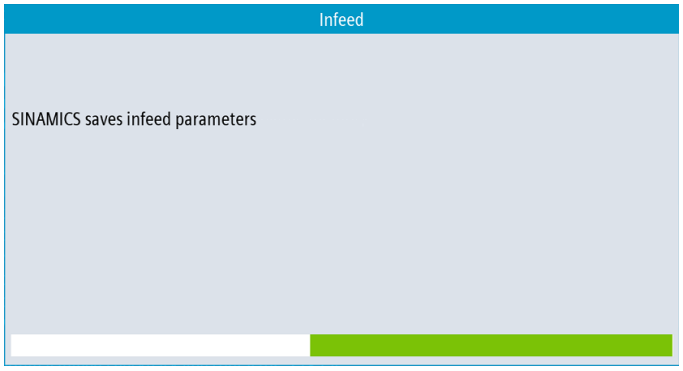


The new BiCo settings need to be saved, select the ***Yes*** softkey.

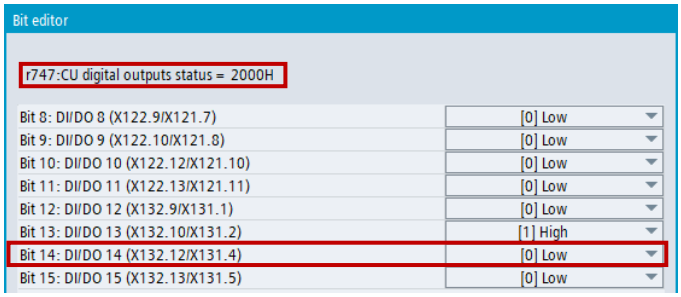
Yes

Notes:

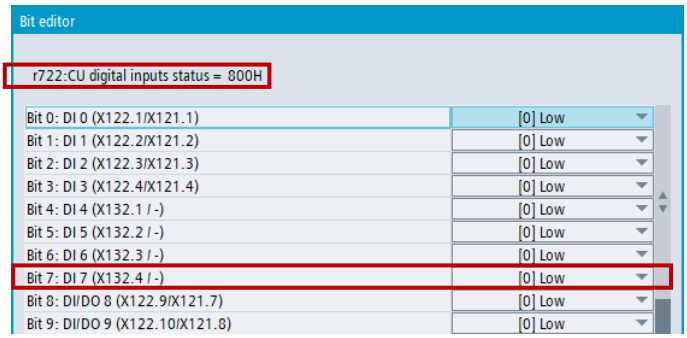
The changes are now saved, as indicated by the dialogue box which disappears once completed.



CU parameter r747 bit 14 - status of the line contactor output signal:

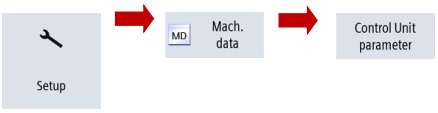


CU parameter r722 bit 7 - status of the line contactor feedback signal:



Checking the Line Contactor Signals

The status of the line contactor feedback signal can be checked via CU parameter r722.7 (X132.4)
The status of the line contactor coil digital output can be checked using CU parameter r747.14 (X132.12)



NCK Digital Inputs and Outputs

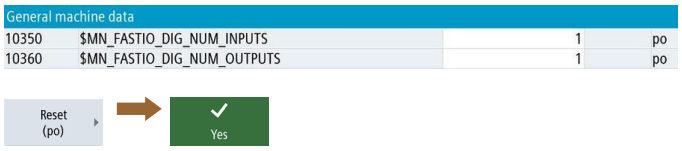
NCK digital I/O can be addressed directly from machining programs.

To address Inputs, for example Digital NCK input 2, the following has to be entered in the part program: **$A_IN[2]**

Inputs cannot be written to (Read only) but their status can be checked and used to influence the machining operation.

Addressing outputs is done in a similar manner, example address for digital output 2: **$A_OUT[2]**

The digital outputs are not active by default. General machine data **10360 $MN_FASTIO_DIG_NUM_OUTPUTS** is used to activate the outputs..
The value entered determines how many output bytes will be available.



Example: Setting an NCK output via a part program.

The NCK outputs can be set directly from a part program. The following example will show how NCK digital input 2 is set and unset.

Select ***MDA*** mode.



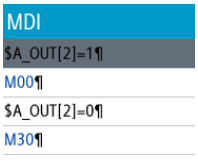
Type in the following text:

$A_OUT[2] = 1 ← Set NCK output 2 to 1

M00 ← Optional program stop, program will halt until Cycle Start is pressed.

$A_OUT[2] = 0 ← Set NCK output 2 to 0

M30 ← Program end



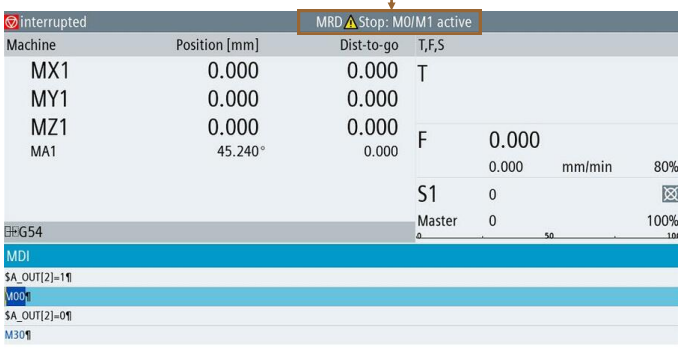
Press Cycle Start:



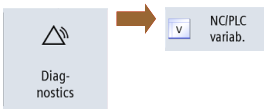
Notes:

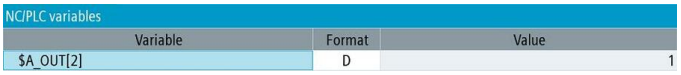
The program will stop at the ***M00*** command (optional stop) and a message will be displayed.

Optional stop active.



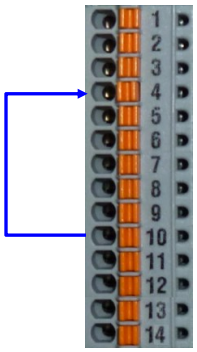
The status of the PLC variable can be checked.





Note: Interface X142 on training equipment has been wired so that X142 pin 4 ($A_IN[2]) and X142 pin 10 ($A_OUT[2]) are linked.

This allows demonstration of the use of an NCK input in MDA mode ands also shows that, X142 pin 10, does actually output a physical voltage.



X142 pin 4 - $A_IN[2]

X142 pin 10 - $A_OUT[2]

Example: Using NCK inputs in a part program

The NCK inputs can also be used in the part program. The following program checks the status of NCK input 9 and displays a relevant message.

Type the following into the MDA screen:

$A_OUT[2] = 1 ← Setting NCK output 2 to 1 will also set NCK input 2.

IF $A_IN[2] == 1 GOTOF LABEL1 ← If NCK input 2 = 1 jump forward to LABEL1.

MSG (“NCK INPUT 2 IS LOW”) ← Message line showing that the input is low.

M00 ← M00 Program stops until cycle start is pressed.

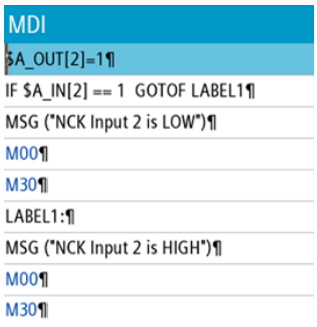
M30 ← M30 Program end.

LABEL1: ← LABEL1 used for the jump command (GOTOF).

MSG (“NCK INPUT 2 IS HIGH”) ← Message line for showing that the input is high.

M00 ← M00 Program stops until cycle start is pressed.

M30 ← M30 Program end.

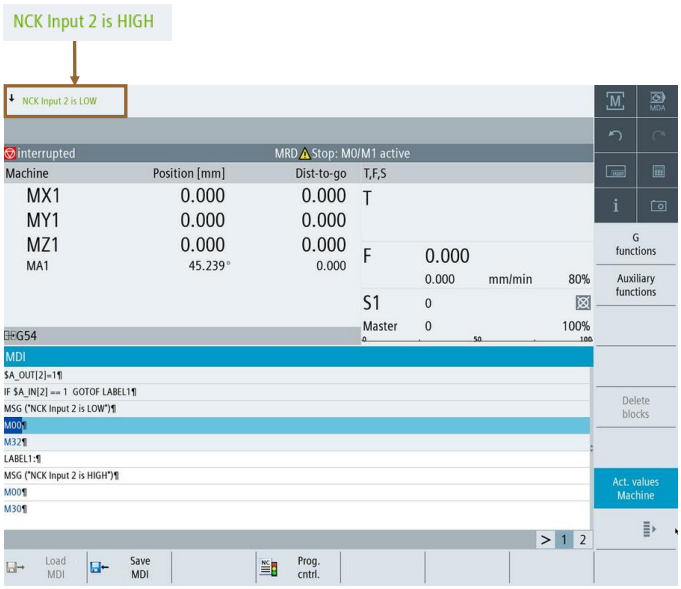


Start the program via the ***Cycle Start*** button.



Notes:

When the line, ***IF $A_IN[2] == 1 GOTOF LABEL1***, is reached, the system will check the status of digital input $A_IN[2] for a value of 1. If this is true the program will jump to LABEL1 and continue from there.
The program message will be displayed and the cycle will pause because of the M00 instruction.



The status of the PLC variables can be checked:

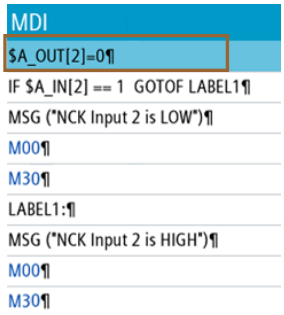
NC/PLC variables

Variable	Format	Value
$A_OUT[2]	D	1
$A_IN[2]	D	1

When the cycle start is pressed again the program will see the M30 instruction and the program will be reset, the message will clear.



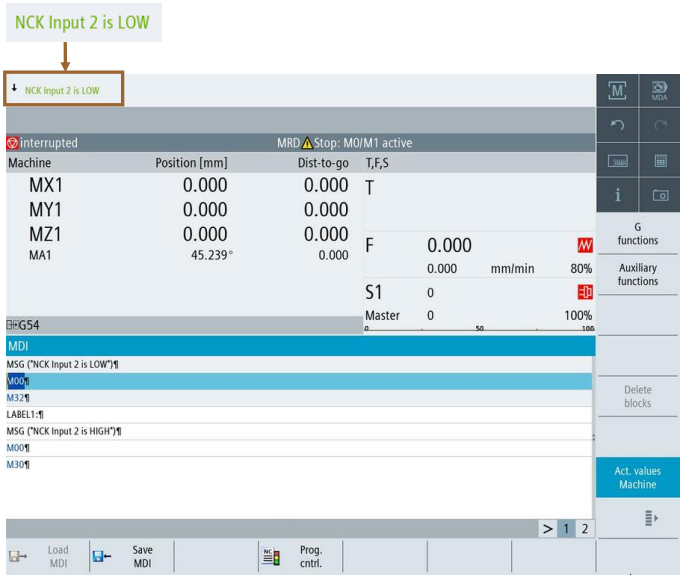
The program now needs to be edited so that NCK output 2 is set to 0.



Start the program via the ***Cycle Start*** button.



The program will not process the GOTOF command because of the status of Input 2.
Instead, it will process the next block which is the message ***NCK Input 2 is LOW*** the program will halt due to the M00 command.



The status of the PLC variables can be checked:

NC/PLC variables

Variable	Format	Value
$A_OUT[2]	D	0
$A_IN[2]	D	0

When the cycle start is pressed, again, the program will read the M30 instruction and the program will be reset. The message will clear.



Notes:

Binary Connector (BiCo) Technology:

Interconnecting signals:

Every CU/Servo contains a large number of interconnectable input/output variables and internal control variables.

BiCo technology (Binector Connector Technology) allows the drive to be adapted to a wide variety of requirements.

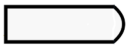
Digital and analog signals, which can be interconnected as required by means of BiCo parameters, are identified by the prefix BI, BO, CI, or CO in their parameter name.

Binectors/Connectors:

BI: Binector Input, BO: Binector Output

A binector is a digital (binary) signal without a unit which can assume the value 0 or 1.

Binectors are subdivided into binector inputs (signal sink) and binector outputs (signal source).

Abbreviation	Symbol	Name	Description
BI		Binector input (signal sink)	Can be interconnected to a binector output as source. The number of the binector output must be entered as a parameter value.
BO		Binector output (signal source)	Can be used as a source for a binector input.

CI: Connector Input, CO: Connector Output

A connector is a digital signal, e.g. in 32-bit format. It can be used to emulate words (16 bits), double words (32 bits) or analogue signals. Connectors are subdivided into connector inputs (signal sink) and connector outputs (signal source).

Abbreviation	Symbol	Name	Description
CI		Connector input (signal sink)	Can be interconnected to a connector output as source. The number of the connector output must be entered as a parameter value.
CO		Connector output (signal source)	Can be used as a source for a connector input.

Interconnecting signals using BiCo technology

To interconnect two signals, a BiCo input parameter (signal sink) must be assigned to the required BiCo output parameter (signal source). The following information is required for connecting a binector/connector input to a binector/connector output:

Connectors with no index: drive object ID and Parameter number

Binectors: Drive Object ID, Parameter number and Bit number e.g.:

SERVO_3.3:3 P849[0] BiCo **1**:722.1
SERVO_3.3:3 P848[0] BiCo **63**:2090.2

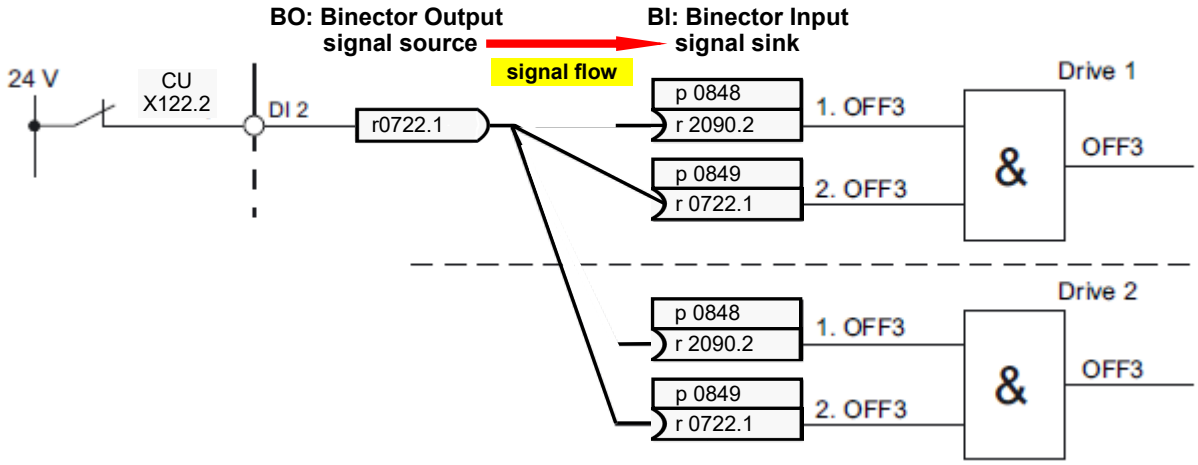
Drive Object ID 63 = own drive object

Notes:

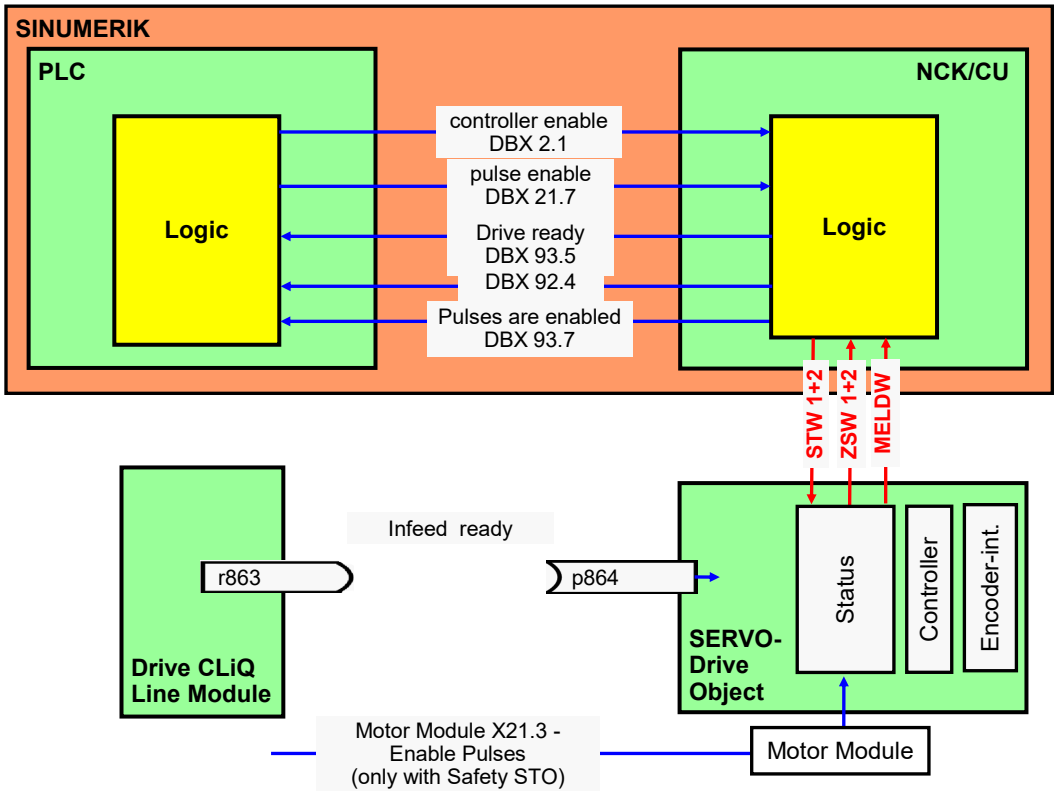
Connection of OFF3 to several drives

Each drive has two ***OFF3*** binector inputs: ***1. OFF3*** and ***2. OFF3***.
The ***1. OFF3*** is an internal signal which monitors the status of the individual drive via the the DRIVECLiQ Telegram.
The ***2. OFF3*** is an external signal and is connected as standard via terminal X122.2 on the Control Unit.

The two signals are processed via AND logic to STW1.2 (OFF3).



The signal flows of the signal source now (digital input of the CU) to the target (on the function)



Notes:

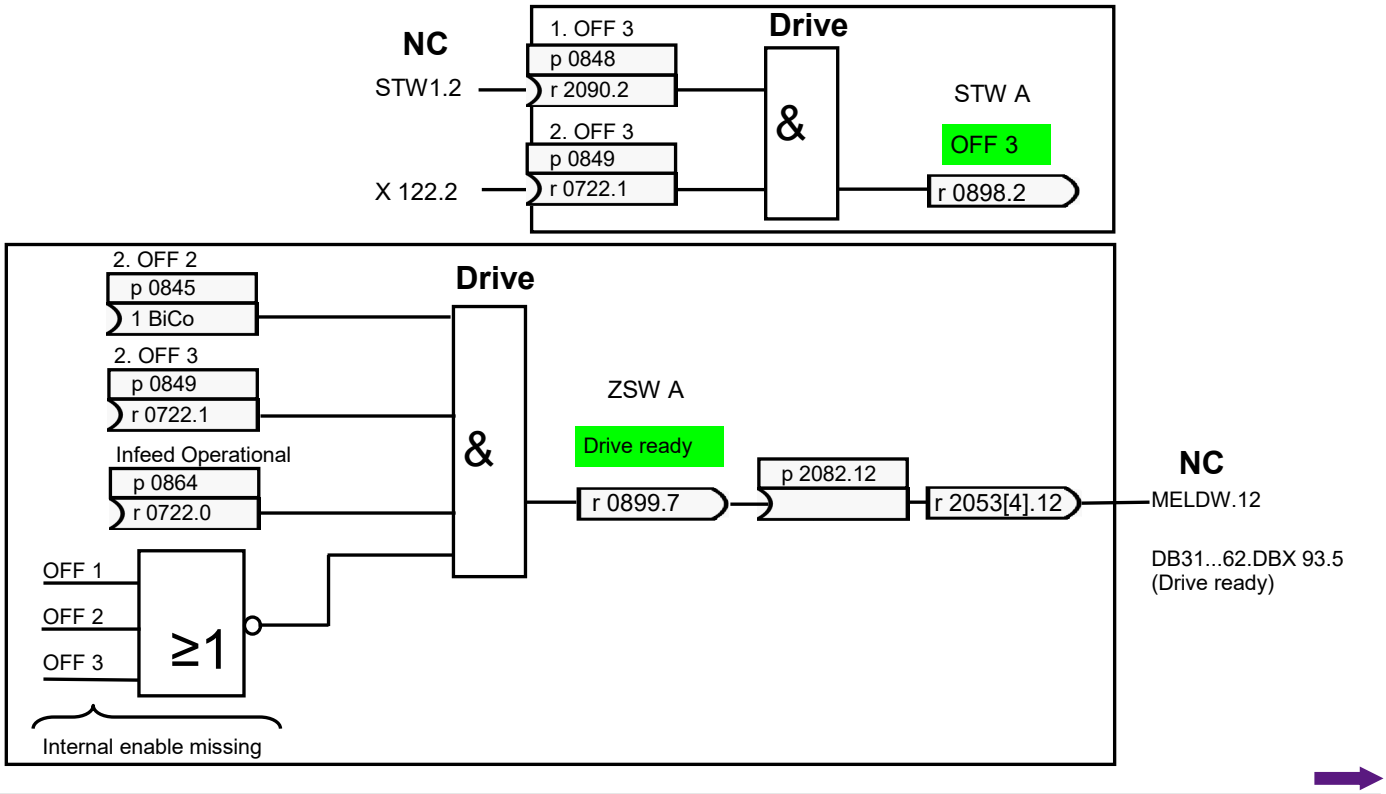
Telegram 136: Application class 4 with DSC (Dynamic Servo Control) - position control with servo cycle time (Default 0.125ms)

Telegram 136 - Application Class 4		
PZD1	STW1	ZSW1
PZD2	NSOLL_B	NIST_B
PZD3		
PZD4	STW2	ZSW2
PZD5	MOMRED	MELDW
PZD6	G1_STW	G1_ZSW
PZD7	G2_STW	G1_XIST1
PZD8	XERR	
PZD9		G1_XIST2
PZD10	KPC	
PZD11		G2_ZSW
PZD12	M_VST	G2_XIST1
PZD13	DSC_STW	
PZD14	res	G2_XIST2
PZD15	T_SYMM	
PZD16		AIST_GLATT
PZD17		MSOLL_GLATT
PZD18		PIST_GLTT
PZD19		ITIST_GLATT

Drive Telegram Control Words (STW), Status Words (ZSW) & Message Words (MELDW)

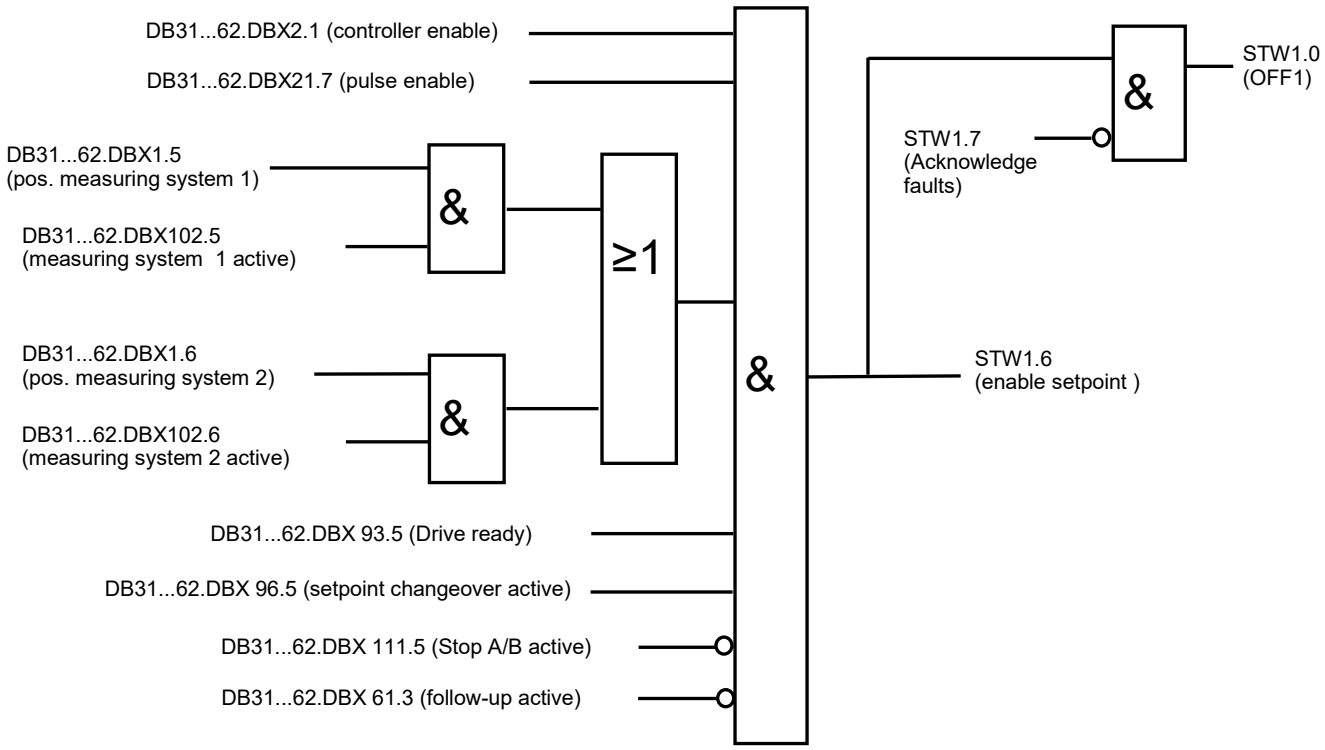
Signal	Description	
STW1/2	Control word PB - telegram - R 2090 / R 2094	NC → DO
ZSW1/2	Status word PB - telegram - R 2053 [0/3]	DO → NC
MELDW	Message word PB - telegram - R 2053 [4]	DO → NC
STWA	Internal control word - sequence control - R 0898	DO
ZSWA	Internal status word - sequence control - R 0899	DO

OFF 3 and Drive ready (schematic diagram)



Notes:

OFF1 and enable setpoint (schematic diagram)



A009: END

Notes:

A009: END

Notes:

A011

SINUMERIK ONE

Axis and Drive Diagnostics

Module Description:

In the event of a problem with the machine it may be necessary to check the status of an axis/drive. The information obtained from these status screens can be very useful to technical personnel and help speed up the fault finding process.

Module objective:

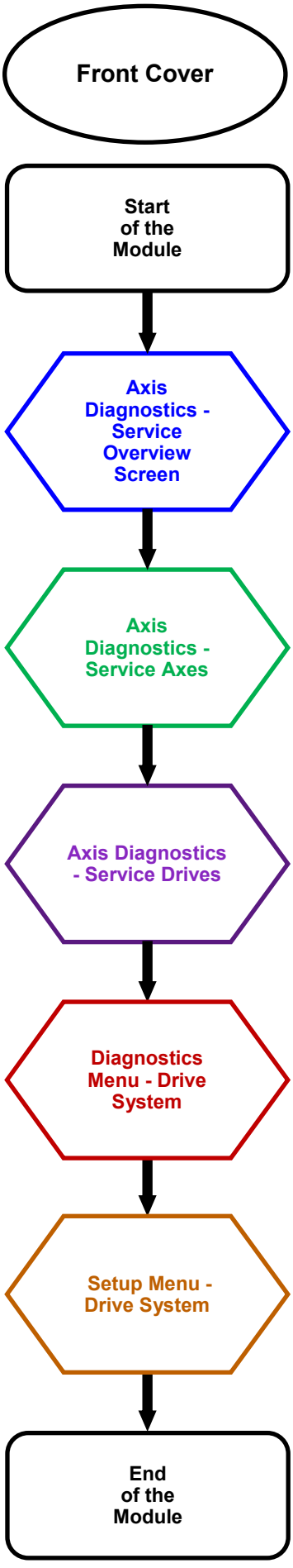
Upon completion of this module you will be able to use the Axis & Drive diagnostic functions.

Content:

Axis Diagnostics - Service Overview Screen
Axis Diagnostics - Service Axes
Axis Diagnostics - Service Drives
Diagnostics Menu - Drive System
Setup Menu - Drive System

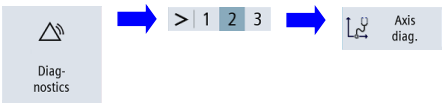
Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

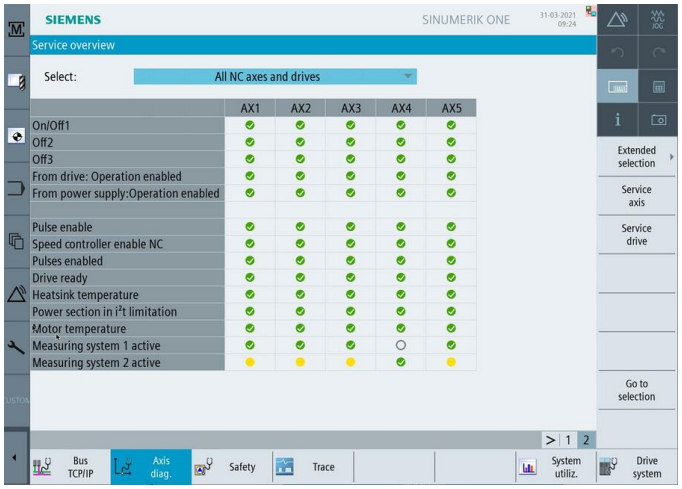

Front Cover
Start of the Module
Axis Diagnostics - Service Overview Screen
Axis Diagnostics - Service Axes
Axis Diagnostics - Service Drives
Diagnostics Menu - Drive System
Setup Menu - Drive System
End of the Module

Axis Diagnostics - Service Overview Screen

To navigate to the relevant diagnostics page please use the following key presses.



The screen shows the status of various enable/monitoring signals via indicators.



Status	Explanation
	The signal is healthy. In the case of an enable, the signal is high. For a monitoring signal there is no fault present.
	The signal is missing i.e. low (Enable signals only).
	The signal is indicating a fault (Monitoring signals only).
	The signal is inactive/not in use.

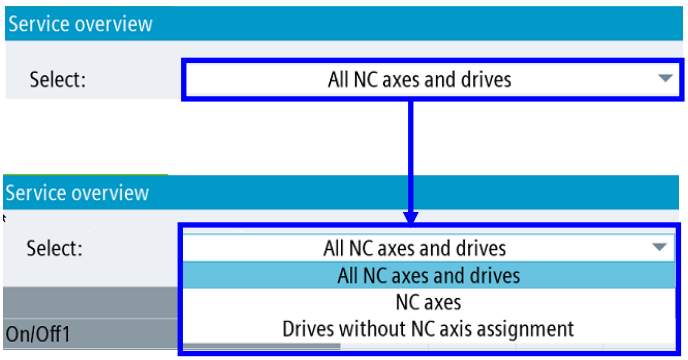
Customising the Service Display:

On systems with a large number of axes it maybe useful to select which axes to display.

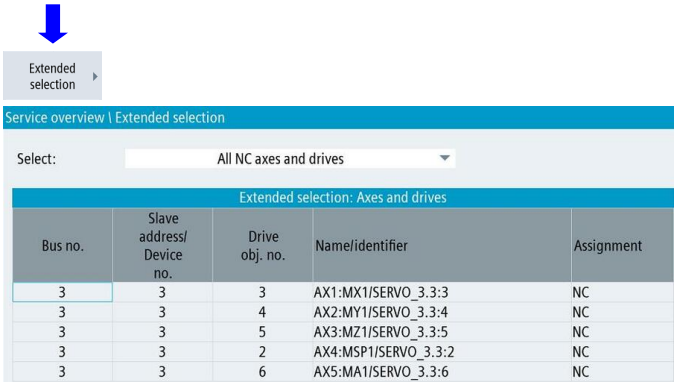
Basic selection:

The drop-down menu allows the selection between:

- All NC axes and drives.
- NC axes
- Drives without NC axis assignment

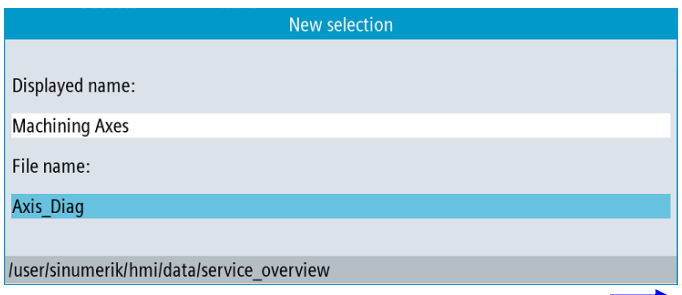


Extended selection:





Choose a service display name and a file name:



Notes:

OK

The new selection now appears in the list:

Machining Axes
All NC axes and drives
NC axes
Drives without NC axis assignment
Machining Axes

Service overview \ Extended selection

Select: Machining Axes

Edit selection

Select the axes to be displayed.

Extended selection: Axes and drives

Please select axes or drives for display:

	Bus no.	Slave address/ Device no.	Drive obj. no.	Name/identifier	Assignment
☑	3	3	3	AX1:MX1/SERVO_3.3:3	NC
☑	3	3	4	AX2:MY1/SERVO_3.3:4	NC
☑	3	3	5	AX3:MZ1/SERVO_3.3:5	NC
☐	3	3	2	AX4:MSP1/SERVO_3.3:2	NC
☐	3	3	6	AX5:MA1/SERVO_3.3:6	NC

OK

Extended selection: Axes and drives

Bus no.	Slave address/ Device no.	Drive obj. no.	Name/identifier	Assignment
3	3	3	AX1:MX1/SERVO_3.3:3	NC
3	3	4	AX2:MY1/SERVO_3.3:4	NC
3	3	5	AX3:MZ1/SERVO_3.3:5	NC

Back

Service overview

Select: Machining Axes

	AX1	AX2	AX3
On/Off1	✓	✓	✓
Off2	✓	✓	✓
Off3	✓	✓	✓
From drive: Operation enabled	✓	✓	✓
From power supply:Operation enabled	✓	✓	✓
Pulse enable	✓	✓	✓
Speed controller enable NC	✓	✓	✓
Pulses enabled	✓	✓	✓
Drive ready	✓	✓	✓
Heatsink temperature	✓	✓	✓
Power section in i²t limitation	✓	✓	✓
Motor temperature	✓	✓	✓
Measuring system 1 active	✓	✓	✓
Measuring system 2 active	●	●	●

Axis Diagnostics - Service Axis

The ***Service axis*** soft-key shows more detailed axis information:

Service axis

AX1:MX1/SERVO_3.3:3

Service axis/spindle — AX1:MX1/SERVO_3.3:3

Signal	Value Unit
Following error	0.000 mm
Controller error	0.000 mm
Contour deviation (axially)	0.000 mm
Servo gain factor (calculated)	0.000 1000/min
Active measuring system	1
Status of measuring system 1	Active
Status of measuring system 2	Park
Actual pos.value measuring system1	-0.000 mm
Actual pos.value measuring system2	0.000 mm
Position setpoint	-0.000 mm
Abs. compens. value meas. system 1	0.000 mm
Abs. compens. value meas. system 2	0.000 mm
Compensation sag + temperature	0.000 mm
Act.speed value of active encoders	0.000 %
Setpoint speed of drive	0.000 %
Pos. offset to master axis/spindle act. value	0.000 mm
Pos. offset to master axis/spindle setpoint	0.000 mm
Override	100.000 %
Parameter set (axis)	1
Controller mode	Stop
Feed forward control mode	Inactive

Axis +, Axis -, Select axis, Back

Bus TCP/IP | Axis diag. | Safety | Trace | System utiliz. | Drive system

The axis to be viewed can be selected with the following softkeys:

Axis + Or Axis - Or Select axis

Axis Diagnostics - Service Drive

The ***Service drive*** soft-key shows more detailed drive information:

Service drive

AX1:MX1/SERVO_3.3:3

Service drive — AX1:MX1/SERVO_3.3:3

Signal	Value Unit
Pulse enabled PLC	Yes
Speed controller enable NC	Yes
Pulses enabled	Yes
Drive ready	Yes
Status indicator	Operation - everything enabled
Message ZK1 drive alarm	No
DC link voltage smoothed	540.157 V
Speed setpoint smoothed	0.000 rpm
Speed setpoint limited	0.000 rpm
Actual speed smoothed	-0.050 rpm
Current actual value torque-generating smoot...	0.006 A
Motor temperature	29.320 °C
Integrator inhibit	No
Parking axis	No
Target drive data set	DDS0
Actual drive data set	DDS0
Target motor data set	MDS0
Actual motor data set	MDS0
Target encoder data set of active encoders	EDS0
Actual encoder data set of active encoders	EDS0
Number of DDS per MDS	1
Open-loop/closed-loop control mode	Speed control (with encoder)

Drive +, Drive -, Select a drive, Back

Bus TCP/IP | Axis diag. | Safety | Trace | System utiliz. | Drive system

The drive to be viewed can be selected with the following softkeys:

Drive + Or Drive - Or Select a drive

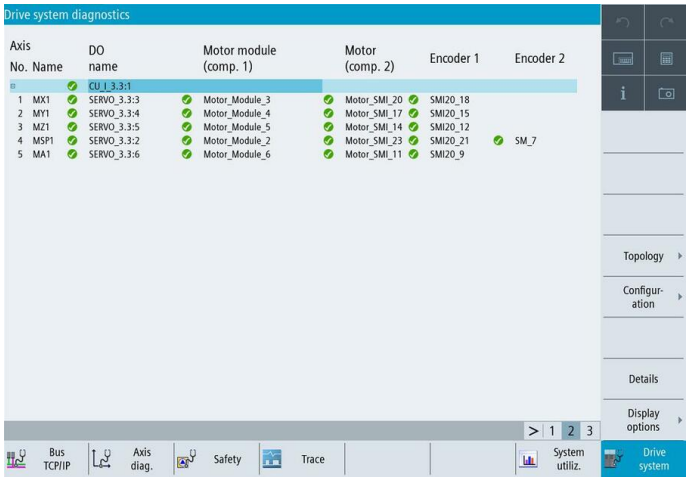
Notes:

Diagnostics Menu ⇨ Drive System.

Further drive diagnostic information selecting the ***Drive system*** soft key.



The drive objects and components (Control Unit, motor modules, encoders) are displayed.

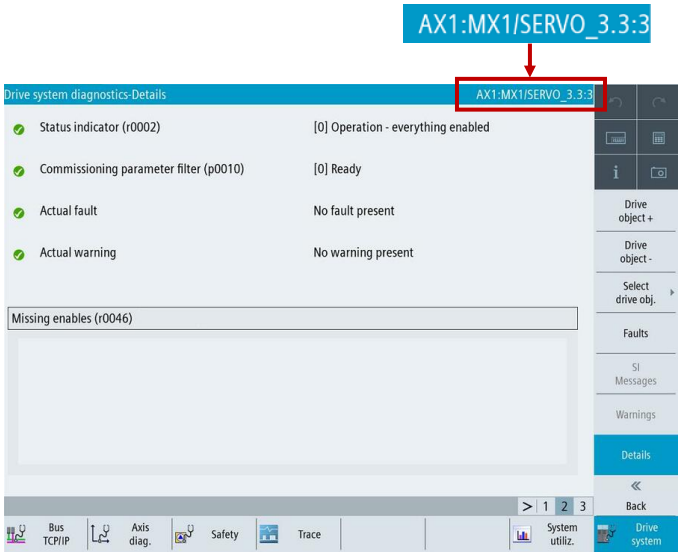


The green tick symbol indicates a healthy drive object. Other symbols will appear if there is a problem

Status	Explanation
	The signal is healthy. In the case of an enable, the signal is high. For a monitoring signal there is no fault present.
	The signal is missing i.e. low (Enable signals only).
	The signal is indicating a fault (Monitoring signals only).

Detailed information for each drive object can be viewed:



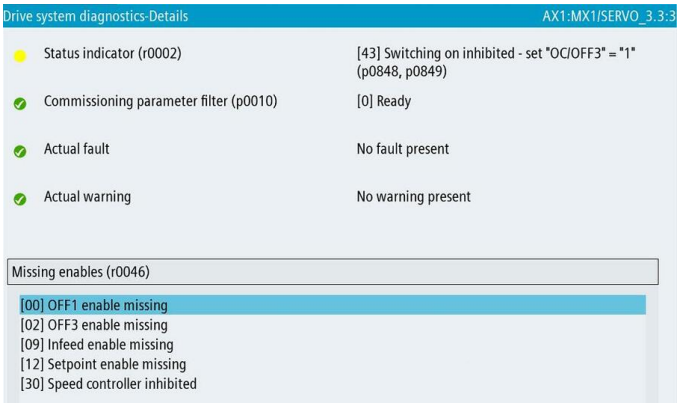


The drive object to be viewed can be selected with the following softkeys:



Faults, Safety Integrated (SI) messages and warnings:

A brief description of missing signals/enables is displayed in the lower half of the screen.



Notes:

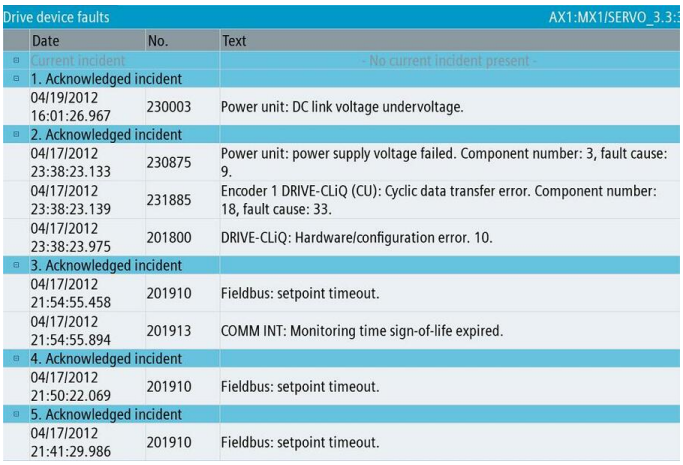


Faults, Safety Integrated messages and warnings can be viewed by selecting the relevant soft-key

Example:

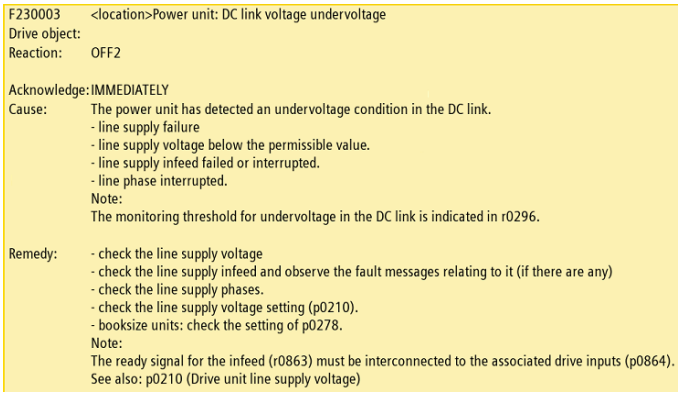


A list of current and past faults is displayed.





The Information key reveals detailed information for the selected fault:





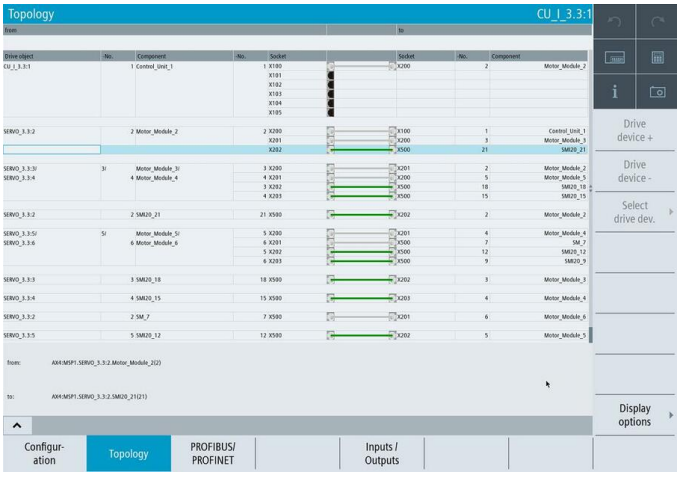
Exit the ***Details*** area:



Diagnostics Menu ⇨ Drive System ⇨ Topology:

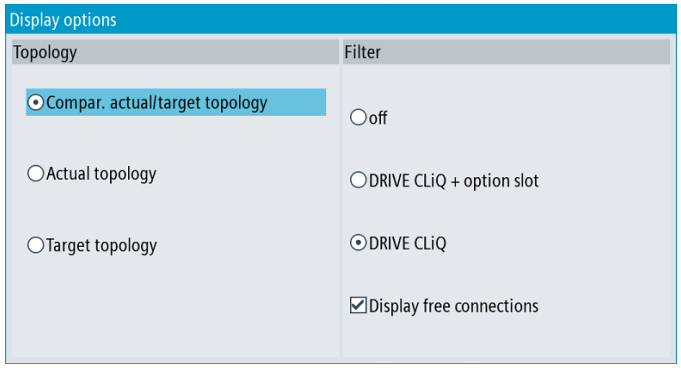
The topology shows the DRIVECLiQ connections of the drive system.







Display options can be used to customise the topology screen.





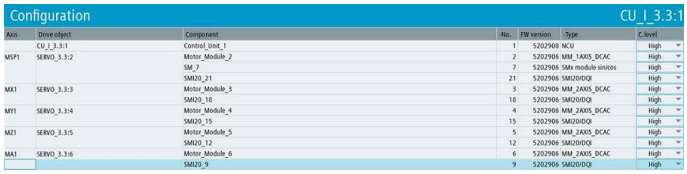


Notes:

Diagnostics Menu ⇨ Drive System ⇨ Configuration:

The configuration screen shows Control Unit , Motor Module and encoder information.

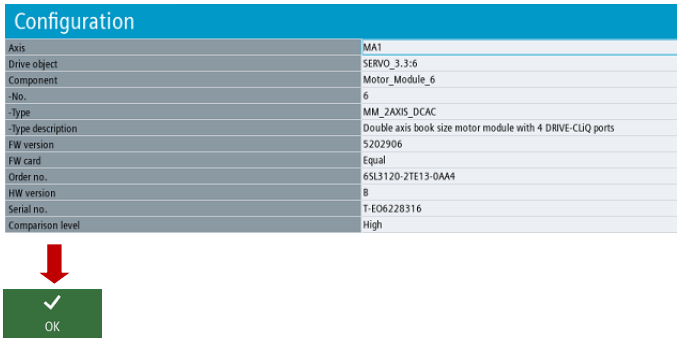




Detailed information can be viewed by selecting the Details soft-key:



Useful information like part numbers can be seen:



Diagnostics Menu ⇨ Drive System ⇨ Configuration ⇨ Profinet/Profibus:

Sinamics drive information is available with the Profinet/ Profibus soft-key.





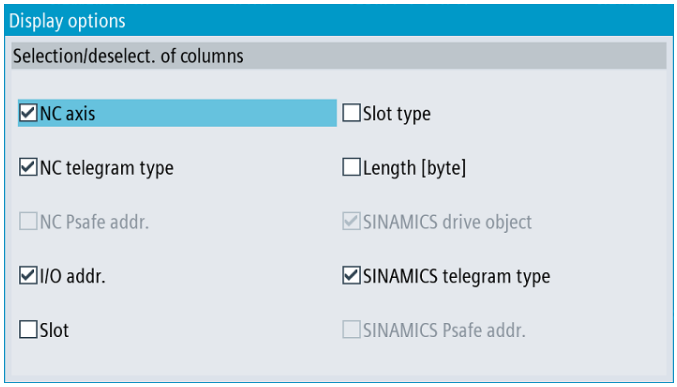
NC axis	NC telegram type	I/O addr.	SINAMICS drive object	SINAMICS telegram type
AX4:MSP1	SIEMENS 136	4100	SERVO_3.3:2(2)	SIEMENS 136
		4100		
AX1:MX1	SIEMENS 136	4140	SERVO_3.3:3(3)	SIEMENS 136
		4140		
AX2:MY1	SIEMENS 136	4180	SERVO_3.3:4(4)	SIEMENS 136
		4180		
AX3:MZ1	SIEMENS 136	4220	SERVO_3.3:5(5)	SIEMENS 136
		4220		
AX5:MA1	SIEMENS 136	4260	SERVO_3.3:6(6)	SIEMENS 136
		4260		
AX6:MC1	--	4300	-- (255)	--
		4300		
--	SIEMENS 391	6500	CU_I_3.3:1(1)	SIEMENS 391
		6500		
--	--	6514	-- (255)	--
		6514		

The telegram types Input/output addresses and drive object numbers can be viewed..

Diagnostics Menu ⇨ Drive System ⇨ Configuration ⇨ Display Options:



Display options can be used to customise the Profi-bus/Profibus screen.

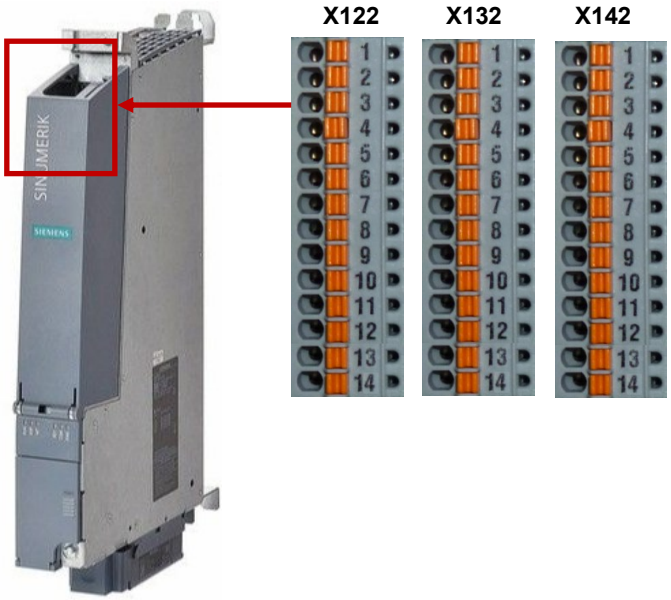


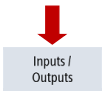


Notes:

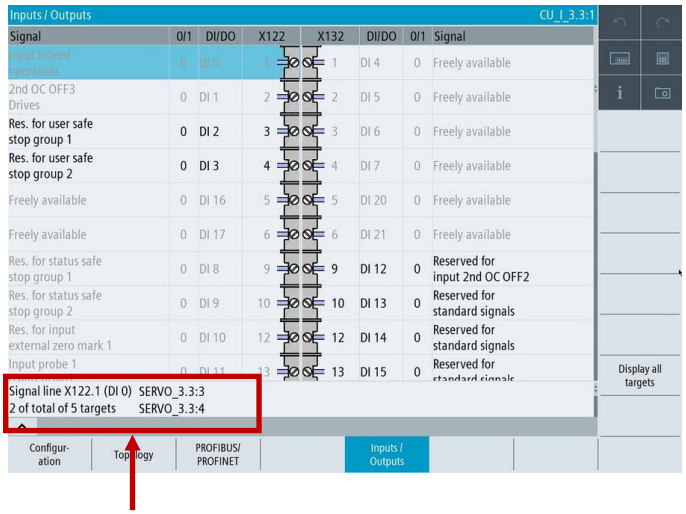
Diagnostics Menu ⇨ Drive System ⇨ Configuration ⇨ Input/Outputs:

The Status of the Sinamics inputs and outputs can be monitored:
The terminals for the Sinamics I/O, X122 & X132, can be found on the top front of the NCU





The left of the screen shows X122 information while the right hand side shows X132 information.



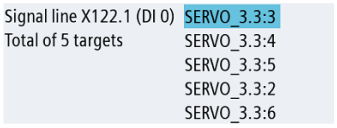
Targets:

When an input/output is highlighted the target information is displayed at the bottom of the screen.

The default number of targets is two.
The ***Display all targets*** soft-key will list all the targets associated with the selected input/output.



The following example shows that X122.1 (infeed operation) is assigned to five servos:

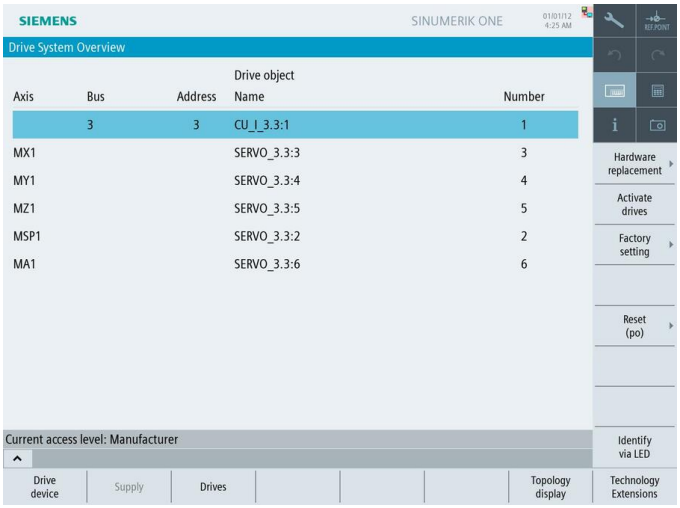


Notes:

Setup Menu ⇨ Drive system.

For further drive diagnostic information, select the **Setup/ *Drive system*** soft key.
The ***Setup*** menu area is primarily for commissioning tasks, however, this area has similar and additional diagnostic information in comparison to the ***Diagnostics/Drive system*** area.

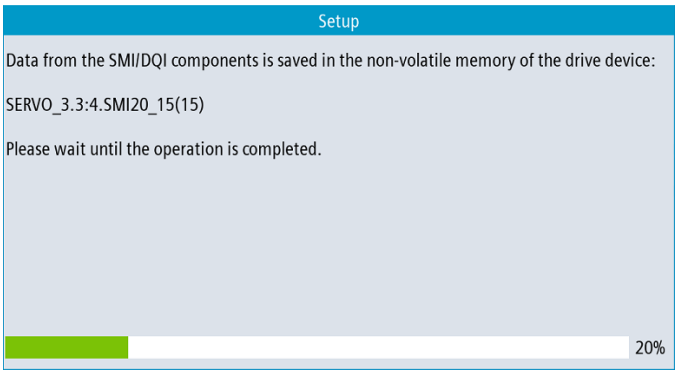




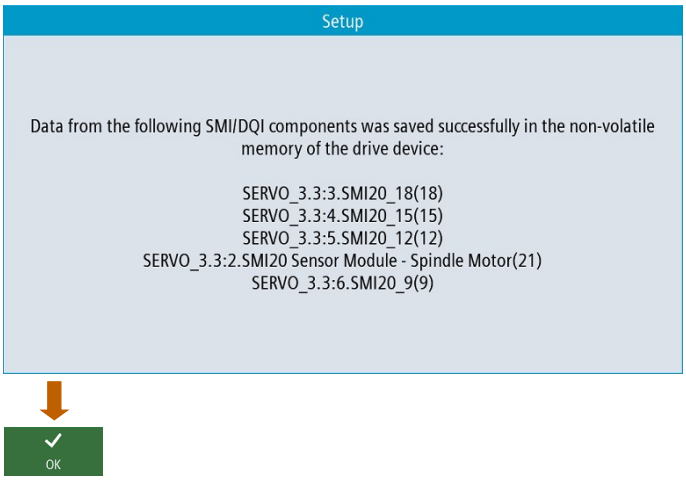
Setup Menu ⇨ Drive system ⇨ Hardware replacement ⇨ Save SMI/DQI data:

The SMI (Sensor Module Internal)/DQI (DRIVECLiQ Internal) encoder data can be saved.
This is used when a motor is replaced.





A summary is displayed:



Load SMI/DQI Data:

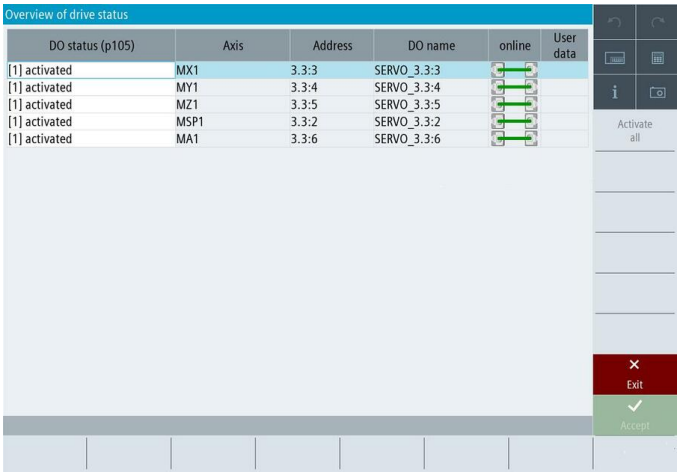
When a motor with an SMI/DQI encoder is replaced the ***Load SMI/DQI*** soft-key will be available.



Setup Menu ⇨ Drive system ⇨ Activate drives.

It is possible to activate/deactivate drives.
The drive/motor can then be removed from the DRIVECLiQ This depends on the way the drive objects and components are connected to the DRIVECLiQ network.





Notes:

Activate all Soft-key:
The Activate all soft-key will be grayed out if all the drives are active



Deactivating Drives:

The drives are deactivated individually.



DO status (p105)	Axis
[1] activated	MX1
[1] activated	MY1
[1] activated	MZ1
[1] activated	MSP1
[1] activated	MA1

p105 setting	Symbol	Meaning
[0]		Component deactivated and not connected.
[0]		Component deactivated and connected.
[1]		Component activated and connected.
[1]		Component activated and connected - fault present.
[2]	No symbol	Component deactivated and not connected. Component is included in the target topology

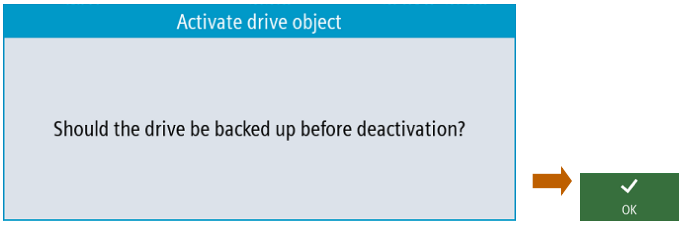
Example - deactivating the MA1 axis by entering a 0 into DO status (p105).

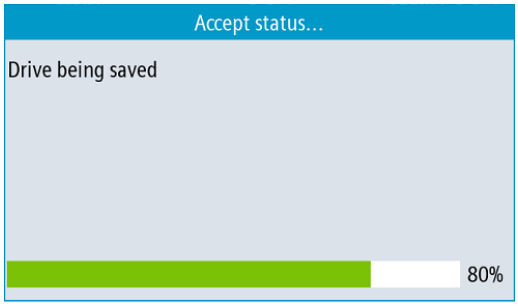
DO status (p105)	Axis
[1] activated	MX1
[1] activated	MY1
[1] activated	MZ1
[1] activated	MSP1
[0]* deactivated	MA1



Accept the changes.

The drive data should be backed up. It can be decided whether to use this if/when the drive is re-activated.





The drive is now deactivated. The ***Save*** symbol indicates the drive data was backed up.

DO status (p105)	Axis	Address	DO name	online	User data
[1] activated	MX1	3.3:3	SERVO_3.3:3		
[1] activated	MY1	3.3:4	SERVO_3.3:4		
[1] activated	MZ1	3.3:5	SERVO_3.3:5		
[1] activated	MSP1	3.3:2	SERVO_3.3:2		
[0] deactivated	MA1	3.3:6	SERVO_3.3:6		

Save symbol 

The ***Reset (po)*** procedure is now required to clear an encoder alarm caused by the deactivation procedure.



The motor, for example, can now be unplugged from the DRIVECLiQ.
It may also be possible to remove the Motor Module providing that the communication to the remaining modules is not interrupted.

Notes:

Activate drives.

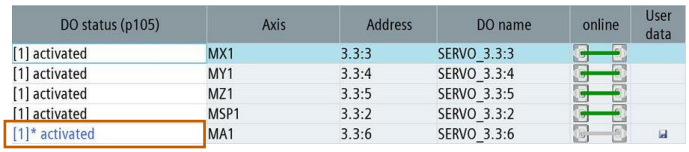
The deactivated drive(s) can also be activated from the same screen.
Ensure that the drive to be activated is correctly connected to the DRIVECLiQ.







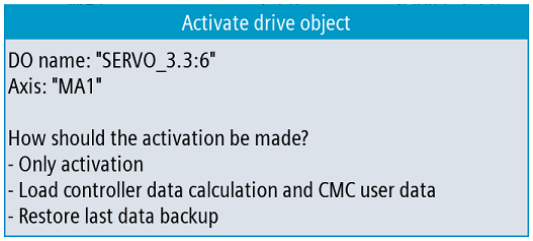
This will activate any deactivated drives.





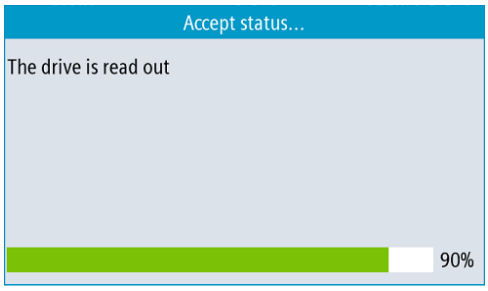
Depending on the requirements the following activation choices are available:

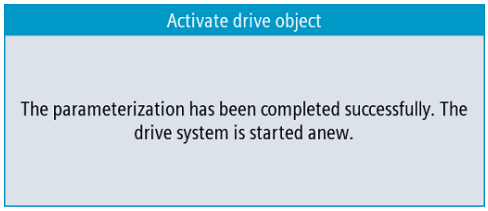
- Drive activation only
- Load controller data calculation and CMC user data - if available.
- Restore last data backup.



Example - loading the last data backup which was saved when the drive was deactivated.





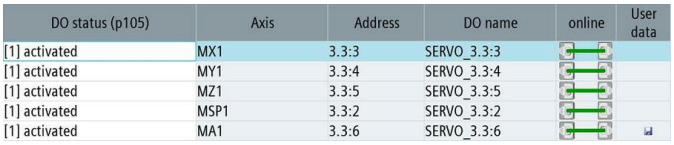




Wait for the dialogue box to clear:



The drives are now activated.





- **Setup Menu ⇨ Drive system ⇨ Factory Setting.**

Note: the ***Factory setting*** soft-key is for commissioning use and is described in the relevant module

- **Setup Menu ⇨ Drive system ⇨ Identify via LED.**

The ***RDY*** LED of the selected DRIVECLiQ drive object/ component can be made to flash for identification purposes.

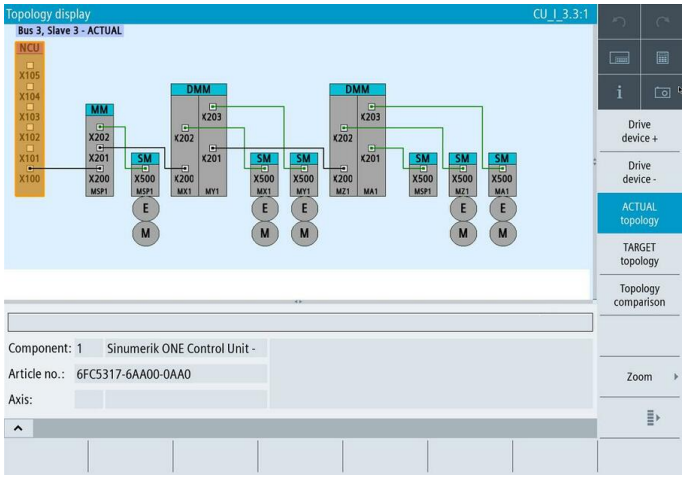


Notes:

Setup Menu ⇨ Drive system ⇨ Topology:

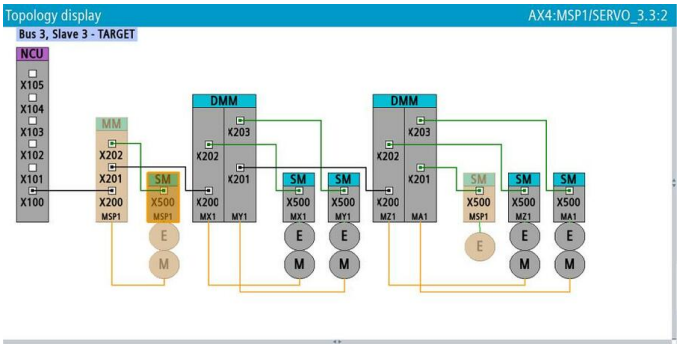
The topology display shows a graphical representation of the connection of the DRIVECLiQ components.
The default screen is the actual topology i.e. the topology of the components connected to the DRIVECLiQ

Topology display → ACTUAL topology



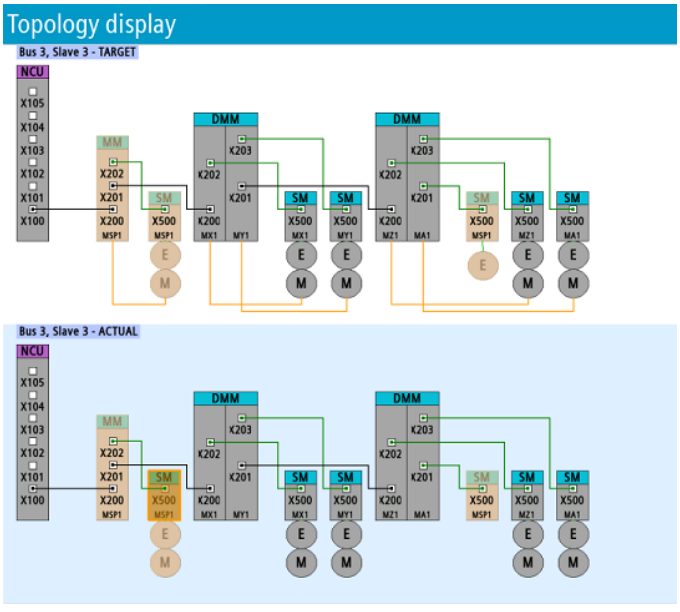
TARGET topology

The target topology is the topology that is stored the system and is used for topology comparison with the actual topology.

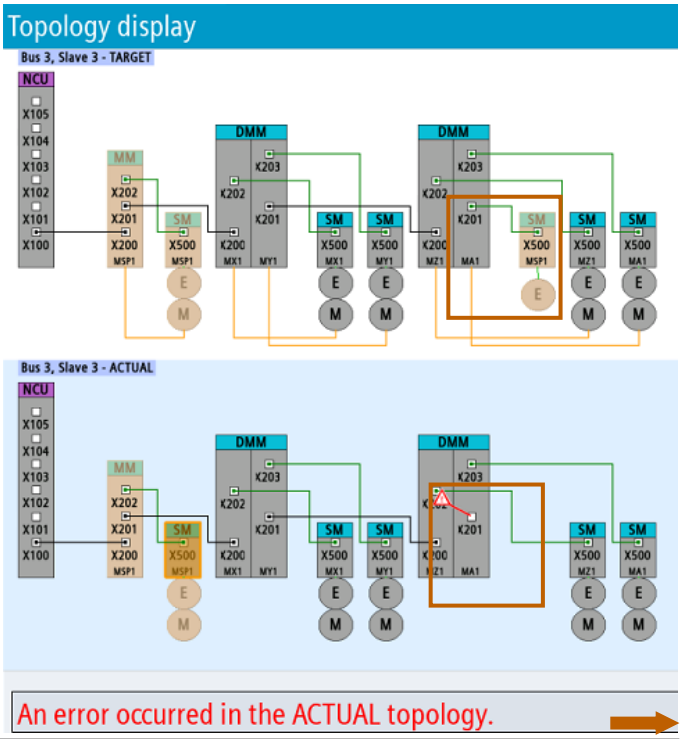


Topology comparison

Topology comparison is useful for checking for problems with the connected drive objects/ components.

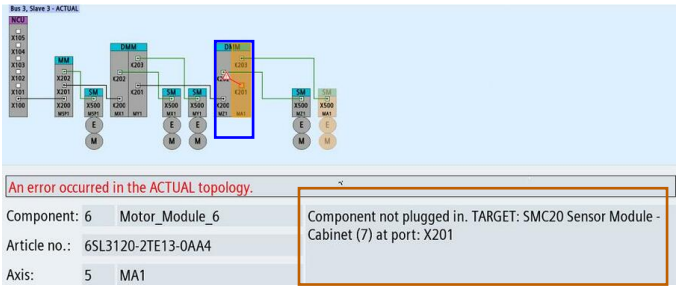


Example - the Topology comparison showing a problem with the actual topology on X201 of the last double motor module (DMM). In this case the SMC had been unplugged



Notes:

Selecting the faulty module shows more detailed fault information



- Additional functions - Identify via LED.

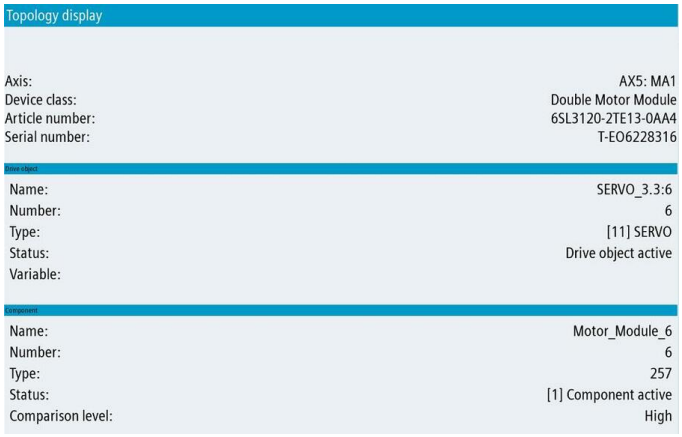


The ***RDY*** LED of the selected DRIVECLiQ drive object/ component can be made to flash for identification purposes.

- Additional functions - Properties.

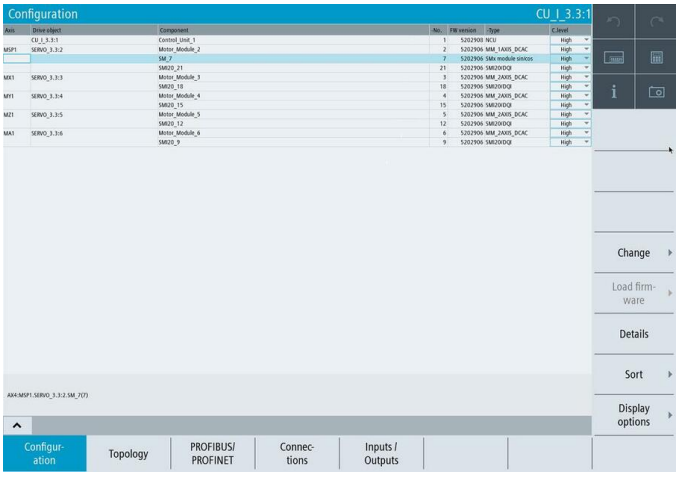


The properties of the selected drive object/ component are displayed.



Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Configuration:



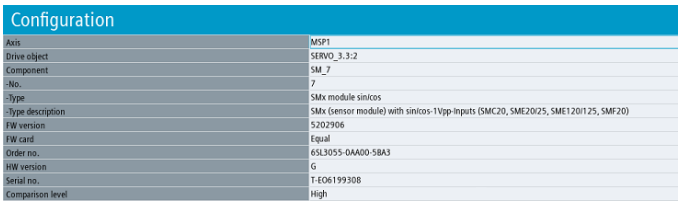


Various information and functions are accessed with the **Configuration** soft-key:

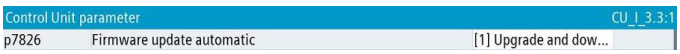
- Edit the axis/object names.
- Check drive object/component information.
- Manual firmware update.
- Set the comparison level.

Details:
Part numbers, firmware versions etc. can be seen for the selected drive object/component.





Load firmware:
The firmware for the Sinamics drive objects and components is automatically downloaded due to the default setting of Control Unit parameter p7826 = 1 (upgrade and downgrade)



The ***Load firmware*** soft-key is available for manual downloading the firmware if p7826 = 0



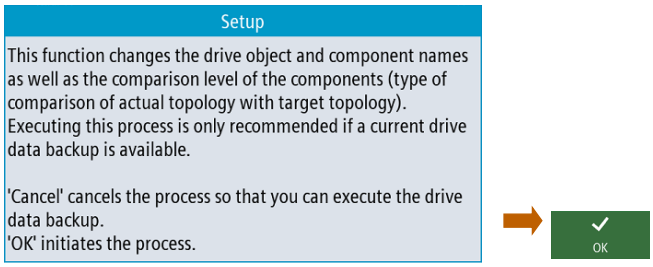
Notes:

Change:
The default component names can be modified to suit.
The comparison level can also be set to facilitate the switching of drive objects/components within the existing drive system.

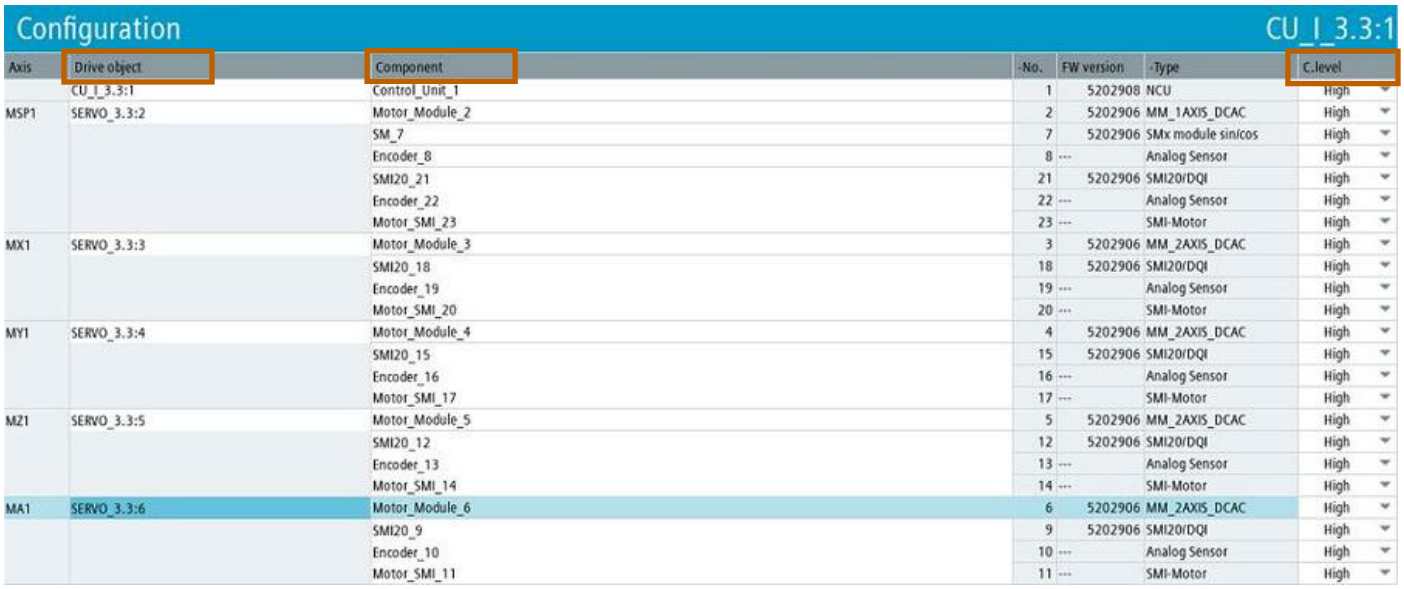
Note: the drives must be disabled before changes can be made.



A message is displayed which advises to ensure drive data backup is available.



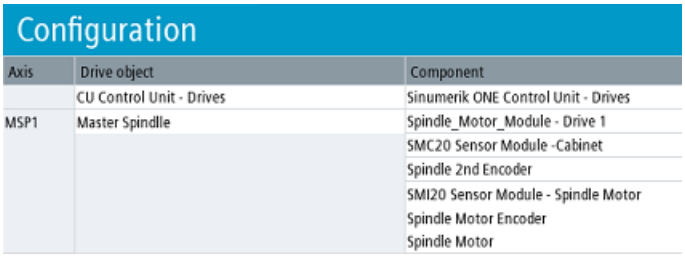
It is possible to change:
- Drive object names
- Component names
- C. level (comparison level)



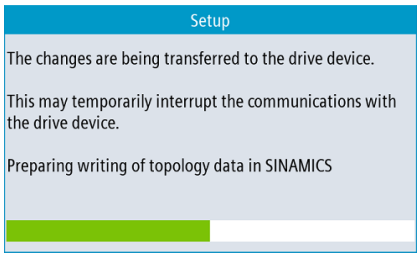
Axis	Drive object	Component	No.	FW version	Type	C.level
	CU_I_3.3:1	Control_Unit_1	1	5202908	NCU	High
MSP1	SERVO_3.3:2	Motor_Module_2	2	5202906	MM_1AXIS_DCAC	High
		SM_7	7	5202906	SMx module sin/cos	High
		Encoder_8	8	---	Analog Sensor	High
		SMI20_21	21	5202906	SMI20/DQI	High
		Encoder_22	22	---	Analog Sensor	High
		Motor_SMI_23	23	---	SMI-Motor	High
MX1	SERVO_3.3:3	Motor_Module_3	3	5202906	MM_2AXIS_DCAC	High
		SMI20_18	18	5202906	SMI20/DQI	High
		Encoder_19	19	---	Analog Sensor	High
		Motor_SMI_20	20	---	SMI-Motor	High
MY1	SERVO_3.3:4	Motor_Module_4	4	5202906	MM_2AXIS_DCAC	High
		SMI20_15	15	5202906	SMI20/DQI	High
		Encoder_16	16	---	Analog Sensor	High
		Motor_SMI_17	17	---	SMI-Motor	High
MZ1	SERVO_3.3:5	Motor_Module_5	5	5202906	MM_2AXIS_DCAC	High
		SMI20_12	12	5202906	SMI20/DQI	High
		Encoder_13	13	---	Analog Sensor	High
		Motor_SMI_14	14	---	SMI-Motor	High
MA1	SERVO_3.3:6	Motor_Module_6	6	5202906	MM_2AXIS_DCAC	High
		SMI20_9	9	5202906	SMI20/DQI	High
		Encoder_10	10	---	Analog Sensor	High
		Motor_SMI_11	11	---	SMI-Motor	High

Notes:

Example: modified object and component names for the Control Unit and Spindle.







Comparison Level (C. level)

The DriveCLiq drive bus is continually monitored for any differences between the set and actual topology.
This is to protect against faults developing and components accidentally being connected into the wrong DRIVCLiQ interface.
The comparison level determines how critical the monitoring is. This can be adjusted in the event of service cases, for example, where it may be necessary to interchange existing components on the drive system.

Comparison Levels Description:

- High - the complete rating plate, including the serial number, is compared. Any component not connected to the correct interface will generate an alarm.

.

- Medium - the component type and part number are compared. Example: two identical motor modules could be interchanged without causing an alarm.

- Low - the component type is compared. Example: two motor modules of different ratings but of the same type can be interchanged without causing an alarm

- Minimum - the component class is compared. Example any two motor modules or SMCs etc. can be interchanged.

Comparison Level Requirement:

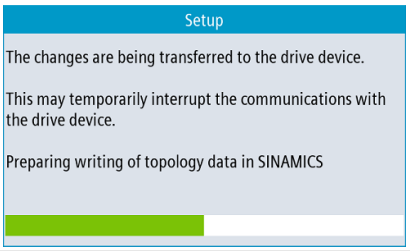
Service case 1:
If an identical component is to be fitted to a system to replace a faulty one it is not necessary to change the comparison level.
The system should automatically load the correct firmware to the new component.

Service case 2:
If two existing components are to be interchanged to prove a fault e.g. Motor Module 1 & Motor Module 2, the system will detect this and issue alarms depending on the setting of the comparison level.
In this case setting the relevant component comparison levels to low would allow the modules to be interchanged without alarms occurring.

Example - if the X and Y axis drive components require interchanging then setting the C. level to low would allow this without errors.







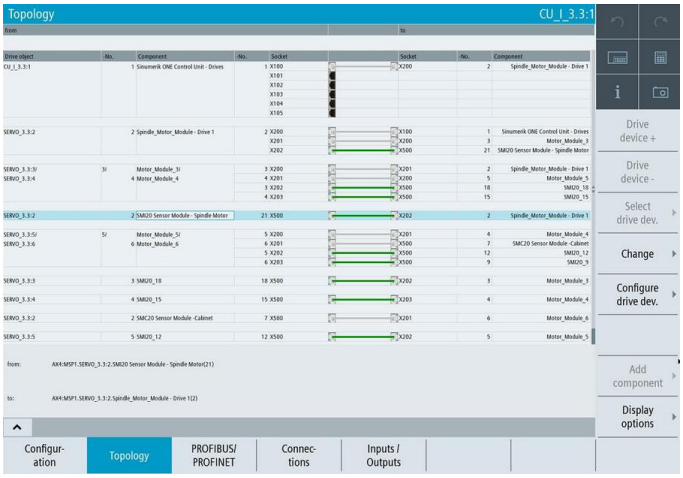
Warning: it is the responsibility of the engineer to decide whether it is suitable to modify the comparison level and interchange drive components.
Careful consideration must be given to the safety aspects etc.

Notes:

Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Topology.

The topology shows the DRIVECLiQ connections of the drive system.

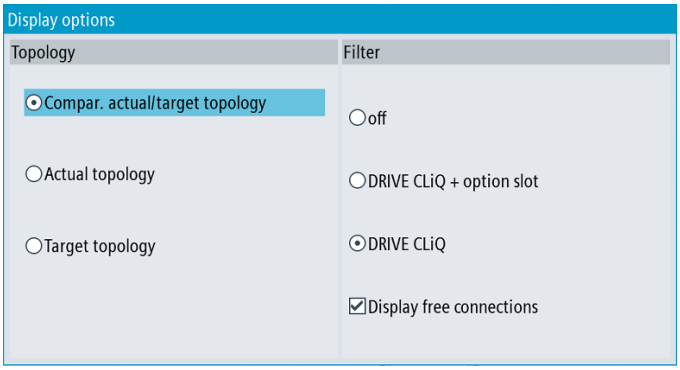




Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Topology ⇨ Display Options



Display options can be used to customise the topology screen.





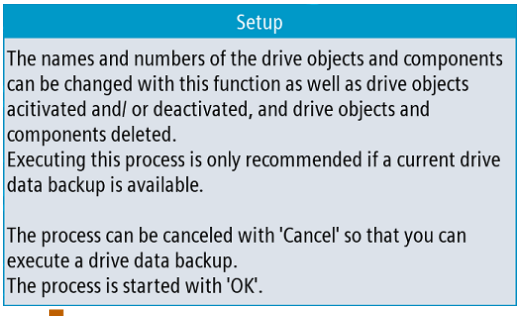
Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Topology ⇨ Change:

With the Change soft-key it is possible to:

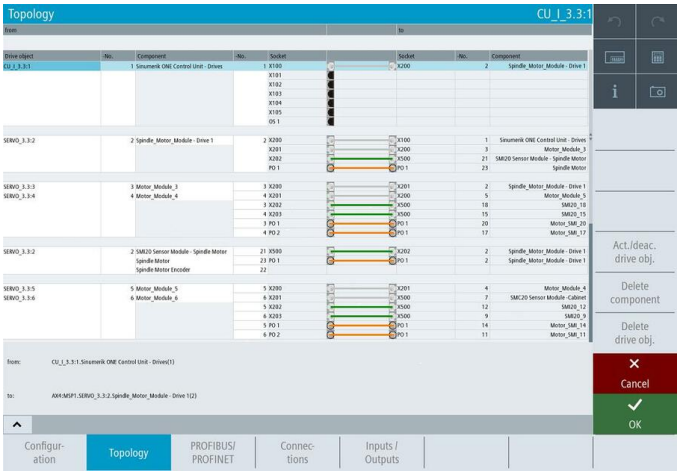
- Change drive object/component names
- Activate drive objects/components
- Deactivate drive objects/components
- Delete drive objects/components

! This is for experienced personnel. Always ensure that a current data backup is available.











Drive objects can be activated/deactivated. This sets parameter p105 of the selected drive object.



Drive components can be deleted.



Drive objects can be deleted.



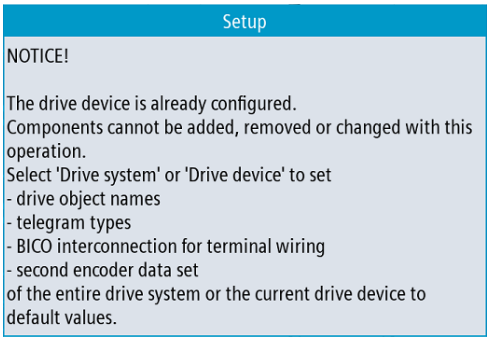
Notes:

Use the ***Information*** key for more details

Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Topology ⇨ Configure Drive Device.

Configure drive dev.

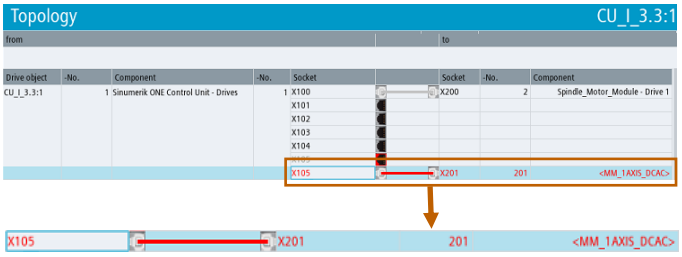
The default drive settings can be loaded.



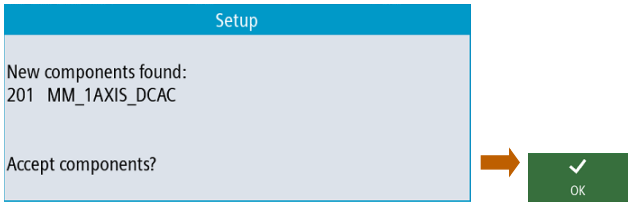
Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Topology ⇨ Add component.

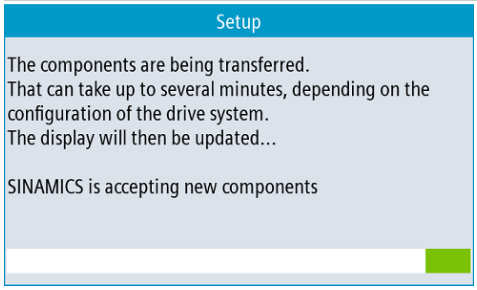
If additional DRIVECLiQ equipment is connected to an existing drive system it can be added using the ***Add component*** soft-key.

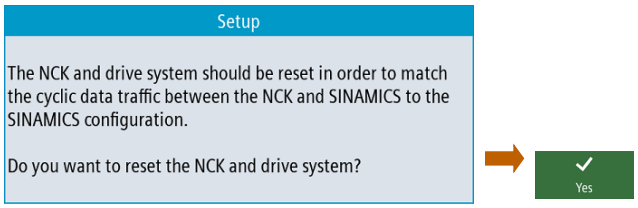
Additional components are displayed in red.



Add component



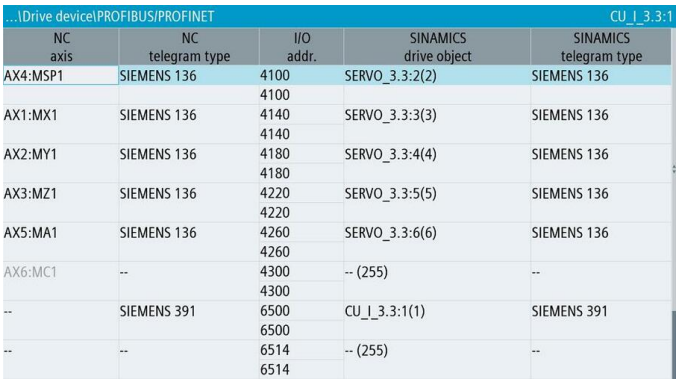




Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Profinet/ Profibus.

Sinamics drive information is available with the Profinet/ Profibus soft-key

PROFIBUS/ PROFINET



NC axis	NC telegram type	I/O addr.	SINAMICS drive object	SINAMICS telegram type
AX4:MSP1	SIEMENS 136	4100	SERVO_3.3:2(2)	SIEMENS 136
		4100		
AX1:MX1	SIEMENS 136	4140	SERVO_3.3:3(3)	SIEMENS 136
		4140		
AX2:MY1	SIEMENS 136	4180	SERVO_3.3:4(4)	SIEMENS 136
		4180		
AX3:MZ1	SIEMENS 136	4220	SERVO_3.3:5(5)	SIEMENS 136
		4220		
AX5:MA1	SIEMENS 136	4260	SERVO_3.3:6(6)	SIEMENS 136
		4260		
AX6:MC1	--	4300	-- (255)	--
		4300		
--	SIEMENS 391	6500	CU_I_3.3:1(1)	SIEMENS 391
		6500		
--	--	6514	-- (255)	--
		6514		

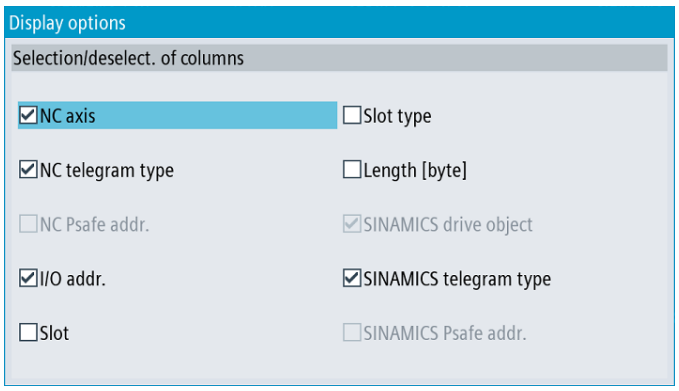
The telegram types Input/output addresses and drive object numbers can be viewed.

Notes:

Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Profinet/ Profibus ⇨ Display Options:

Display options

Display options can be used to customise the Profinet/Profibus screen.

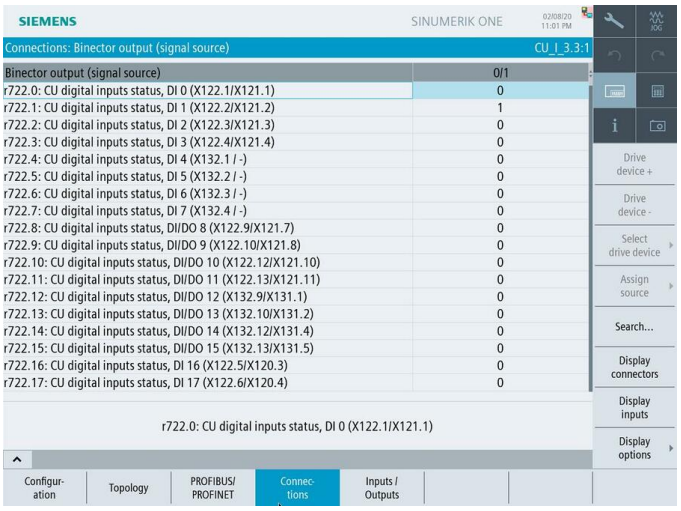


OK

Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Connections.

The ***Connections*** soft-key allows:

- Sinamics Binector inputs and outputs to be monitored.
- Sinamics Connector inputs and outputs to be monitored.
- Assignment of a signal source to the Binector/Connector inputs



Display binectors → Display outputs

Connections: Binector output (signal source) CU_I_3.3:1

Binector output (signal source)	0/1
r722.0: CU digital inputs status, DI 0 (X122.1/X121.1)	0
r722.1: CU digital inputs status, DI 1 (X122.2/X121.2)	0
r722.2: CU digital inputs status, DI 2 (X122.3/X121.3)	0
r722.3: CU digital inputs status, DI 3 (X122.4/X121.4)	0
r722.4: CU digital inputs status, DI 4 (X132.1 / -)	0
r722.5: CU digital inputs status, DI 5 (X132.2 / -)	0
r722.6: CU digital inputs status, DI 6 (X132.3 / -)	0
r722.7: CU digital inputs status, DI 7 (X132.4 / -)	0
r722.8: CU digital inputs status, DI/DO 8 (X122.9/X121.7)	0
r722.9: CU digital inputs status, DI/DO 9 (X122.10/X121.8)	0
r722.10: CU digital inputs status, DI/DO 10 (X122.12/X121.10)	0
r722.11: CU digital inputs status, DI/DO 11 (X122.13/X121.11)	0
r722.12: CU digital inputs status, DI/DO 12 (X132.9/X131.1)	0
r722.13: CU digital inputs status, DI/DO 13 (X132.10/X131.2)	0
r722.14: CU digital inputs status, DI/DO 14 (X132.12/X131.4)	0
r722.15: CU digital inputs status, DI/DO 15 (X132.13/X131.5)	0
r722.16: CU digital inputs status, DI 16 (X122.5/X120.3)	0
r722.17: CU digital inputs status, DI 17 (X122.6/X120.4)	0

Display binectors → Display inputs

Connections: Binector input (signal sink) CU_I_3.3:1

Source	0/1	Binector input (signal sink)
r2090.0: IF1 PROFIdrive PZD1 receive bit-serial, Bit 0	0	p681.0: Central measuring probe synchroni...
0	0	p738.0: CU signal source for terminal DI/DO 8
0	0	p739.0: CU signal source for terminal DI/DO 9
0	0	p740.0: CU signal source for terminal DI/DO...
0	0	p741.0: CU signal source for terminal DI/DO...
0	0	p742.0: CU signal source for terminal DI/DO...
0	0	p743.0: CU signal source for terminal DI/DO...
0	0	p744.0: CU signal source for terminal DI/DO...
0	0	p745.0: CU signal source for terminal DI/DO...
0	0	p2080.0: IF1 binector-connector converter s...
0	0	p2081.0: IF1 binector-connector converter s...
0	0	p2082.0: IF1 binector-connector converter s...
r722.11: CU digital inputs status, DI/DO 11 (X122.13/X...	0	p2083.0: IF1 binector-connector converter s...
0	0	p2084.0: IF1 binector-connector converter s...
r2090.7: IF1 PROFIdrive PZD1 receive bit-serial, Bit 7	0	p2102.0: Acknowledge all faults
r2090.7: IF1 PROFIdrive PZD1 receive bit-serial, Bit 7	0	p2103.0: 1st acknowledge faults
0	0	p2104.0: 2nd acknowledge faults
0	0	p2105.0: 3rd acknowledge faults

Notes:

Display connectors → Display outputs

Connections: Connector output (signal source) CU_I_3.3:1

Connector output (signal source)	Value	Unit
r565[0]: Probe time stamp	0	-
r565[1]: Probe time stamp	0	-
r565[2]: Probe time stamp	0	-
r565[3]: Probe time stamp	0	-
r565[4]: Probe time stamp	0	-
r565[5]: Probe time stamp	0	-
r565[6]: Probe time stamp	0	-
r565[7]: Probe time stamp	0	-
r565[8]: Probe time stamp	0	-
r565[9]: Probe time stamp	0	-
r565[10]: Probe time stamp	0	-
r565[11]: Probe time stamp	0	-
r565[12]: Probe time stamp	0	-
r565[13]: Probe time stamp	0	-
r565[14]: Probe time stamp	0	-
r565[15]: Probe time stamp	0	-
r566[0]: Probe time stamp reference	0H	-
r566[1]: Probe time stamp reference	0H	-

Display connectors → Display inputs

Connections: Connector input (signal sink) CU_I_3.3:1

Target	Value	Unit	Connector input (signal sink)
50[2]: IF1 PROFIdrive PZD receive word, PZD 3	0H	-	p682[0]: Central measuring probe control ...
0	0	-	p771[0]: Test sockets signal source, T0
0	0	-	p771[1]: Test sockets signal source, T1
0	0	-	p771[2]: Test sockets signal source, T2
r2050[0]: IF1 PROFIdrive PZD receive word,...	1000H	-	p2045[0]: PB/PN clock synchronous controll...
r2089[1]: IF1 send binector-connector conv...	6F00H	-	p2051[0]: IF1 PROFIdrive PZD send word, P...
r2089[2]: IF1 send binector-connector conv...	0H	-	p2051[1]: IF1 PROFIdrive PZD send word, P...
r688[0]: Central measuring probe status w...	0H	-	p2051[2]: IF1 PROFIdrive PZD send word, P...
r687[0]: Central measuring probe measurin...	0	-	p2051[3]: IF1 PROFIdrive PZD send word, P...
r686[0]: Central measuring probe measurin...	0	-	p2051[4]: IF1 PROFIdrive PZD send word, P...
r687[1]: Central measuring probe measurin...	0	-	p2051[5]: IF1 PROFIdrive PZD send word, P...
r686[1]: Central measuring probe measurin...	0	-	p2051[6]: IF1 PROFIdrive PZD send word, P...
0	0	-	p2051[7]: IF1 PROFIdrive PZD send word, P...
0	0	-	p2051[8]: IF1 PROFIdrive PZD send word, P...
0	0	-	p2051[9]: IF1 PROFIdrive PZD send word, P...
0	0	-	p2051[10]: IF1 PROFIdrive PZD send word, ...
0	0	-	p2051[11]: IF1 PROFIdrive PZD send word, ...
0	0	-	p2051[12]: IF1 PROFIdrive PZD send word, ...

Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Connections ⇨ Display Options:

Display options

Display options can be used to customise the ***Connections*** screen.

Display options: connections

MD display filter active

Filtering by indices: from 0 to 0

Hide 0 BICO Hide 1 BICO

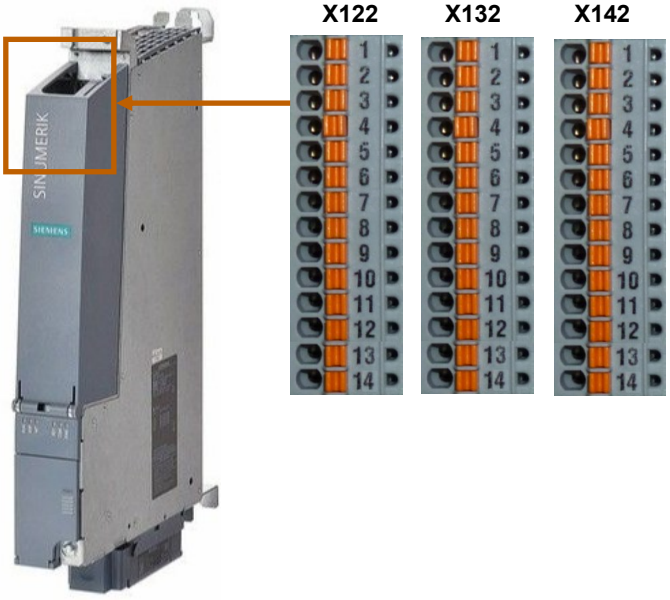
Filtering by display groups

D0(	Displays, signals	D1!	Trace / function generator
D0'	Power section	D1:	Modulation
D0:	Motor	D1&	Motor identification
D0:	Encoder	D1(	Communication
D0<	Applications	D2(	Alarms
D0(	Commands	D2'	Technology
D0:	Terminals	D2!	Topology
D0(	Setpoints	D2<	Safety Integrated
D1'	Functions	D2:	Free function blocks
D1:	V/f control	D2(	Factory settings
D1:	Control	D3(	General startup parameters
D1<	Data sets		All others

OK

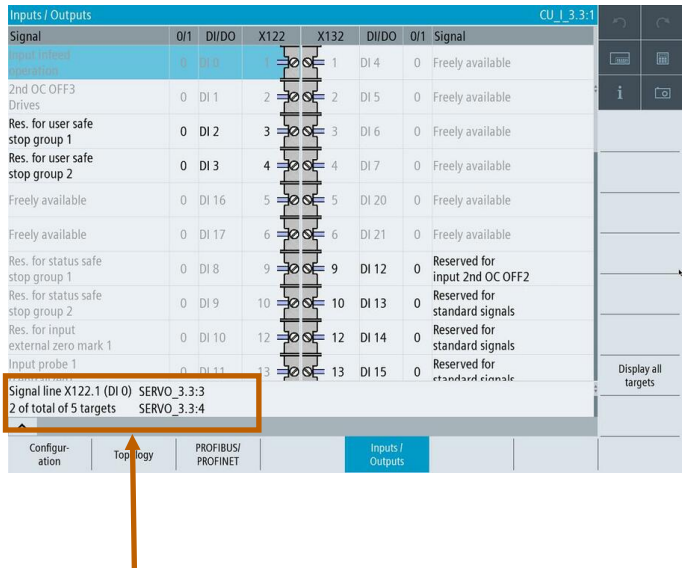
Setup Menu ⇨ Drive system ⇨ Drive Device ⇨ Connections ⇨ Input/Outputs:

The status of the Sinamics inputs and outputs can be monitored:
The terminals for the Sinamics I/O, X122 & X132, can be found on the top front of the NCU



Notes:

The left of the screen shows X122 information while the right hand side shows X132 information.



Targets:

When an input/output is highlighted the target information is displayed at the bottom of the screen.

The default number of targets is two.
The ***Display all targets*** soft-key will list all the targets associated with the selected input/output.



The following example shows that X122.1 (infeed operation) is assigned to five servos:

Signal line X122.1 (DI 0)	SERVO_3.3:3
Total of 5 targets	SERVO_3.3:4
	SERVO_3.3:5
	SERVO_3.3:2
	SERVO_3.3:6

Assign Terminal

Certain terminals are reserved and assigned to a Sinamics function by default, e.g. X122.2 - OFF3 enable
Some terminals are reserved but have not been assigned to a Sinamics function, e.g. X122.3
The reserved and unassigned terminal can be assigned using the ***Assign terminal*** soft-key



Example - X122.3 is reserved for the drive based Safety Integrated Safe Stop function but it has not been assigned to a drive or drives.



Select the drive on which the function is required.

Example:





Confirm the selection.



The new assignment can be viewed at the bottom of the screen.

Signal line X122.3 (DI 2)	SERVO_3.3:3
Total of 3 targets	SERVO_3.3:4
	SERVO_3.3:5

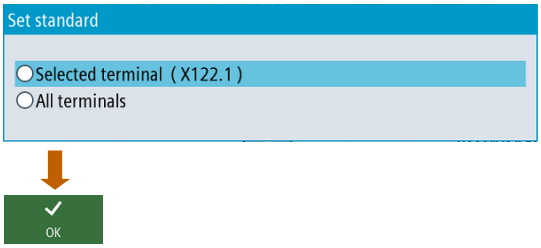
Notes:

Set Standard

The standard settings of the terminals can be applied.



The standard settings can be applied to Individual terminals or all terminals.



A011: END

Notes:

015

SINUMERIK

Alarm Structure

Module Description:

The SINUMERIK system offers various possibilities for diagnosis, one of which is displaying alarms and messages.

Alarms and messages are generated to indicate the machine status and faults in the control unit, drive system or machine. They are used to assist the user with identifying the cause of machine stoppages.

Alarms/messages are sub-divided into the following ranges:

- NCK
- HMI
- SINAMICS
- PLC

Module objective:

Upon completion of this module you will be able to perform the following tasks:

- Understand alarm and message structure of the SINUMERIK system.

Content:

NCK Alarms and Messages Structure

Alarm Display Description

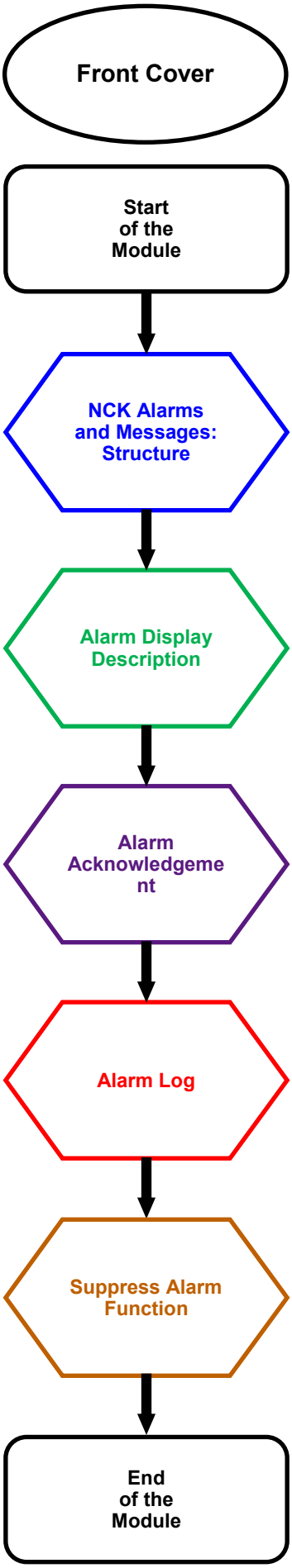
Alarm Acknowledgement

Alarm Log

Suppress Alarm Function

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
NCK Alarms and Messages: Structure
Alarm Display Description
Alarm Acknowledgement
Alarm Log
Suppress Alarm Function
End of the Module

NCK Alarms and Messages: Structure

The NCK alarm structure is divided into three areas. Each area has been assigned a separate range of alarm/message numbers.

Message/Alarm Nr.	Function range
NCK-Alarms/Messages	
000000 - 009999	General alarms
010000 - 019999	Channel alarms
020000 - 029999	Axis/spindle alarms
030000 - 039999	Peripheral alarms
040000 - 040000	Base program alarms
060000 - 064999	Cycle alarms SIEMENS
065000 - 069999	Cycle alarms user
070000 - 079999	Compile cycles Manufacturer and OEM
HMI-alarms/messages	
100000 - 100999	Basic system
101000 - 101999	Diagnosis
102000 - 102999	Services
103000 - 103999	Machine
104000 - 104999	Parameter
105000 - 105999	Programming
106000 - 106999	Reserved
107000 - 107999	OEM
108000 - 108999	HiGraph
109000 - 109999	Distributed systems (M to N)
110000 - 110999	Cycles
111000 - 111999	ShopMill, ShopTurn
113000 - 113999	Expanding User Interface (Easy screen)
114000 - 114999	HT6
119000 - 119999	OEM
120000 - 129999	HMI advanced messages
130000 - 139999	OEM
142000 - 142099	RCS Host/Viewer
149000 - 149999	ePS

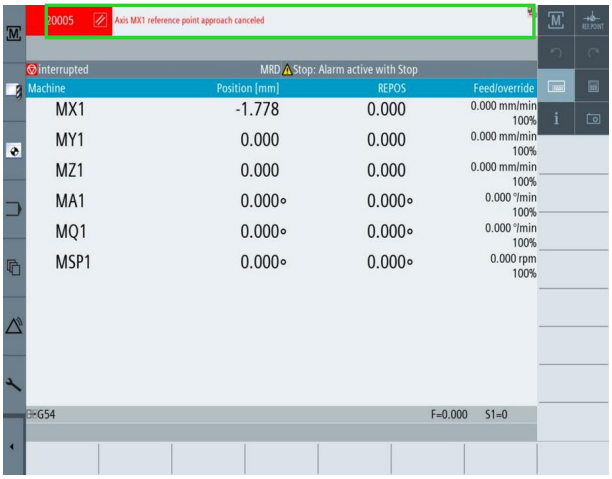
Message/Alarm Nr.	Function range
PLC Alarms/messages	
400000 - 499999	General PLC alarms
500000 - 599999	Channel PLC alarms
600000 - 699999	Axis/spindle PLC alarms
700000 - 799999	User PLC alarms
800000 - 899999	User PLC alarms
900000 - 999999	User PLC alarms
2000000 - 2499999	Alarms from PLC-1500 from TiA

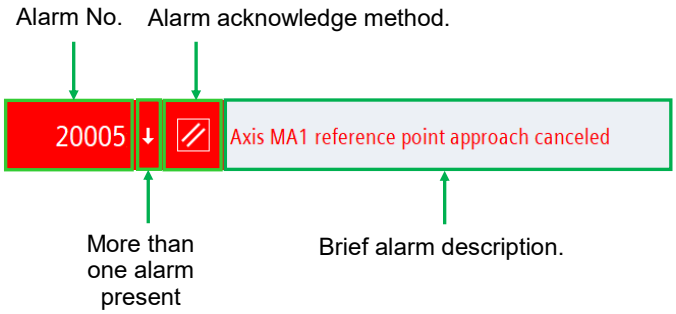
Message/Alarm Nr.	Function range
SINAMICS-Alarms	
201000 - 203999	Control unit (CU)
204000 - 204999	Reserved
205000 - 205999	Power Unit
206000 - 206999	Infeed
207000 - 207999	Drive
208000 - 208999	Option board
209000 - 209999	Reserved
213000 - 213002	Licence
230000 - 230999	DriveCLiQ component power unit
231000 - 231999	DriveCLiQ component encoder 1
232000 - 232999	DriveCLiQ component encoder 2
233000 - 233999	DriveCLiQ component encoder 3
234000 - 234999	Reserved
235000 - 235999	Terminal Module 31 (TM31)
236000 - 236999	Reserved
240000 - 240999	Controller extension (NX32)
241000 - 248999	Reserved
249000 - 249999	Sinamics GM7SM7GL
250000 - 250999	Communication board (Comm Board)
250500 - 259999	Reserved

Notes:

Alarm Display Description

When an alarm is activated it will be displayed on the top line of the controller display.





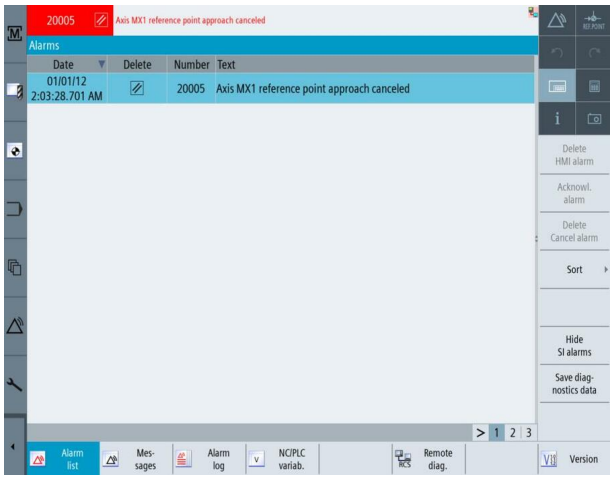
If more than one alarm is present (Indicated by the arrow) they can be made to rotate in turn by setting Display Machine Data 9056.



Display machine data			
9009	$MM_KEYBOARD_STATE	2	po
9032	$MM_HMI_MONITOR	""	po
9056	$MM_ALARM_ROTATION_CYCLE	300 ms	po
9057	$MM_ENABLE_CHANNEL_MSG_FILTER	1	po

With values of less than 500 there is no rotation cycle. Values of 500 to 10000 (milliseconds) set the time that each alarm/ message is displayed for.
The machine data is activated with an NC reset.





A full list of alarms can be seen in the “Diagnostics” area.

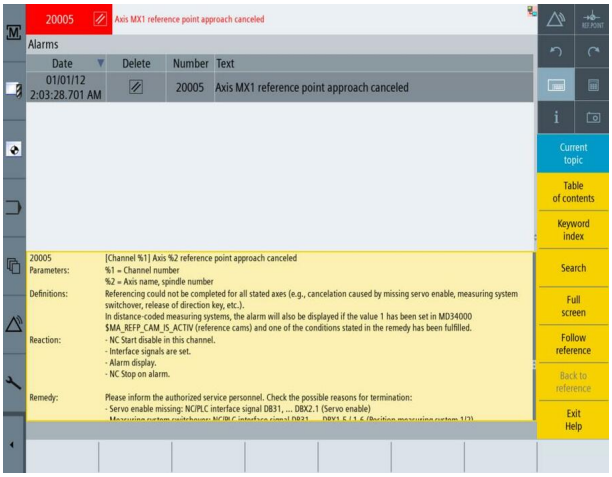


To see extra information, highlight an alarm and select the “Help” key.



Notes:

The information will be displayed in a pop-up box.



Help Screen Function Description:

Current topic — Detailed display of the selected alarm when the “Help” key is selected.

Table of contents — On board operating and programming manuals.

Keyword index — A list of key words is displayed, each is a shortcut to the correct help-file.

Search — The partial text of an alarm can be searched for.

Full screen — The pop-up box will be displayed on the whole screen.

Follow reference — Used to open the link to the help-file from a keyword or search item.

Back to reference — Returns to the keyword or search item.

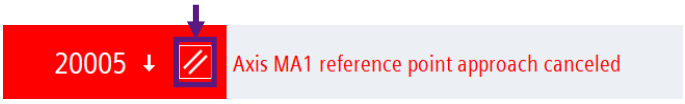
Exit Help — Exits the help screens.

Alarm Acknowledgement

When an alarm has occurred and the cause rectified, the alarm will have to be acknowledged.
The type of acknowledgement depends upon the type/ seriousness of the alarm.
Some alarms needs a reset of the system while others require acknowledgement via a pushbutton.

Acknowledgement types:

The required acknowledgement is indicated to the right of the alarm number on the alarm display line.



Requires action by the operator before continuing with the Cycle Start. Alarm can also be cleared by the MCP “Reset” button

Controller needs to be powered off/on or an NCK power on via the “Reset (po)” soft-key in the “Setup” area.



Acknowledgement is via the MCP “Reset” button.
This will also reset the machining program.

Acknowledgement is via the NC keyboard “Alarm cancel” button.
This should not reset the machining program.

Acknowledgement is via the “Delete HMI alarm” in the “Diagnostic” area.



This is specifically related to PLC user alarms.
A button/key/switch should be provided by the manufacturer to facilitate the acknowledgement of the alarms.

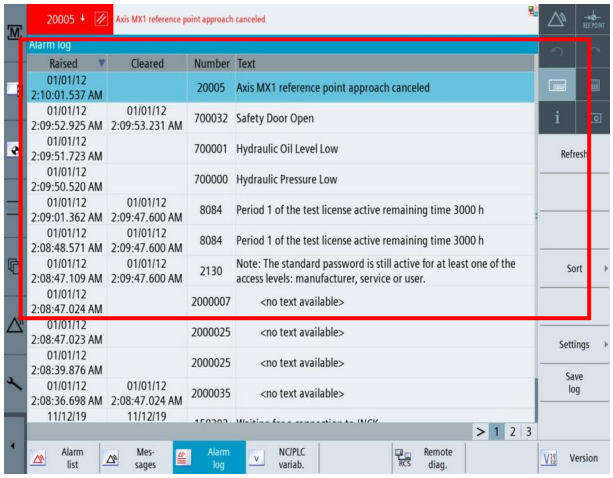
Notes:

Alarm Log

A log of alarms that have occurred can be seen by selecting the “Alarm protocol” soft key.

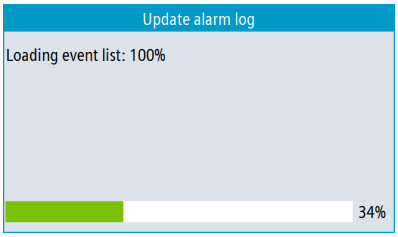


This shows the time when the alarm occurred and when it was cleared.



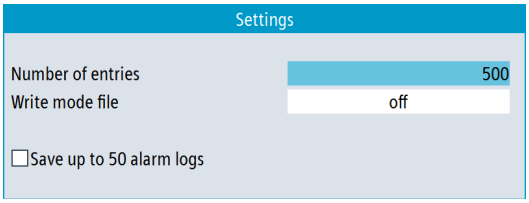
The “Refresh” soft key is used to refresh the list and display any alarms that have recently occurred.





The “Settings” soft key is used to configure the alarm log.



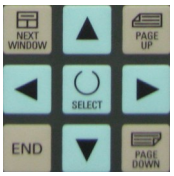


There are a number of options which can be selected by using the cursor and select keys and confirmed with the “OK” soft key:

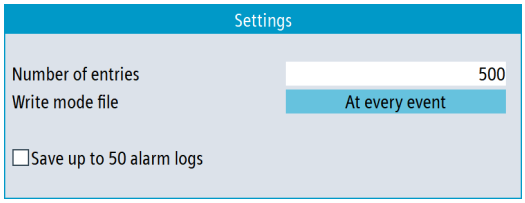
Number of entries:
This sets the maximum number of alarms displayed on the alarm log page.

Write mode file: Off
No storing of the log file will take place to the internal flash memory (SD card)
At every event: Each time an alarm occurs this will be recorded to the compact flash memory.
When this option is selected, a warning is displayed. By leaving this option permanently set, the life of the internal flash card (NCU) / SSD (IPC) will be reduced.

Notes:

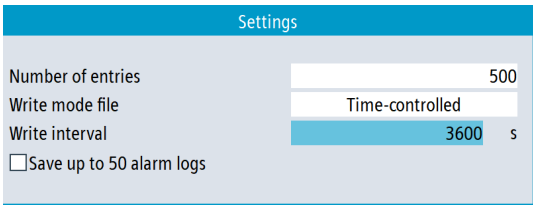








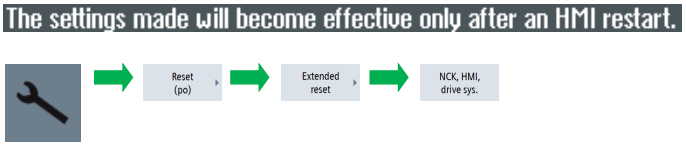
Write mode file: Time controlled:
The log will be saved at the interval entered in the Write interval area (max. 32000 seconds).





This will cause less wear and tear of the SD card but still carries similar risk of damage to the SD card.

The HMI will require restarting for the changes to become effective



Suppress Alarm Function

Certain alarms can be suppressed via General Machine Data 11410 & 11415.

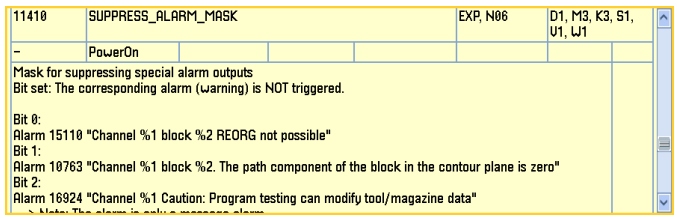


The bit editor is viewed by pressing the “select” key.



More detailed information is viewed with the “Help” key





015: END

Notes:

A017

SINUMERIK ONE

Onboard Monitoring and Tracing of Variables

Module Description:

The SINUMERIK ONE controller has onboard diagnostic capabilities to aid users with commissioning and fault finding procedures.

Monitoring:
It is possible to monitor the status of user PLC variables (inputs, outputs etc.) and also variables of the system (NC, axis, channel, Mode groups, global user data).
The display format of the variable being monitored can be chosen to suit (bit, byte, word, double word, decimal, floating point number, ASCII)

Tracing:
It is possible to trace the status of variables over defined time periods. Numerous variables can be traced at one time. The result is displayed in graphical format. Tools (cursors, zoom etc.) are available to facilitate the detailed analysis of the traces.
The following variables can be traced: system, NC, axis, channel, Mode groups, servo, drive, global user data, PLC, user PLC alarms.

Module Objective:

Upon completion of this module you will be able to:

- Monitor the status of User PLC Variables
- Monitor the status of system variables
- Use the trace function

Content:

NC/PLC Variable Formats

Data Block Status

Timer and Counter Status

NC/PLC Variable Comments

Saving the NC/PLC Variable List

System Variables

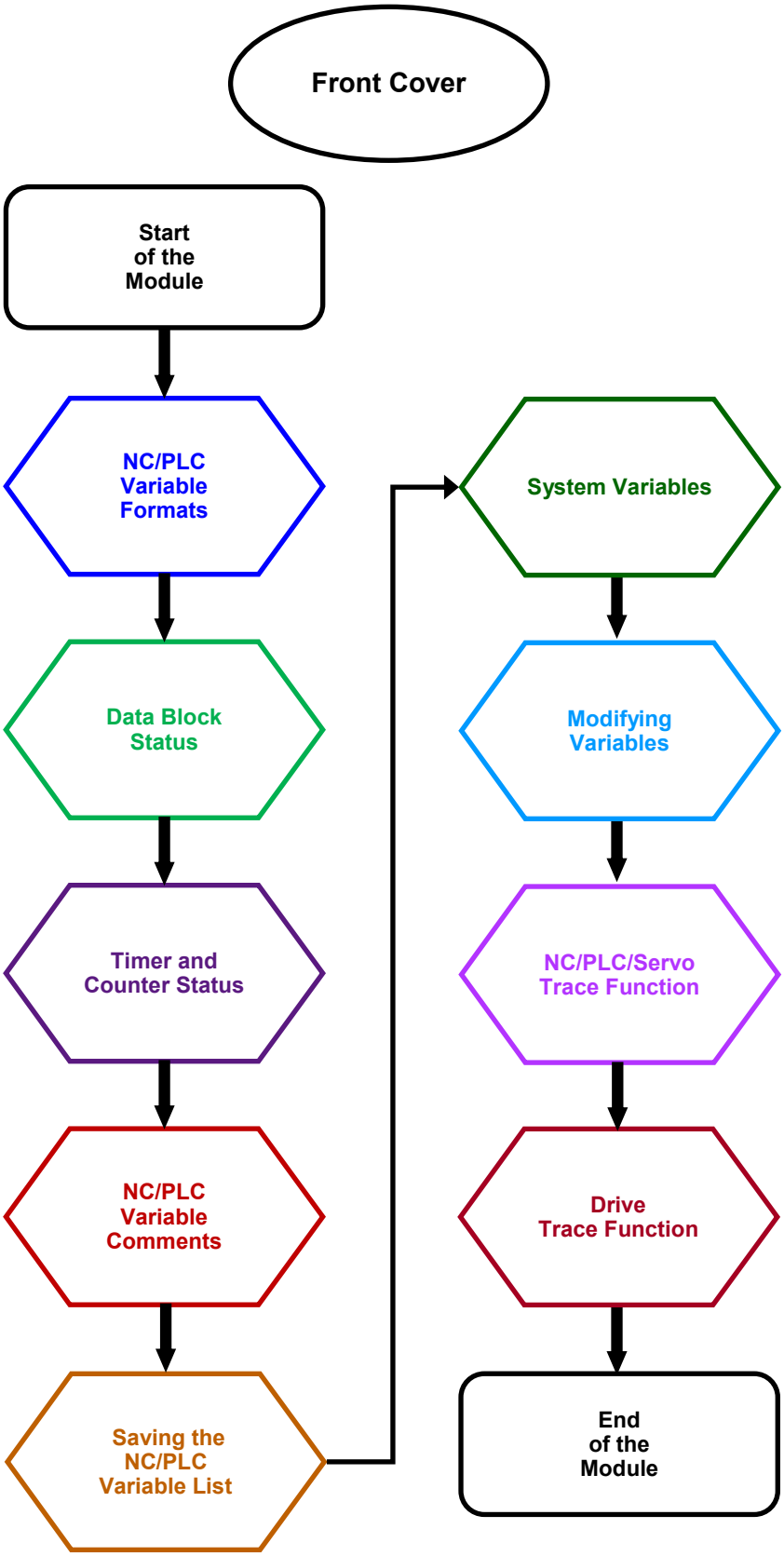
Modifying Variables

NC/PLC/Servo Trace

Drive Trace

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
NC/PLC Variable Formats
Data Block Status
Timer and Counter Status
NC/PLC Variable Comments
Saving the NC/PLC Variable List
System Variables
Modifying Variables
NC/PLC/Servo Trace Function
Drive Trace Function
End of the Module

PLC Variable Formats

Address Formats

The format of a variable (Operand), which is entered on the diagnostics screen, will be dependent on the type of data being diagnosed.
The format can be in single bit, byte, word & double word format.

Bit- This refers to a single element of logic e.g. I4.0, Q5.5, M100.6, DB31.DBX2.1. The Bit part is the number after the decimal point. The value of a bit is either 1 or 0.

Byte- A Byte is a collection of 8 bits of logic in the same group. e.g. Input Byte 4 (IB4) contains the bits I4.0, I4.1, I4.2, I4.3, I4.4, I4.5, I4.6, I4,7.
When working with bytes it must be remembered that the numbering counts up from the right as shown below.

	IB4							
Bit no	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**

In this example I4.1, I4.2, I4.5 & I4.7 are equal to 1. Another term for being equal to 1 is that these bits are High.
The bits equal to 0 are termed as being Low.

Output Bytes (QB), and Marker Bytes (MB) can also be used in the same way.

Word– A Word is a collection of 16 bits of logic which encompasses two consecutive Bytes.

A Word can only be addressed as 0 or an even number (**IW0, IW2, IW4** etc) odd numbers are not allowed.

As with the Byte format the numbering starts from the right.

Example **IW4**

	IB4								IB5							
Bit no	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**1**

In this example I4.0, I4.1, I4.4, I4.5, I4.6, I5.0, 5.2, I5.5, I5.7 are equal to 1.Output Words (QW) and Marker Words (MB) can also be used in the same way.
Double Word– A Double Word is a collection of 32 bits of logic which encompasses two consecutive Words (Four consecutive Bytes).

A Double Word can only be addressed as 0 or as a number which is a multiple of four, (**ID0, ID4, ID8, ID12** etc) odd numbers are not allowed.

Example **ID4**

	IB4								IB5								IB6								IB7							
Bit no	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**1**

Output Words (QD) and Marker Words (MD) can also be used in the same way.

Data Blocks (DB):
System Data Blocks are used for interfacing between different areas of the controller and the User PLC. ***Appendix C*** of the training documentation describes this in more detail.
User Data Blocks can be created by the users for their own specific requirements.

Data Blocks can be written in different formats e.g. :

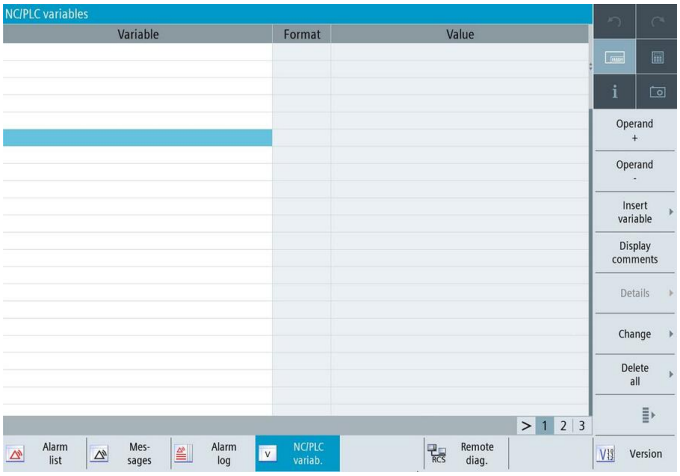
DB31.DBX0.0	Data Block 31, Data Block Bit 0.0
DB31.DBB1	Data Block 31, Data Block Byte 1
DB31.DBW10	Data Block 31, Data Block Word 10
DB31.DBD12	Data Block 31, Data Block Double Word 12

Notes:

Locate the ***NC/PLC variable*** screen from within the ***Diagnostics*** menu.

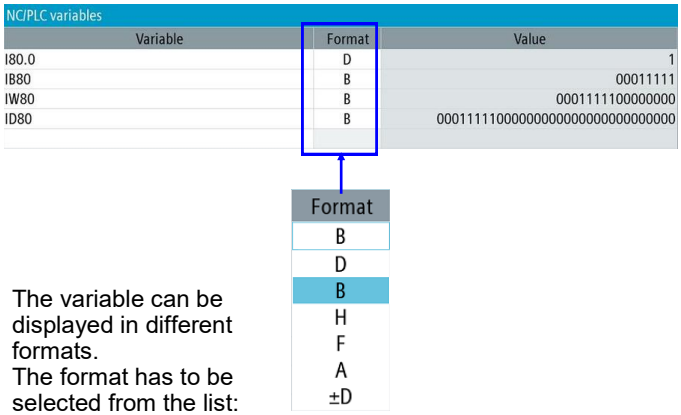


The following screen will be displayed, it is possible that there are entries already present on the screen.



The name of the PLC variable to be monitored needs to be typed into the ***Variable*** column (Bit, byte, word, double word).

Example - inputs being used to show the different display format possibilities.



The variable can be displayed in different formats.
The format has to be selected from the list:

Format description:

Formats	Description
B	Binary
H	Hexadecimal
D	Decimal without sign
±D	Decimal with sign
F	Floating point number (for double words)
A	ASCII characters

Variable	Possible Format
Inputs	Ix.x, IBx, IWx, IDx
Outputs	Qx.x, QBx, QWx, QDx
Bit memory	Mx.x, MBx, MWx, MDx
Timers	Tx
Counters	Cx
Data Blocks	DBx.DBXx.x, DBx.DBBx, DBx.DBWx, DBx.DBDx

The variable will be displayed in the selected format:



PLC variables are the most commonly checked during fault finding, e.g. inputs from switches, outputs to coils etc.

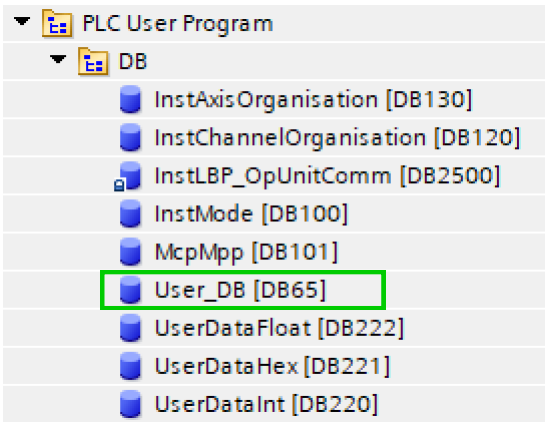
The following table shows the PLC variables that can be monitored:

Notes:

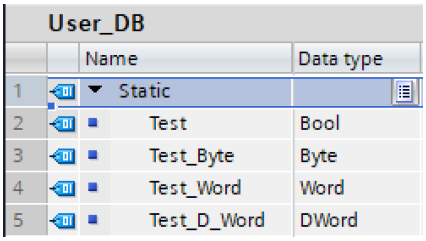
Data Block Status

User Data Blocks

Example - User_DB [DB65] in the PLC project:



User_DB [DB65] data entries:

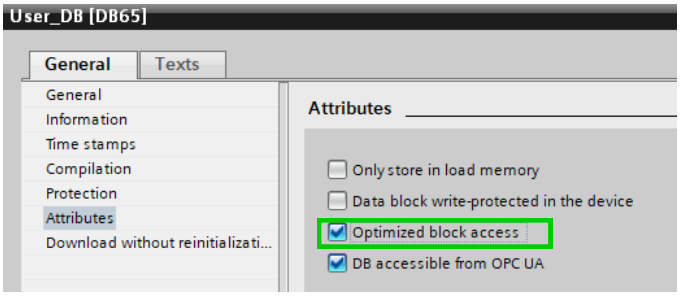


Status Display Format

It is possible to view the status using absolute or symbolic formats, depending on the ***Optimized block access*** setting. This is found in the properties of the relevant data block.

Any changes to the settings will require the data block to be downloaded to the controller.

Example - the properties of ***User_DB [DB65]*** data block showing the default setting of the ***Optimised block access*** attribute:



The default setting requires the variable to be entered in symbolic format on the ***NC/PLC variables*** screen.

Note: Symbolic information should be sought from the manufacturer's technical documentation.

Example - status display of ***User_DB.Test*** can only be monitored in ***Symbolic*** format:

NC/PLC variables

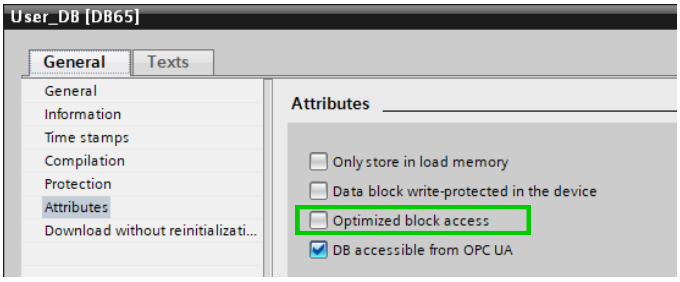
Variable	Format	Value
"User_DB".Test	D	0
DB65.DBX0.0	B	#

Example - status display of ***User_DB.Test*** can now be monitored in ***Symbolic*** and ***absolute*** formats:

NC/PLC variables

Variable	Format	Value
"User_DB".Test	D	0
DB65.DBX0.0	B	0

Absolute addressing - to view the Data Block status in absolute format the ***Optimised block access*** attribute needs to be deselected:



System Data Block Status

The ***Optimised block access*** attribute setting for system data blocks is deselected.
This allows the data block status to be displayed in absolute format.

Example - Axis data block ***DB31.DBX2.1***

NC/PLC variables

Variable	Format	Value
DB31.DBX2.1	D	1

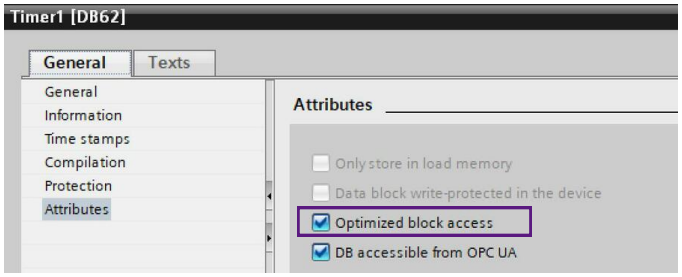
Notes:

Timers and Counters

Instance Data Blocks

Timers and counters use instance data blocks for data handling.
The contents of the data block can be displayed on the status screen.
It is possible to view the status using absolute and/or symbolic formats, depending on the ***Optimized block access*** setting.
This is found in the properties of the relevant instance data block.

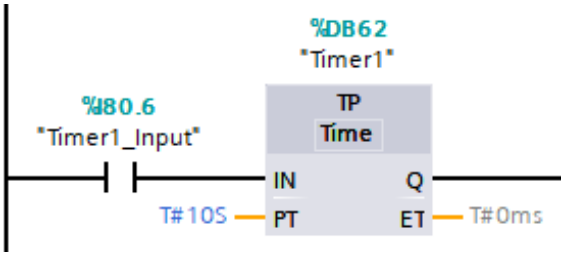
Example - properties of ***Timer 1 [DB62]*** instance data block showing the default setting of the ***Optimised block access*** attribute:



The default setting requires the variable to be entered in symbolic format on the ***NC/PLC variables*** screen.

The parameter values of timers and counters can be monitored.
The symbolic name of the timer/counter is required.
The symbolic addresses are chosen by the person responsible for the PLC and they will vary between machine manufacturers.
Each of the parameters of the timer/counter can be monitored.

Example - ***Timer1*** in the PLC project:

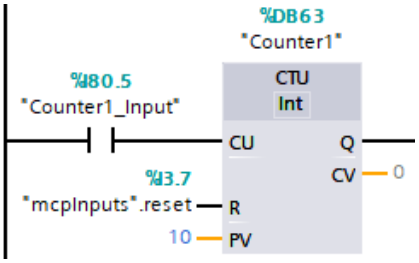


Example - monitoring the parameters of a ***TP*** (***Pulse)*** timer which has the symbolic name of ***Timer1:***

NC/PLC variables

Variable	Format	Value
"Timer1".PT	D	10000
"Timer1".ET	D	0
"Timer1".IN	D	1
"Timer1".Q	D	1

Example - ***Counter1*** in the PLC project



Example - monitoring the parameters of a ***CTU*** (***count up)*** counter which has the symbolic name of ***Counter1:***

NC/PLC variables

Variable	Format	Value
"Counter1".CU	D	1
"Counter1".R	D	0
"Counter1".PV	D	10
"Counter1".CV	D	5

Note: Symbolic information should be sought from the manufacturer's technical documentation.

Notes:

NC/PLC Variable Comments:

The ***Display Comments*** soft-key will display any pre-existing comments and allow additional ones to be added.



NC/PLC variables	
Variable	Comment
I6.0	
I6.1	
Q4.0	
Q4.1	

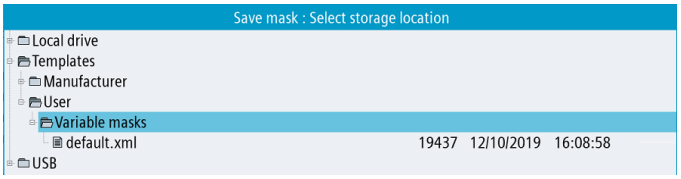
Useful comments can be added:

NC/PLC variables	
Variable	Comment
I6.0	Turret clamped switch
I6.1	Turret unclamped switch
Q4.0	Turret clamp solenoid
Q4.1	Turret unclamp solenoid

Saving the NC/PLC Variable List

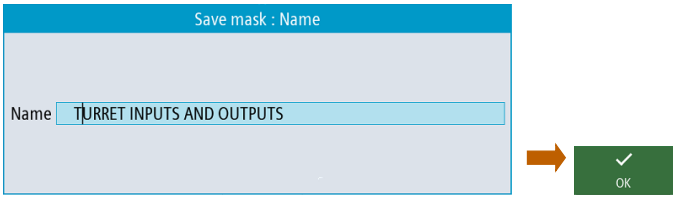
It may be useful to save the screen entries as a mask. Multiple masks, each relating to a specific area of the machine, for example, can be saved both locally or externally.





Choose the destination and enter a suitable name:





Loading the Mask

Masks are easily loaded.



Choose the source, locate the mask and select the ***OK*** soft-key



Any entries on the screen will be overwritten by the mask contents.

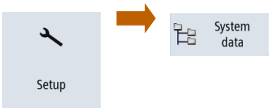
The default mask contains no entries and will clear all entries from the screen.

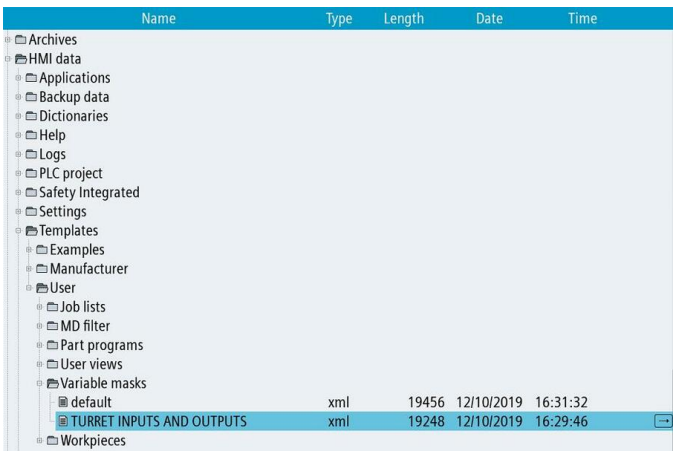
Note: Virtual machine to real machine

The templates are not transferred via archive from the virtual machine to the real machine.
The templates must be manually transferred to the real machine.

Local storage Location for the Masks

The locally saved masks are stored in the following default location:





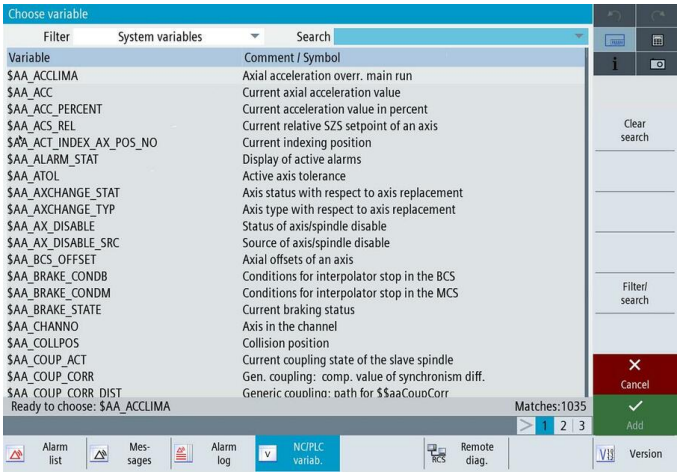
The masks can be conveniently transferred to a USB device for use in other machines.

Notes:

System Variables

A list of available system variables is displayed via the ***Insert variable*** soft-key.
The available variables are listed when the soft-key is selected.



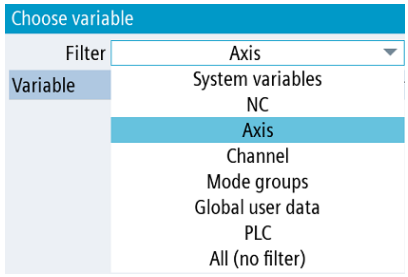


The types of variable available for selection are:

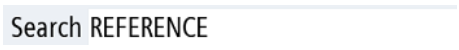
System variables
NC
Axis
Channel
Mode groups
Global user data
PLC

The filter is used to narrow down the variables on view.

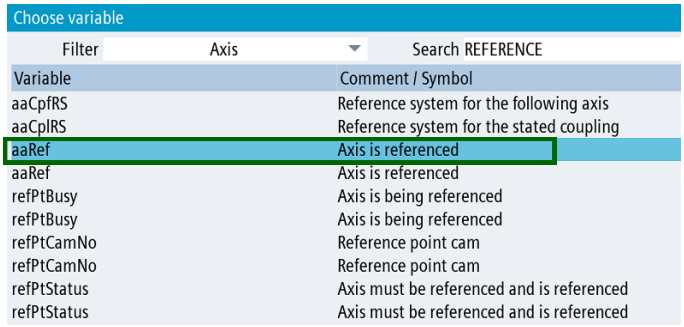




The search window can be used to locate the required variable. Partial or whole names can be entered.

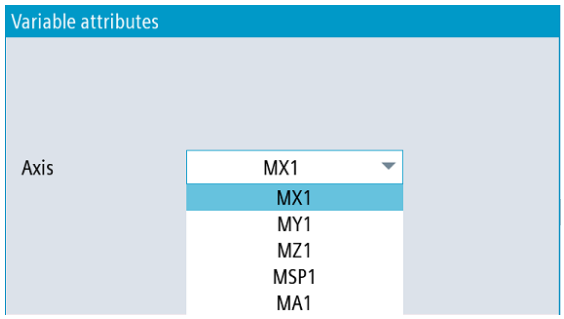


Choose the variable from the list, e.g. ***aaRef Axis is referenced.***



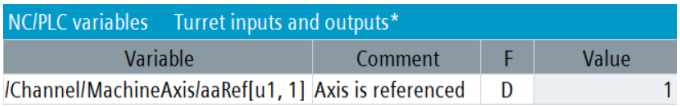


The variable used in this example is an axis variable, therefore an axis has to be chosen.





The chosen variable will appear on the screen.



Notes:

The values within the square brackets identify the individual variables e.g.:

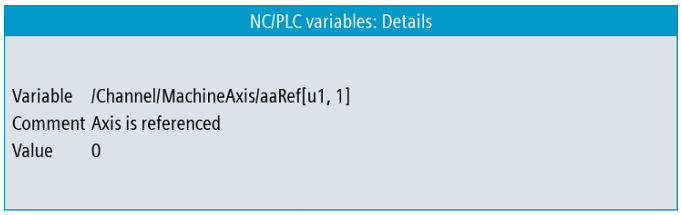


U1= Channel number
1 = Axis number

Details:



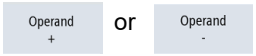
The Details soft-key shows information about the variable.

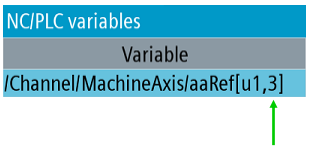




Operand +/- Softkeys

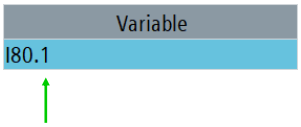
The ***Operand +*** and ***Operand -*** soft-keys are used for locating operands (Variables) within the same group.





The axis No. can be incremented/decremented

The operand +/- function can also be used with other variables:



The bit number of a variable can be incremented/ decremented if it is displayed in bit format.

Likewise, the byte, word & double word numbers can be incremented/decremented for variables displayed in these formats

Modifying Variables

For test purposes it may be necessary to manually change (Force) a variable.
This is achieved with the ***Change*** soft-key.

The operand to be changed has to be highlighted and the ***Change*** soft-key selected.

Variable	Comment	F	Value
M50.0		B	0



The ***Value*** column is highlighted. The new value needs to be entered here

Example - changing M50.0 from a value of 0 to a value of 1

Variable	Comment	F	Value
M50.0		B	0

Variable	Comment	F	Value
M50.0		B	1



The variable is now set to the specified value:

Variable	Comment	F	Value
M50.0		B	1

It is also possible to modify the variables in all the formats, depending on the functionality of the individual variable:

B	Binary
H	Hexadecimal
D	Decimal without sign
±D	Decimal with sign
F	Floating point number (for double words)
A	ASCII characters

Note:
It is useful to have this function in the commissioning stages. After data loss, it may also be necessary to enter values into Data Blocks using this method. This depends upon the machine manufacturer.

Caution should be taken when attempting to manually enter data as unwanted movements may occur.

It is not possible to force operands which are being addressed by the PLC program.

Notes:

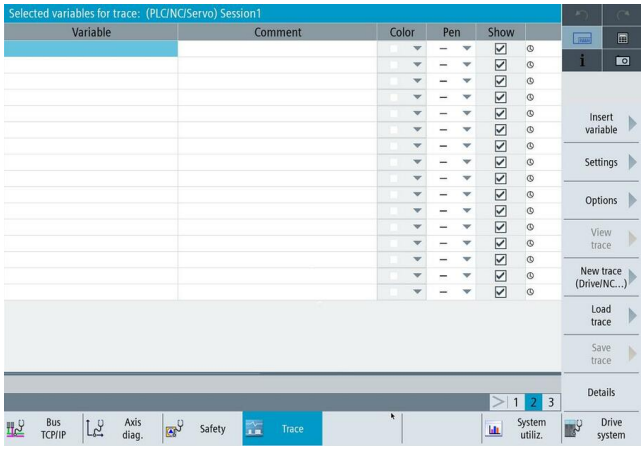
NC/PLC/Servo Trace Function

The ***Trace*** function is used to trace the status of variables over a given time. The results of the trace are displayed on the screen in a graph format.

Virtual Machine Note: full functionality is not available on the virtual machine.
The following examples are shown using a real machine.

To locate the ***Trace*** screen, follow the soft-key sequence:



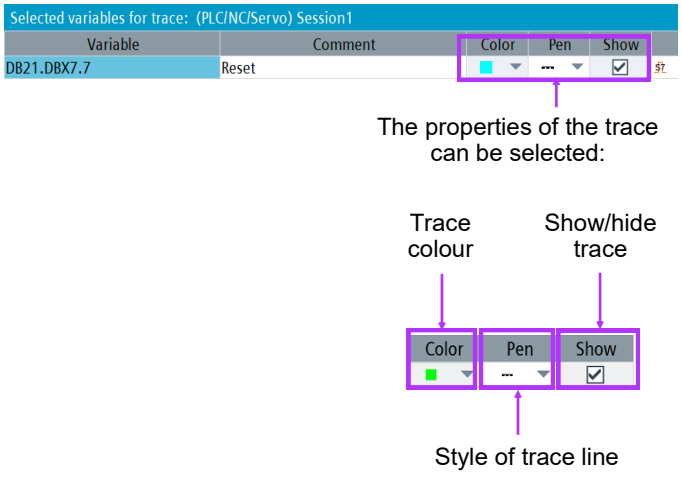


PLC/NCK/Servo Variables.

Many types of signals can be traced.
The default setting is for the tracing of PLC/NCK/Servo variables. It is also possible to trace drive variables (described later)

Manual Entries:
The variables can be typed directly into the ***variable*** column:

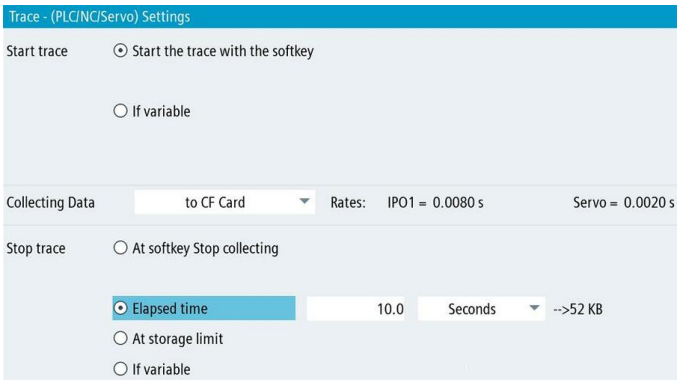
Example - DB21.DBX7.7 (channel 1 reset signal controlled by MCP reset button) entered as the variable to be traced:



Trace Settings



The trace settings are used to set up the conditions for starting and stopping the trace.



The default trace settings are:

- Start the trace with softkey.
- Collecting data to the CF card.
- Stop the trace after 10 seconds.

Note: This is for a real machine without IPC.

Description of the Trace Settings

Setting	Description/comments
Start Trace	Condition for starting the trace
Start the trace with the softkey	The trace is started via the ***Start trace*** softkey
If variable	The trace is started by the condition of a variable. The variable is chosen by the user.
Collecting Data	Trace data storage location.
to CF card/NCK	Real machine - No IPC
to Hard disk/NCK	Real machine - with IPC
to Hard disk/NCK	Virtual machine
Stop Trace	Condition for stopping the trace
At softkey stop collecting	The trace is started via the ***Stop trace*** softkey.
Elapsed time	The trace is stopped after the specified time. (milliseconds, second, minutes, hours, days)
At storage limit	The trace is stopped when the trace reaches the specified size - limit depends on whether the destination is CF card, Hard Disk or NCK
If variable	The trace is stopped by the condition of a variable. The variable is chosen by the user.
Post triggered with...	The stop trace signal can be delayed. (only with ***If variable*** setting)
Discard oldest data at storage limit	On lengthy traces the oldest data will be discarded when the specified storage limit is reached. (only with ***If variable*** setting)

Notes:

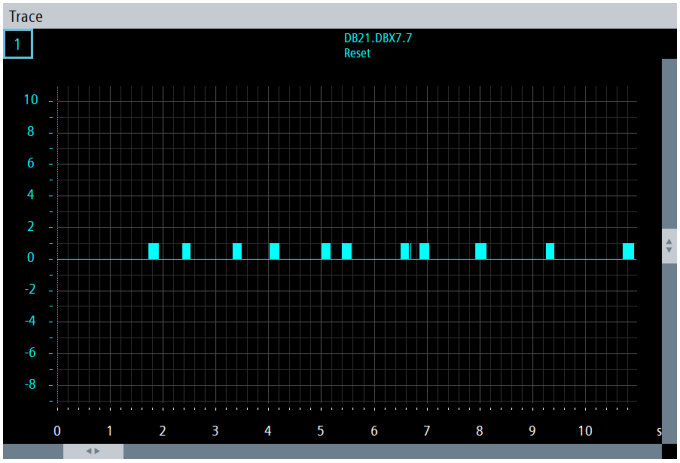
Manually Starting and Viewing the Trace:

The following example will use the default settings to trace DB21.DBX7.7 (via MCP Reset button)



- Press the MCP reset button several times within 10 seconds.

After 10 seconds the trace will stop and the results can be analysed.



Scaling and Zooming

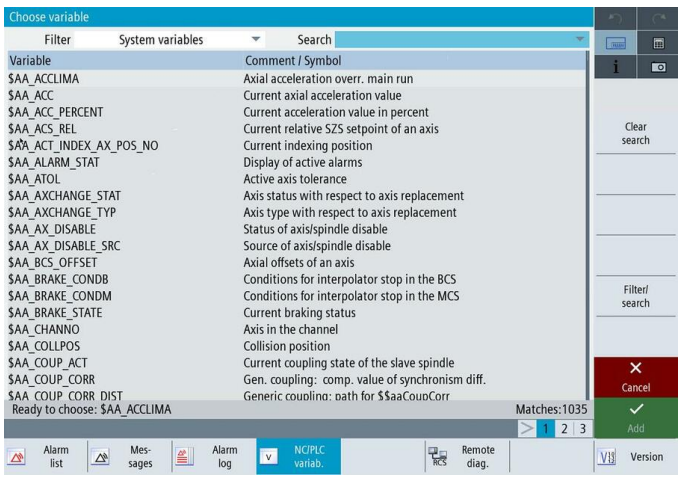
Various options are available to achieve a more detailed analyses of the trace.



Insert Variable Soft-key:

System variables can be chosen from the lists.

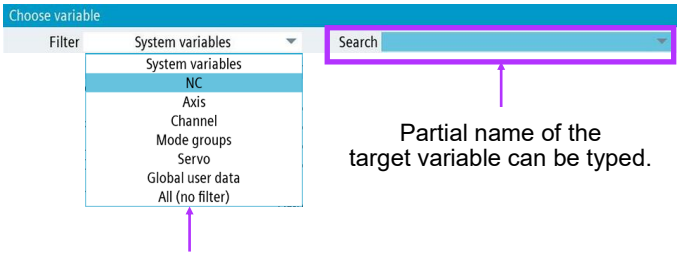




Filter/Search

The Filter/Search feature is available for use in locating the required variable.





Once the correct variable has been located, confirm with the ***Add*** softkey:



The selected variable will now appear on the variables screen in readiness for the trace to be activated.

Notes:

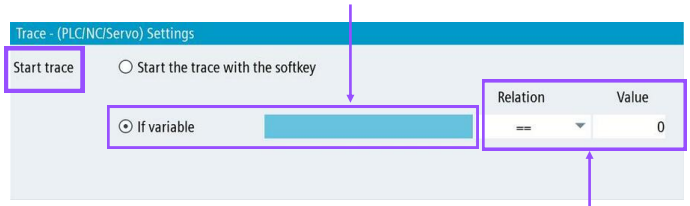
Starting and Stopping the Trace with a Variable:

The ***If variable*** setting allows the trace to be started and stopped with a variable.



Start Trace:

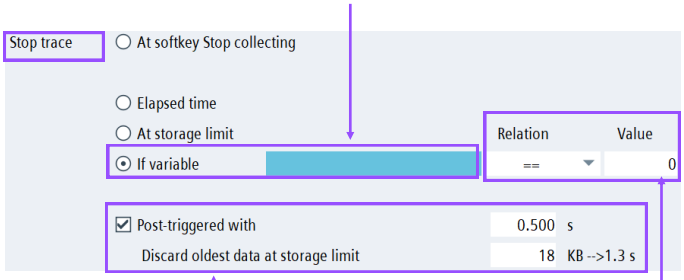
The variable for starting the trace can be entered directly.
Alternatively it can be chosen via the ***Insert variable*** or ***Quick list*** soft-keys.



The ***Relation*** and ***Value*** settings are used determine the variable status required to start the trace.

Stop Trace:

The variable for stopping the trace can be entered directly.
Alternatively it can be chosen via the ***Insert variable*** or ***Quick list*** soft-keys.



The ***Relation*** and ***Value*** settings are used determine the variable status required to stop the trace.

Post-triggered with: an additional setting is available to delay the stopping of the trace after the ***Stop trace*** condition is met.

Discard oldest data at storage limit: on long traces the limit can be set at which point the oldest data is deleted.

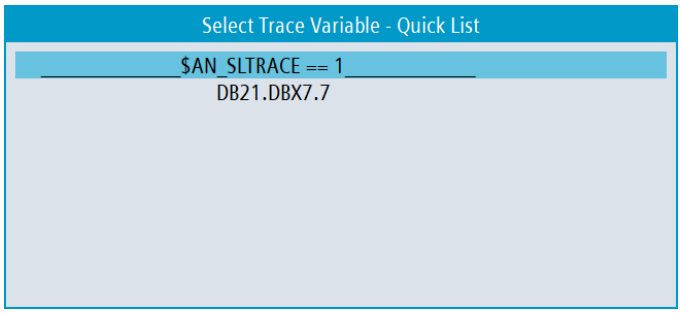
***If variable* description:**

The variable for starting or stopping the trace can be manually entered or selected from a list.

Quick list - the Quick list soft-key will show the variables already selected for tracing plus a system variable - ***$AN_SLTRACE ==1***.
This system variable can be controlled from the part program.

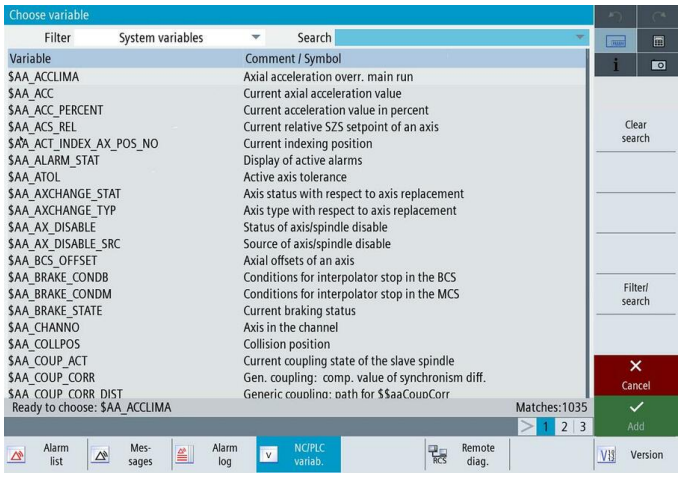
An example of how to use the ***$AN_SLTRACE*** variable is described later in the module





Insert variable - a suitable variable for starting or stopping the trace can be selected from the lists.

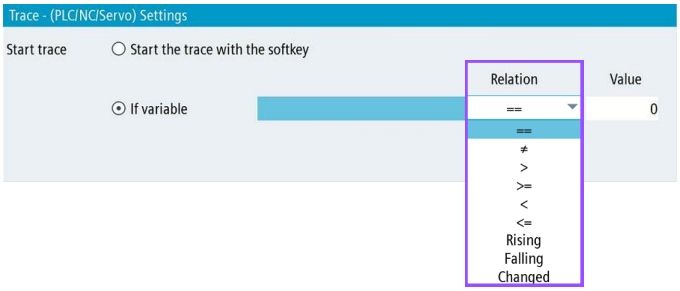




Notes:

Relation and Value

The ***Relation*** and ***Value*** settings are used define the conditions of the variable required to start/stop the trace.

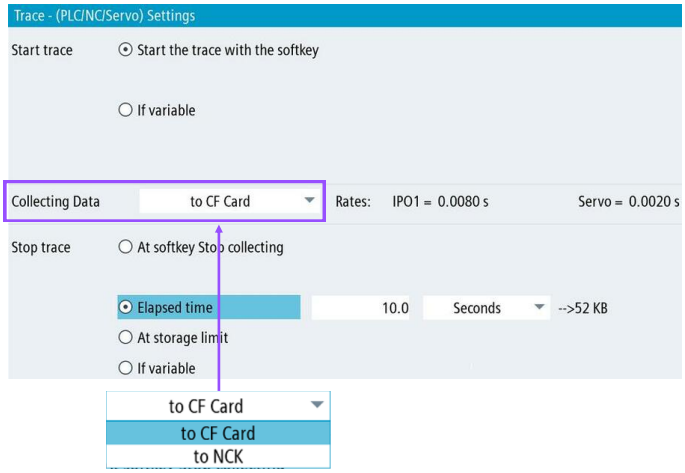


Description of the Trace Start Conditions:

Trace Start Conditions	
Relation	**Variable Status**
==	Equal to the ***Value***
≠	Not equal to the ***Value***
>	Greater than the ***Value***
>=	Greater than or equal to the ***Value***
<	Less than the ***Value***
<=	Less than or equal to the ***Value***
Rising	***Rising edge*** of the ***Value***
Falling	***Falling edge*** of the ***Value***
Changed	***Change of state*** of the ***Value***

Collecting Data.

The collecting data setting is used to determine the storage location of the trace file.



The choices will differ depending on whether an IPC or Virtual Machine is in use.

Collecting Data	Equipment
to CF(SD) card/NCK	Real machine - No IPC
to Hard disk/NCK	Real machine - with IPC
to Hard disk/NCK	Virtual machine

Trace storage information.

- With ***to NCK,*** the recorded data is saved in the NC memory until the recording is stopped.
  Only then is the session file saved on the CF(SD) card.
- With ***to CF card*** or ***to hard disk***, the data is saved during the recording on the CF(SD) card or hard disk (e.g. of the IPC).
  As a consequence, more signals can be processed. and for longer time periods.

Collecting data to the NCK offers the least trace time/storage space.
The NCK trace time is limited to milliseconds/seconds/minutes.

Collecting data to the CF(SD) card/hard drive offers trace times in milliseconds/seconds/minutes/hours/days.

All of the above are influenced by the sample rates - see below:

Sample Rates

The sample rate should be chosen to suit the trace requirements.

Smaller sample rates offer higher accuracy, for example, when tracing positional values.

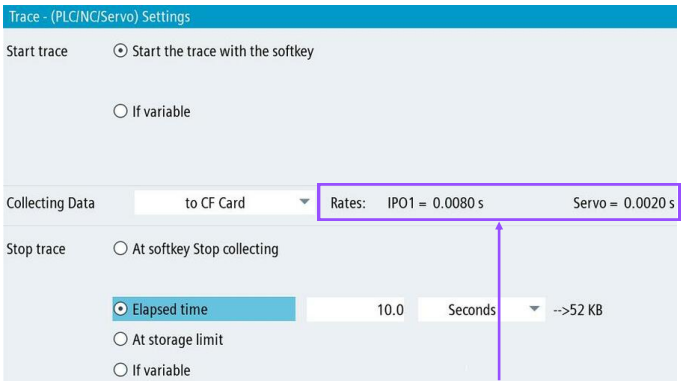
Notes:

The restriction of using smaller sample rates is that the trace periods are limited in size and length.

Another restriction is caused by the number of variables being traced. More variables results in less trace storage capacity. This is especially true if collecting data to the NCK.

The user has to decide which sample rate suits the trace requirements.

The active sample rates are displayed along side the collecting data selection



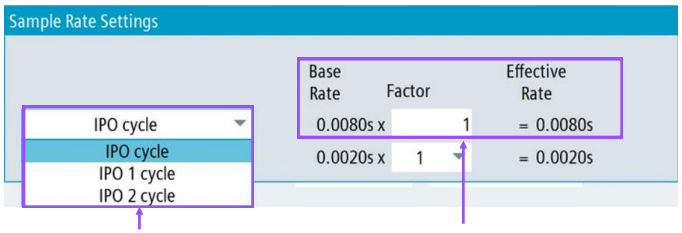
Current sample rates



Available settings:

The sample rates can be adjusted by applying a factor to the base rate of the interpolator clock cycle (IPO) or the Servo cycle.

IPO sample rate selection:
Three different IPO values can declared (IPO, IPO1, IPO2). The required IPO cycle is chosen from a drop down menu.



Selecting the required IPO cycle.

A factor can be entered (1 to 7500)

The IPO Cycle sample rate settings are effective when tracing the following variables:

- NC
- Axis
- Channel
- Mode groups
- Global user data
- PLC

Note: the trace destination, number and type of variables, plus the sample rate factor all have an effect on the maximum trace length/size.

Elapsed time/at storage limit: The elapsed time/storage limits are inextricably linked and both are affected in same way by the sample rate.

Example - tracing a Channel signal with default sample settings:

To the CF(SD) card - max 16.5 days.



Collecting data to Hard Drive - max 16.5 days.



Collecting data to NCK- max 4.5 minutes.



In the following example the IPO cycle factor was set to 2 (0.016 s). This allows the maximum trace time of 30.0 days for collecting to the CF (SD) card.



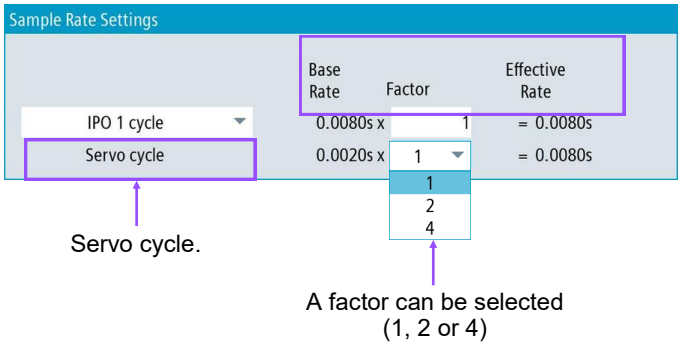
Notes:

Servo sample rate selection

The Servo cycle sample rate settings are effective when tracing the following variables:

- Servo

The required IPO cycle is chosen from a drop down menu.



Note: the trace destination, number and type of variables, plus the sample rate factor all have an effect on the maximum trace length/size.

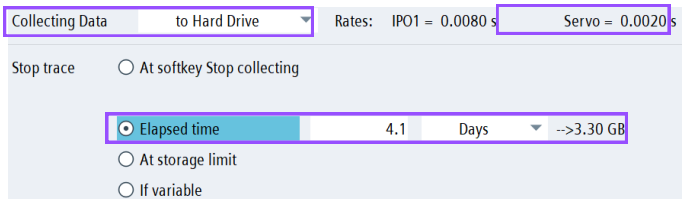
Elapsed time/at storage limit: The elapsed time/storage limits are inextricably linked and both are affected in same way by the sample rate.

Example - tracing a servo signal with default sample settings:

To the CF(SD) card - max 4.1 days.



Collecting data to Hard Drive - max 4.1 days.



Collecting data to NCK - max 2.7 minutes.



In the following example the servo cycle factor was set to 4 (0.008 s). This allows the maximum trace time of 16.5 days for collecting to the CF (SD) card.



Description and example of the *$AN_SLTRACE* variable

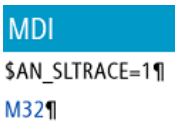
The ***$AN_SLTRACE*** variable can be set from the machining programs and cycles. This gives great flexibility for starting and stopping the trace.

Example: Using the ***$AN_SLTRACE*** signal to start the tracing of DB21.DBX7.7 (via MCP reset button)

The trace will be started from ***MDA mode*** by setting the variable, ***$AN_SLTRACE,*** to 1.



Enter the following in order to set the variable to 1 when the cycle start button is pressed.



Locate the trace setup screens:



Notes:

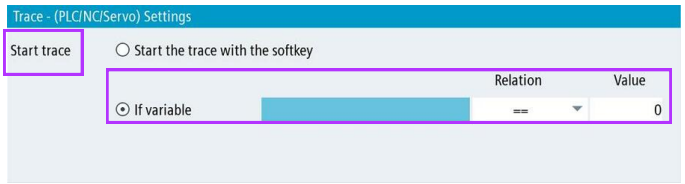
DB21.DBX7.7 is the variable to be traced:



Selecting the variable for starting the trace:

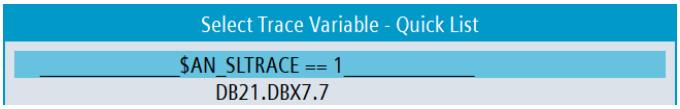


Select the ***Settings*** area and select **Start trace/ If variable** setting.





Select Quick list soft-key. The default condition for the variable is ***$AN_SLTRACE == 1.***
The trace will be started when the variable is equal to 1.



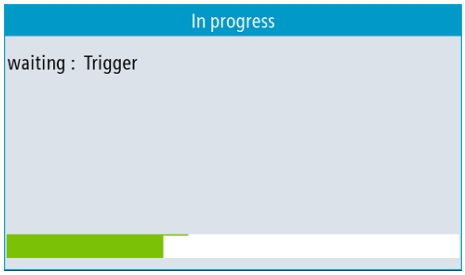






The conditions are now set for the signal to be traced.

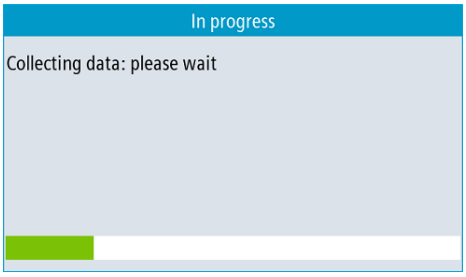




Waiting: Trigger - The trigger is the $AN_SLTRACE = 1 signal from the MDI program.

- Press the cycle start button:

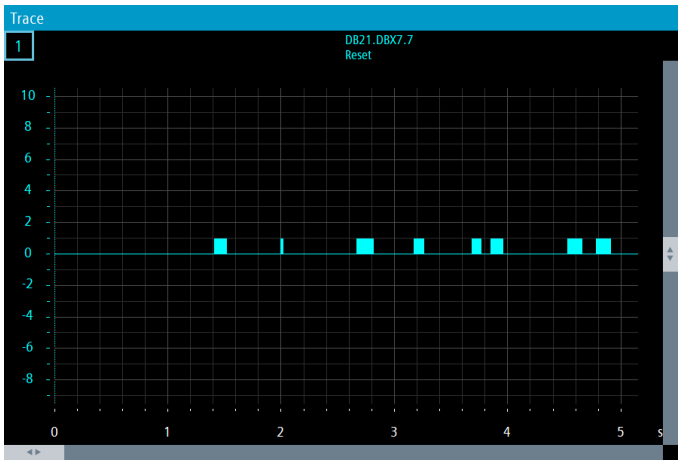




- Press the MCP reset button, several times, within 10 seconds.

After 10 seconds the trace will stop and the results can be analysed.

The result of the tracing of DB21.DBX7.7 :



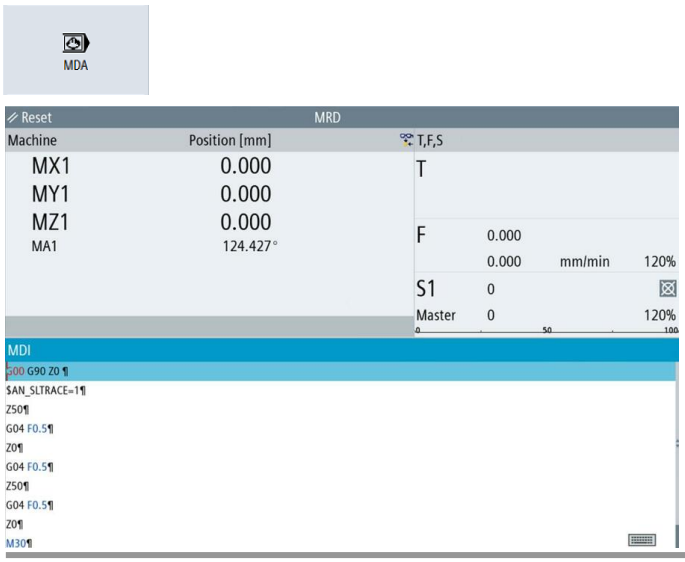
Notes:

Tracing Axis Positions

The following example shows the tracing of the 1st and 2nd measuring circuits of an axis fitted with two encoders.

The measuring circuits of the Z axis are to be traced.
The trace will be triggered from MDA mode using the variable $AN_SLTRACE.

An example program can be seen below:



Program explanation:

G00 G90 Z0	Select rapid traverse. Select absolute positioning. Position Z axis to 0mm.
$AN_SLTRACE=1	Trigger the trace.
Z50	Position Z axis to 50mm.
G04 F0.5	Dwell for 0.5 seconds.
Z0	Position Z axis to 0mm.
G04 F0.5	Dwell for 0.5 seconds.
Z50	Position Z axis to 50mm.
G04 F0.5	Dwell for 0.5 seconds.
Z0	Position Z axis to 0mm.
M30	End of program.

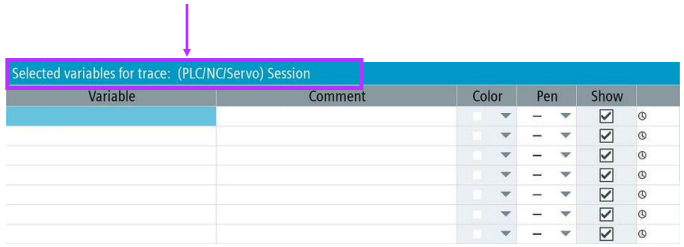
The program is now ready and will be used once the trace signals and settings have been selected.

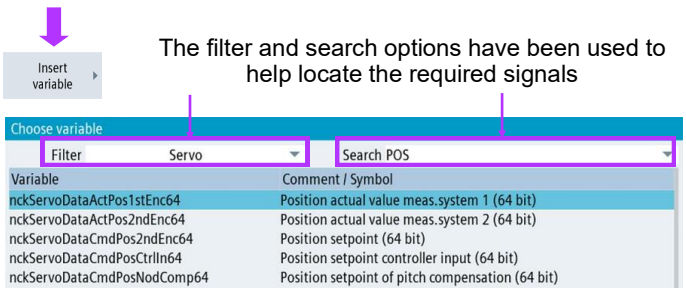
Selecting the trace signals:

Locate the trace setup screens:



The signals to be traced are servo signals





The filter and search options have been used to help locate the required signals

The signals to be traced are:

nckServoDataActPos1stEnc64
(Position actual value meas. System 1 - 64bit)

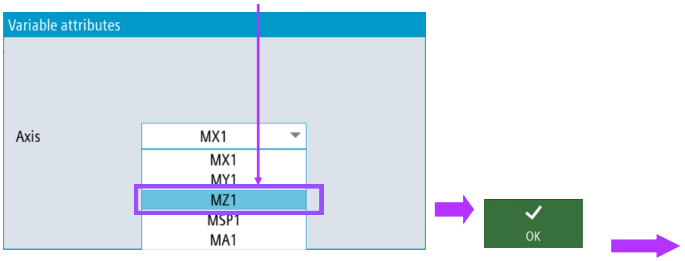
nckServoDataActPos2ndEnc64
(Position actual value meas. System 2 - 64bit)

nckServoDataCmdPos2ndEnc64
(Position Setpoint - 64bit)

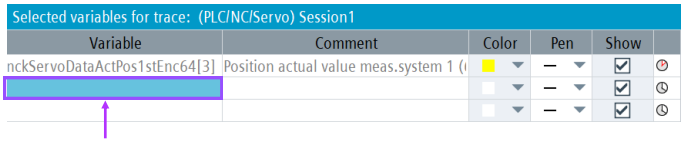
Select and add the first signal:



Choose the Z axis from the list of available axes



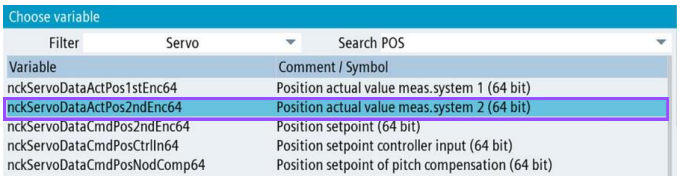
Notes:



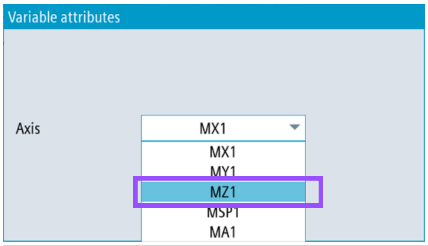
Select the 2nd row.

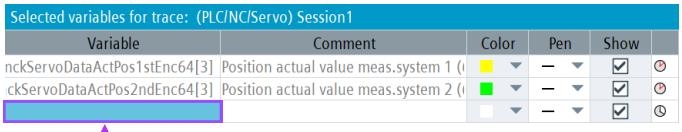
Insert variable

Select the next required variable:



Add

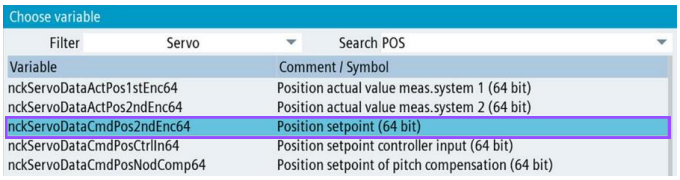




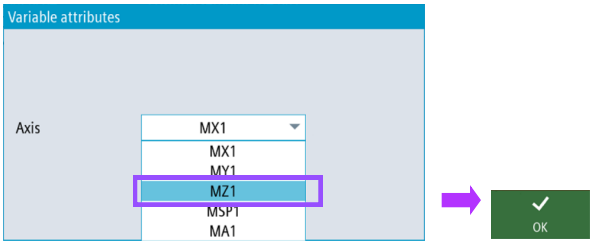
Select the 3rd row.

Insert variable

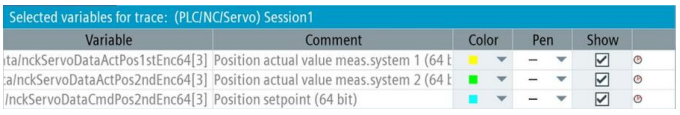
Select the next required variable:



Add



The required selection is complete:

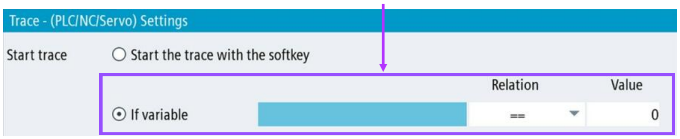


Trace settings:

The trace start and trace length requirements need to be set.

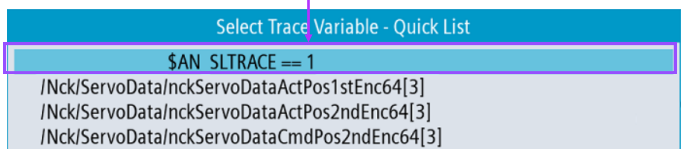
Settings

The trace start signal is to be initiated from the MDA program - select ***If variable:***



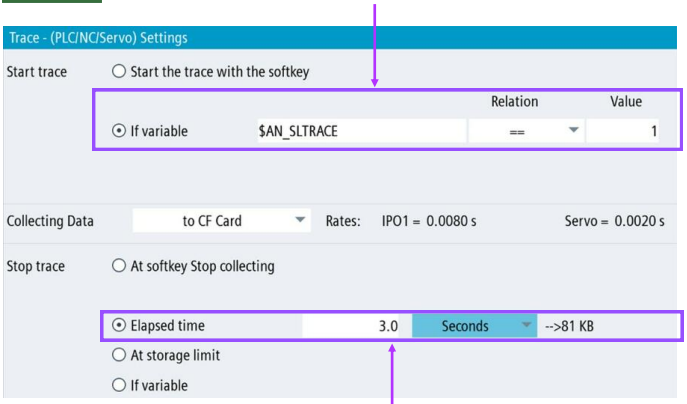
Quick list

The required variable, ***$AN_SLTRACE***, is located on the ***Quick list.***



OK

The trace will start when, ***$AN_SLTRACE = 1.***



The example requires that the trace will run for 3 seconds.

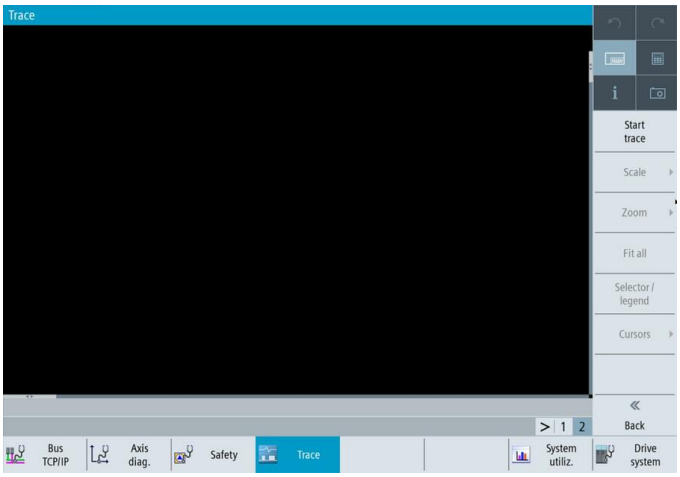
The required signal selection and trace set-up is complete.

OK

Notes:

Initialising and Starting the Trace.

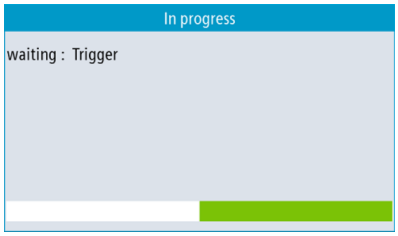






The ***Start trace*** soft-key initialises the trace but doesn't actually start the trace. This is because the ***If variable*** setting has been chosen.

The ***Waiting trigger*** message is displayed:

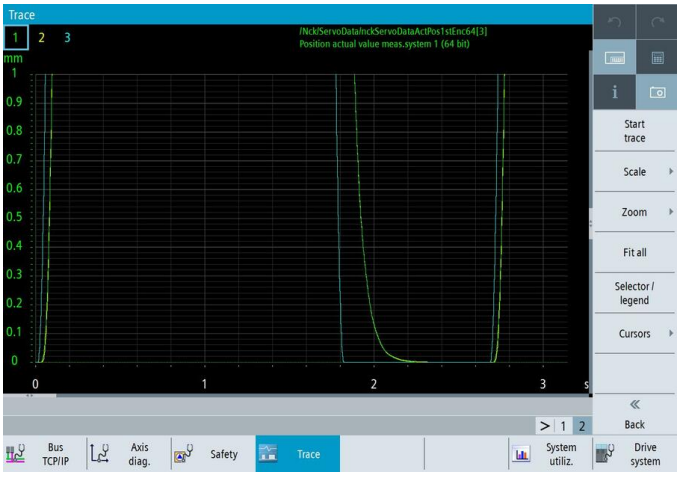


Conditions required for starting the trace:

- Emergency Stops buttons released.
- Drive switched on, enabled and operational.
- Feed-hold button selected.
- Feed-rate at 100%.
- MDA mode selected.
- Press the Cycle Start button.



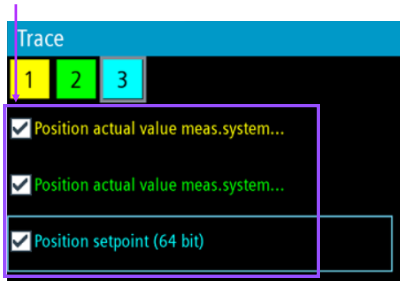
The axis will move several times as programmed.
After 3 seconds the trace will stop and the results will be displayed:



In order to compare the traces together it is necessary to change some settings.

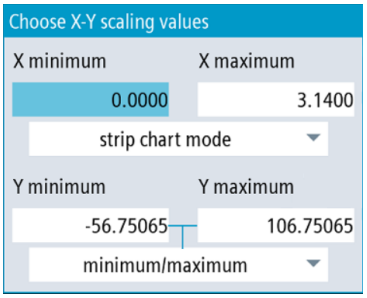


Select all three traces via the boxes on the left side.



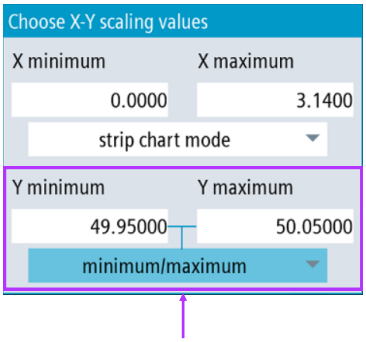


The Scale feature is used to accurately highlight a section of a trace.



Notes:

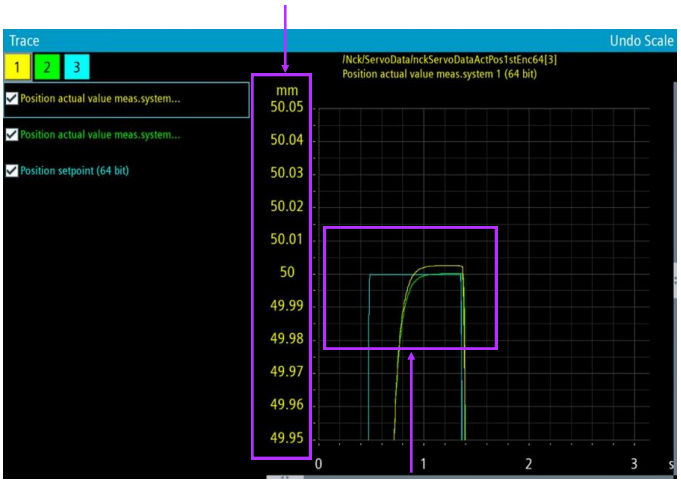
The following example shows the trace scale settings for examining the axis position at 50mm (Y axis plane).



The Y axis scale has been set with a minimum value of 49.95mm and a maximum of 50.05mm.



Y axis is now scaled as required (49.95mm to 50.05mm)



A detailed examination of the axis positional behaviour, as it approaches 50mm, can be seen.

Note: this example was performed on an untuned axis with standard data loaded.

Notes:

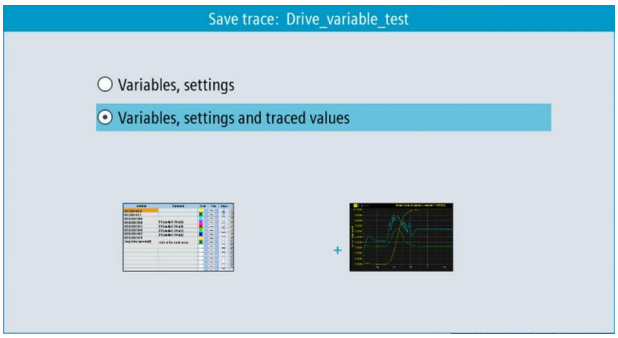
Saving and Loading Trace Files:

Save Trace.
Files containing the trace information can be saved locally on the controller or externally to USB memory devices or networked computers.

Note: save options are available via the ***Trace options*** soft-key. These are described on the next page.

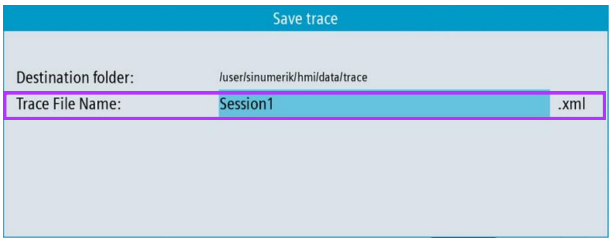


The trace variables and settings can be saved without traced values or with traced values (default):



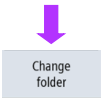


The trace file name can be entered or modified.

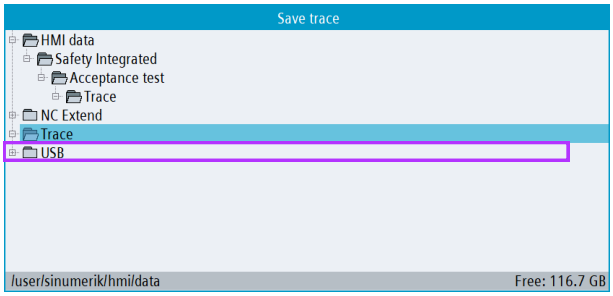




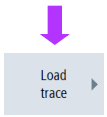
The file will be saved to the default location - *user\Sinumerik\hmi data\ trace.*



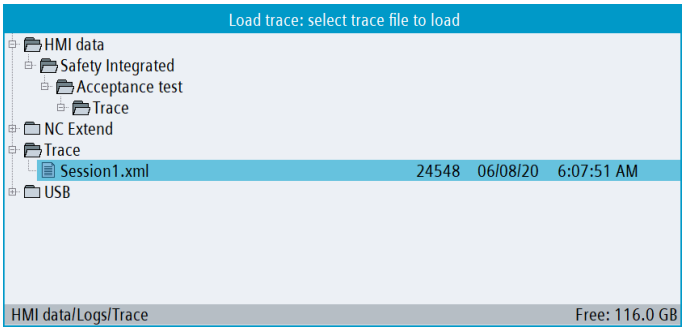
Selecting the ***Change folder*** soft-key allows the destination to be changed to a USB device, for example.



Load Trace.
Files containing the trace information can be loaded from local locations of the controller or from external locations such as USB memory devices or networked computers.

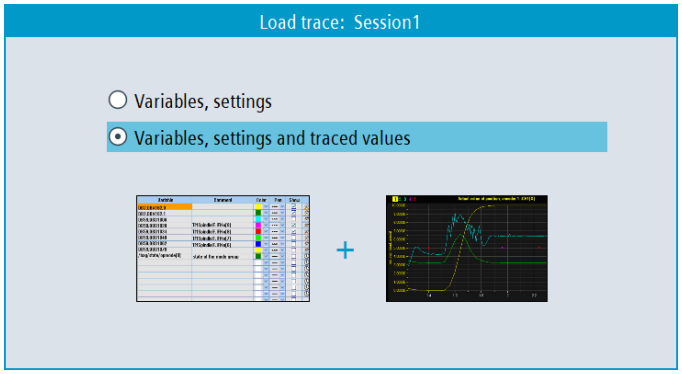


Locate the file to be loaded:





If the trace file contains traced values it can be decided whether to include them in the load operation:





The trace information which was included in the trace file will now be available for viewing.



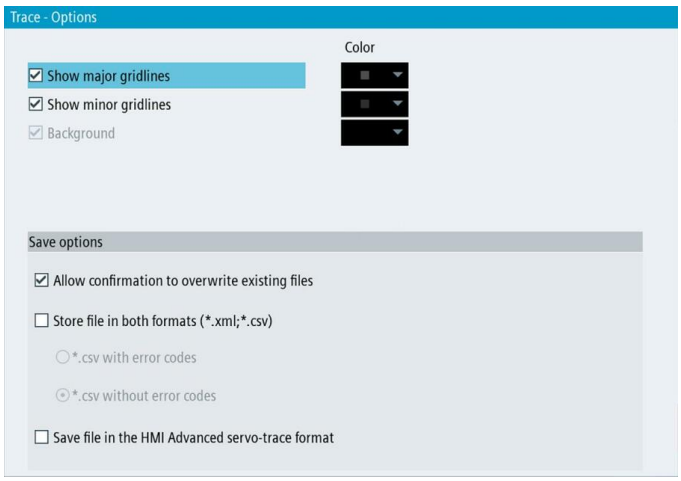
Notes:

Trace Options:

Certain options of the trace feature can be changed.



The default trace options can be seen below:



Trace options.

Show major gridlines:

- the gridline can be displayed or hidden.
- the gridline colour can be changed.

Show minor gridlines:

- the gridline can be displayed or hidden.
- the gridline colour can be changed.

Background:

- the background colour can be changed.

Save options

Allow confirmation to overwrite existing files:

- a confirmation prompt can be shown or hidden each time a trace session is to be overwritten.

Store file in both formats (*.xml; *.csv)

- save file in both formats (*.xml; *.csv) with or without error codes. Note: The file in csv format cannot be imported again.

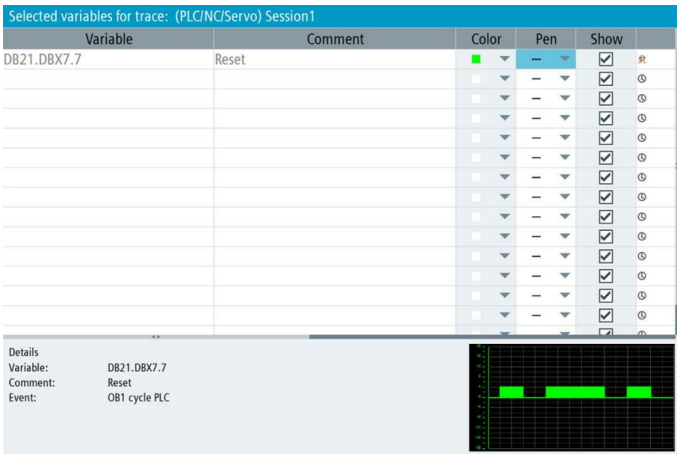
Save file in HMI Advanced Servo Trace Format

- store additional export data in the HMI Advanced servo-trace format.
  The following file formats are supported:
  SVT files (trace session settings)
  st1 files (variables of the 1st and 2nd trace sessions)
  st2 files (variables of the 3rd and 4th trace sessions)

Details:

The details soft-key can be used to view the detailed information of the selected variable.





Notes:

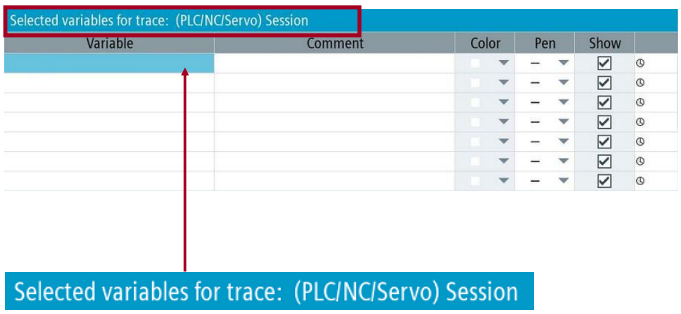
Drive Variable Trace Function

Note: Not for Virtual Machines

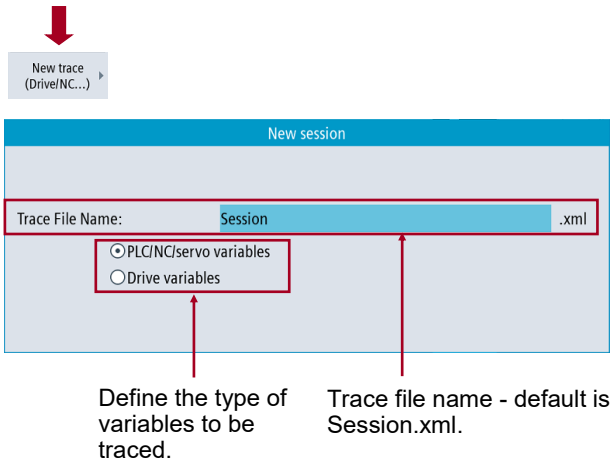
The default trace is for the tracing of NC/PLC/Servo variables. For the tracing of drive variables it is necessary to create a new trace session.



The default trace session is active in this example.



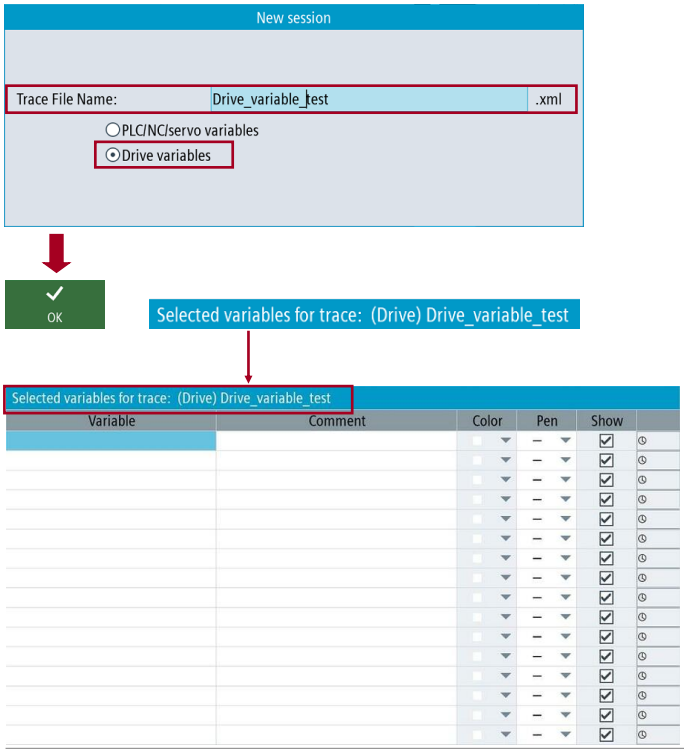
Select the ***New trace*** soft-key



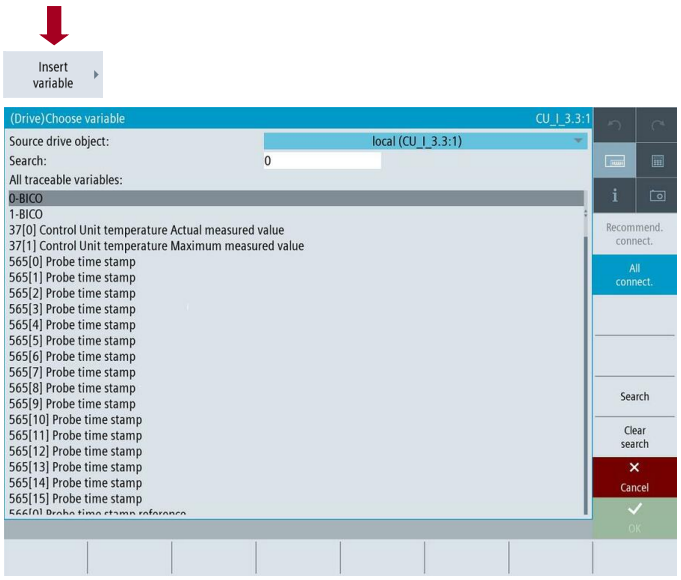
It is possible to create new NC/PLC/Servo and Drive trace sessions.

The file extension is ***.xml*** (Extensible Mark-up Language).

Example: a new drive variables session is to be created with the name of ***Drive_variable_test.xml***.

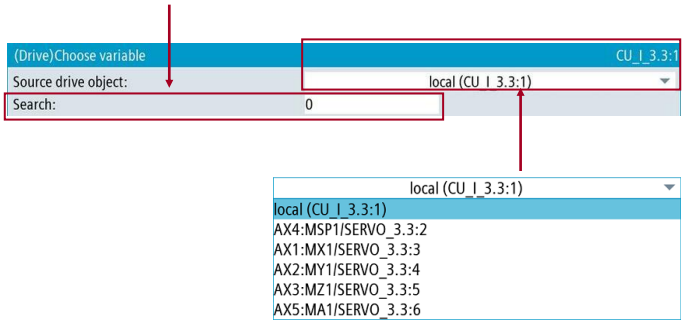


The variables to be traced can be entered manually or selected via the ***Insert variable*** soft-key:



Notes:

The required variable (parameter number or partial name) can be searched for:

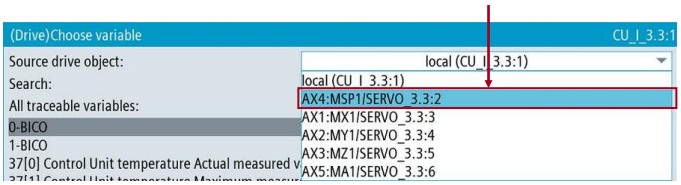


The default drive object is the CU. Other available drive objects can be selected from the drop down list.

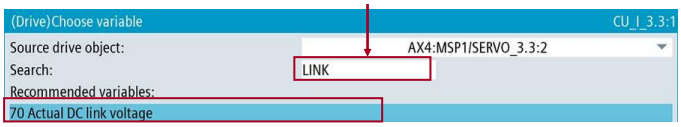
Example: in the following example the DC link voltage variable is being selected.

The DC link voltage drive variable is common to all the servos.

Choose which servo is to be used - the spindle, for example:



The search feature can be used to locate the required variable, example - LINK





Confirm with the ***OK*** soft-key.
The variable will now appear on the list and will be used in a running example later in the module.

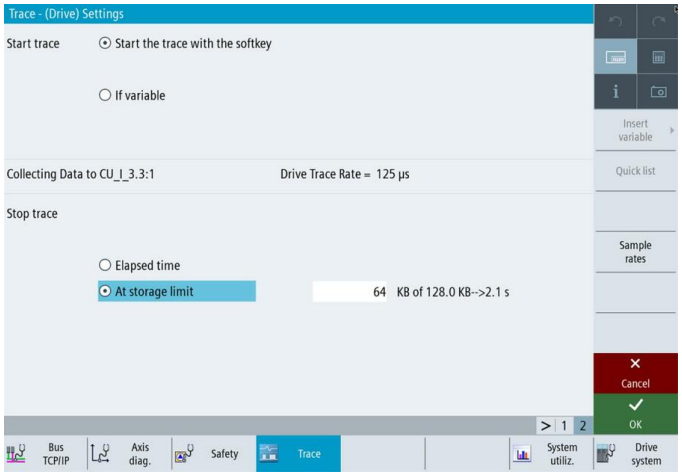


Trace Settings

The trace settings are used to set up the conditions for starting and stopping the trace.



The default selections of the Settings screen:



Note: some of the selections appear in red if a variable has not been previously selected:



Description of the Trace Settings.

Setting	Description/comments
Start Trace	Condition for starting the trace
Start the trace with the softkey.	The trace is started via the ***Start trace*** softkey
If variable	The trace is started by the condition of a variable. The variable is chosen by the user.
Prerun time	Trace trigger delay time (only with ***If variable*** setting)
Collecting Data	Trace data storage location.
to CU_I_3.3:1	Fixed location
Drive Trace Rate	Adjustable sampling rate for the trace
Stop Trace	Condition for stopping the trace
Elapsed time	The trace is stopped after the specified time.
At storage limit	The trace is stopped when the trace reaches the specified size - limit depends on whether the destination is CF card, Hard Disk or NCK

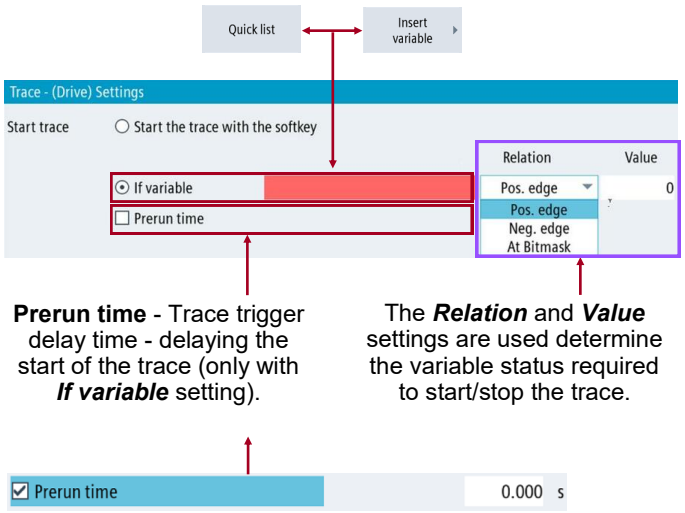
Notes:

Start trace - *If variable:* parameter description:

The variable for starting the trace can be manually entered or selected from a list.

Quick list - the Quick list soft-key will show the variables already selected.

Insert variable - the variable for starting the trace can be chosen from any of the drive objects (CU, NX or servo).

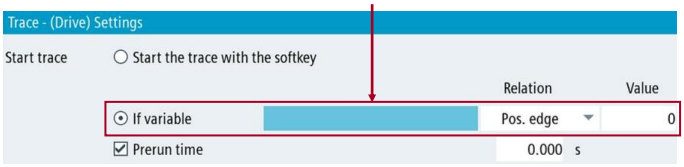


Prerun time - Trace trigger delay time - delaying the start of the trace (only with ***If variable*** setting).

The ***Relation*** and ***Value*** settings are used determine the variable status required to start/stop the trace.

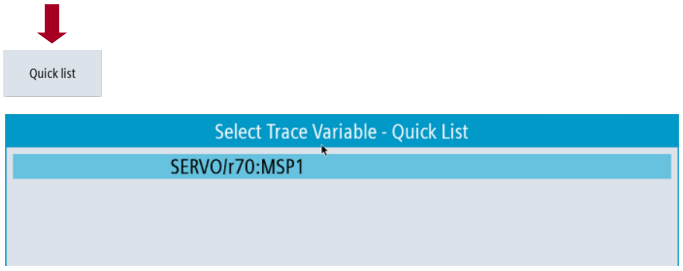
If variable **- Insert variable**

Highlight the entry field:

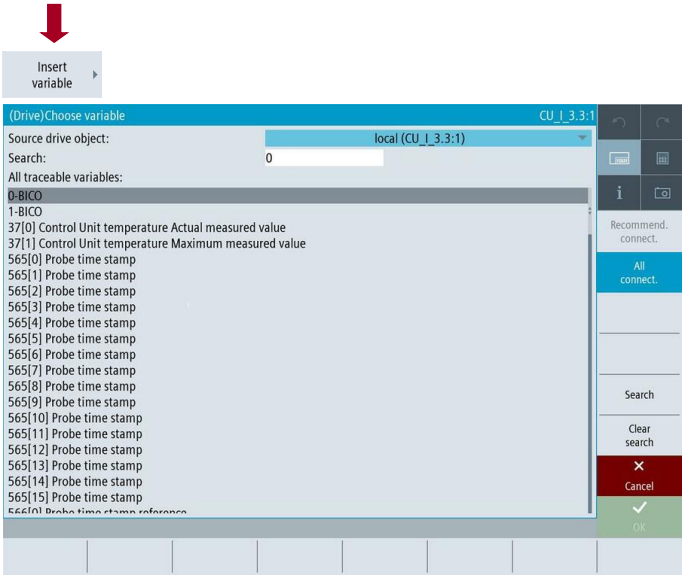


The variable can be either:
Manually entered.
Selected via the ***Quick list*** soft-key.
Selected via the ***Insert variable*** soft-key.

The ***Quick list*** softkey displays the variables which have been selected for tracing.



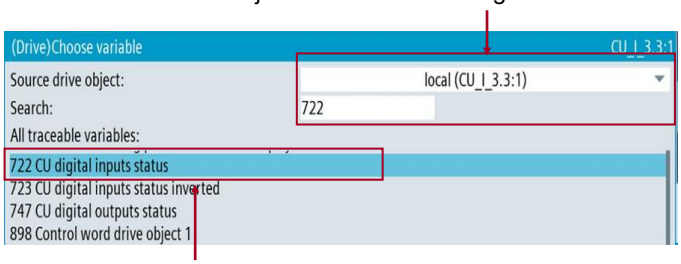
The ***Insert variable*** softkey allows a variable to be selected from any of the drive objects (CU, NX or servo).



Example

The following example will monitor the status of ***CU parameter r722*** which will be used to start the trace using the ***If variable*** setting.

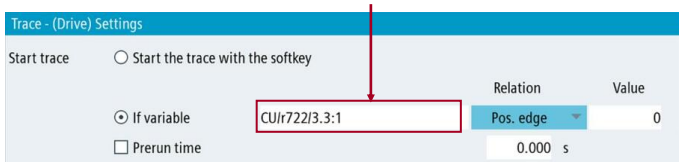
The CU is selected as the Source drive object and the search target is 722:



The required parameter, ***722 CU digital input status***, has been located.



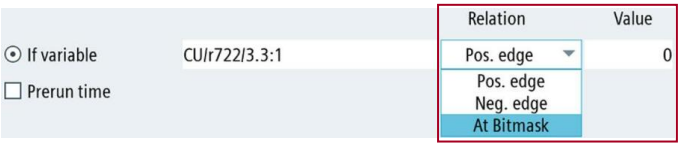
The selected variable appears in the ***If variable*** entry field:



Notes:

Relation/Value Parameters

The ***Relation*** setting is used to determine the required condition of the variable for starting the trace. This is in relation to the ***Value*** setting.

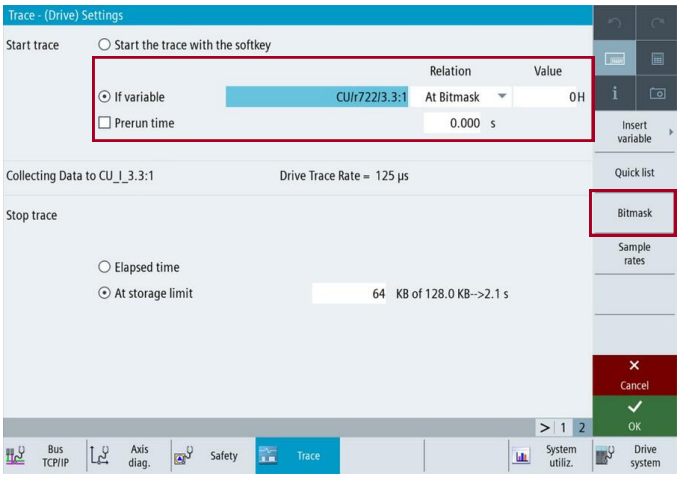


Value - the required value is selected to meet the trace start condition.
This is a decimal value unless the bitmask relation is used, in which case the value becomes a HEX value.
The Bitmask setting is described in detail in the following pages.

Trace Start Conditions	
Relation	***If variable* Status/description**
Positive edge	***Rising edge*** of the ***Value*** entry
Negative edge	***Falling edge*** of the ***Value*** entry
At bitmask	When using ***Hex*** parameter - see the following description.

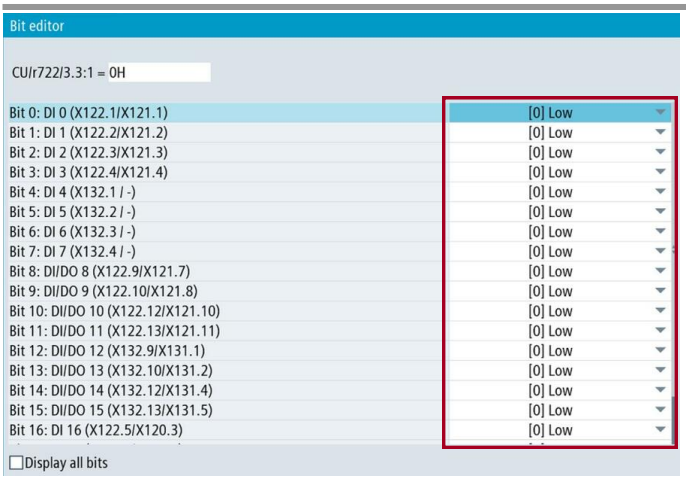
At Bitmask - Relation and Value Description

When the ***If variable*** setting is active and one of its entry fields is selected, the ***Bitmask*** vertical soft-key will be available.



The ***Bitmask*** soft-key is used for the selection of the bits of the variable whose status will be monitored and used to start the trace.



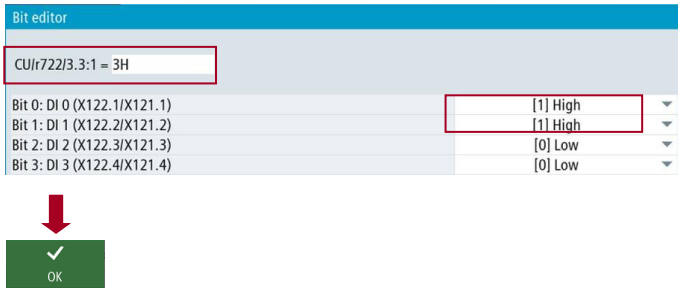


Description:

The bitmask default is for all the bits to have the ***Low*** selection set.
This setting will result in the trace starting regardless of the status of the selected variable.
For the variable to have an effect on the trace start, the bitmask should have at least one bit set to ***High***.
The trace start behaviour is then determined by the value entered in the ***Value*** parameter.
This parameter is used to set whether the selected bit is checked for a high or low status.

Bitmask & Value Example.

Bitmask:
The following example shows that bitmask bits 0 & 1 have been set to High.
This means that their statuses will be monitored when the trace start is selected.





The result of the selection is displayed in Hexadecimal format, above the variable.

Notes:

Value:
The default setting of the ***Value*** parameter is 0.
This means that the bitmask bits with the ***High*** setting are being monitored for a logic of 0.
For a bit to be monitored for logic 1, it has to be specified as such using the ***Value*** parameter.

Example: the requirement is for bit 0 to be monitored for logic 0 and for bit to be monitored for logic 1.
In this case the Value should be set to 2H - this represents bit 1 and sets it to be monitored for logic 1 when the trace start is active.

The following tables show some examples off Bitmask/Value relationship.

Table 1

Bitmask = F(H)		Value = F(H)	Result
Bit 0	High	Bit 0	Checked for logic 1
Bit 1	High	Bit 1	Checked for logic 1
Bit 2	High	Bit 2	Checked for logic 1
Bit 3	High	Bit 3	Checked for logic 1

Table 2

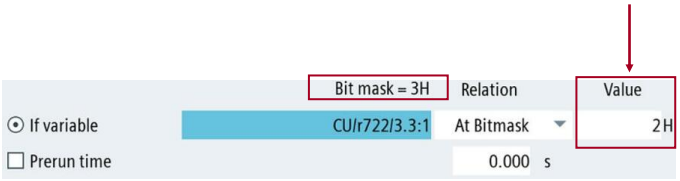
Bitmask = F(H)		Value = 10(H)	Result
Bit 0	High	-	Checked for logic 0
Bit 1	High	Bit 1	Checked for logic 1
Bit 2	High	-	Checked for logic 0
Bit 3	High	Bit 3	Checked for logic 1

Table 3

Bitmask = 3(H)		Value = 2(H)	Result
Bit 0	High	-	Checked for logic 0
Bit 1	High	Bit 1	Checked for logic 1
Bit 2	Low	-	Not checked
Bit 3	Low	-	Not checked

The settings from ***Table 3*** have been used in the following example.

Bit 1 is being monitored for logic 1 due to the value of 2H.



The trace will start if bit 0 = 0 and bit 1 = 1.
All other bits have no effect.

Note: this will be demonstrated using a running example later in the module.

Stop Trace:

Two options are available for setting the condition to stop the trace (note - a variable must be selected in order to be able to use the options).

Stop trace - *Elapsed time:* The trace will stop after the specified time has elapsed.



A choice of time units is available. The system calculates the resulting file size and displays it to the left.

If the entry is out of limits it will be displayed in red and cannot be accepted.

Stop trace - *At storage limit:* the trace will be stopped when the specified file size is reached (default 64KB, maximum 128KB).



If the entry is out of limits it will be displayed in red and cannot be accepted.

The system calculates the trace time based upon the selected file size and displays it to the right.

Sample rates



The Sample rate factor can be modified from the default setting (125µs x 1).
Increasing the sample rate will result in an increase in the recording time.



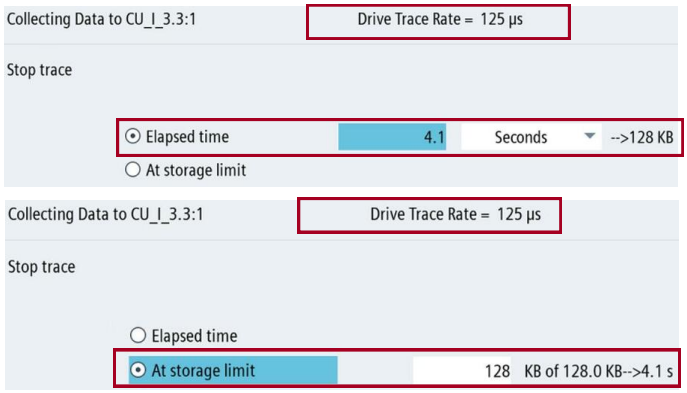
Notes:

Sample Rate Information:

The sample rate should be chosen to suit the trace requirements.
Smaller sample rates offer higher accuracy, for example, when tracing positional values.
The restriction of this is that the trace periods are limited in size and length.
The user has to decide which sample rate suits the trace requirements.

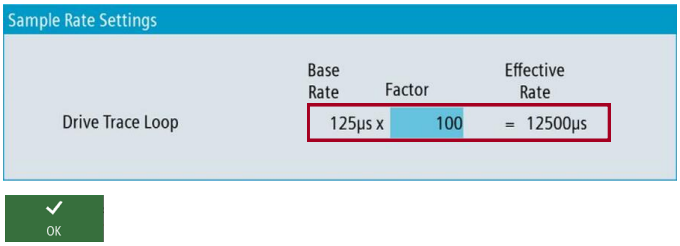
Default Sample Rate Limitations:

The default sample rate of 125µs restricts the trace time to 4.1 seconds with a trace file size of 128KB.

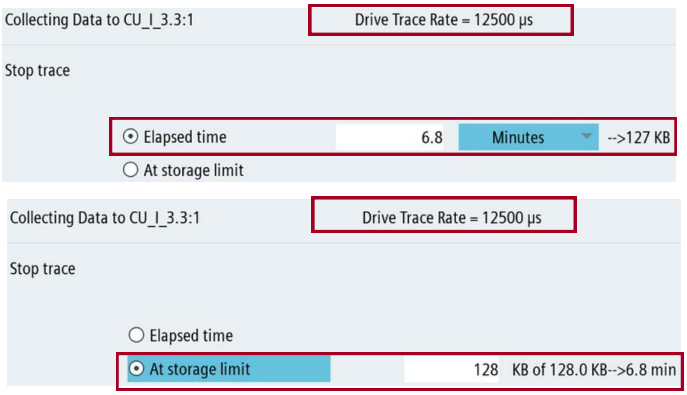


Increasing the sample rate:

Example - an in increase in the sample rate factor to 100 allows longer trace times:



The sample rate of 12500µs allows a maximum trace time of 6.8 minutes.



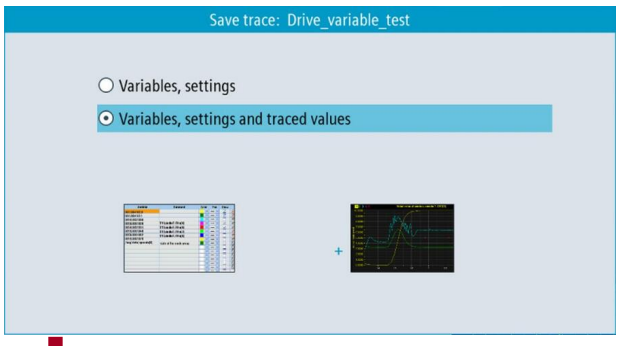
Saving and Loading Trace Files:

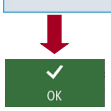
Save Trace.
Files containing the trace information can be saved locally on the controller or externally to USB memory devices or networked computers.

Note: save options are available via the ***Trace options*** softkey. These are described on the next page.

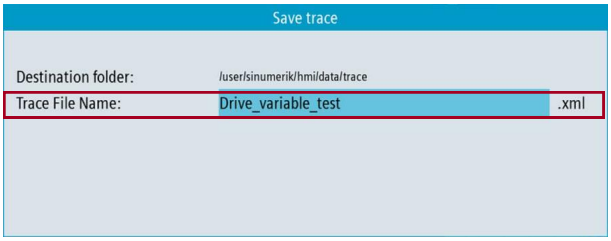


The trace variables and settings can be saved without traced values or with traced values (default):



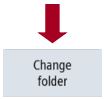


The trace file name can be entered or modified.

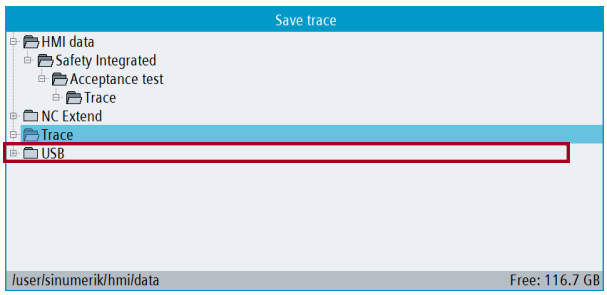




The file will be saved to the default location - *user\Sinumerik\hmi data\ trace.*

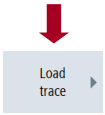


Selecting the ***Change folder*** soft-key allows the destination to be changed to a USB device, for example.

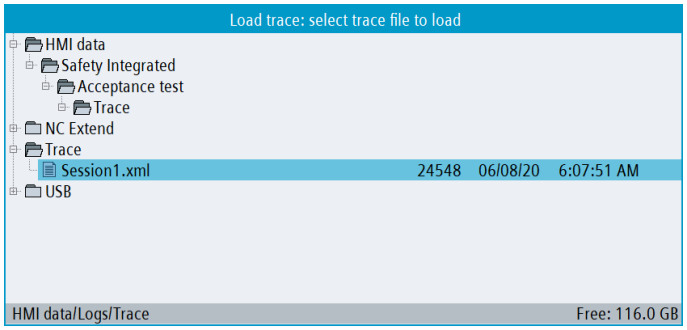


Notes:

Load Trace.
Files containing the trace information can be loaded from local locations of the controller or from external locations such as USB memory devices or networked computers.

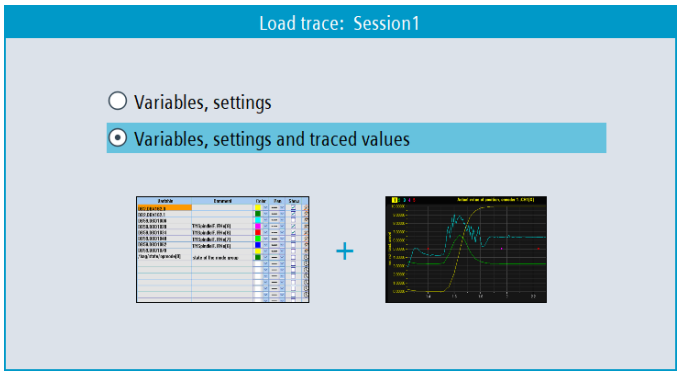


Locate the file to be loaded:





If the trace file contains traced values it can be decided whether to include them in the load operation:





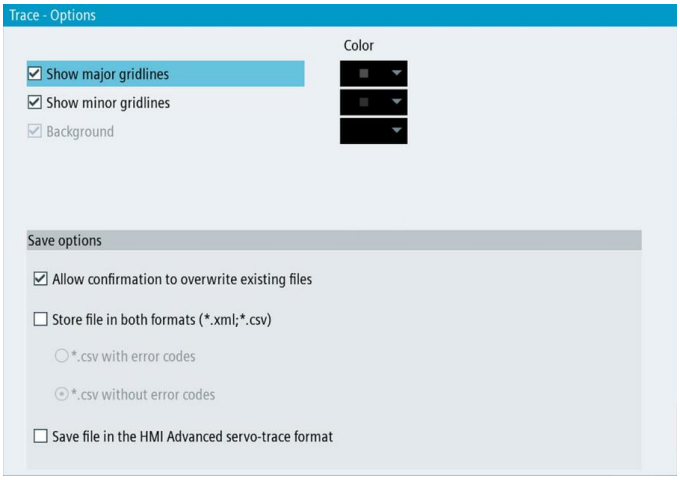
The trace information which was included in the trace file will now be available for viewing.

Trace Options:

Certain options of the trace feature can be changed.



The default trace options can be seen below:



Trace Options - Description.

Show major gridlines:
- the gridline can be displayed or hidden.
- the gridline colour can be changed.

Show minor gridlines:
- the gridline can be displayed or hidden.
- the gridline colour can be changed.

Background:
- the background colour can be changed.

Save options

Allow confirmation to overwrite existing files:
- a confirmation prompt can be shown or hidden each time a trace session is to be overwritten.

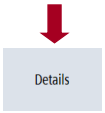
Store file in both formats (*.xml; *.csv)
- save file in both formats (*.xml; *.csv) with or without error codes. Note: The file in csv format cannot be imported again.

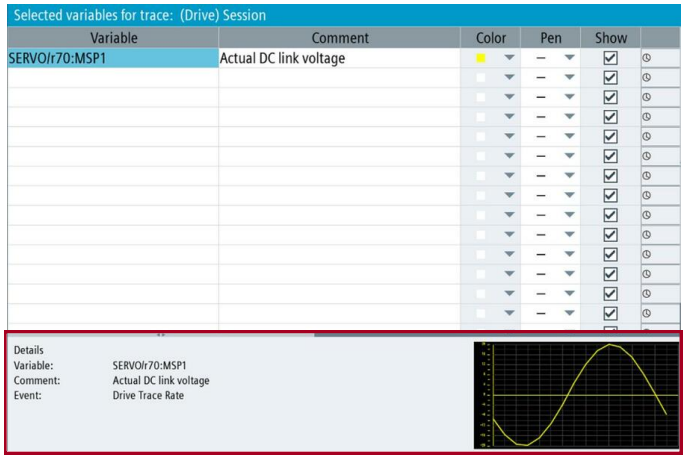
Save file in HMI Advanced Servo Trace Format
- store additional export data in the HMI Advanced servo-trace format.
  The following file formats are supported:
  SVT files (trace session settings)
  st1 files (variables of the 1st and 2nd trace sessions)
  st2 files (variables of the 3rd and 4th trace sessions)

Notes:

Details:

The details soft-key can be used to view the detailed information of the selected variable.





Example - Tracing a Drive Variable

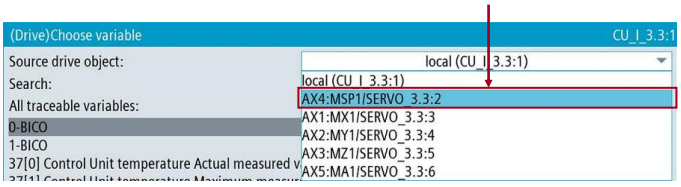
The following example will show the tracing of the DC link voltage drive variable (70 Actual DC link voltage).



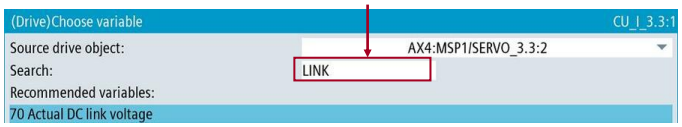
Example: tracing the DC link voltage.

The DC link voltage drive variable is common to all the servos.

Chose which servo is to be used - the spindle, for example:



The search feature can be used to locate the required variable, example - LINK



Confirm with the correct selection with the OK soft-key. The variable will now appear on the list.

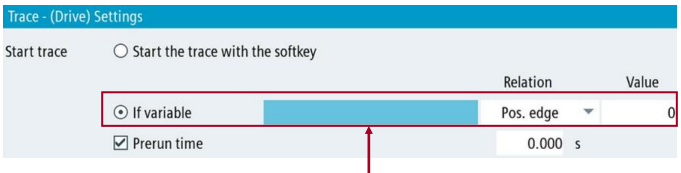




Trace settings.

The example will monitor the status of ***CU parameter r722*** which in turn will be used to start the trace using the ***If variable*** setting.



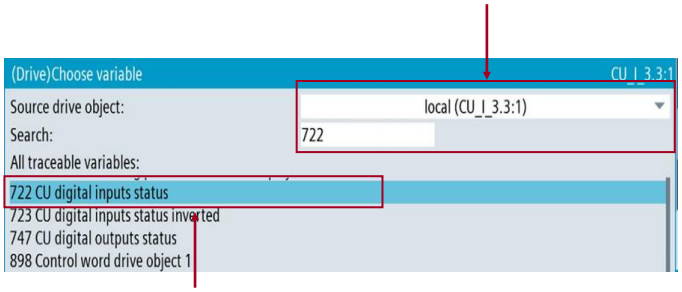


Highlight the entry field shown above



Notes:

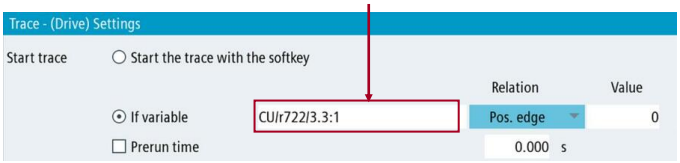
The CU is selected as the Source drive object and the search target is 722:



The required parameter, ***722 CU digital input status***, has been located.



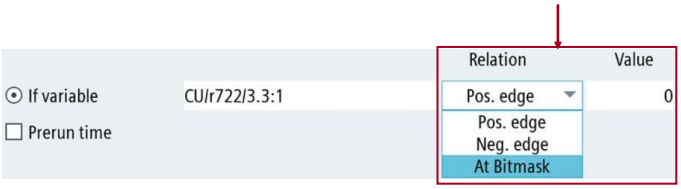
The selected variable appears in the ***If variable*** entry field:



Selecting the trace start conditions.

The example requires that the trace is started when r722 bit 0 = 0 and bit 1 = 1.

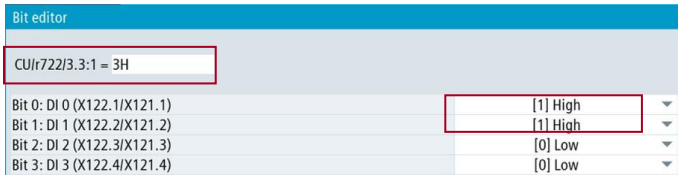
Select the ***At Bitmask*** relation:



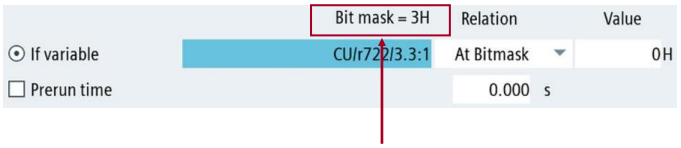
Select the ***Bitmask*** soft-key:



- Set bits 0 & 1 to **High:**



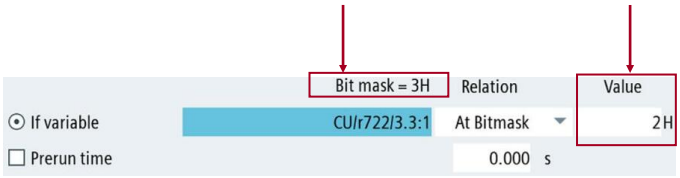




The result of the selection is displayed above the variable in Hexadecimal format.

Bitmask = 3H:
Bit 0 & 1 are being monitored

Value = 2H:
Bit 1 is being monitored for logic.



These settings determine that the two signals are to be monitored for the correct status in order to start the trace.
The correct status is declared using the ***Value*** setting.
By default, the value of 0 requires that the two bits have a logic = 0 to start the trace.
The example requires that bit 0 = logic 0 and bit 1 = logic 1.
To achieve this the ***Value*** has to be set to 2 (monitor bit 1 for logic = 1)
See the following table:

The trace will start if bit 0 = 0 and bit 1 = 1.
All other bits have no effect.

Requirements for the training example.

Bitmask = 3(H)		Value = 2(H)	Result
Bit 0	High	-	Checked for logic 0
Bit 1	High	Bit 1	Checked for logic 1
Bit 2	Low	-	Not checked
Bit 3	Low	-	Not checked

Sample rates



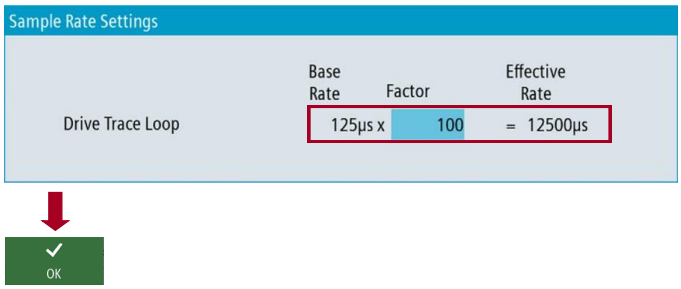
The Sample rate factor can be modified from the default setting (125µs x 1).
Increasing the sample rate will result in an increase in the recording time.



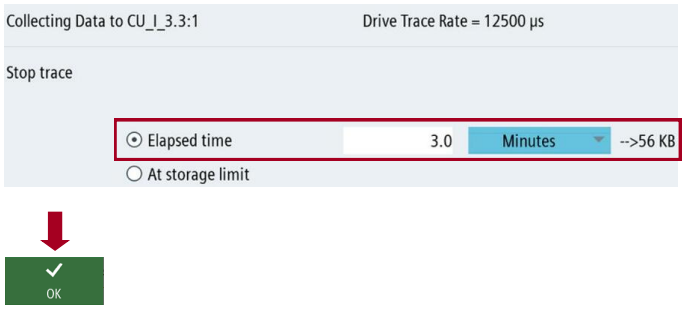


Notes:

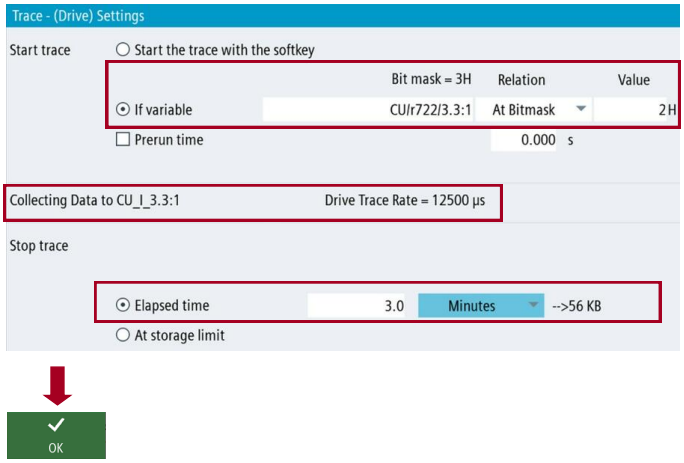
The example will require a trace time of 3 minutes. This requires the sample rate to be increased.
An increase in the sample rate to 100 will increase the trace capacity to allow a 3 minute trace time to be set.



The required trace time of 3 minutes can now be set:



The required trace settings are now complete:



Summary:

- Drive variable, ***SERVO/r70:MSP1***, has been selected for tracing.
- The trace start will be initiated when ***CU/r722/3.3.1***, bit 0 = 0 and bit 1 = 1.
- The trace will record for 3 minutes

Additional information:

The variable, ***CU/r722/3.3.1***, chosen to start the trace is a CU parameter and displays the status of Sinumerik ONE interface X122.

The connections of the training equipment are shown below:

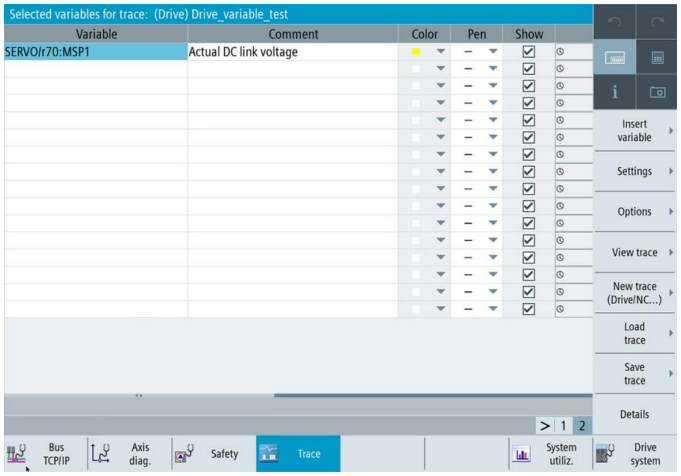
X122 terminal	CU r722 bit	Training equipment connections
1	0	OFF1 Drive on/off/Emergency Stop
2	1	OFF3 I80.0 on main console
3	2	
4	3	
5	16	
6	17	
7	n/a	
8	n/a	
9	8	
10	9	
11	n/a	
12	10	Spindle BERO
13	11	
14	n/a	

Pre-conditions:

To test the trace example it is necessary to:

- Release the Emergency Stop buttons - MCP & drive simulator.
- Acknowledge the Emergency Stop - ***Act-E-Stop*** button on the drive simulator.
- Switch on the drives - ***Drive ON*** button on the drive simulator.

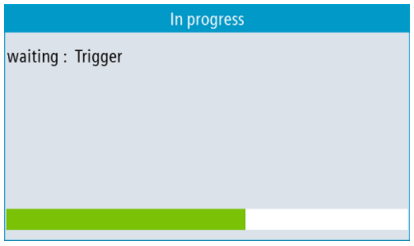
Starting the trace sequence:



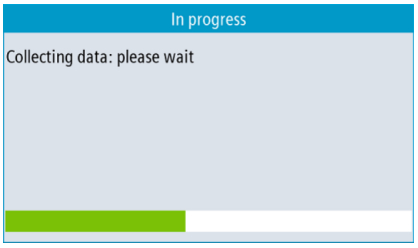
Notes:



The system is now waiting for the correct trace trigger conditions:

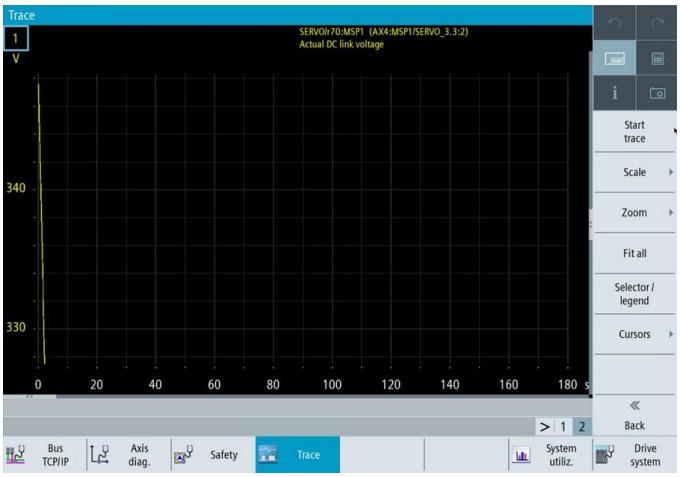


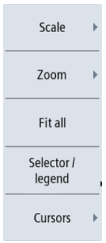
Press the Emergency Stop button. After a short delay the trace will start. This is due to the drive signal to CU r722.0 (X122.1) being set to 0.



The trace settings determined that the trace would continue for 3 minutes.

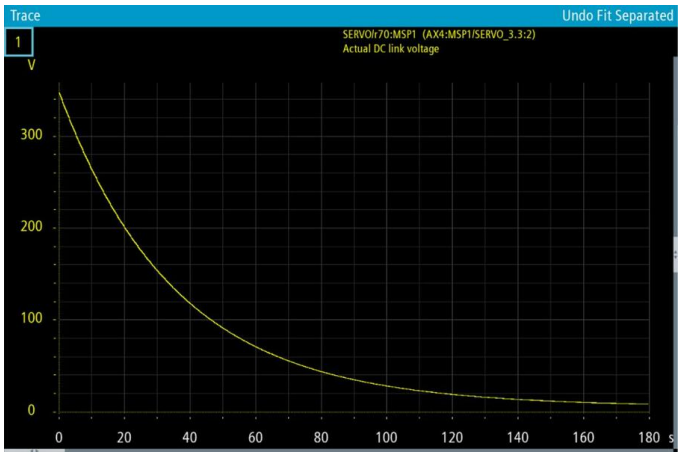
After 3 minutes the result is displayed





Various tools are available for analysing the trace.





The results show the decay of the DC link voltage over the 3 minute period.

A017: END

Notes:

019

SINUMERIK

Electrical Cabinet Design

Module Description:

Technological advances have resulted in the reduction in size of electronic equipment. The result being that the equipment is more prone to heat, dust and moisture damage.

By calculating the power loss of the components and therefore ensuring the correct cooling requirements of a particular cabinet, it is possible to reduce the failure rate of these and other components.

It is also essential that the cabinets are designed to prevent/minimise the ingress of contaminants.

Note: the installation and wiring of electrical enclosures is covered by various international standards and directives. These should be consulted and adhered to along with any equipment manufacturer's recommendations.

Module Objective:

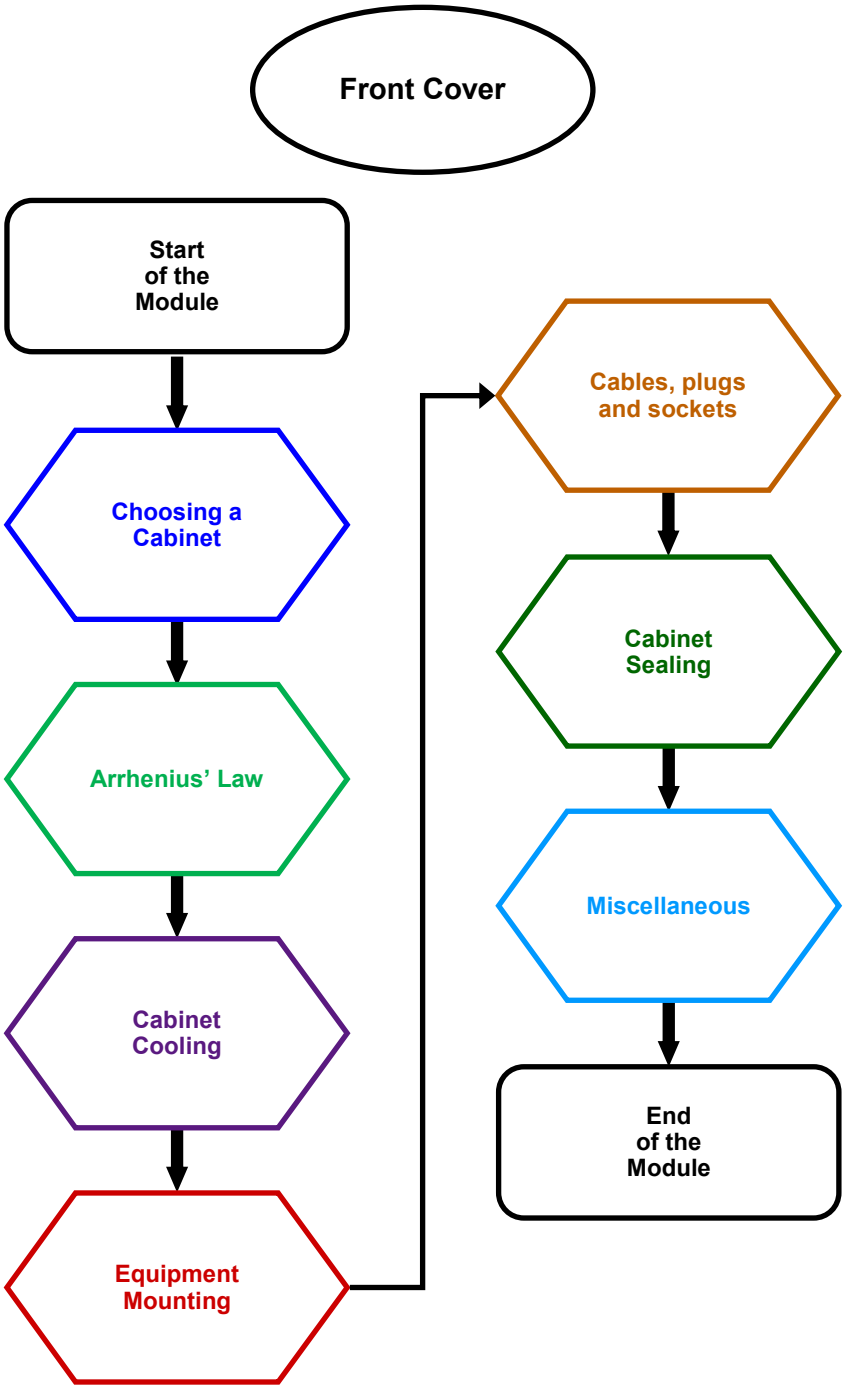
Upon completion of this module you will know the design criteria of electrical cabinets.

Content:

Sinumerik

- Arrhenius' Law
- Choice of Cabinet
- Cabinet Cooling
- Equipment Mounting
- Cables, plugs and sockets
- Cabinet Sealing
- Miscellaneous

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Choosing a Cabinet
Arrhenius' Law
Cabinet Cooling
Equipment Mounting
Cables, plugs and sockets
Cabinet Sealing
Miscellaneous
End of the Module

Choosing a Cabinet

Sinamics drives are available in internally cooled formats.

Heat generated by the drive modules and any additional cabinet equipment will need to be dissipated in order to keep the cabinet temperature within the recommended limits.

This will usually require additional cooling to help regulate the temperature.

Depending on the application, a simple fan/filter setup might be sufficient.

It may be preferable to always use cooling units as this allows the cabinet to be sealed and is the best option for stopping the entry of contaminants.

The technical specifications of the individual hardware component will provide details of the power loss (measured in Watts/ kilowatts).

Calculations are made to see if the cabinet can dissipate the heat naturally.

The cabinet manufacturer can supply details of the heat-loss of the cabinet.

Natural heat dissipation occurs through the cabinet walls, providing that the wall is in free air and not against another structure, for example.

A general rule is that the energy dissipated is approximately 50Watts/m² of free cabinet wall assuming a 10° temperature difference between the cabinet and the outside air.

Example: A cabinet with a width of 3m, a height of 2m and a depth of 1m sited with its back to a wall has an effective surface area of approximately 13 m².

2m x 3m = 6m² (Front)
2m x 1m = 2m² (left side)
2m x 1m = 2m² (Right side)
3m x 1m = 3m² (Top)
Total = 13m²

Therefore, this cabinet is able to dissipate approximately 650 W. (50W x 13m²)
If the actual losses of the cabinet equipment are higher than this, other cooling measures have to be implemented.

Power loss examples:
36Kw Active Line Module = 406W
30A Single Motor Module = 260W typical (max. 310W)
18A Double Motor Module = 280W typical (max. 350W)
9A Single Motor Module = 60W typical (max. 100W)

These items total 1006W which would require extra cooling if fitted into the cabinet described above, especially when any additional cabinet equipment is taken into consideration.

Cabinet Ambient Temperature

The ideal ambient temperature for the cabinet is **35°C**.

The cooling unit should be set to achieve this temperature.

If the temperature is allowed to get too high the lifetime of the components will be reduced.

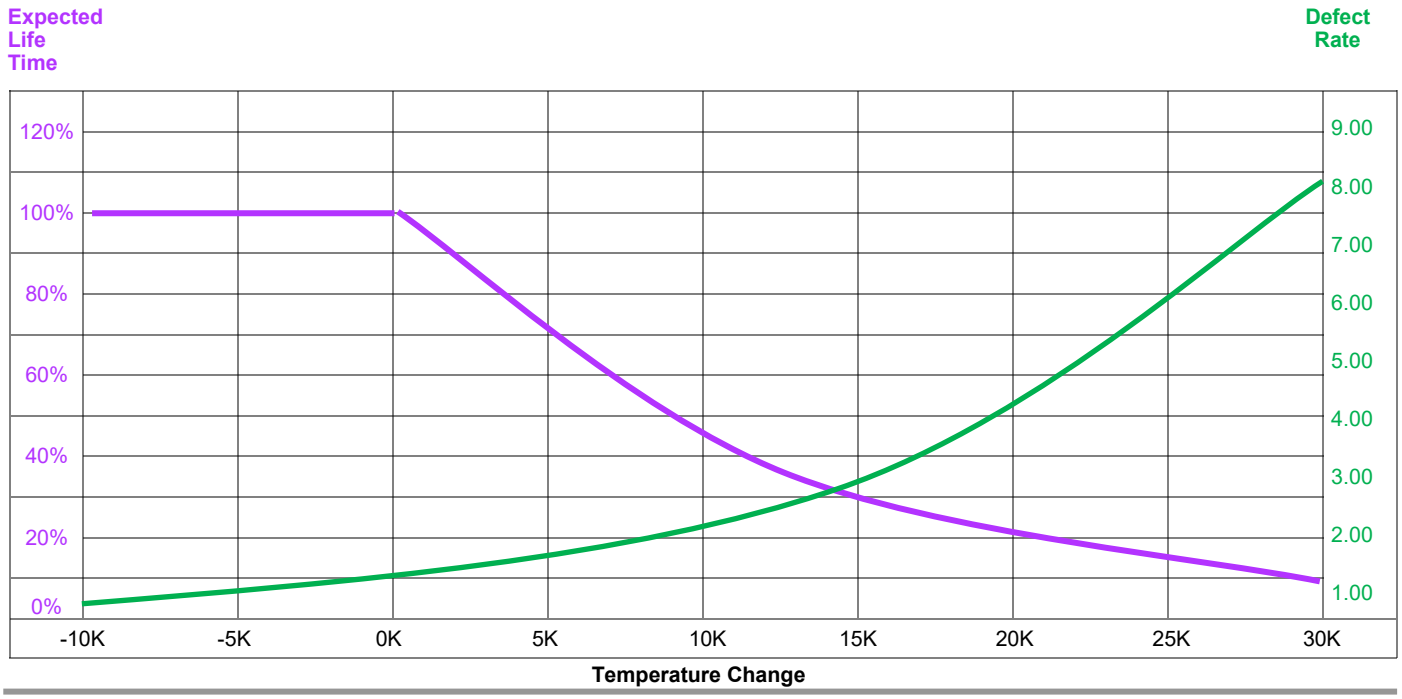
If the temperature is set too low there is a risk of condensation forming on the electronic components which can also be detrimental to the equipment.

Within the cabinets it may also be necessary to install fans to ensure the air is circulated and prevent ***hot spots*** from forming.

Notes:

Arrhenius' Law

The following graph demonstrates Arrhenius Law and shows the effects that increased temperature has on the lifetime of electronic components.



Notes:

Cabinet Cooling

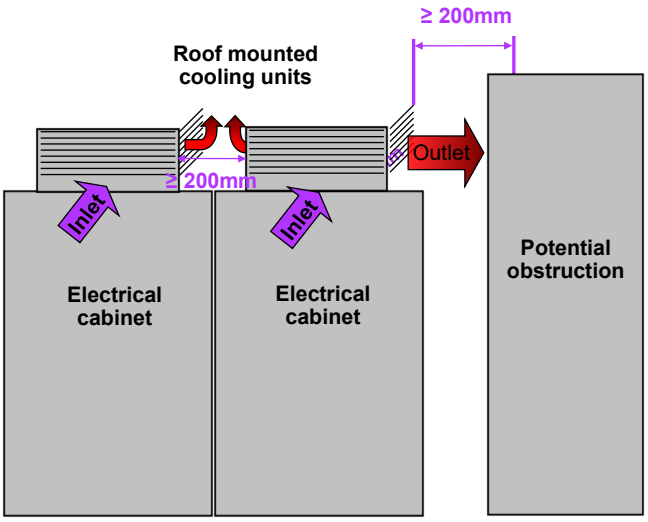
Cabinet cooling is usually accomplished using fans and filters or cooling units.

The fans and filter option would normally be used on low power installations or where the drive units are of the externally cooled type.
A drawback of this method is that the fans pull air from outside the cabinet via filters. Even if the filters are serviced regularly it is impossible to keep all contaminants out of the cabinet.

Cooling units allow the cabinets to be air-tight as the air within the cabinet is re-circulated through the cooling unit.
A thermostat should be employed to regulate the temperature to the recommended 35°C.
If a higher temperature is unavoidable (max 50°C) the drive units have to be de-regulated.

It is important to position the cabinet and cooling devices to ensure correct operation.
A cabinet placed in such a position where it restricts the ventilation inlet/outlet can prevent the cabinet from being cooled correctly or result in the cooling unit becoming damaged due to excessive work cycles.
The cooling device should also be positioned to ensure the cool air is reaching the desired areas and that the warm air is extracted and not allowed to cause hot spots within the cabinet.

The inlets and outlets of the cooling units must have a clearance of at least 200mm.
Example:



The use of baffles is recommended to prevent ventilation short circuits where the exhaust air is taken directly into the cooling units.

Cabinets with cooling units.

Correct installation of cooling units.

The cool air should be directed to the bottom of the drive. The internal drive fans and natural convection will draw the air through the drives.

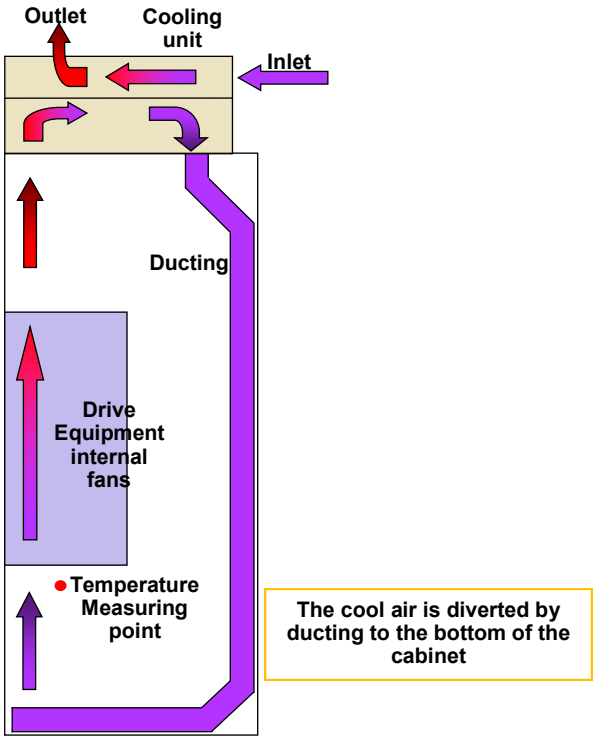
The cool air from the cooling unit should not be directed straight onto/into the drives, it should be allowed to mix with the warmer air already in the cabinet, This will minimise the risk of condensation forming.

Cabinet top mounted cooling units require a method of directing the cool air from the cooling unit to the bottom of the drives.

In the case of a door mounted unit it may be necessary to direct the cool air with the use of baffles.

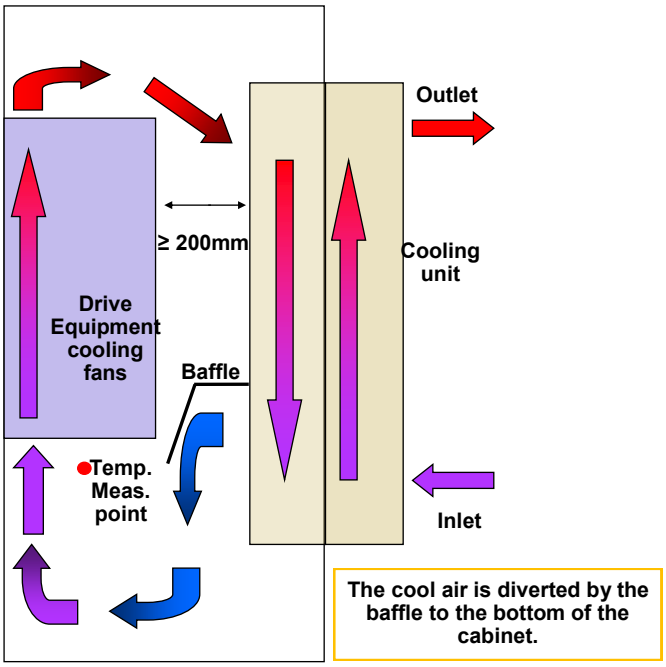
Example of cabinet cooling with roof mounted cooling units:

Cabinet top cooling unit.



Notes:

Cabinet door cooling unit:



It is advisable that the cooling units are switched off automatically when the cabinet door is opened.

Large differences between the outside air temperature and the cabinet temperature can lead to condensation forming when the cabinet doors are opened. This should be taken into consideration when the cabinet thermostat is set.

A by-product of cooling units is water. For obvious reasons it must be ensured that this water is not allowed to enter the electrical cabinet . This is particularly important for roof mounted cooling units as any leaking water could drip directly onto the drive equipment.

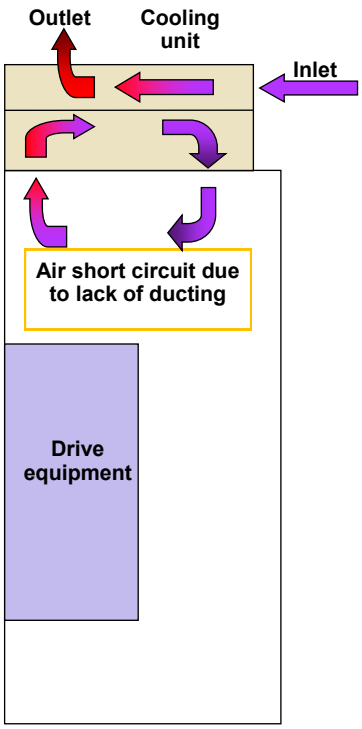
The outside of the cooling unit must be protected from any possible contamination such as coolant, machine waste etc.

Cabinet cooling sins.

The following diagrams are examples of cabinet cooling sins:

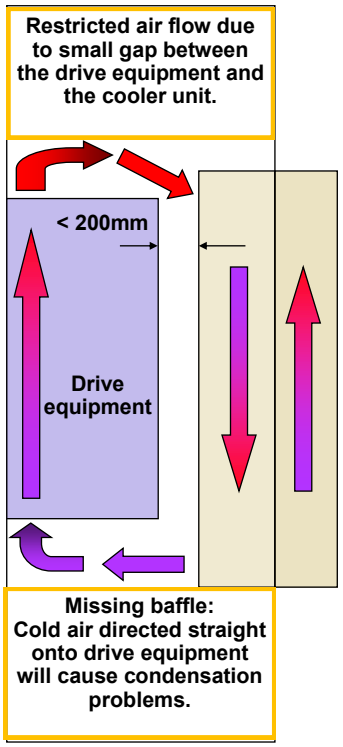
Top mounted cabinet cooling unit - without ducting.

The cool air does not reach the components.

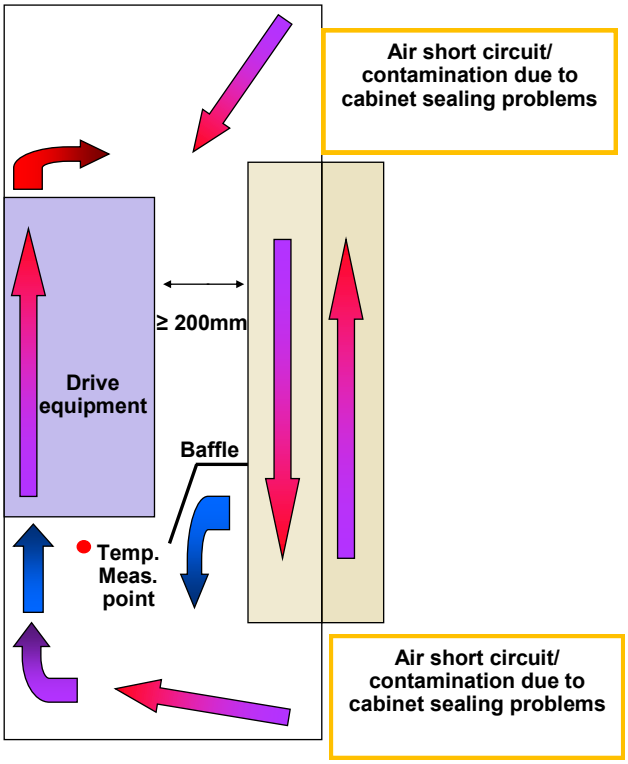


Notes:

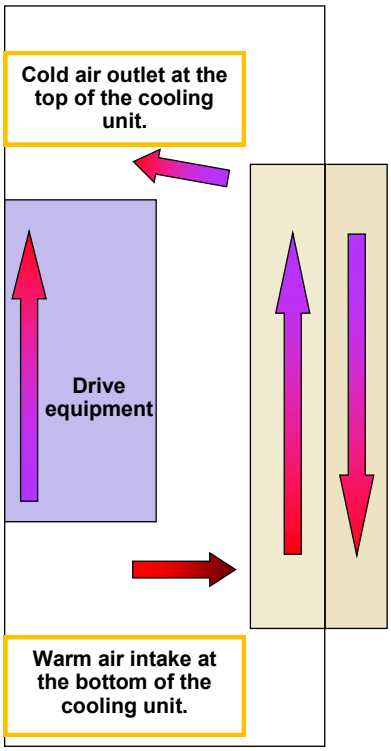
Air intake restricted due to narrow gap between cooling unit and drive equipment/missing baffle.



Air short circuit/contamination due to cabinet sealing problems:



Wrong type of cooling unit used.

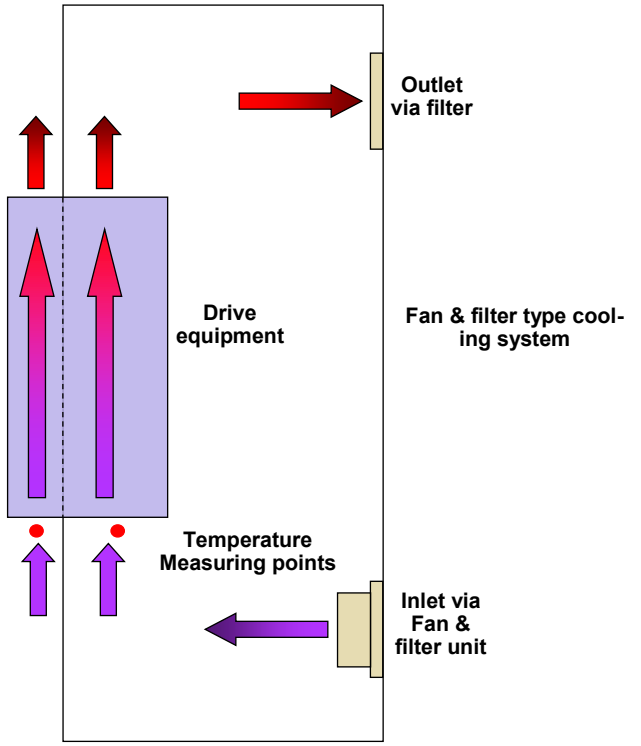


Notes:

Cabinets Without Cooling Units

For cabinets with externally cooled drive units it is possible that the cooling can be achieved with fans drawing air from the outside via a filter.
Natural convection causes the warm air to exit via a filter.
The fact that air is being drawn directly from the outside means that this system can never be airtight.
Having a fan for the inlet side ensures that the cabinet is positively pressurised which helps keep contaminants from entering.
It is extremely important that the filters are serviced regularly to avoid ingress of contaminants and to maintain the efficiency of the cooling process.

Example: Cabinet with externally cooled drive units.



Notes:

Equipment Mounting

The correct mounting of equipment is important with respect to the cooling process and for Electromagnetic compatibility measures.

Air flow to drive units:

It is essential that the air flow to the drive units is unrestricted. Other electrical equipment, including cable ducting and cables, must be mounted a suitable distance away from the top and bottom sides of the drive units. The minimum clearance distance for Sinamics S120 drives is 80mm.

Examples: Correct mounting methods for drive equipment.

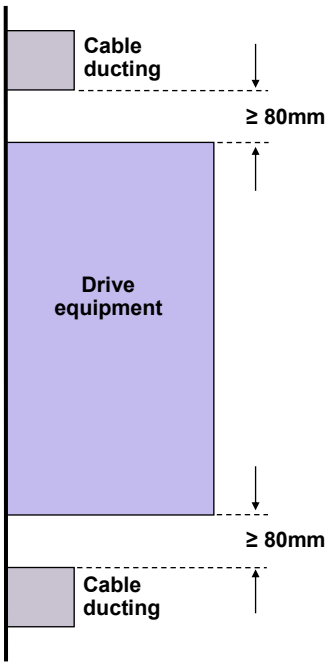


Diagram showing minimum distance between drive units and other cabinet components.



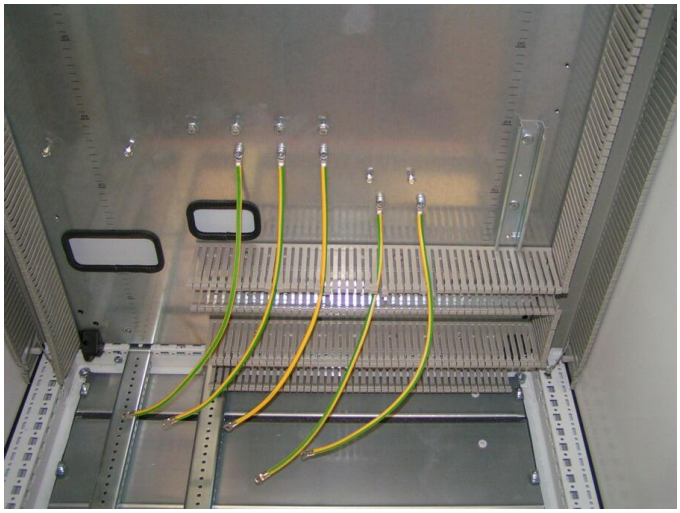
Photograph of correctly installed drive units. Cables and ducting are not causing restrictions to the air flow.

Notes:

Electromagnetic Compatibility (EMC)

To cut down on the interference effects caused by the high frequency drive equipment certain measures can be taken. One of these includes the positioning of the equipment within the cabinet.
By positioning the line module, reactor and filter in close proximity to each other, cables lengths can be kept to a minimum thus cutting down on emissions.
Further reductions in emissions can be achieved by screening the cables.
The mounting plate for the equipment should be one of bare metal, this acts as a central earth point and also improves the EMC.

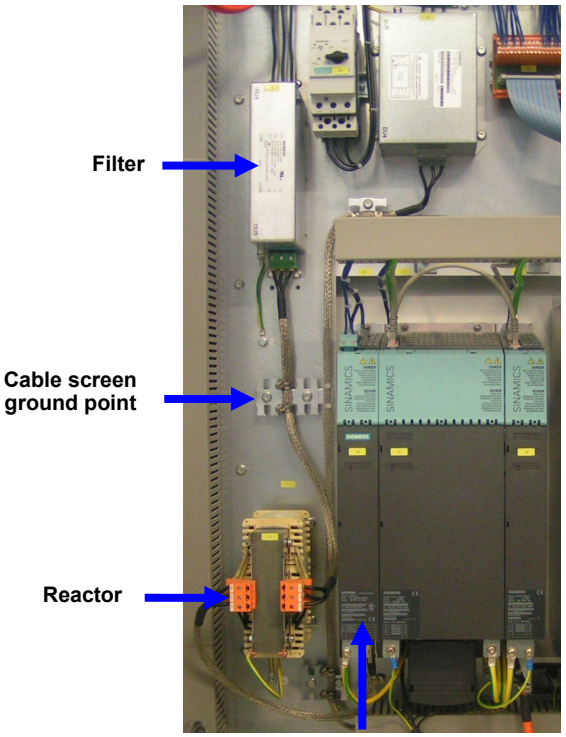
Example: A bare metal backplane ready for the drive modules to be mounted, note the earth bonding cables, one for each drive module.



Example: How shielded cable can be grounded.



Example of a filter, reactor and line module installation.



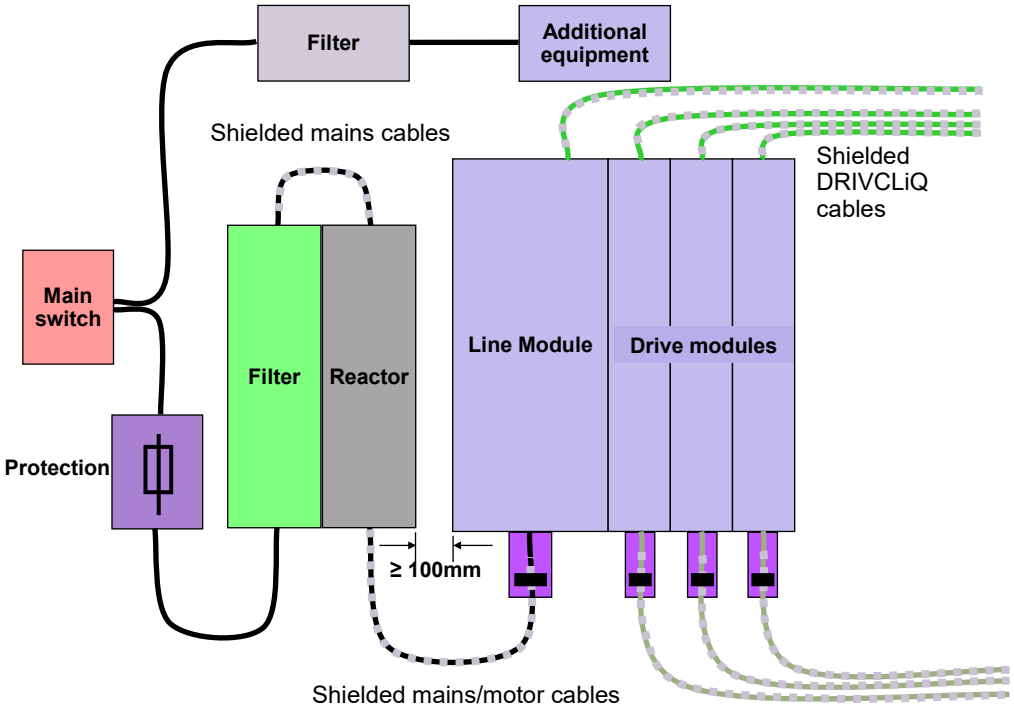
Notes:

A typical cabinet layout demonstrating the close proximity of the filter, reactor and line module in order to reduce the interference effects of the high frequency drive current.

The use of shielded cables is also necessary to further eliminate this Interference.
Shielding plates offer the best method for grounding the shields.

The power cables and the signal cables should also be run separately from each other throughout the machine and not just in the electrical cabinet.

It is not permissible to connect any other equipment between the line module and the filter and reactor.



Notes:

Cables, Plugs and Sockets

Shielding

Shielded power cables are recommended between the Sinamics line filter, reactor and line module. If the cable length is 1m or less the cables can be twisted and not shielded.

Signal cables and power cables should be kept separate by a distance of at least 0.2m.

Cables should be kept as close as possible to objects with a ground potential, this reduces crosstalk between cables.

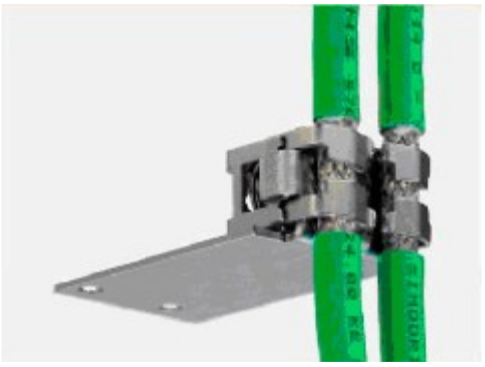
Where the cable shield is grounded it should be done with the use of grounding plates, this allows a large surface area of the shield to be connected. The use of a wire to connect the shield to ground is not permitted as the wire then becomes an aerial which can introduce problems.

Failure to observe all of the above can lead to intermittent faults.

The shielding of cables has a downside in that it causes additional losses in the line filter, reactor, in-feed and motor modules. This becomes worse the longer the cable length and can cause overheating of the drive components.

A maximum of 50m is allowed for each motor module and a maximum of 150m for each drive system.

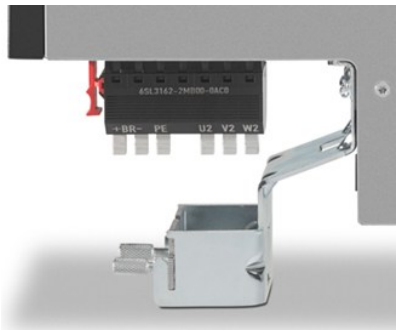
Example: Correctly shielded cables



Shielding that is connected via a plug & socket housing must use a metallic housing with a continuous metallic contact. Metalized housings are not permitted, over a period of time the metalized coating becomes worn and fails to be conductive to the levels required.

Shielding of motor power cables

Below: The Sinamics S120 drives are equipped with a shielding frame which should always be used.





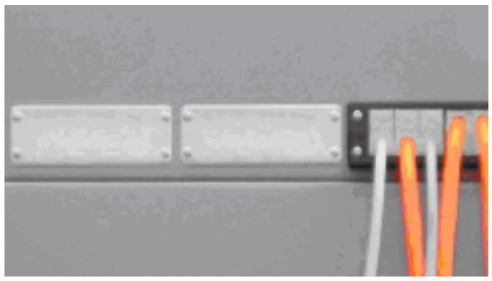
Example - an Incorrectly shielded motor cable:



Examples of cabinet cable entries:

To maintain the airtight properties of the cabinet, cable outlets should use methods which form a seal and any unused openings blanked off.

Without plugs and sockets:



With plugs and sockets:



Notes:

Cabling Sins.

Encoder and power cables in close proximity:



Encoder and motor cables in close proximity.



Cabinet Sealing

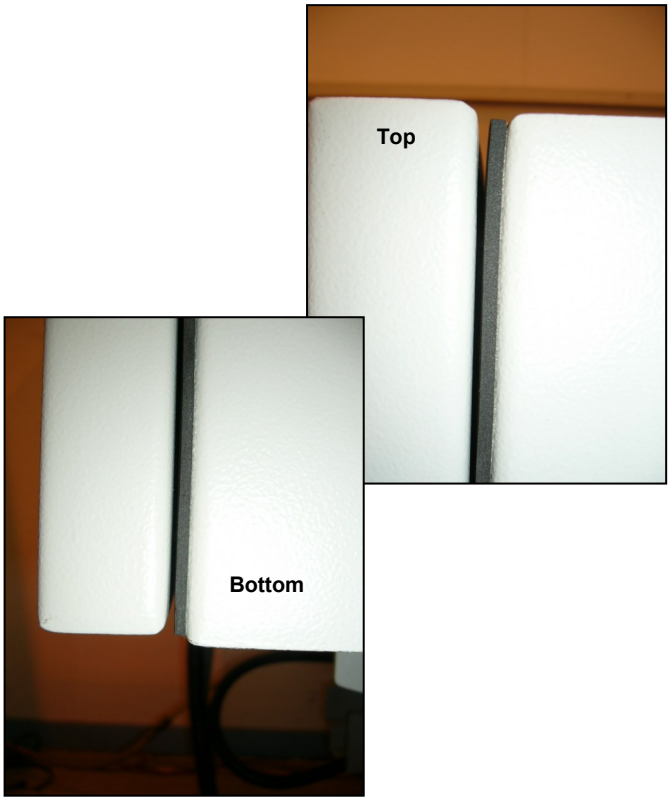
A cabinet that isn't sealed correctly can lead to a multiple of problems. Poor Sealing can be caused by cable entries, poorly fitting doors, etc.

A poorly sealed cabinet can:

- Influence the air circulation and cause overheating due to hotspots forming and air short circuits.
- Lead to a high moisture content which can in turn lead to component failure.
- Emit electrical interference, causing problems with other nearby equipment.
- Allow harmful, conductive contaminants to enter causing component failure.

The cabinet door should always be closed and locked during normal running operation, access should only be with a key or tool and by authorised personnel.

Right:-An example of a poorly fitting cabinet door.



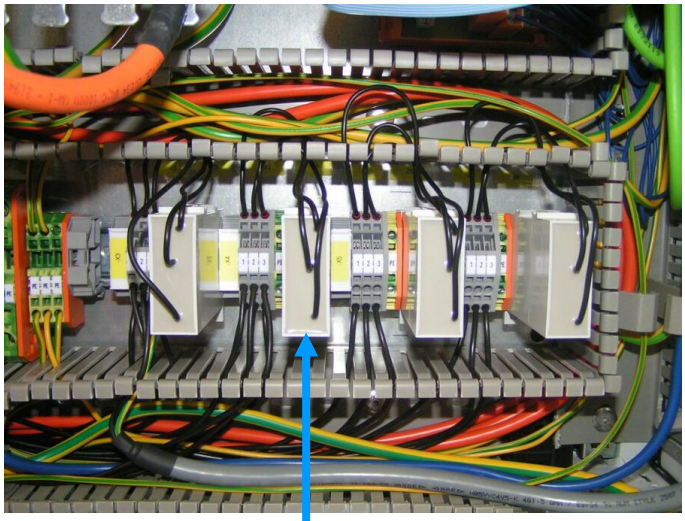
Notes:

Miscellaneous

Coil Suppression

Poor Electromagnetic compatibility can also affect other cabinet mounted components.
To deal with this suppressing devices can be used on any item with a coil. (Except servo motors)
AC components use RC networks for suppression while DC components use diodes.

Below:-Example of 3 phase AC suppressors.



AC Suppressor

019: END

Notes:

Notes:

SIEMENS

MACHINE TEST PROTOCOL FOR SINUMERIK

Product / Version	Comment
SINUMERIK ONE **SINUMERIK 840D sl** **SINUMERIK 828D**	Virtual and Real

Machine Description	**Machine Features**

Electrical Cabinet Protocol Attached	☐	Safety Integrated Protocol Attached	☐

Title **VMC 600**	DEX Machine Tool Incorporated
Document no.	Date
	Protocol version 6.1

Copyright © Siemens AG 2020 All Rights Reserved.

Contents

1 Responsible persons: 3
1.1 Schedule 3
1.2 First commissioning contacts 3
2 Machine Test summary 4
2.1 Summary 4
2.2 Result 4
2.3 Release of the machine recommended (YES/NO/Reason): 4
2.3.1 Exception List 4
2.3.2 The machine has been released for sale with the following exceptions: 4
2.4 Changes to standard data – as reference data 4
2.5 Data Backup 4
3 Change Log 5
3.1 Changes to the machine. 5
Components contained in the machine configuration. 5
3.2 Control 5
4 Test objective 6
4.1 Note 6
5 Functions tested 7
Detailed Test 8
5.1 Machine Area 8
5.1.1 Jog → Ref 8
5.1.2 Jog → MCP 9
5.1.3 Jog → HMI Soft keys 11
5.1.4 MDA → Teach-in 11
5.1.5 MDA → HMI Softkeys 12
5.1.6 MDA → MDA → Teach-in 13
5.1.7 AUTO 14
5.2 Technology 15
5.2.1 Milling General 15
5.2.2 Milling 3+2 15
5.2.3 Milling 5 Axis 16
5.2.4 Turning Basic 16
5.2.5 Turning Advanced 17
5.3 Parameter 18
5.3.1 Machine Builder Options 18
5.3.2 Measure tool 18
5.3.3 Setting data 18
5.3.4 Magazine 18
5.3.5 Tool List 19
5.4 Geometry 19
5.4.1 Circularity test HMI 19
5.4.2 Ball-bar circularity test 19
5.5 Safety Integrated 19

Copyright © Siemens AG 2020 All Rights Reserved.

5.5.1 Safety related test 19
5.6 Password protection 20
5.6.1 Securing data areas 20
5.7 Various 21

1 Responsible persons:

1.1 Schedule

Start date of First Commissioning:

End date of First Commissioning:

1.2 First commissioning contacts

Name	Company / Department	Contact information
Mr. Peter Writer	*Machine Builder ABC*	*PLC Program Contact*

Copyright © Siemens AG 2020 All Rights Reserved.

2 Machine Test summary

2.1 Summary

2.2 Result

2.3 Release of the machine recommended (YES/NO/Reason):

2.3.1 Exception List

2.3.2 The machine has been released for sale with the following exceptions:

Function:	Test Number	Remark

2.4 Changes to standard data – as reference data

Extend table as necessary

Machine data name + ID number	Original Value	New Value

2.5 Data Backup

Create data backup after completing the test and attach to the protocol.

Appendices:

- (NC Archive)
- (PLC Archive)
- (Drive Archive)
- (HMI Archive)
- (Complete archive) (Backup)(Setup)
- (Clone TGZ file for series production)
  (Image for restore)
- (Ghost image IPC)
- (License key data)

Archive format dependent upon control type:

SINUMERIK ONE	DSF Format
SINUMERIK 840D sl	ARC Format
SINUMERIK 828D	ARD Format

Copyright © Siemens AG 2020 All Rights Reserved.

3 Change Log

3.1 Changes to the machine.

This table describes features added and tested after the protocol was completed, it serves as a confirmation for the Machine Tool Builder that prior to machine delivery all features were tested without exception.

Date	Change	Reason	Tested	Responsible person	Signature	Contact information
			☐			
			☐			
			☐			
			☐			
			☐			
			☐			
			☐			

Components contained in the machine configuration.

Machine

Type of Machine	
Serial No.	
Configuration	
Axes	
Spindle	
Miscellaneous	

3.2 Control

NC type	SINUMERIK ONE / Sinumerik 840D sl / Sinumerik 828D
Serial No.	
Machine control panel	
Software version:	
User software	
PLC Application	

Copyright © Siemens AG 2020 All Rights Reserved.

4 Test objective

4.1 Note

The test plan extends the commissioning documentation of Sinumerik Operate.

The application is tested with the current test plan, which helps to achieve for the most part an error-free interaction of SINUMERIK with the machine. No guarantee for fault-free machine application can be guaranteed.

Siemens is not responsible for the PLC safety related functions which are implemented on the Machine. The Machine tool builder is responsible in satisfying the receiving countries "Safety regulations regarding CNC machine tools" regulations.

Basic knowledge of operation and programming of Sinumerik is required for carrying out the tests. In case of some test tasks, it is necessary to start an automatic program or to trigger machine functions. It is, therefore, explicitly indicated that all operations that trigger machine functions should only be carried out by the employee of the machine manufacturer, who is authorized to do this.

The necessary programs are to be created directly at the machine.

ALL RELEVANT TESTS HAVE TO BE CARRIED OUT WITH THE FOLLOWING CONDITIONS:	
1) Siemens Code (DIN) and ShopMill/ShopTurn with →	**Metric system of units**
2) Siemens Code (DIN) and ShopMill/ShopTurn with →	**Inch System of Units**
3) ISO / dialect with →	**Metric system of units – if activated**
4) ISO / dialect with →	**Inch system of units – if activated**

Copyright © Siemens AG 2020 All Rights Reserved.

5 Functions tested

Confirm the functions as per the test plan

Test.	Function	Result / Accepted/To Do
5.1	**Machine**	
5.1.1	Jog → Reference	*All tests completed*
5.1.2	Jog → MCP	*To Do´s in list*
5.1.3	Jog → HMI Soft keys	
5.1.4	MDA → Teach-in	
5.1.5	MDA → HMI Soft keys	
5.1.6	MDA → MDA → Teach-in	
5.1.7	AUTO → DRF, Jog	
5.2	**Technology**	
5.2.1	Milling General	
5.2.2	Milling 3+2	
5.2.3	Milling 5 Axis	
5.2.4	Turning Basic	
5.2.5	Turning Advanced	
5.3	Parameter	
5.3.1	Machine Builder Options	
5.3.2	Measure Tool	
5.3.3	Setting data	
5.3.4	Magazine	
5.3.5	Tool List	
5.4	**Machine Geometry**	
5.4.1	Circularity Test HMI	
5.4.2	Ball bar circularity Test	
5.5	**Safety integrated**	
5.5.1	Safety Integrated protocol	
5.6	General	
5.6.1	Securing data areas	
5.7	Various	

Copyright © Siemens AG 2020 All Rights Reserved.

Detailed Test
5.1 Machine Area
5.1.1 Jog → Ref

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	3	Reference not necessary (Absolute measuring system) (Start-up in JOG)	☐		☐			X	X	X
T+M	4	Prevention of Axis faults, when starting in axis operation, the Spindles with axis functionality should be synchronized in jog prior to NC Start	☐		☐			X	X	X
T+M	5	Additional equipment requiring reference point approach should be referenced prior to the first NC Start (Turret). “Alarm prevention”	☐		☐			X	X	X
T+M	9	The feedrate and Spindle enable signals should not be active immediately after power-on of the machine. The operator must enable these signals with the respective key presses on the MCP	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.2 Jog → MCP

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	If the axis is resting on the Hardware limit switch, it should be possible to JOG the axis away in the opposite direction.	☐		☐			X	X	X
T+M	2	It should be possible in the RESET state to change to JOG mode	☐		☐			X	X	X
T+M	3	When changing from REF to Jog - Operator must select again the required axis. There should be no pre-selected axis.	☐		☐			X	X	X
T	5	With Turning machines, it should also be possible to move more axes simultaneously. The machine builder has to realize this in the PLC.	☐		☐			X	X	X
M	6	With Milling machines, it is also possible to Jog axes simultaneously; however, the machine builder has to realize this.	☐		☐			X	X	X
T+M	7	The machine builder should also ensure that the rotary axis can be jogged, including rotary axis functionality with Spindles. C+ button = spindle direction M4 C- button = spindle direction M3	☐		☐			X	X	X
T+M	8	The feedrate override should be effective on all axes which can be Jogged. All axes should come to standstill when the feed override is set to zero.	☐		☐			X	X	X
T+M	9	When the feed stop button is pressed the Jog plus and minus buttons should have no effect on any axis motion. The axis should all come to a stop.	☐		☐			X	X	X
T+M	10	When the spindle stop button is pressed the Jog plus and minus buttons should have no effect on the spindle motion. The spindles should all come to a stop.	☐		☐			X	X	X
T	11	When Spindle direction buttons are available, the spindle should move clockwise or anti-clockwise dependent upon which button is pressed. The default speed is defined in the machine data.	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

T+M	12	The Spindle override should correct the currently controlled spindle/s.	☐		☐			X	X	X
T+M	13	Do all axis move according to the work plane defined in the DIN 6625 standard.	☐		☐			X	X	X
T+M	14	When direction keys plus and minus are pressed simultaneously, the axis in motion should stop immediately. Further motion can only be achieved when both buttons are once again released, and the desired direction button pressed again	☐		☐			X	X	X
T+M	15	WCS >><< MCS With key not pressed (MCS) the axis should be moved using the Machine axis interface, and the axis will move on an individual basis, without active rotations. With key pressed (WCS) the axis should be moved using the Channel axis interface, and the axis will move on an individual basis, with active rotations.	☐		☐			X	X	X
T+M	16	Repos With interruption in Automatic, it should be possible to switch to Jog, move axis away from the contour, subsequently return to Automatic / Repos and Jog the axis back to the previous position.	☐		☐			X	X	X
T+M	17	TCS Tool Coordinate System Jog should be activated dependent upon machine kinematic	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.3 Jog → HMI Soft keys

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	T,S,M → T-D Select a tool from the tool list, using SK TOOL and IN MANUAL. Press cycle start tool should be transferred into the spindle or positioned before the work piece. The tool must not rotate if already present in the spindle or present before the work piece. Repeat the process with other tools from the magazine. Prior to changing from a driven tool, the rotating tool should be stopped prior to changing.	☐		☐			X	X	X
T+M	2	Spindle → Set a spindle speed and activate "Cycle Start" (together with spindle M function) Repeat the procedure with further spindles.	☐		☐			X	X	X
T+M	3	Spindle gear change→ Change gear, if a gear box is available on the machine.	☐		☐			X	X	X
T+M	4	Spindle direction → Program M3 and M4 on all spindles which are available on the machine and ensure for the programmed code that spindle rotates in the correct direction.	☐		☐			X	X	X
T+M	5	Measure Tool → Manual → Test with all types of possible tools Automatic → Test with all types of possible tools	☐		☐			X	X	X
T+M	6	Position → All axis must be available, ensure that it is possible to move the spindle (C-Axis function) without first running in spindle operation	☐		☐			X	X	X
T+M	7	MCS / WCS → Should change only the display from MCS/WCS	☐		☐			X	X	X
T+M	8	Hand wheel → Test in both MCS and WCS Syncron-actions→. show currently active SYNACT Settings → Test and acknowledge functionality of the Setting data's	☐		☐			X	X	X

5.1.4 MDA → Teach-in

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Teach-in → It should be possible to Jog the axis in MDA - Teach-in **OPTION 828D**	☐		☐			X	X	X
	2		☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.5 MDA → HMI Softkeys

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Program Control → When the PRT checkbox is set the axis should work internally, and not physically move the axis. The signal should work only in Automatic mode - when changing to Jog/MDA the signal should not be active except when an external switch is used.	☐		☐			X	X	X
T+M	2	Program Control → DRY - When selected, the axis should move with the feedrate which is defined in Settings. This should only be available in Automatic.	☐		☐			X	X	X
T+M	3	Program Control → M01 - When selected, the next occurrence of M01 in the NC program should cause the NC program to stop and wait for the next NC start.	☐		☐			X	X	X
T+M	4	Program Control → When DRF is selected it should be possible in the Automatic mode of operation to move the axis with the Hand wheel, the position will be stored in the DRF offset. **OPTION 828D**	☐		☐			X	X	X
T+M	5	Program Control → SKP - When selected, the next NC block with the Skip block instruction will not be executed.	☐		☐			X	X	X
T+M	6	G functions → G - Functions for the reset state of the machine - To be checked for: Turning Milling	☐		☐			X	X	X
T+M	7	Auxiliary functions → Machine tool builder should check that the functions which he does not allow in MDA are not processed.	☐		☐			X	X	X
T+M	8	>> Extend SK → All G functions G - Functions for the reset state of the machine - To be checked for: Turning Milling	☐		☐			X	X	X
T+M	9	> Extend menu → Hand wheel → Move axis using Hand wheel in the machine axis interface or using Geometry interface in connection with Teach-in mode **OPTION 828D**	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.6 MDA → MDA → Teach-in

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	MDA → Tool change in MDI window - T01 + Cycle start for Turning machines T01 M06 + Cycle start for Milling machines.	☐		☐			X	X	X
T+M	2	MDA → It should not be possible to Jog the axis in MDA - exception: Teach-in is active	☐		☐			X	X	X
T+M	**3**	Teach-in → Hand wheel → Move axis using Hand wheel in the machine axis interface or using Geometry interface in connection with Teach-in mode **OPTION 828D**	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.7 AUTO

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Move axis using Hand wheel/Jog/Inc in the machine axis interface to generate a DRF offset **OPTION 828D**	☐		☐			X	X	X
T+M	2	Jog in Automatic → Jog Mode Mask is set to 0 as standard, which means you are not able to Jog the axis in Automatic, with the exception of Teach-in. Jog Mode Mask is set to 1, means you are able to Jog the axis in Automatic. The machine builder should explicitly set this machine data when he requires this functionality.	☐		☐			X	X	X
T+M	3	AUTO → Cycle start	☐		☐			X	X	X
T+M	4	AUTO → Cycle Stop	☐		☐			X	X	X
T+M	5	AUTO → Cycle single block	☐		☐			X	X	X
T+M	6	AUTO → Program reset	☐		☐			X	X	X
T+M	7	AUTO → Feed Stop in Program effect	☐		☐			X	X	X
T+M	8	AUTO → Spindle stop in Program - effect	☐		☐			X	X	X
T+M	9	AUTO → Feed override = 0	☐		☐			X	X	X
T+M	10	AUTO → Spindle override = 50	☐		☐			X	X	X
T+M	11	AUTO → Effect of E-Stop	☐		☐			X	X	X
T+M	12	Program control → PRT - Program Test	☐		☐			X	X	X
T+M	13	Program control → DRY - Dry run feedrate	☐		☐			X	X	X
T+M	14	Program control → M01 - Program stop	☐		☐			X	X	X
T+M	15	Program control → DRF - Differential hand wheel offset **OPTION 828D**	☐		☐			X	X	X
T+M	16	Program control → SKP - Block skip	☐		☐			X	X	X
T+M	17	Block search → M,S,T,H - Effect Search to Tool	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.2 Technology

5.2.1 Milling General

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
M	2	Customer Cycles CUST800 etc.	☐		☐			X	X	X
M	3	Settings for Technology and measuring cycles	☐		☐			X	X	X
M	4	Settings for Tool management views	☐		☐			X	X	X
M	5	Cycle settings when multitasking Turning on Milling machines Milling on Turning machines	☐		☐			X	X	
M	6	Specific customer cycle CUST832	☐		☐			X	X	X
M	7	3/5 Axis M Dynamics MD Test	☐		☐			X	X	X
M	8	Tool change with DRF Deactivation M6	☐		☐			X	X	X
M	9	Cycle 996/9960 Measuring Kinematic	☐		☐					
M	10		☐		☐					
M	11		☐		☐					
M	12		☐		☐					

5.2.2 Milling 3+2

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
M	1	Settings for Swiveling	☐		☐			X	X	X
M	2	Tool change with active CYCLE800 / Transmit / Tracyl	☐		☐			X	X	X
M	3	Settings Transmit / Tracyl	☐		☐			X	X	X
M	4	Reset State with Head/ Table / Mixed kinematic	☐		☐			X	X	
M	5	Power off state / Head / Table / Mixed kinematic	☐		☐			X	X	

Copyright © Siemens AG 2020 All Rights Reserved.

5.2.3 Milling 5 Axis

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
M	1	Settings for Traori	☐		☐			X	X	
M	2	Settings for CUST832	☐		☐			X	X	X
M	3	Correct Trafo Type set Or Trafo based upon Kinematic Chain	☐		☐			X	X	X
M	4	Tool Change with active Traori	☐		☐			X	X	
M	5	Reset State with Head/ Table / Mixed kinematic	☐		☐			X	X	
M	6	Power off state / Head / Table / Mixed kinematic	☐		☐			X	X	
M	7	Settings for Orientation / ORIAXES reset ORIEULER ORIMKS ORIWKS reset	☐		☐			X	X	

5.2.4 Turning Basic

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T	1	Is the coordinate system displayed correctly in the editor?	☐		☐			X	X	X
T	2	Are the softkeys for the standard cycles displayed in the editor?	☐		☐			X	X	X
T	3	Is the “plunge-cutting” and “plunge-turning” softkey displayed in the editor?	☐		☐			X	X	X
T	4	Are the softkey for the remaining-material machining displayed in the editor?	☐		☐			X	X	X
T	5	Is the “Tailstock” parameter displayed in the program header? (only in ShopTurn)	☐		☐			X	X	X
T	6	Is the “parts gripper” parameter displayed in the cut-off cycle?	☐		☐			X	X	X
T	7	Does the parts gripper in the cut-off cycle function correctly?	☐		☐			X	X	X
T	10	Is a program simulated correctly with a open working-area door?	☐		☐			X	X	X
T	11	Is the simulation aborted when the “Reset” key is pressed?	☐		☐			X	X	X
T	12	Is the calculated program run time updated during the simulation?	☐		☐			X	X	X
T	13	Is a program executed correctly when you press “Cycle Start”?	☐		☐			X	X	X
T	14	Is the “Base block” softkey displayed?	☐		☐			X	X	X
T	15	Is the” Simultaneous recording” softkey displayed?	☐		☐			X	X	X
T	16	Does “Block search” work correctly?	☐		☐			X	X	X
T	17	Does” REPOS” work correctly?	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.2.5 Turning Advanced

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T	1	Is a Face milling contour milled correctly on main spindle?	☐		☐			X	X	X
T	2	Is a Face milling contour milled correctly on counter spindle?	☐		☐			X	X	X
T	3	Is a Peripheral milling contour milled correctly on main spindle?	☐		☐			X	X	X
T	4	Is a Peripheral milling contour milled correctly on counter spindle?	☐		☐			X	X	X
T	5	Is a Y-Axis milling contour milled correctly on main spindle?	☐		☐			X	X	X
T	6	Is a Y-Axis milling contour milled correctly on counter spindle?	☐		☐			X	X	X
T	7	Is a B-Axis milling contour milled correctly on main spindle?	☐		☐			X	X	
T	8	Is a B-Axis milling contour milled correctly on counter spindle?	☐		☐			X	X	
T	9	Is a position pattern drilled/tapped correctly on main spindle?	☐		☐			X	X	X
T	10	Is a position pattern drilled/tapped correctly on counter spindle?	☐		☐			X	X	X
T	11	Does the counter spindle cycle correctly? (only in ShopTurn)	☐		☐			X	X	X
T	12	Does turning with B axis on the main spindle correctly?	☐		☐			X	X	
T	13	Does turning with B axis on the counter spindle correctly?	☐		☐			X	X	
T	14	Are the softkeys for the measuring cycles displayed in the editor?	☐		☐			X	X	X
T	17	Does a workpiece measurement program work correctly?	☐		☐			X	X	X
T	18	Does the measurement result pictures display correctly?	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.3 Parameter

5.3.1 Machine Builder Options

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Test the commissioning of machine builder options **EASY EXTEND 828D**	☐		☐			X	X	X
T+M	2	Test the activation / deactivation of machine builder options **EASY EXTEND 828D**	☐		☐			X	X	X
T+M	3	Test Regular Maintenance Tasks **MAINTENANCE PLANNER 828D**	☐		☐			X	X	X

5.3.2 Measure tool

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Measure the length of the Tool	☐		☐			X	X	X
T+M	2	Measure the Diameter / radius of the tool	☐		☐			X	X	X

5.3.3 Setting data

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Spindle data - minimum spindle speed should be set to a value not equal to 0	☐		☐			X	X	X

5.3.4 Magazine

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Relocate a tool in the Magazine	☐		☐			X	X	X
T+M	2	Position the Magazine	☐		☐			X	X	X
T+M	3	Rename a tool	☐		☐			X	X	X
T+M	4	Disable magazine locations -	☐		☐			X	X	X
T+M	5	Test oversize tools	☐		☐			X	X	X
T+M	6	Test tools with fixed locations	☐		☐			X	X	X
T+M	7	Test interruption scenarios in tool changer ■ Reset the channel ■ Emergency stop ■ Isolator switch	☐		☐			X	X	X
M	8	Test Hand loaded Tools in program	☐		☐			X	X	X
T+M	9	L6 Optimized, Spindle Tool = Pre-selected Tool	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.3.5 Tool List

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Enter a new tool directly in the spindle	☐		☐			X	X	X
T+M	2	Enter a new tool directly in the Magazine	☐		☐			X	X	X
T+M	3	Enter a new tool directly in the Buffer	☐		☐			X	X	X
T+M	4	Delete a tool directly in Spindle	☐		☐			X	X	X
T+M	5	Delete a tool from the Magazine	☐		☐			X	X	X
T+M	6	Delete a tool from the buffer	☐		☐			X	X	X
T+M	7	Unload the tool in the spindle	☐		☐			X	X	X
T+M	8	Machine with sub-spindle should be possible to create 2 tools for 1 turret position (different tool types !!!)	☐		☐			X	X	X
T+M	9	It must be possible to program/call a tool with tool name (e.g. T="name") and also turret position T1	☐		☐			X	X	X
T+M	10	Relocate a tool in the Spindle	☐		☐			X	X	X
T+M	11	Tool Change in Inch and Metric	☐		☐			X	X	X

5.4 Geometry

5.4.1 Circularity test HMI

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Geometry axis	☐		☐			X	X	X
T+M	2	Thread boring	☐		☐			X	X	X

5.4.2 Ball-bar circularity test

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Machine axis	☐		☐			X	X	X
T+M	2	Geometry axis	☐		☐			X	X	X

5.5 Safety Integrated

5.5.1 Safety related test

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Safety integrated acceptance Test.	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.6 Password protection

5.6.1 Securing data areas

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Key switch → Tools Tool wear length and radius Tool length and radius	☐		☐			X	X	X
T+M	2	Key switch → Create a tool	☐		☐			X	X	X
T+M	3	Key switch → Delete a tool	☐		☐			X	X	X
T+M	4	Programming → Open program	☐		☐			X	X	X
T+M	5	Programming → Edit program	☐		☐			X	X	X
T+M	6	Programming → Select program (NC)	☐		☐			X	X	X
T+M	7	Offsets → fine coarse	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.7 Various

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	3	Clone backup (TGZ) available for Series machines.	☐		☐			X	X	X
T+M	4	Country specific safety regulations included in PLC program + door interlocks etc.	☐		☐			X	X	X
T+M	5	NCK, HMI, PLC and Drive archives tested.	☐		☐			X	X	X
T+M	6	TGZ file for series production tested	☐		☐			X	X	X
T+M	7	Adaptive Friction compensation active	☐		☐			X	X	X
T+M	8	Top Surface date set and tested	☐		☐			X	X	X
T+M	9	Customer Cycles optimized and tested	☐		☐			X	X	X
T+M	10	Top Speed data set and Tested	☐		☐			X	X	X
T+M	11	Save Data VRam copy	☐		☐			X	X	X
+M	12	PLC Written according to 828D design document	☐		☐					X
T+M	13	ESR has been commissioned	☐		☐			X	X	X
T+M	14	ELG has been commissioned	☐		☐			X	X	
T+M	15	CA ECO Commissioned	☐		☐			X	X	X
T+M	16	CA Advanced	☐		☐			X	X	
T+M	17	Sidescreen activated	☐		☐			X	X	X
T+M	18	Display Manager activated	☐		☐			X	X	X
T+M	19	WinCC installed and configured	☐		☐			X	X	
T+M	20	Spindle Load monitoring	☐		☐			X	X	
T+M	22	Axis optimization with AST	☐		☐			X	X	X
T+M	23	Manual axis optimization	☐		☐			X	X	X
T+M	24	Compile cycles loaded	☐		☐			X	X	
T+M	25	Tools for Sinumerik integrate	☐		☐			X	X	
T+M	26	OPC UA Server commissioned	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	29	ILC Intelligent Load control	☐		☐			X	X	X
T+M	30	IDC Intelligent Dynamic Control	☐		☐			X	X	X
T+M	31	APC ECO	☐		☐			X	X	X
T+M	32	Nodding Compensation	☐		☐			X	X	X
T+M	33	IPC Data backup	☐		☐			X	X	
T+M	34	Deactivate Softkeys	☐		☐			X	X	
T+M	35	Create Machine data user views	☐		☐			X	X	
T+M	36	IPC/PCU Ghost image	☐		☐			X	X	
T+M	37	CNC Lock Feature	☐		☐					X
+M	38	Standard password changed	☐		☐			X	X	X
T+M	39	Machine registration data	☐		☐			X	X	X
T+M	40	License key back-up	☐		☐			X	X	X
T+M	41	1st Setup complete	☐		☐			X	X	X
T+M	42	2nd Setup complete	☐		☐			X	X	X
T+M	43	Warranty log file	☐		☐			X	X	X
T+M	44		☐		☐					
T+M	45		☐		☐					
T+M	46		☐		☐					
T+M	47		☐		☐					
T+M	48		☐		☐					
T+M	49		☐		☐					
T+M	50		☐		☐					
T+M	51		☐		☐					

Copyright © Siemens AG 2020 All Rights Reserved.

Service and Commissioning Tools

Module Description:

In order to integrate the SINUMERIK ONE CNC to a machine tool, various software tools are required. These tools enable the machine tool builder to create the PLC program for the machine, configure the drives, alarm text etc.

Software to be installed:

- TiA Portal (V20)
- PLC Toolbox DVD (PLC basic program, Sinumerik Add-on for TiA, NC Var selector)
- Create My Virtual Machine (V1.5.0.1)
- CNC Software (V6.25)
- Access MyMachine - File transfer, Remote control (Viewer) (V5.0)
- Create MyConfig (V6.8)

Module objective:

Upon completion of this module you will be able to perform the following tasks:

- Install the TiA Portal software
- Install the software tools from the PLC toolbox DVD.
- Setup Communication settings
- Setup the network of the host computer
- Install the Create MyVirtual Machine software tool.
- Install the CNC software
- Install the Access MyMachine software tool.
- Connect to the controller using each of the software packages.
- Install the Create MyConfig

Content:

Sinumerik

TiA Portal Installation

PLC Toolbox Installation.

Access MyMachine Installation.

Create MyVirtual Machine Installation

Create MyConfig Installation

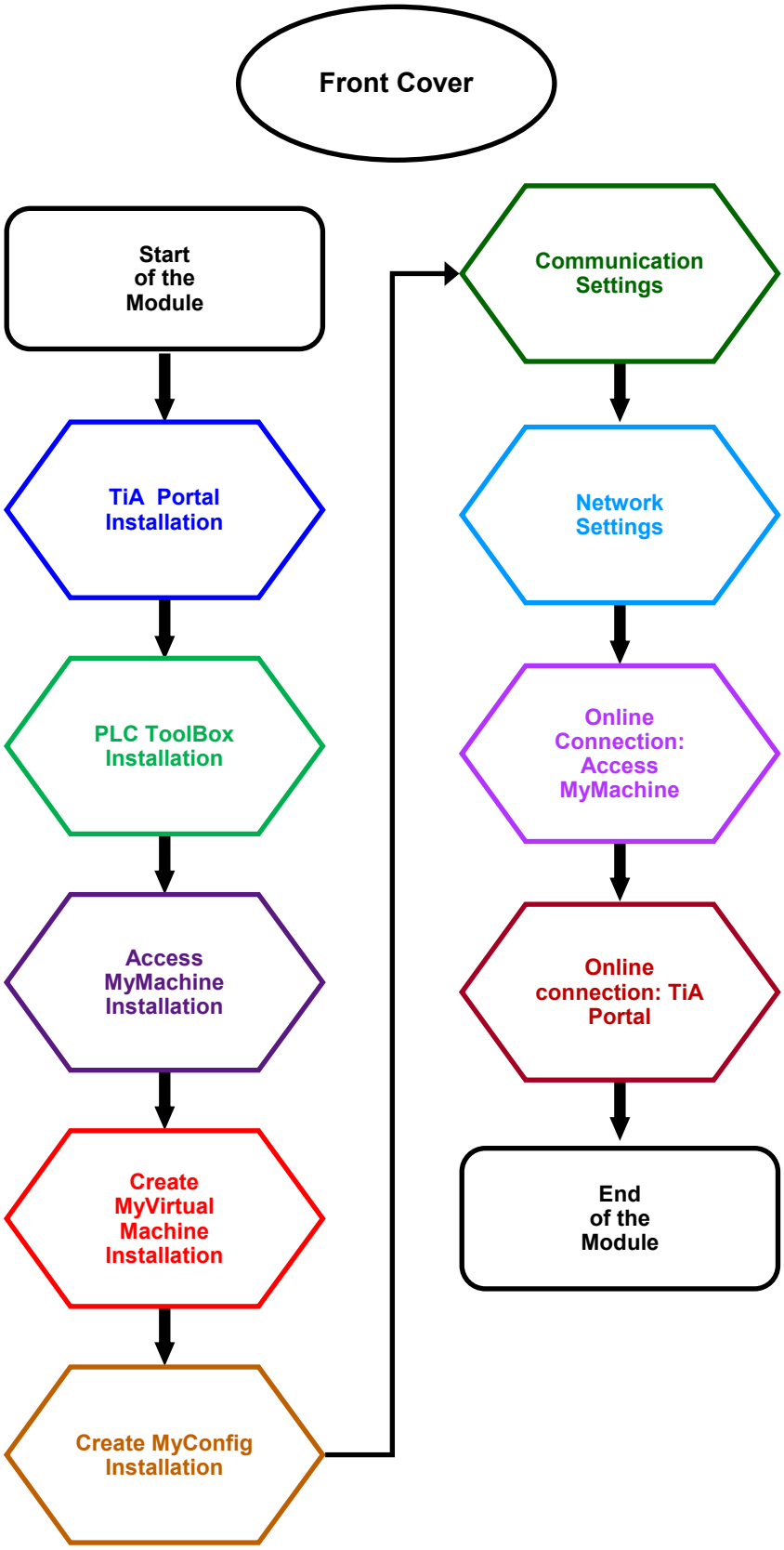
Commination settings

Network Settings

Online Connection - Access MyMachine

Online Connection TiA Portal

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
TiA Portal Installation
PLC ToolBox Installation
Access MyMachine Installation
Create MyVirtual Machine Installation
Create MyConfig Installation
Communication Settings
Network Settings
Online Connection: Access MyMachine
Online connection: TiA Portal
End of the Module

TIA Portal installation:

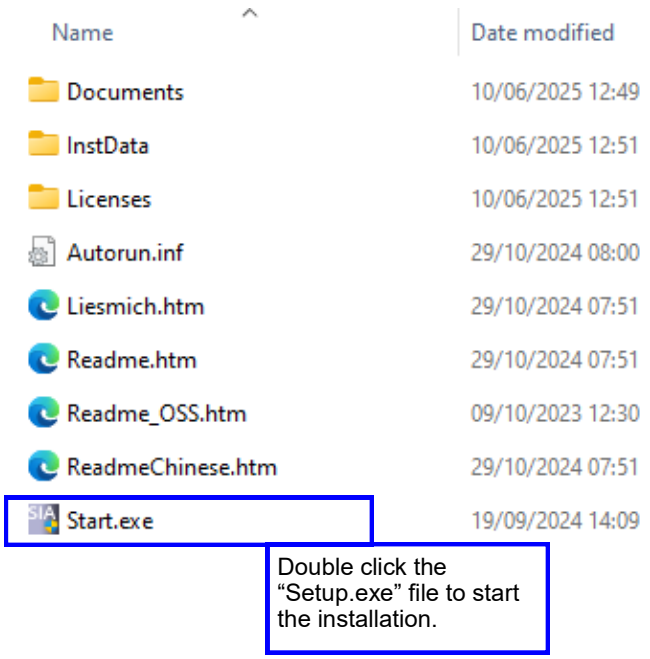
The *Totally Integrated Automation Portal* forms the working environment for a uniform configuration using the following components:

SIMATIC STEP 7 Professional:
SIMATIC STEP 7 PLCSIM:
SIMATIC WinCC:
SIMATIC STEP 7 Safety:

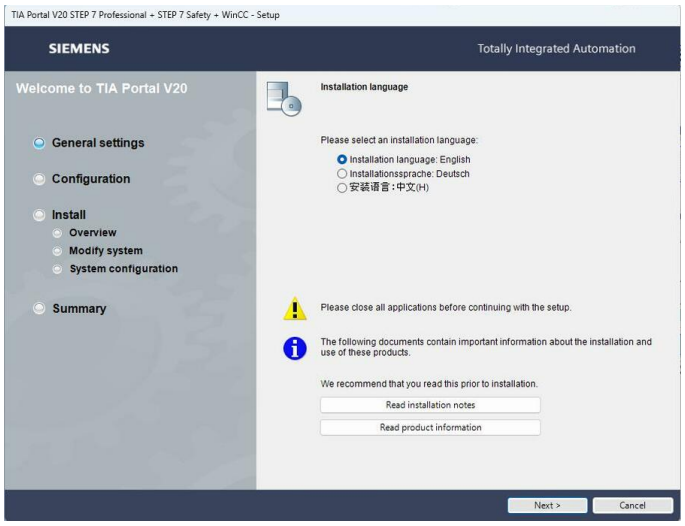
Note:
The Access MyMachine software is not part of the toolbox DVD.

Installation:

To install the TiA Portal software, locate the “Setup.exe” file in the DVD. Start the installation by double clicking the file.



Select the *installation language*

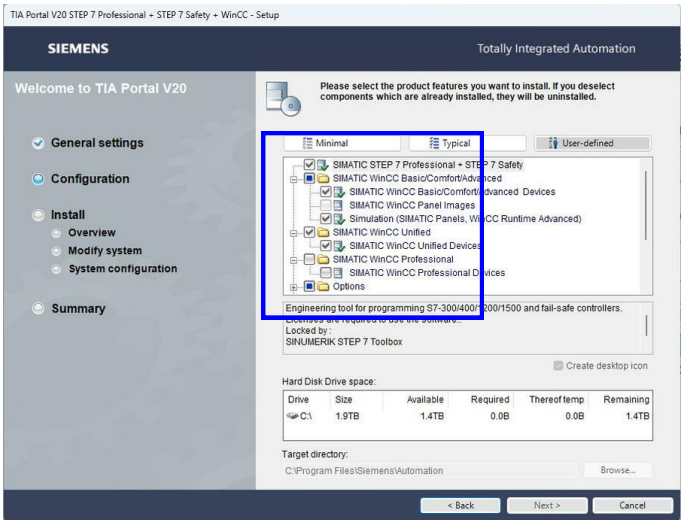


Next >

Select the *product language*

Next >

Select the *product configuration*



Next>

Notes:

Accept the license conditions

Next >

Accept the security settings

Next >

Overview

Next >

Installation

Install

Restart the computer

Restart

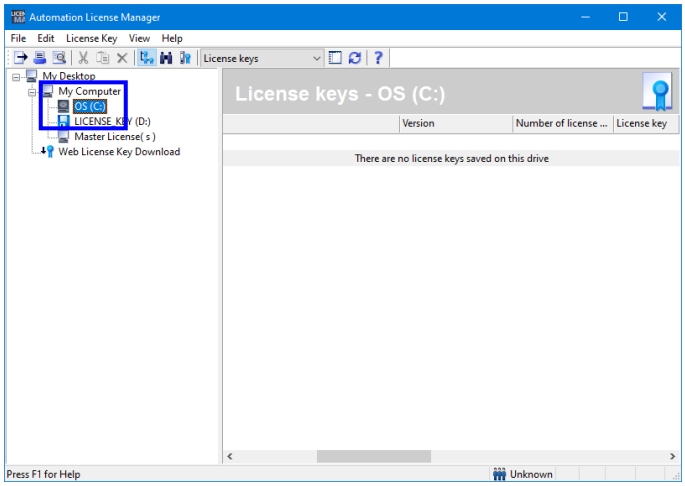
During the installation of the TIA Portal, a **shortcut icon** is created on the PC desktop. The **version** is displayed after right-clicking **Properties/General**.

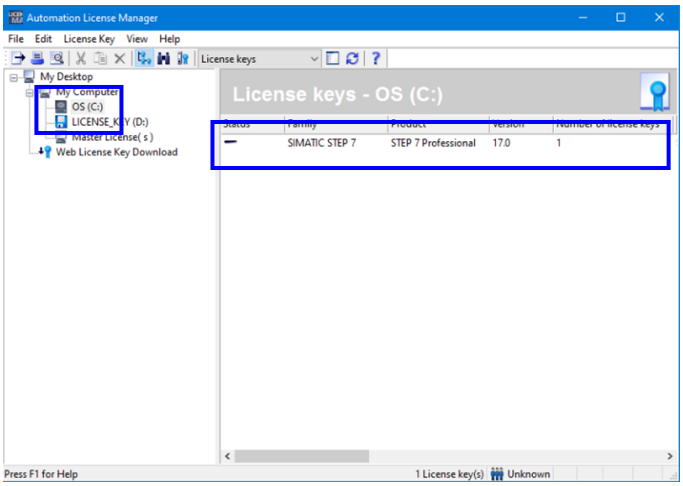


License transfer

If no license carrier (e.g. USB stick) has been found for TiA Portal, the licensing can be performed subsequently in the License Manager.

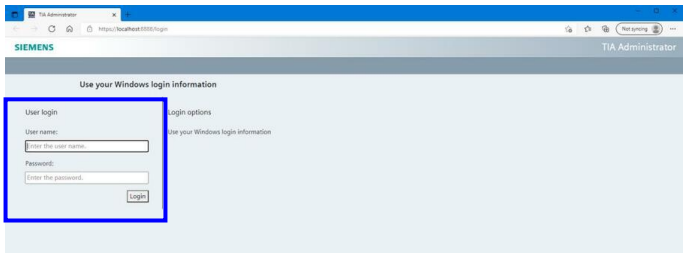
A licence token can be transfered from a USB stick to the C: drive of the PC or PG



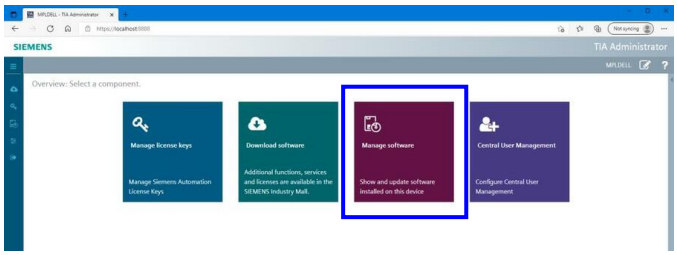


Notes:

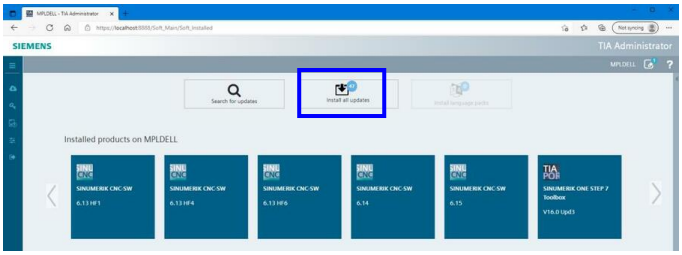
After a successful installation, the TIA Administrator software will run.



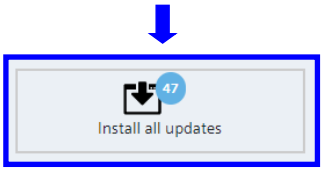
Enter your windows logon name and password.



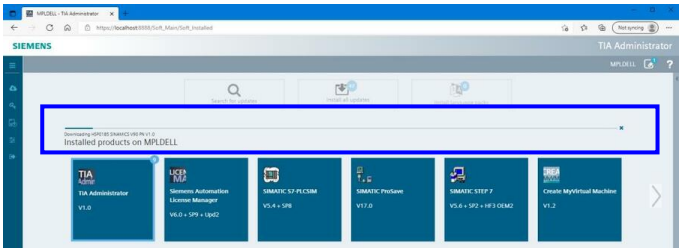
Select “Manage software“



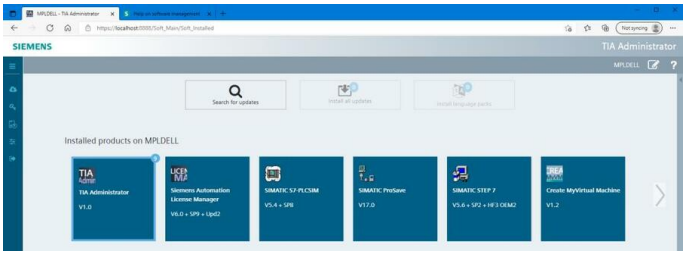
Software searching for any updates, if any updates are found a window will show how many updates are avaliable.

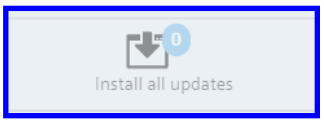


Selection of this icon will start the download.

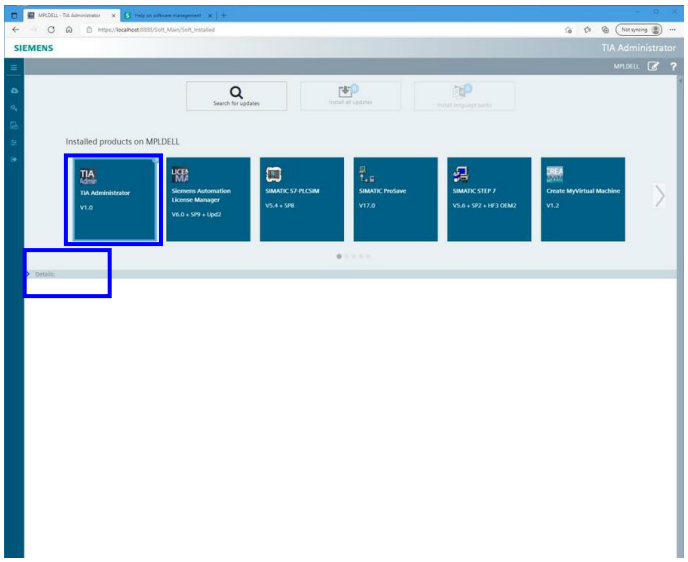


The TIA Administrator, will search for available Siemens, SIMATIC, SINUMERIK,and SINAMICS products and Support Packages and download and installed them.



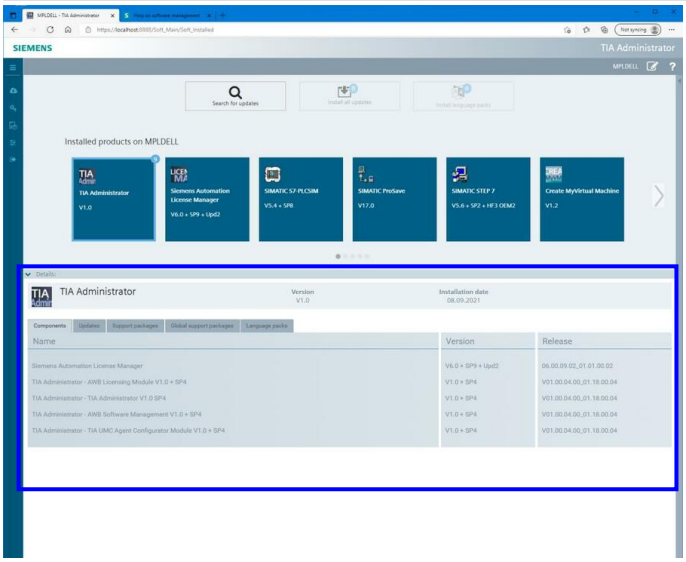


No updates are avaliable.



Details of an installed Siemens, SIMATIC, SINUMERIK,and SINAMICS product can be seen by selecting the “Installed products“ in the top window, and selecting “details“.

Notes:

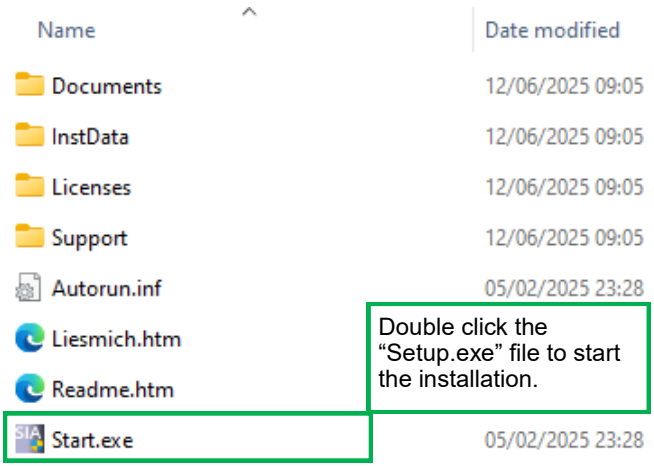


A list of all it's components are shown.

PLC ToolBox Installation

Toolbox content:
PLC Basic Program for SINUMERIK 840D sl
PLC Basic Program for SINUMERIK ONE
Sinumerik Add-on for TiA
NC VAR selection

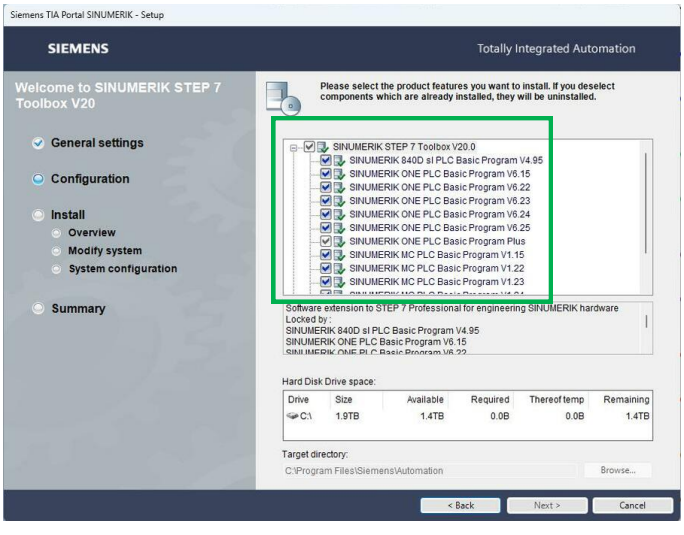
Installation Process:
Installation is started with the "**setup.exe"** file.



The dialog for the **selection of the *installation language*** opens.

➡ Next >

Selection of the *product configuration*
If you select all of the basic program versions, all versions are available subsequently in the library.



➡ Next >

Accept the license conditions

➡ Next >

Accept the security settings

➡ Next >

Overview

➡ Next >

Installation

➡ Install ➡ Finish

Restart the computer

➡ Restart

Notes:

Access MyMachine Installation

Access MyMachine allows a remote operation between the SINUMERIK ONE controller and a computer running windows.

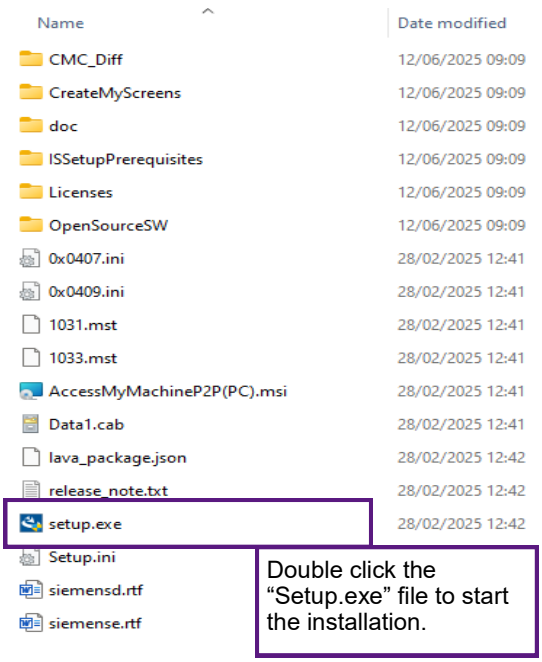
Data such as part programs can be transferred between the remote computer and the controller.

A viewer is included which is used to view and change the controller settings remotely (Dependant upon access rights).

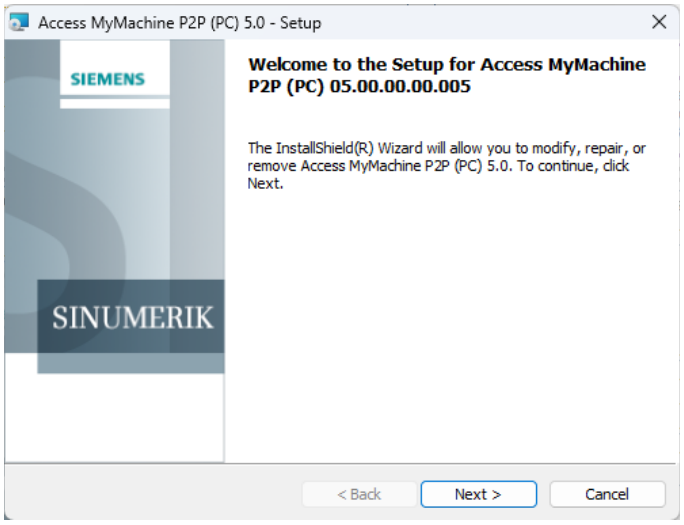
Note:
The Access MyMachine software is not part of the toolbox DVD.

Installation:

To install the Access MyMachine software, locate the “Setup.exe” file in the root of the Access MyMachine DVD. Start the installation by double clicking the file.



Select the *installation version*



Next >

Accept the license conditions

Next >

Select the *Destination folder*

Next >

Overview

Next >

Select to *Install*

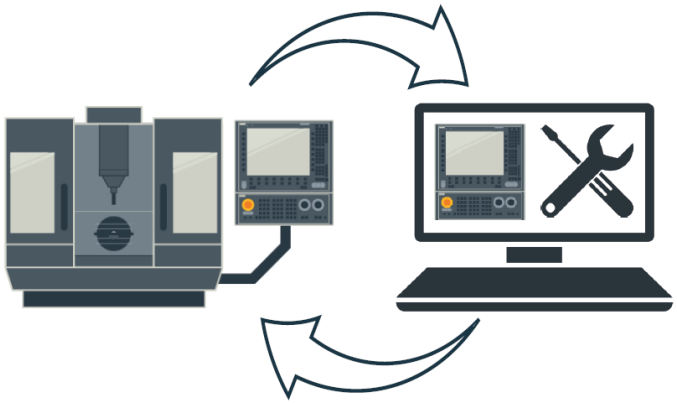
Install

Select the *Finish*

Finish

Notes:

Create MyVirtual Machine Installation



Create MyVirtual Machine is a virtual CNC-System, which simulates a SINUMERIK ONE on a PC. The Hardware components of the controller are simulated as software components

Note:
Create MyVirtual Machine software is not part of the toolbox DVD.

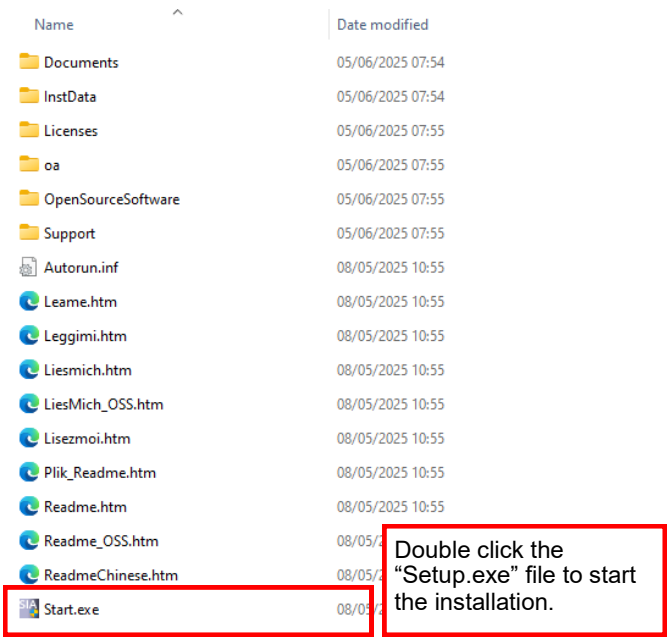
With Create MyVirtual Machine you can develop and test a CNC-Machines NCK, PLC and HMI without any hardware. Several Steps of the machine comissioning can be prepared already on the virtual model. This helps to shorten the commissioning time on the real machine significantly.

Components of Create MyVirtual Machine:

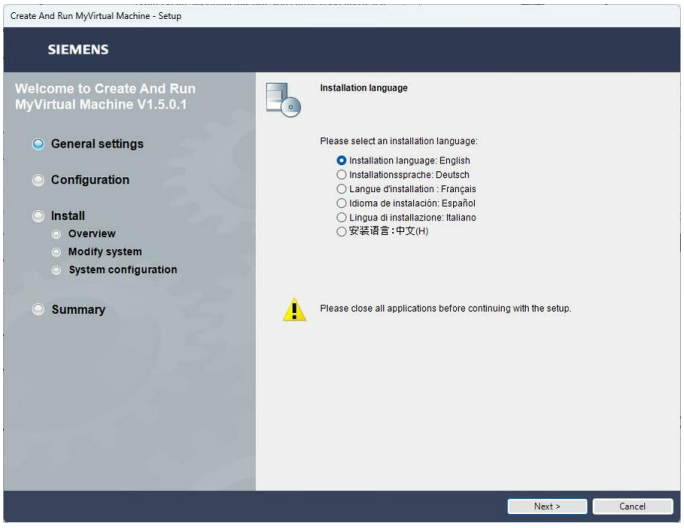
- simNCK: Simulation of the NCK
- SIMATIC S7-PLCSIM Advanced: PLC Simulation based on S7-1500
- HMI: SINUMERIK ONE with a virtual Machine Control Panel
- simDrive (Drive Simulation):Drive Replacement Component for SINAMICS S120
- Open-Interface for the connection of external Simulation Tools
- Integrated Peripheral-Simulation

Installation:

To install Create MyVirtual Machine software, locate the “Setup.exe” file in the DVD. Start the installation by double clicking the file.



Double click the “Setup.exe” file to start the installation.



Next

Notes:

Accept the license conditions

Next >

Accept the security settings

Next >

Overview

Next >

Installation

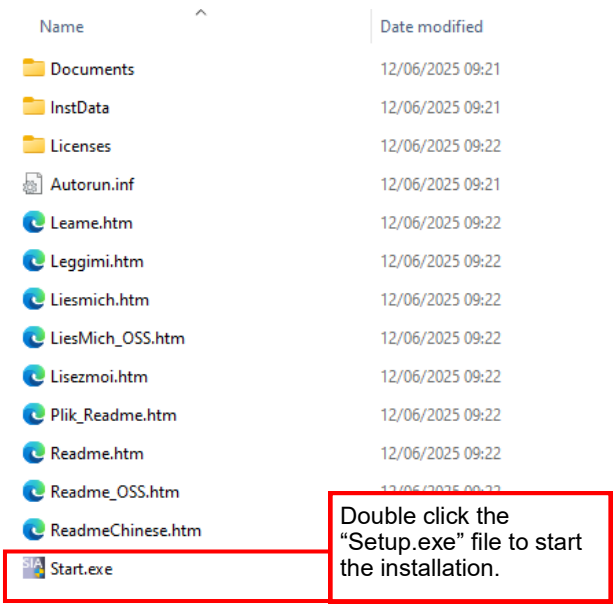
Install

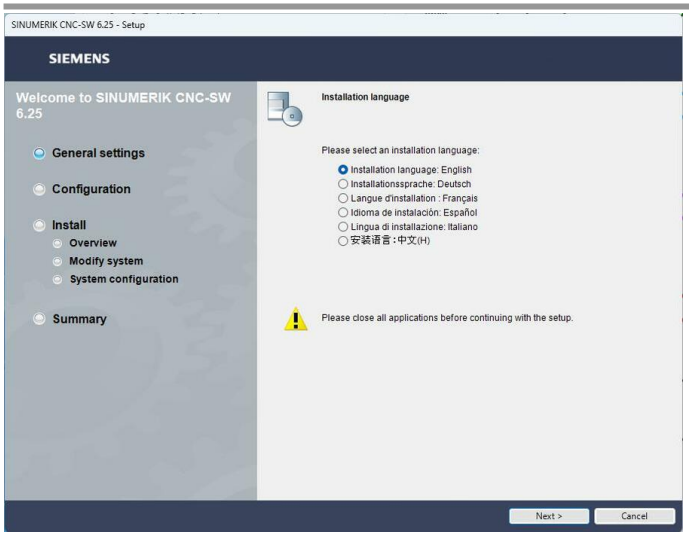
Restart the computer

Restart

After Windows has rebooted, install the **"Virtual SINUMERIK CNC-Software SW 6.25"**

Installation Process:
Installation is started with the "**setup.exe"** file.





Next >

WinPcap is only required if installing CNC SW6.15 xx or earlier CNC SW versions

Accept the license conditions

Next >

Accept the security settings

Next >

Overview

Next >

Installation

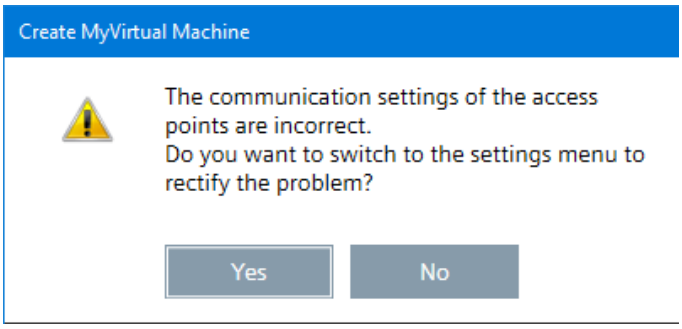
Install

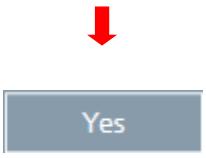
Restart the computer

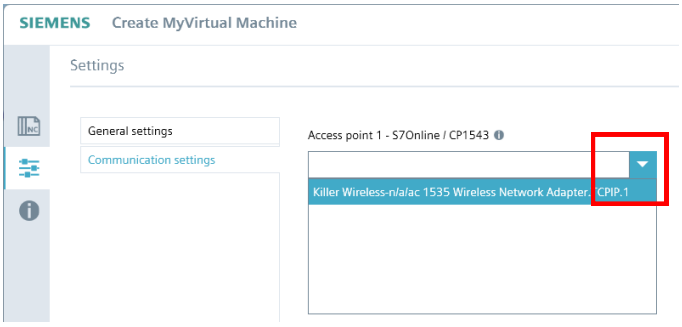
Restart

Notes:

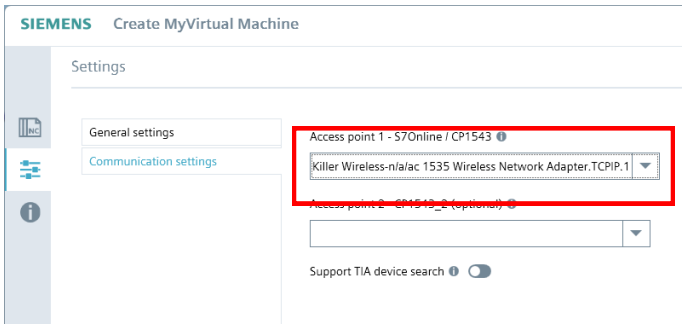
On first power-up of the software a communications error windown is posted, as shown below.







Use the pull down menu to select the the wireless adaptor of the host PC/Laptop.





Notes:

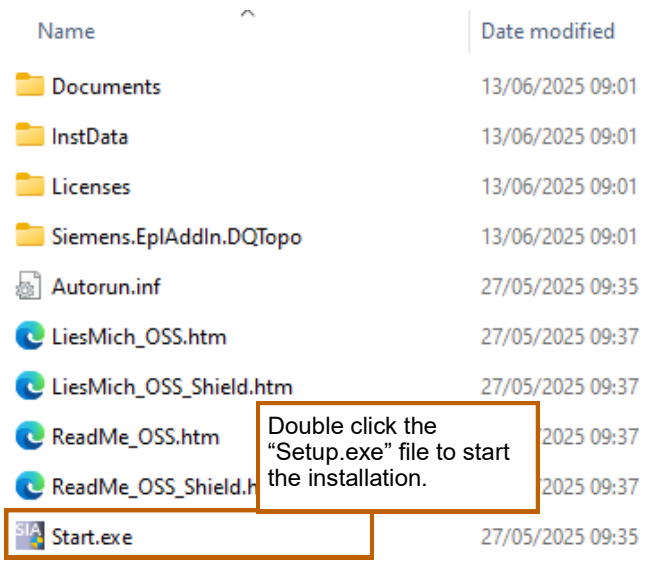
Create MyConfig Installation

CMC is used by machine manufacturers to create and implement a project for automated commissioning during machine production with SINUMERIK ONE.

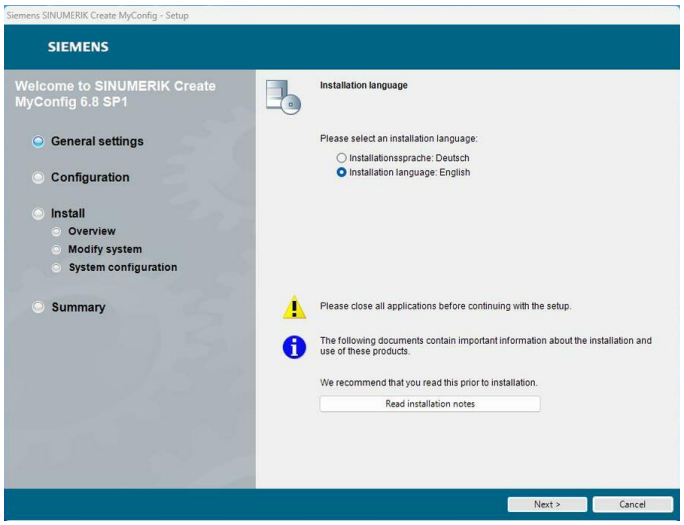
The Diff and Topo components are also used for carrying out diagnostics and service on test machines or machines in the field.

Installation:

To install the Create MyConfig software, locate the “Setup.exe” file in the root of the Create MyConfig DVD. Start the installation by double clicking the file.



Select the *installation version*



Next >

Accept the license conditions

Next >

Select the *Destination folder*

Next >

Overview

Next >

Select to *Install*

Install

Select *Finish*

Finish

Notes:

Communication Settings

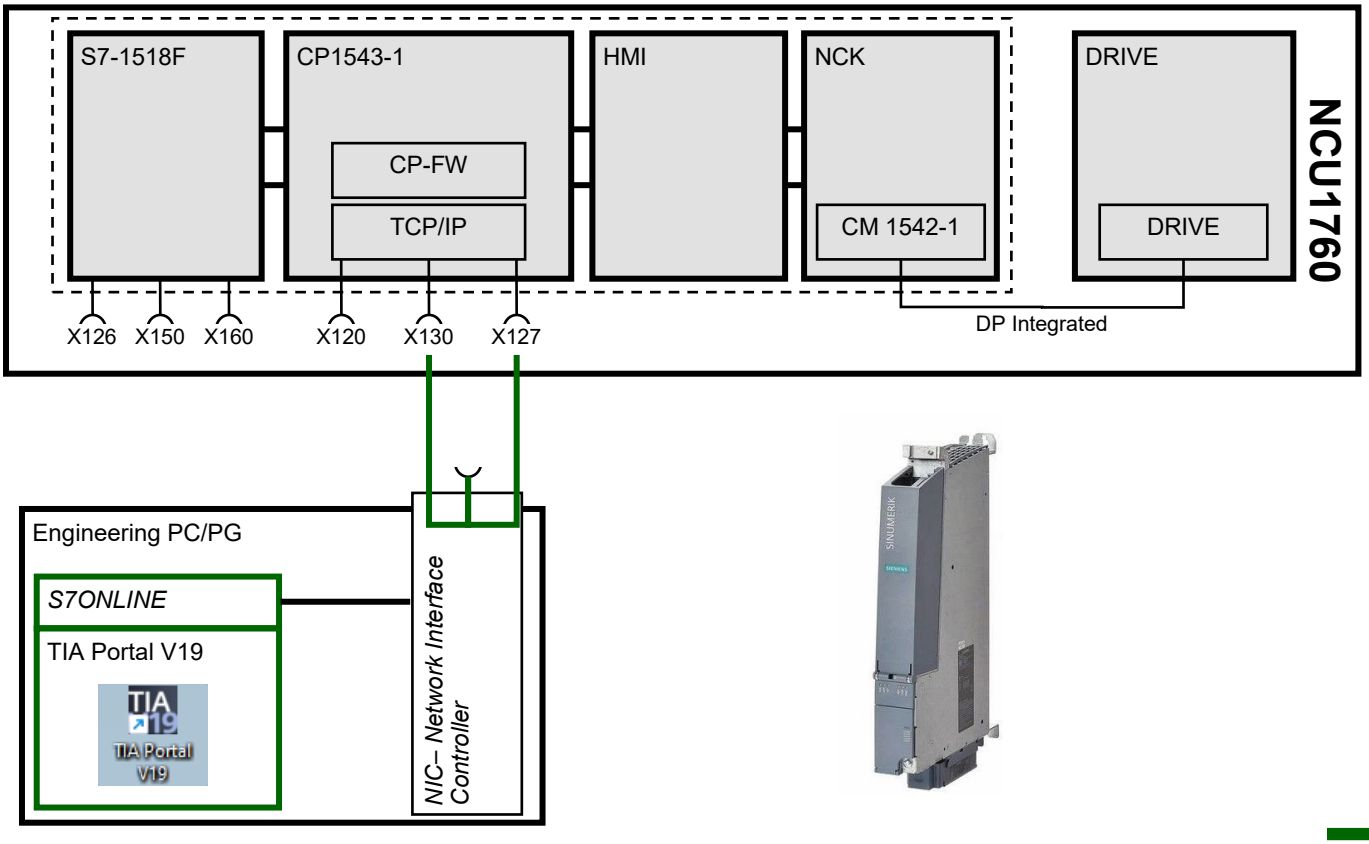
Network Interface on a SINUMERIK ONE(real)
SINUMERIK ONE uses PLC-, CP– and CM-Communication mechanisms based on the S7-1500 family

Interfaces on a SINUMERIK ONE
With the present Version the following interfaces are accessible:

- Ethernet interfaces X120, X130 and X127
- PROFINET interfaces X150, X160
- PROFIBUS DP X126

Load a TIA Portal project via the CP to the PLC
With the present version of SINUMERIK ONE the interfaces X127 or X130 are preferred.
With X127 only the PLC project can be loaded.

The principal schematic of an Engineering PC with TIA Portal connected to a SINUMERIK ONE NCU1760 is shown here:



Notes:

Network Interface on a virtual SINUMERIK ONE
SINUMERIK ONE uses PLC-, CP– and CM-Communication mechanisms based on the S7-1500 family

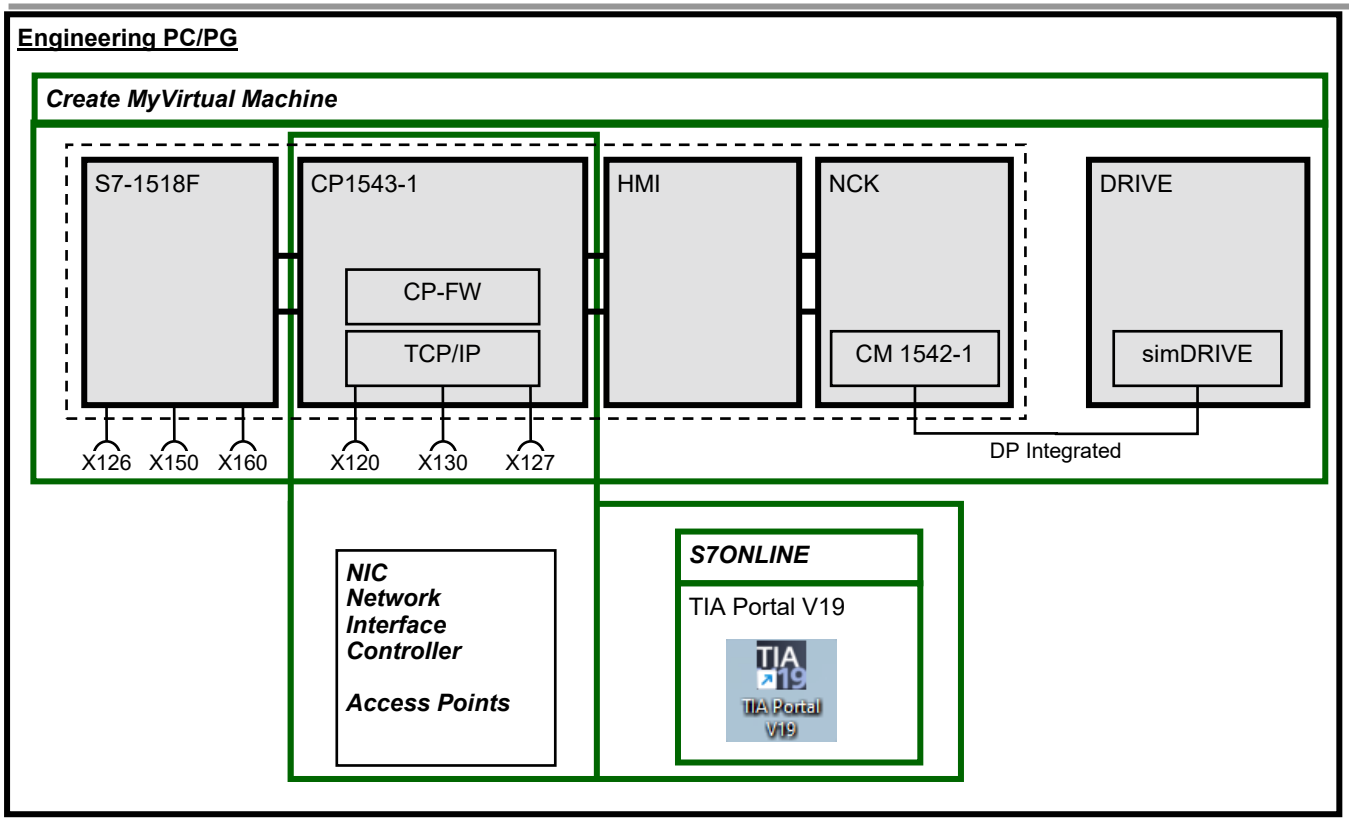
Interfaces on a virtual SINUMERIK
The virtual SINUMERIK uses the same configurable Communciation Interefaces as the real SINUMERIK ONE. With the present Version the interfaces X120, X130 and X127 are accessible.

Specific features and Limitations
The interfaces of the virtual SINUMERIK ONE are treated as one interface during simulation. They are only accessible by the one IP-Address of the Host. The interfaces X126, X150 and X160 are not available for PLC-Periphery. Existing PLC-Periphery within the Virtualization is accessible with direct Read/Write into the PLC I/O interface via the Open-Interface simulated by Create MyVirtual Machine.

The virtual SINUMERIK ONE supports the following communication methods:

- Load a TIA Portal project via the CP to the PLC
- Communication between PLC and TIA Portal or the HMI (e.g. Read/Write Variables)
- S7-Communication between HMI and NCK
- Support of the DCP protocol
- Support of the IPv4 protocol

The principal schematic of an Engineering PC with TIA Portal and a Create MyVirtual Machine installation is shown here:



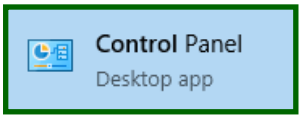
Notes:

Communication Settings for the Create MyVirtual Machine

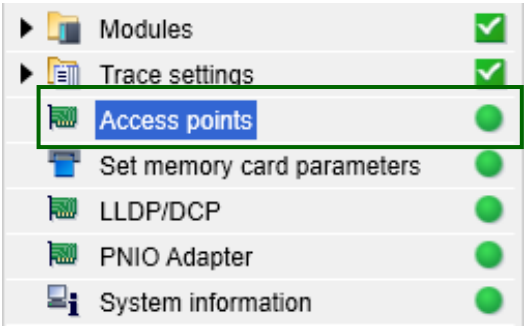
Go to the Windows **Control Panel**, **Communication Settings** and check the **Access Points** for **S7ONLINE** and **SINUMERIK_CP1543** for the correct network adapter.

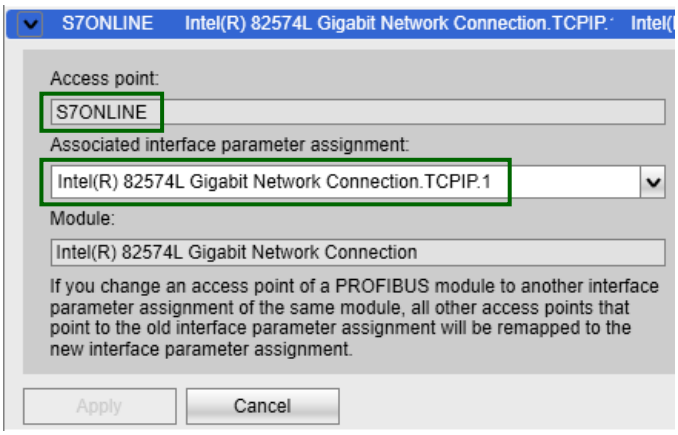
Choose your PC's Ethernet Adapter (e.g. **"Intel(R) 82574L Gigabit Network Connection.TCPIP.1"**)

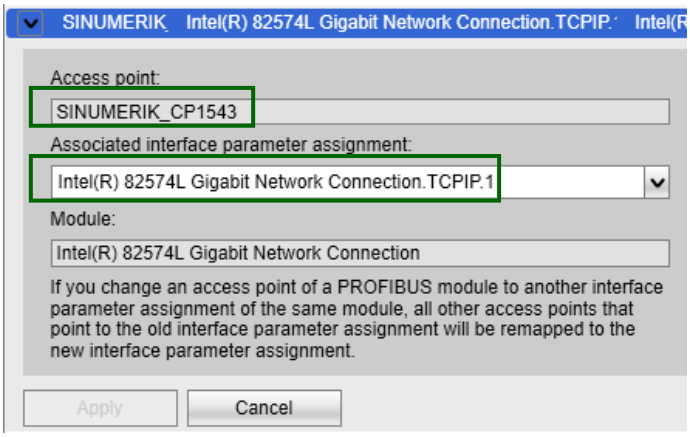
Always select „***.TCPIP.1**"











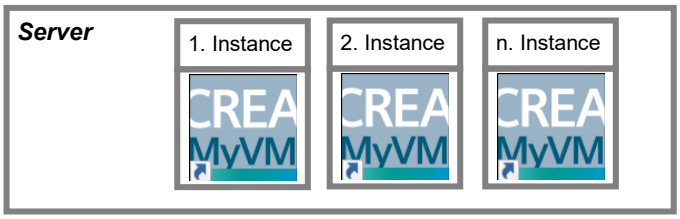
Firewall Settings and Port Forwarding for a virtual SINUMERIK ONE

With the virtualization of SINUMERIK ONE it is possible to run the virtualization on any PC or Server. E.g. several instances of Create MyVirtual Machine can run in a company server.

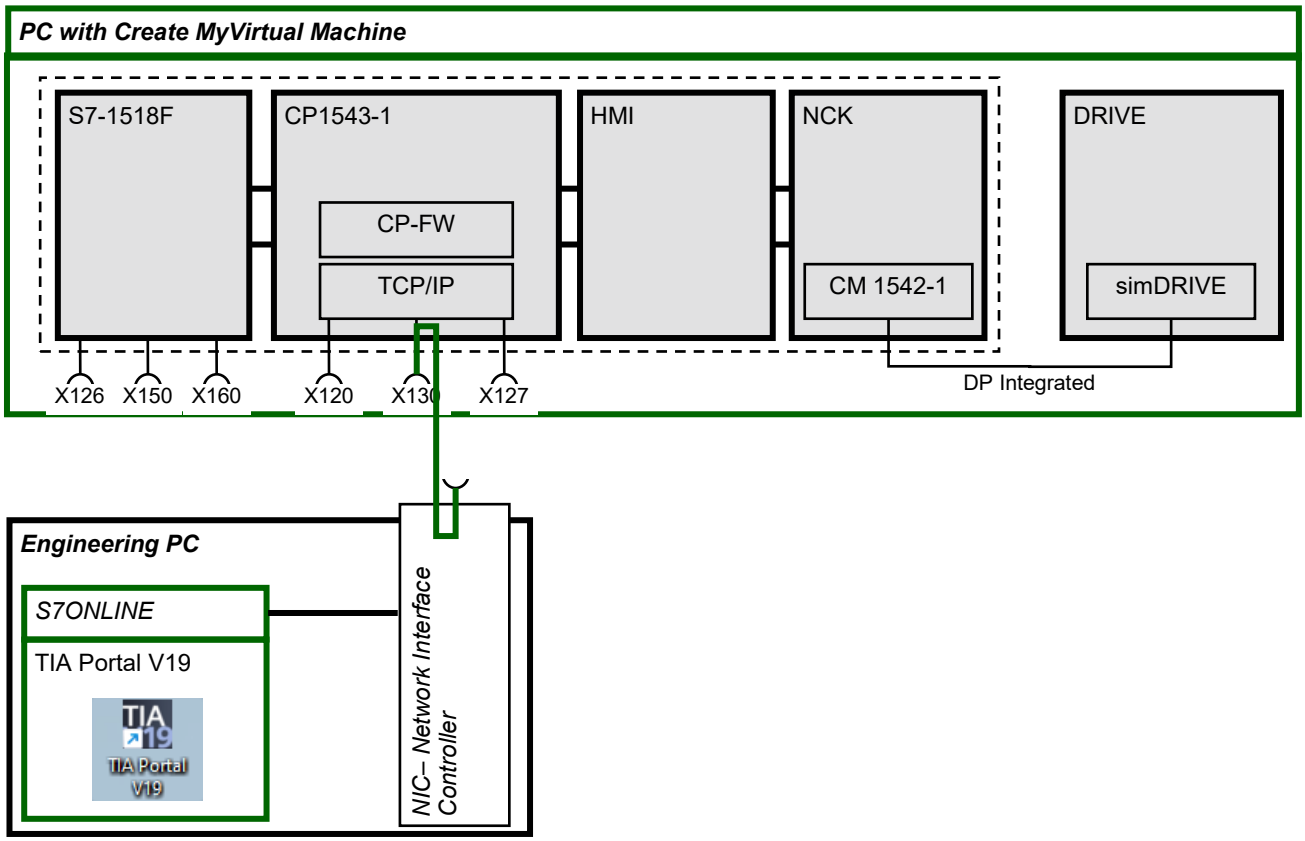
When TIA Portal and Create MyVirtual Machine are installed on different PC's, the TCP Communication has to be set accordingly.

On the PC with the installed Create MyVirtual Machine the firewall needs an open Port 102 for incoming connections. This is required for downloading and online connections for TIA Portal Step7.

For this procedure you will need administrator rights on your PC. The sequence for enabling Port 102 in the windows firewall is explained on the following pages.

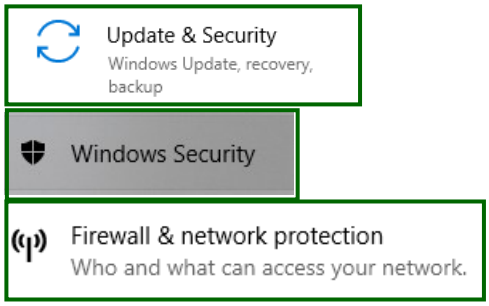


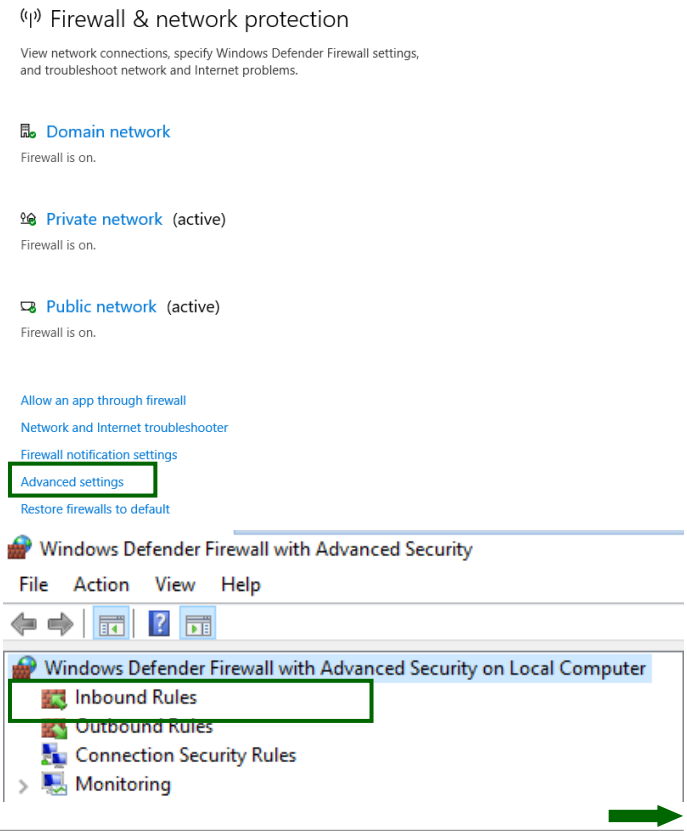
Notes:



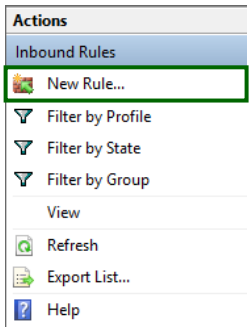
Apply advanced Settings to the windows firewall

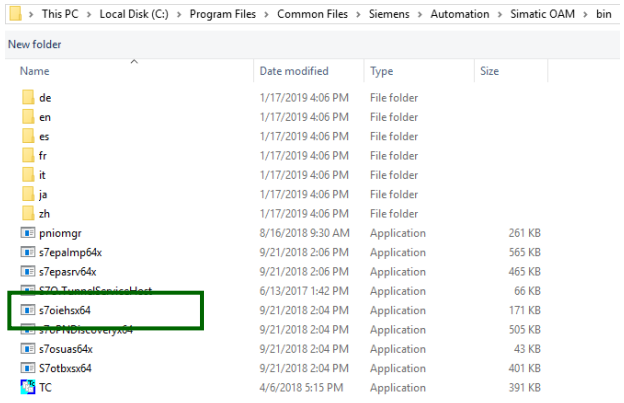
1. Go to Windows Settings
2. Select „Update & Security“
3. Select „Windows Security“
4. Select „Firewall & network protection“
5. Select „Advanced settings“
6. After the firewall has opened select „Inbound Rules“
7. Then select „New Rule“ and „Program“
8. Select the Programs: „%ProgramFiles%\Common Files\Siemens\Automation\Simatic OAM\bin\s7oiehsx64.exe“
9. „Allow the connection“

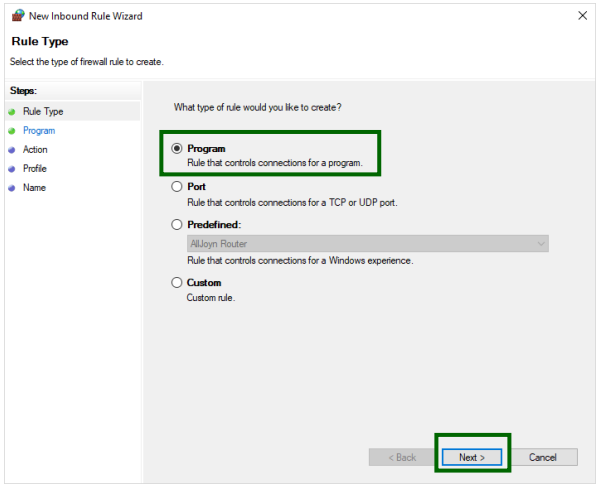


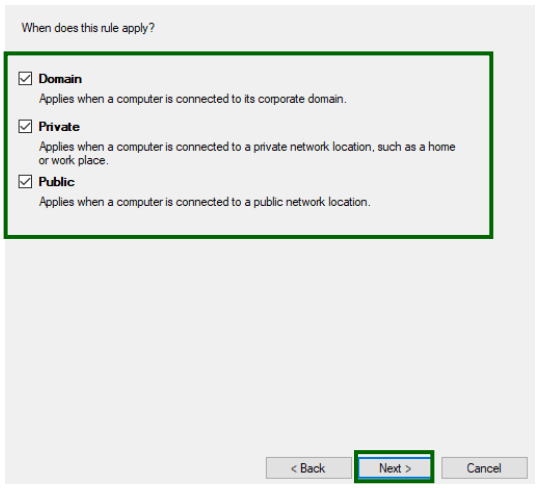


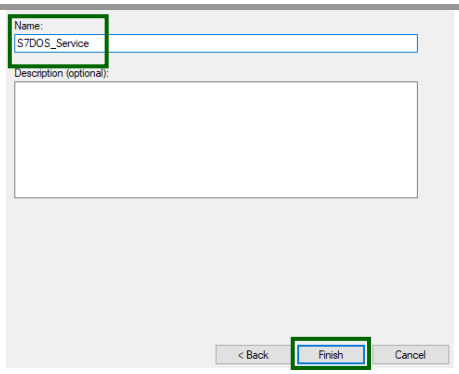
Notes:







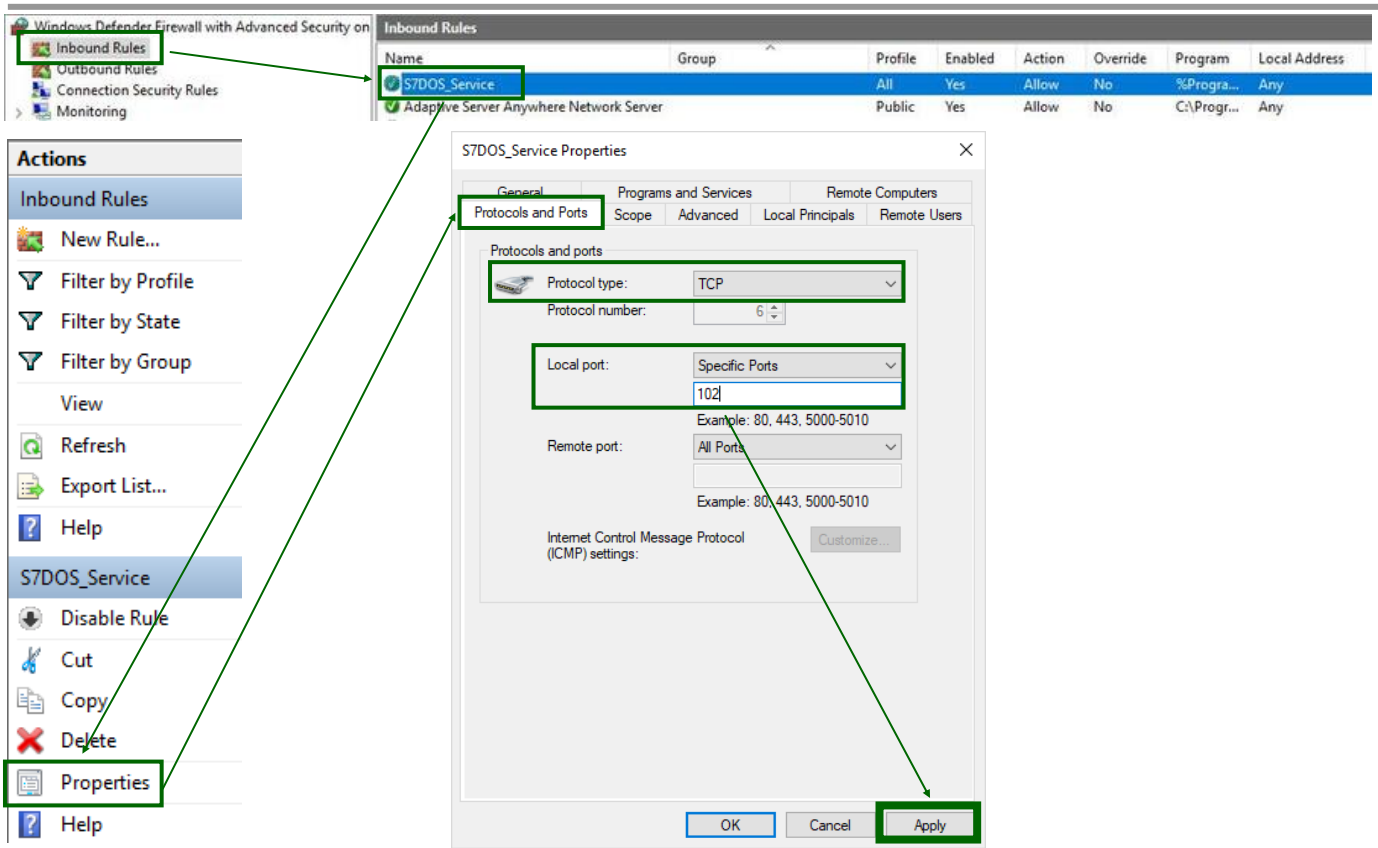


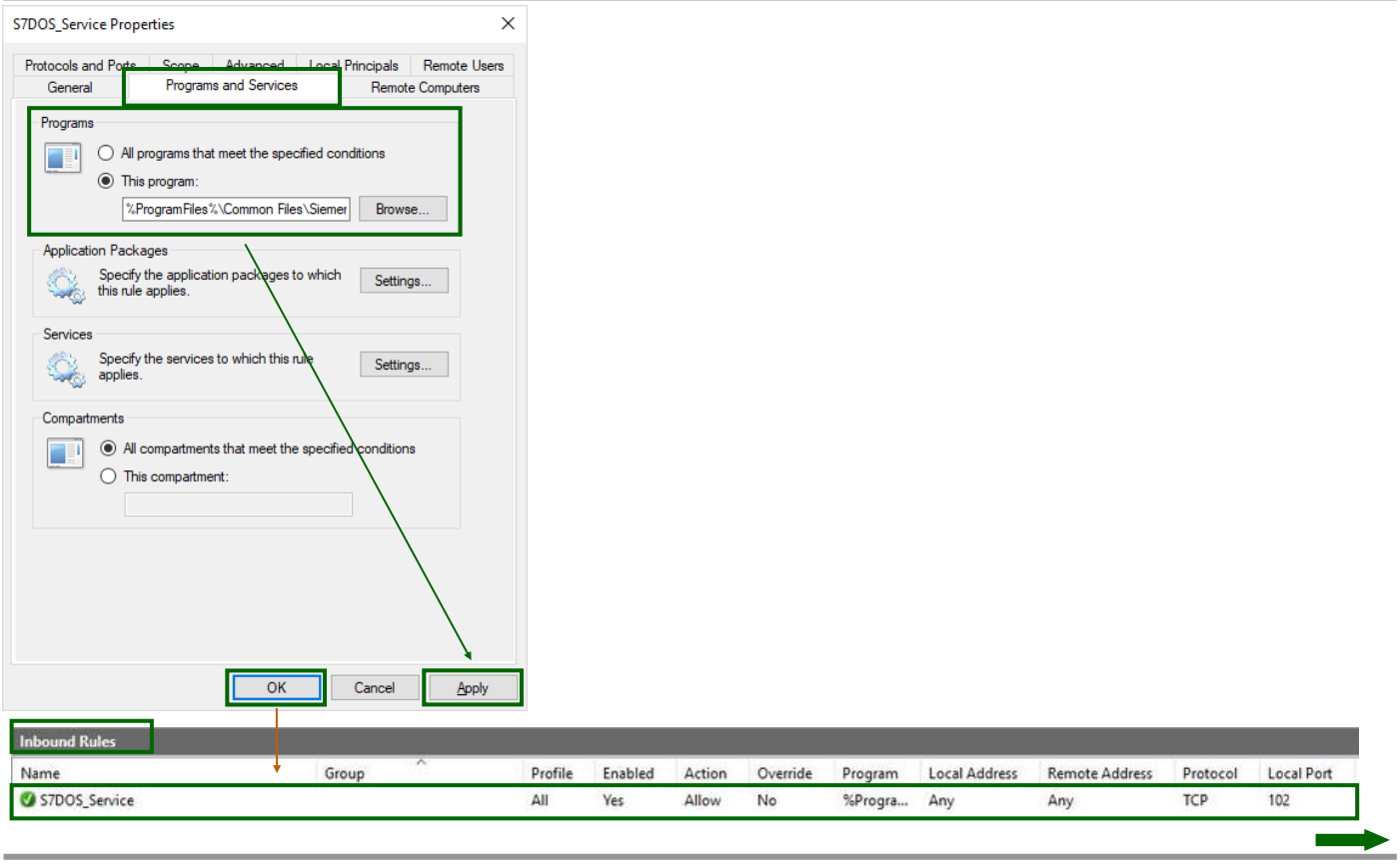


Continue with:

1. „When does the rule apply?“ Select all
2. Assign a name for the rule e.g. „S7DOS_Service“
3. Click on Finish
4. Go to „Inbound Rules“ and select „S7DOS_Service“
5. Select „Properties“
6. Go to „Protocol and Ports“
7. Set „Protocol type“ to „TCP“, „Local Port“ to „Specific Ports“ and type 102
8. Check „Programs and Services“ for correct settings
9. Click „Apply“ and „OK“ to store the settings
10. Check the final settings in the „Inbound Rules“ Screen

Notes:


Windows Defender Firewall with Advanced Security on
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring
Inbound Rules
Name
Group
Profile
Enabled
Action
Override
Program
Local Address
S7DOS_Service
All
Yes
Allow
No
%Progra...
Any
Adaptive Server Anywhere Network Server
Public
Yes
Allow
No
C:\Progr...
Any
Actions
Inbound Rules
New Rule...
Filter by Profile
Filter by State
Filter by Group
View
Refresh
Export List...
Help
S7DOS_Service
Disable Rule
Cut
Copy
Delete
Properties
Help
S7DOS_Service Properties
General
Programs and Services
Remote Computers
Protocols and Ports
Scope
Advanced
Local Principals
Remote Users
Protocols and ports
Protocol type:
TCP
Protocol number:
6
Local port:
Specific Ports
102
Example: 80, 443, 5000-5010
Remote port:
All Ports
Example: 80, 443, 5000-5010
Internet Control Message Protocol (ICMP) settings:
Customize...
OK
Cancel
Apply


S7DOS_Service Properties
Protocols and Ports
Scope
Advanced
Local Principals
Remote Users
General
Programs and Services
Remote Computers
Programs
All programs that meet the specified conditions
This program:
%ProgramFiles%\Common Files\Siemer
Browse...
Application Packages
Specify the application packages to which this rule applies.
Settings...
Services
Specify the services to which this rule applies.
Settings...
Compartments
All compartments that meet the specified conditions
This compartment:
OK
Cancel
Apply
Inbound Rules
Name
Group
Profile
Enabled
Action
Override
Program
Local Address
Remote Address
Protocol
Local Port
S7DOS_Service
All
Yes
Allow
No
%Progra...
Any
Any
TCP
102

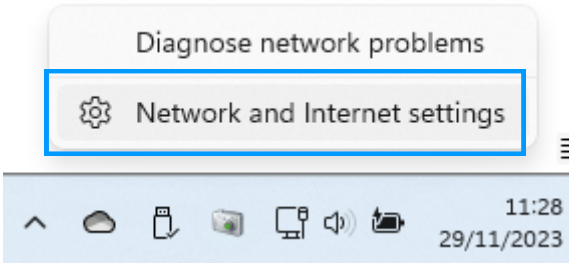
Notes:

Network Settings

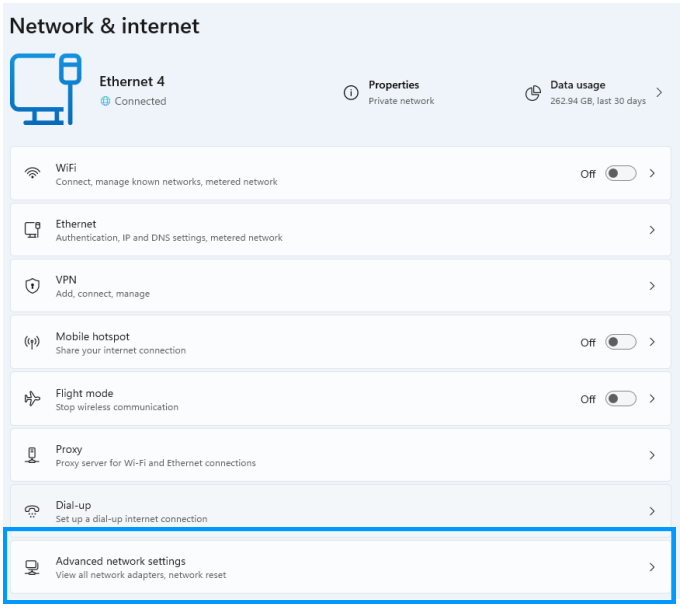
X127 is a DHCP server with a fixed IP address of 192.168.215.1 and subnet mask of 255.255.255.224
IP addresses from 192.168.215.2 to 192.168.215.31 are automatically issued to any PC connected to X127.
The name of the network adapter of the PC is required.
This can be found in the “Network and Sharing Center” area of the computer:



Right click the network icon.



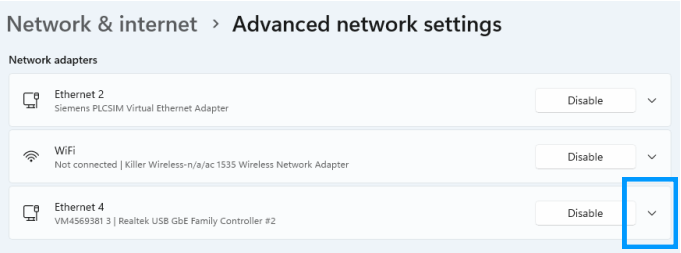
Select ”Network and Internet settings”



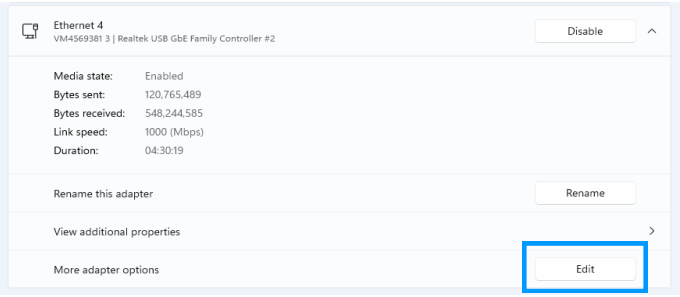
Select ”Advanced network settings”

Note the name of the network adapter associated with the Local Area Connection.
In the **example** below the network adapter is of the type: **Realtek USB GbE Family Controller #2**.

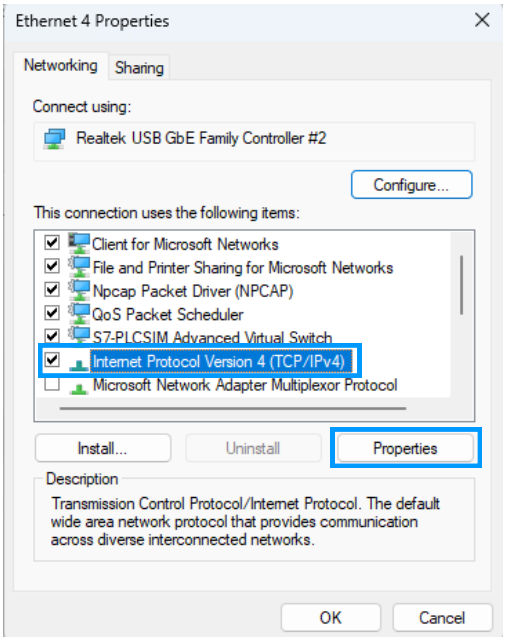
Note: Network adapter may be different between PC makes and types.



Click the expand icon, then on the “Edit” icon:



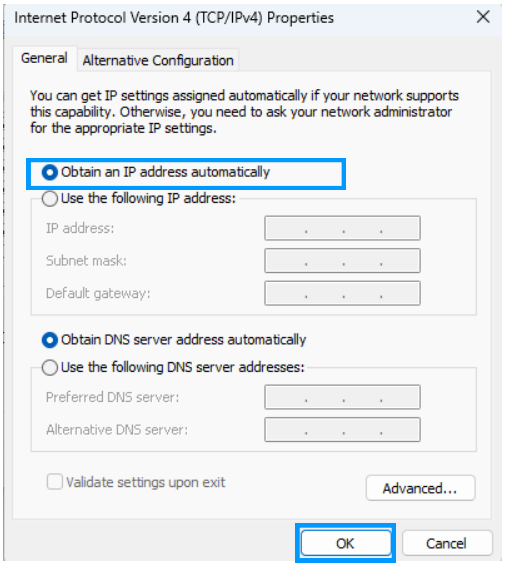
Select the option “**Internet Protocol Version 4 (TCP/IPv4)**” followed by the “Properties” button:



Properties

Notes:

Select the “**Obtain an IP address automatically**” option.

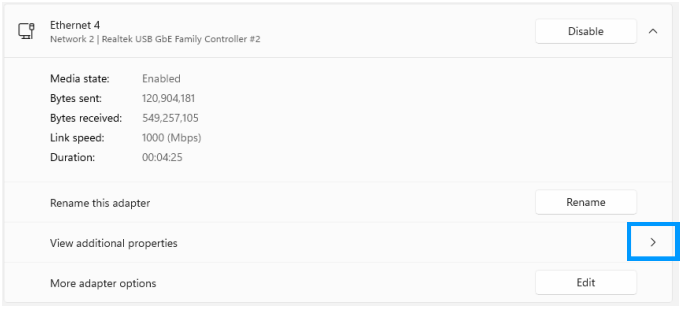


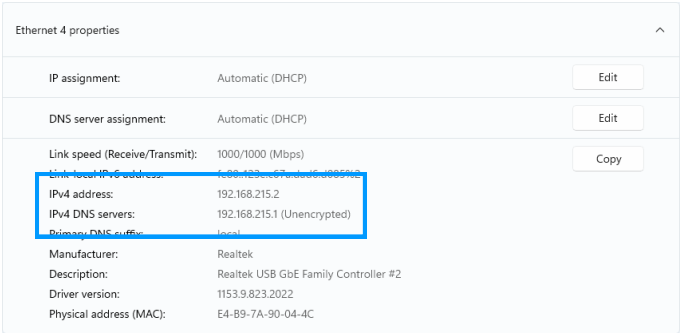
OK

The PC is now set to receive an IP address from a DCHP server, i.e. X127 of the Sinumerik ONE Controller.

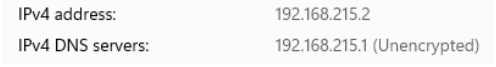
Close

Connect a suitable Ethernet cable between the PC and X127 of the controller. In some situations this may need to be a cross-over cable.
The connection status can be checked via the “View additional properties” option of the Ethernet connection.





The PC has been assigned the first available IP address of 192.168.215.2 by the DHCP server of X127.

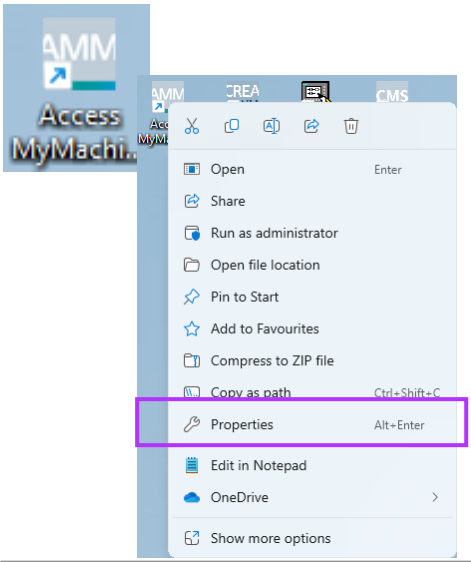


Notes:

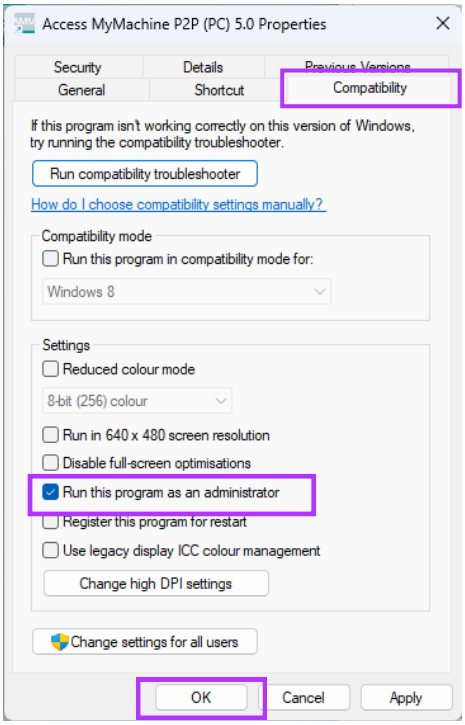
Online Connection-Access MyMachine

Administrator Privilege Level
The Toolbox programs should always be run with privileges set to Administrator.

Setting the Administrator Privilege:
Right click on the icon of the Access MyMachine program and select “Properties“.

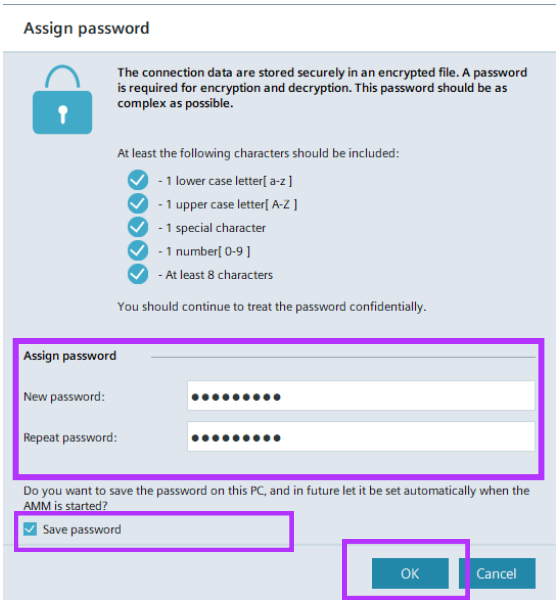


Set the privilege level to “Run this program as an administrator“





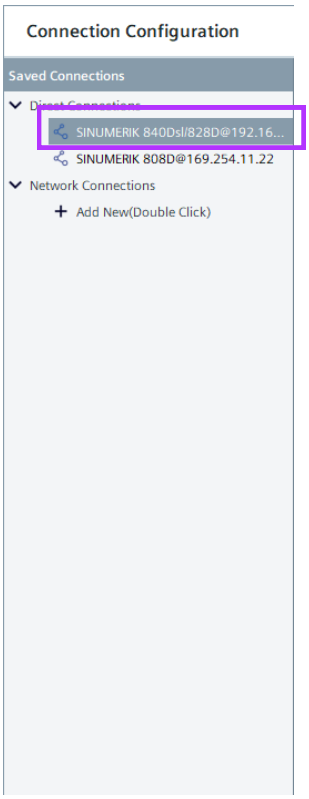
When the program is opened for the first time after the installation, it is necessary to assign a password for encryption purposes.



Make a connection by selecting the “icon” shown below.

A dialogue box appears where the choice of connection is made.
A default connection is available, this is specifically for connecting to **X127** of the controller (IP address **192.168.215.1**).

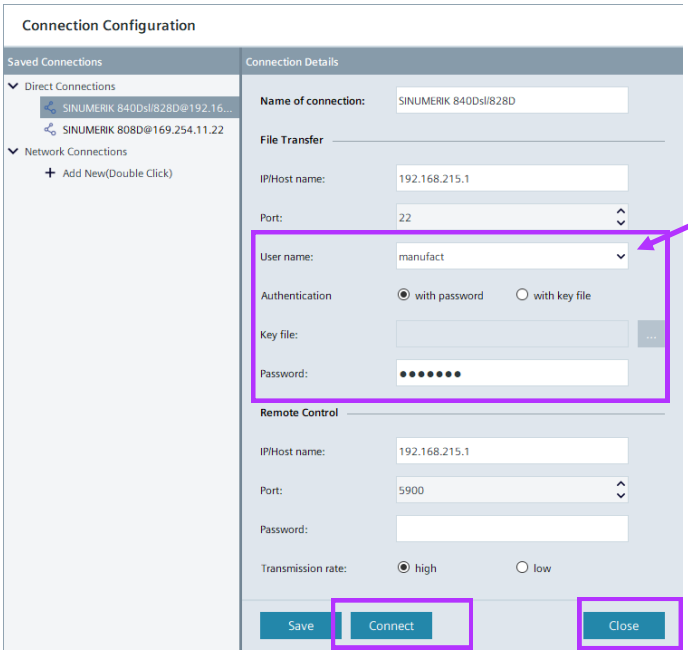
Notes:



It is necessary to login with a password depending on the access level required:

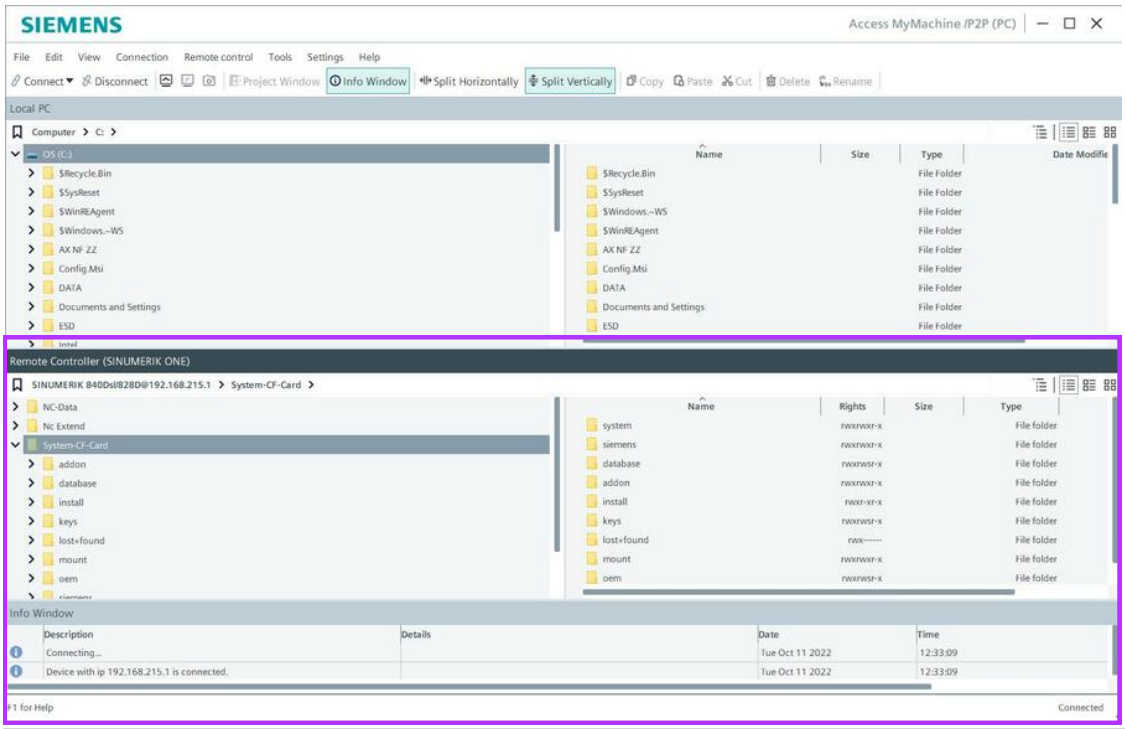
The access level determines which data can be accessed via Access MyMachine.
The “Manufacturer” login grants the most access, the “User” login is the most restrictive.
Select the login and enter the password.

Passwords:
Manufacturer-SUNRISE
Service-EVENING
User-CUSTOMER



Notes:

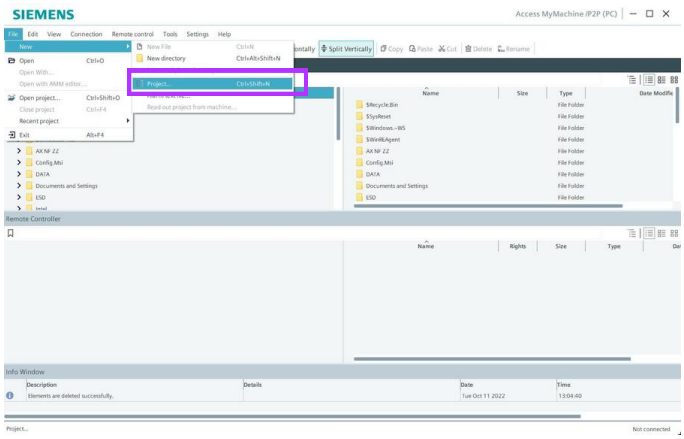
After a short delay the data folders of the controller will appear in the lower half of the Access MyMachine program, this indicates that a successful connection was made.



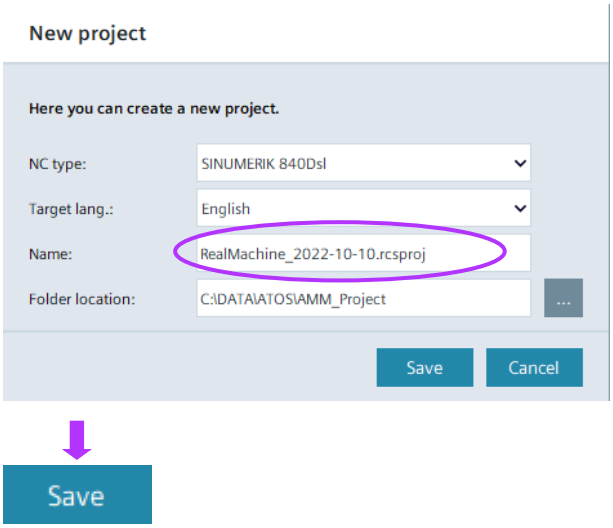
Creating a "Project" within Access MyMachine

Start the Access MyMachine program.

Under the pull down menus:
File --> New --> Project……..

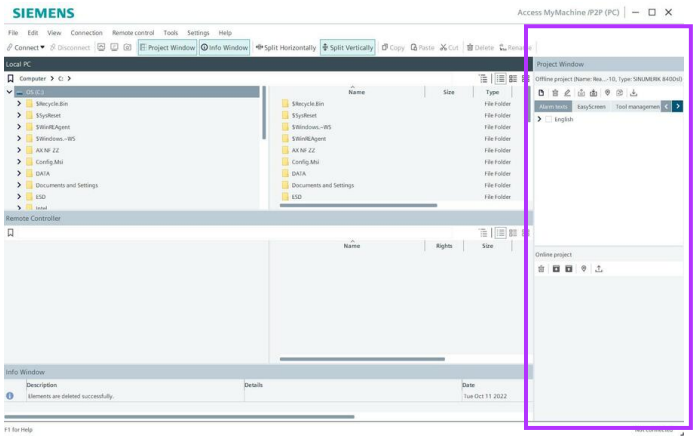


Enter a name to identify the project.



Notes:

The Project window will appear to the right-hand side.



Loading files directly to the control system and from the control system to the PC.
Managing the control system's NC data. You can copy files directly from the PC to the NC and from the NC to the PC.
Monitoring processes and remote control of the SINUMERIK.
Saving the screenshot of the HMI on the PC.
Editing the following user files on the PC:
- PLC alarm texts (oem_alarms_plc)
- Cycle alarm texts (oem_alarms_cycles)
- Part program messages (oem_partprogram_)
- EasyScreen language strings
- Tool management language strings

Creating and loading an archive for the control system.
Managing user files with the following options:
Creating a project OFFLINE.
Copying individual files using the project dialog to the control system or copying from the control system to the project.
Deleting user files using the project dialog on the control system.

Once a project has been created, AMM provides the following functions:

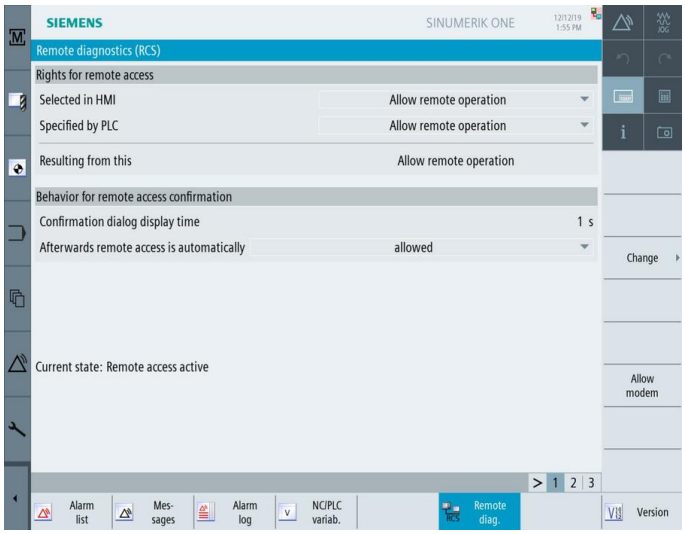
Viewer

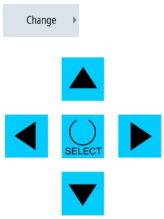
The external viewer duplicates the controller display on the external computer including menu and soft-key functionality. When the viewer connects, access can be restricted on the HMI or by the PLC interface.

To configure the HMI use the following soft-key sequence:









Viewing Rights-HMI
Rights for remote access - Selected in HMI
Do not allow remote access-Prevents remote connections.
Allow remote monitoring-The remote computer can monitor the controller.
Allow remote operation-The remote computer can access the various screens of the controller and changes can be performed.

Viewing Rights-PLC
Rights for remote access, Specified by PLC (Not settable on screen)
The access can be restricted/blocked via the user PLC with interface signals. These signals have priority over the "Selected in HMI" settings. It may be necessary to Contact the machine manufacturer if the PLC is restricting the viewer access.

Notes:

Behaviour for remote access:

Confirmation dialog display time

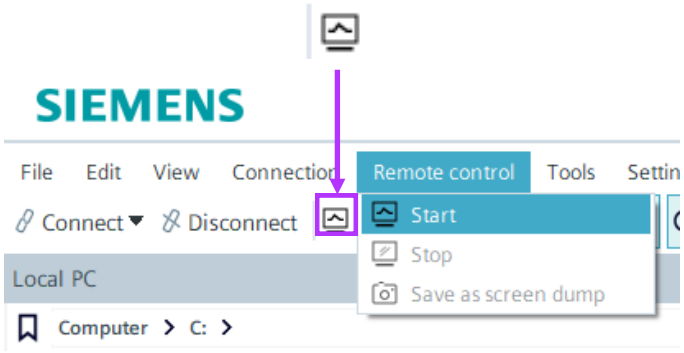
When a connection attempt is made a dialogue box will appear which requires acknowledgement, this determines how long the dialogue box stays on the screen.

Afterwards remote access is automatically

When set to “allowed”, then the controller will allow the connection, even when the dialogue box has not been acknowledged.
If set to “Denied” the connection will be blocked if the dialogue box is not acknowledged.

The settings are saved by selecting the “OK” soft-key.

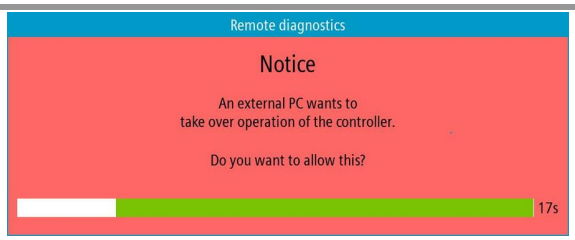




Connecting the Remote Viewer

To start this function select the following icon from the Access MyMachine tool bar.
Alternatively select “Start” from the “Remote control” drop down menu.

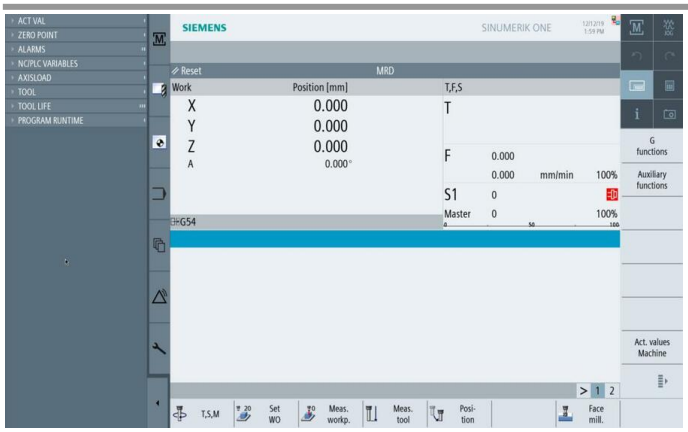
When the viewer attempts to connect, providing that the remote viewer has not been blocked, the screen of the controller will be displayed.
If an acknowledgement is required, a dialogue box will appear. The person at the controller should acknowledge with the “Yes” soft-key.
This should be done within the time limit specified on the HMI-set-up screen.





When the connection is successful the viewer will display the same screen as the controller.

Notes:



The following keys are used to navigate the controller from the remote PG/PC:

F1 to F8 - Horizontal soft-keys
Shift & F1 to F8 - Vertical soft-keys
F9 - Page Back/Up
F10 - Menu Select
Shift & F9 - Page right
F12 - Help function
Escape - Alarm Cancel
Select - Space Bar

Programs, machine data etc can be directly changed from the remote computer, archives can also be created.

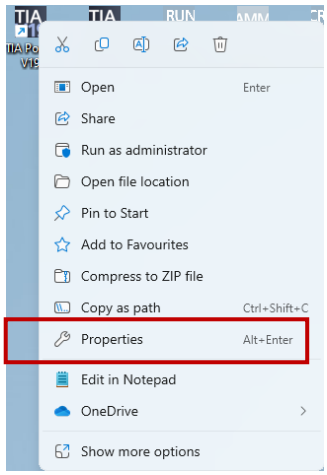
Online Connection: TiA Portal



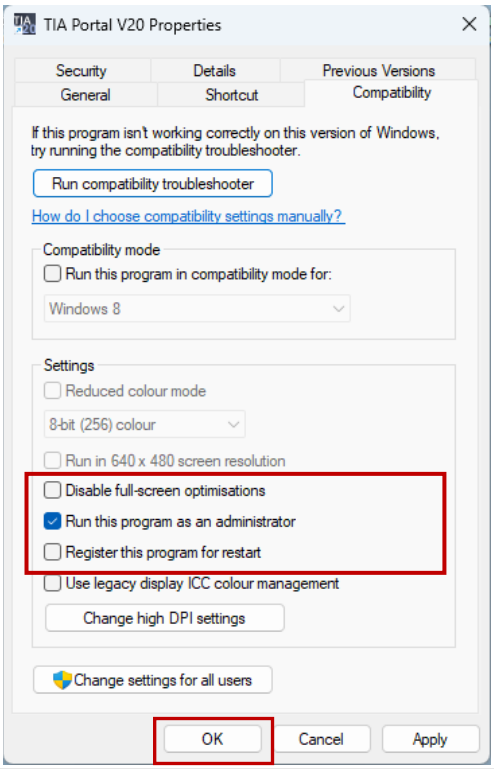
A TiA Portal shortcut icon is placed on the PC desktop during the installation process.

Administrator Privilege Level
The service/commissioning programs should always be run with privileges set to Administrator.

Setting the Administrator Privilege:
Right click on the icon of the **SIMATIC Manager** program and select “Properties“



Set the privilege level to “**Run the program as administrator**“



Notes:

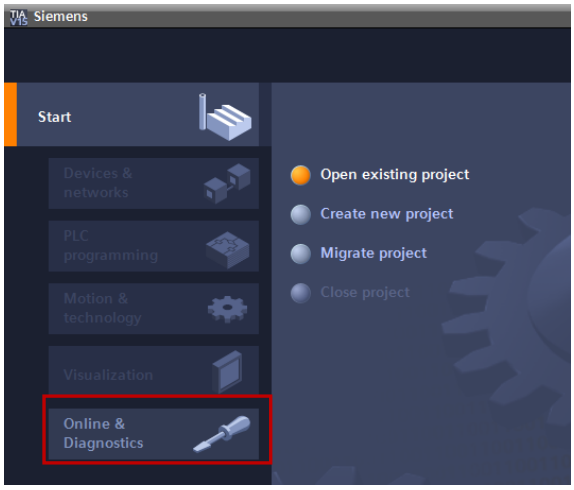
Setting the PG/PC Interface:

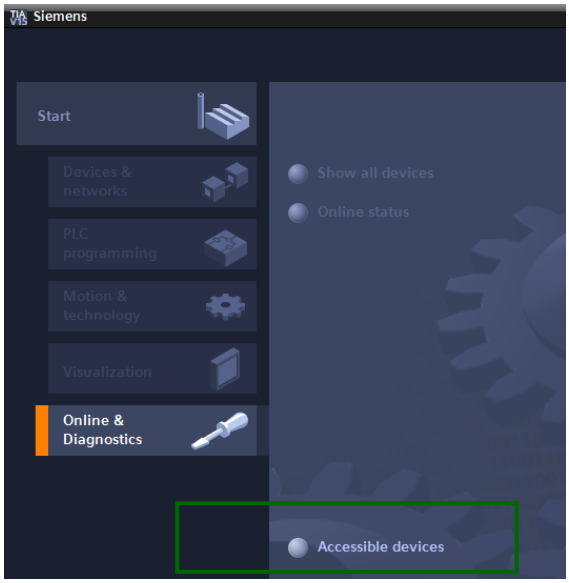
To allow the program to communicate via Ethernet it is necessary to set the PG/PC interface.



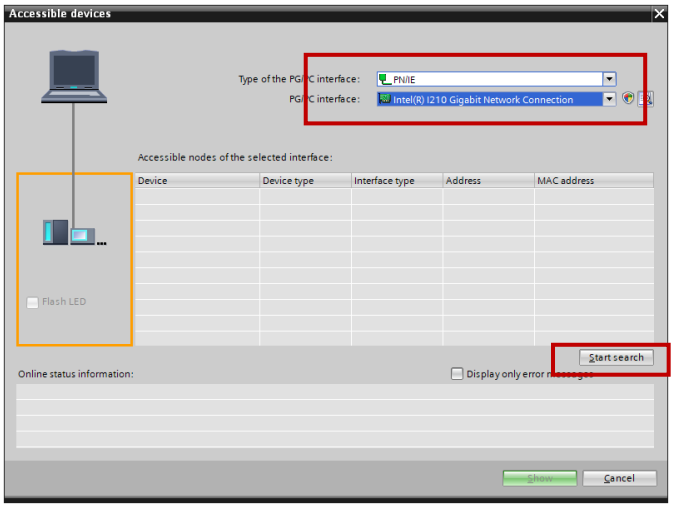
Start the TiA Portal program

The setting for the connection can now be changed, select “Online & Diagnostics” Interface.





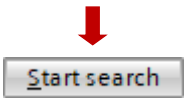
The Ethernet adapter of the computer needs to be selected from the list.
In the **example** below, the network adapter is of the type:
Intel® 1210 Gigabit Network.



Checking the online connection:

Ensure that a suitable Ethernet cable is connected between the PC and X127 of the controller.

To check the connection select “Start Search” in the window

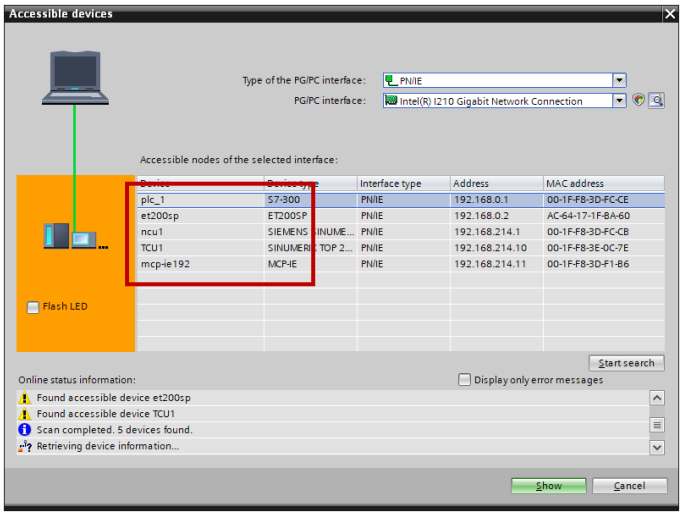


After a short delay the available hardware will be displayed. Accessible nodes are the hardware components which are connected to X120. The internal routing of the controller allows these to be accessed via X127
Seeing the window below shows that the connection to the controller has been established.

Notes:

The accessible hardware in this case are:

PLC (plc_1)
ET200SP (et200sp)
Numerical Control Unit (ncu1)
Thin Client Unit (TCU1)
Machine Control Panel (mcp-ie192)



A101: END

Notes:

A103

SINUMERIK ONE

Create MyVirtual Machine

Module Description:

This modules describes the usage of the Software „Create MyVirtual Machine. This Software is the Virtual SINUMERIK ONE. In Combination with TIA Portal V20 and the SINUMERIK ONE Toolbox the commissioning sequence of a SINUMERIK can be done completely virtual..

Module objective:

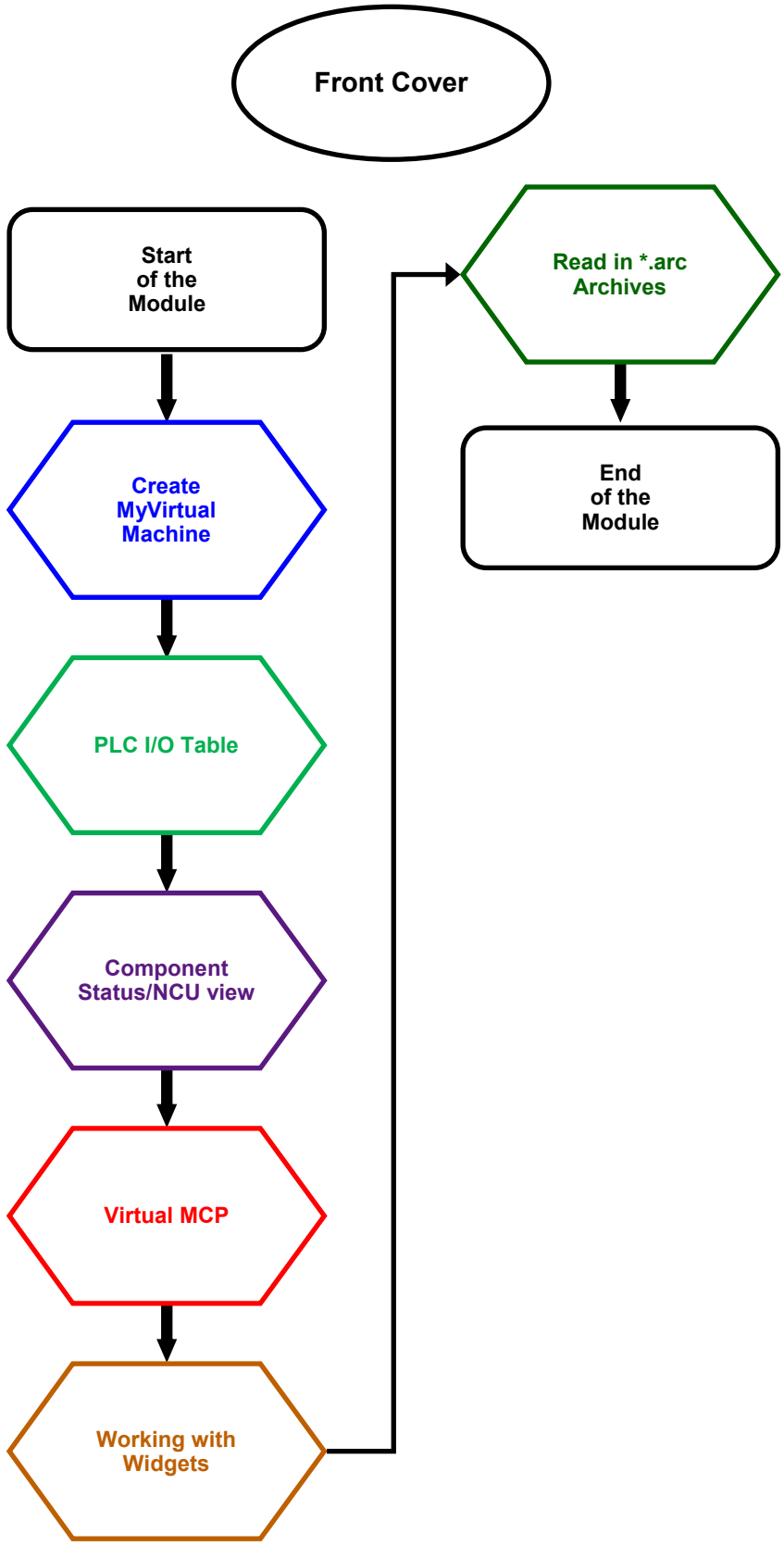
At the end of this module, you will be familiar with the functions of the Tool Create MyVirtual Machine. You will be able to create and modify a project and know how to import and export the *.vcp files. You know the handling of archives and the usage of the virtual SD-Card.
Also you learn how to manipulate the virtual machines status by using the I/O-tables.

Content:

- Create a new Project, manage Projects
- PLC I/O table
- Component Status/NCU view
- Virtual MCP
- Working with Widgets
- Read in *.arc Archives

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

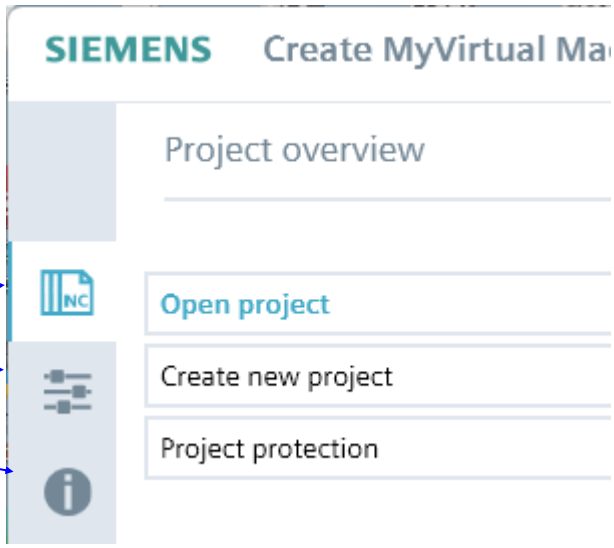

Front Cover
Start of the Module
Create MyVirtual Machine
PLC I/O Table
Component Status/NCU view
Virtual MCP
Working with Widgets
Read in *.arc Archives
End of the Module

Create MyVirtual Machine

The Software can be started by double clicking on the desktop icon.

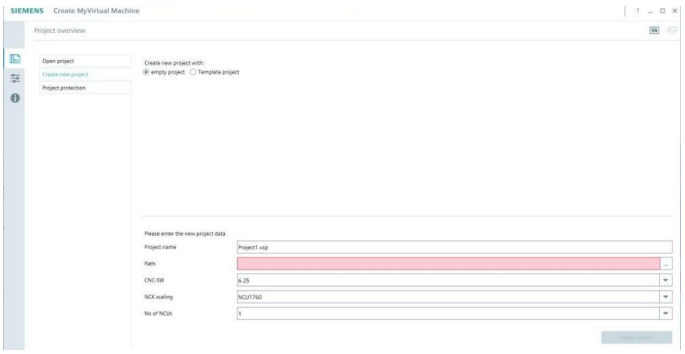


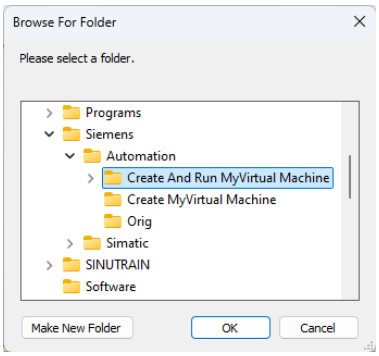
With the Main Menu you can do the following:

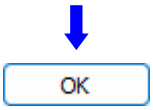


1. **Manage Projects**
2. **Change Settings**
3. **View Application Information**

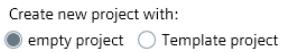
When you create a new project, you have to specify the name and the folder for storing and also the software version.
After clicking on “Create Project“ the project window is opened.









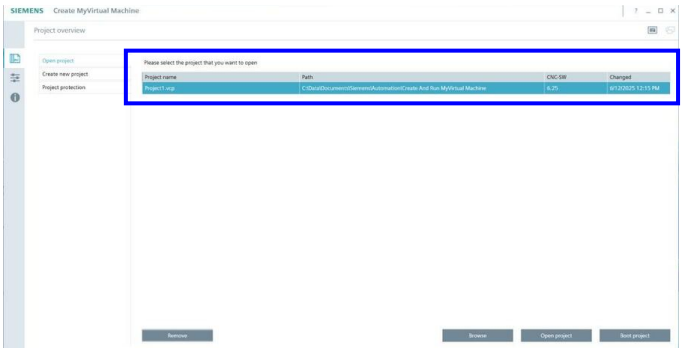


Notes:

Managing projects: Open an existing project

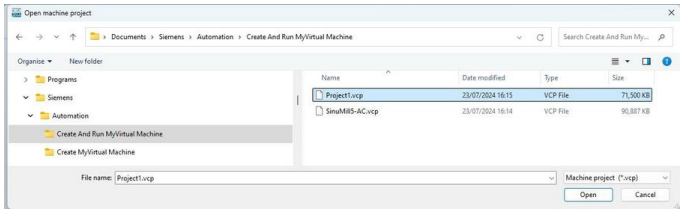
You can either select a project listed in the table or open a project from a folder on your PC.

The Project booting must be selected manually.



Open project

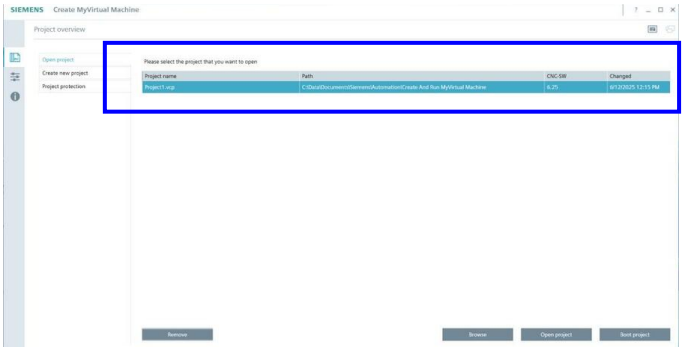
Browse



Open

You can also directly open and automatically Boot an existing project.

This project will power up as a switched on machine.

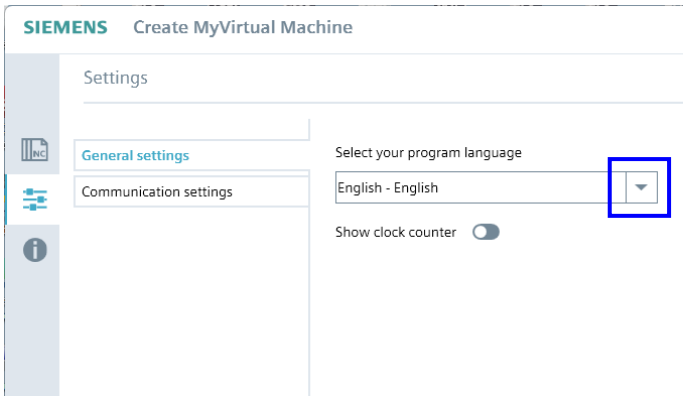


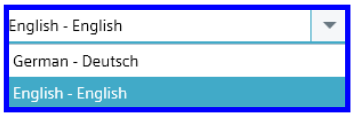
Boot project

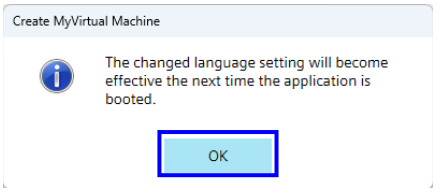
Notes:

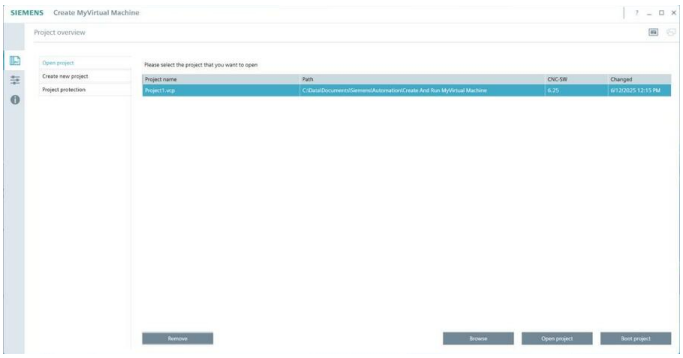
Changing Settings: CMVM Settings

You can select the language. German and English are supported.



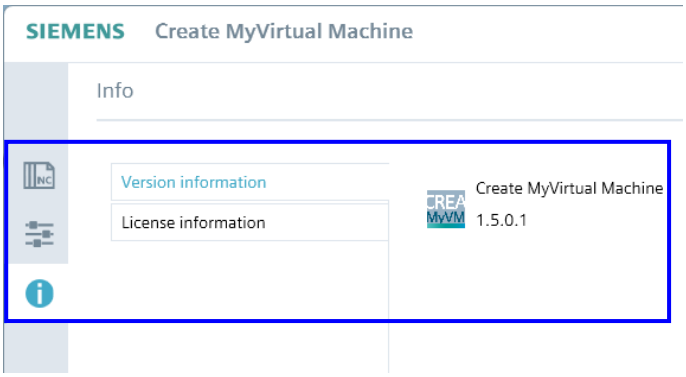






Viewing application information: CMVM Info

The Info tab displays the actual version of Create MyVirtual machine.



Closing the application:

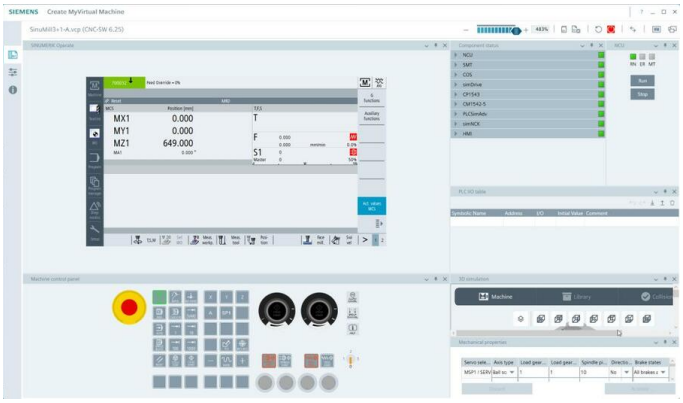
By Clicking on the Close Symbol, Create MyVirtual Machine is switched off immediatly.



Notes:

The project in simulation view

If you push the button create project the CMVM Software will start the project in a simulation view.
The Simulated machine is displayed in a switched off state.



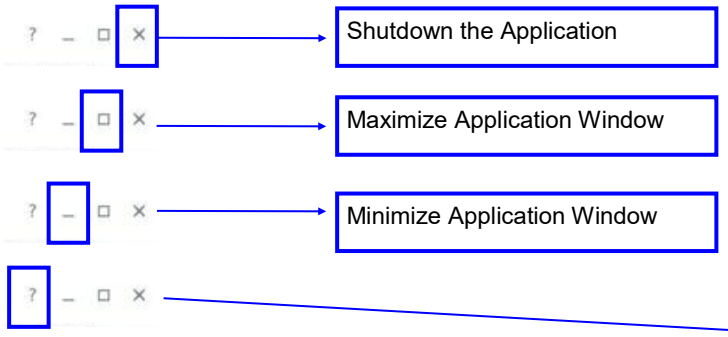
When you push the green button the simulated SINUMERIK ONE is switched on.
The run up of the controller is visible on the right hand side of the application. All components will change one after the other from a grey coloured LED to a green coloured LED:

After the run up is completed, the HMI and also the machine control panel become visible.

By pressing the red button the simulation controller can be Turned Off.

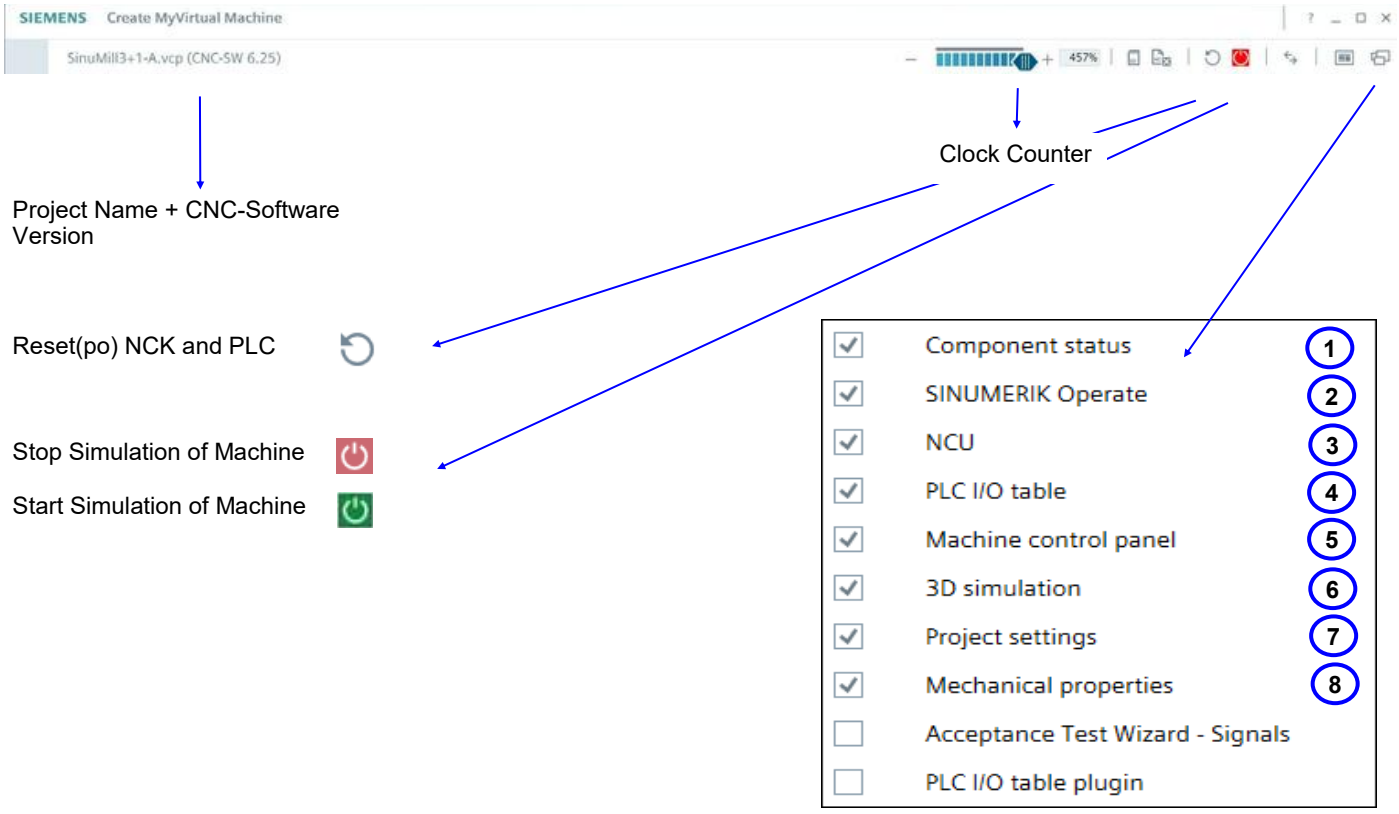
Create MyVirtual Machine Menu bar

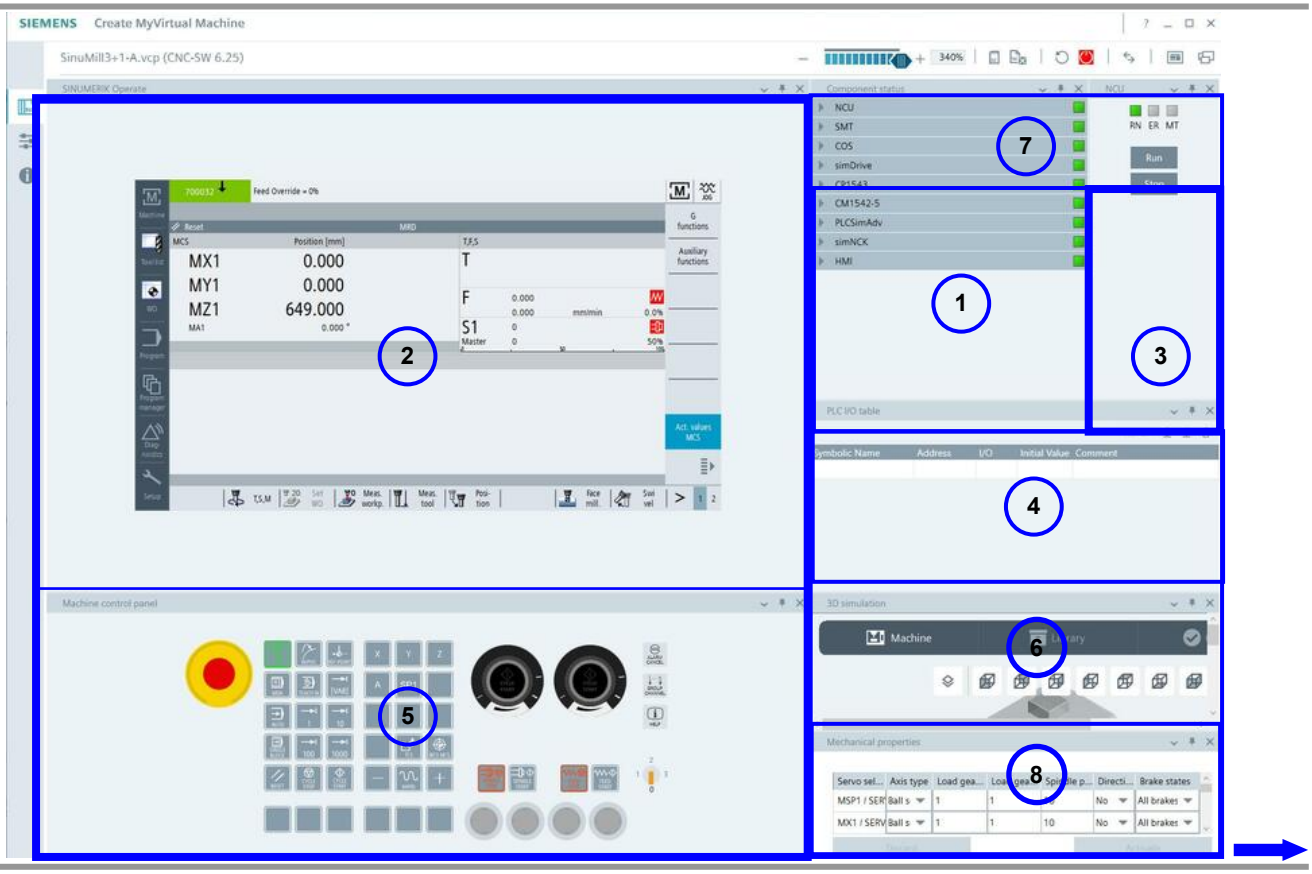






Notes:

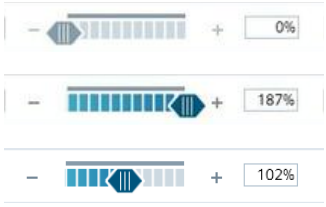

SIEMENS Create MyVirtual Machine
SinuMill3+1-A.vcp (CNC-SW 6.25)
457%
Clock Counter
Project Name + CNC-Software Version
Reset(po) NCK and PLC
Stop Simulation of Machine
Start Simulation of Machine
Component status
SINUMERIK Operate
NCU
PLC I/O table
Machine control panel
3D simulation
Project settings
Mechanical properties
Acceptance Test Wizard - Signals
PLC I/O table plugin
1
2
3
4
5
6
7
8


SIEMENS Create MyVirtual Machine
SinuMill3+1-A.vcp (CNC-SW 6.25)
340%
MX1 0.000
MY1 0.000
MZ1 649.000
1
2
3
4
5
6
7
8

Notes:

Manage the Machine Simulation

Speed Settings:
The Machines Simulation Speed can be modified between Pause System to maximum Speed. The Bar can have 3 Positions, Left Middle and Right.



Pause, System is stopped

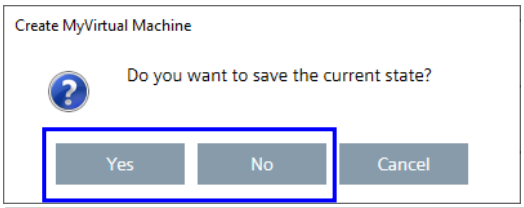
Maximum Speed, limited by the Host Hardware

Approximately Real Machines Speed

Close the actual Simulation and return to the Main Menu:

With this icon the simulation can closed. If needed the changes should be saved.

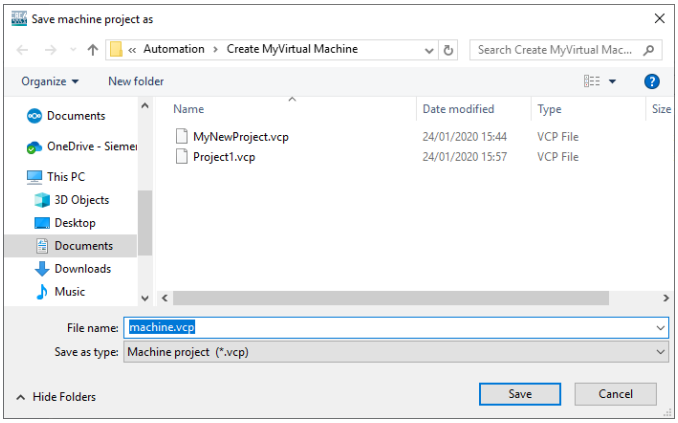




Save the actual Simulation:

The Machine Simulation can only be saved, if it is switched off. The Machine Simulation can be saved in its destination folder(**Save**) or it can be saved under a different location (**Save As**).





The virtual SINUMERIK ONE has a **virtual SD-Card** for data and program handling.

This virtual SD-Card is created temporarily on each startup of a machine project in the Windows User Folder. Each machine project has its own virtual SD-Card.

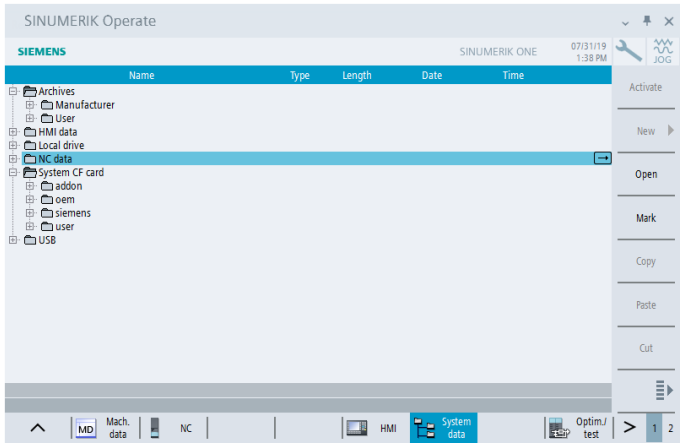
After closing a machine project, the virtual SD-Card content and structure is stored in the project. The temporary Windows User folder will be **deleted.**

On the next run up of a machine project the data will be copied again to the temporary Windows User folder. The structure and handling of the virtual SD-Card is similar to the real SD-Card of a SINUMERIK ONE.

The temporary Windows User Folder is the following and can be opened from the menu(„auduser“ must be replaced by your user name):
„C:\Users\auduser\AppData\Local\Siemens\Automation\SINUMERIK ONE\ncu\card“





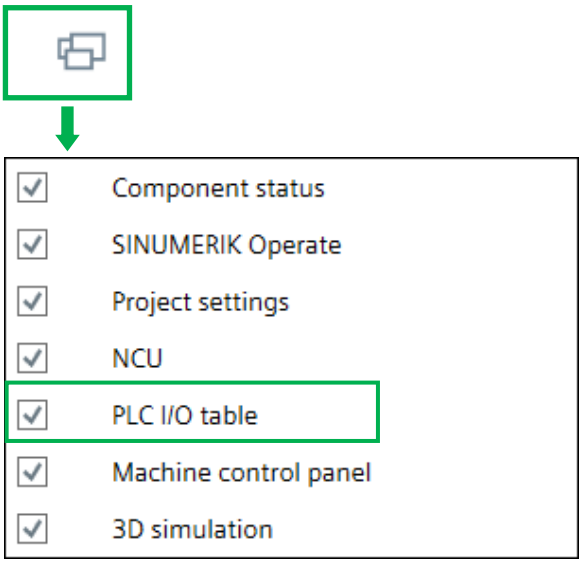


Notes:

PLC I/O Table

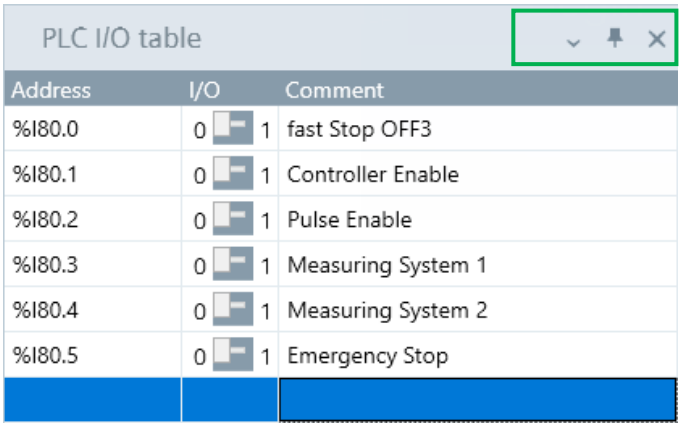
The PLC I/O Table can be switched ON/OFF from the Menu Bar Icon.
It can also be switched OFF by the x in the right upper corner.

It is also possible to dock/undock and minimize the PLC I/O Table Widget by the pin Icon.



The purpose of the PLC I/O Table is the possibility for the CMVM Operator to manipulate the S7-1500 PLC of the virtual SINUMERIK ONE. This is essential during the virtual commissioning phase to simulate non existing machine equipment as far as possible.

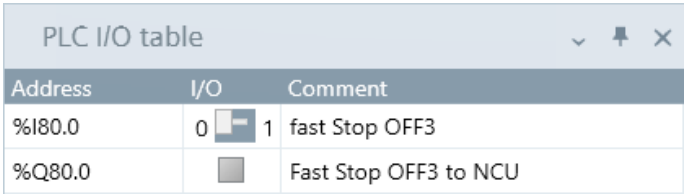
On the first boot up of a new CMVM project, the PLC I/O Table is empty. It can be used with PLC Inputs and Outputs only!



Address	I/O	Comment
%I80.0	0 1	fast Stop OFF3
%I80.1	0 1	Controller Enable
%I80.2	0 1	Pulse Enable
%I80.3	0 1	Measuring System 1
%I80.4	0 1	Measuring System 2
%I80.5	0 1	Emergency Stop

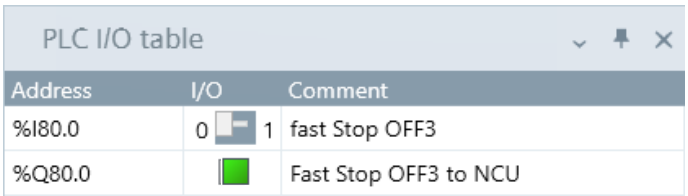
The I/Os must be written in the correct syntax to the PLC I/O Table as in the S7-1500PLC..

Each operand starts with a **%.**
Outputs are written with **%Qx.x**
Inputs are written with **%Ix.x**



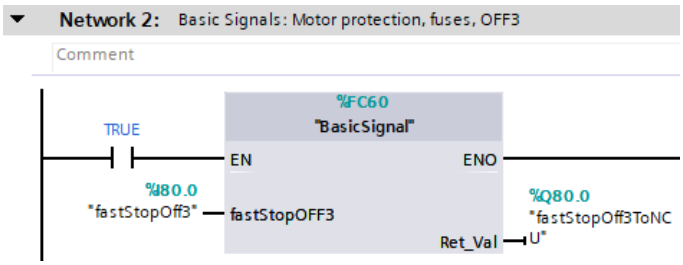
Address	I/O	Comment
%I80.0	0 1	fast Stop OFF3
%Q80.0		Fast Stop OFF3 to NCU

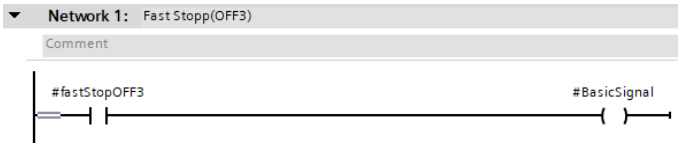
Comments are very useful to identify the I/O's usage. If an Output is ON, the LED's status is green. If its OFF, the LED's status is grey. Inputs are either 0 (FALSE) or 1 (TRUE) which is indicated by the switch position.



Address	I/O	Comment
%I80.0	0 1	fast Stop OFF3
%Q80.0		Fast Stop OFF3 to NCU

Here is the sample Logic behind the I/O's **%I80.0** and **%Q80.0**





Notes:

Component Status/NCU view

The **Component Status** can be switched ON/OFF from CMVM Menu. It can also be switched OFF by the x in the right upper corner.

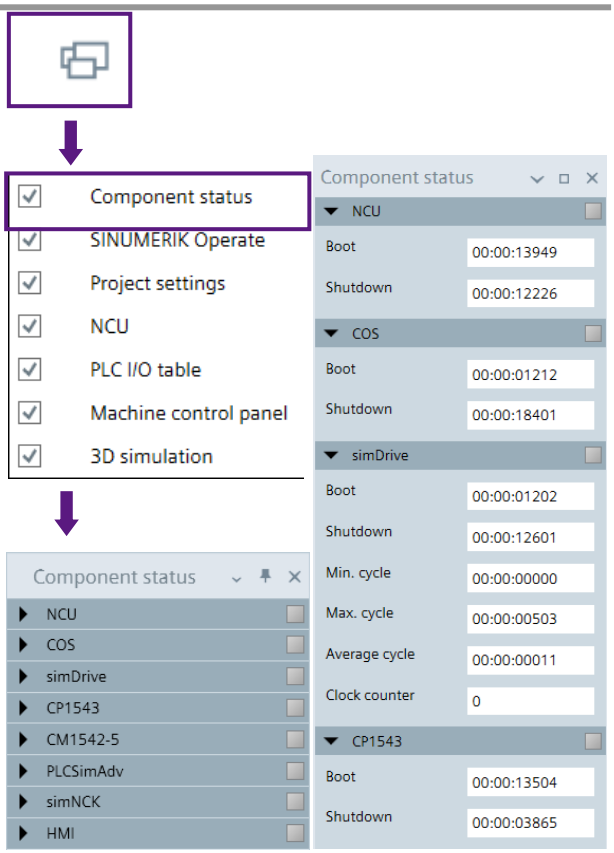
In the component status you see the status information for each Software component of the virtual SINUMERIK ONE.
The most important information is the Status LED on the right hand side of each component.

- Component started successfully (green)
- Component is starting (grey/green)
- Component is shutting down (green/grey)
- Component not started (grey)
- Faulty component (red)

Component faulty means the component did not reply in a certain waiting time (**timeout**) during start, cyclic operation or shutting down. In the component status display the time measurements are displayed.

For the cyclic components **simNCK**, **PLCSimAdv** and **simDrive** you can see the Boot and Shutdown time as well as the cycle time information. The cycle time information consists of minimum, maximum, average cyclic time and the cycle timer. This information is mainly used for debugging or R&D purposes.

The **NCU view** can be switched ON/OFF from CMVM Menu. It can also be switched OFF by the x in the right upper corner.



RN Run/Stop LED
- grey: PLC is switched OFF
- green: PLC is in Run
- yellow: PLC is in Stop
- 2 colored blinking: PLC changes mode

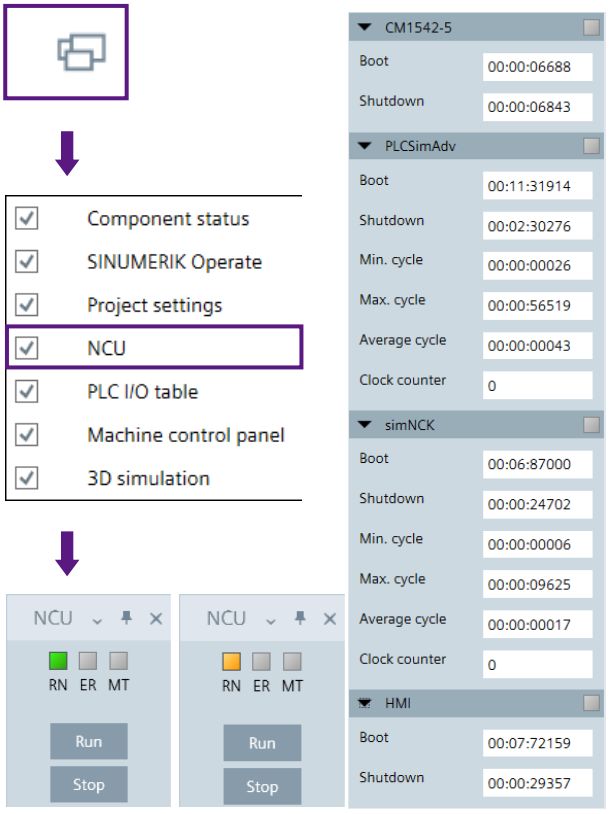
ER Error LED
- grey: No diagnosis event or PLC is switched OFF
- red blinking: diagnosis event

MT Maintenance LED
- grey: No maintenance request or PLC is switched OFF
- yellow: There is a maintenance request
- Yellow blinking: There is a maintenance demand

The PLC Mode selector switch is a simulation specific Addon for changing the actual mode of the PLC. The functionality is similar to the „CPU operator panel - Mode selector“ of the PLC device in TIA Portal STEP 7.

RUN
PLC-Mode RUN is requested. The PLC program is processed
STOP
PLC-Mode STOP is requested. The processed PLC program is stoped and all PLC Outputs will be set to default

Remark:
After changing a running PLC to STOP Mode, you can switch it back to RUN Mode. A NCK Power On Reset is neccessary to synchronize the components NCK and PLC again!



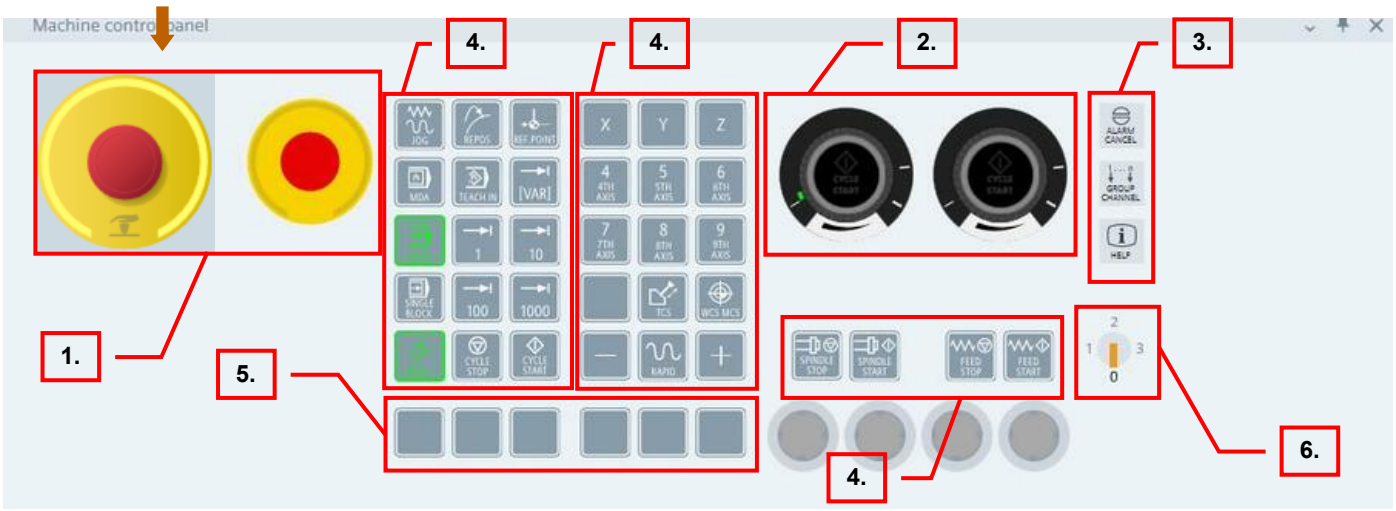
Notes:

Virtual MCP

1. Emergency Stop Button (PLC I/O's **%I9.0** and **%Q19.0**)
2. Feed and Spindle Override
3. Alarm-, Channel-, Help-Key
4. Standard MCP keys, Modes, Reset, NC-Start/Stop, Spindle/Feed Start/Stop, Axis selection, MCS/WCS etc.
5. Freely assignable customer keys T1-T15
6. Key Switch (0-3)

The simulated MCP can be used for Ethernet and PROFINET MCP's in the project.

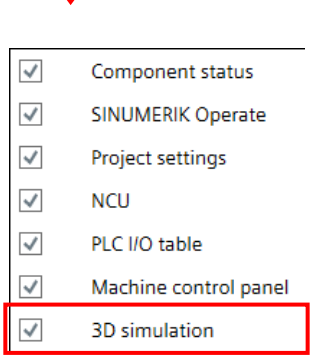
Customization of the standard virtual MCP is not possible at the moment!

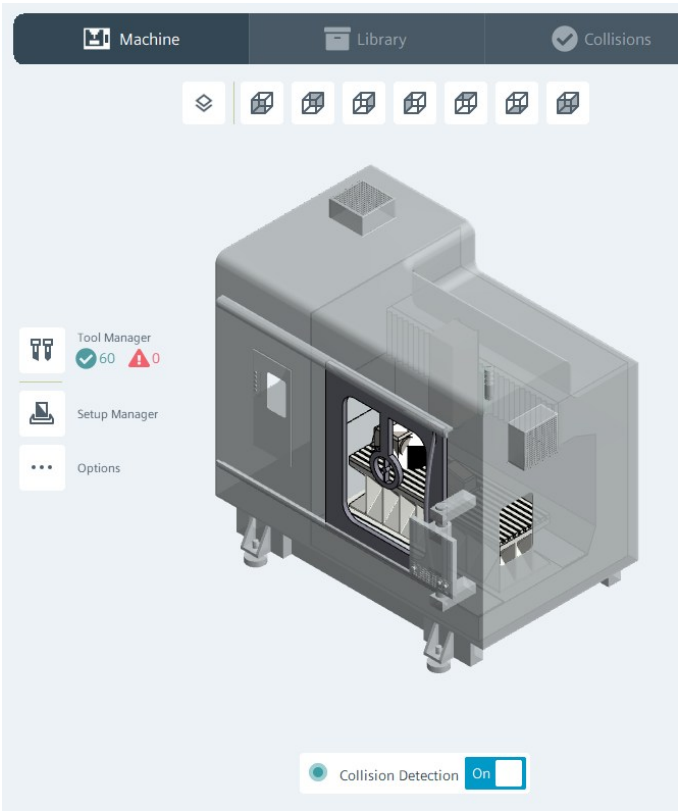


3D Simulation

It is possible to add a 3D Simulation Model to the virtual machine. This is not part of the course content and will be described in the modules for programing and operation.







Notes:

Signals from MCP (keys)
Input image

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IB n + 0	Spindle speed override				Operating mode			
	D	C	B	A	JOG	TEACH IN	MDA	AUTO
IB n + 1	Machine function							
	REPOS	REF	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
IB n + 2	Key-switch position 0	Key-switch posi-tion 2	Spindle start	*Spindle stop	Feed start	*Feed stop	NC Start	*NC Stop
IB n + 3	Reset	Key-switch posi-tion 1	Single block	Feedrate override				
				E	D	C	B	A
IB n + 4	Direction keys			Keyswitch position 3	Direction keys			
	plus R15	minus R13	Rapid trav-erse R14		X R1	4th axis R4	7th axis R7	R10
IB n + 5	Axis selection							
	Y R2	Z R3	5th axis R5	Traverse command MCS/WCS	R11	9th axis R9	8th axis R8	6th axis R6
IB n + 6	Unassigned customer keys							
	T9	T10	T11	T12	T13	T14	T15	
IB n + 7	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

Signals to MCP (LEDs)
Output image

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
QB n + 0	Machine function				Operating mode			
	1000 INC	100 INC	10 INC	1 INC	JOG	TEACH IN	MDA	AUTO
QB n + 1	Feed start	Feed stop	NC Start	NC Stop	Machine function			
					REPOS	REF	Var. INC	10000 INC
QB n + 2	Direction key minus R13	Axis selection				Single block	Spindle start	Spindle stop
		X R1	4th axis R4	7th axis R7	R10			
QB n + 3	Axis selection							Direction key plus R15
	Z R3	5th axis R5	Travel com-mand MCS/WCS R12	R11	9th axis R9	8th axis R8	6th axis R6	
QB n + 4	Unassigned customer keys							Y R2
	T9	T10	T11	T12	T13	T14	T15	
QB n + 5	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

Notes:

Working with Widgets

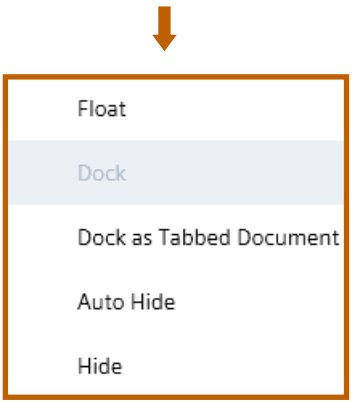


Enable Auto Hide Widget

Disable Auto Hide Widget

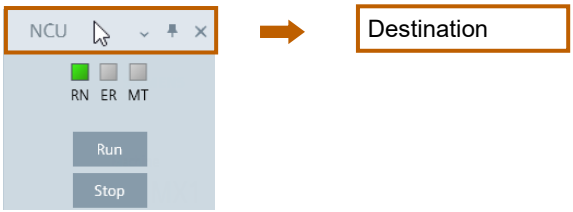
Hide Widget

Widget Option Menu



Drag and Drop Widgets

Widgets can easily be managed with the mouse. You can Drag and Drop them to various Locations.



There for click and hold with the mouse on the upper dark grey bar of the widget and move it to:

Outside of the Machine Simulation Screen

- As separate Window (Float)
- Into an already separate Window/Widget

Inside the Machine Simulation Screen

- As main Widget on the Top, Bottom, Left Side or Right Side of the Machine Simulation Screen
- As additional Widget on the Top, Bottom, Left Side or Right Side of another Widget



Place Widget at the Top of the Machine Simulation Screen or Widget



Place Widget at the Left of the Machine Simulation Screen or Widget



Place Widget at the Right of the Machine Simulation Screen or Widget

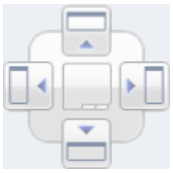


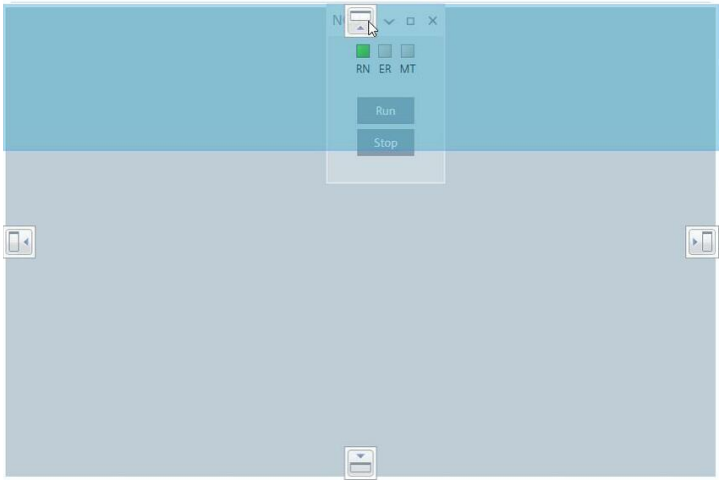
Place Widget at the Bottom of the Machine Simulation Screen or Widget

Procedure

1. Click and hold with the left mouse key on the upper dark grey bar of the widget
2. Move the mouse cursor to a target zone.
3. The Target Zone will be marked as light blue rectangle.
4. Release the left mouse key

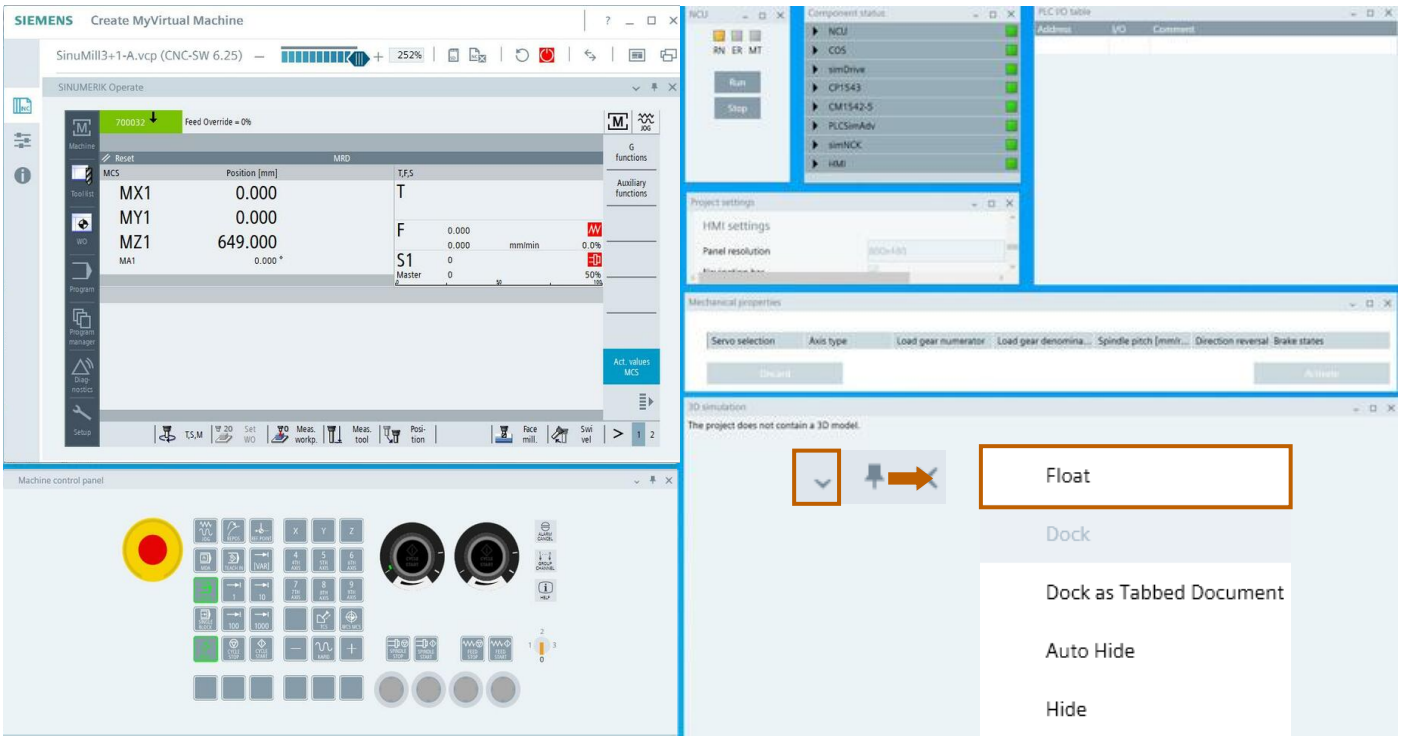
In case another widget is already in the target zone, you can select on which side of this widget the next one is placed.

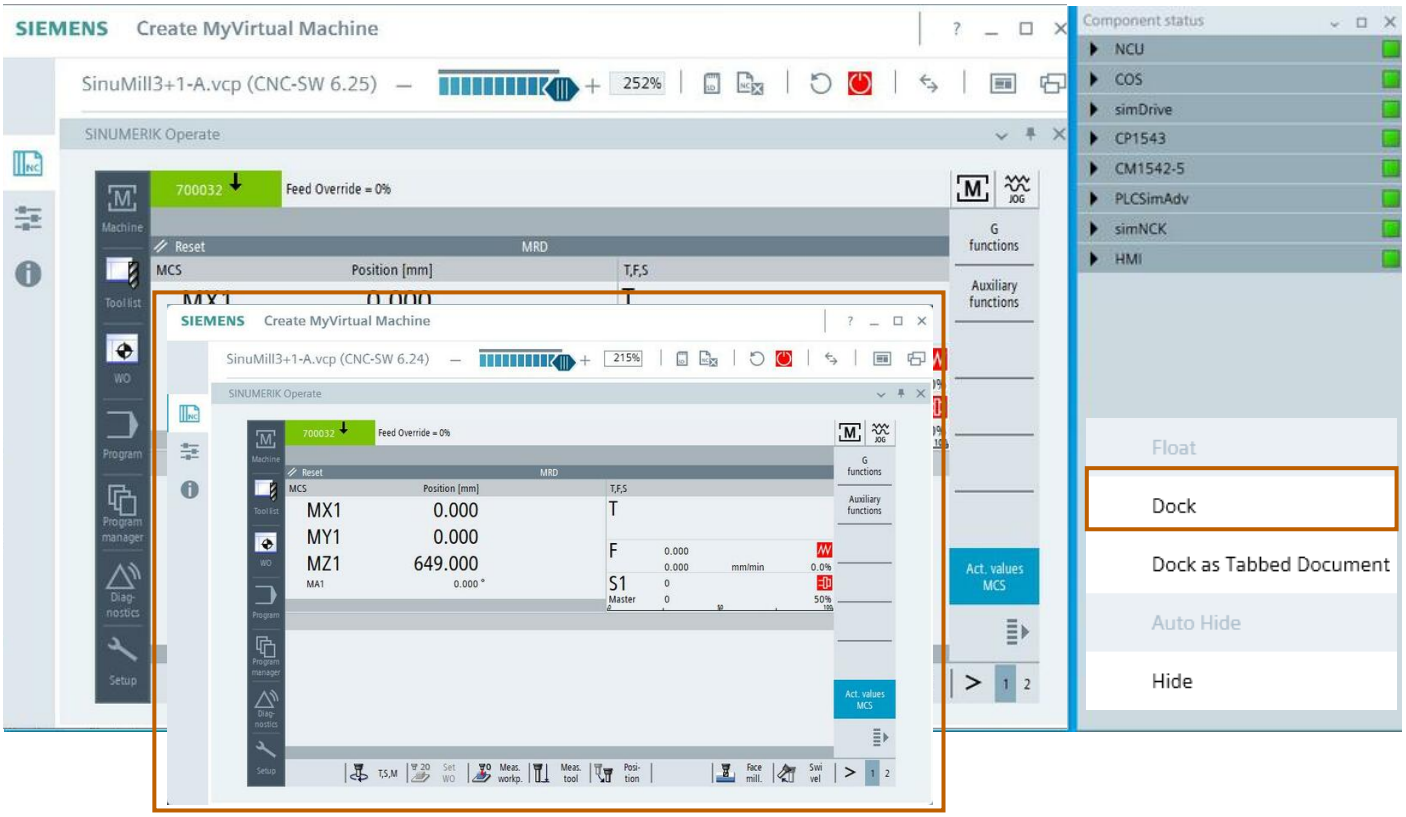




Notes:

Floating and Docking Widgets



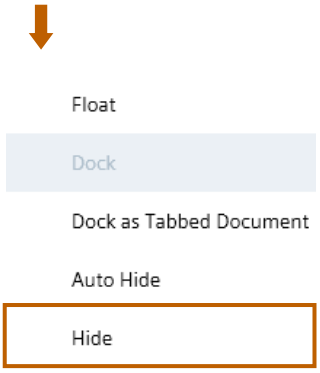


Notes:

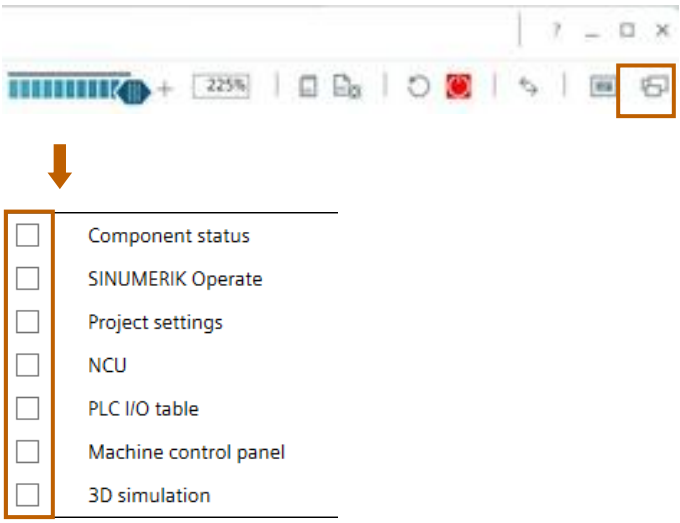
Hide Widgets



If you click hide a widget, it is completely removed from the Simulations screen.



Unhide can be done from the main menu bar by selecting the widget again.



Auto Hide Widgets



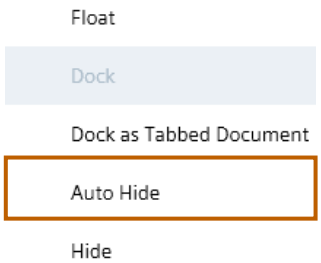
With this function the widget is hidden as a **Tab**.

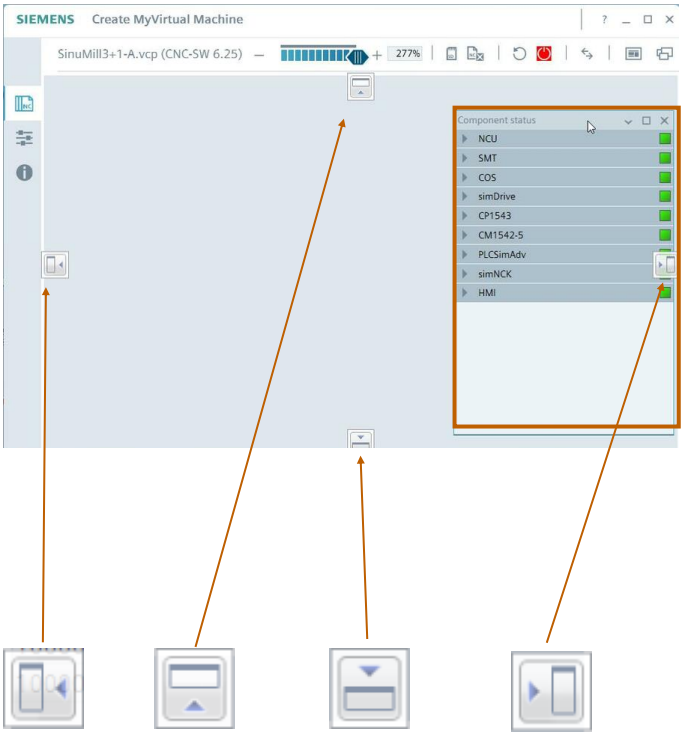
The Tab Location depends on the previous position of the widget in the simulation view.

Right side placed Widget are tabbed to right side tab-bar. Top Side at the Top Tab-bar, and so on.

The Hidden widgets are diplayed automatically when the mouse cursor touches the Tab.

If the mouse is leaving the tab, the widget is hidden again.





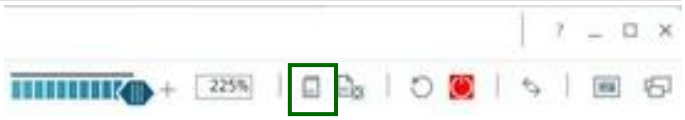
Notes:

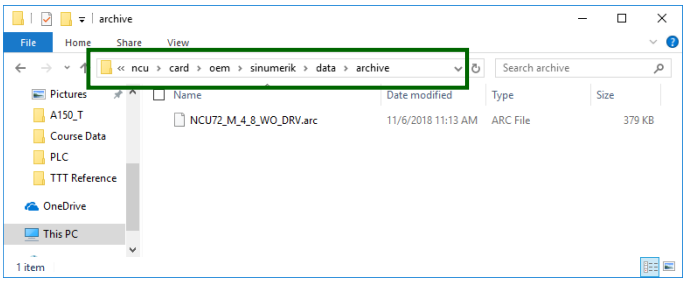
Read in *.arc Archives

*.arc archives from SINUMERIK 840Dsl can be read in to a virtual SINUMERIK ONE to import the NCK Settings. The new archive format of SINUMERIK ONE is described in the chapter “**A127 Data Management**”.

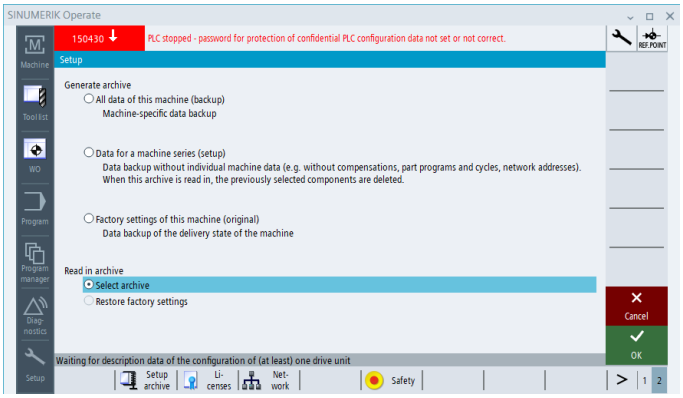
Open the virtual SD-Card and navigate to the following folder, card/oem/sinumerik/data/achives (Machine builder Archive folder).
Copy your *.arc (SINUMERIK 840Dsl) archive to this folder

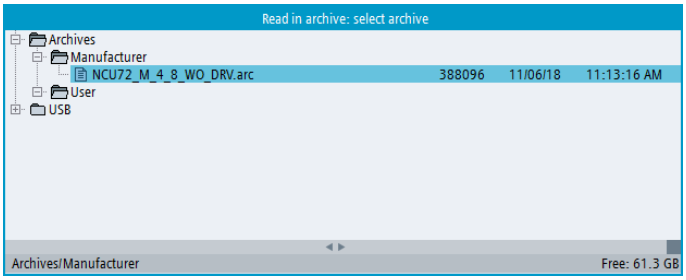
C:\Users\“your user name“\AppData\Local\Siemens\Automation\SINUMERIK ONE\ncu\card\oem\sinumerik\data\archive

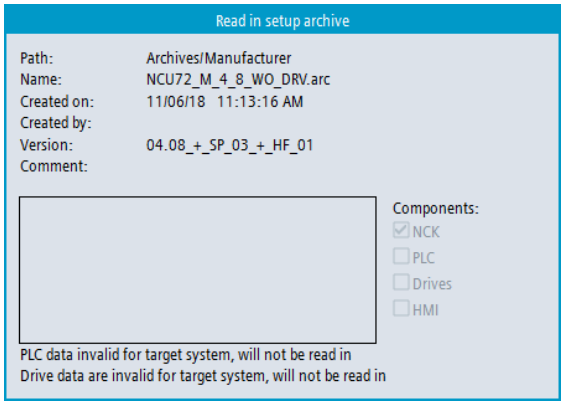




Go to the „**Setup**“ area and then press „**Setup archive**“. After this go to your SINUMERIK 840Dsl *.arc Archive and press the Soft Key „**Ok**“. From the existing SINUMERIK 840D sl archive you can **only** read in the **NCK** component! Press **OK** to continue.



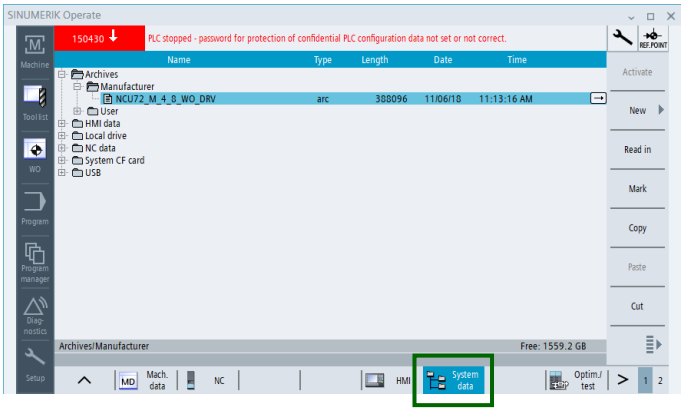


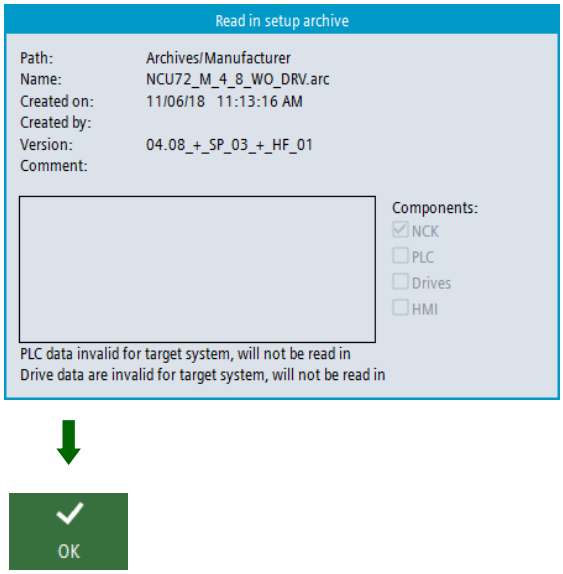


Notes:

Also you can go to the „**System data**“ area. After this go to your SINUMERIK 840Dsl *.arc Archive and press the Soft Key „**Read In**“. From existing SINUMERIK 840D sl archives you can **only** read in the **NCK** component!

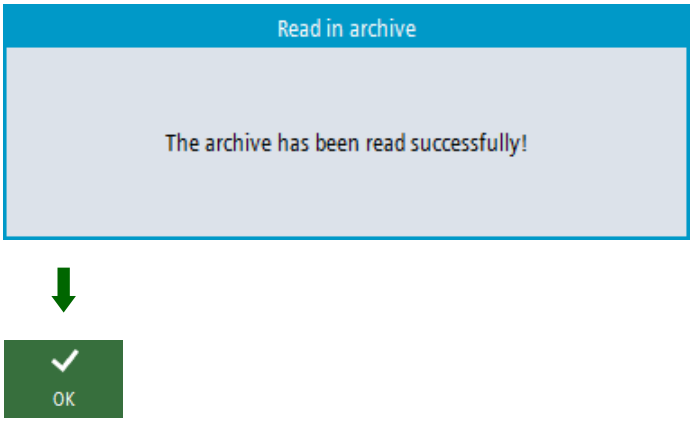
Press **OK** to continue.

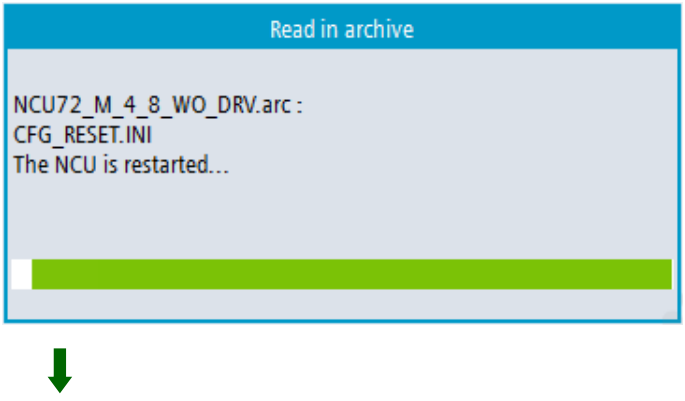




The archive is now read in. After finishing this process the message appears **“The archive has been read successfully“**

Confirm this by pressing **OK**.





A103: END

Notes:

A107

SINUMERIK ONE

Access Levels / Time and Date settings

Module Description:

Access Levels: The data of the Sinumerik ONE controller is protected by seven access levels.
Four of the access levels are set via a key-switch and three by password.
Each access level allows varying degrees of viewing/editing rights of the data.
While the majority of the data viewing/editing rights are determined by default, it is possible for the manufacturer/end user to set access rights for some data types.

Date and Time: It is important that the system date and time are set correctly.
This is useful for checking when an alarm occurred or when a file (Part program, archive etc) was created or edited.
Siemens warranty is dependant on the specific date/time that the components were purchased/installed.
An incorrect date/time setting could delay an On Site Service (OSS) visit.
This is part of the Global Service Platform (GSP) Ibase and is described in Module A007

Note: The virtual machine time and date are automatically taken from the host PC.

Module objective:

Upon completion of this module you will be able to:

- Set the different access levels using a password or key switch.
- Change the viewing/editing/execution rights of individual NC programmes
- Determine the access levels for various Soft-keys.
- Set the date and time of the system

Content:

Password and Key-switch Information

Setting and Deleting the Password

Real Machine: Changing the passwords

Virtual Machine: Changing the passwords

Protection and Attributes of Various Data Areas

Protection of Individual Files

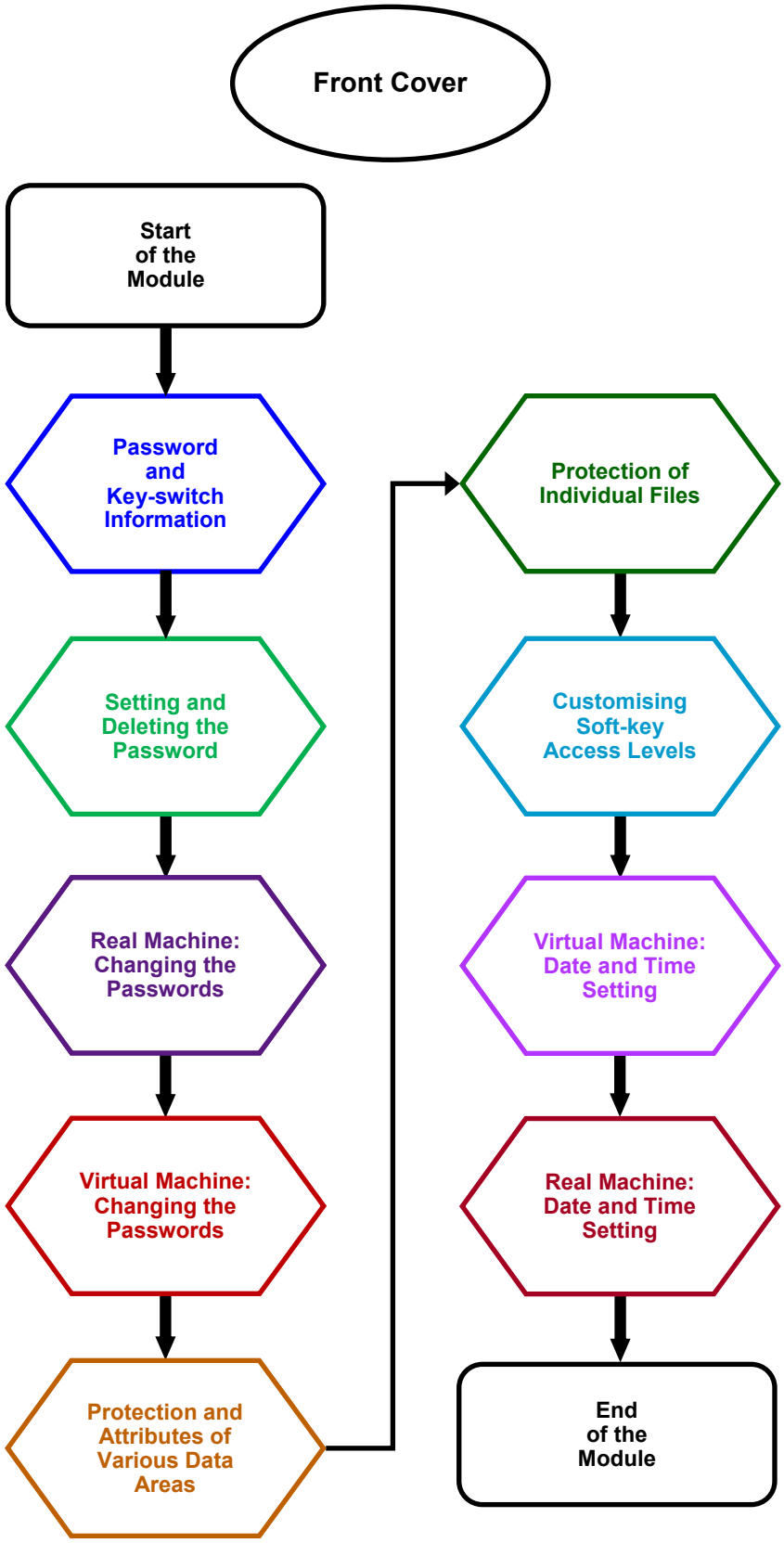
Customising Soft-key Access Levels

Virtual Machine: Date and Time Setting

Real Machine: Date and Time Setting

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Password and Key-switch Information
Setting and Deleting the Password
Real Machine: Changing the Passwords
Virtual Machine: Changing the Passwords
Protection and Attributes of Various Data Areas
Protection of Individual Files
Customising Soft-key Access Levels
Virtual Machine: Date and Time Setting
Real Machine: Date and Time Setting
End of the Module

Password and key-switch information

The following table shows the password/key-switch positions and associated access levels/users.

Access level	Intended user	Protection by	Access to:
1	Manufacturer	Password: SUNRISE	Defined functions, programs and data; e.g. entering options.
2	Installation Engineer	Password: EVENING	Defined functions, programs and data; e.g. most machine data
3	End User/service	Password: CUSTOMER	Assigned function, programs and data.
4	Programmer, Setter	Key-switch position 3	Less than protection levels 0 –3. Defined by the manufacturer/end user.
5	Qualified operator	Key-switch position 2	Less than protection levels 0 –3. Defined by the end user.
6	Trained operator	Key-switch position 1	Example: program selection only. Tool wear and offset entry.
7	Semi-skilled operator	Key-switch position 0	Example: No tool wear and offset entries or program selection possible

Real Machine Security Recommendation:

The above passwords are default. For IT security reasons it is recommended that individual passwords are assigned to protect unauthorised machine modifications.

The following alarm is displayed as a reminder:
2130 Note: The standard password is still active for at least one of the access levels: manufacturer, service or user.

Changing of the passwords is described later in the module.

General Access Information:

- Protection level 1 provides the highest access rights, protection level 7 the least.
- If certain access rights are granted to a protection level, these protection rights automatically apply to to any higher protection level.
- Conversely, protection rights for a certain protection level can only be altered from a higher protection level.
- Access rights for password levels 1 to 3 are assigned by Siemens and cannot be altered.
- Access rights can be set by querying the current key-switch positions and comparing the passwords entered. When a password is entered it overwrites the access rights of the key-switch position.
- Options can be protected on each protection level. However, option data can only be entered in protection level 1. Access rights for protection levels 4 to 7 are only suggestions and can be altered by the machine tool manufacturer or end user.

Key-switch Information:

The key switch is located on the MCP and has four positions. Three colour coded keys are available, each of which allow a different access level.
This is standard on Siemens machine control panels. It is possible that a manufacturer may use a different method to set the interface signal.



Key switch


Colour coded keys

Access levels of the three keys:

Key colour	Switch position	Access level
Key not required	0	7
Black	0 & 1	6-7
Green	0, 1 & 2	5-7
Orange	0, 1, 2 & 3	4-7

It is the responsibility of the OEM to ensure that the key-switch signals are transferred to the PLC interface.

Appendix C shows the relevant interface signals for key-switch

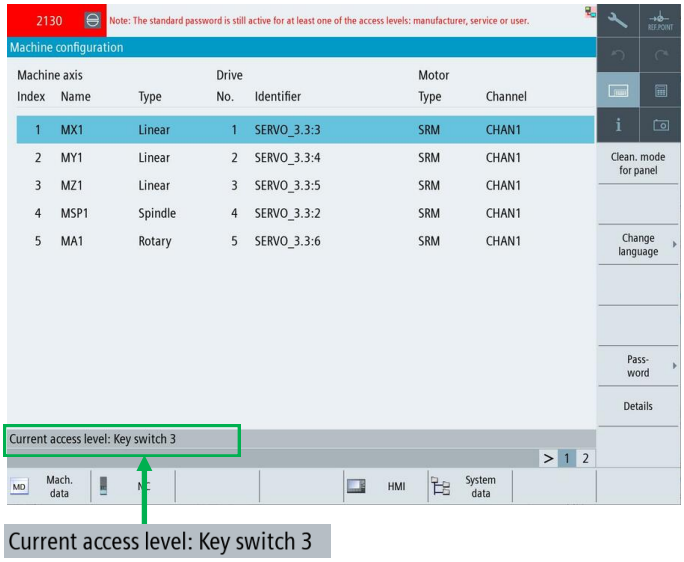
Notes:

Setting and Deleting the Password

The screen for entering the password is accessed via the ***Setup*** menu



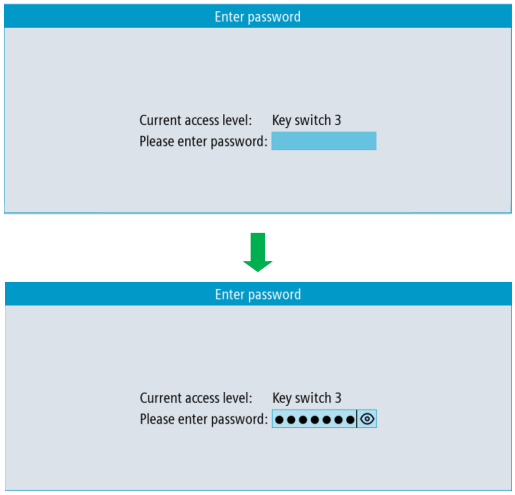
The current access level is displayed to the lower left of the screen.



Current access level: Key switch 3

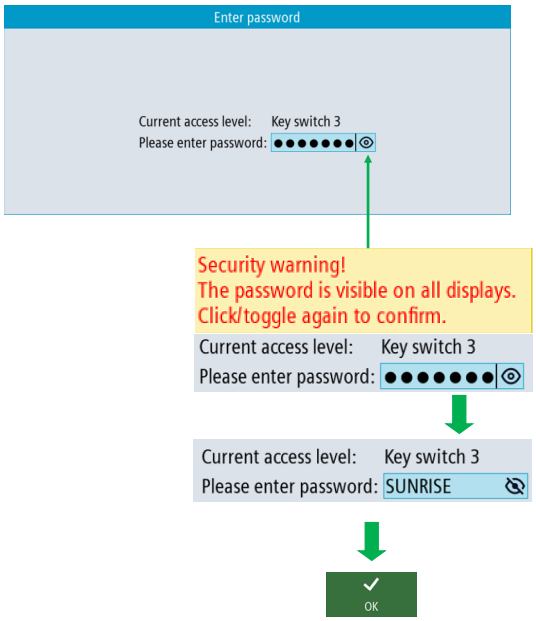
Pass-word → Enter password

A dialogue box, which shows the current access level, is opened. The required password needs to be entered using the keyboard.

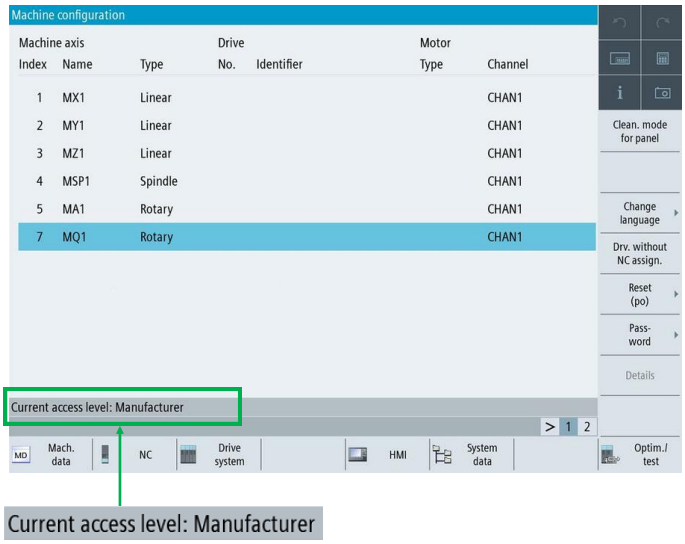


Viewing the Password:

The password characters can be viewed by selecting the ***eyeball*** symbol - a security warning is issued.



The dialogue box clears and the current access level is displayed above the left hand horizontal soft-keys.

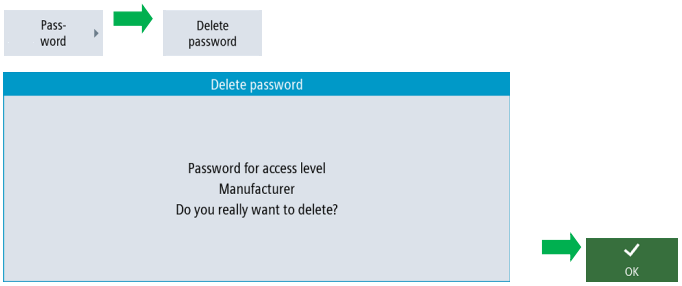


Current access level: Manufacturer

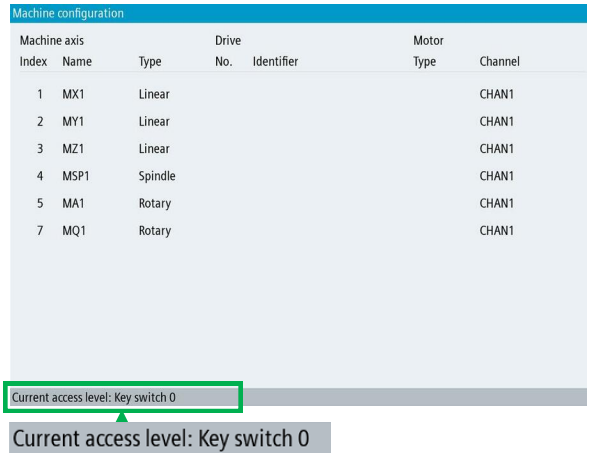
Note: The password remains active after power off/on. Module A125 explains how password deletion can be automated.

Notes:

Deleting the Password



The access level is now dependant on the key-switch.



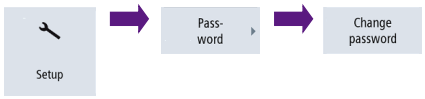
Real Machine: Changing the Passwords

For IT security reasons it is recommended that individual passwords are assigned to protect unauthorised machine modifications.

Alarm - 2130 - is displayed if one are more of the standard passwords is active.



Change the password using the ***Change password*** soft-key:





Note the rules for strong passwords:

A password must contain at least 8 and can contain up to 32 characters.

Permissible characters are:
- Uppercase characters
- Lowercase characters
- Numbers
- Special characters (ASCII 0x20 - 0x7E)

The character combination of a password must contain:
- At least one number
- At least one uppercase letter
- At least one lowercase letter

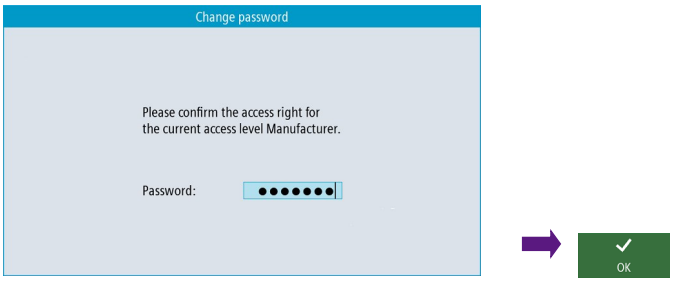
A password must not be too similar to the following names:
- User names (manufacturer, service, user)
- Computer name
- The additional values that can be saved for a user in the Linux password file (so-called GECOS data)

We recommend that you confine yourself to the characters available on the operator panel when selecting a password.

Note:
Once a password has been changed it is not possible to change back to the standard password.
This is because it will not meet the minimum password requirements.

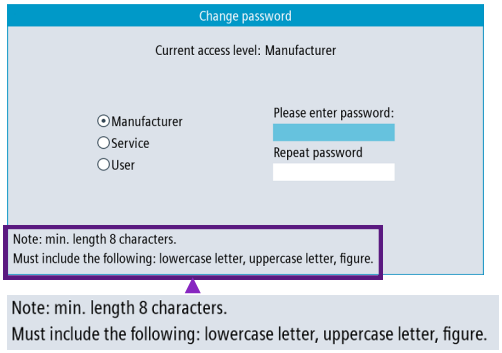
Notes:

The current password of the active access level needs to be confirmed e.g. SUNRISE



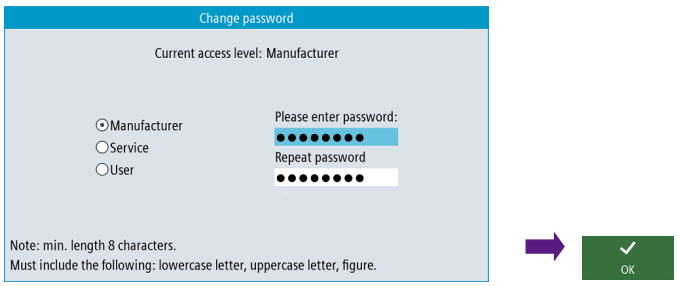
The access level (or higher) of the password to be changed must be active.

Example - all three passwords can be changed when the ***Manufacturer*** access level active .



Note: min. length 8 characters.
Must include the following: lowercase letter, uppercase letter, figure.

- Note the password rules.
- Select the access level to be changed
- Enter the new password.
- Repeat the new password.
- Confirm.



The new password is now active.

Resetting the Passwords to Default:

The passwords can only be returned to default by restoring a TGZ image using emergency boot system and selecting the option:
Update NCU software --> Clear all + reinstall system sw. (with or without system restore point).

Refer to module, A067 Preparation of NCU and IPC, for more information.

Notes:

Virtual Machine: Changing the Passwords

The change password function between the real machine and the virtual machine is not identical.

The virtual machine does not show alarm - 2130. This warns the user that at least one Siemens standard passwords is still active and that it should be changed for security reasons.

The passwords of the virtual machine can be changed but the criteria of the new passwords is not as strict as with the real machine.

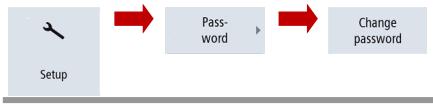
Unlike the real machine, once a password has been changed it can be easily set back to the original settings.

Password settings from the virtual machine are not transferred to the real machine via the loading of an archive.

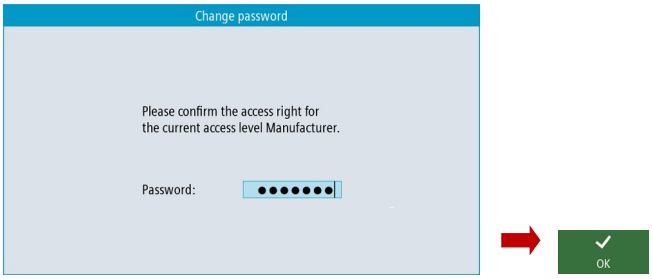
Standard Password Information:

- A password may contain up to eight characters.
- It is recommend that the characters available on the operator panel are used when defining the password.
- Where a password consists of less than eight characters, the additional characters are interpreted as blanks.

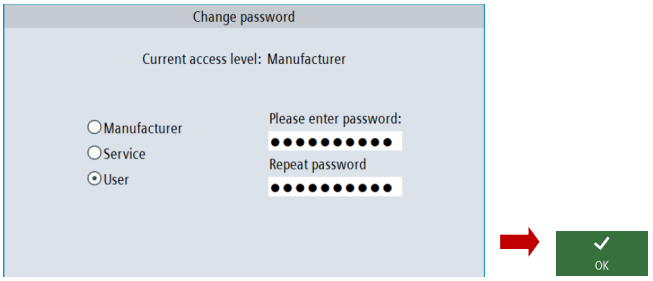
Changing the Password:



The current password of the active access level needs to be confirmed e.g. SUNRISE



- Select the access level to be changed
- Enter the new password.
- Repeat the new password.
- Confirm.



Protection and Attributes of System/OEM/User Data

The manufacturer/end user can determine the read/write/ execution properties of various data areas via ***General Machine Data.***

A list of machine data with the default access levels can be seen below:

General machine data		
11160	$MN_ACCESS_EXEC_CST	7
11161	$MN_ACCESS_EXEC_CMA	7
11162	$MN_ACCESS_EXEC_CUS	7
11165	$MN_ACCESS_WRITE_CST	-1
11166	$MN_ACCESS_WRITE_CMA	-1
11167	$MN_ACCESS_WRITE_CUS	-1
11170	$MN_ACCESS_WRITE_SACCESS	7
11171	$MN_ACCESS_WRITE_MACCESS	7
11172	$MN_ACCESS_WRITE_UACCESS	7
51044	$MNS_ACCESS_SHOW_SBL2	7
51045	$MNS_ACCESS_TEACH_IN	4
51046	$MNS_ACCESS_CLEAR_RPA	4
51047	$MNS_ACCESS_READ_GUD_LUD	7
51048	$MNS_ACCESS_WRITE_GUD_LUD	7
51049	$MNS_ACCESS_WRITE_PRG_COND	7
51050	$MNS_ACCESS_WRITE_PROGRAM	4
51051	$MNS_ACCESS_WRITE_RPA	7
51052	$MNS_ACCESS_WRITE_SEA	7
51053	$MNS_ACCESS_WRITE_BASEFRAME	7
51054	$MNS_ACCESS_WRITE_CYCFRAME	7
51055	$MNS_ACCESS_WRITE_EXTFRAME	7
51056	$MNS_ACCESS_WRITE_PARTFRAME	7
51057	$MNS_ACCESS_WRITE_SETFRAME	7
51058	$MNS_ACCESS_WRITE_TOOLFRAME	7
51059	$MNS_ACCESS_WRITE_TRAFRAME	7
51060	$MNS_ACCESS_WRITE_USERFRAME	4

General machine data		
51061	$MNS_ACCESS_WRITE_WPFRAME	7
51062	$MNS_ACCESS_WRITE_FINE	6
51063	$MNS_ACCESS_SET_ACT_VALUE	4
51064	$MNS_ACCESS_WRITE_PROGLIST	4
51070	$MNS_ACCESS_CAL_TOOL_PROBE	4
51071	$MNS_ACCESS_ACTIVATE_CTRL_E	1
51072	$MNS_ACCESS_EDIT_CTRL_E	2
51073	$MNS_ACCESS_SET_SOFTKEY_ACCESS	3
51074	$MNS_ACCESS_WRITE_WPC_COUNTER	7
51075	$MNS_ACCESS_WRITE_MDI	7
51160	$MNS_ACCESS_WRITE_CA_MACH_JOG	6
51161	$MNS_ACCESS_WRITE_CA_MACH_AUTO	3
51162	$MNS_ACCESS_WRITE_CA_TOOL	1
51198	$MNS_ACCESS_READ_TM_ALL_PARAM	1
51199	$MNS_ACCESS_WRITE_TM_GRIND	4
51200	$MNS_ACCESS_WRITE_TM_GEO	5
51201	$MNS_ACCESS_WRITE_TM_WEAR	6
51202	$MNS_ACCESS_WRITE_TM_WEAR_DELTA	7
51203	$MNS_ACCESS_WRITE_TM_SC	7
51204	$MNS_ACCESS_WRITE_TM_EC	7
51205	$MNS_ACCESS_WRITE_TM_SUPVIS	7
51206	$MNS_ACCESS_WRITE_TM_ASSDNO	7
51207	$MNS_ACCESS_WRITE_TM_WGROUP	7
51208	$MNS_ACCESS_WRITE_TM_ADAPT	7
51209	$MNS_ACCESS_WRITE_TM_NAME	4
51210	$MNS_ACCESS_WRITE_TM_TYPE	4
51211	$MNS_ACCESS_READ_TM	7
51214	$MNS_TM_WRITE_LIMIT_MASK	7H
51215	$MNS_ACCESS_WRITE_TM_ALL_PARAM	4
51216	$MNS_ACCESS_TM_TOOL_CREATE	4
51217	$MNS_ACCESS_TM_TOOL_DELETE	4
51218	$MNS_ACCESS_TM_TOOL_LOAD	4
51219	$MNS_ACCESS_TM_TOOL_UNLOAD	4
51220	$MNS_ACCESS_TM_TOOL_MOVE	4
51221	$MNS_ACCESS_TM_TOOL_REACTIVATE	4

Notes:

General machine data		
51221	$MNS_ACCESS_TM_TOOL_REACTIVATE	4
51222	$MNS_ACCESS_TM_TOOL_MEASURE	4
51223	$MNS_ACCESS_TM_TOOLEDGE_CREATE	4
51224	$MNS_ACCESS_TM_TOOLEDGE_DELETE	4
51225	$MNS_ACCESS_TM_MAGAZINE_POS	4

The Help function shows the machine data description including the access levels.

Example:
General Machine data - ***11160 ACCESS_EXEC_CST***



11160	ACCESS_EXEC_CST				N01	-
-	Execution right for /_N_CST_DIR			BYTE	PowerOn	
-						
System	Dimension	Default value	Minimum value	Maximum value	Protection	Class
standard	-	7	1	7	7/2	M

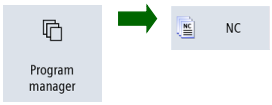
Description: Execution right assigned to the program stored in directory /_N_CST_DIR :
Value 1: Machine OEM password
Value 2: Password of setup engineer, service
Value 3: End user password
Value 4: Keyswitch position 3
Value 5: Keyswitch position 2
Value 6: Keyswitch position 1
Value 7: Keyswitch position 0
Machine data can only be written with value 1 if the password for machine OEM is also active.

This machine data is used to set the access level required for executing a program stored in the CST (Standard Cycles) directory.

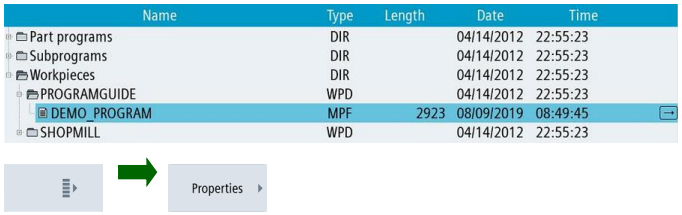
Individual File Protection and Attributes

The properties of individual files can be set to protect from editing, deletion or execution. It is also possible to hide a file from view.
The previously described access levels are used in conjunction with file properties to achieve the required protection.

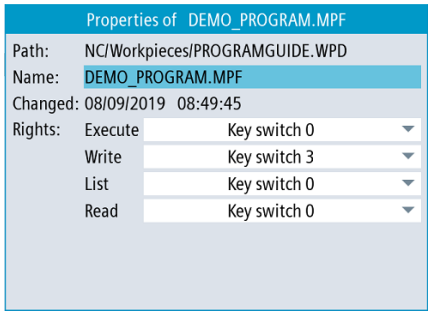
The following example shows the setting of the properties of a machining program.



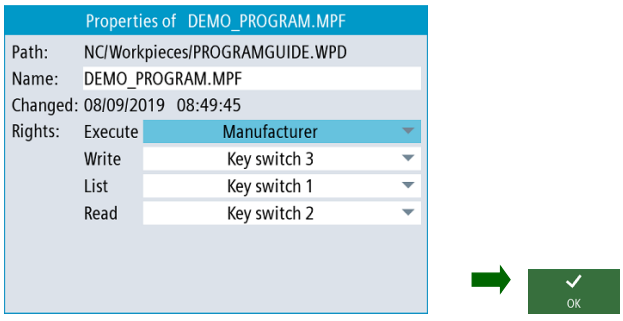
Example: the ***DEMO_PROGRAM*** has been selected



The default properties of the ***DEMO_PROGRAM*** :



The modified properties of the ***DEMO_PROGRAM*** :



The modified properties mean that the program:

- Can only be executed if the Manufacturer access level is set.
- Can only be written to (Edited) if the Key switch or higher access level is active.
- Will only be listed if the key switch position 1 or higher is active
- Can only be read if the key switch position 2 or higher is active

Note: changes made in the virtual machine are transferred to the real machine with the loading of an NC archive.

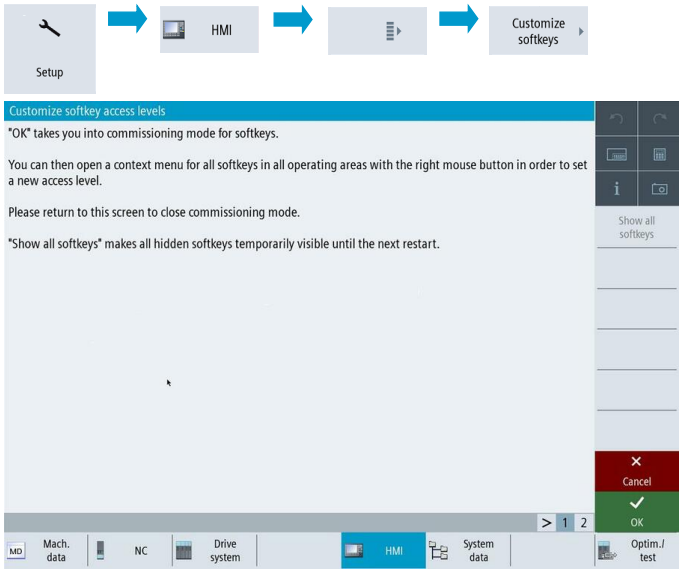
Notes:

Customising Soft-key Access Levels

It is possible to set the access levels for most soft-keys.
The function must first be activated in the Setup area.

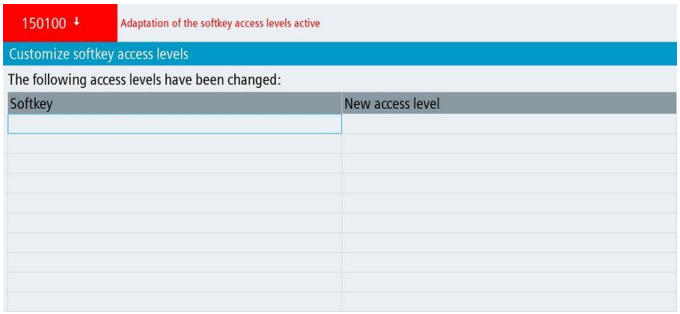
Notes: The minimum access level of *User* and also a mouse is required for this function.

Locate the Customize soft-key area:



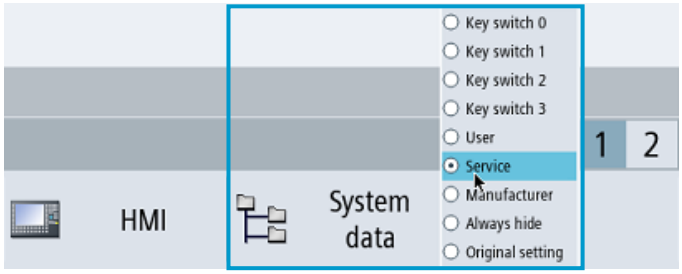


The message, 150100, will be displayed while the soft-key adaption is active.

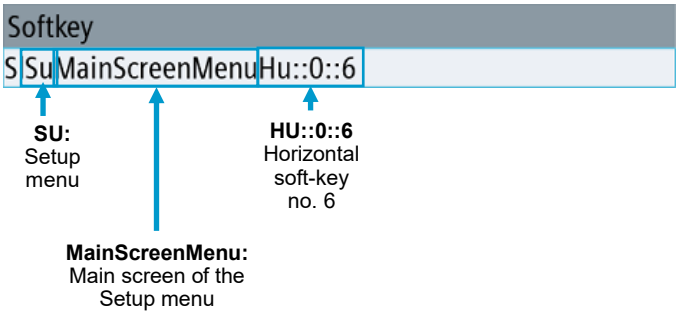


- Locate the soft-key, which requires customising.
- Use the mouse ***right click*** function on the soft-key
- Select the required access level from the menu using the mouse ***left click*** function.

Example: the ***System data*** soft-key adapted to the ***Service*** access level.



The soft-key can be identified from the text:



A summary of adapted soft-keys is listed on the ***Customize soft-keys*** screen.





Additional features:

Show all softkeys — All soft-keys will be displayed, regardless of soft-key adaption, until the next system restart

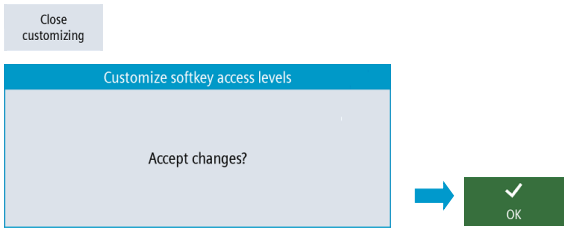
Original setting — The original settings can be reapplied to each soft-key on an individual basis.



Notes:

Accepting the changes

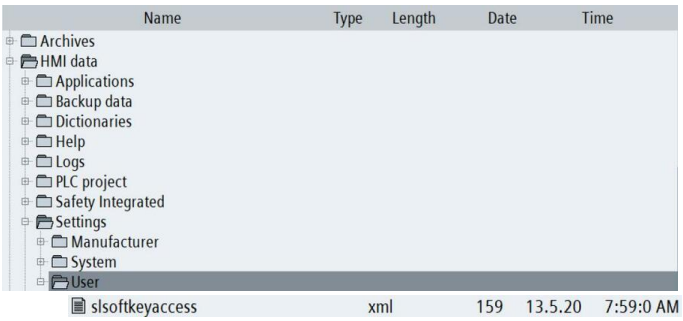
Select the ***Close customizing*** soft-key to accept the changes:



The soft-key availability can be tested by setting the relevant access levels.

File location:





- Transfer the file to the same location on the real machine.
- Initiate an HMI restart to activate the settings.

Virtual Machine to Real Machine

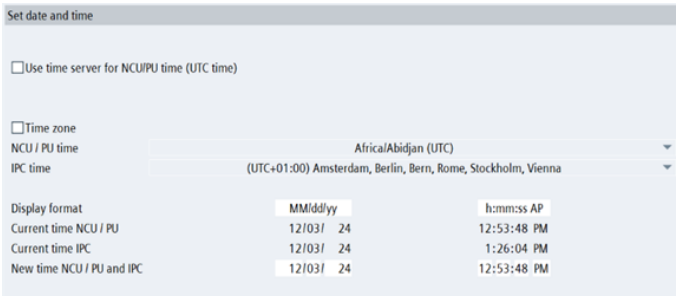
When the customizing of soft-keys feature is used the system creates the following file: ***slsoftkeyaccess.xml***

The file is not part of a virtual machine archive.
It must therefore be manually copied from the virtual machine to real machine.

Virtual Machine: Setting the Date and Time



Note: the functionality of the Date/Time settings is limited with the virtual machine.



Making date/time changes:

The password for the ***User*** or higher must be active before any changes to the date/time can be made.

Date/time settings from the virtual machine are not transferred to the real machine via the loading of archives.

Various options are available

- **Use time server for NCU/PU time (UTC time):**
  A time server address can be entered to automatically update the date and time e.g. pool.ntp.org
  Note: this is not functioning on the virtual machine.
- **Time zone:**
  An extensive list of time zones is available for selection.
  Note: this is not functioning on the virtual machine.
- **Manually setting:**
  - **IPC** - this is the time/date system of the host PC (real-time clock).
    It is possible to change the date and time of the host PC from the virtual machine.
    Likewise, any changes to the host PC time and date will result in them being transferred to the virtual machine.
    The date and time stamp for editing/creating files/ programs etc. are taken from the IPC time.
    This also applies to the raising and clearing of alarms.
  - **NCU / PU** - this is the time/date system of the virtual machine and has no influence on the system.

Changes are confirmed with the ***Accept*** and ***OK*** softkeys:



Notes:

Real Machine: Setting the Date and Time

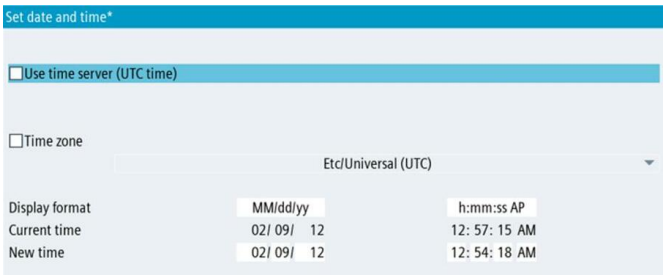
The system clock is present on the NCU. It is buffered by a capacitor. After a lengthy period in the powered off state the date/time will need correcting.
IPCs also have a clock. The time/date setting screens are therefore slightly different if an IPC is in use.

Before the date and time displays can be edited, the password for the ***User*** or higher must be entered.

The date/Time screen is reached with the following soft key sequence:

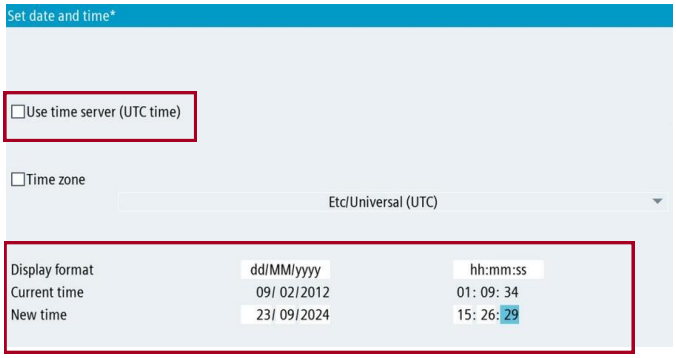


Time and Date setting via the ITC (no IPC present).



Option: Manual Time & Date Adjustment

To manually adjust the settings, ensure that the Use time server option is ***not*** selected.



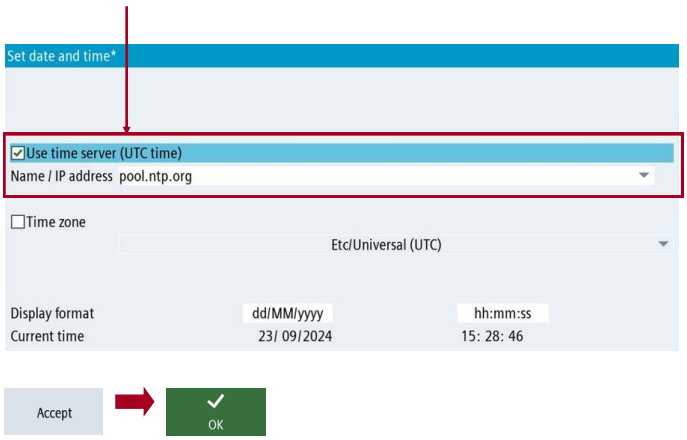
Enter the required date/time/format and confirm with the **Accep** & ***OK*** soft-keys.



Option: Use time server

If the controller is connected to the internet the option, ***Use time server,*** can be set.
The address/name of the time server needs to be entered and the relevant time zone selected from the time zone drop down menu.
The date and time format also needs to be selected.

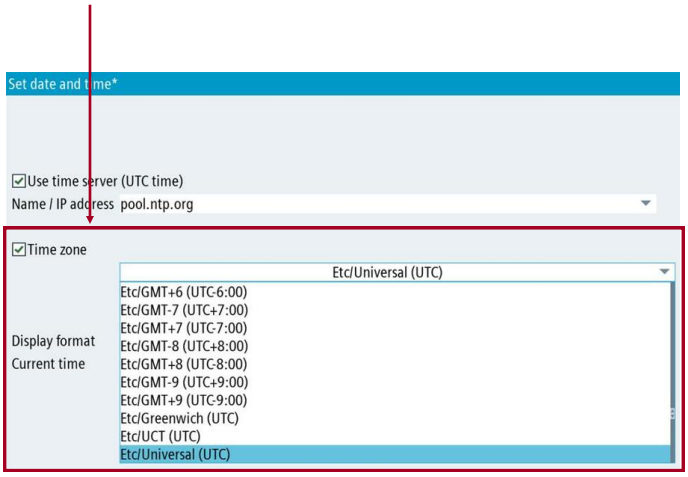
Example - Date & time set via the ***Time server*** option:



Option: Use time zone

A time zone can also be specified by setting the ***Use time zone*** option and selecting the desired time zone from ***Time zone*** list.

Example - Date & time set via the ***Use time zone*** option:



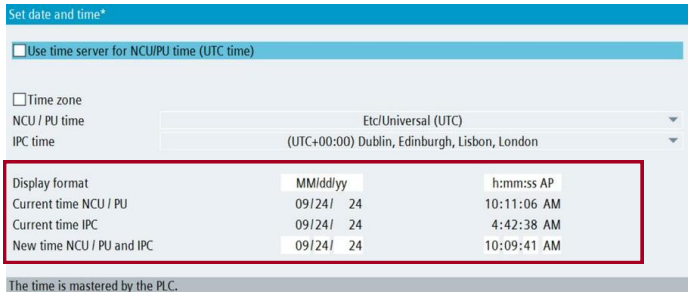


Notes:

Time and Date setting via the IPC.

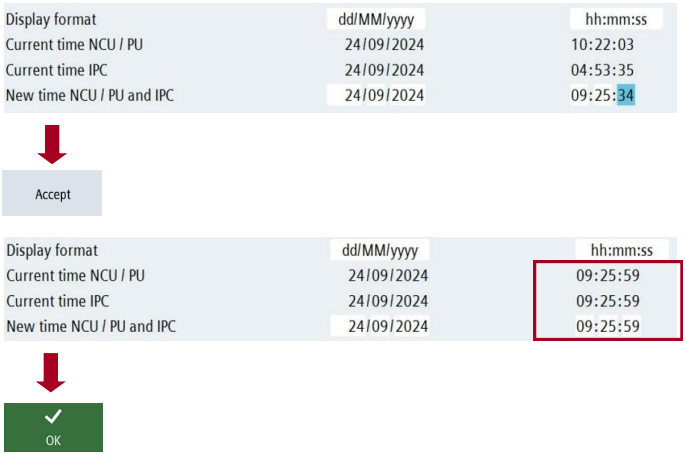
The clock of the IPC has to be taken into consideration when setting the date and time.





Note: the Time server/Time zone option function as previously described in this module (Page 12)

When the correct date and time is entered/accepted, the NCU and IPC clocks will be automatically synchronised.



A107: END

Notes:

Notes:

A109

SINUMERIK ONE

License and Option Management

Module Description:

The Sinumerik ONE system SD card includes basic functionality in the scope of delivery.

When extra functionality is required it is necessary to purchase licences.

A licensing website is used to manage the licences and to generate a licence key which is uniquely linked to the system SD card serial number.

When the licence key is entered on the controller it is then possible to activate the required options and operate the controller without any restriction.

A time limited test licence is available which allows unlicensed options to be set without affecting controller functionality.

This module explains the licensing procedure and how to activate options for both the virtual and real machine situations.

Module objective:

Upon completion of this module you can manage the licence key using the Sinumerik Operate HMI.

Content:

Licence and Option Overview

Virtual Machine: Option Setting

Real Machine: Option Setting

Real Machine: Test Licences

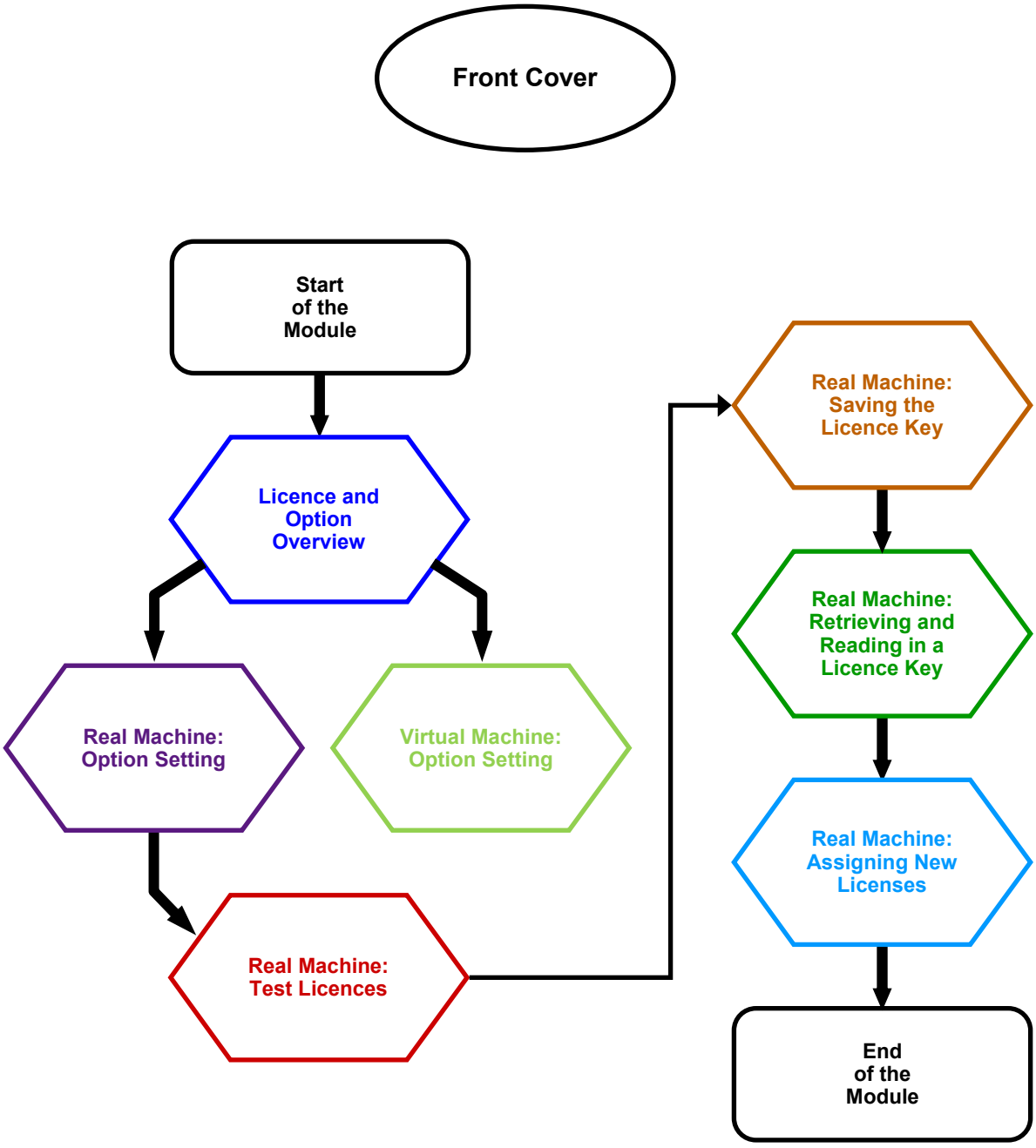
Real Machine: Saving the Licence Key

Real Machine: Retrieving and Reading in a Licence Key

Real Machine: Assigning New Licenses

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Licence and Option Overview
Real Machine: Option Setting
Virtual Machine: Option Setting
Real Machine: Test Licences
Real Machine: Saving the Licence Key
Real Machine: Retrieving and Reading in a Licence Key
Real Machine: Assigning New Licenses
End of the Module

Licence and Option Overview

Real Machine

Basic functionality, e.g. maximum 3 axes, is included in the scope of delivery of the Sinumerik ONE.

Expansion of this basic functionality requires options to be purchased, licenced and set.

Purchased options need assigning to the relevant Sinumerik ONE system SD card. (via the Siemens Web License Manager website).

When the purchased options are assigned to the SD card, a licence key is generated.

The licence key is unique and can only be used with the designated system SD card.

The licence key has to be loaded/entered via the licencing screen of the controller.

Activating an unlicensed option results in the inhibiting of the cycle start function.

Test licences are available to temporarily allow unlicensed options to be set and used without inhibiting the cycle start function.

Virtual Machine

All options are available and licenced.

The required options for the real machine can be set here.

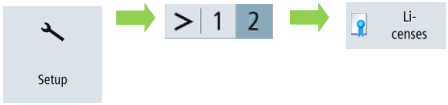
These selections are transferred to the real machine via the NC archive.

The test licence philosophy does not apply.

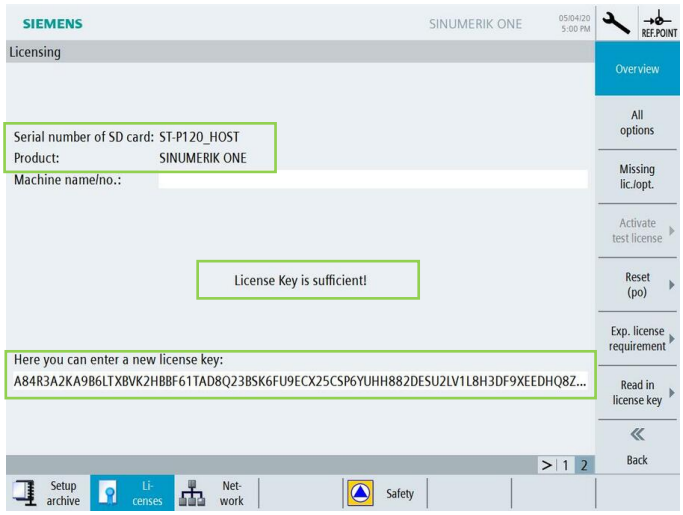
The serial number of the SD card and the license key, which are displayed on the licensing screen, are particular to the virtual machine software installation and have no relevance on the real machine.

Virtual Machine: Option Setting

The licencing screens are accessed via the Setup menu:

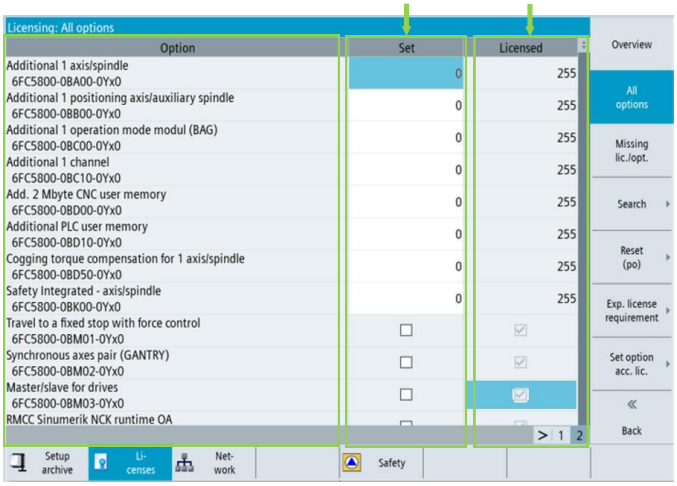


An example of a virtual machine licencing overview screen:



A list of available options is accessed via the ***All options*** softkey:





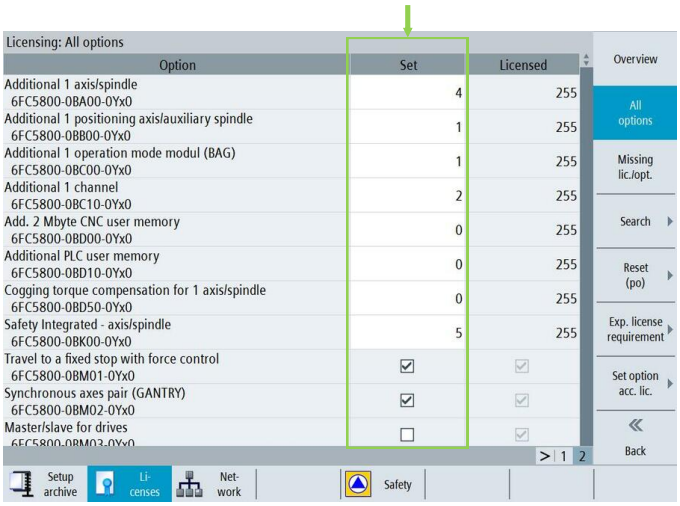
The left column shows the option name and number.
The middle column shows which options have been set.
The right hand column shows the fully available licensed options

Notes:

Setting the Options

Any options required on the real machine can be set in the virtual machine.
These settings are part of the NC archive and will be transferred to the real machine.

Example - various options which have been manually set to suit the requirements of the real machine:



An NCK reset is required to activate the settings



Virtual Machine Licensing Notes:

The virtual machine licensing feature has limited functionality.

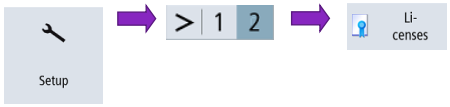
Do not use the ***Set option acc. Lic***. soft-key. This will set every SINUMERIK ONE option and result in conflicts and impossible situations.

The test licence philosophy does not apply.

Exporting the license requirement/Reading in the license key serves no functional purpose.

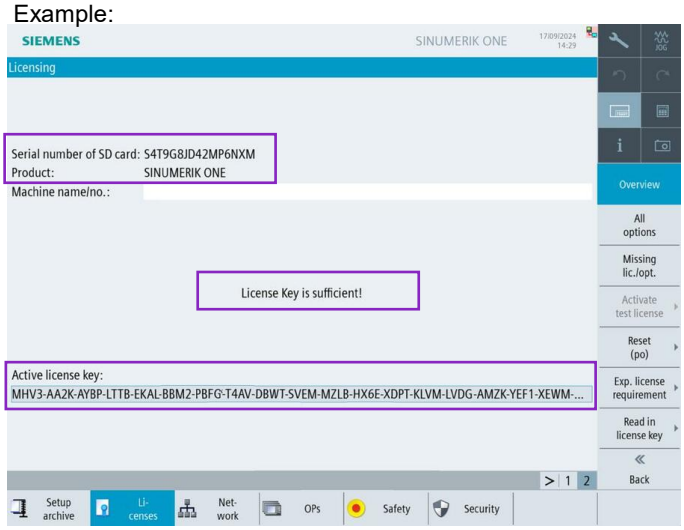
Real Machine: Setting the Options

Use the following key selection to reach the licence screens:



Information regarding the license and the SD card serial number is displayed.

Example:

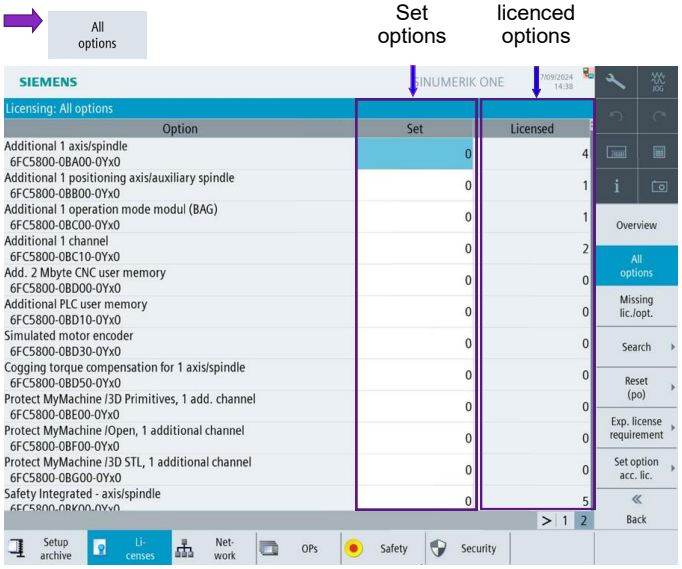


Option Management.

Options are used to expand the functionality of the Sinumerik ONE controller, for example, an additional axis.

The available options are displayed via the ***All options*** soft-key.

Example: licenced options are available, but have not yet been set.



Notes:

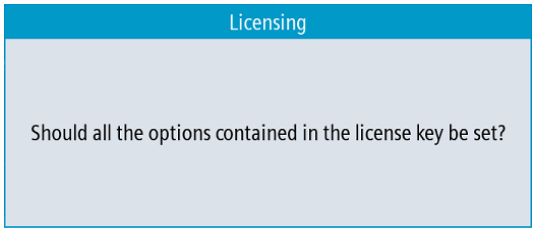
The options which are licenced for that particular System SD card are displayed in the right hand ***Licensed*** column.

The ***Set*** column shows options that have been set.

Virtual machine: if an NC archive from a virtual machine, which includes set options, is loaded, then this will result in the same options being set on the real machine.

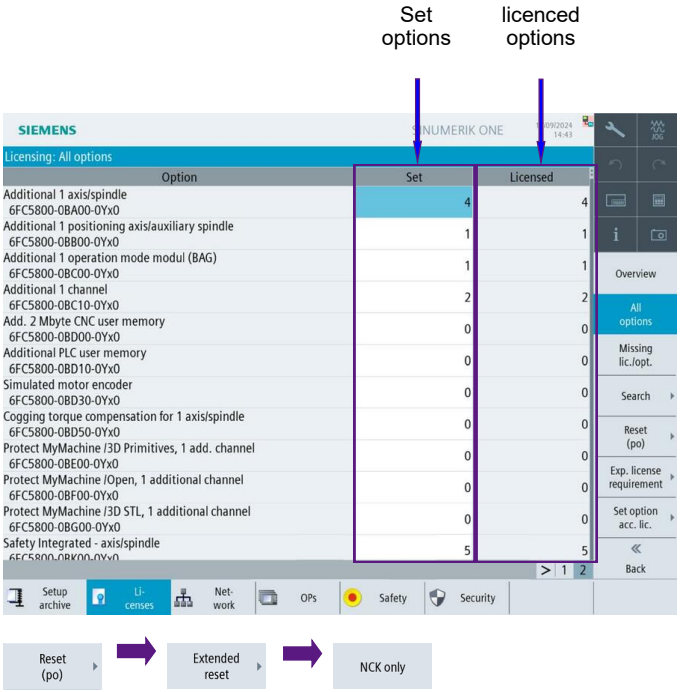
The options can be set individually. Alternatively, all the licenced options can be set, automatically, by selecting the ***Set option acc. Lic.*** Soft-key.







The licenced options have now been set.
They are activated by an NCK Reset.



Notes:

Real Machine: Test Licences:

A missing/incorrect license key will result in alarm number 8081.
The same applies in the event of an unlicensed option being set.



Example: An unlicensed option and alarm 8081



Note: The cycle start function is blocked if the licence key is not sufficient.

In the event of an unlicensed option being set it is possible to activate a ***Test licence***.

Up to six time monitored test licences are available should a licence key be unavailable/incorrect.

Sinumerik ONE Test Licence Periods:

Sinumerik ONE Test Licence Periods		
Type:	Period 1:	Periods 2 to 6:
PPU 1740	1000 hours	150 hours
NCU 1740	1000 hours	150 hours
NCU 1750	2000 hours	150 hours
NCU 1760	3000 hours	150 hours

Test Licence use and philosophy:

Test licence 1:

- OEMs can freely activate and test options during the commissioning stage. Once satisfied which options are required, these can be officially purchased and licensed. The complexity of the controller is reflected by the different time periods.

Test licenses 2- 6:

- Dealers/End users can demonstrate/test options.
- Service engineers can activate a test licence in the event of a faulty system SD card. This eliminates any production delays while awaiting the delivery of a new licence from the Siemens Hotline.

Additional test licence information:

- The test licence period will only decrement if the system has at least one drive assigned to an axis.

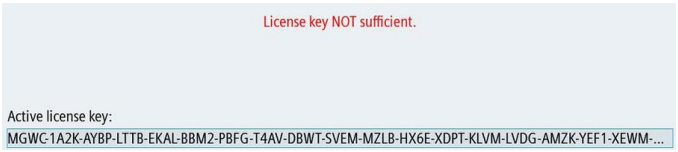
The test licence period:

- Starts with control power on and stops with power off.
- Is paused when the correct licence key for the activated options is entered.
- Is paused if an unlicensed option is deactivated.

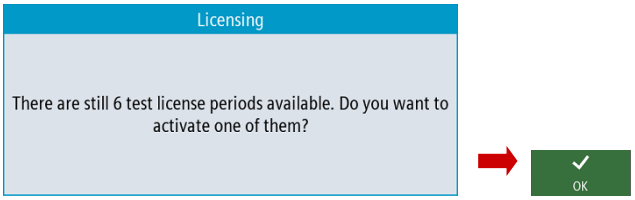
Activating the test licence:

By selecting the ***Activate test licence*** soft-key it is possible to have full functionality of the option.







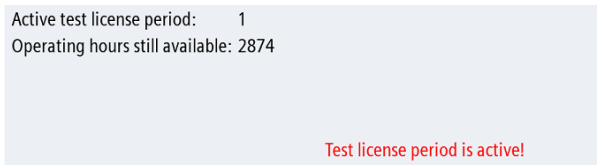


Alarm number ***8084*** will replace the ***8081*** alarm number. This can be cleared with the ***cancel/clear*** button but will periodically reappear as long as a test licence is active. The remaining licence time will be seen in the alarm description.

Alarm 8084:



The main licence screen also shows the status of the test licence:



Notes:

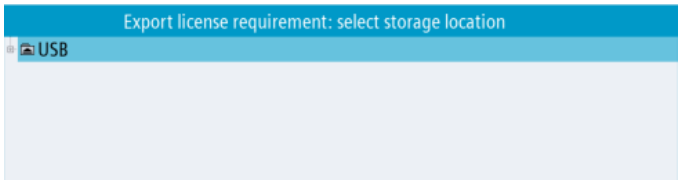
Real Machine: Saving the License Key:

It is possible to save the license number to a file.
The license file has the ***.alm*** extension.
This type of file cannot be read-in to the controller but it can be used in with the Siemens Weblicense manager for licensing of additional options of an existing SD card.

A valid licence key must be already entered and active.



Choose the destination for the License key e.g. a USB stick.





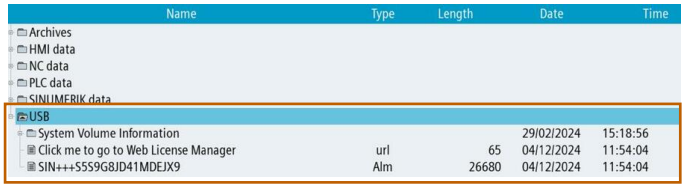
A confirmation message is displayed, momentarily, above horizontal soft-keys:



The licence key is saved as an ***alm*** type file.
The name of the file is generated automatically and contains the serial number of the SD card.
A ***url*** link to the Siemens web licensing website is also created.

The saved licence key can be viewed via the ***System data*** area:





Notes:

Real Machine: Retrieving and Reading in a Licence Key

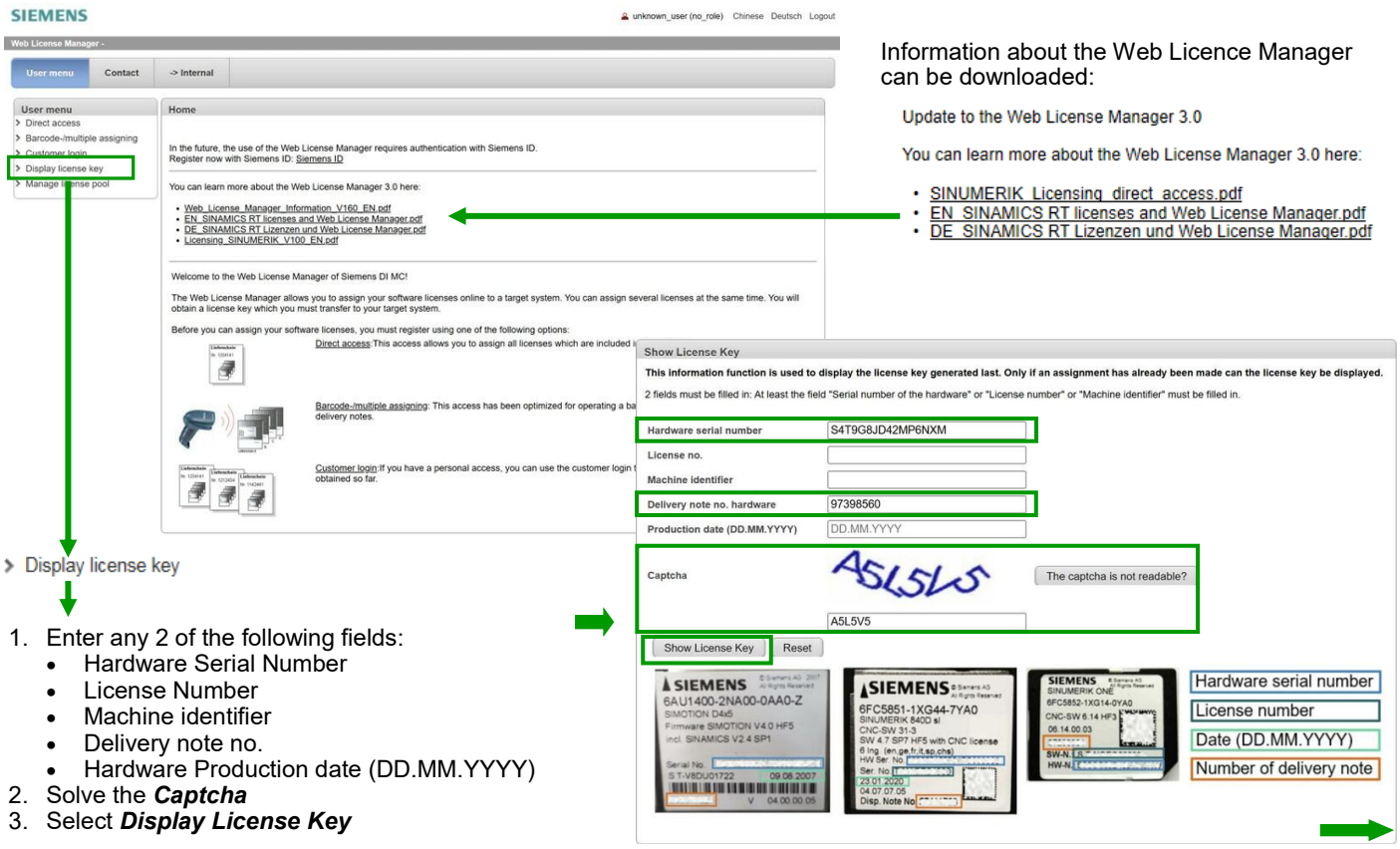
The SD card will arrive with documentation regarding the license key and options. This needs to be kept in a secure place.

The License key should be saved at first commissioning and prior to attempting any work where the NC data might be lost/ reloaded.

If the License key is lost it is possible to retrieve it via the internet. The serial number of the internal SD card will be needed to recover the License key.

The licensing website can be found at:
www.siemens.com/automation/license

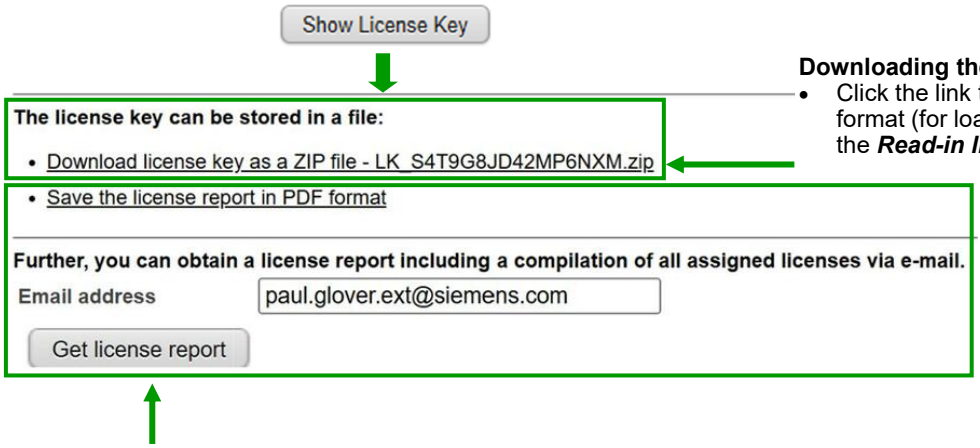
To retrieve a lost key use the ***Show License Key*** option, found in the left hand column.
From the ***Please select*** menu, choose ***Hardware serial number***.
This is the serial number of the internal SD card taken from the license screen shown of the controller.



1. Enter any 2 of the following fields:
   - Hardware Serial Number
   - License Number
   - Machine identifier
   - Delivery note no.
   - Hardware Production date (DD.MM.YYYY)
2. Solve the ***Captcha***
3. Select ***Display License Key***

Notes:

- Select ***Show License Key*** :



Downloading the license key:

- Click the link to download the license file in ***.zip*** format (for loading directly into the controller via the ***Read-in licence key*** soft-key).

Saving the License Report

The full license report can be saved or emailed .pdf format.

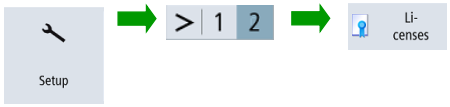
- ***Save the license report in PDF format*** (save report to PC)
- ***Get license report*** (email report)

Restoring the License Key:

Note: this function only works with ***.zip*** type licence files.

Example - reading in the licence key file from a USB stick:

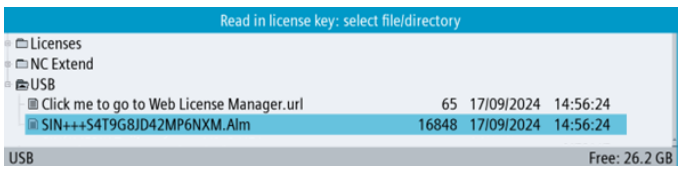
- Copy the ***.zip*** licence key file to a USB stick.
- Insert the USB stick into the controller.
- Locate the licencing screen.



- Select the ***Read in license key*** soft-key:



- Locate the License key file and press the ***OK*** soft-key:





The License key will now appear in the License key box.



Notes:

Example licence certificate:

Real Machine: Assigning New Licenses

Activating an unlicensed option, without a test licence being active, results in alarm number 8081.
If a new option is required it can be licensed via the ***Web Licence Manager*** website.

Example of an unlicensed option that has been set:



- Purchase the required option.

Each purchased option has a licence certificate issued.

The certificate contains the License and dispatch note numbers.

These are required to be able to generate a new license key for a Sinumerik ONE SD card.

It is possible for multiple options of the same type to be assigned to the same license number.

The following example will show how an additional axis is assigned to an existing SD card.



Sequence:

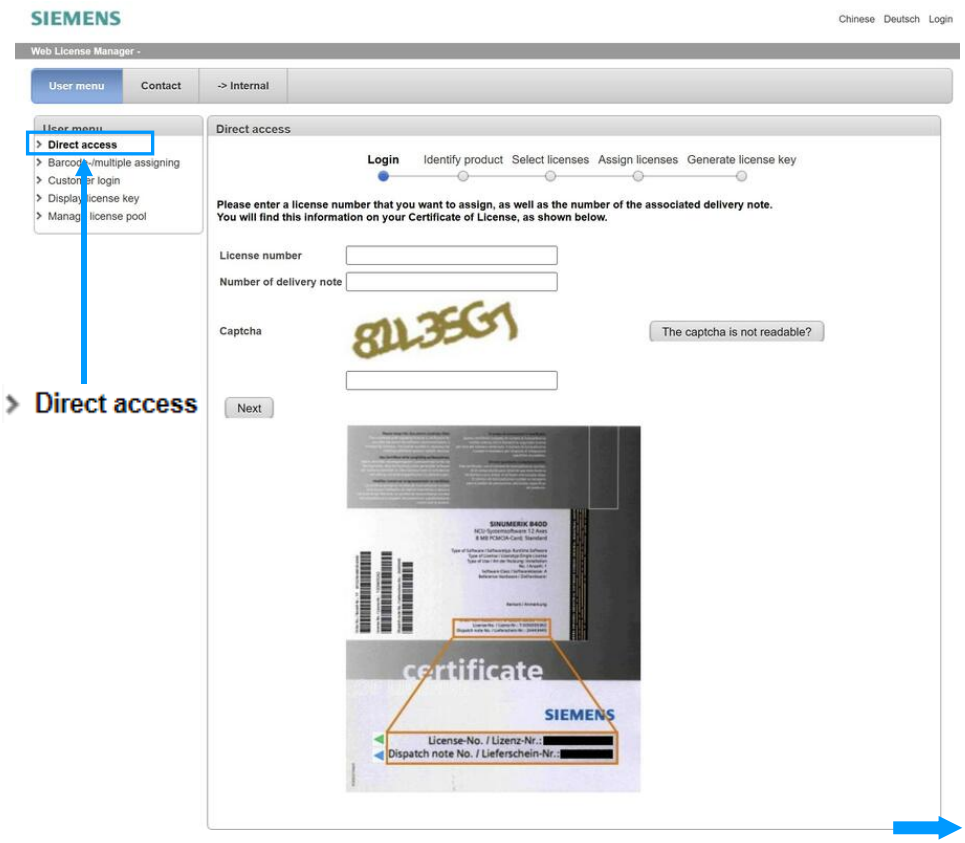
The licensing is achieved by a series of 5 steps via the ***Siemens Web License Manager***.
www.siemens.com/automation/license

The hardware serial number (System SD card serial number) and the hardware type are required.

These can be seen on the licensing screen.

- Select the ***Direct access*** button

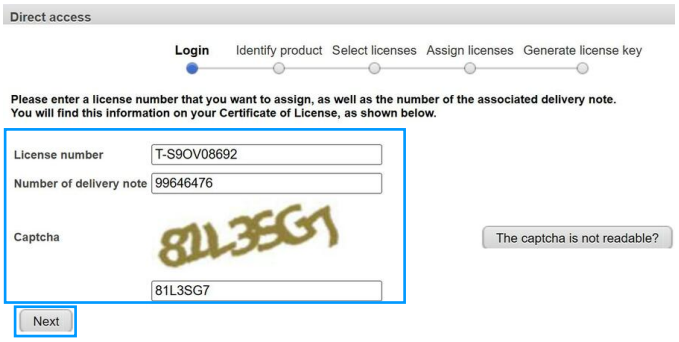
  Note: the ***captcha*** will not be required if it was previously entered during the current licensing session.



Notes:

Step1 - login:

Enter the ***license*** and ***delivery note*** numbers taken from the relevant certificate:

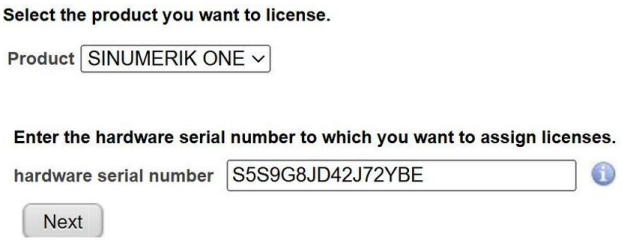


Continue to step 2 with the ***Next*** button.



Step 2 - Identify Product:

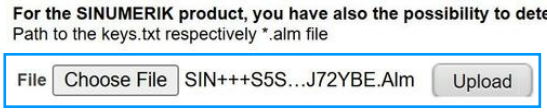
Enter the hardware serial number and select the product.
If a saved license file is available, it can be loaded as an alternative to manually entering the information (see below).



Continue to step 3 with the ***Next*** button.

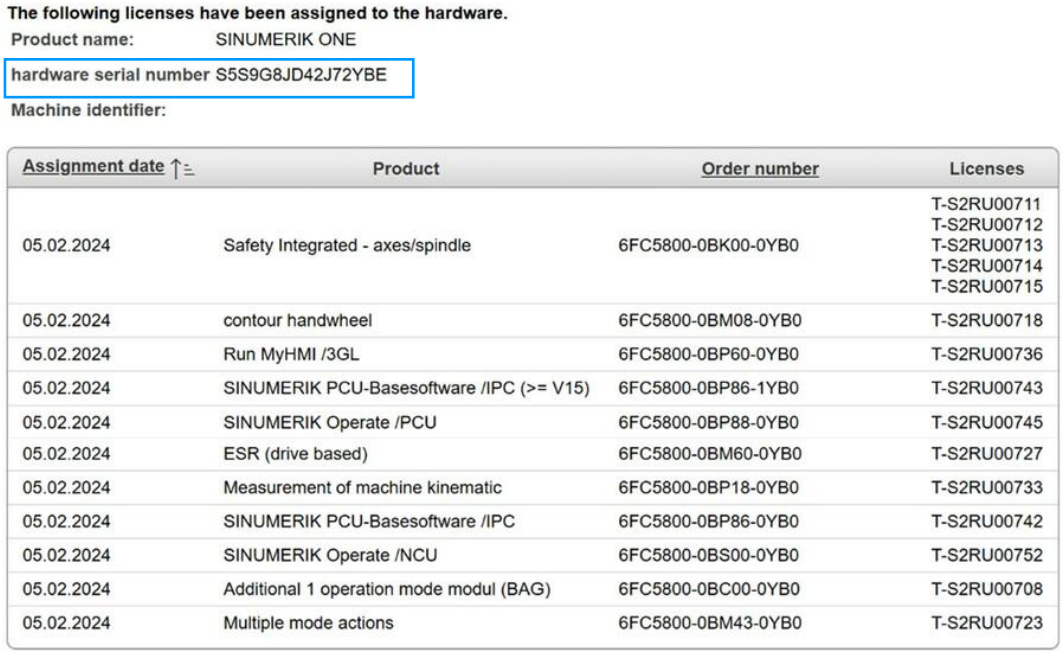


An alternative method of identifying the product is by uploading an ***alm*** **or** ***zip*** type license file via the ***Browse*** and ***Upload*** buttons.



Step 3 - Select Licences:

A list of licenses, already assigned to the card, is displayed.



Continue with the ***Next*** button.



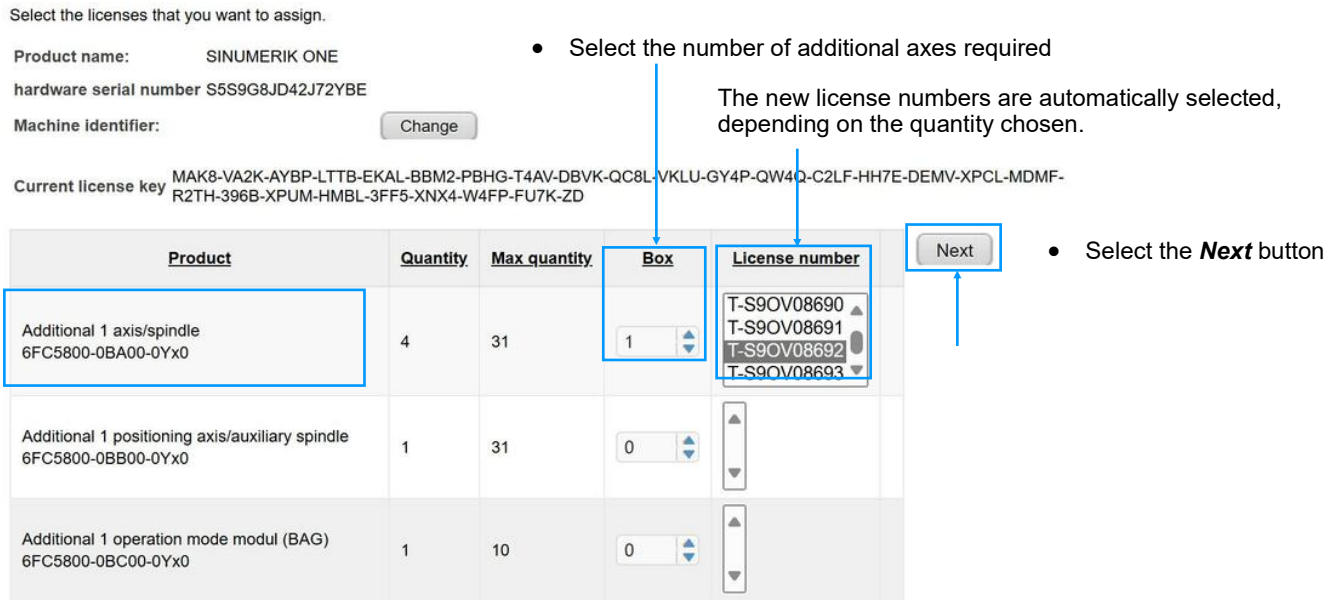
Notes:

Step 3 - Select Licences:

- Select the ***Next*** button.



The current license information of the SD card is listed.
The additional axis option is also on the list.



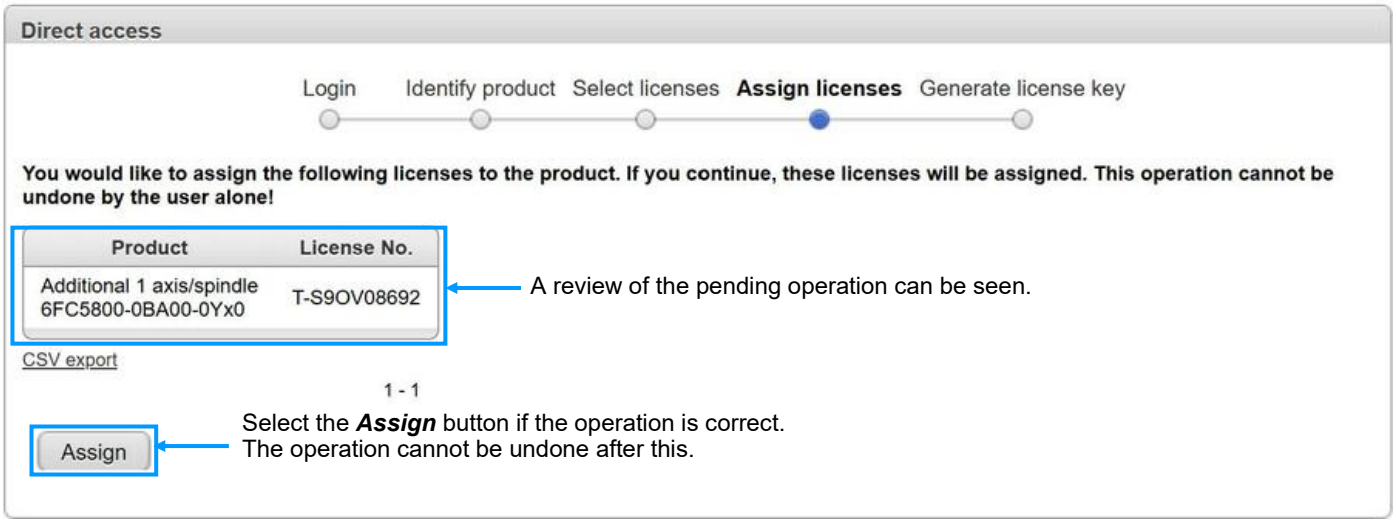
- Select the number of additional axes required

The new license numbers are automatically selected, depending on the quantity chosen.

- Select the ***Next*** button

Step 4 - Assign Licences:

The option is now ready to be assigned:



A review of the pending operation can be seen.

Select the ***Assign*** button if the operation is correct.
The operation cannot be undone after this.

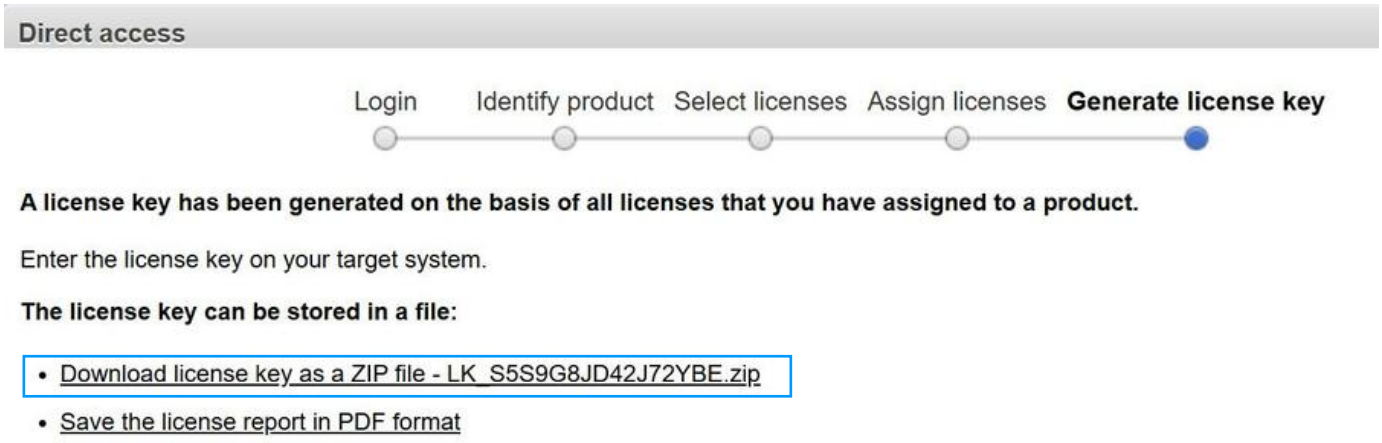
Notes:

Step 5 - Generate/Save License Key

A new license key file has now been generated.
This file can be saved in TWO formats (.zip & .pdf).

- The ***zip*** type license file ***can*** be loaded to the controller
- The ***pdf*** type is a written record of the assigned licenses

Choose the ***zip*** type license file and save it to a suitable location. The file can then be loaded to the controller and the additional option can then be set.

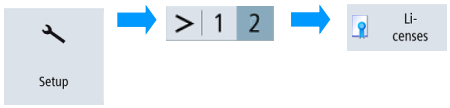


Reading in the License Key:

Note: this function only works with ***.zip*** type licence files.

Example - reading in the licence key file from a USB stick:

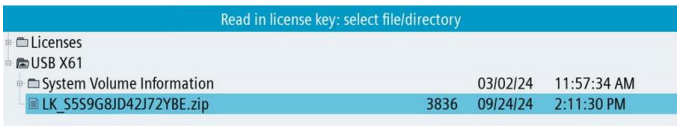
- Copy the ***.zip*** licence key file to a USB stick.
- Insert the USB stick into the controller.
- Locate the licencing screen



- Select the ***Read in license key*** soft-key:



- Locate the License key file and press the ***OK*** soft-key:



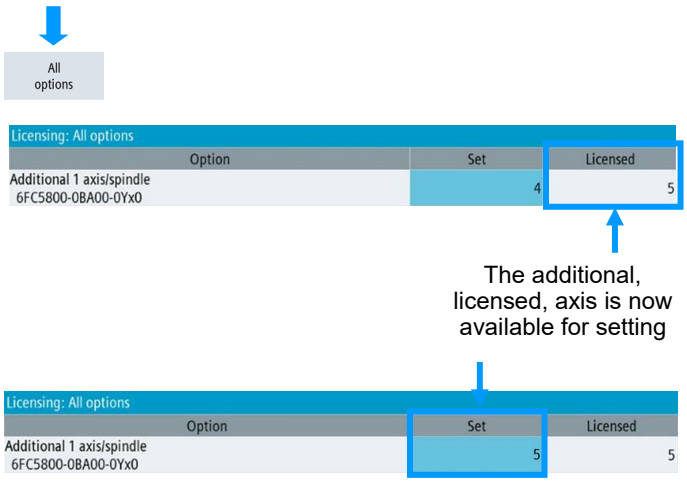


The License key will now appear in the License key box.



Notes:

Checking and setting the additional option:



Then new setting is activated with a ***Reset (po)*** procedure



A109: END

Notes:

Notes:

A111

SINUMERIK ONE

Machine and Setting Data

Module Description:

Machine and Setting data's are used in many ways to influence the operation of a machine.
During commissioning some of the machine data is automatically set while other machine data has to be manually set.
Over time, as the machine mechanically wears it may be necessary to edit certain machine data such as compensation data's.
This module shows the different machine data areas and how to activate the machine data.

Module objective:

Upon completion of this module you will be able to perform the following tasks:

- Locate the different Machine/Setting data areas and be able to carry out adjustments if required.
- You will also be able to create a customised user list containing machine and setting data which are specific to a particular function/area of the machine.

Content:

Machine & Setting Data Overview

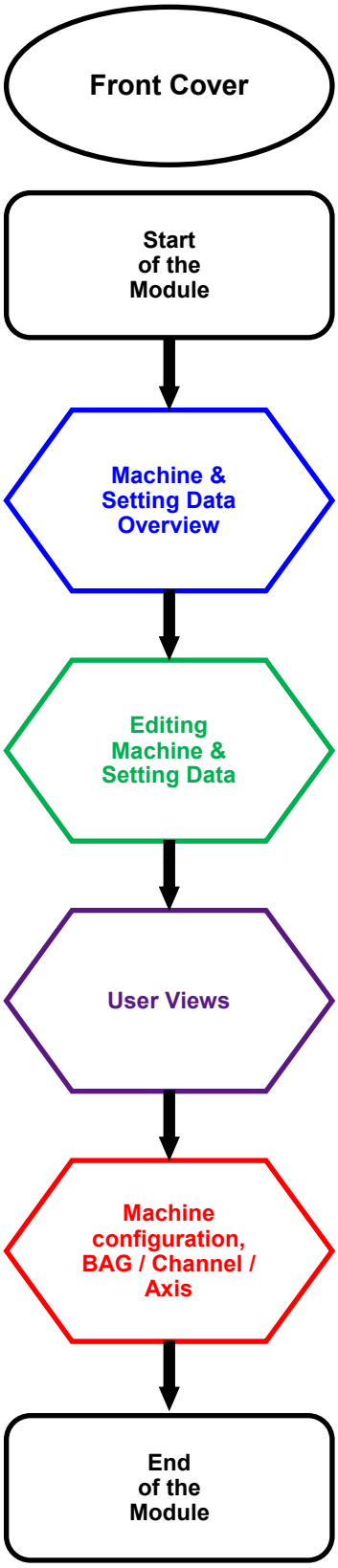
Editing Machine & Setting Data

User Views

Machine configuration, BAG / Channel / Axis

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Machine & Setting Data Overview
Editing Machine & Setting Data
User Views
Machine configuration, BAG / Channel / Axis
End of the Module

Machine & Setting Data Overview

General assignment

The Machine/Setting data are divided into the following ranges:

Range	Description
9000 - 9999	Display machine data
10000 - 19999	General machine data
20000 - 29999	Channel-specific machine data
30000 - 39999	Axis-specific machine data
41000 - 41999	General setting data
42000 - 42999	Channel-specific setting data
43000 - 43999	Axis-specific setting data
51000 - 51999	General cycle machine data
52000 - 52999	Channel-specific cycle machine data
53000 - 53999	Axis-specific cycle machine data
54000 - 54999	General cycle setting data
55000 - 55999	Channel-specific cycle setting data
56000 - 56999	Axis-specific cycle setting data

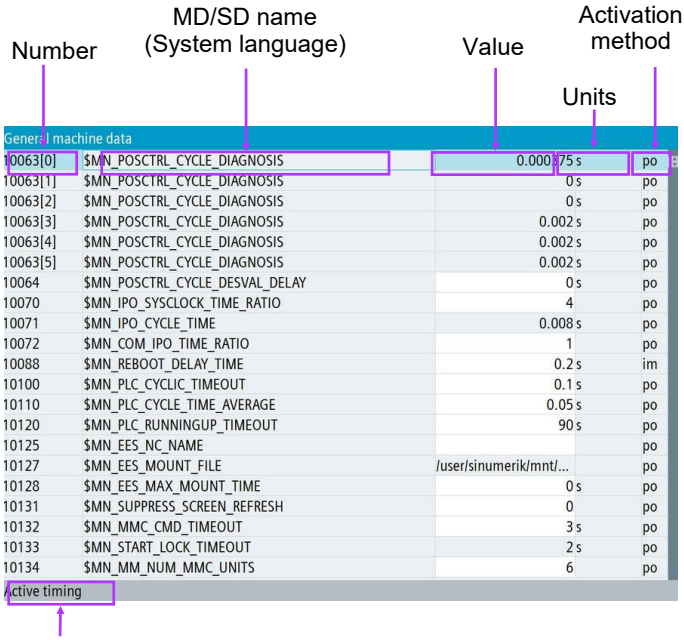
Each range contains the individual machine/setting data. Machine data is often referred to as **MD**, while setting data is referred to as **SD.**

Machine/Setting Data Location:

The following soft-key selection is used to locate the MD/SD area.



The data is listed in numerical order with the MD/SD number in the left column. The following columns show the description/name, value, units and activation requirement.



MD/SD identification:

The MD/SD name contains identifiers so that the system can identify that particular data.

1st character
$ = System variable (Present on all MD & SD)

2nd character (Defines between MD or SD)
M = Machine data
S = Setting data

3rd character (Defines type of MD/SD)
M = Display Machine Data
N = General Machine Data / Setting data
C = Channel Machine Data / Setting data
A = Axis Machine Data / Setting Data

4th character
S = Cycle Data (Defines MD/SD as cycle data)

Examples:
9900 $MM_MD_TEXT_SWITCH Display MD
54600 $SNS_MEA_WP_BALL_DIAM General Cycle SD
55623 $SCS_MEA_EMPERIC_VALUE Channel Cycle SD
30200 $MA_NUM_ENCS Axis MD

Notes:

The following data ranges are used for the machine data:

$MM_	Display machine data
$MN_	General machine data
$MC_	Channel-specific machine data
$MA_	Axis-specific machine data
$MNS_	General cycle machine data
$MCS_	Channel-specific cycle machine data
$MAS_	Axis-specific cycle machine data

The following data ranges are used for the setting data:

$SN_	General setting data
$SC_	Channel-specific setting data
$SA_	Axis-specific setting data
$SNS_	General cycle setting data
$SCS_	Channel-specific cycle setting data
$SAS_	Axis-specific cycle setting data

Indexes:

Certain MD/SD have indexes. The function and number of indexes depends on the data in question.

Example:
General MD **10000 $MN_AXCONF_MACHAX_NAME_TAB**

This MD is used for defining the names of the axes in the system. Each axis name is assigned to an index the MD.

General machine data		
10000[0]	$MN_AXCONF_MACHAX_NAME_TAB	MX1
10000[1]	$MN_AXCONF_MACHAX_NAME_TAB	MZ1
10000[2]	$MN_AXCONF_MACHAX_NAME_TAB	MCS1
10000[3]	$MN_AXCONF_MACHAX_NAME_TAB	MCS3
10000[4]	$MN_AXCONF_MACHAX_NAME_TAB	MQ1
10000[5]	$MN_AXCONF_MACHAX_NAME_TAB	MY1
10000[6]	$MN_AXCONF_MACHAX_NAME_TAB	MZS2
10000[7]	$MN_AXCONF_MACHAX_NAME_TAB	MCS2

Further examples:
30240[0] $MA_ENC_TYPE = The type of encoder connected to the **1st** measuring system input.
30240[1] $MA_ENC_TYPE = The type of encoder connected to the **2nd** measuring system input.

Help Function:

The function of each MD/SD is described in detail via the “Help” key.
The MD/SD first needs to be selected and then the “Help” key pressed. The key is found on the NC keyboard



A detailed description will be displayed. This will include units, limits etc as well as the functionality.

10000	AXCONF_MACHAX_NAME_TAB				N01, N11	K2, F1, G2, F2, K5, M1
-	PowerOn					

List of the machine axis identifiers.
The name of the machine axis is entered in this MD.
In addition to the fixed, defined machine axis identifiers "AX1", "AX2" ..., user-defined identifiers for the machine axes can also be assigned in this data.
The identifiers defined here can be used parallel to the fixed, defined identifiers for addressing axial data (e.g. MD) and machine axis-related NC functions (reference point approach, axial measurement, travel to fixed stop).

Special cases:



Activation of the MD/SD:

There are four methods available for activating the MD/SD.
The method depends on the MD/SD in question.
The right hand column of the screen displays the required method.

po	Active via “Reset (po)” soft-key or controller power off/on.	
cf	Active on selection of “Set MD active (cf)” soft key.	
re	Active on selection of the appropriate PLC interface signal, usually via MCP “Reset” key.	
im	Active immediately the “Input” key is selected to confirm the value.	



Notes:

Units for MD/SD

The function of the MD/SD determines the units in use.
The following table shows the abbreviations and descriptions of the units.

Unit	Description
mm/sec	Millimetres/second: Linear velocity/speed
rpm	Revolutions/minute: rotational velocity/speed
m/s²	Meters/second-squared: Linear acceleration
rev/s²	revolutions/second-squared: rotary acceleration
kgm²	Kilogram/meter-squared: Moment of inertia
s	seconds: measurement of time.
Hz	Cycles/second: Frequency.
deg	Degrees: Angular measurement.

Unit	Description
mm/rev	millimetres/revolution: linear feed-rate determined by a rotary axis.
m/s³	Meters/second-cubed: linear jerk
rev/s³	revolutions/second-cubed: rotary jerk
Kg	Kilogram: Measurement of mass
mm	millimetre: Linear measurement.
Nm	Newton-metre
mH	milli-Henry: Measurement of inductivity
A	Amperes (Amps): Electrical current
V	Volts: Electrical potential

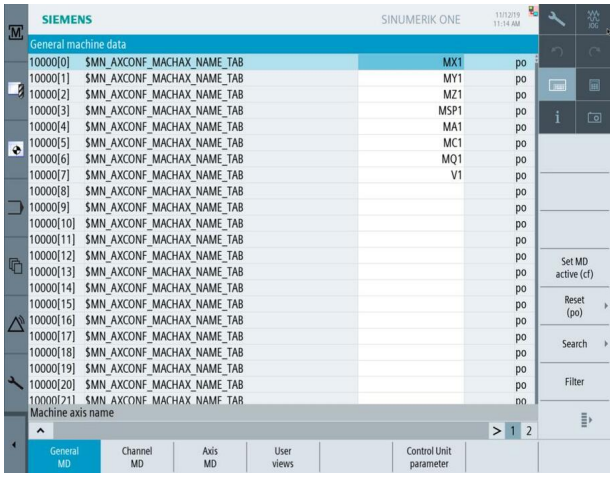
Some MD/SD have no units associated with them. In this case use the “Help” function to determine the data type required.

Editing Machine & Setting Data

To locate the machine data screens first ensure that the relevant access level is active (Module A107 Protection Levels) then use the following key sequence:-

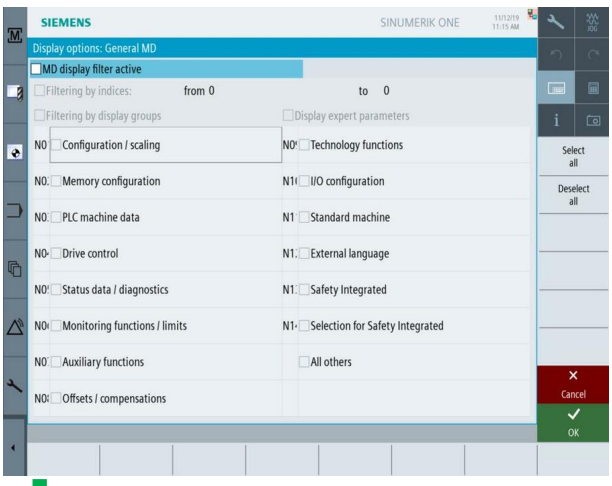


The default is for the General Machine screen to be displayed.



The machine data which are available for viewing can be restricted via the “Display options” soft-key





Notes:

The Machine & Setting Data is separated into different groups.

General MD — General MD and general Cycle MD.

Channel MD — Channel-specific MD and channel-specific Cycle MD.

Axis MD — Axis-specific MD and axis-specific Cycle MD.

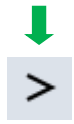
User views — User Views- Allows the user to create list of specific machine data.

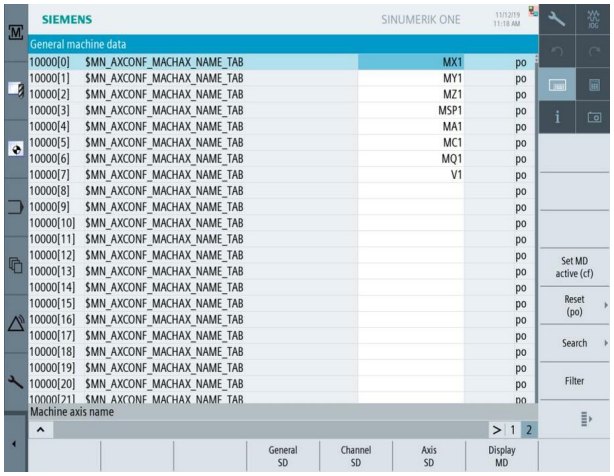
Control Unit parameter — Control Unit MD.

Infeed parameter — Infeed parameter (ALM/SLM-only DriveCLiQ types).

Drive parameter — Drive MD (Motor Modules)

For further data the “Menu Extension” key must be selected. This allows access to the Setting Data and also the Display Machine Data.



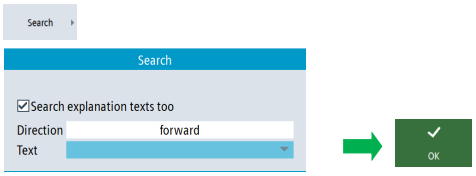


General SD — General SD and general Cycle SD.

Channel SD — Channel-specific SD and channel-specific Cycle SD.

Axis SD — Axis-specific SD and axis-specific Cycle SD.

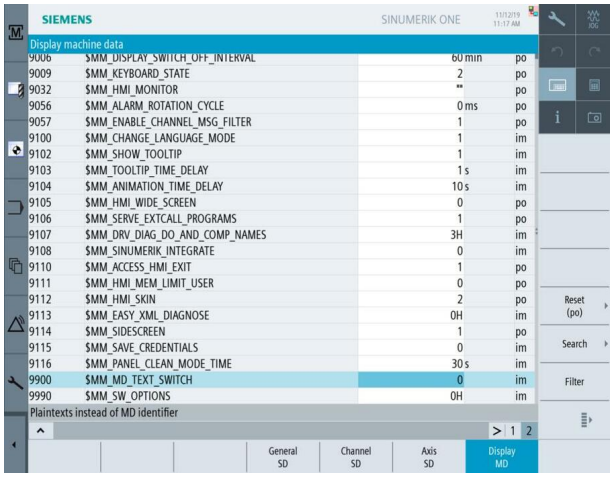
Display MD — Display MD.



When the correct MD has been found, a value can be entered. In this example we are going to change Display MD 9900 from 0 to 1. This will change the MD/SD description from system language to plain text.

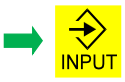
The correct machine data should first be located. This is done by either using the page/cursor keys or the search function.

The search function is activated with the “Search” soft key. If the Machine Data number is known this can be typed into the dialogue box. It is also possible to enter part or all of the MD name in text format.

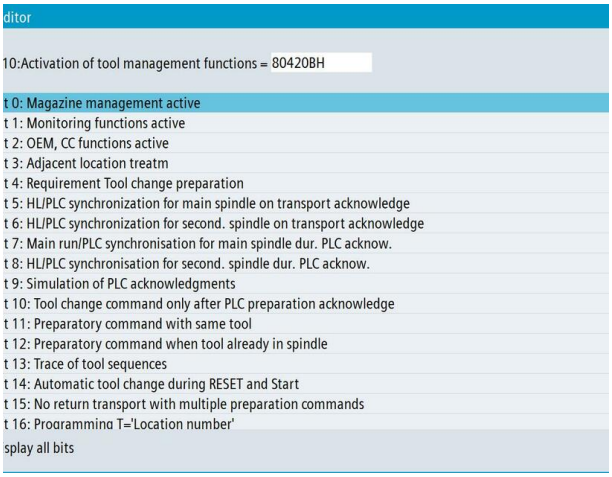


Notes:

The correct machine data has to be highlighted. The number 1 is entered and confirmed with the “Input” key.



On pressing the “Input” key the text of the machine data changes immediately (see below). This will apply to all MD/SD.

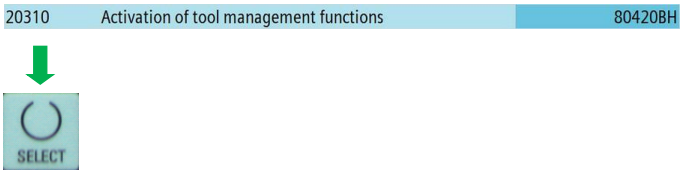


Some MD/SD have no units associated with them. In this case use the “Help” function to determine the data type required.

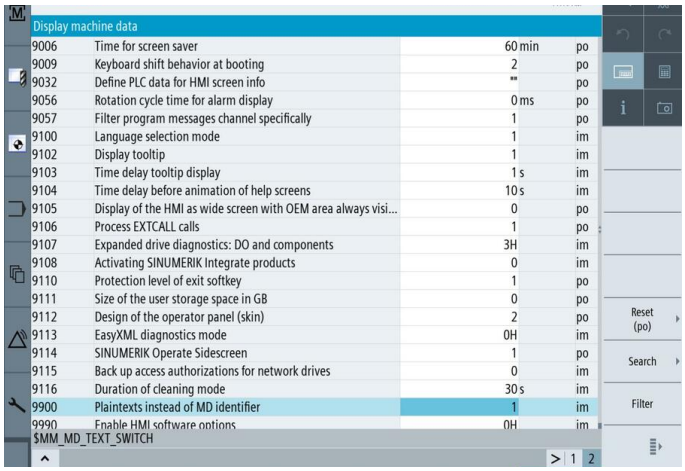
MD/SD with “Bit patterns”

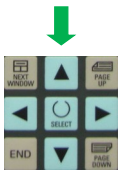
If the letter “H” appear as the last digit in the “Value” column the MD/SD requires a value in Hexadecimal format. A bit pattern can be viewed.
To view the bit pattern, highlight the MD/SD and then use the “Select” key found on the NC keyboard.

Example: Channel MD 20310



A descriptive list of each individual bit is displayed. Use the cursor keys to highlight the required bit, use the select key to activate/deactivate the bit.





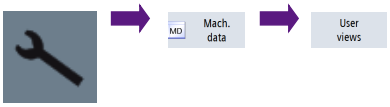


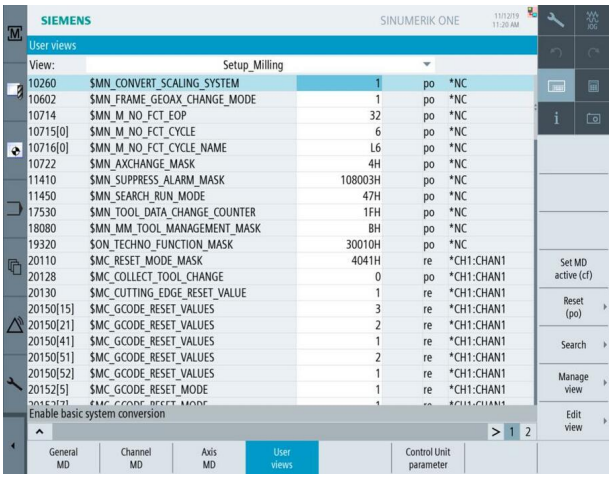
REMEMBER:-
Create a data backup.
Any changed MD/SD should be saved in an archive.
See Module A137.

Notes:

User Views

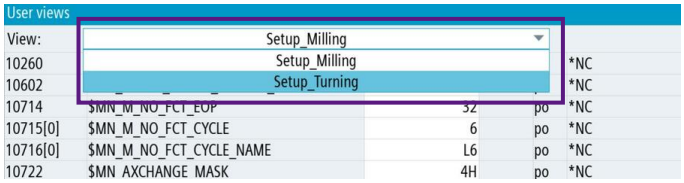
The “User views” function allows a customised list of machine data to be made. MD/SD can be listed that are of importance to a specific task/function of the machine.



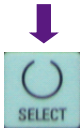


By default on the control there are two “User view” lists. “Setup_Milling” and “Setup_Turning”

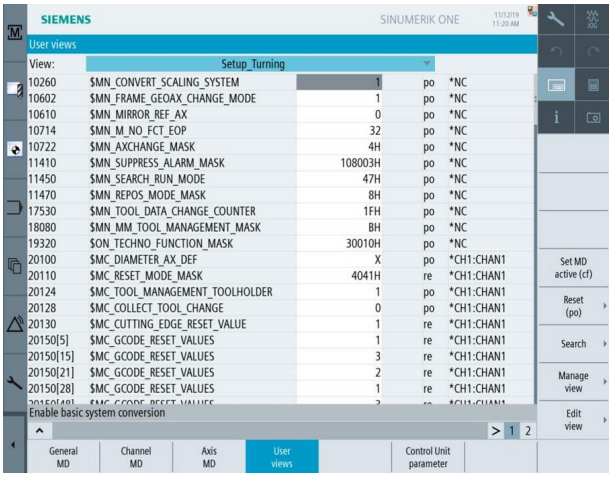
These views list all MD/SD required for setting the technology Milling or Turning.



Module A121 Setup Milling and A509 Setup Turning describe the minimum setting for the respective MD and SD that should be applied in each case.
When a bit pattern is described these bits MUST be set, in addition to any that have been set by the OEM.



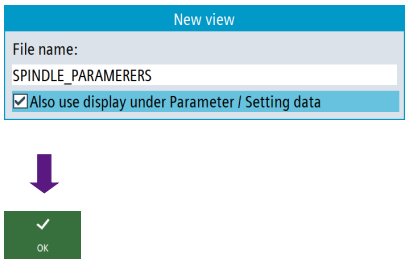
The “SELECT” button can be to show the Setup_Turning view.



Edit existing machine data list or create new machine data list.

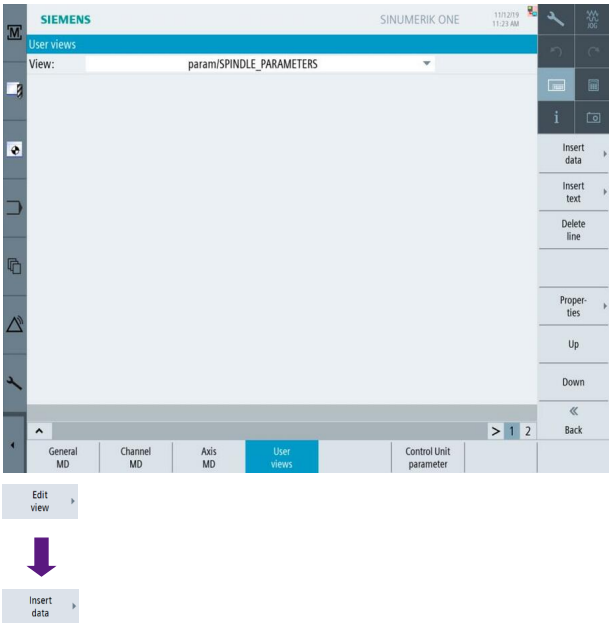


Enter a name for the list e.g. SPINDLE_PARAMETERS. The “User views” soft-key will also be displayed in the “Parameter/Setting data” area if the option is selected.

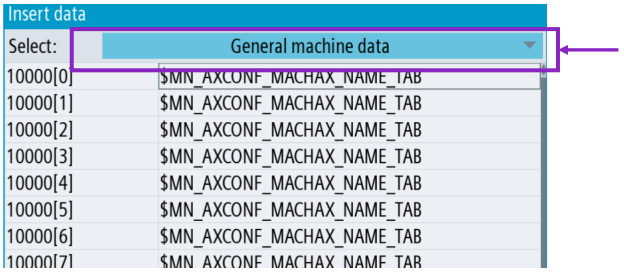


Notes:

The new entry will be displayed in the View menu.
It is now possible to add the required MD/SD.

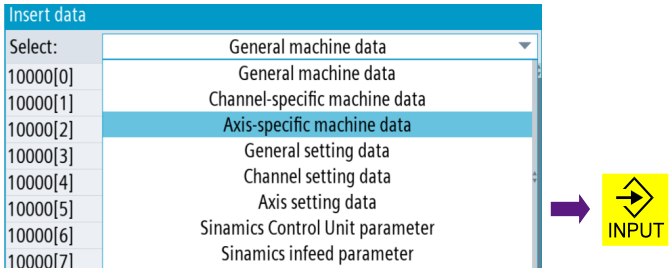


The General machine data will be displayed by default.
A list of all types of data can be viewed on the “Select” list

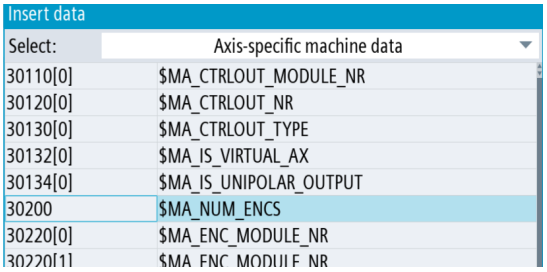


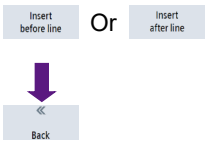
In the following example Axis MD for the spindle will be added.

The required MD/SD type is selected from the list and confirmed with the “Input” key.

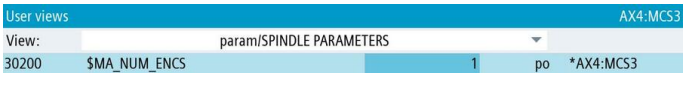


Select the data to be added to the list.



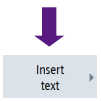


The selected MD will appear on the screen.



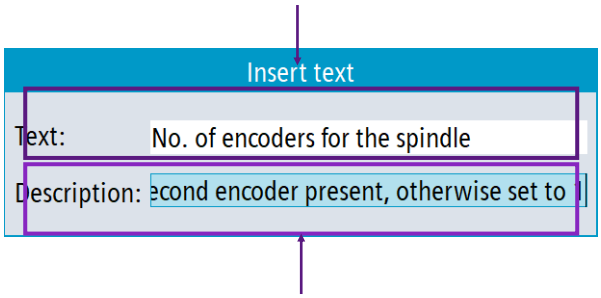
Repeat the procedure to add further data to the list.

Text for information purposes can be added:



The text entered in the “Text” area will be displayed on the list. The text entered in the “Description” area is displayed at the bottom of the screen.

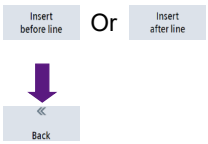
Text to be displayed on the list.
(Maximum 30 displayable characters)



Detailed text to be displayed at the bottom of the screen. (Maximum 80 characters)

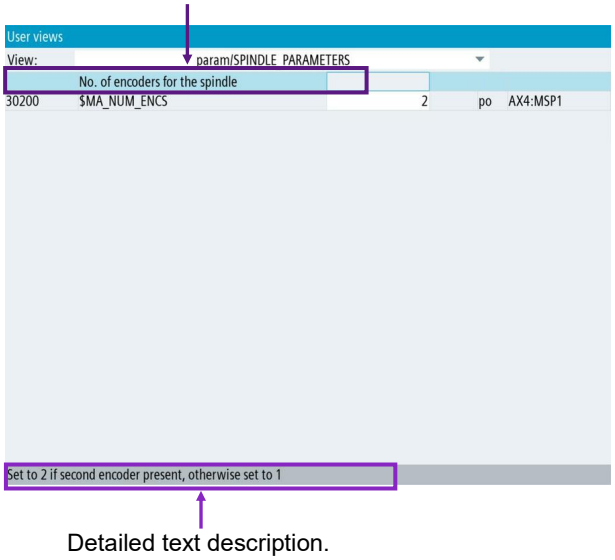
Notes:

The position of the text line can be selected when editing is complete.



Example of a list of spindle specific MD complete with descriptive texts:

Short text description.

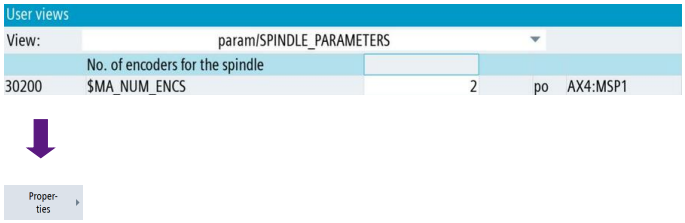


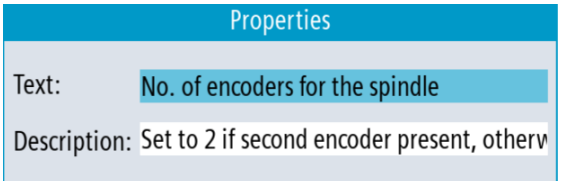
Detailed text description.

Modifying a MD/SD entry:

The entries on the list can be modified by selecting the “Properties” soft-key.

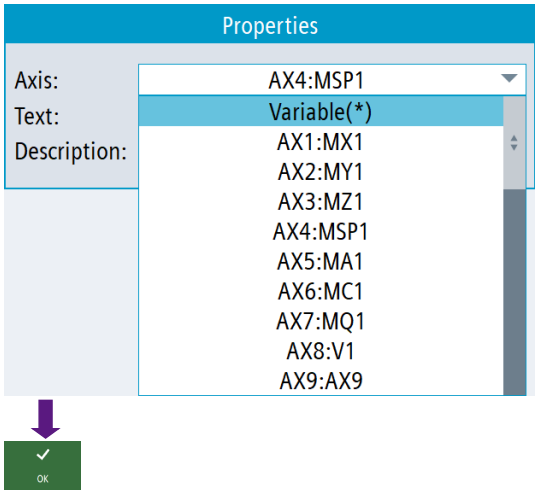
Select the entry to be modified and press the “Properties” soft -key.



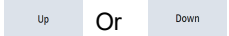


To change the order that the items appear on the list, first select the item to be moved and then move it up/down the list via the “Up” or “Down” soft-keys.

Each press will move the item one position.



The changes that can be made will depend on the MD/SD in question.
In this example, a different axis for the MD has been selected.



Any unwanted items are removed via the “Delete” Soft-key.



Notes:

Machine configuration, BAG / Channel / Axis

After HMI configuration the NCK should be configured.

This NCK configuration includes :

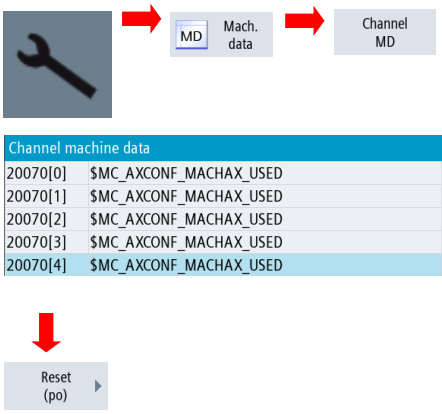
- Number of Mode Groups
- Number of Channels / assignment
- Number of Axis / Spindles
- Axis names
- Assignment of Axis

Note:

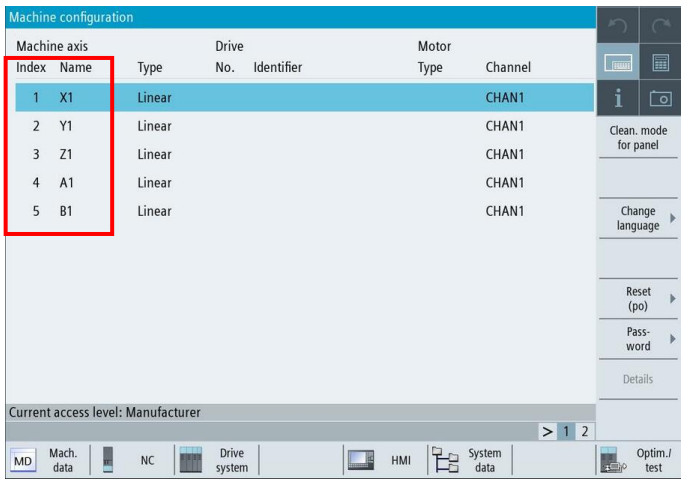
In our example a 4 axis milling machine with one spindle will be configured.

The number of machine axis / spindles therefore has to be increased to 5. This is done by entering the additional axis numbers in the **Channel Machine Data 20070[0....n]**, in the respective index.

In order to activate the new axis, an NCK reset has to be performed.



On the machine configuration screen it can bee seen that the axes are available with default names.



Notes:

Axis Names:

The axis names can now be changed according to the requirement of the machine tool builder.

In this example the following names and axis types will be used:

MX1	Machine axis 1	Linear
MY1	Machine axis 2	Linear
MZ1	Machine axis 3	Linear
MSP1	Machine axis 4	Rotary / Spindle
MA1	Machine axis 5	Rotary

These are the names of the axis which will be seen in Jog/Ref screen at every power on of the machine, and are not the axis names which appear in the part program, an exception to this is when directing the axis to go to reference position in the NC part program, other exceptions exist and can be found in the NC part program guide.

These names are defined in the corresponding **General machine data 10000[0....n].**

Naming the axes in the General Machine Data area:

General machine data

10000[0]	$MN_AXCONF_MACHAX_NAME_TAB	MX1	po
10000[1]	$MN_AXCONF_MACHAX_NAME_TAB	MY1	po
10000[2]	$MN_AXCONF_MACHAX_NAME_TAB	MZ1	po
10000[3]	$MN_AXCONF_MACHAX_NAME_TAB	MSP1	po
10000[4]	$MN_AXCONF_MACHAX_NAME_TAB	MA1	po

The following naming convention is used for the part programming:

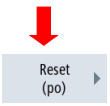
X1	Channel axis 1	Linear
Y1	Channel axis 2	Linear
Z1	Channel axis 3	Linear
SP1	Channel axis 4	Rotary / Spindle
A	Channel axis 5	Rotary

These names are defined in the corresponding **Channel machine data 20080[0....n].**

Naming the axes in the Channel Machine Data area:

Channel machine data — CH1:CHAN1

20080[0]	$MC_AXCONF_CHANAX_NAME_TAB	X1	po
20080[1]	$MC_AXCONF_CHANAX_NAME_TAB	Y1	po
20080[2]	$MC_AXCONF_CHANAX_NAME_TAB	Z1	po
20080[3]	$MC_AXCONF_CHANAX_NAME_TAB	SP1	po
20080[4]	$MC_AXCONF_CHANAX_NAME_TAB	A	po



It can bee seen in the following screenshot that although the required axes are available, they are of not the required axis type.
The axes are set as linear by default and have to be converted to their required type using the axis machine data.

The A axis has to be converted to the Rotary type and the SP1 has to be converted also to type “Rotary“ and be assigned as a Spindle.

Machine configuration

Index	Machine axis Name	Type	Drive No.	Identifier	Motor Type	Channel
1	MX1	Linear				CHAN1
2	MY1	Linear				CHAN1
3	MZ1	Linear				CHAN1
4	MSP1	Linear				CHAN1
5	MA1	Linear				CHAN1

Rotary Axes & Spindles:

The following axis machine data: **30300**, **30310** and **30320** are necessary in order to define an axis as rotary.
These settings are also the prerequisite setting for spindles.

Axis machine data — AX4:MSP1

30300	$MA_IS_ROT_AX	0	po
30310	$MA_ROT_IS_MODULO	0	po
30320	$MA_DISPLAY_IS_MODULO	0	po

The machine data should be changed to the value of 1.
Upon changing MD30300, the alarm message 4070 will be displayed, indicating that the measuring system units will be changed at the next NCK reset.
The default linear measuring unit is the mm, this will changed to degrees for rotary axes / spindles.

Axis machine data — AX4:MSP1

30300	$MA_IS_ROT_AX	1	po
30310	$MA_ROT_IS_MODULO	1	po
30320	$MA_DISPLAY_IS_MODULO	1	po

Notes:

Repeat for each rotary axis, i.e. MA1 for the training equipment.

MA1 axis defined as rotary:

Axis machine data			AX5:MA1
30300	$MA_IS_ROT_AX	1	po
30310	$MA_ROT_IS_MODULO	1	po
30320	$MA_DISPLAY_IS_MODULO	1	po

Defining the MSP1 axis as a Spindle:

Following the definition of the rotary axes, the MSP1 axis can be assigned as a Spindle.
This definition is made with **Axis Machine Data 35000**.
A value other than 0 (1 to 20) defines the axis as a spindle.
The value depends on the number of spindles/machine configuration.
The training equipment has one spindle to be defined, therefore set MSP1 Axis MD35000 = 1.

Axis machine data			AX4:MSP1
35000	$MA_SPIND_ASSIGN_TO_MACHAX	1	po

Defining the Master Spindle:

The master spindle is defined with **Channel Machine Data 20090.**
The value entered (1 to 20) is linked to the spindle defined in Axis MD35000.
By default, Channel Machine Data 20090 = 1, therefore, MSP1 is defined as the master Spindle for the defined channel.
As a result, Spindle MSP1 is programmed with the address "S"

Channel machine data			CH1:CHAN1
20090	$MC_SPIND_DEF_MASTER_SPIND	1	po

Reset (po)

Checking the Configuration:

The machine configuration screen displays the defined axes.

It should be noted however that this is only the basic machine configuration and not the end requirement. Many more machine data have to be changed in order to achieve this.

It can be seen that our main configuration is reached.

3 Linear axes
1 Spindle
1 Rotary axis

Machine configuration screen:

Machine configuration

Machine axis Index	Name	Type	Drive No.	Identifier	Motor Type	Channel
1	MX1	Linear				CHAN1
2	MY1	Linear				CHAN1
3	MZ1	Linear				CHAN1
4	MSP1	Spindle				CHAN1
5	MA1	Rotary				CHAN1

The configured axes can also be seen on the machine screen:

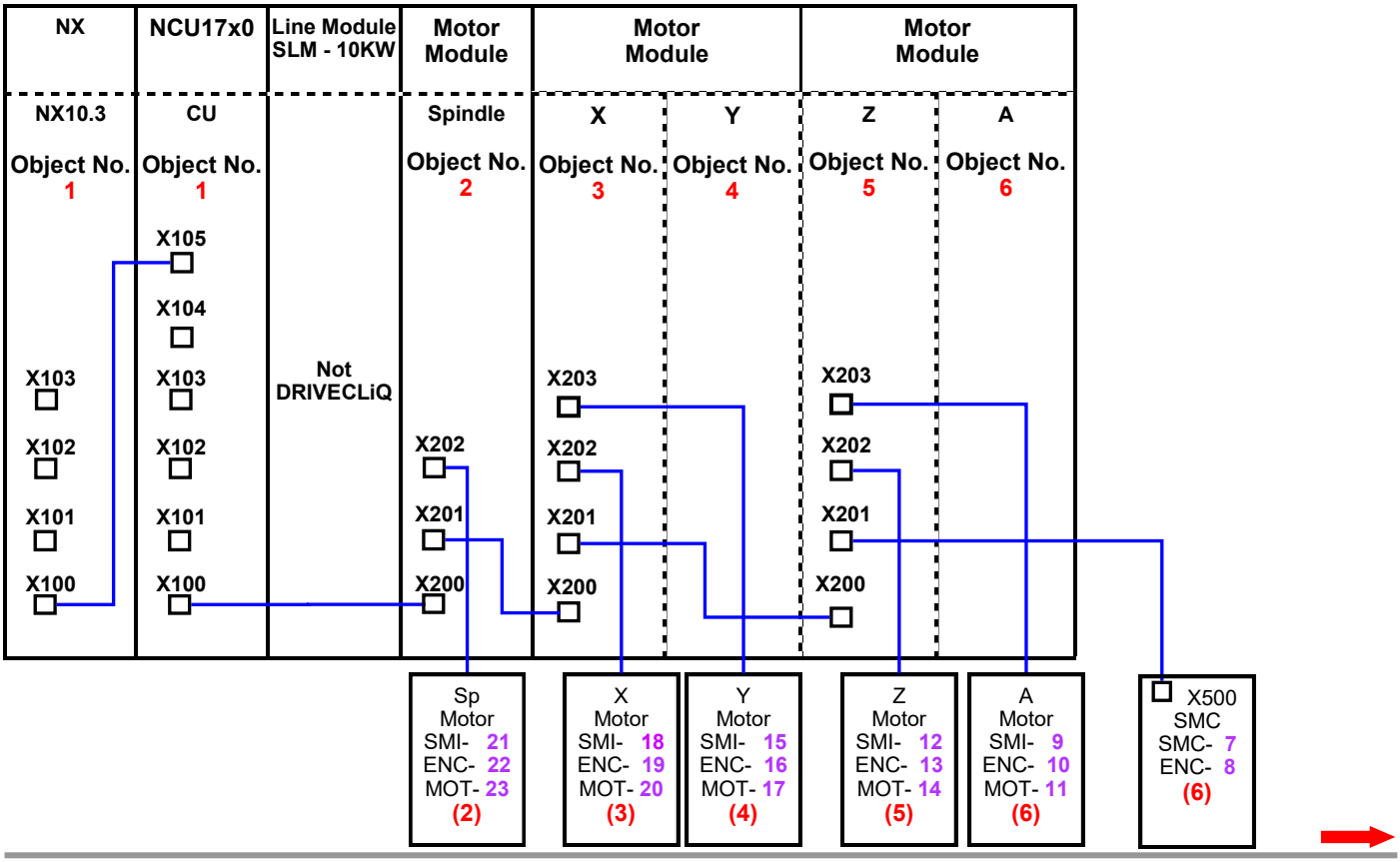


Notes:

Topology

Example - 5 axis system (Spindle, X, Y, Z, A) Commissioned with the ***automatic topology recognition*** procedure.

The topology of the training equipment is shown below.
Objects are numbered in **red** text.
Components are numbered in **purple** text.



Notes:

Defining of axis reference MD

The following axis machine data: **34000**, **34200** and **34210** are necessary in order to define referencing of each axis. These settings are also required for the spindle.

Axis machine data		AX1:MX1/SERVO_3.3:3	
34000	$MA_REFP_CAM_IS_ACTIVE	1	re
34200[0]	$MA_ENC_REFP_MODE	1	po
34210[0]	$MA_ENC_REFP_STATE	0	im

Axis machine data		AX2:MY1/SERVO_3.3:4	
34000	$MA_REFP_CAM_IS_ACTIVE	1	re
34200[0]	$MA_ENC_REFP_MODE	1	po
34210[0]	$MA_ENC_REFP_STATE	0	im

Axis machine data		AX3:MZ1/SERVO_3.3:5	
34000	$MA_REFP_CAM_IS_ACTIVE	1	re
34100[0]	$MA_REPF_SET_POS	649	re
34200[0]	$MA_ENC_REFP_MODE	1	po
34210[0]	$MA_ENC_REFP_STATE	0	im

Axis machine data		AX4:MSP1/SERVO_3.3:2	
34000	$MA_REFP_CAM_IS_ACTIVE	0	re
34200[0]	$MA_ENC_REFP_MODE	1	po
34200[1]	$MA_ENC_REFP_MODE	1	po
34210[0]	$MA_ENC_REFP_STATE	0	im
34210[1]	$MA_ENC_REFP_STATE	0	im

Axis machine data		AX5:MA1/SERVO_3.3:6	
34000	$MA_REFP_CAM_IS_ACTIVE	0	re
34200[0]	$MA_ENC_REFP_MODE	0	po
34210[0]	$MA_ENC_REFP_STATE	0	im

Note:

For an explanation of MD 34210[0] = 2 (Absolute encoder) for the A axis, please refer to

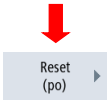
Appendix_J_EN Encoder Adjustments

The following axis machine data: 36100, 36110 sets the software limit switches.

Axis machine data		AX1:MX1/SERVO_3.3:3		
36100	$MA_POS_LIMIT_MINUS	-750	mm	cf
36110	$MA_POS_LIMIT_PLUS	750	mm	cf

Axis machine data		AX2:MY1/SERVO_3.3:4		
36100	$MA_POS_LIMIT_MINUS	-300	mm	cf
36110	$MA_POS_LIMIT_PLUS	300	mm	cf

Axis machine data		AX3:MZ1/SERVO_3.3:5		
36100	$MA_POS_LIMIT_MINUS	0	mm	cf
36110	$MA_POS_LIMIT_PLUS	650	mm	cf



The following machine data allows for the logical I/O address of the CU.

General machine data			
13120[1]	$MN_CONTROL_UNIT_LOGIC_ADDRESS	6530	po

The following machine data allows for the continuous operation for the JOG-INC and reference point approach.

General machine data			
11300	$MN_JOG_INC_MODE_LEVELTRIGGRD	0	po

The following axis machine data: as set out below are necessary in order to define velocities for each axis. These settings are also required for the spindle and rotary axis

Axis machine data		AX1:MX1/SERVO_3.3:3	
32010	$MA_JOG_VELO_RAPID	5000	re

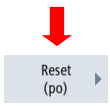
Axis machine data		AX2:MY1/SERVO_3.3:4	
32010	$MA_JOG_VELO_RAPID	5000	re

Axis machine data		AX3:MZ1/SERVO_3.3:5	
32010	$MA_JOG_VELO_RAPID	5000	re

Axis machine data		AX4:MSP1/SERVO_3.3:2	
32000	$MA_MAX_AX_VELO	1000	cf
32010	$MA_JOG_VELO_RAPID	100	re
35100	$MA_SPIND_VELO_LIMIT	1000	re
35110[0]	$MA_GEAR_STEP_MAX_VELO	1000	cf
35110[1]	$MA_GEAR_STEP_MAX_VELO	1000	cf
35130[0]	$MA_GEAR_STEP_MAX_VELO_LIMIT	1000	cf
35130[1]	$MA_GEAR_STEP_MAX_VELO_LIMIT	1000	cf
36200[0]	$MA_AX_VELO_LIMIT	1100	cf
36200[1]	$MA_AX_VELO_LIMIT	1100	cf

Notes:

Axis machine data		AX5:MA1/SERVO_3.3:5	
32000	$MA_MAX_AX_VELO	100	cf
32010	$MA_JOG_VELO_RAPID	50	re
32020	$MA_JOG_VELO	10	re
36200[0]	$MA_AX_VELO_LIMIT	110	cf
36200[1]	$MA_AX_VELO_LIMIT	110	cf



Note:

To allow the drive data to become visible, complete module A113_EN Creating a PLC Program

The following axis machine data: **30130, 30200, 31020, 30240** and **36400** are necessary in order to define setpoint output and encoder type of each axis.
These settings are also required for the spindle.

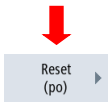
Axis machine data		AX1:MX1/SERVO_3.3:3	
30130[0]	$MA_CTRLOUT_TYPE	1	po
30240[0]	$MA_ENC_TYPE	1	po

Axis machine data		AX2:MY1/SERVO_3.3:4	
30130[0]	$MA_CTRLOUT_TYPE	1	po
30240[0]	$MA_ENC_TYPE	1	po

Axis machine data		AX3:MZ1/SERVO_3.3:5	
30130[0]	$MA_CTRLOUT_TYPE	1	po
30240[0]	$MA_ENC_TYPE	1	po

Axis machine data		AX4:MSP1/SERVO_3.3:2	
30130[0]	$MA_CTRLOUT_TYPE	1	po
30200	$MA_NUM_ENC	2	po
30240[0]	$MA_ENC_TYPE	1	po
30240[1]	$MA_ENC_TYPE	1	po
31020[0]	$MA_ENC_RESOL	2048	po
31020[1]	$MA_ENC_RESOL	1024	po
36400	$MA_CONTOUR_TOL	10 °	cf

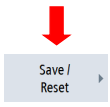
Axis machine data		AX5:MA1/SERVO_3.3:6	
30130[0]	$MA_CTRLOUT_TYPE	1	po
30240[0]	$MA_ENC_TYPE	4	po
31020[0]	$MA_ENC_RESOL	512	po



The following drive parameters: **p408** are necessary in order to define encoder pulses of spindle and A-axis.

Drive parameters		AX4:MSP1/SERVO_3.3:2	
p408[0]	Rotary encoder pulse number	2048	po
p408[1]	Rotary encoder pulse number	1024	po

Drive parameters		AX5:MA1/SERVO_3.3:6	
p408[0]	Rotary encoder pulse number	512	po



Using CMVM, go to the “Mechanical properties” and change the spindle axis data.

Mechanical properties

Servo selection	Axis type	Load gear numerator
MSP1 / SERVO_3.3:3	Ball screw drive	1
MX1 / SERVO_3.3:4	Ball screw drive	1
MY1 / SERVO_3.3:5	Ball screw drive	1
MZ1 / SERVO_3.3:6	Ball screw drive	1
MA1 / SERVO_3.3:7	Ball screw drive	1
MB1 / SERVO_3.3:8	Ball screw drive	1

By default, all axis are given an “Axis type” of “Ball screw drive”.

In the case of a spindle, it must be changed to rotary so that a second encoder can be used.

Mechanical properties

Servo selection	Axis type	Load gear numerator
MSP1 / SERVO_3.3:3	Ball screw drive	1
MX1 / SERVO_3.3:4	Ball screw drive	1
MY1 / SERVO_3.3:5	Ball screw drive	1
MZ1 / SERVO_3.3:6	Ball screw drive	1
MA1 / SERVO_3.3:7	Ball screw drive	1
MB1 / SERVO_3.3:8	Ball screw drive	1

Using the pull down menu, select “Rotary”.

Mechanical properties

Servo selection	Axis type	Load gear numerator
MSP1 / SERVO_3.3:3	Rotary	1
MX1 / SERVO_3.3:4	Ball screw drive	1
MY1 / SERVO_3.3:5	Ball screw drive	1
MZ1 / SERVO_3.3:6	Ball screw drive	1
MA1 / SERVO_3.3:7	Ball screw drive	1
MB1 / SERVO_3.3:8	Ball screw drive	1

Notes:



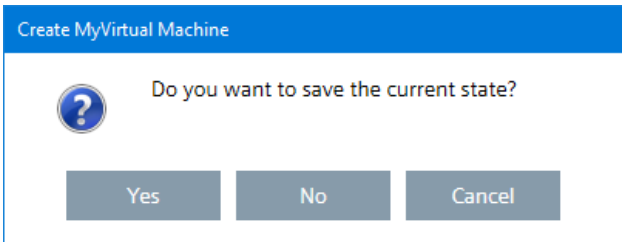
Mechanical properties

Servo selection	Axis type	Load gear numerator
MSP1 / SERVO_3.3:3	Rotary	1
MX1 / SERVO_3.3:4	Ball screw drive	1
MY1 / SERVO_3.3:5	Ball screw drive	1
MZ1 / SERVO_3.3:6	Ball screw drive	1
MA1 / SERVO_3.3:7	Ball screw drive	1
MB1 / SERVO_3.3:8	Ball screw drive	1

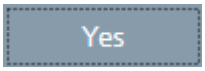




Make sure the *.vcp file is saved with the new “Machanical properties”.







A111: END

Notes:

A113

SINUMERIK ONE

Creating a PLC program

Module Description:

The TIA Portal STEP 7 Professional PLC Programming package is used to create the machine interface program for the SINUMERIK ONE.

In order to use the STEP 7 Professional package to its upmost, it is necessary to know the structure of the SINUMERIK controller and the structure of its interface; this knowledge is required in order to produce a machine tool oriented PLC program which adheres to the relevant safety regulations of the destination country.

Module objective:

Upon completion of this module you will know how to use the STEP 7 Professional Programming tool to:

- Create a PLC program
- Install the SINUMERIK PLC Basic Program
- Configure the PLC Hardware
- Download a PLC project to the virtual and the real controller from a computer

Content:

Sinumerik

S7-1500 PLC Cyclic Operation

SINUMERIK ONE PLC

Integrate Basic PLC

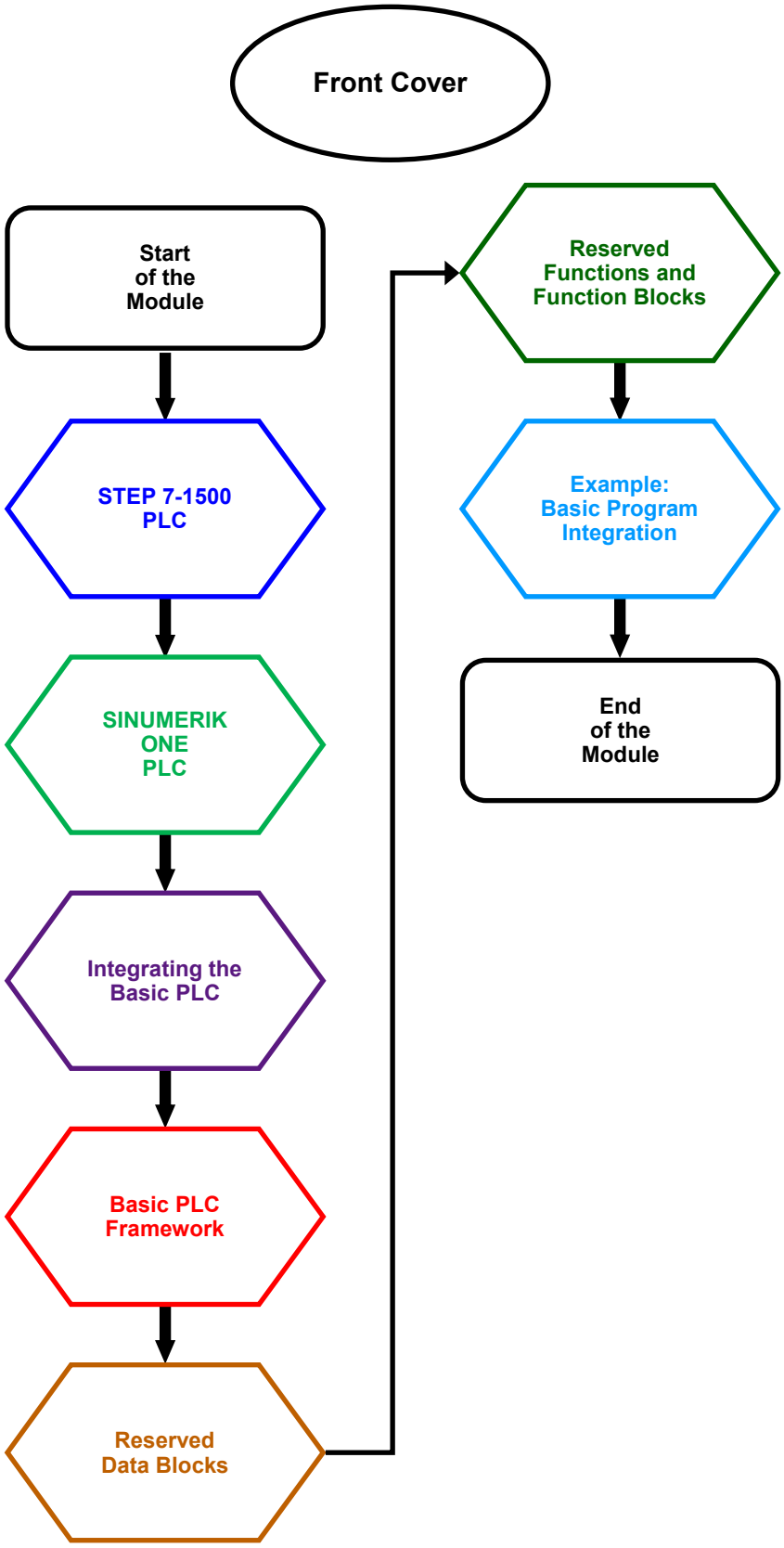
Basic PLC Framework

Reserved Functions

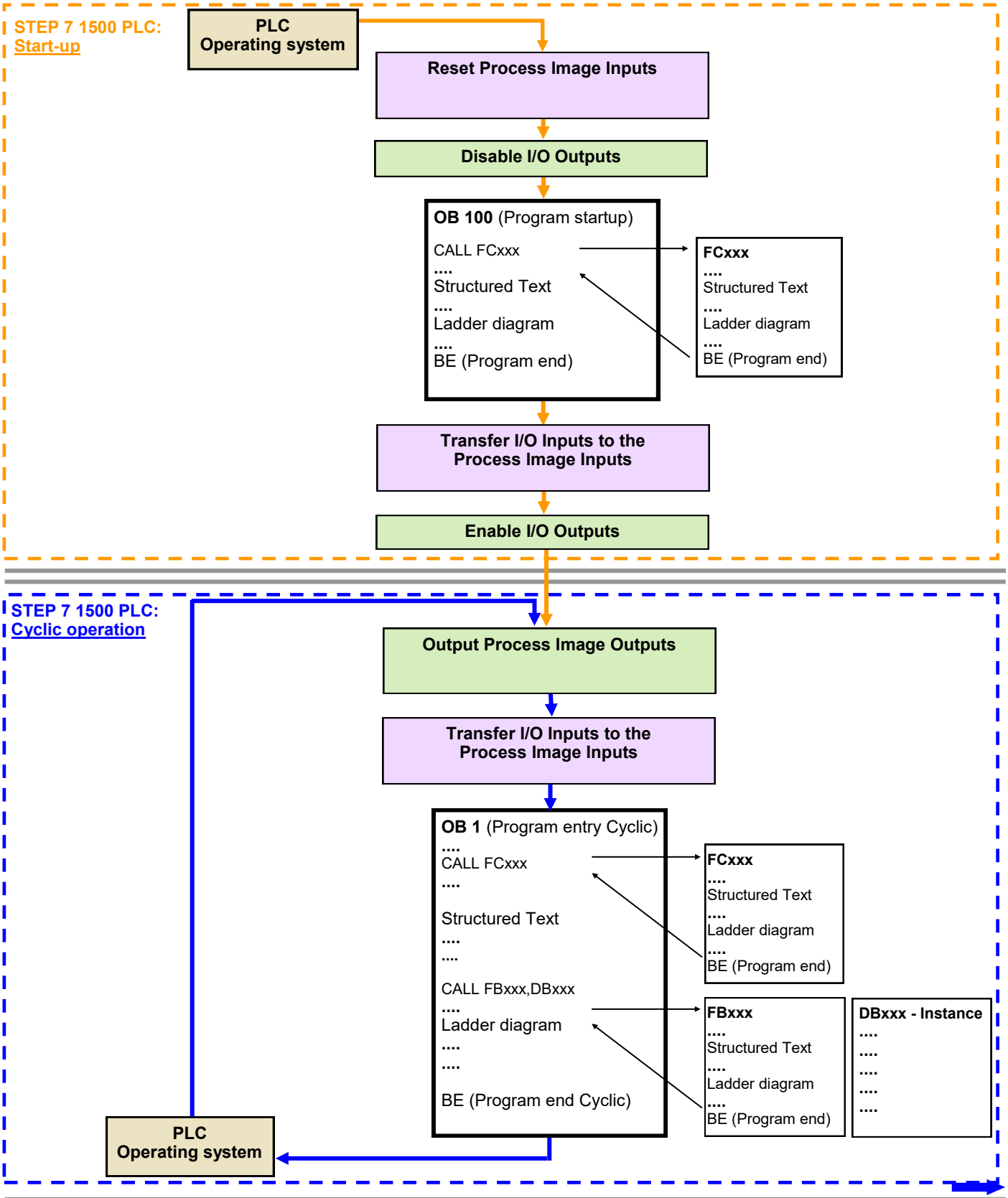
Reserved Data

Example Base Program

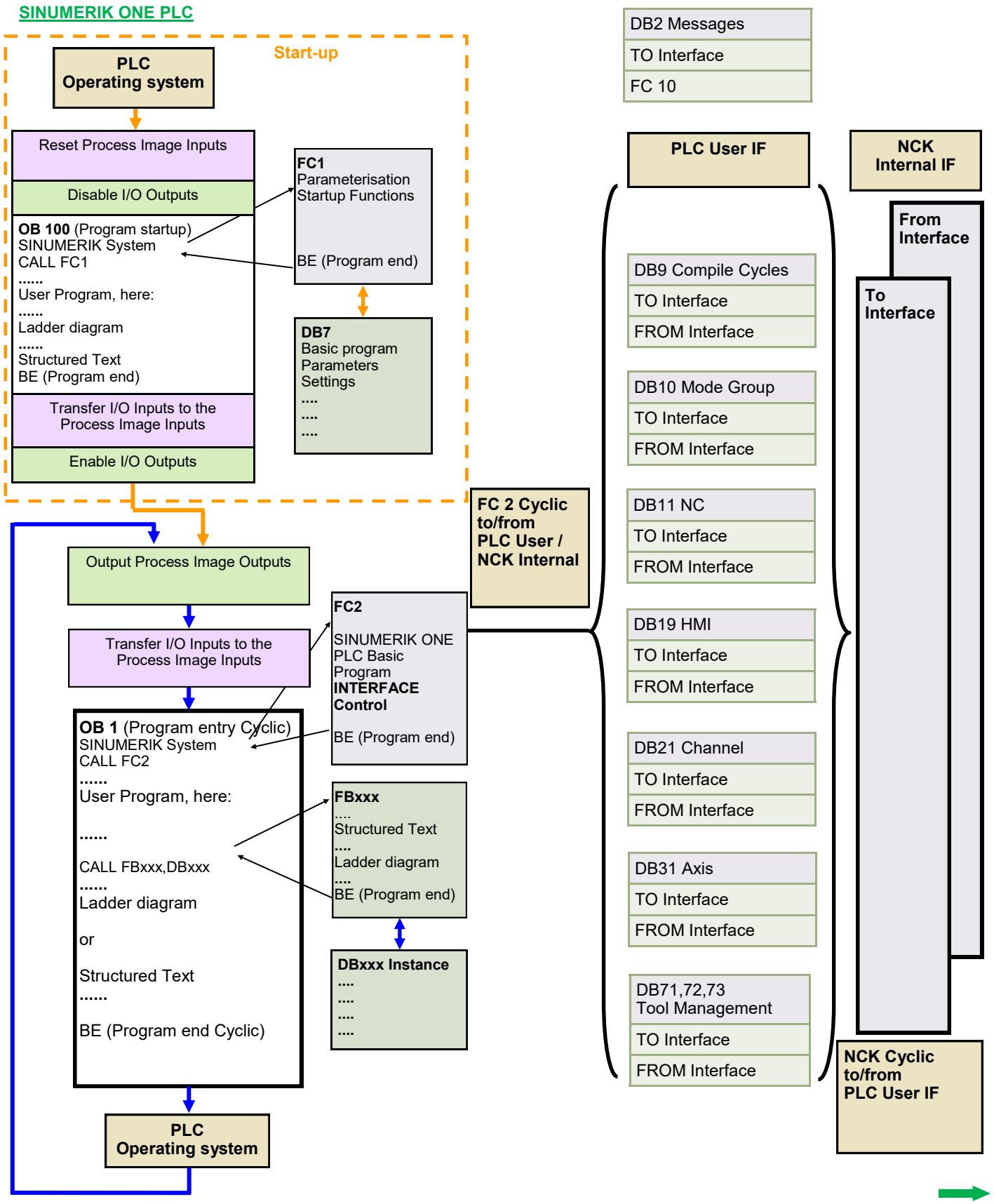
This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
STEP 7-1500 PLC
SINUMERIK ONE PLC
Integrating the Basic PLC
Basic PLC Framework
Reserved Data Blocks
Reserved Functions and Function Blocks
Example: Basic Program Integration
End of the Module

STEP 7-1500 PLC



Notes:


SINUMERIK ONE PLC
Start-up
PLC Operating system
Reset Process Image Inputs
Disable I/O Outputs
OB 100 (Program startup)
SINUMERIK System
CALL FC1
.....
User Program, here:
.....
Ladder diagram
.....
Structured Text
BE (Program end)
Transfer I/O Inputs to the Process Image Inputs
Enable I/O Outputs
FC1
Parameterisation
Startup Functions
BE (Program end)
DB7
Basic program
Parameters
Settings
....
....
....
Output Process Image Outputs
Transfer I/O Inputs to the Process Image Inputs
OB 1 (Program entry Cyclic)
SINUMERIK System
CALL FC2
.....
User Program, here:
.....
CALL FBxxx,DBxxx
.....
Ladder diagram
or
Structured Text
.....
BE (Program end Cyclic)
PLC Operating system
FC2
SINUMERIK ONE PLC Basic Program
INTERFACE Control
BE (Program end)
FBxxx
....
Structured Text
....
Ladder diagram
....
BE (Program end)
DBxxx Instance
....
....
....
....
FC 2 Cyclic to/from PLC User / NCK Internal
DB2 Messages
TO Interface
FC 10
PLC User IF
NCK Internal IF
From Interface
To Interface
DB9 Compile Cycles
TO Interface
FROM Interface
DB10 Mode Group
TO Interface
FROM Interface
DB11 NC
TO Interface
FROM Interface
DB19 HMI
TO Interface
FROM Interface
DB21 Channel
TO Interface
FROM Interface
DB31 Axis
TO Interface
FROM Interface
DB71,72,73
Tool Management
TO Interface
FROM Interface
NCK Cyclic to/from PLC User IF

Notes:

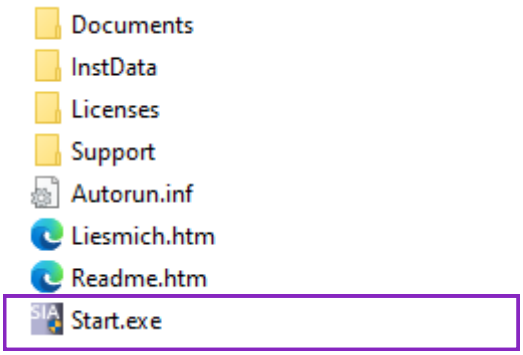
Integrating the Basic PLC

Installation of SINUMERIK ONE PLC Toolbox V20

SINUMERIK ONE TIA Portal Toolbox is an option package for SIMATIC STEP 7 Professional for SINUMERIK ONE as of V6.15 with an additional setup. The current PLC Toolbox V20 can be ordered as DVD from Sales.

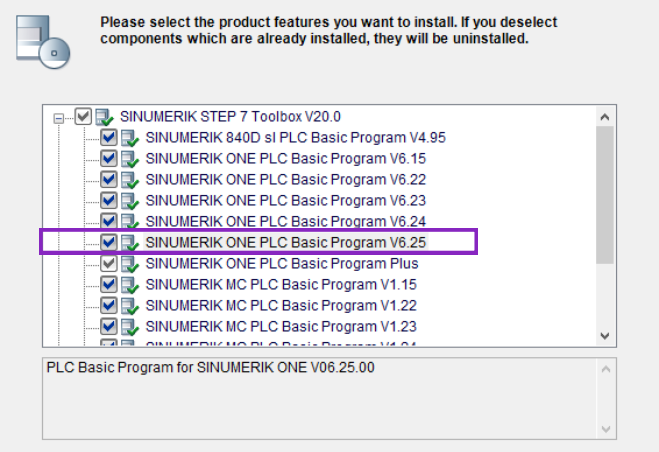
Installation of the Toolbox:

The installation is started with the "Start.exe" file on the Toolbox DVD.



Selecting the product configuration

When all basic program versions are selected, all available versions of the libraries are then available.



Note:
The installation of the PLC Toolbox is described in the module A101 “Service and Commissioning Tools”.

The installed versions can be displayed in the TIA Portal at **Help / Installed software...**.



Content of PLC Toolbox V20

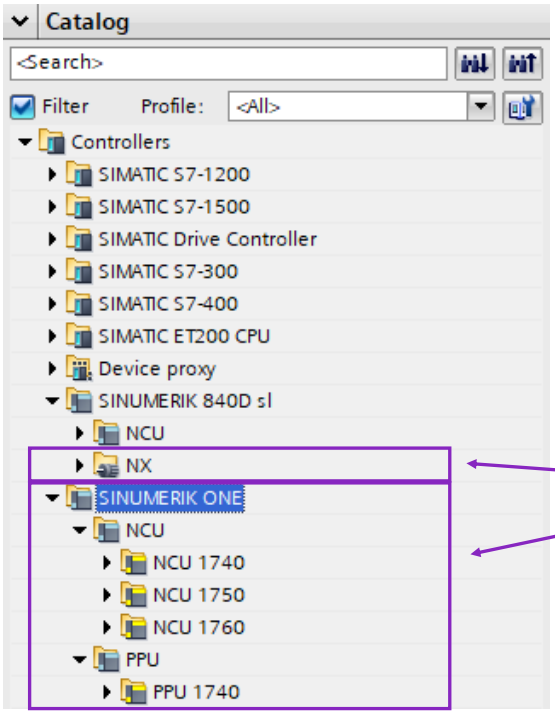
SINUMERIK ONE TIA Portal Toolbox V20 allows the configuring of the SINUMERIK hardware in the TIA Portal with the following functionality and tools:

- Augment the hardware catalog with SINUMERIK NCU and the NX module
- Support the hardware migration
- SINUMERIK PLC basic program
- Export alarms and text lists for SINUMERIK Operate
- Support PROFINET IO IRT for NCK
- Create SINUMERIK PLC commissioning archives
- Support SINUMERIK Safety Integrated F-PLC
- NC VAR Selector (external tool)

Since the SINUMERIK ONE TIA Portal Toolbox V20 is an option package, the installation of SIMATIC STEP 7 Professional V20 is assumed.

Notes:

SINUMERIK ONE Hardware

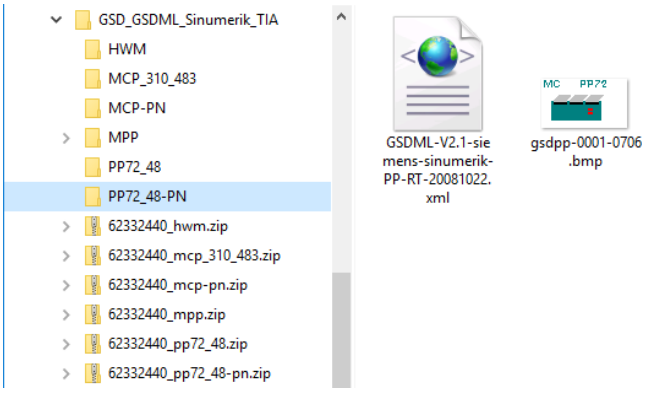


After the Toolbox installation, the **Hardware Catalog** contains the SINUMERIK ONE NCU/NX components that can be added to the project via drag-and-drop.

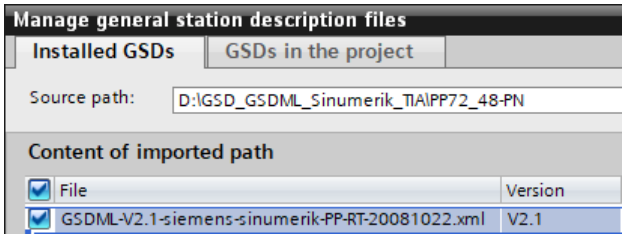
To add further SINUMERIK components, such as the machine control panel or the PP 72/48 I/O module, to the hardware catalog, the corresponding GSD and GSDML files must be installed. Some of these are identical for TIA Portal and S7 Classic. The associated description must be observed.

The current versions of the device description files can be found on the Service&Support pages on the Internet.
SINUMERIK peripherals: GSD/GSDML files:

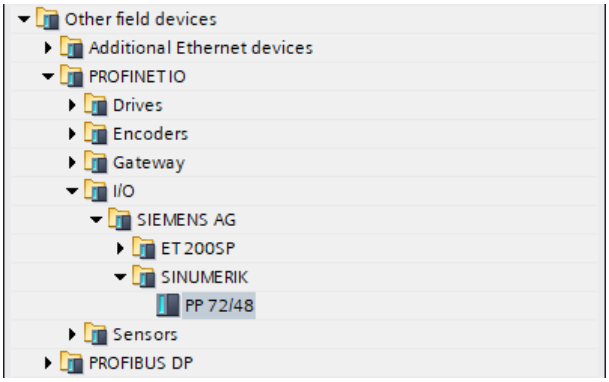
They must be unpacked on the PG/PC for the installation.



The corresponding GSD/GSDML (here PP72/48-PN) is installed via **Options / Manage GSD**.

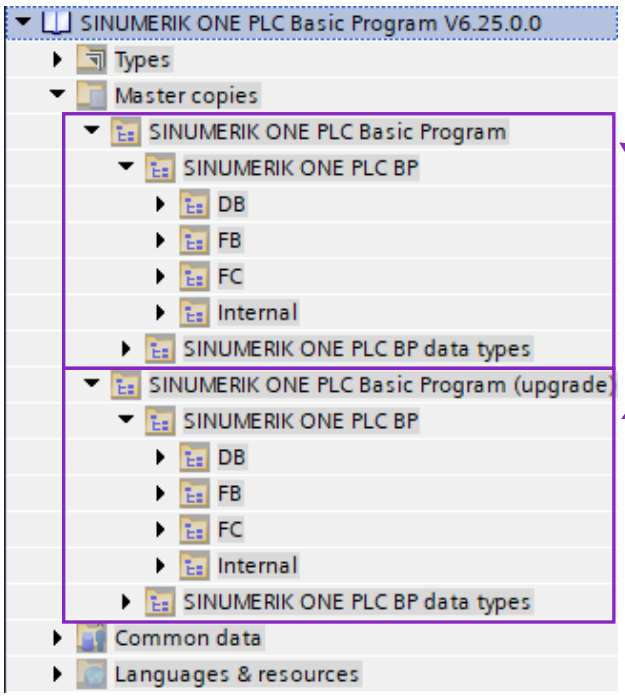


After the successful installation, the hardware catalog is updated.



Notes:

SINUMERIK ONE PLC Basic Program



After the Toolbox installation, the **global libraries** contain the SINUMERIK ONE PLC basic program in the available versions V6.15, V6.22, V6.23, V6.24 and V6.25.

They contain the following copy templates from which the required blocks can be copied to the program blocks via drag-and-drop.

SINUMERIK ONE PLC Basic Program for a complete new project.
Contains all required blocks for the full variant (31 axes, 10 channels), sorted into subfolders, as well as all required data types for DBs, MCPs and NC variables.

SINUMERIK ONE PLC Basic Program (upgrade) for a NC software upgrade.
Contains all know-how protected blocks for the full variant without the blocks to be changed by the user: FC12, DB5, DB75, DB78 and DB1001.

User data blocks from the basic program:

Unlike SINUMERIK Toolbox for STEP 7 V5.x, the user data blocks from the basic program are supplied from the Engineering System for the TIA Portal Toolbox:

- Axes/spindles: Axis1 [DB31]…Axis31 [DB61]
- Channels: Chan1 [DB21]…Chan10 [DB30]
- PLC / mode group signals: Mode group [DB11]
- DB2 alarms of the PLC (alarm numbers 500000 to 799999): ALMSG_DB [DB2]
- Tool management TMLoadIF [DB71]...TMRevIF_MT [DB1073]
- PLC/HMI signals: MMC [DB19]
- M groups decoding [DB75/DB76]

Notes:

Overview SINUMERIK ONE PLC Basic program

Data blocks (DB)

Address	Symbol	Description	PLC PB	PLC PB (Upgrade)
DB2	LBP_AlarmMsgs	PLC alarms/messages	x	x
DB5	LBP_ParamAlarmMsgs	Message type in LBP_AlarmMsgs	x	-
DB7	LBP_ConfigData	Configuration data	x	x
DB10	LBP_NC	Signals from/to NC/PLC and HMI software	x	x
DB11	LBP_ModeGroup	Mode group specific signals	x	x
DB16	LBP_PIServices	PI services	x	x
DB19	LBP_HMI	Signals from/to Operator panel (OP)	x	x
DB21… DB30	LBP_Chan1… LBP_Chan10	Channel specific signals	x	x
DB31… DB61	LBP_Axis1… LBP_Axis31	Axis/spindle signals	x	x
DB71	LBP_TMLoad	Interface for loading/unloading magazine	x	x
DB1071	LBP_TMLoadMT	Interface for loading/unloading magazine (Multitool)	x	x
DB72	LBP_TMSpindle	Interface for spindle as change position	x	x
DB1072	LBP_TMSpindleMT	Interface for spindle as change position (Multitool)	x	x
DB73	LBP_TMTurret	Interface for turret	x	x
DB1073	LBP_TMTurretMT	Interface for turret (Multitool)	x	x
DB75	LBP_MFuncDecListConfig	Decoding list for M Decoding	x	-
DB76	LBP_MFuncDecListSignals	Signal list for M Decoding	x	x
DB78	LBP_PowerrideConfig	Preconfigured behaviour for Powerride	x	-
DB1000	LBP_CtrlEnergy	Energy saving profiles	x	x
DB1001	LBP_SentronPac	SENTRON PAC	x	-
DB1002	LBP_SpindleTempSensor	Spindle temperature sensor	x	x

Function blocks (FB)

Address	Symbol	Description	PLC PB	PLC PB (Upgrade)
FB2	LBP_ReadVar	Read NC variable	x	x
FB3	LBP_WriteVar	Write NC variable	x	x
FB5	LBP_ReadGUD	Read GUD variable	x	x
FB7	LBP_ReqPIService	Request PI service	x	x
FB21	LBP_Powerride	Powerride	x	x
FB25000	LBP_OpUnitComm	Operator unit communication	x	x
FB25002	LBP_OpUnitWriteRecord	Write record to MCP	x	x

Notes:

Functions (FC)

Address	Symbol	Description	PLC PB	PLC PB (Upgrade)
FC1	LBP_ConfigPB	Basic program, start section	x	x
FC2	LBP_MainPB	Basic program, cyclic section	x	x
FC3	LBP_NCKProcessIRT	Basic program, alarm triggered section	x	x
FC6	LBP_TMTransferMT	Transfer block for Tool management and Multitool	x	x
FC7	LBP_TMTransferTurret	Transfer block for Tool management with turret	x	x
FC8	LBP_TMTransfer	Transfer block for Tool management	x	x
FC9	LBP_ReqASUP	Start of asynchronous subprogram	x	x
FC10	LBP_GenerateAlarmMsgs	Alarm and operating messages	x	x
FC12	LBP_CallBackAuxFunc	Call interface for user with auxiliary functions	x	-
FC13	LBP_HTCtrlHT2	Display control for HT2	x	x
FC17	LBP_CtrlStarDelta	Star-Delta switchover	x	x
FC18	LBP_CtrlAxisSpinIde	Spindle control	x	x
FC19	LBP_MCPCtrlMilling	Transfer of MCP signals to interface (M variant)	x	x
FC21	LBP_TransferSelData	Data exchange NC/PLC	x	x
FC24	LBP_MCPCtrlMillingSmall	Transfer of MCP signals to interface (small variant)	x	x
FC25	LBP_MCPCtrlTurning	Transfer of MCP signals to interface (T variant)	x	x
FC26	LBP_HTCtrlHT8	Transfer of HT8 signals to interface	x	x
FC25002	LBP_Handwheels	Transfer Handwheel signals	x	x

Internal blocks

Address	Symbol	Description	PLC PB	PLC PB (Upgrade)
DB3	LBP_InternalDB3	Only for internal use in the basic program	x	x
DB8	LBP_InternalDB8	Only for internal use in the basic program	x	x
DB1020	LBP_InternalDB1020	Only for internal use in the basic program	x	x
DB25000	LBP_Internal1	Only for internal use in the basic program	x	x
FB1020	LBP_InternalFB1020	Only for internal use in the basic program	x	x
FB25001	LBP_InternalProc2	Only for internal use in the basic program	x	x
FC25000	LBP_InternalProc1	Only for internal use in the basic program	x	x
FC25001	LBP_InternalProc3	Only for internal use in the basic program	x	x
FC25003	LBP_InternalProc4	Only for internal use in the basic program	x	x
FC25004	LBP_InternalProc6	Only for internal use in the basic program	x	x

Notes:

SINUMERIK ONE PLC Data types

Data blocks (DBs)

Symbol	DB No.
LBP_typeAlarmMsgs	DB2
LBP_typeConfigData	DB7
LBP_typeNC	DB10
LBP_typeModeGroup	DB11
LBP_typeHMI	DB19
LBP_typeChanX	DB21
LBP_typeAxisX	DB31
LBP_typeTMLoad/LoadMT	DB71/DB1071
LBP_typeTMSpindle/SpindleMT	DB72/DB1072
LBP_typeTMTurret/TurretMT	DB73/DB1073

Function parameter (FCs)

Symbol	FC No.
LBP_typeNCKProcessIRTFlags	FC3
LBP_typeTMMagazines	FC1

NC variables

Symbol	Data type
LBP_typeNCVariableToken	
LBP_typeNCVariableTokenBool	BOOL
LBP_typeNCVariableTokenDateAndTime	DateAndTime
LBP_typeNCVariableTokenDInt	DINT
LBP_typeNCVariableTokenInt	INT
LBP_typeNCVariableTokenLReal	LREAL
LBP_typeNCVariableTokenReal	REAL
LBP_typeNCVariableTokenSInt	SINT
LBP_typeNCVariableTokenString	STRING
LBP_typeNCVariableTokenUDInt	UDINT
LBP_typeNCVariableTokenUInt	UINT
LBP_typeNCVariableTokenUSInt	USINT

Machine control panels (MCPs)

Symbol	MCP type
LBP_typeMCP310StandardInput LBP_typeMCP310StandardOutput LBP_typeMCP310AdditionalInput LBP_typeMCP310AdditionalOutput	MCP310 (Profibus)
LBP_typeMCP483StandardInput LBP_typeMCP483StandardOutput LBP_typeMCP483AdditionalInput LBP_typeMCP483AdditionalOutput	MCP483 (Profibus)
LBP_typeMCP310CPnInput LBP_typeMCP310CPnOutput	MCP310CPN
LBP_typeMCP310PnInput LBP_typeMCP310PnOutput	MCP310PN
LBP_typeMCP483CPnInput LBP_typeMCP483CPnOutput	MCP483CPN
LBP_typeMCP483PnInput LBP_typeMCP483PnOutput	MCP483PN
LBP_typeMCP310StandardInput LBP_typeMCP310StandardOutput	Slimline
LBP_typeMCP483MillingInput LBP_typeMCP483MillingOutput	M variant

Symbol	MCP type
LBP_typeMCP483TurningInput LBP_typeMCP483TurningOutput	T variant
LBP_typeHandwheel	Handwheel

Modular MCP

Symbol	MCP type
LBP_typeModularMcpAxisInput LBP_typeModularMcpAxisOutput LBP_typeModularMcpBaseInput LBP_typeModularMcpBaseOutput LBP_typeModularMcpKeypadRecord LBP_typeModularMcpLedAttribute LBP_typeModularMcpModeGroupInput LBP_typeModularMcpModeGroupOutput LBP_typeModularMcpUniversalInput LBP_typeModularMcpUniversalOutput	Modular MCP
LBP_typePowerrideConfig LBP_typePowerrideConfigPos LBP_typePowerrideInput LBP_typePowerrideOutput	Powerride

Notes:

Notes:

Basic PLC Framework

The PLC program has a modular structure. It consists of two parts:

- PLC Basic Program
- PLC User Program

PLC Basic Program: The PLC Basic Program organizes the exchange of signals and data between the PLC User Program, and the NCK, HMI and machine control panel components.

The PLC Basic Program is part of the SINUMERIK ONE TIA Portal Toolbox V20.

Use the corresponding PLC Basic Program for the firmware version of an NCU:

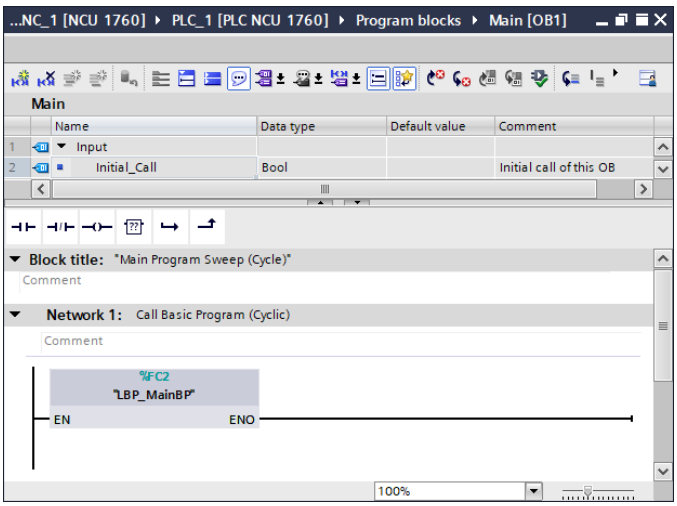
- SINUMERIK ONE PLC Basic Program V6.15
- SINUMERIK ONE PLC Basic Program V6.22
- SINUMERIK ONE PLC Basic Program V6.23
- SINUMERIK ONE PLC Basic Program V6.24
- SINUMERIK ONE PLC Basic Program V6.25

PLC User Program: The PLC User Program is the user-specific part of the PLC program with which the PLC Basic Program is augmented or extended.

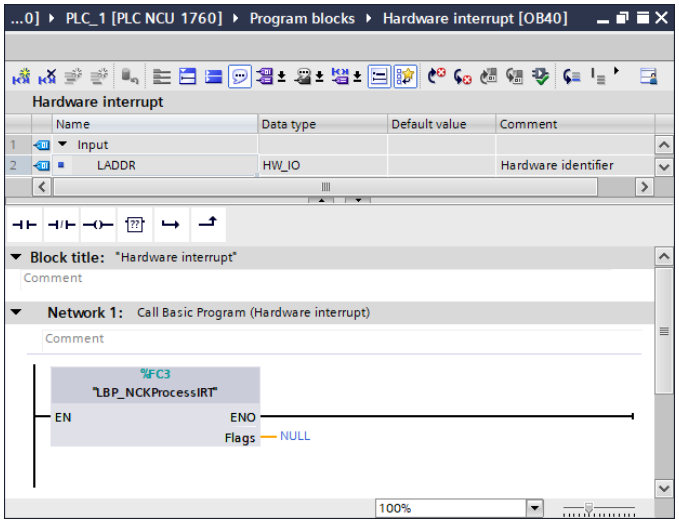
The PLC Basic Program appears when OB1, OB40 and/or OB100 is opened in the STEP 7 editor. The Basic Program is executed by calling FC2 in OB1, by calling FC3 in OB40 and by calling FC1 in OB100.

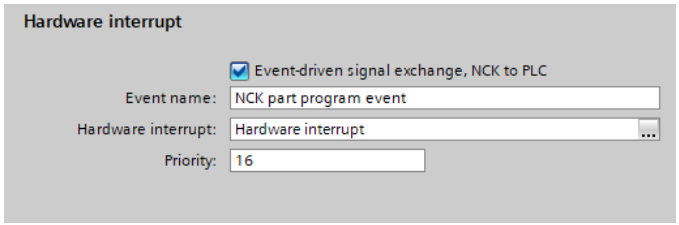
The assignment of the OB40 to the NCK Event is shown by opening the hardware configuration.

OB1:

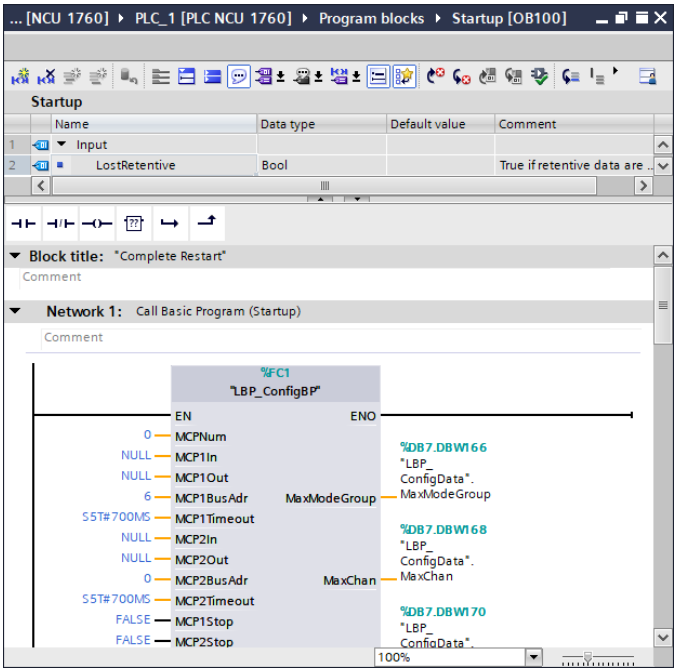


OB40:





OB100:



Notes:

Reserved Data Blocks

Overview of data blocks	
DB no.	**Function**
1	Reserved for Siemens
2 ...3	Basic program
4 ...7	Reserved for Siemens
8	Basic program
9	Reserved for Siemens
10 ... 11	Basic program
12	Interface computer link and transport system
13 ... 15	Reserved for Siemens
16	Basic program
17 ... 18	Reserved for Basic program
19	Basic program
20	Reserved for Siemens
21 ...30	Basic program (Interface for NC channels)
31 ...61	Basic program (Interface for Axes/spindles)

Overview of data blocks	
DB no.	**Function**
62 ...70	Free for user assignment
71 ...73	Basic program (Tool management)
74	Reserved for Siemens
75 ...76	Basic program (M group decoding)
77 ... 80	Reserved for Siemens
81 ... 499	Occupied or reserved for Siemens (ShopMill, ManualTurn)
500 ... 599	Occupied or reserved for Siemens (Transline)
600 ... 999	Occupied or reserved for Siemens (ShopMill, ManualTurn)
1000 ... 1019	Reserved for Siemens
1020	Occupied or reserved for Siemens (Basic program)
1021 ... 1099	Reserved for Siemens
1100 ... 24999	Free for user assignment
25000 ... 29999	Reserved for Siemens
30000... 59999	Default number range for generated F system blocks
60000... 60999	Occupied or reserved for CPU-DB

Notes:

Reserved Functions and Function Blocks

Overview of function blocks	
FB no.	**Function**
0	Reserved for Siemens
1	Reserved for Basic program
2 ...3	Basic program
4	Reserved for Siemens
5	Basic program
6	Reserved for Siemens
7	Basic program
8	Reserved for Siemens
9	Basic program
10 ... 14	Reserved for Siemens
15	Reserved for Basic program
16 ...28	Reserved for Siemens
29	Reserved for Basic program

Overview of function blocks	
FB no.	**Function**
30 ...999	Free for user assignment
1000 ... 1019	Reserved for Siemens
1020	Basic program
1021 ... 1023	Reserved for Siemens
1024 ... 24999	Free for user assignment
25000 ... 32766	Reserved for Basic program
32767... 65535	Default number range for generated F system blocks

Notes:

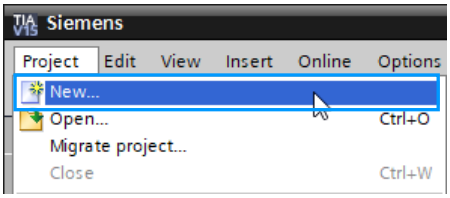
Overview of functions	
FC no.	**Function**
0	Reserved for Siemens
1 ... 3	Basic program
4	Reserved for Siemens
5	Reserved for Basic program
6 ... 29	Reserved for Siemens
30 ...999	Free for user assignment
1000 ... 1023	Reserved for Siemens
1024 ... 24999	Free for user assignment
25000	Basic program
25001 ... 32766	Reserved for Basic program
32767... 65535	Default number range for generated F system blocks

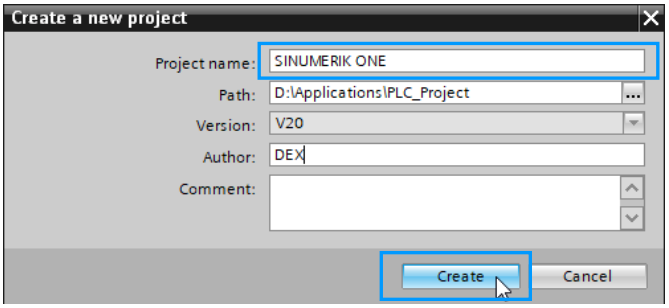
Notes:

Example: Basic Program Integration

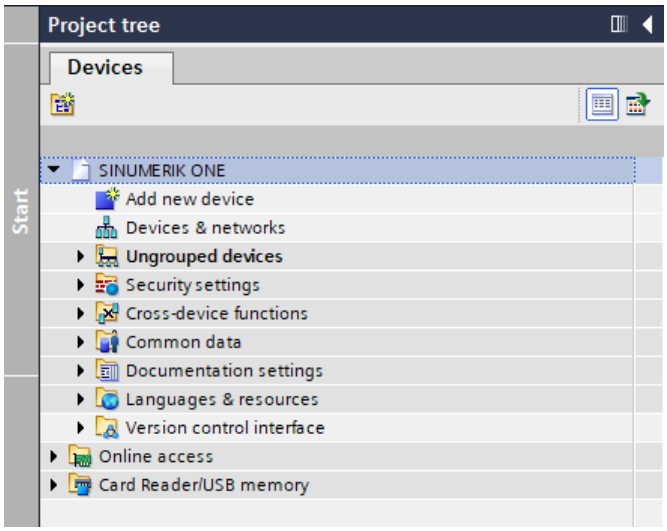
Create a TIA project

Create new project, e.g. SINUMERIK ONE.





The new project is created.



Steps to achieve a running project:

Prerequisites
- SINUMERIK ONE NCU1760 with CNC-SW 6.25
- Create MyVirtualMachine V1.5 with Virtual CNC 6.25
- SIMATIC STEP 7 Professional V20
- SINUMERIK ONE Toolbox V20

Create hardware configuration
- SINUMERIK ONE NCU1760
- ET200SP peripheral
- Profinet Topology
- NX10

Export TIA system messages

Create PLC user program
- Integrate SINUMERIK ONE basic program
- Start-up parametrization
- PLC machine data
- Handwheels
- Machine control panel
- OEM signals, enables, etc.

Load project to Virtual and Real

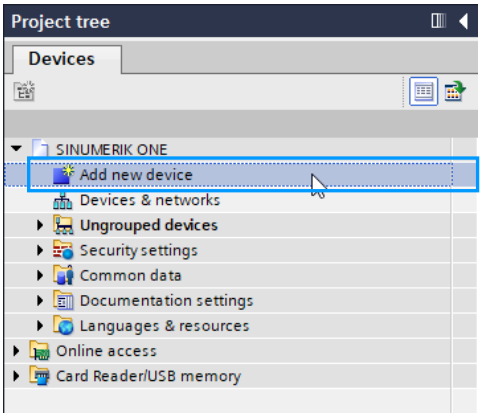
Notes:

Notes:

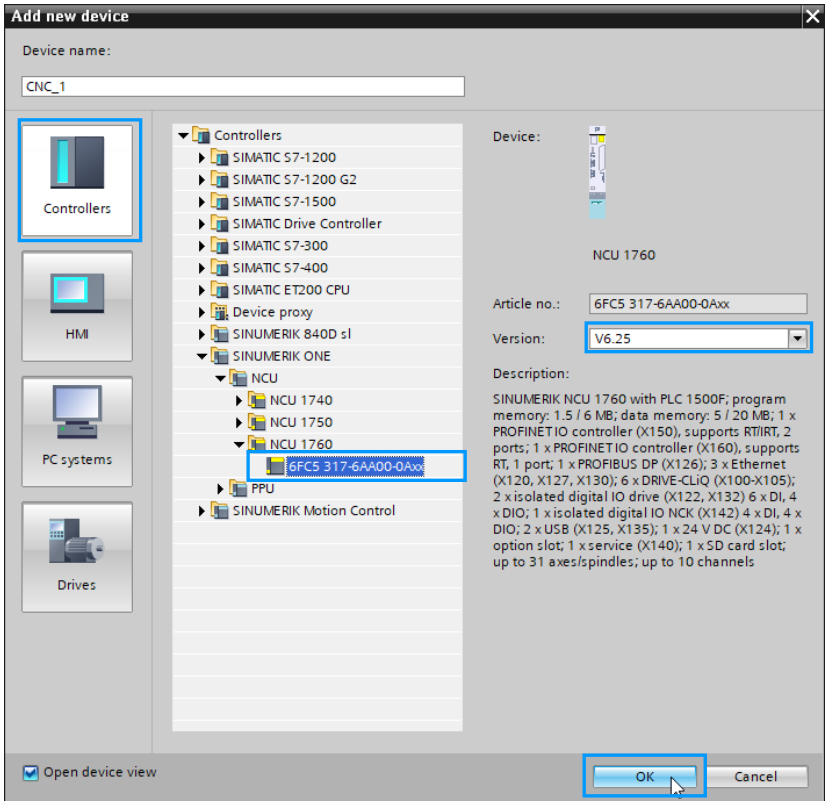
Create Hardware configuration

Adding a SINUMERIK controller

Start the hardware configuration.

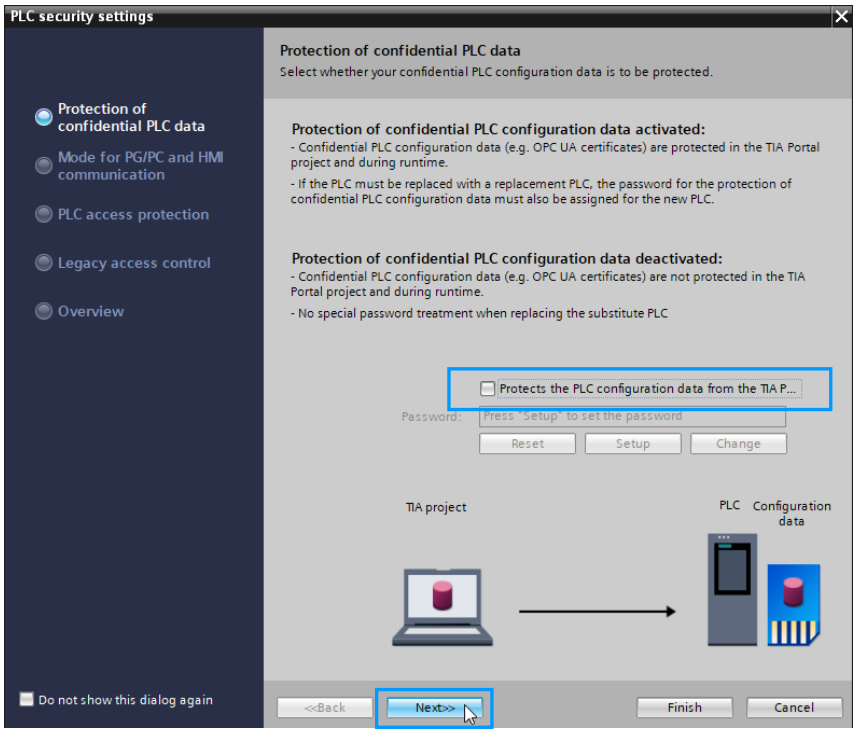


Select the SINUMERIK ONE controller with the suitable version V6.25.



The PLC Security wizard is automatically started.

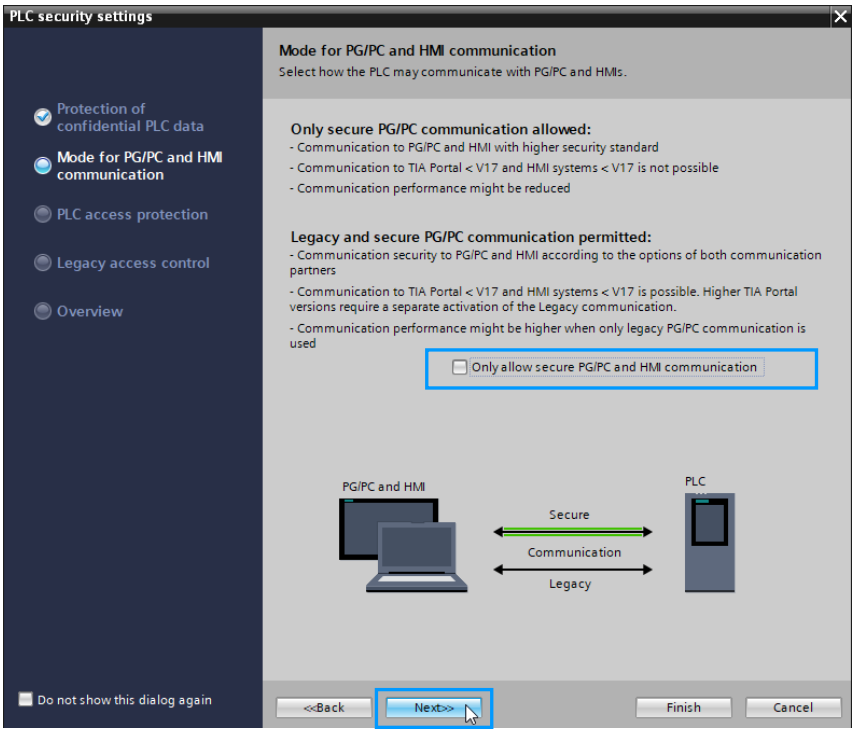
Disable protection of confidential PLC data.



Notes:

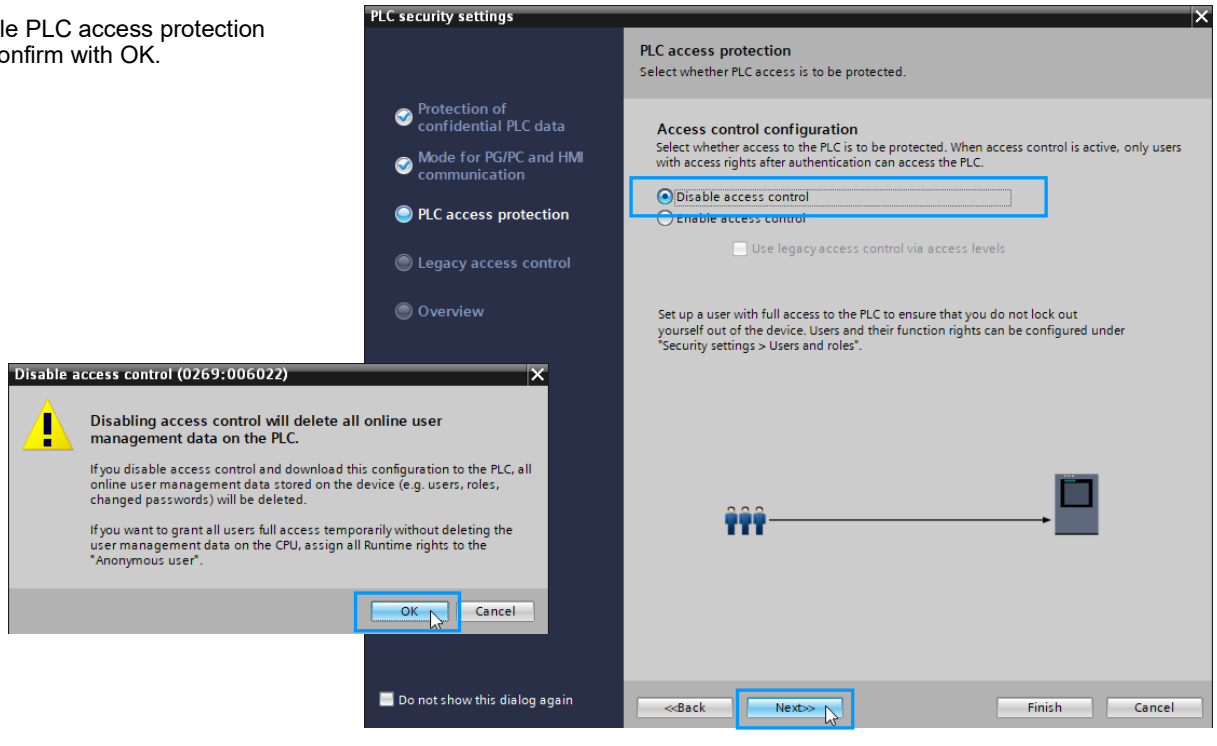
PLC Security wizard - continued.

Disable Secure PG/PC and HMI communication.



PLC Security wizard - continued.

Disable PLC access protection and confirm with OK.

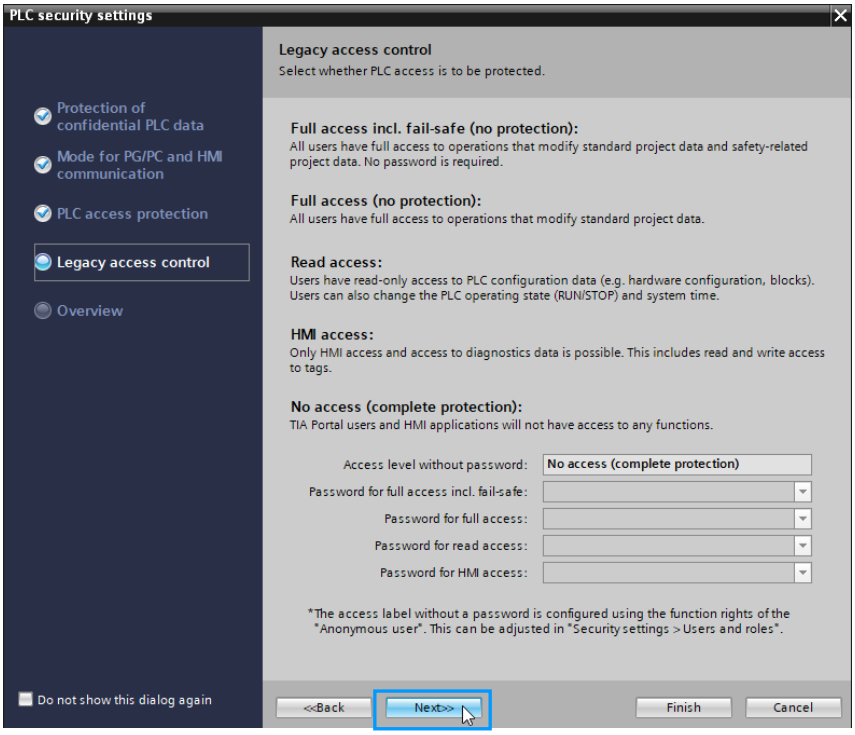


Notes:

PLC Security wizard - continued.

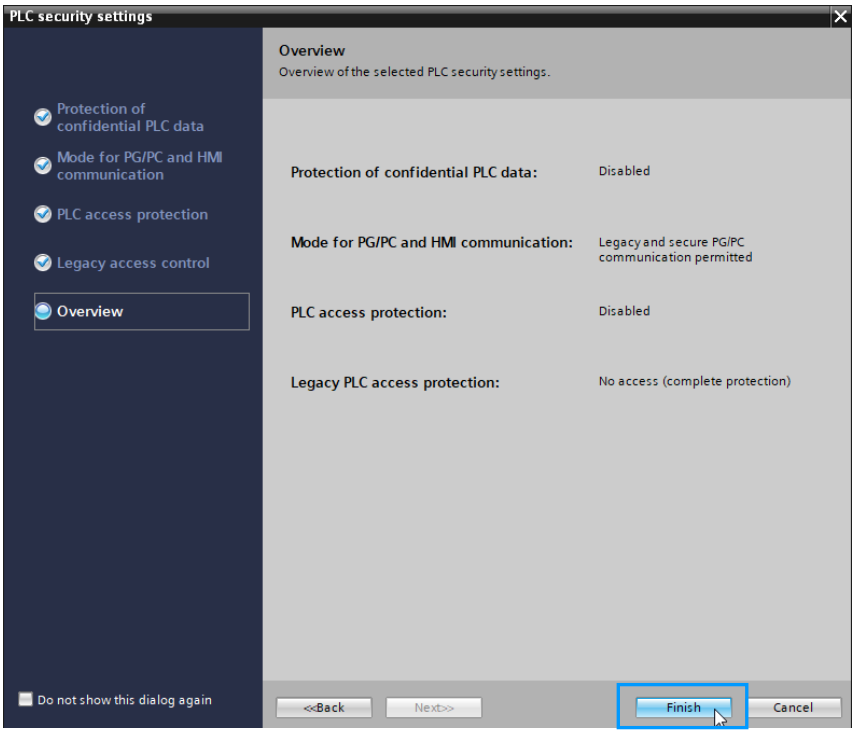
The access level selection cannot be used in this setting From SW 6.24 and later you have to set the access level via the user management of the PLC.

This is done later in Security settings.



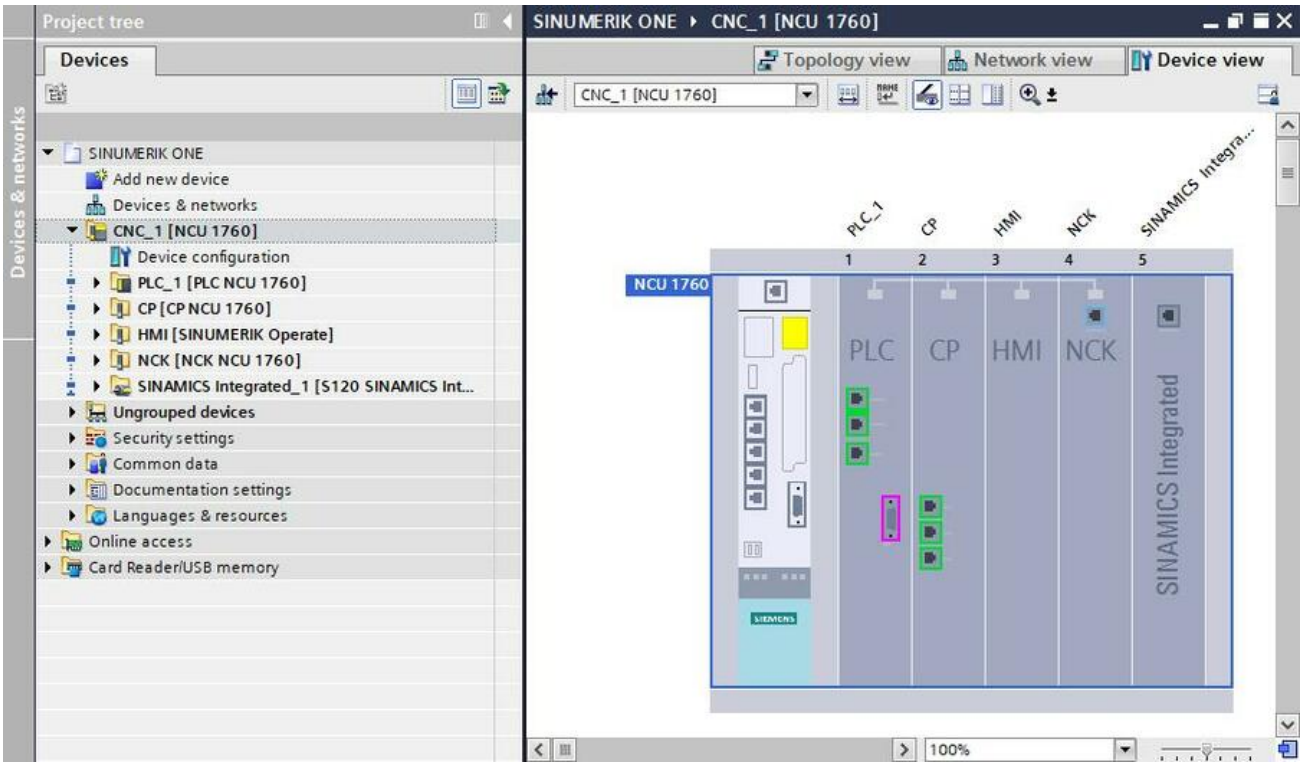
PLC Security wizard - continued.

Finish the Security configuration.



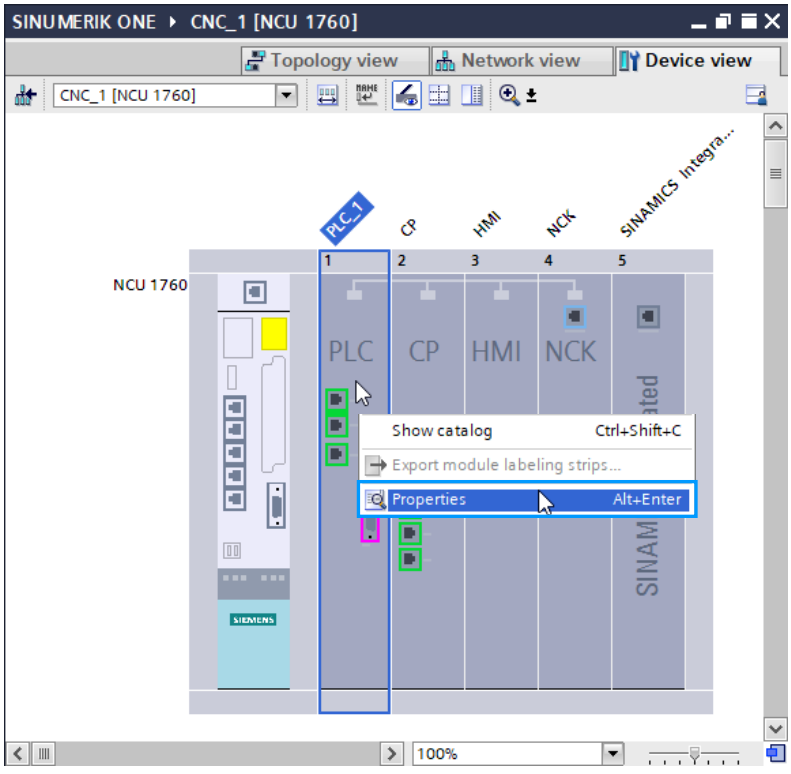
Notes:

The SINUMERIK controller is added.



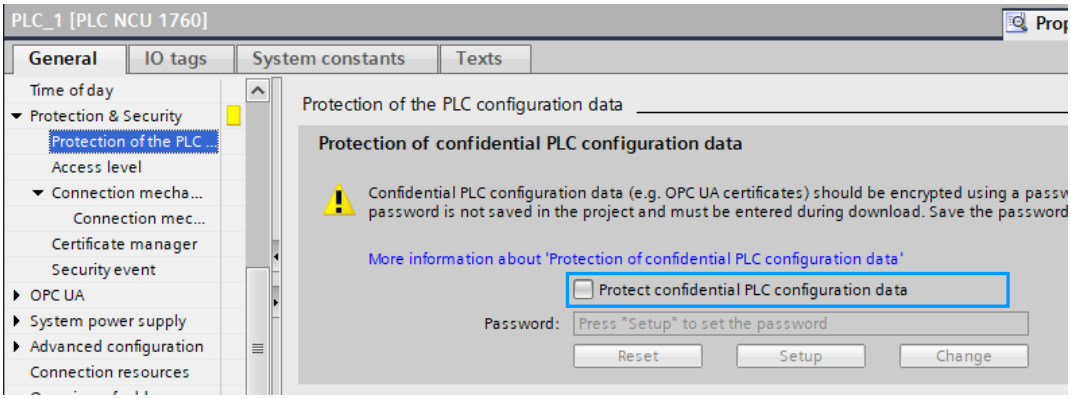
Configure PLC security settings (without the Wizard)

The PLC Security settings can be configured in the hardware configuration, select the properties by right-clicking the object.

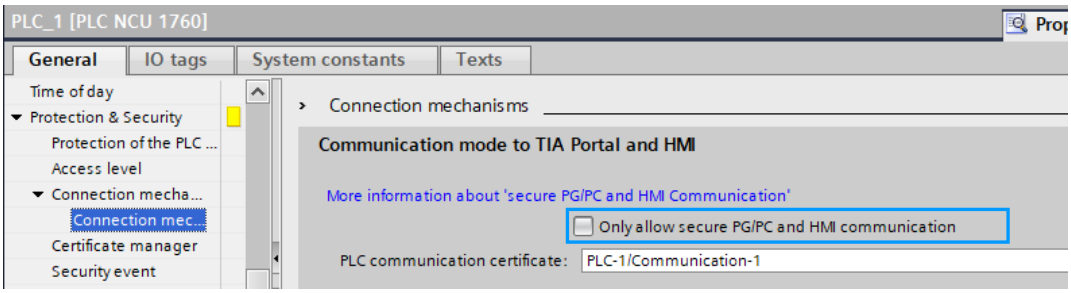


Notes:

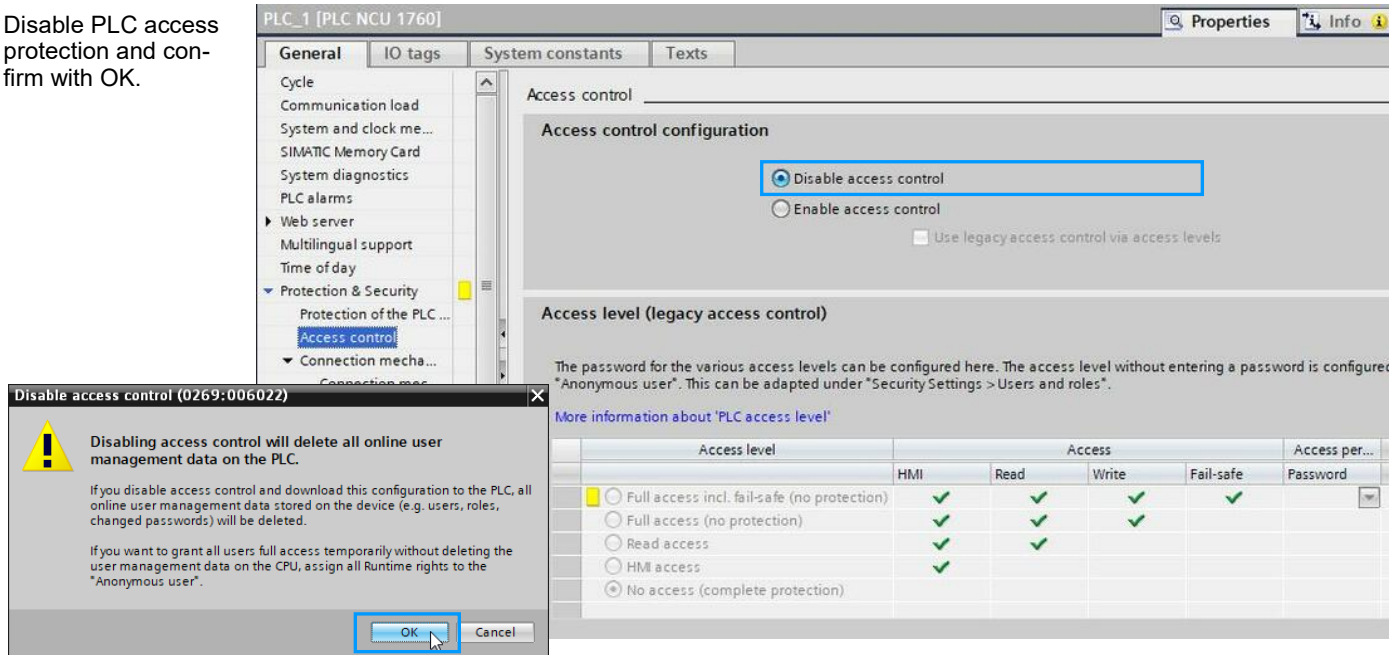
Disable protection of confidential PLC data.



Disable Secure PG/ PC and HMI communication.



Disable PLC access protection and confirm with OK.



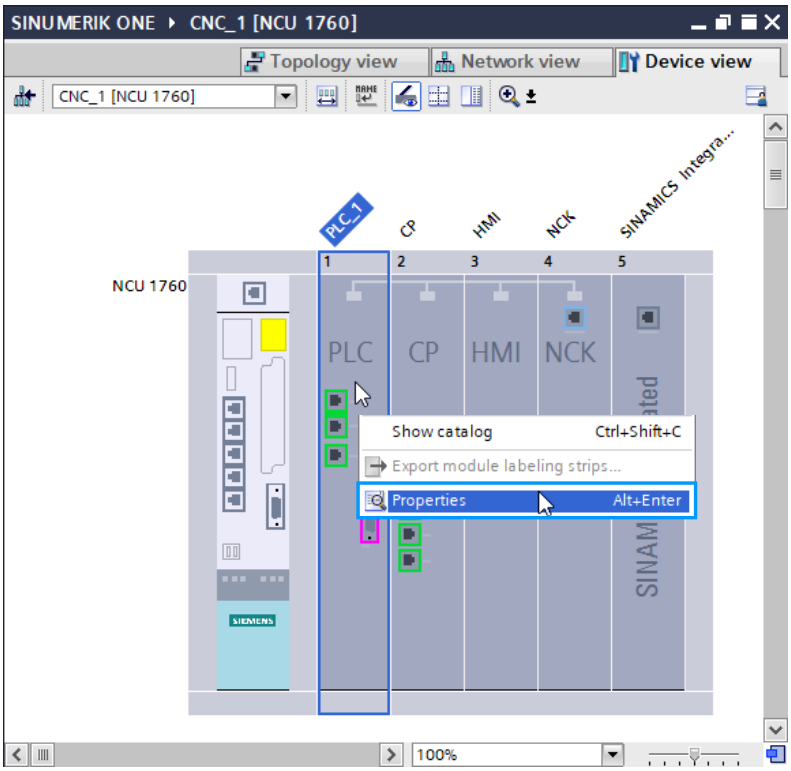
Note:

The access level selection cannot be used in this setting From SW 6.24 and later you have to set the access level via the user management of the PLC. This is done later in Security settings.

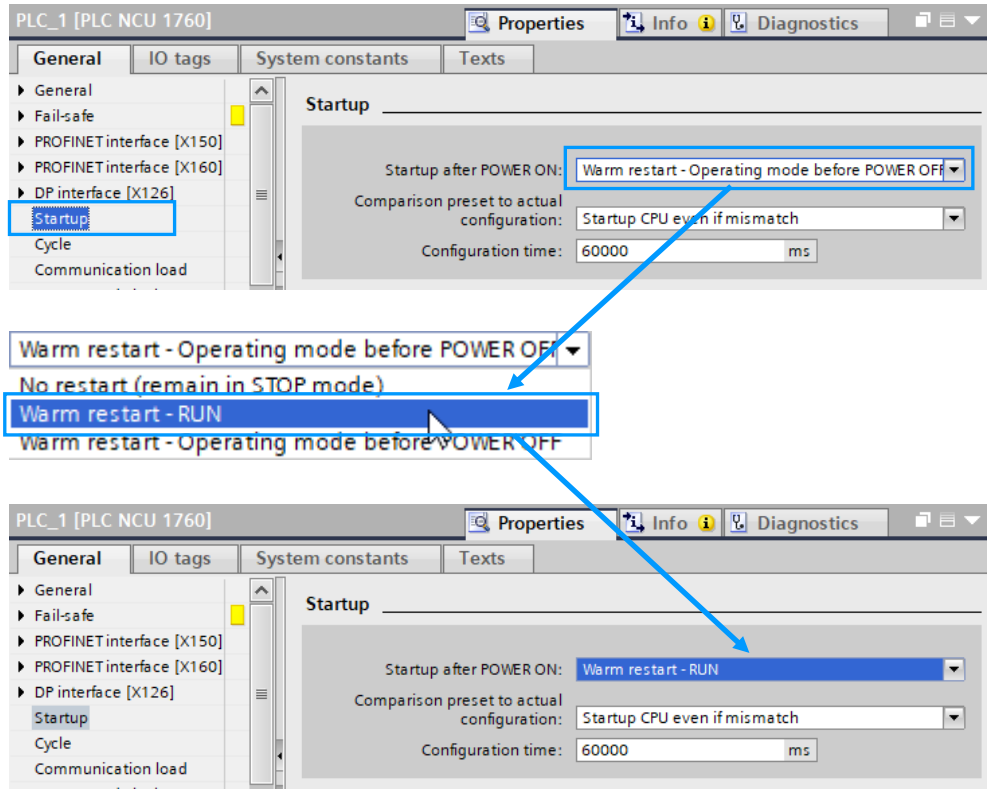
Notes:

Configure start-up after POWER ON

The behavior after POWER ON can be set in the hardware configuration, select the properties by right-clicking the object.



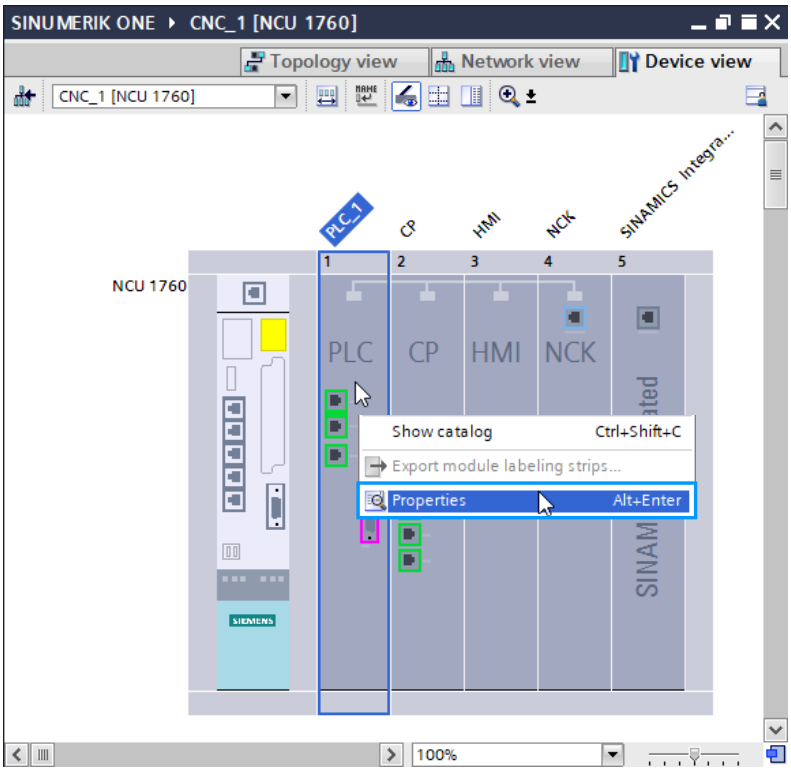
Select start-up behavior **Warm restart - RUN**.



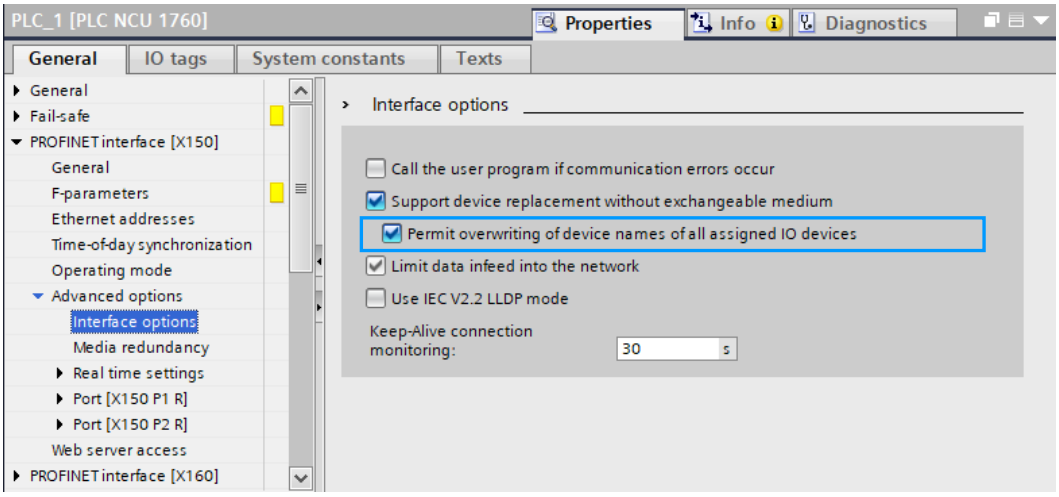
Notes:

Permit overwriting of PROFINET device names

Overwriting of PROFINET device names can be permitted in the hardware configuration, select the properties by right-clicking the object.



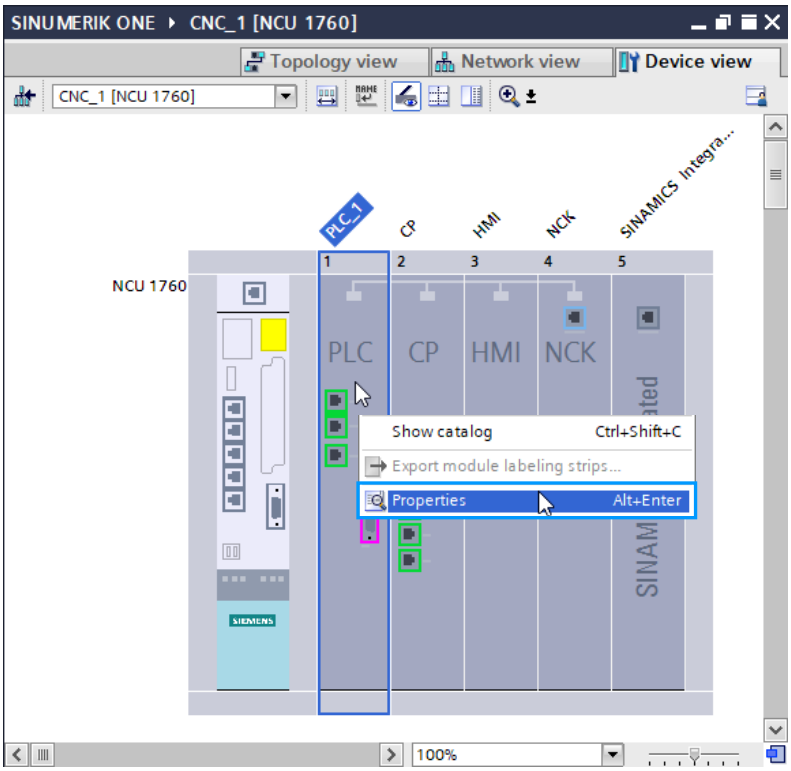
Permit overwriting of PROFINET device names.



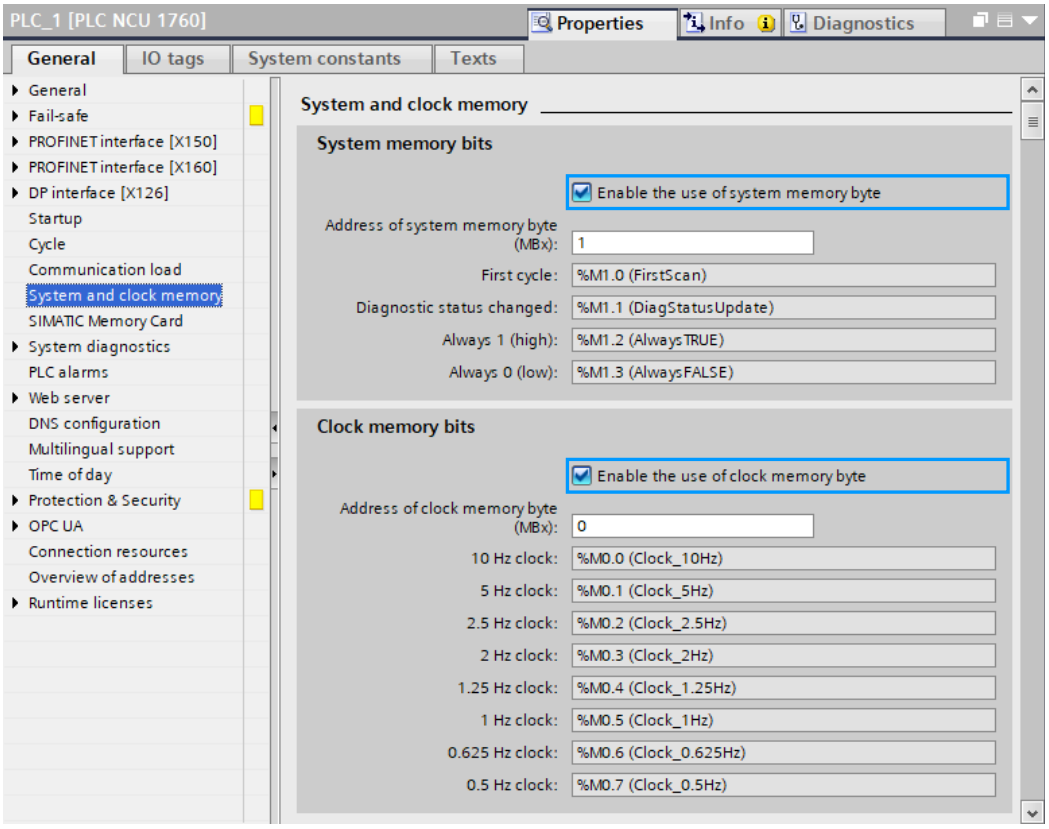
Notes:

Enable system and clock memory bits

The use of the system and clock memory byte can be enabled in the hardware configuration, select the properties by right-clicking the object.

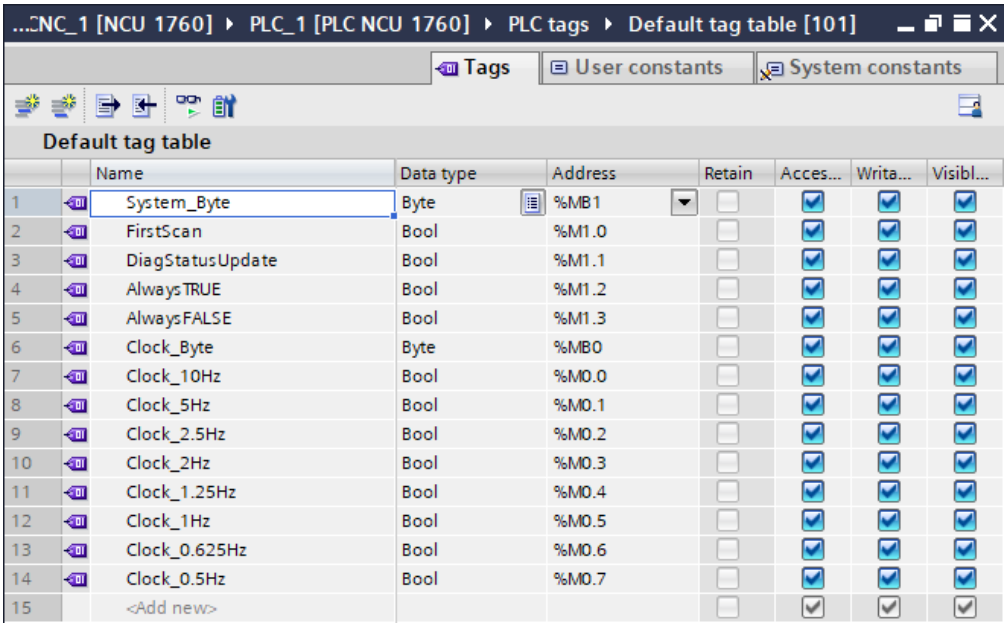


Enable system and clock memory byte.



Notes:

The symbols for the system and clock memory bits are defined in the Default tag table.


...NC_1 [NCU 1760] ▸ PLC_1 [PLC NCU 1760] ▸ PLC tags ▸ Default tag table [101]

Tags | User constants | System constants

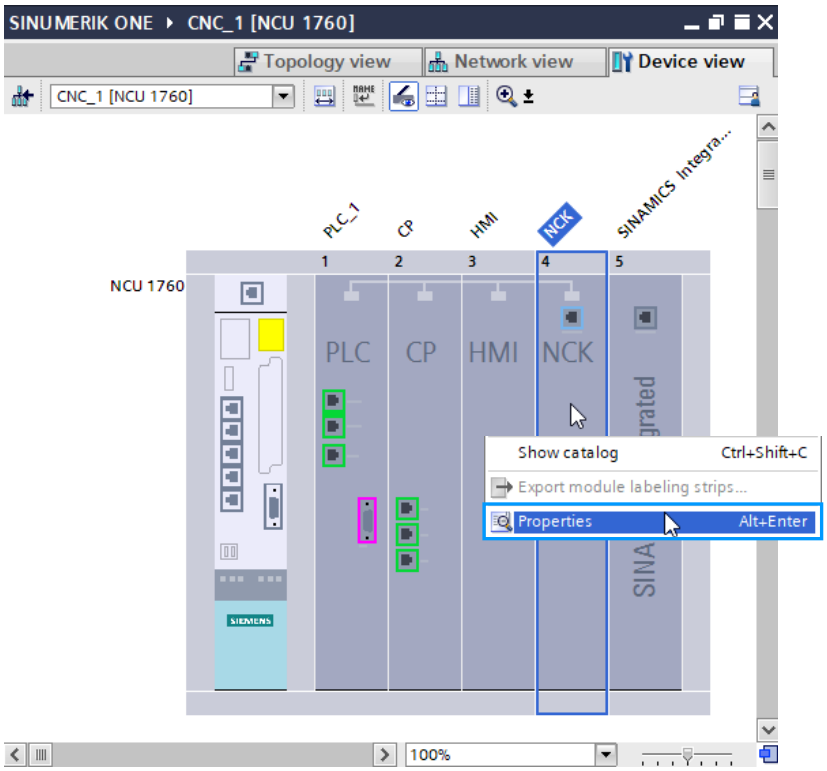
Default tag table

		Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
1		System_Byte	Byte	%MB1		☑	☑	☑
2		FirstScan	Bool	%M1.0		☑	☑	☑
3		DiagStatusUpdate	Bool	%M1.1		☑	☑	☑
4		AlwaysTRUE	Bool	%M1.2		☑	☑	☑
5		AlwaysFALSE	Bool	%M1.3		☑	☑	☑
6		Clock_Byte	Byte	%MB0		☑	☑	☑
7		Clock_10Hz	Bool	%M0.0		☑	☑	☑
8		Clock_5Hz	Bool	%M0.1		☑	☑	☑
9		Clock_2.5Hz	Bool	%M0.2		☑	☑	☑
10		Clock_2Hz	Bool	%M0.3		☑	☑	☑
11		Clock_1.25Hz	Bool	%M0.4		☑	☑	☑
12		Clock_1Hz	Bool	%M0.5		☑	☑	☑
13		Clock_0.625Hz	Bool	%M0.6		☑	☑	☑
14		Clock_0.5Hz	Bool	%M0.7		☑	☑	☑
15		<Add new>				☑	☑	☑

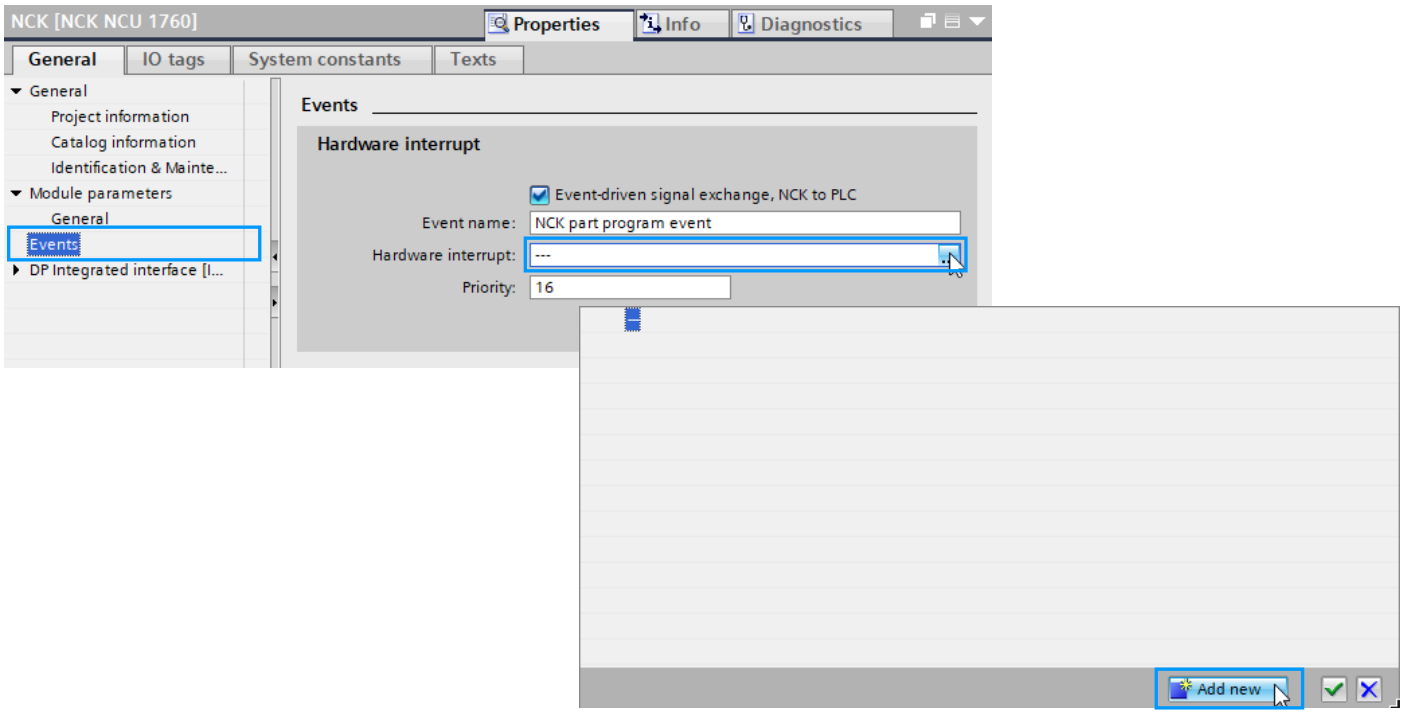
Notes:

Assign hardware interrupt

The process alarms has to be enabled in the NCK, select the properties by right-clicking the object.

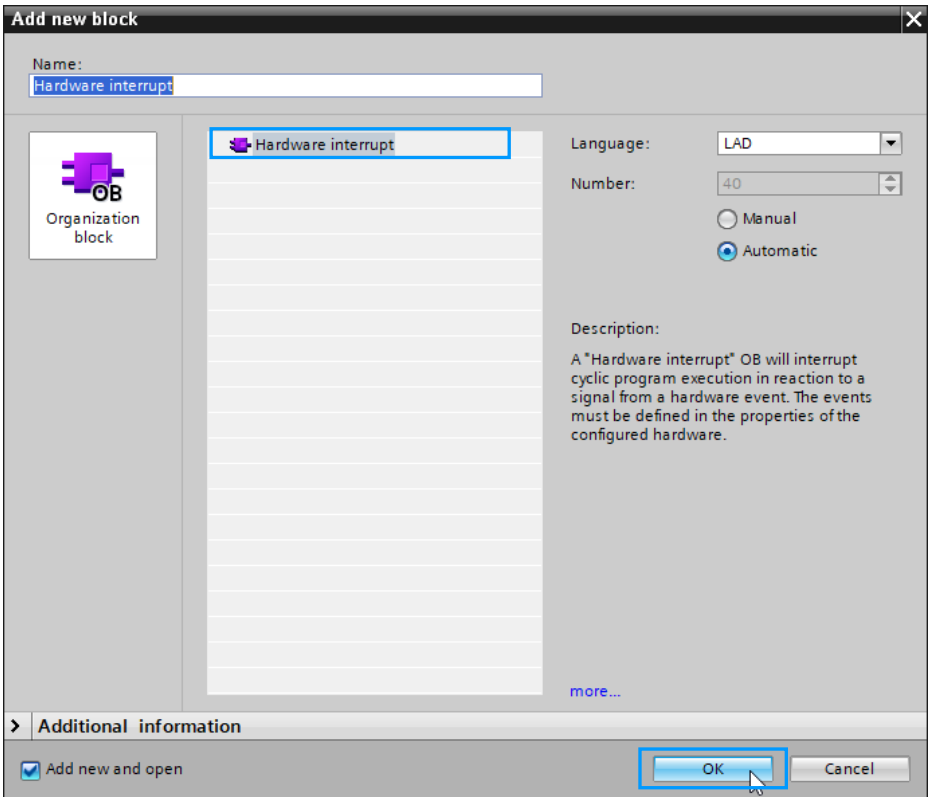


Parametrize a block for the process alarms.

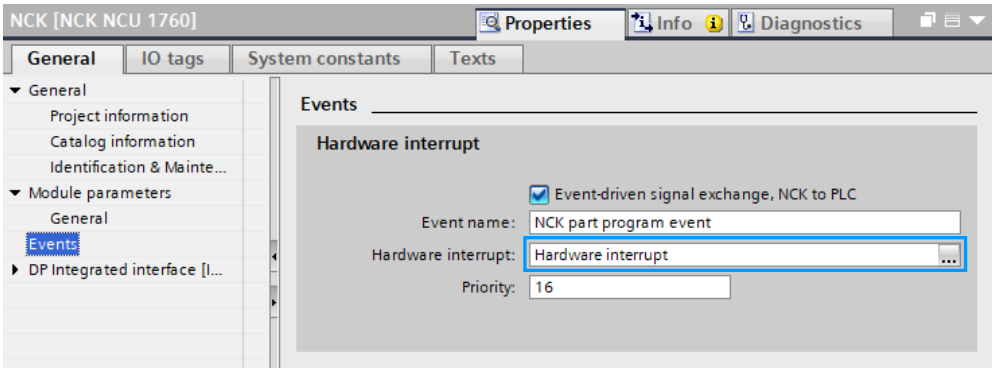


Notes:

The block for the process alarms is newly added.



The block is automatically assigned to the event.



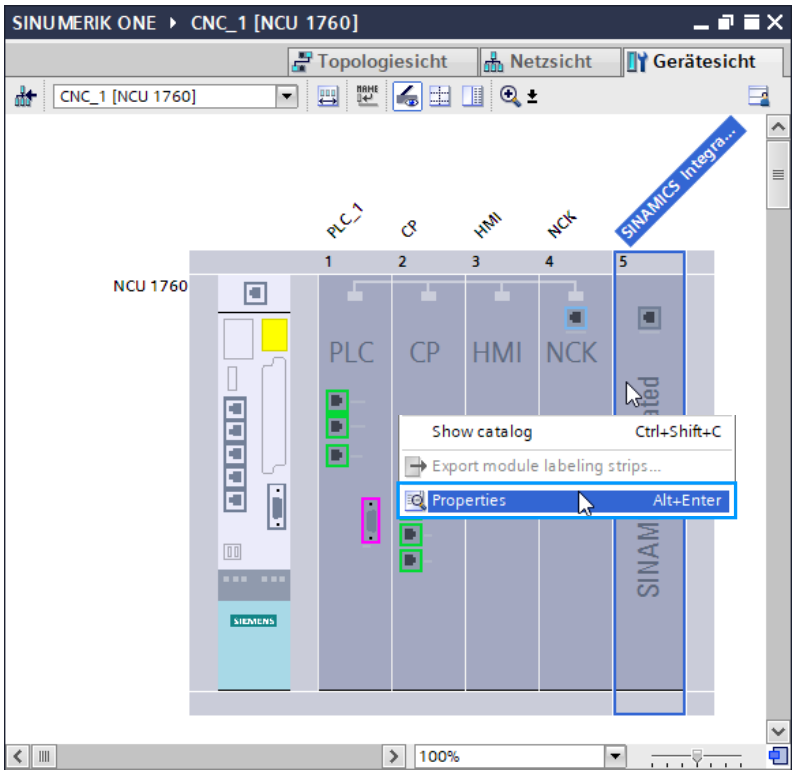
Note:

This link between event and block may be lost if the OB40 is subsequently overwritten from e.g. reference project/library/ external source.

Notes:

Configure SINAMICS telegrams

The logic addresses of the drives has to be set in the hardware configuration, select the properties by right-clicking the object.

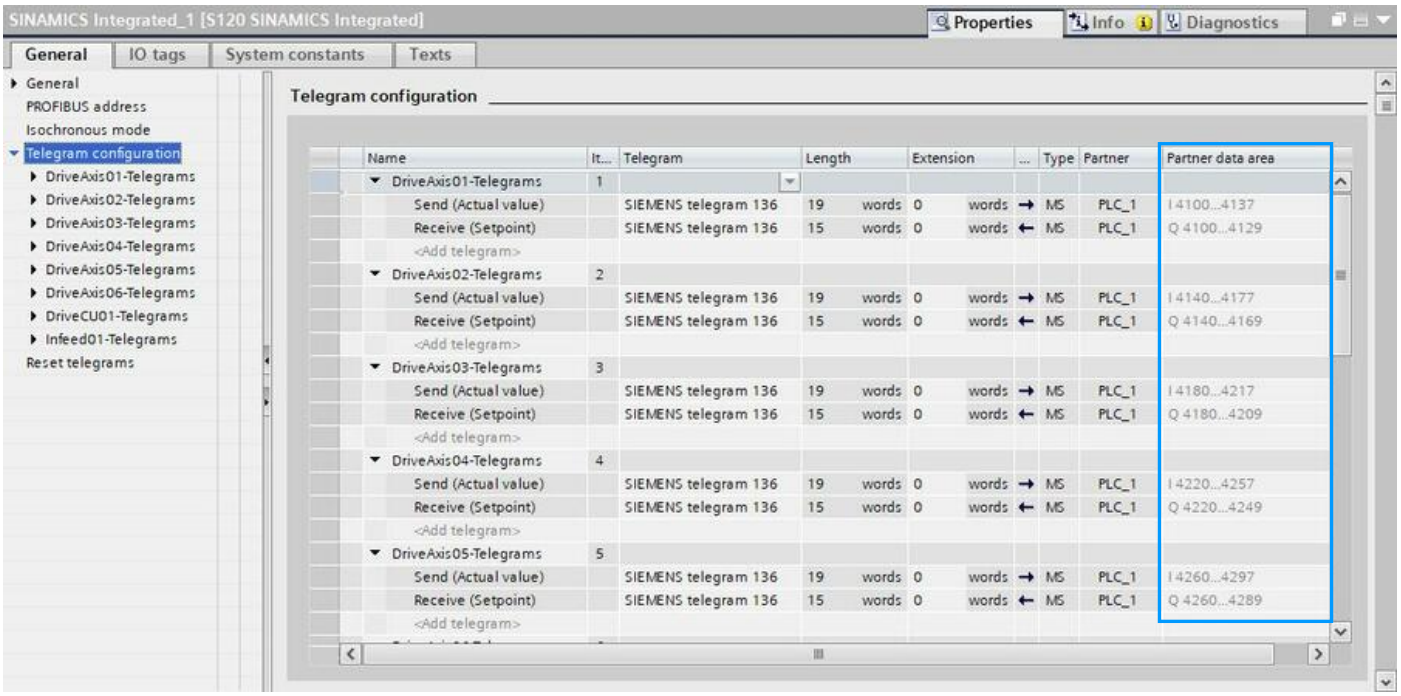


	NC		PLC I/O Adr.	Drive Integr. CU
	MD30110	MD13050		
MX1	[AX1]=1	[0]=4100	4100	DO1
MY1	[AX2]=2	[1]=4140	4140	DO2
MZ1	[AX3]=3	[2]=4180	4180	DO3
MSP1	[AX4]=4	[3]=4220	4220	DO4
MA1	[AX5]=5	[4]=4260	4260	DO5

↓

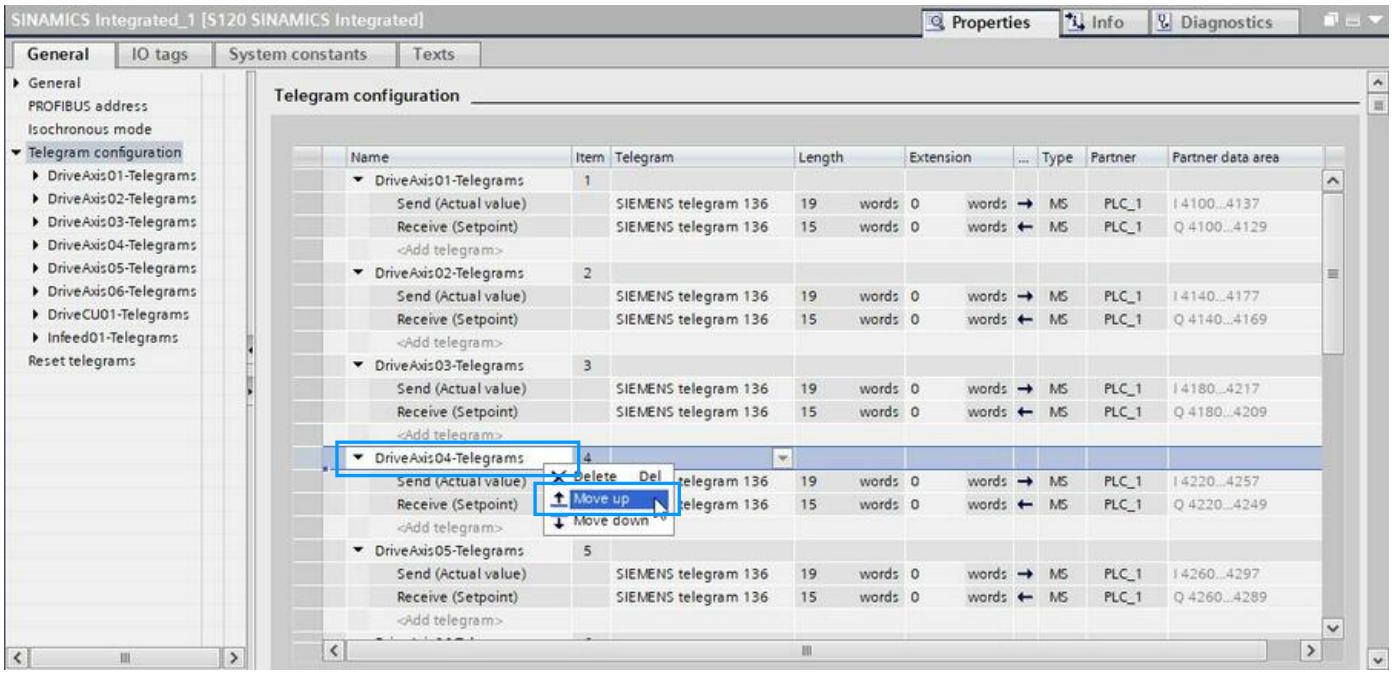
MX1	[AX1]=1	[0]=4100	4220	DO1
MY1	[AX2]=2	[1]=4140	4100	DO2
MZ1	[AX3]=3	[2]=4180	4140	DO3
MSP1	[AX4]=4	[3]=4220	4180	DO4
MA1	[AX5]=5	[4]=4260	4260	DO5

Default setting of the start addresses.

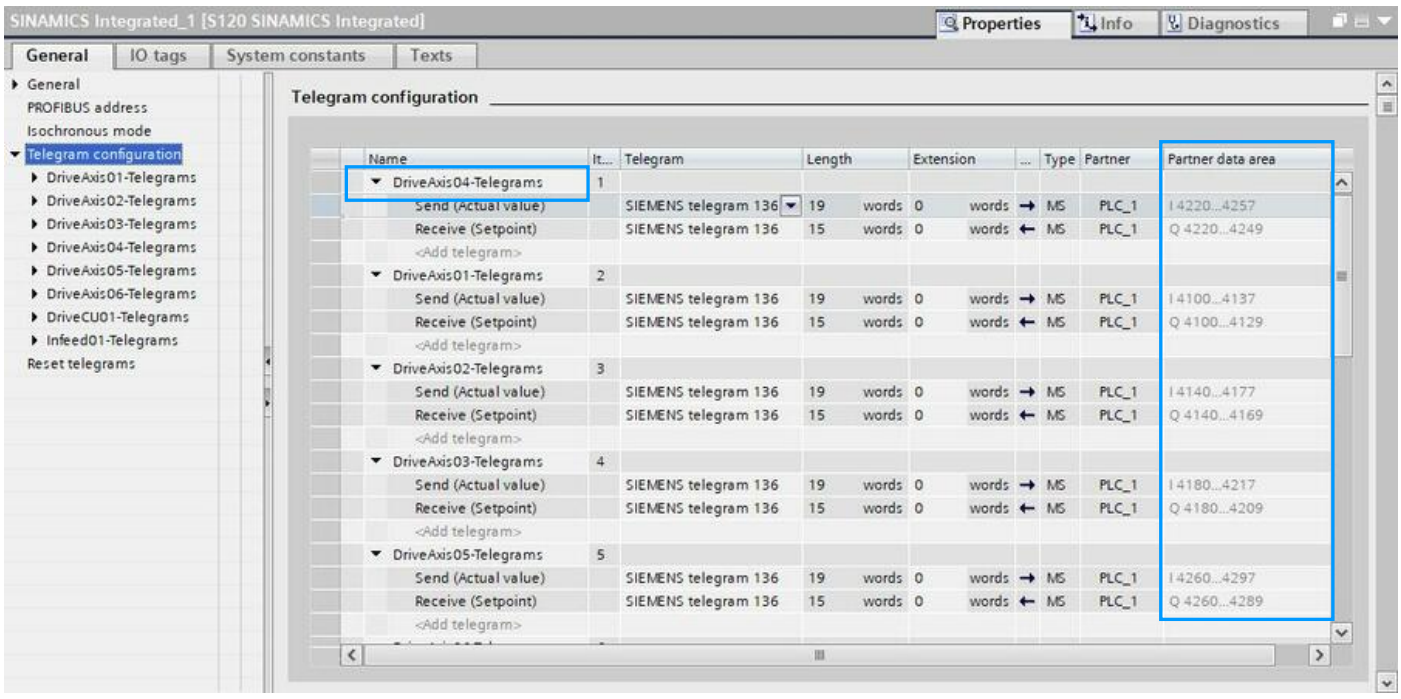


Notes:

Axis containers can be moved in the Telegram configuration using the mouse.



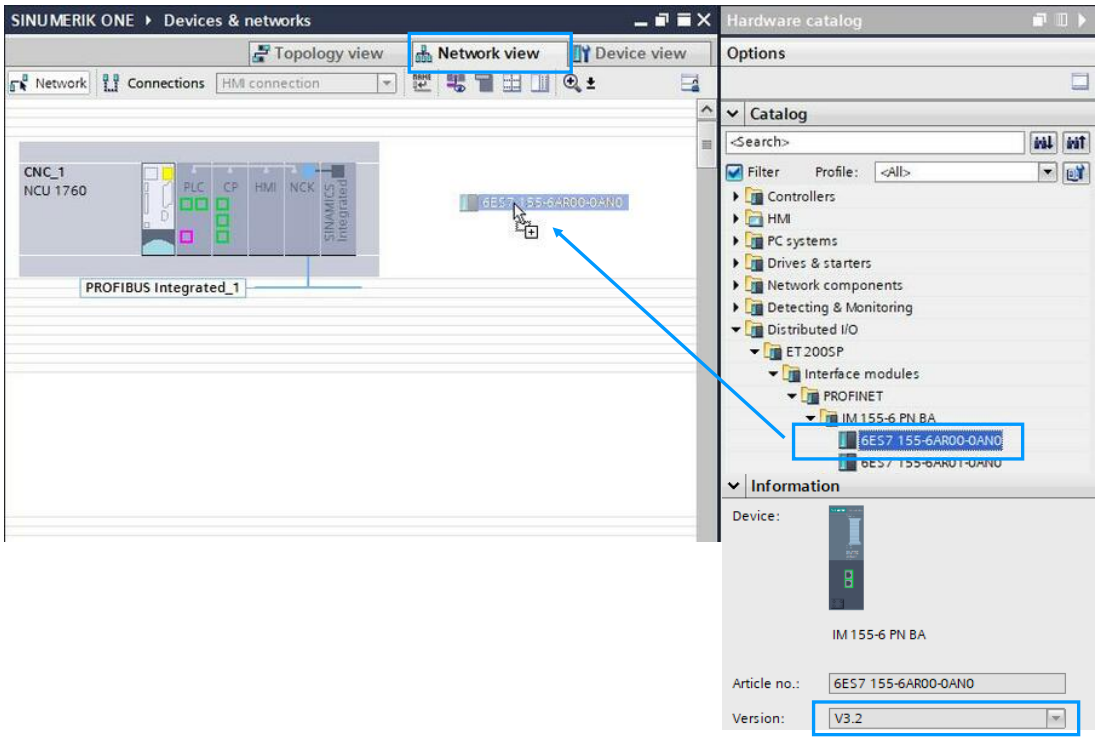
New setting of the start addresses.



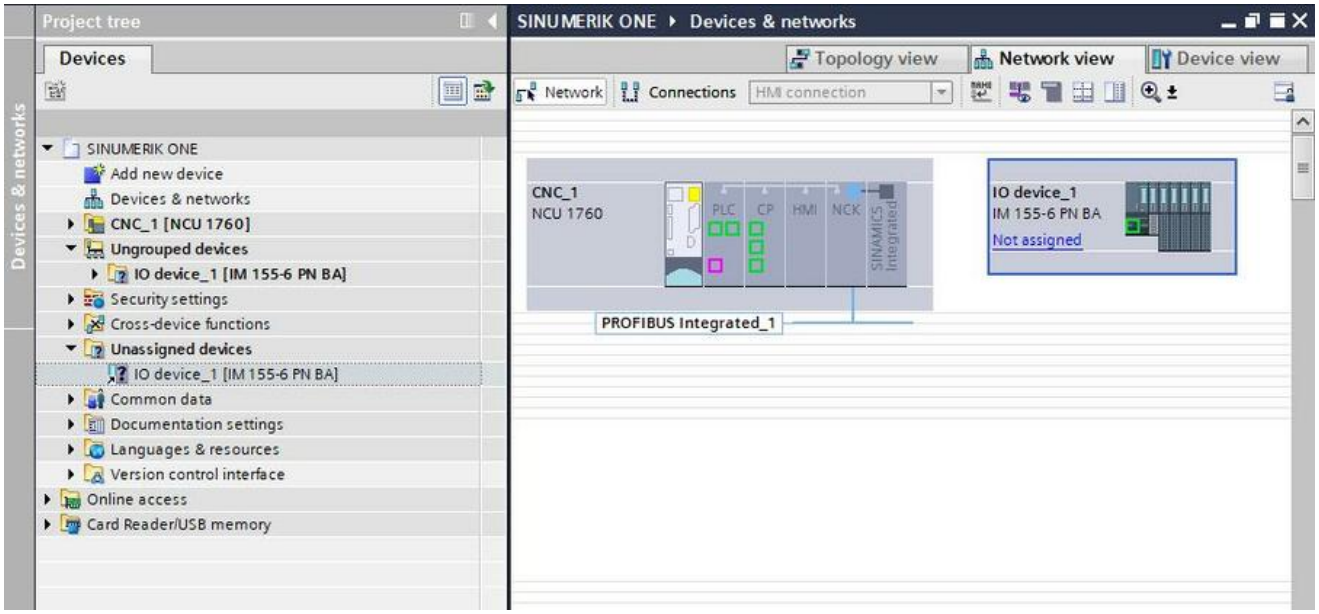
Notes:

Adding a PLC peripheral

Select the PLC peripheral with the suitable version e.g. V3.2 from the hardware catalog.



The PLC peripheral is added.

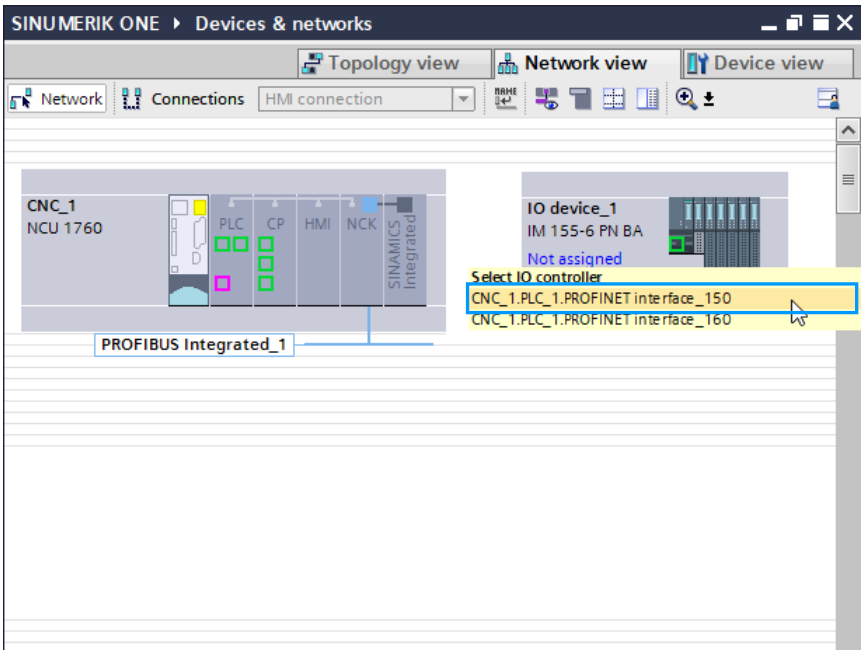


Notes:

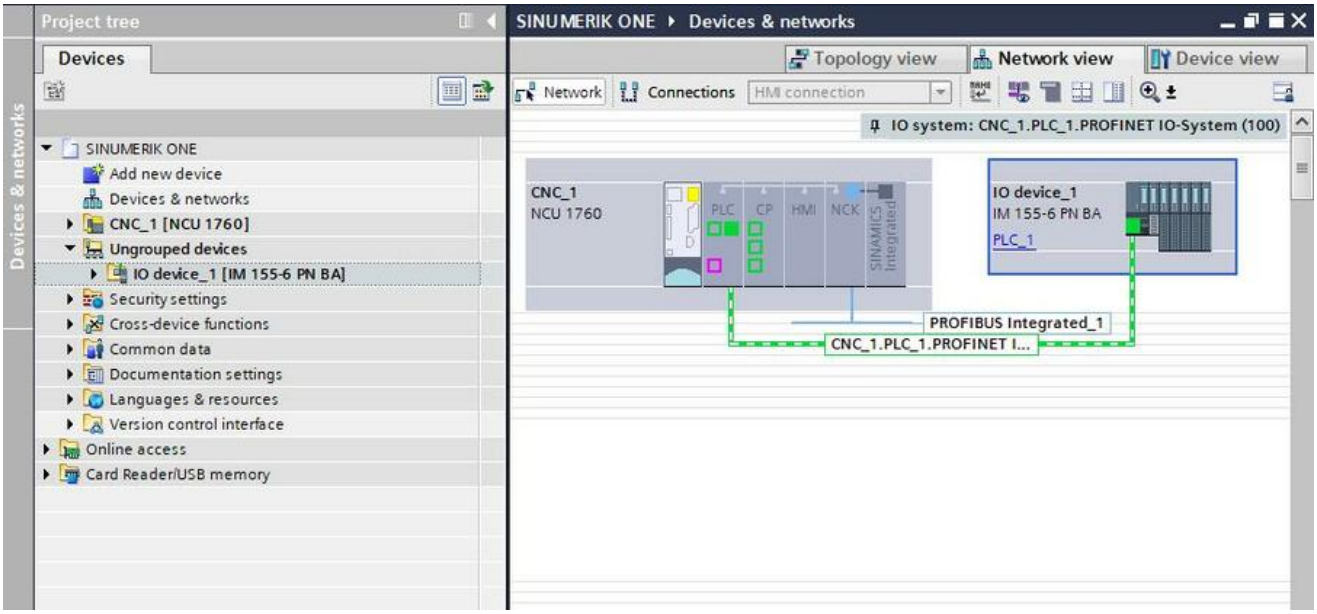
Configuring the PROFINET network

The PROFINET system can now be extended with additional modules.

To assign the module, click Not assigned to select the IO Controller to which the new module should be assigned.



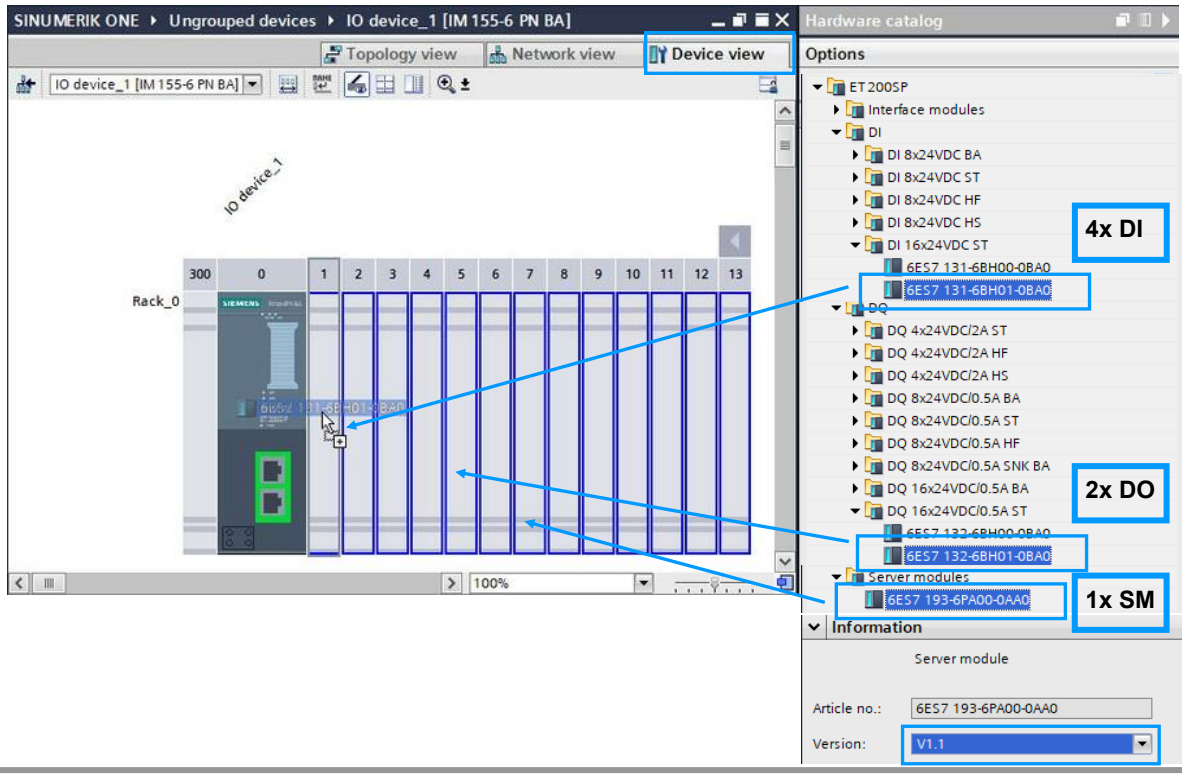
The PROFINET IO system is extended.



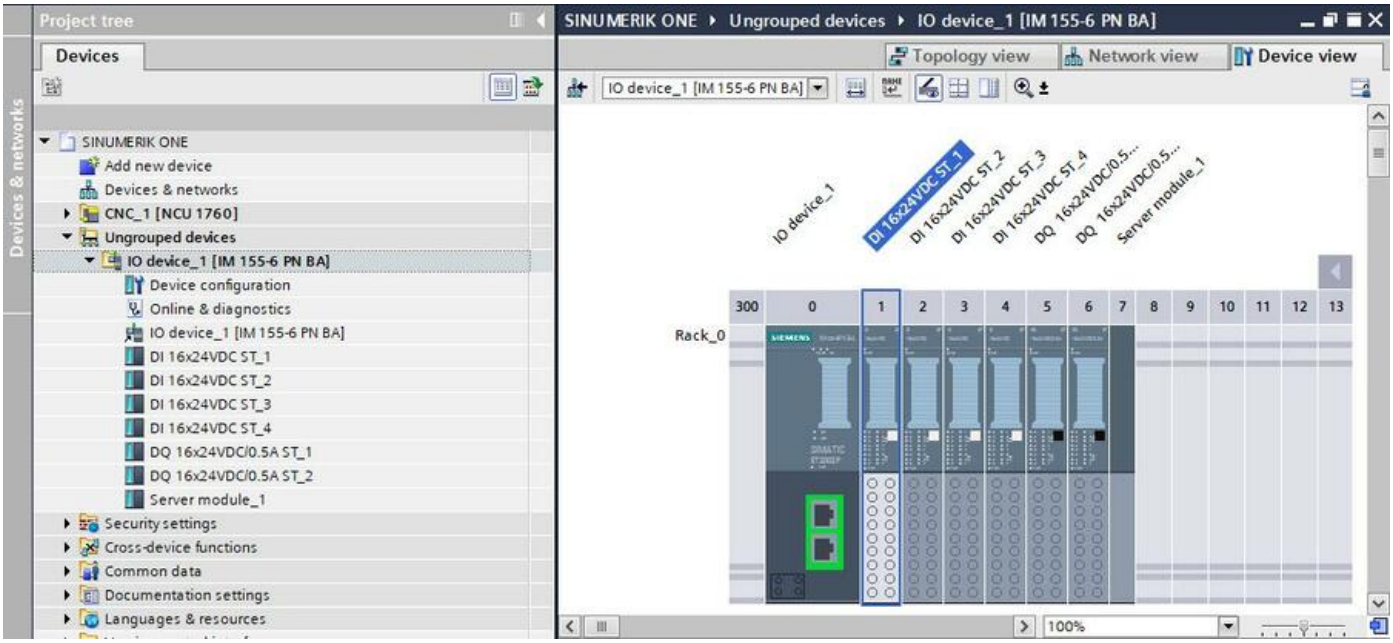
Notes:

Adding peripheral components

Select the individual components with the suitable version e.g. DI/DO V0.0 and Server module V1.1 from the hardware catalog.



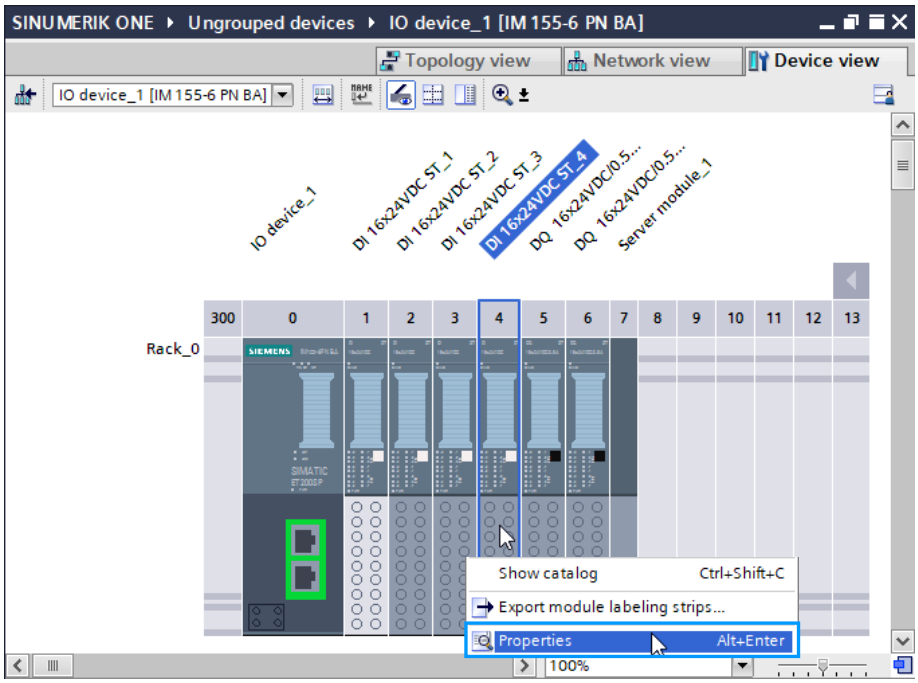
The individual components are added.



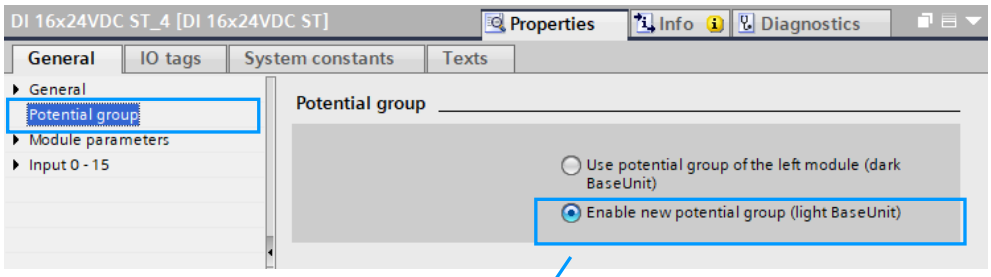
Notes:

Adding a new potential group

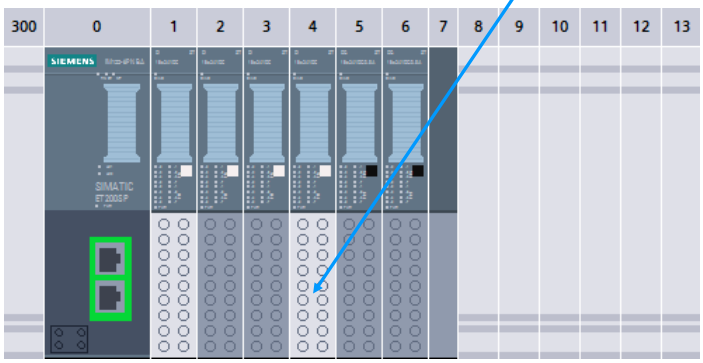
To add a new potential group, select the properties by right-clicking the object.



Select **Enable new potential group**.



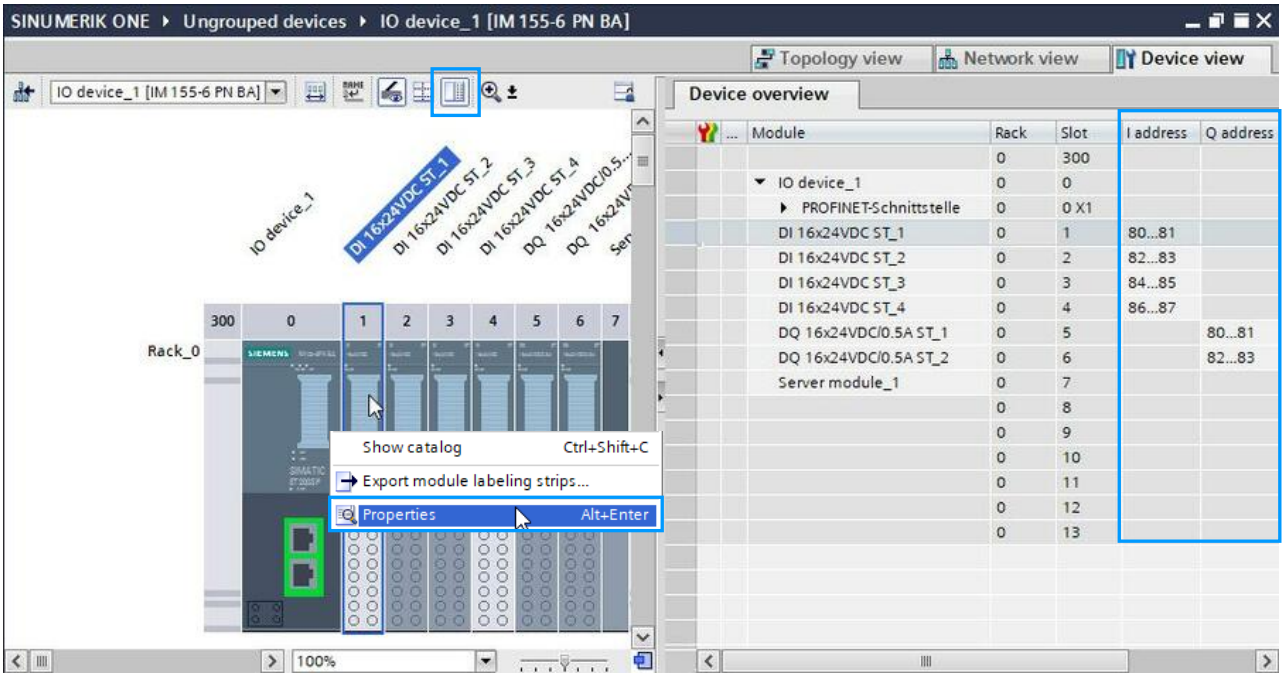
The new potential group is added.



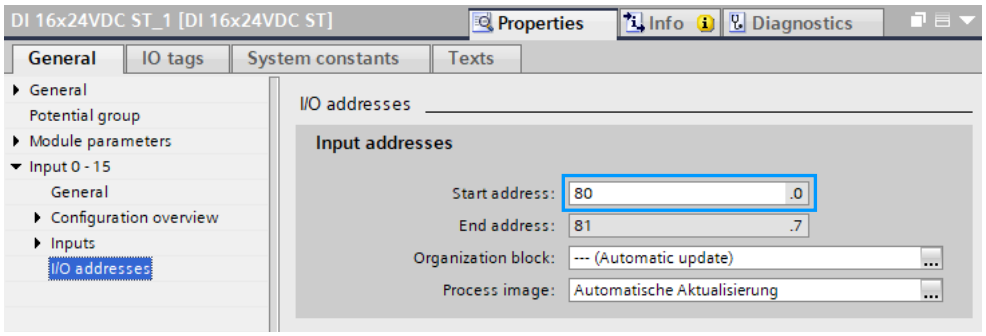
Notes:

Setting I/O addresses

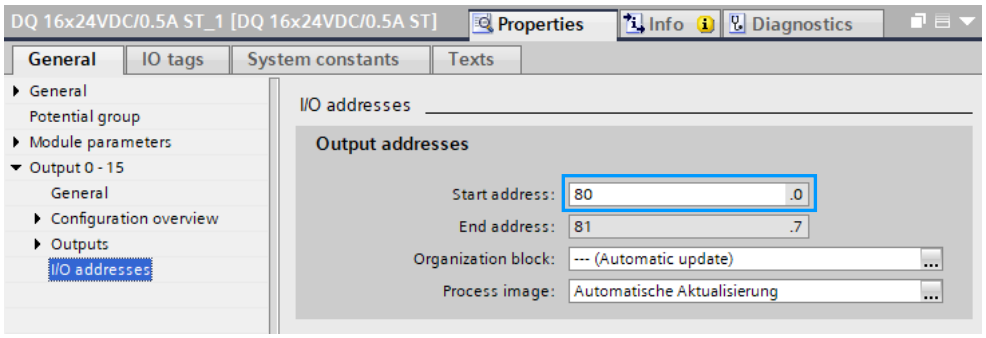
To set the I/O addresses of the components, select the properties by right-clicking the object.



Set start addresses for all DI modules e.g. 80, 82, 84 and 86.



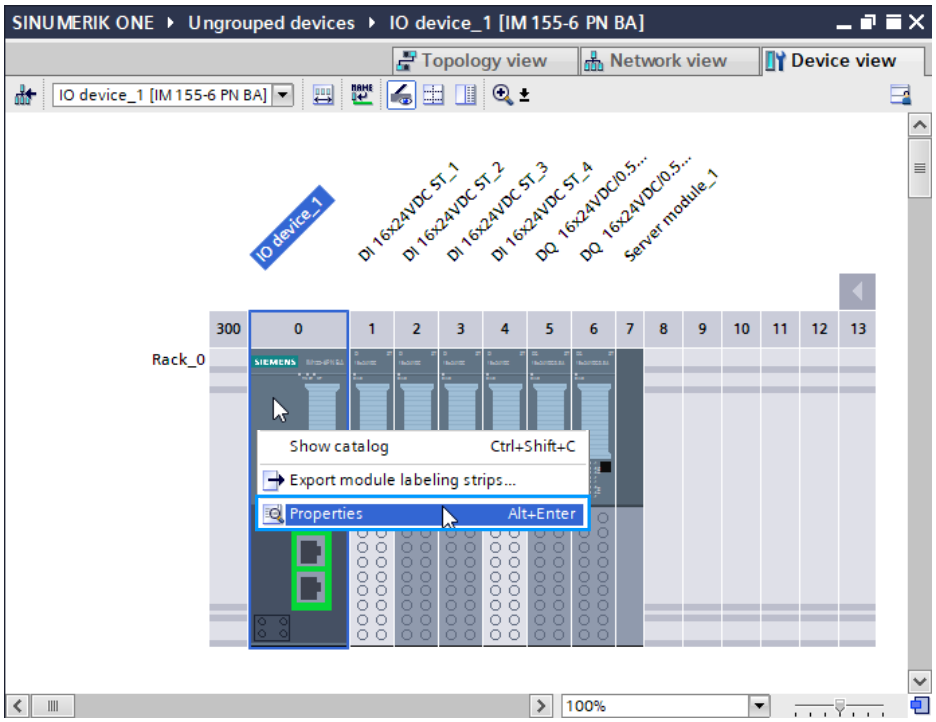
Set start addresses for all DO modules e.g. 80 and 82.



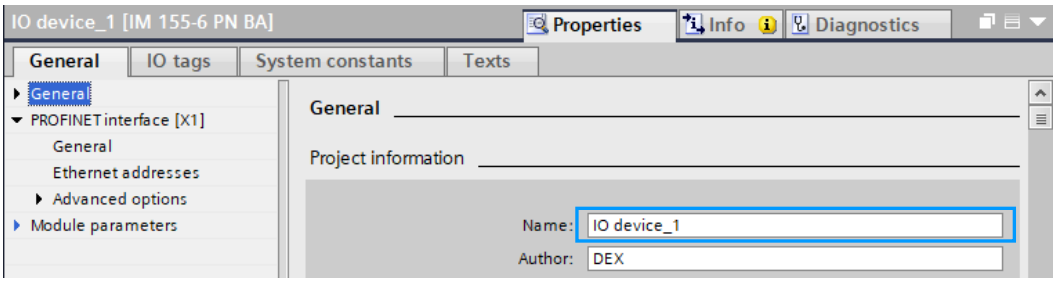
Notes:

Entering the device name

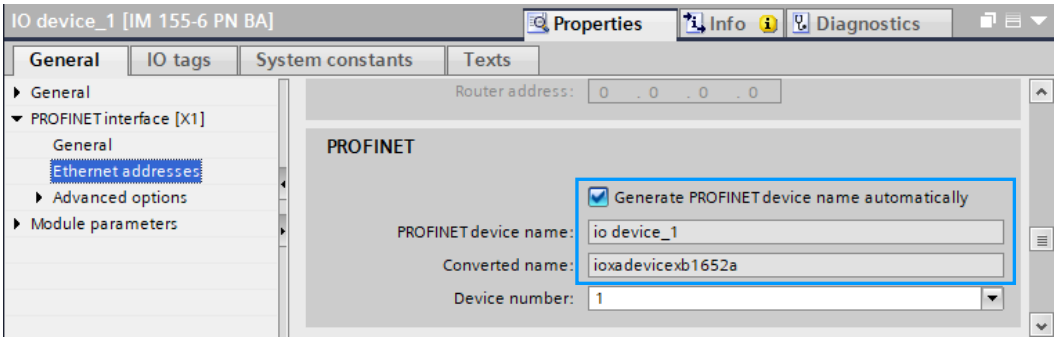
The device name is specified by the system and can be changed, if necessary, select the properties by right-clicking the object.



The device name is assigned automatically by the system, and can be changed, if necessary.



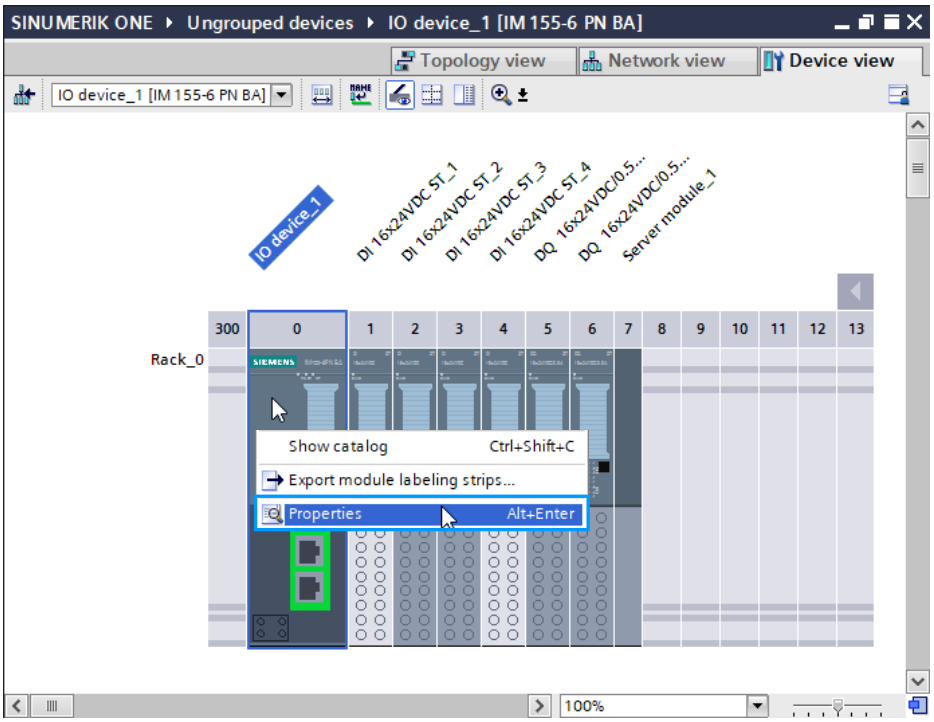
The PROFINET device name is generated automatically from the device name.



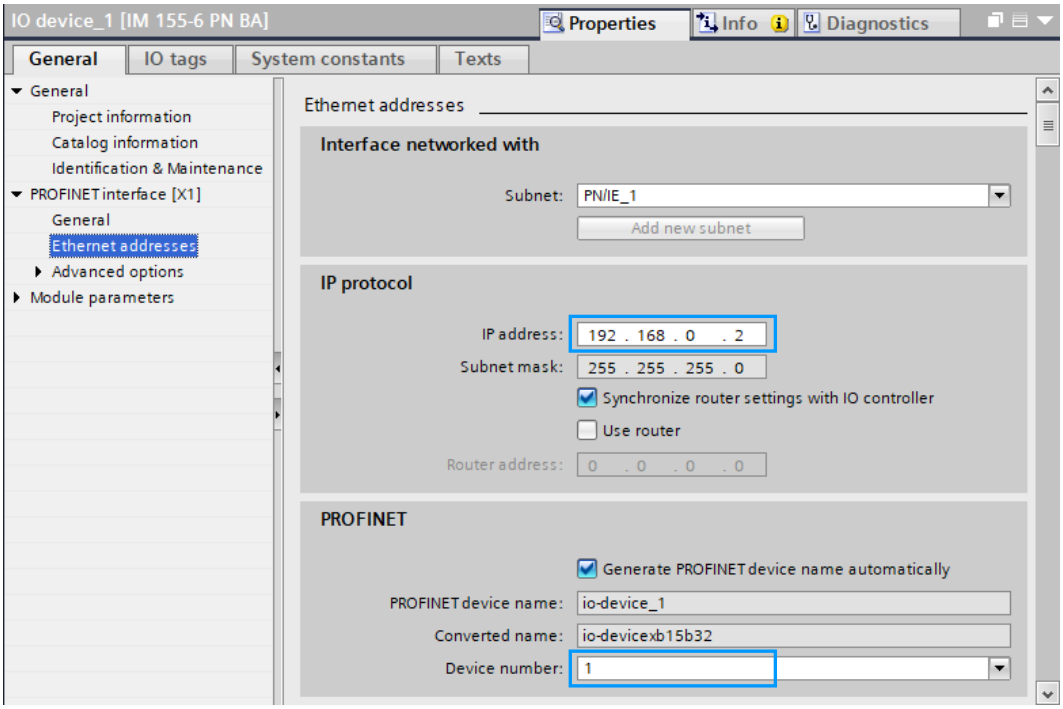
Notes:

Assigning the device number and the IP address

In addition to the topology, the device number and the IP address must be assigned, select the properties by right-clicking the object.



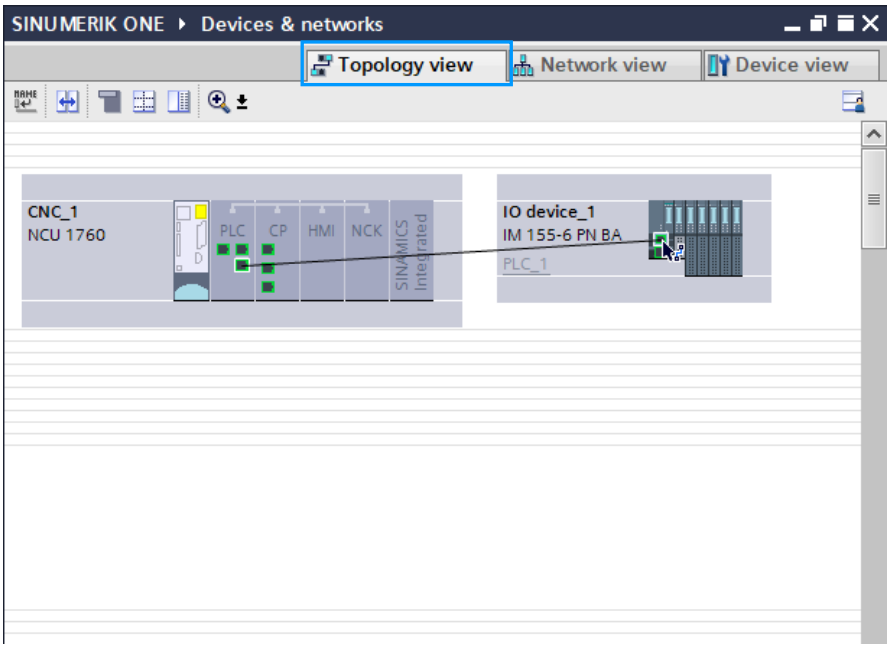
The device number and the IP address are assigned in increasing sequence by the system, and can be changed, if necessary.



Notes:

Configuring the PROFINET topology

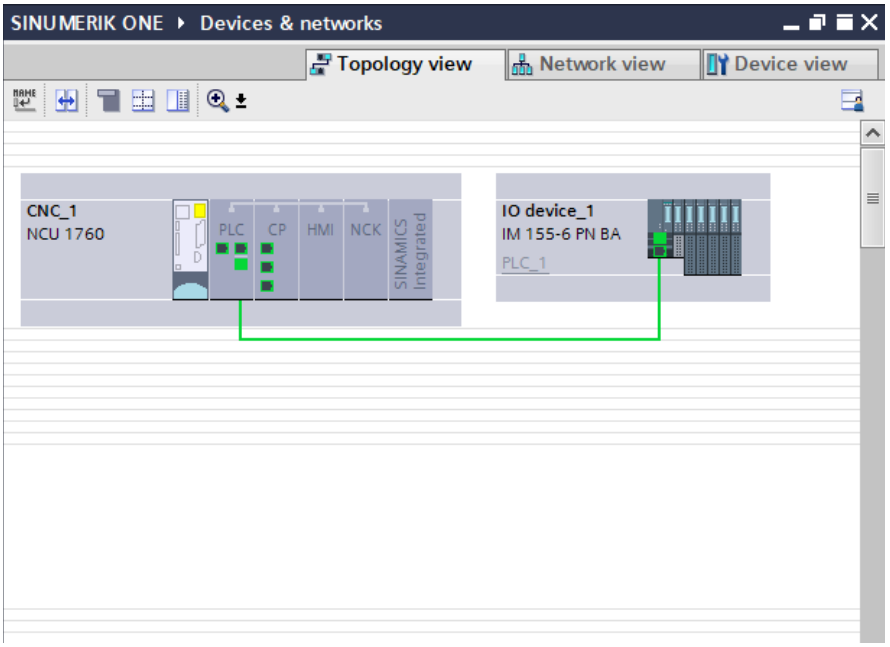
All PROFINET modules must be connected with each other in order to form the PROFINET topology. This is done using the topology view.



Components can be connected in the Topology window using the mouse.

In this case Port_2 [X150.P2] on the NCU is actually connected with Port_1 [X1.P1] on the ET200SP; this must also be represented in the topology.

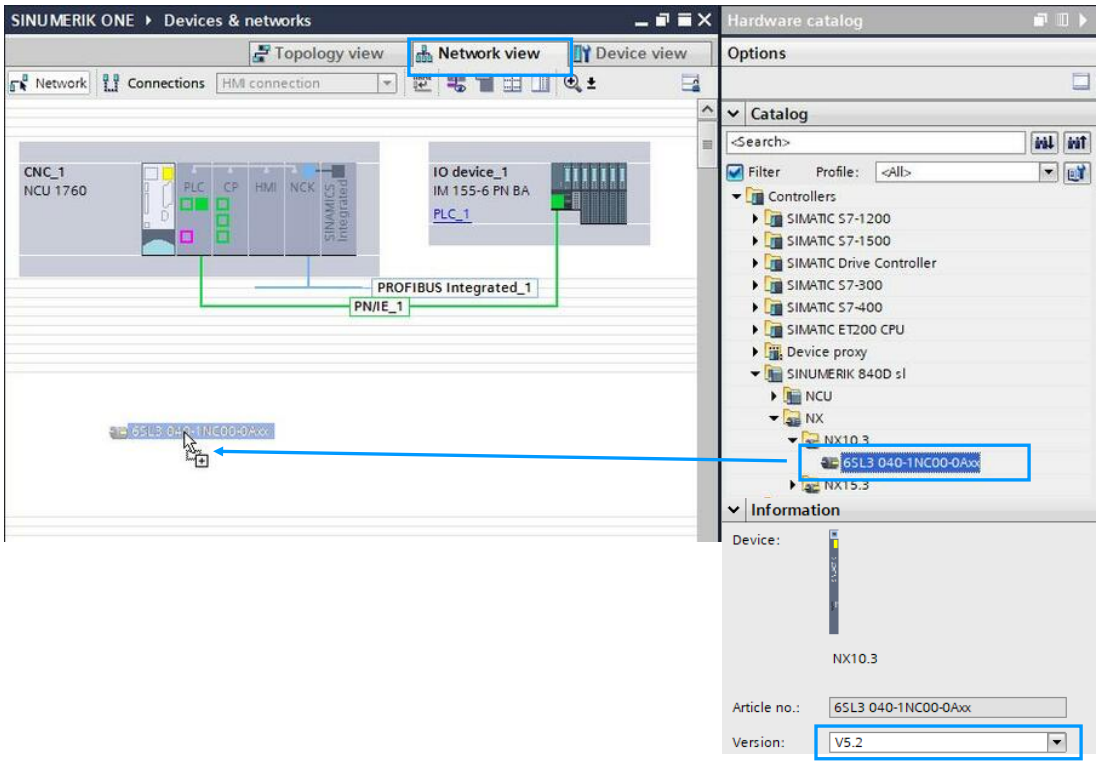
The PROFINET topology is extended.



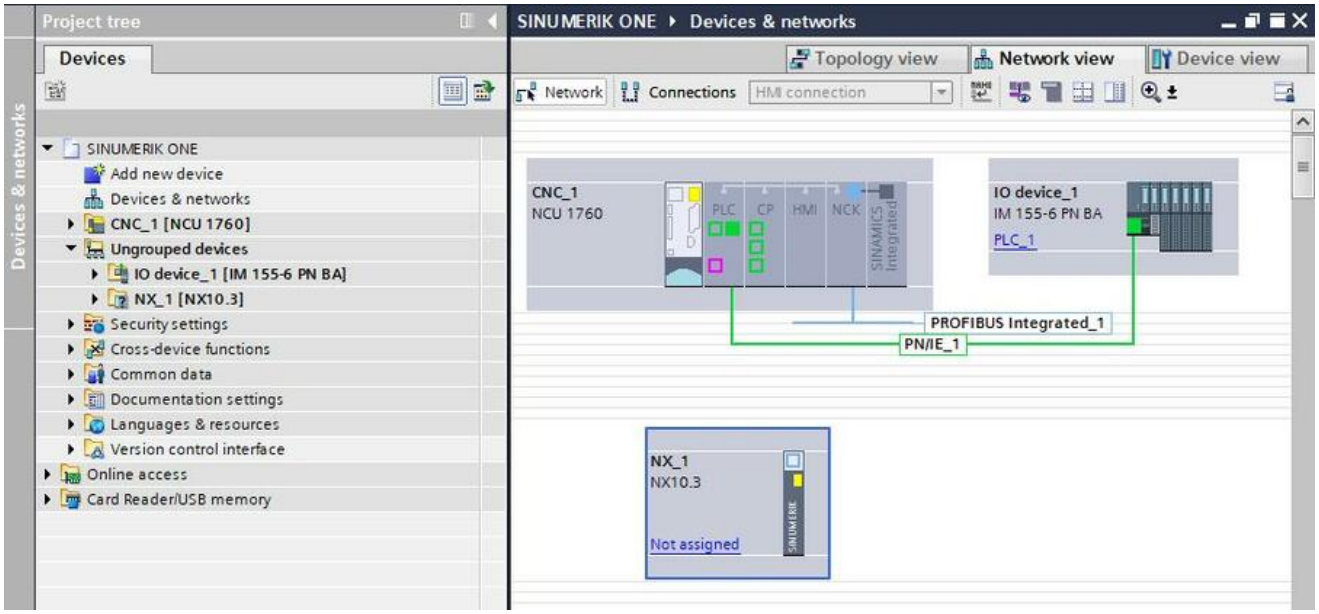
Notes:

Adding additional NC components

Additional axes can be added via NX modules with the suitable version e.g. V5.2.



The NX module is added.

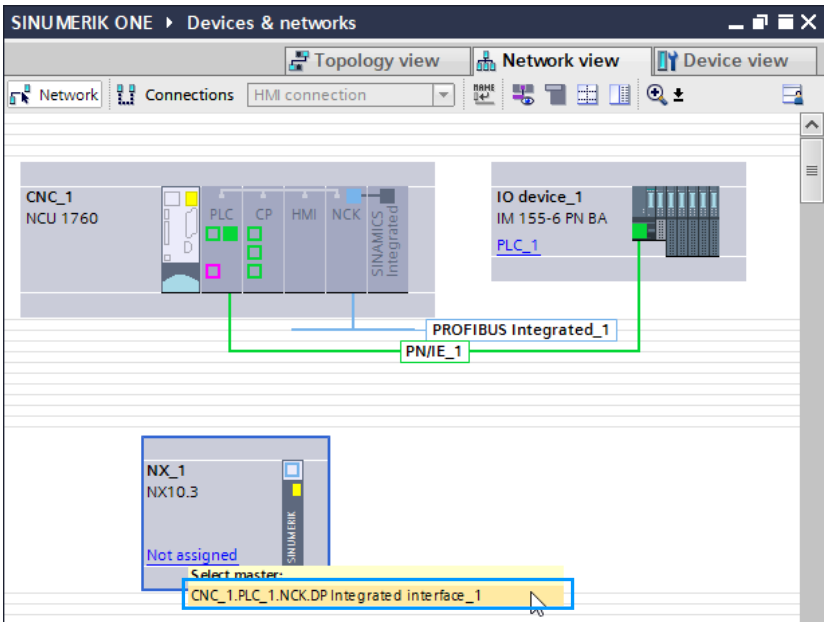


Notes:

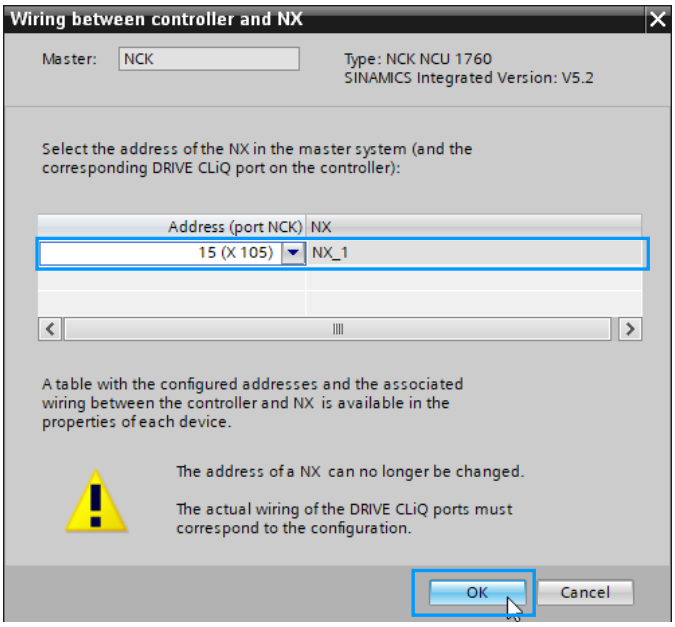
Configuring the SINAMICS Integrated network

The SINAMICS Integrated system can now be extended with additional modules.

To assign the module, click Not assigned to select the master to which the new module should be assigned.

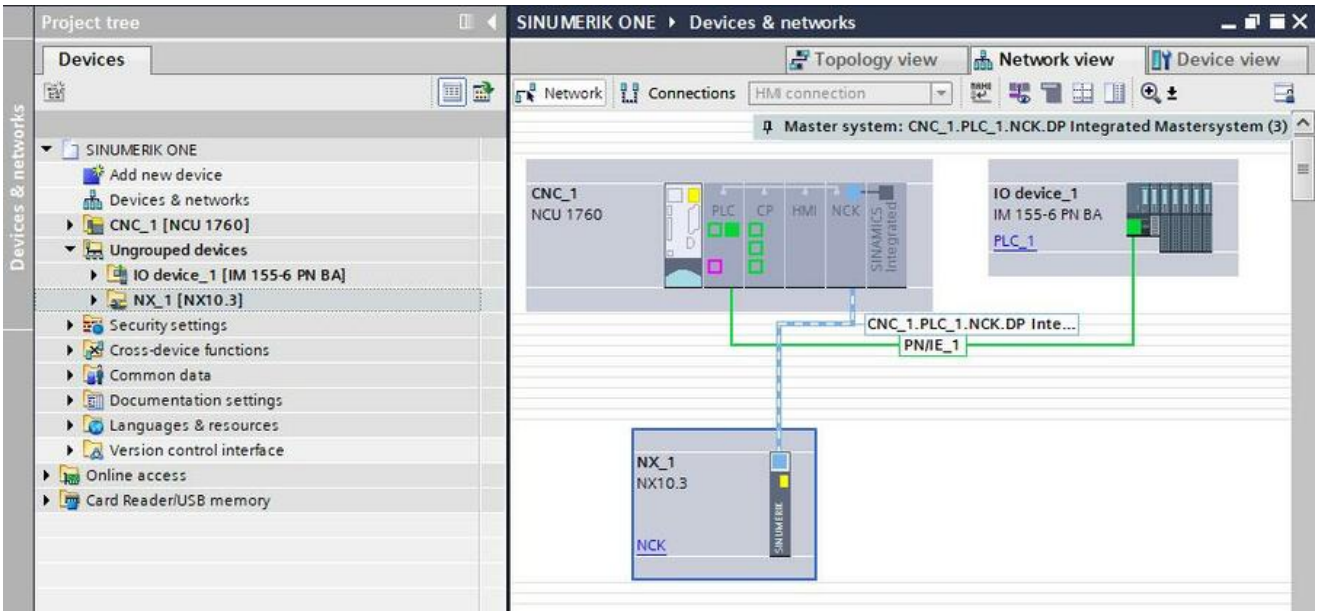


The assignment must be performed in accordance with the SINUMERIK wiring rules: NX_1 -> X105



Notes:

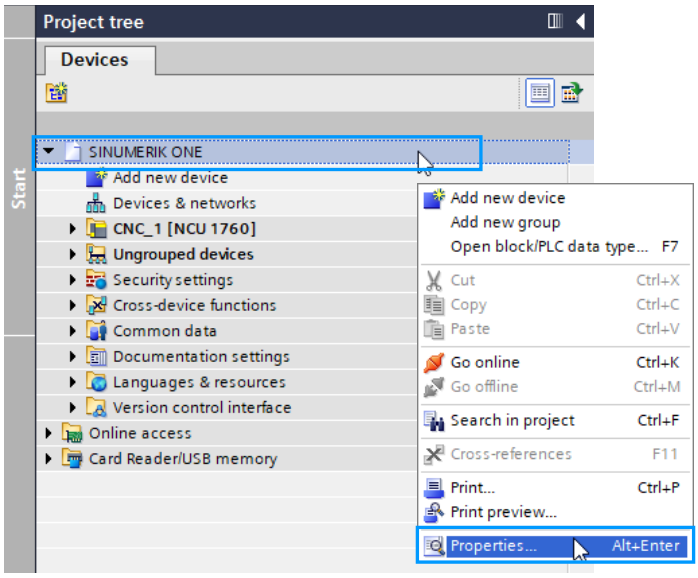
The SINAMICS Integrated system is extended.



Notes:

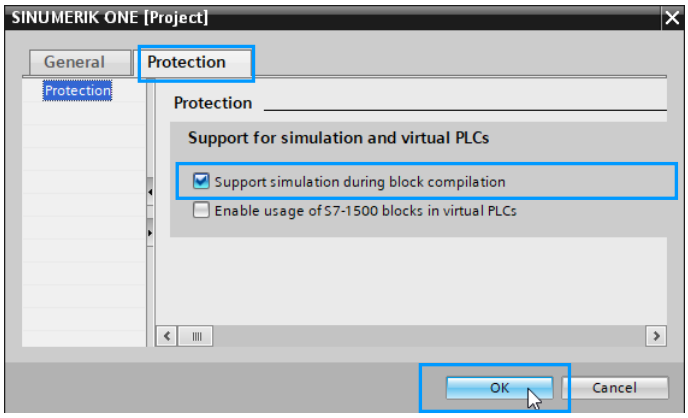
Activate simulation

Select the properties by right-clicking the project.



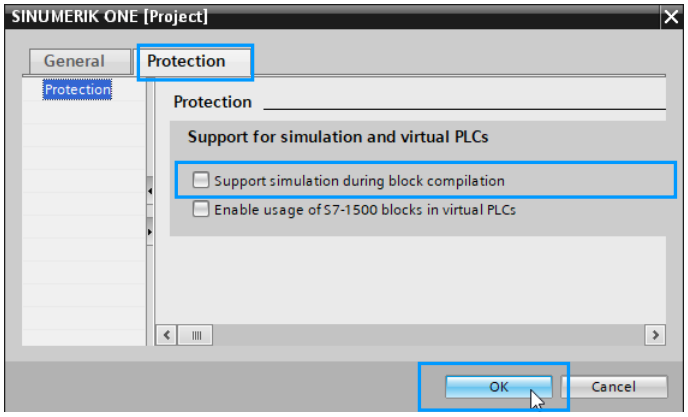
Virtual

Enable simulation with S7-PLCSIM Advanced.



Real

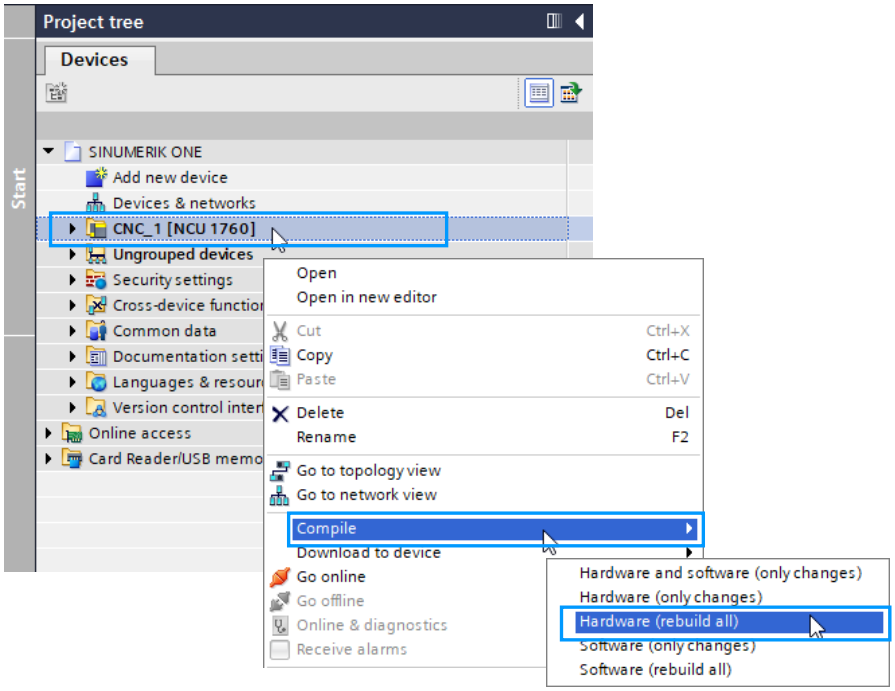
Disable simulation with S7-PLCSIM Advanced.



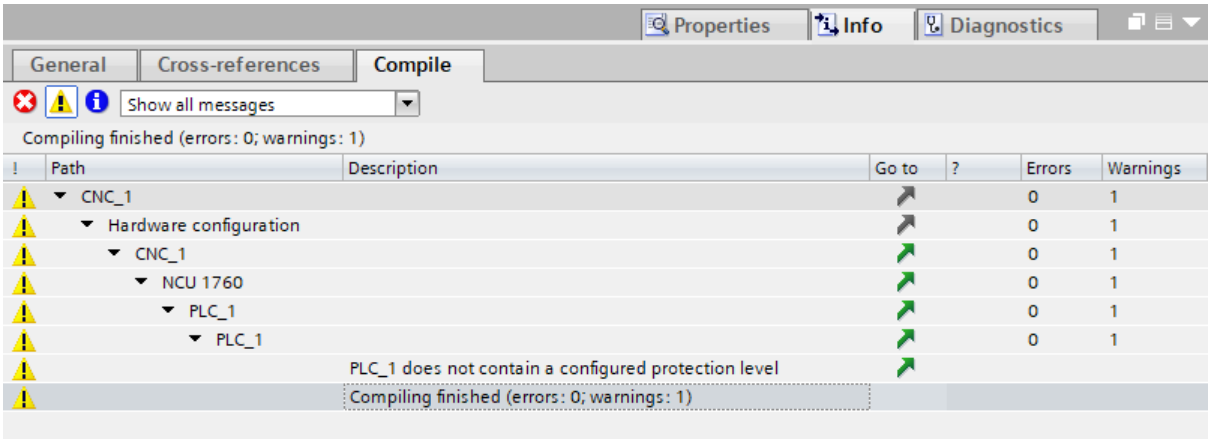
Notes:

Saving and compiling the hardware configuration

To compile the hardware configuration, right-click the marked CNC or PLC.



The compilation result is displayed in the inspector window.

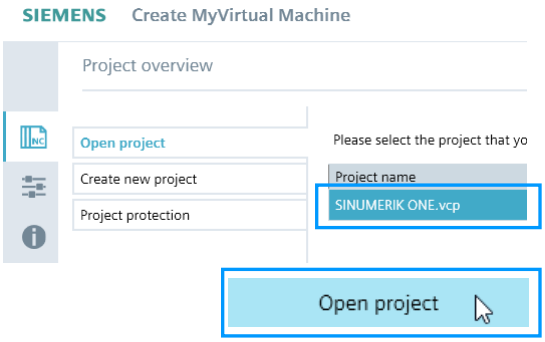


Notes:

Virtual

Starting the virtual machine

Open the project, e.g. SINUMERIK ONE.



Switch on the machine.

Real

Resetting the PLC

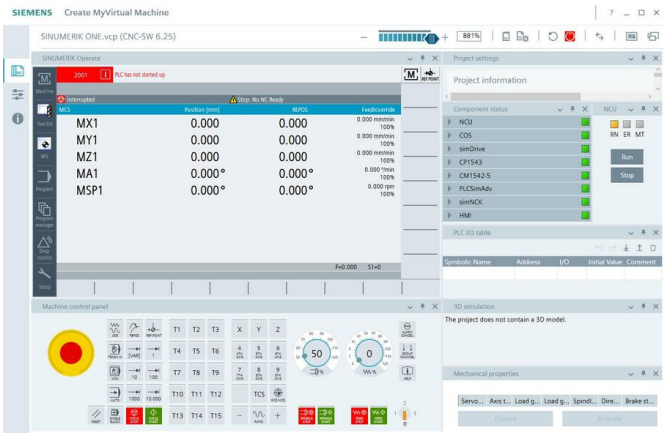
Before the project is loaded, the PLC should be brought into a defined status. This is done by resetting the PLC.

The following steps are necessary:

1. Switch off NCU
2. Set the PLC to STOP (middle switch position) and switch on the NCU
3. Wait until the 7-segment display shows the digit “5”
4. Move the toggle switch to position MRES and hold until the 7-segment display changes from digit “5” to digit “1”
5. Release the toggle switch
6. Within 3 seconds set the toggle switch again to MRES and release -> RUN-LED indicates the overall reset by flashing rapidly
7. Wait until RUN-LED constant yellow
8. Set the PLC to RUN (upper switch position)
9. Finished

Virtual

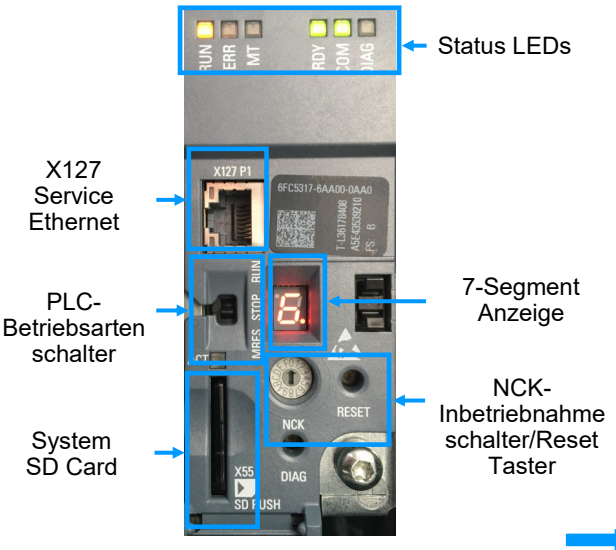
The machine is running, the hardware configuration can now be loaded.



Real

The following LED status should be reached.

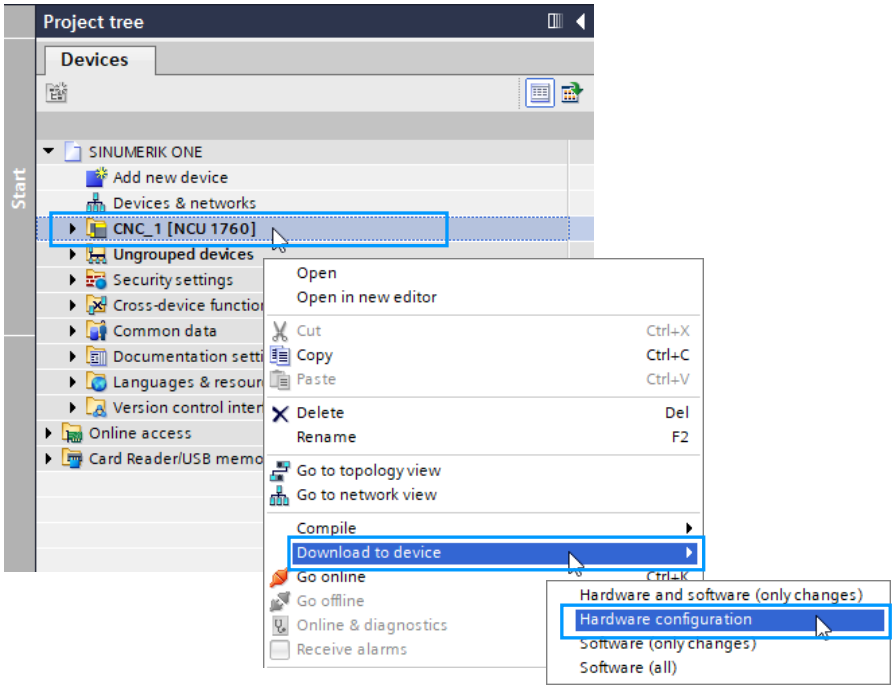
- RUN-LED constant yellow
- RDY-LED flashes green
- COM-LED flashes green
- 7-Segment display on “6“



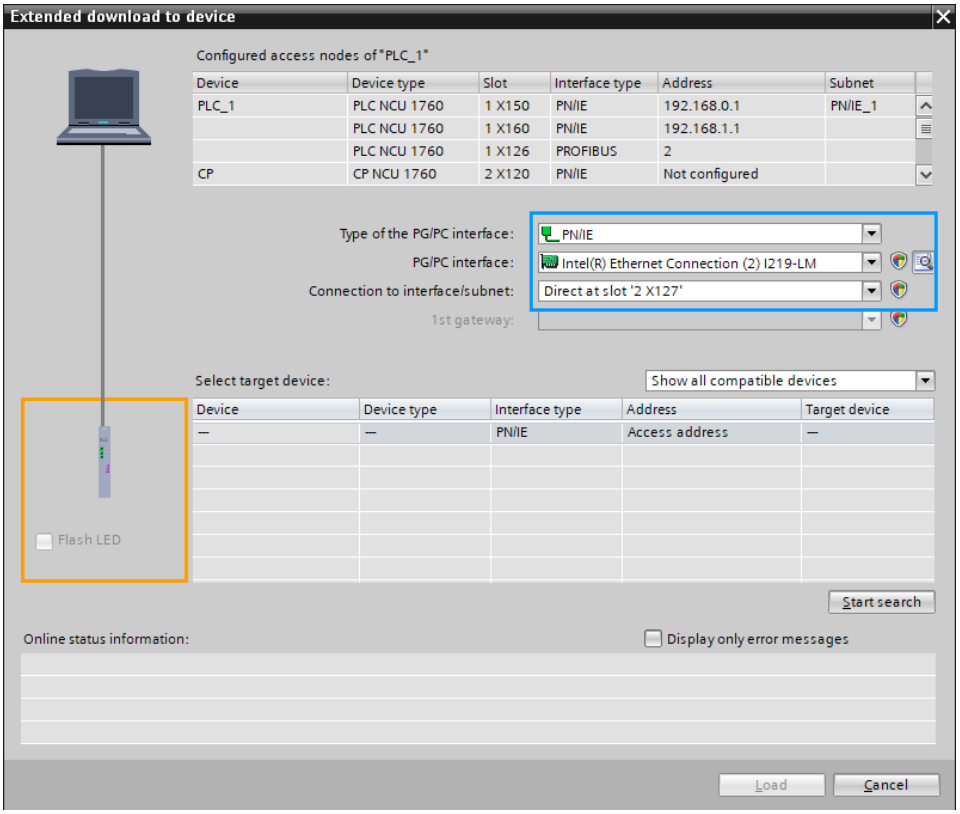
Notes:

Download the hardware configuration to the device

To download the hardware configuration to the device, right-click the marked CNC or PLC.



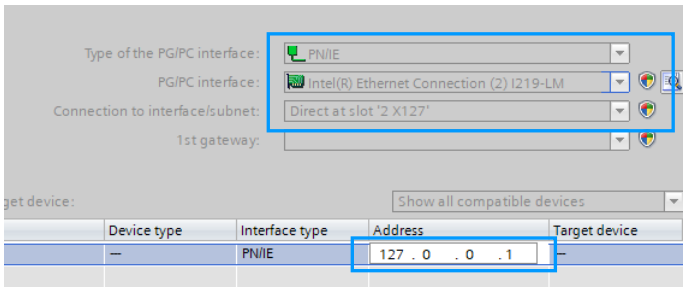
The PG/PC interface is set and the search operation then started. The "Extended download" dialog opens automatically when no connection data exists in the project.



Notes:

The PG/PC interface is set and the search operation then started.

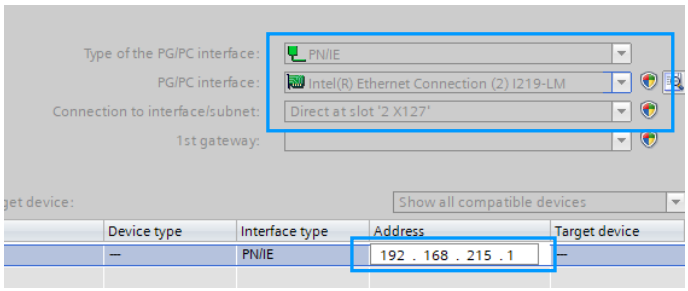
Virtual



Note:

If the Engineering Tools TIA Portal and CMVM are installed on the same PG/PC and no IP address is assigned, 127.0.0.1 can be used for the local host.

Real

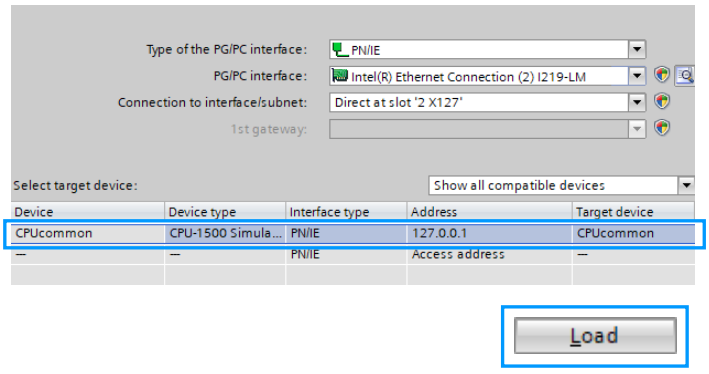


Note:

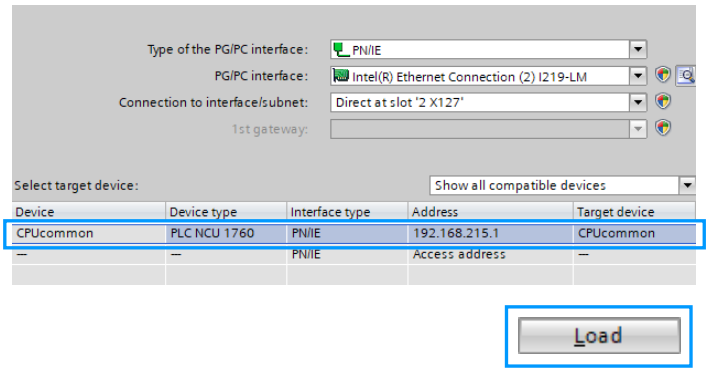
If the PG/PC and NCU are connected via the service interface X127, the IP address 192.168.215.1 can be used.

After the successfully completed scan, the target device is selected and the download started.

Virtual



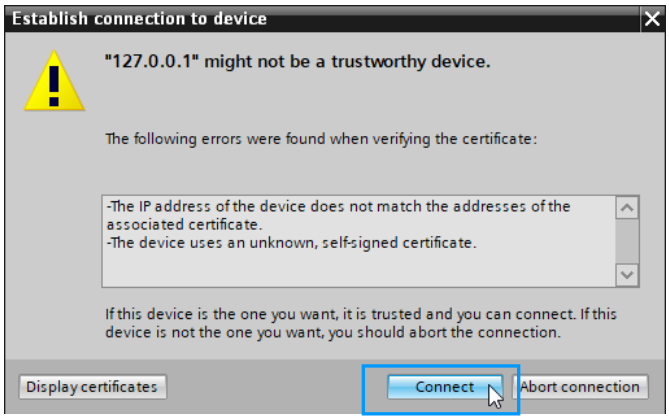
Real



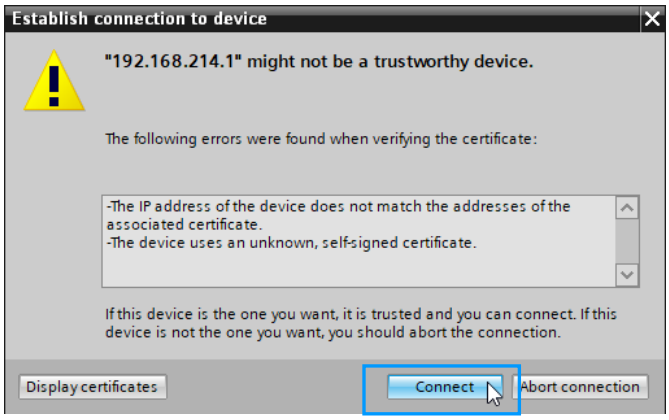
Notes:

If the message below appears select the device as trusted.

Virtual

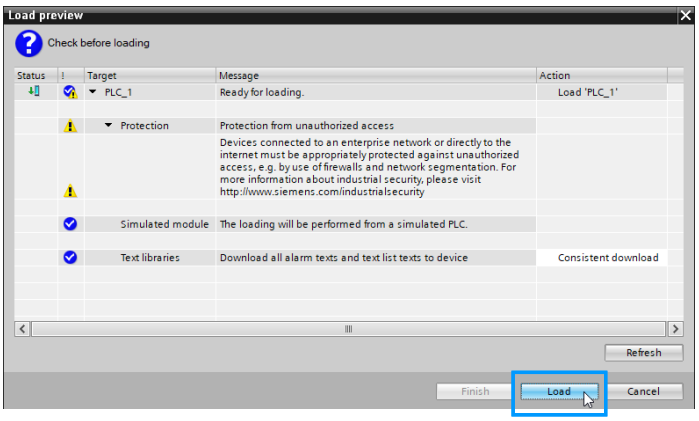


Real

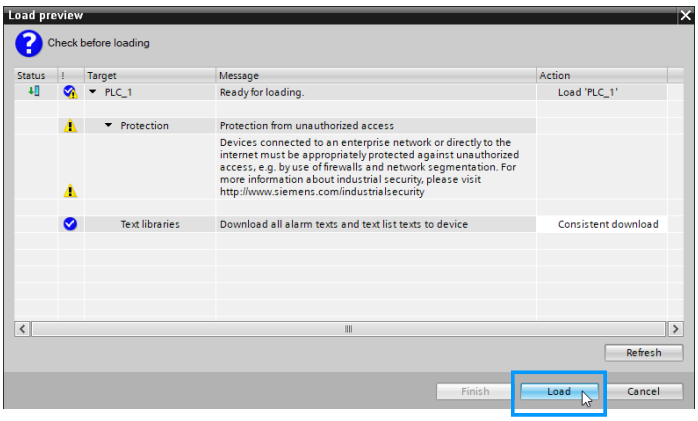


Start the download.

Virtual

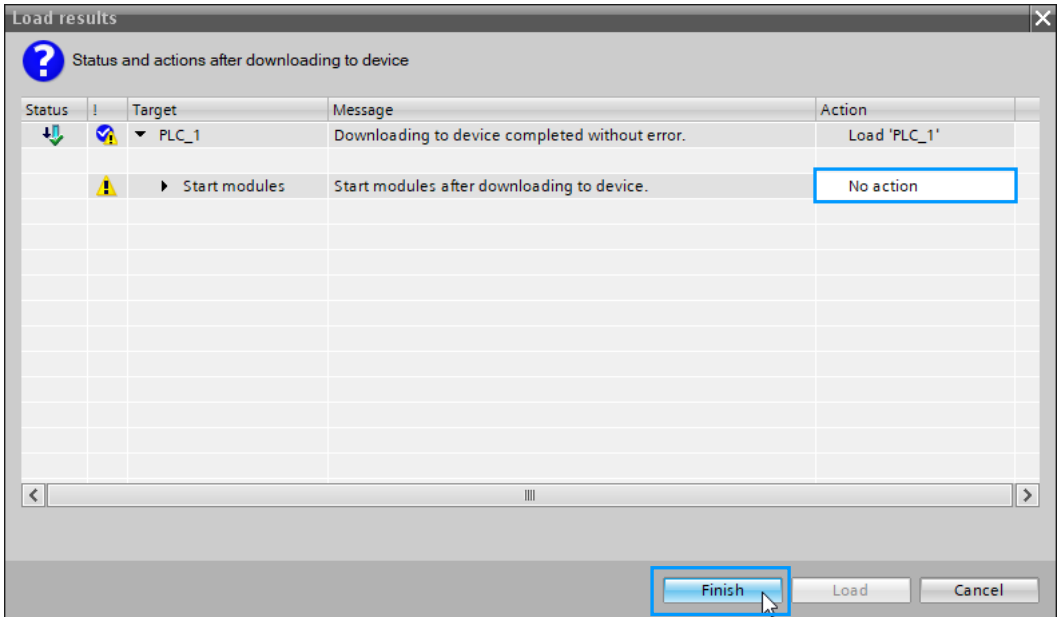


Real

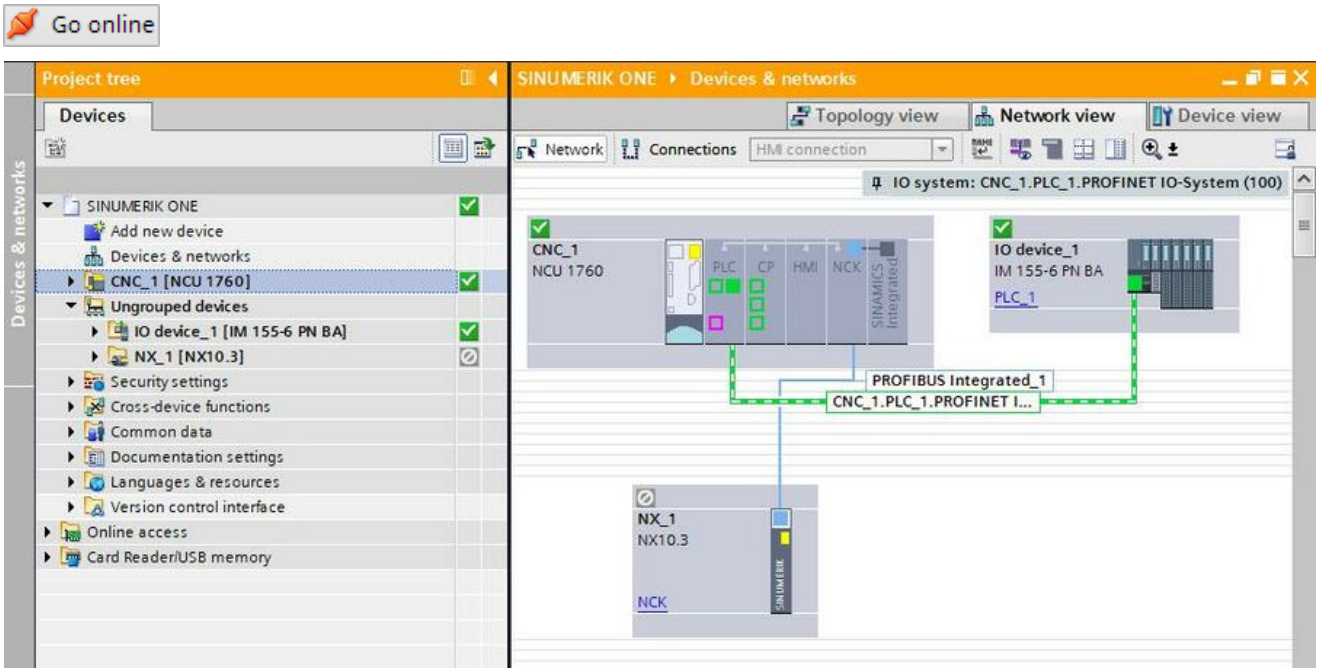


Notes:

Complete the download.



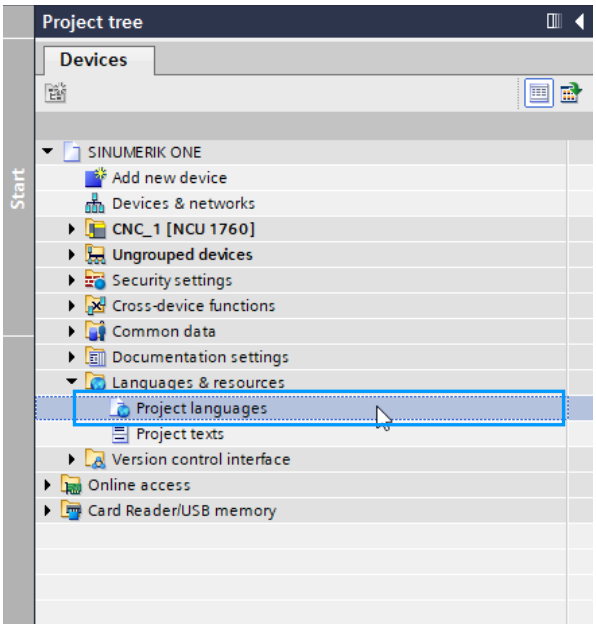
After the successful download, the module status can be checked using the online diagnostics.



Notes:

Import System texts to SINUMERIK Operate

Select Project languages

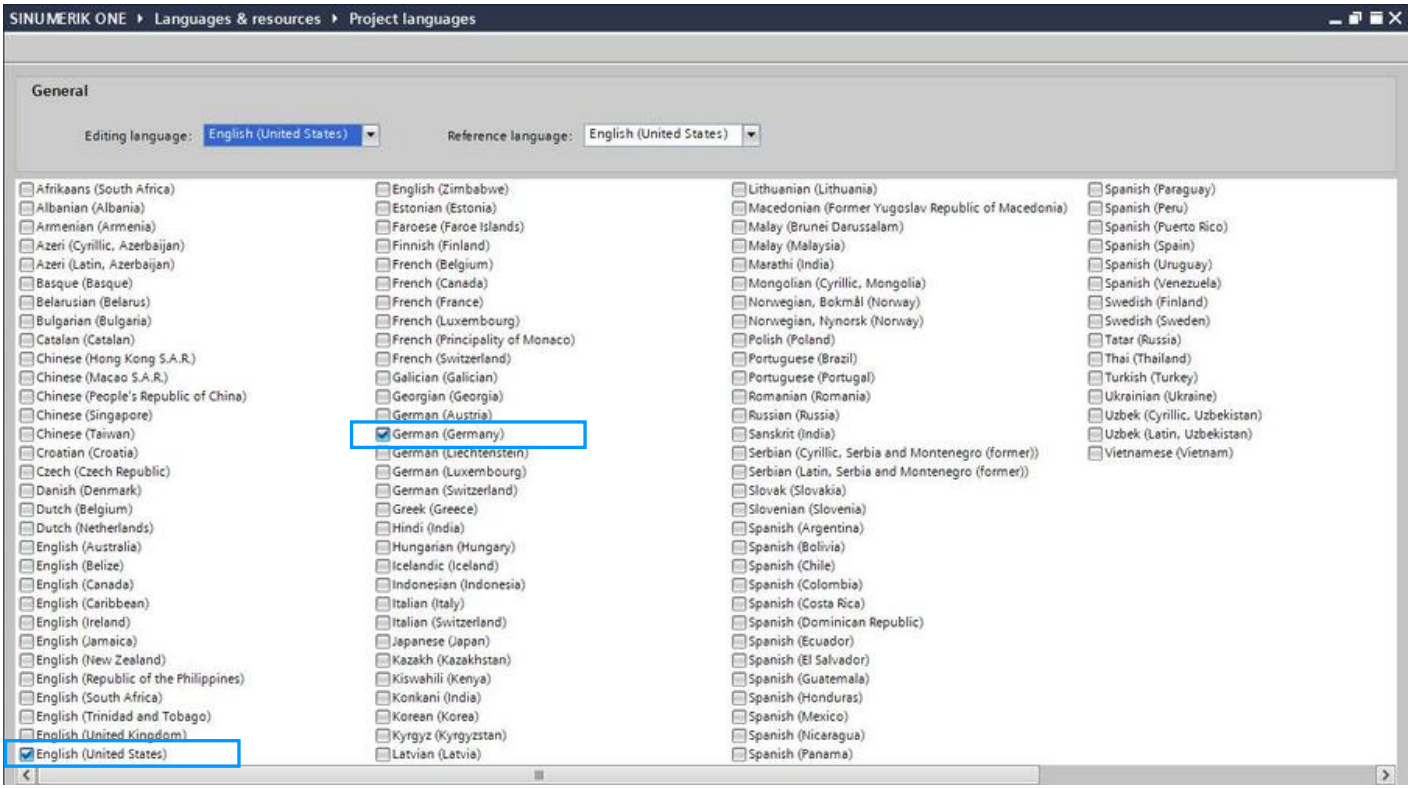


Select HMI/TIA language combinations e.g. German and English.

HMI Language	TIA Project language
German	German (Deutschland)
English	English (United States)

Note:

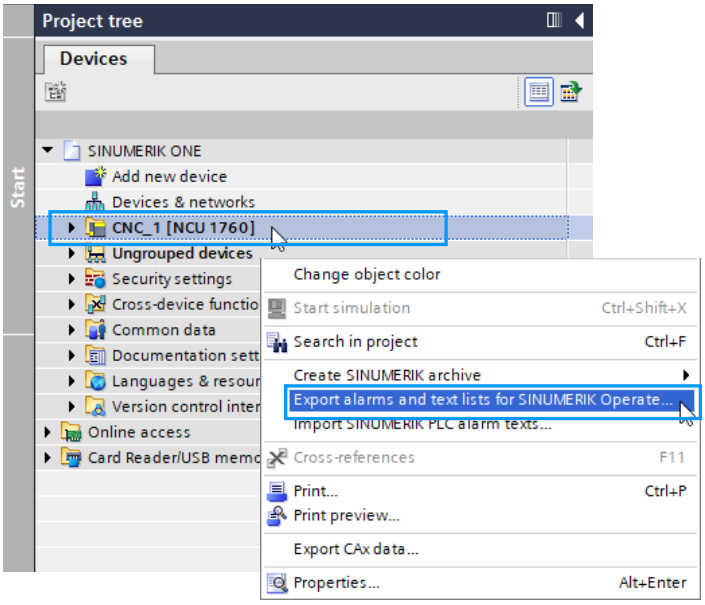
When selecting the project languages, please note that only certain HMI/TIA language combinations are possible.



Notes:

Export the system texts from TIA Portal

To export the system texts for SINUMERIK Operate, right-click the marked CNC or PLC.



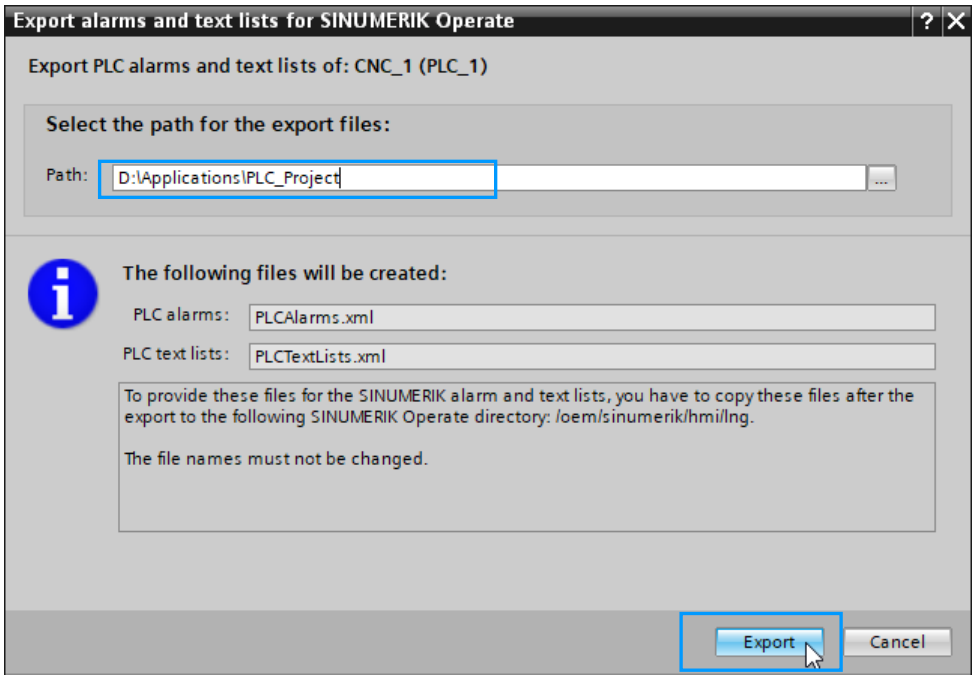
The TIA System messages and the PLC basic program alarms are exported.

Message/Alarm No.	Function
2000000 - 2999999	TIA System messages
400000 - 499999	PLC Basic program alarms

Note:

This step must be repeated after “relevant” changes to the project, e.g. HW configuration gets new devices added, devices are renamed, project contains texts for user alarms, change project languages, change under “Common data” in the project.

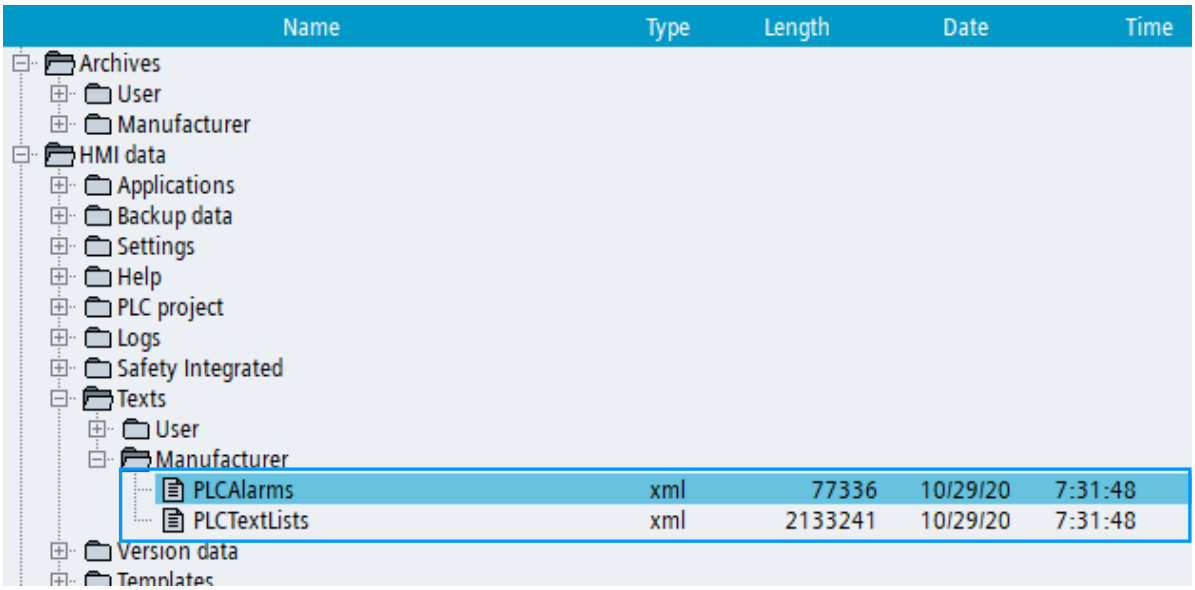
Select the path for the export files.



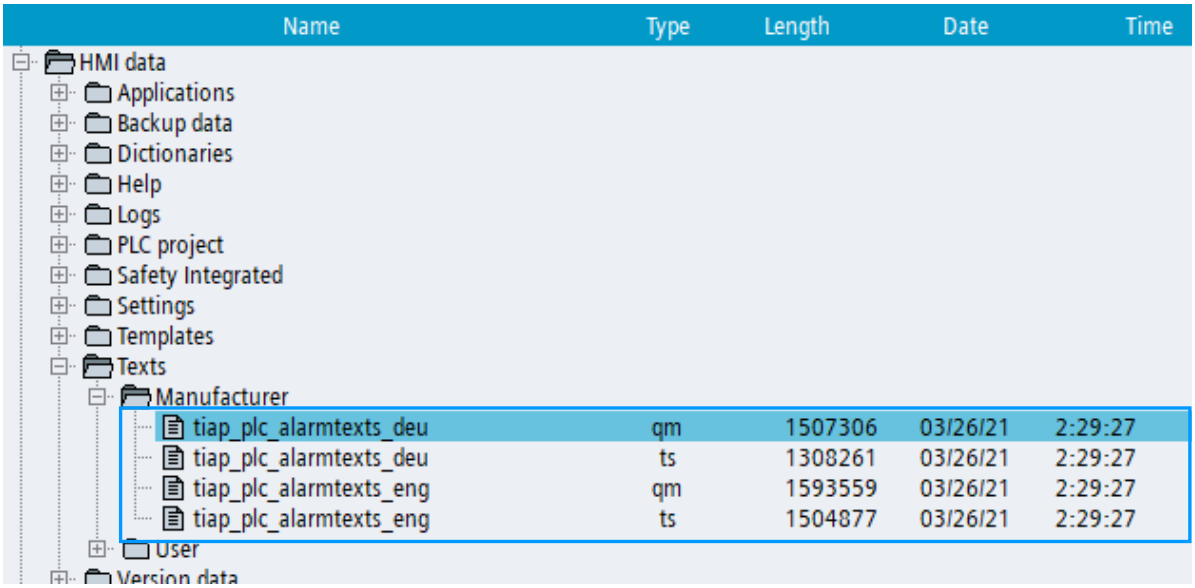
Notes:

Copy the system texts to SINUMERIK Operate

Copy the system texts into SINUMERIK Operate directory
“HMI data\Texts\Manufacturer“.

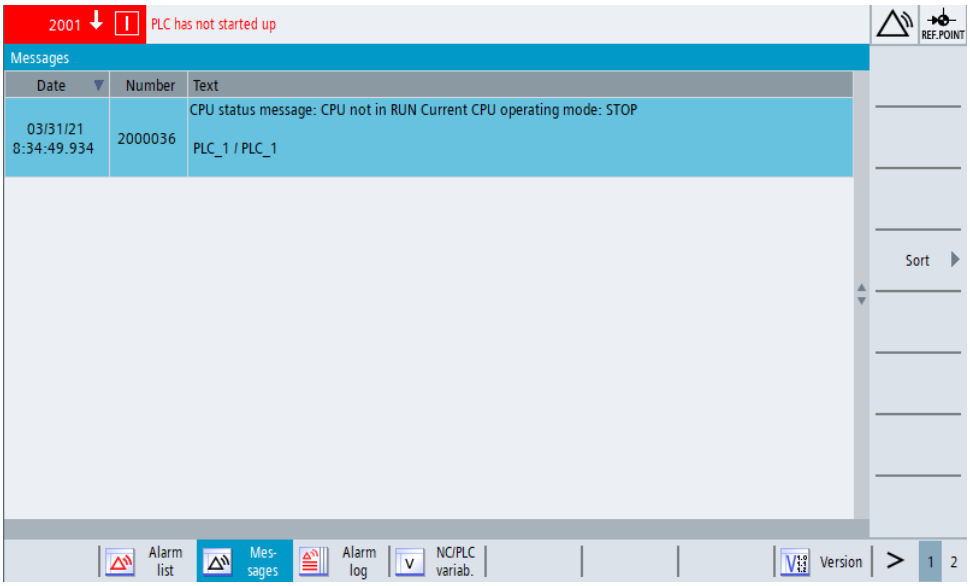


Convert and compile the system texts by switching the control-
ler OFF and ON.



Notes:

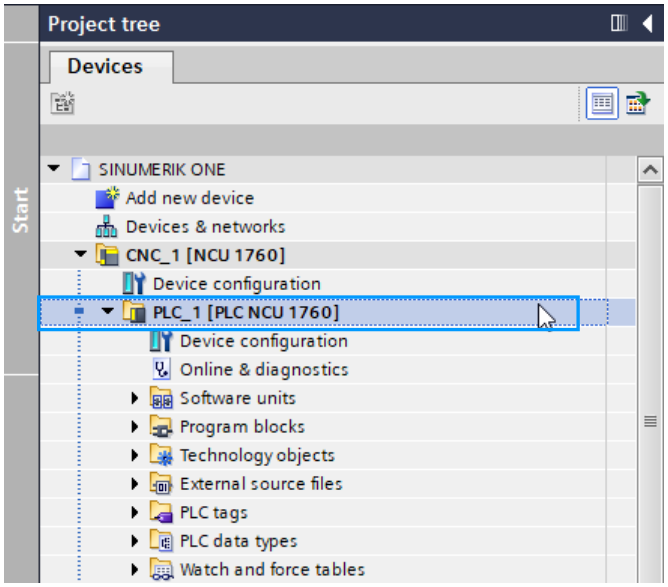
After successful import the system texts are displayed on the screen.



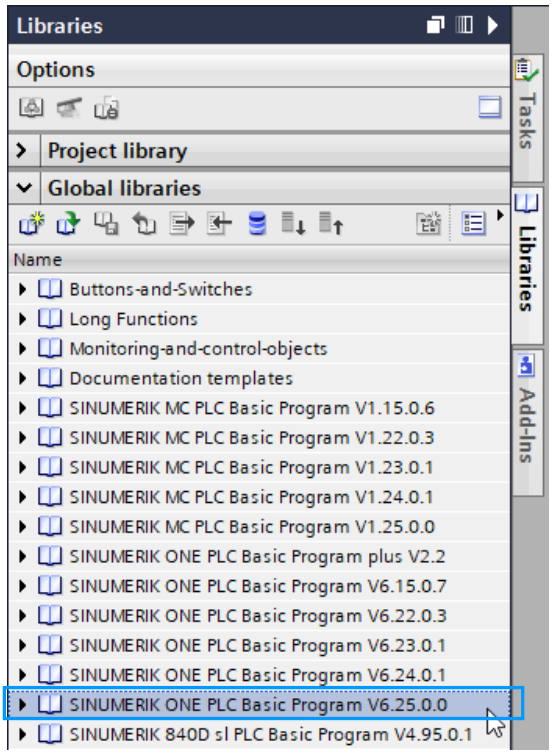
Notes:

Integrating the SINUMERIK ONE Basic Program

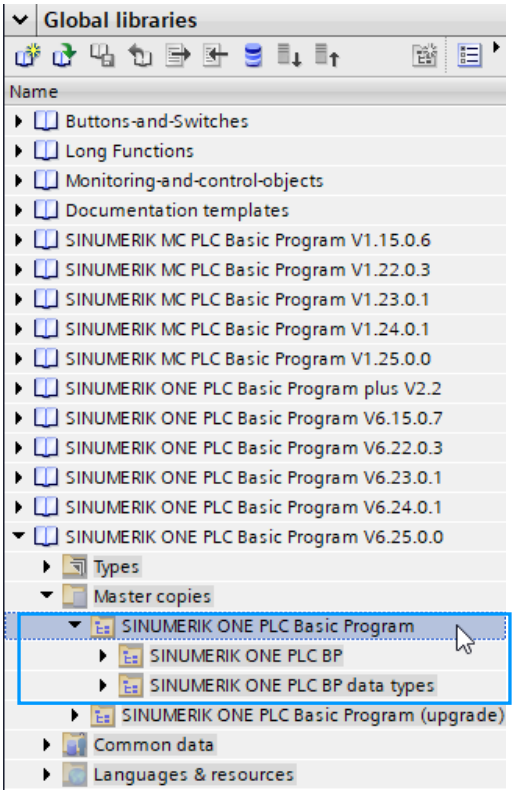
After a successful compilation, the STEP 7 program objects are available for the user.



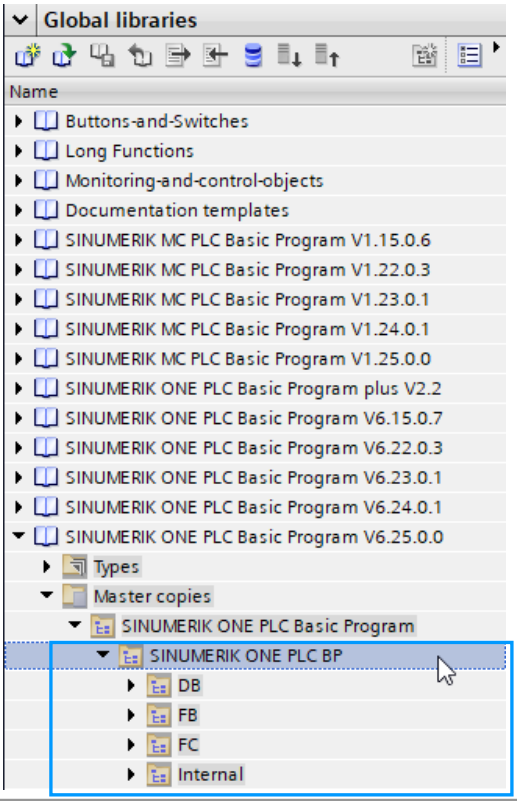
You can now search for the Library which was installed from the Toolbox, and open it in the TIA-Portal.



The Library, as it appears in the TIA-Portal.

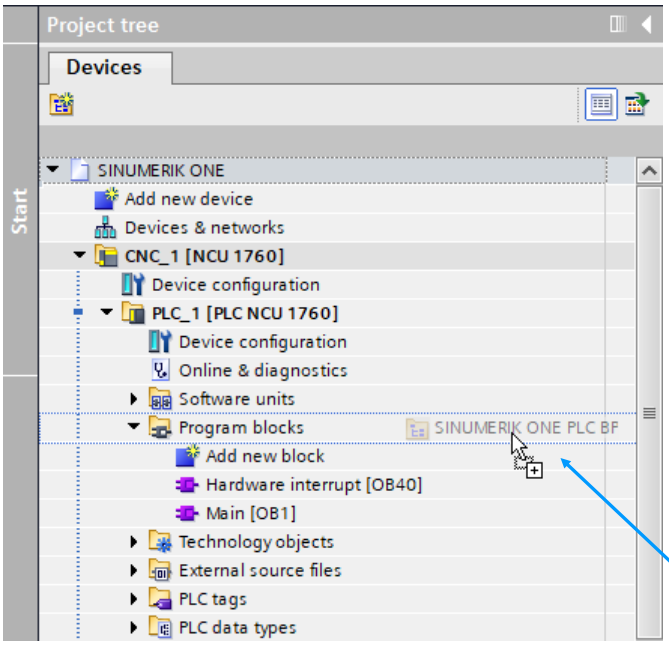


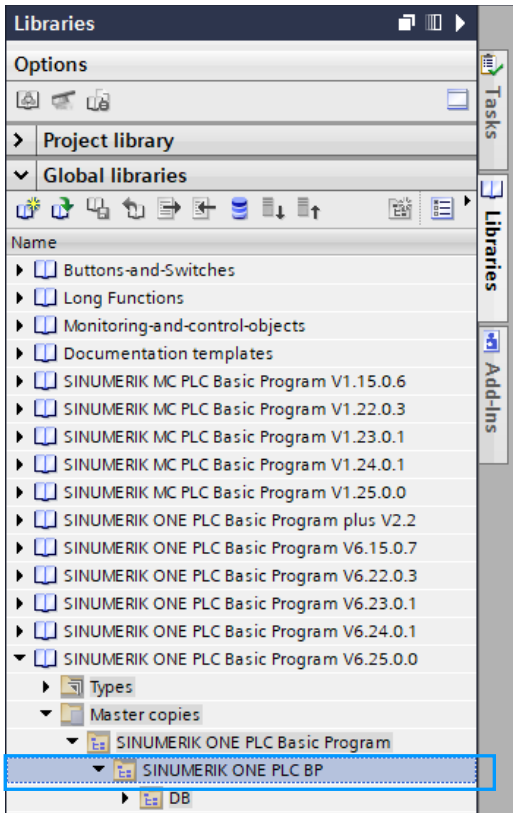
Program blocks in the PLC Basic Program.



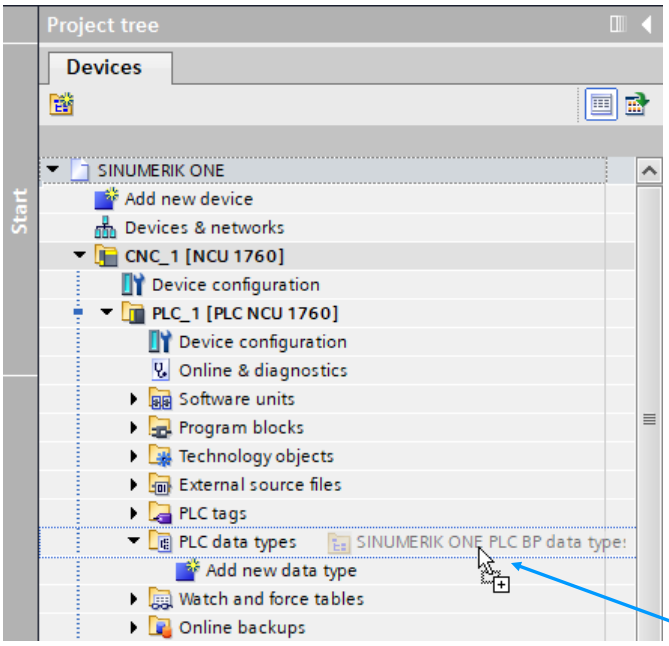
Notes:

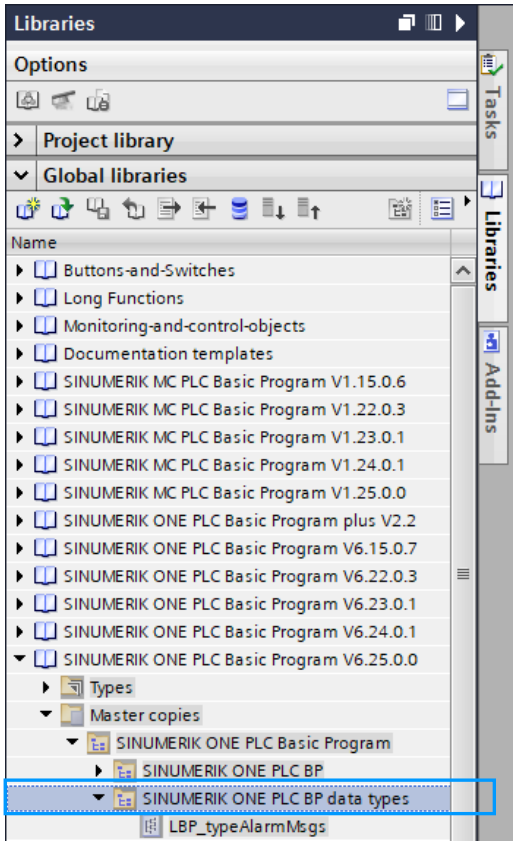
Copy PLC Basic Program blocks from the library to the user program.





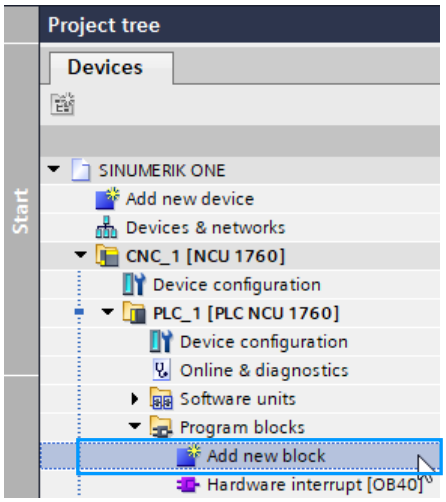
Copy PLC data types from the library to the user program.

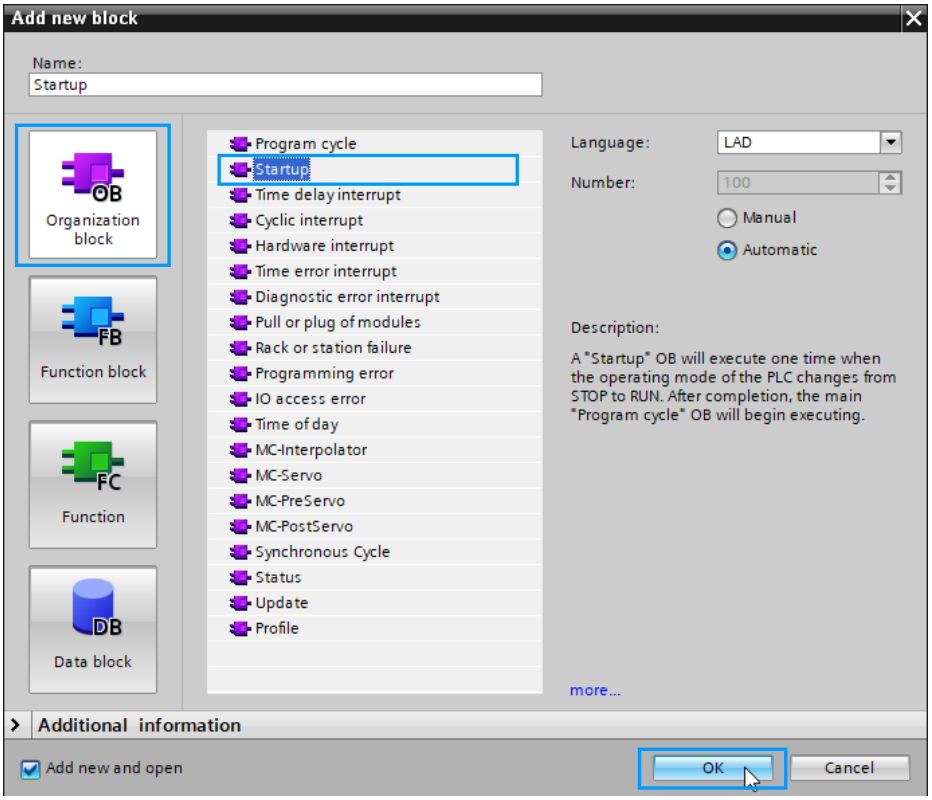




Notes:

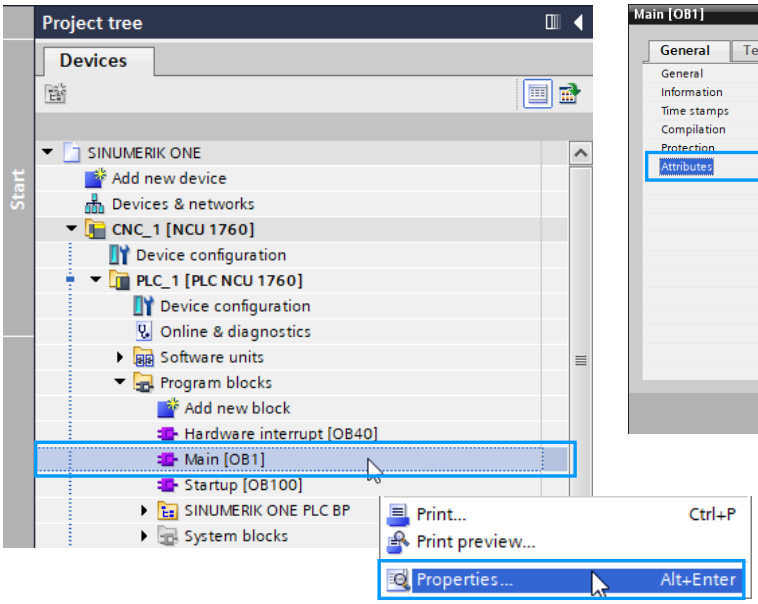
Add new Organization block OB100.

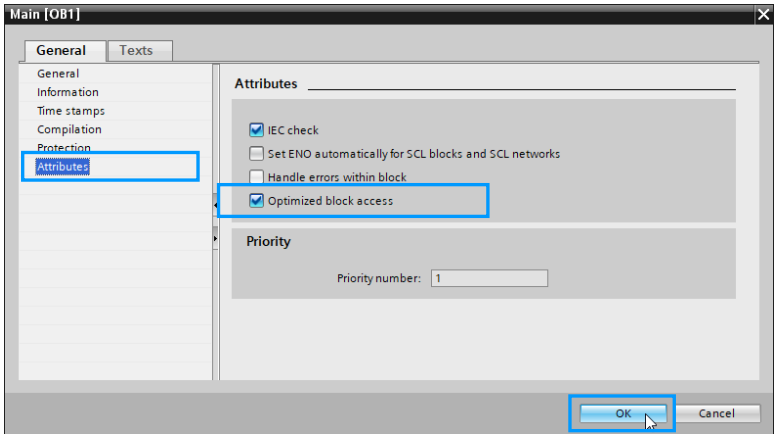




Activate IEC check for OB1, OB40 and OB100

To activate the IEC check, right-click the marked blocks OB1, OB40 and OB100.

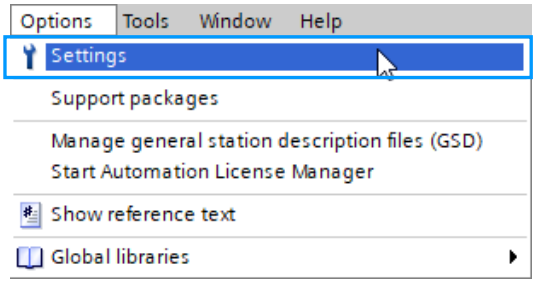




Notes:

Activate IEC check for new blocks

Activate IEC check as default setting for new blocks via the Settings menu.

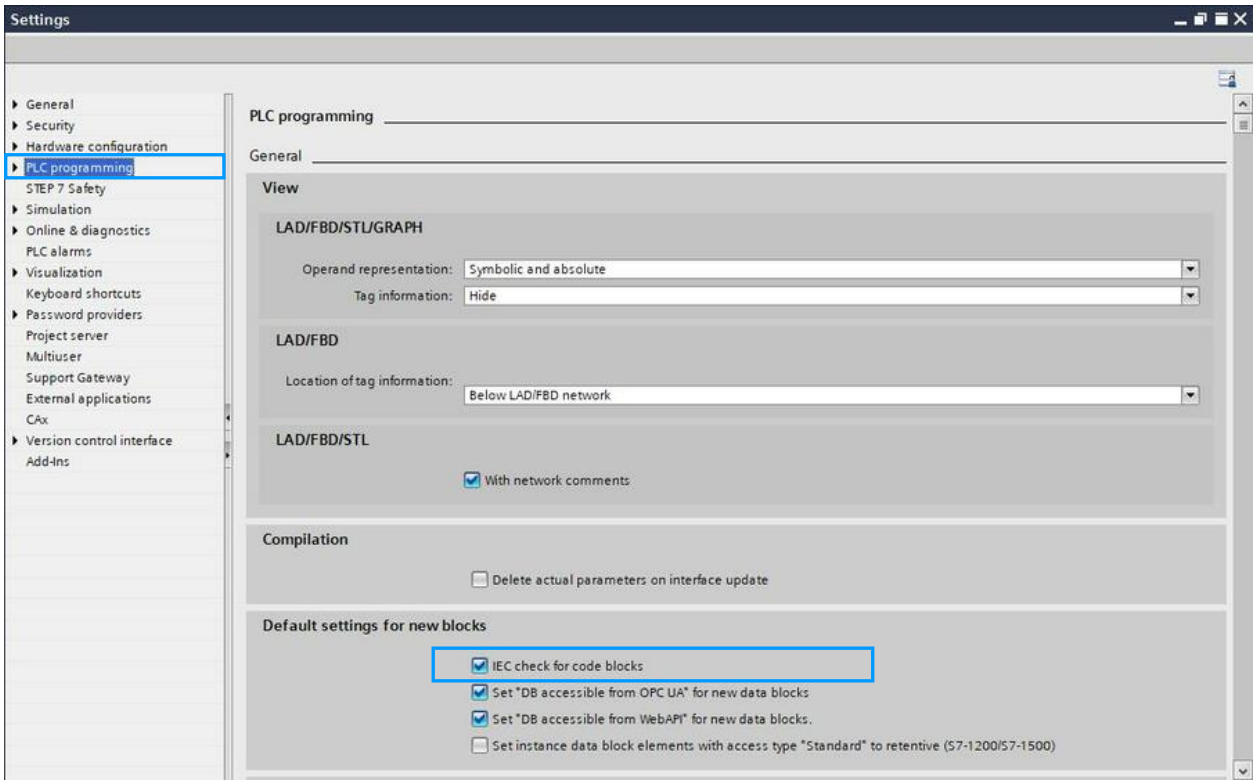


IEC check for code blocks:

The compatibility of the operands in comparison operations and arithmetic operations can be checked in accordance with IEC 61131. Incompatible operands must be converted explicitly.

With IEC check (S7-1500):

- The implicit conversion from BOOL to other data types is not possible.
- Only the BYTE and WORD data types can be converted implicitly.
- The bit length of the source data type must not exceed the bit length of the target data type. For example, an operand of the WORD data type cannot be specified for a parameter when the BYTE data type is expected.

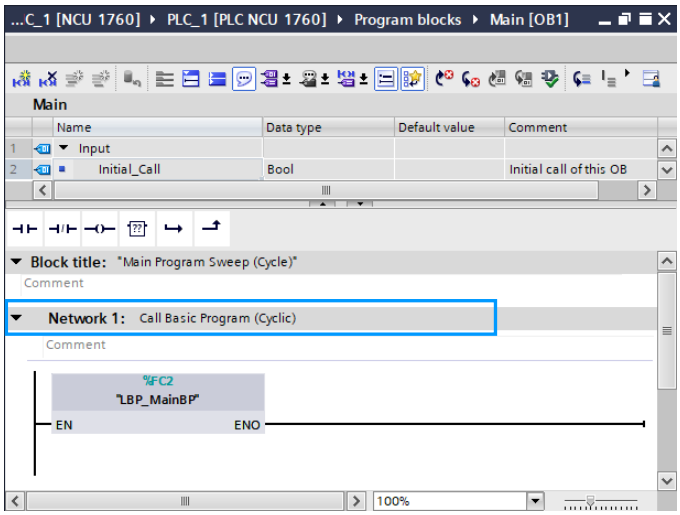


Notes:

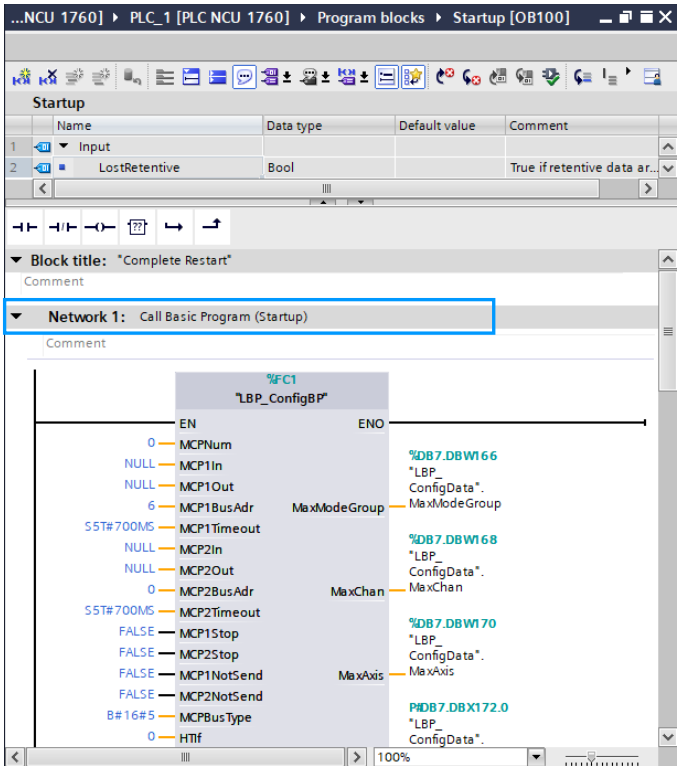
Calling the Basic Program in OB1, OB40 and OB100

The Basic Program is executed by calling FC2 in OB1, by calling FC3 in OB40 and by calling FC1 in OB100.

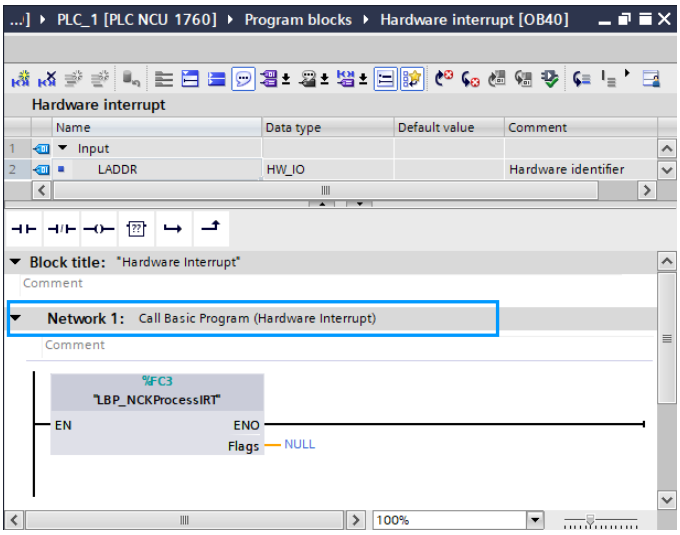
OB1:



OB100:



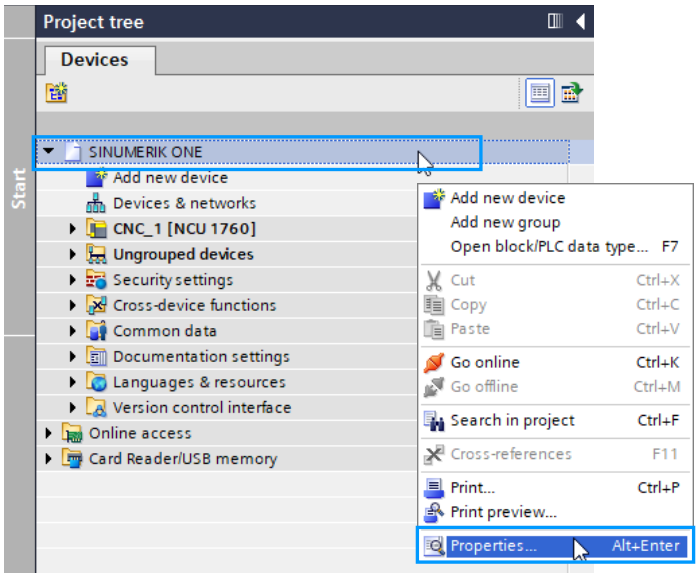
OB40:



Notes:

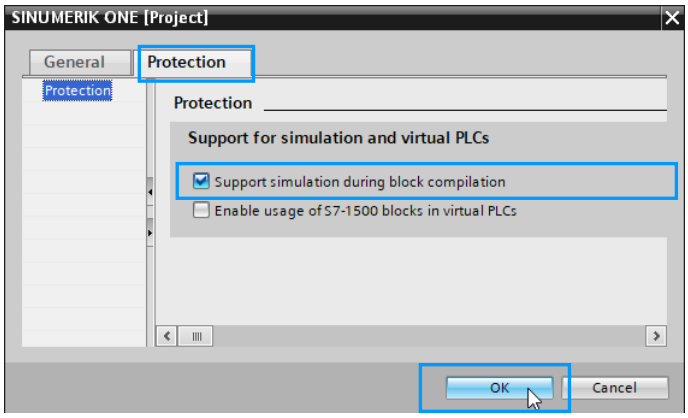
Activate simulation

Select the properties by right-clicking the project.



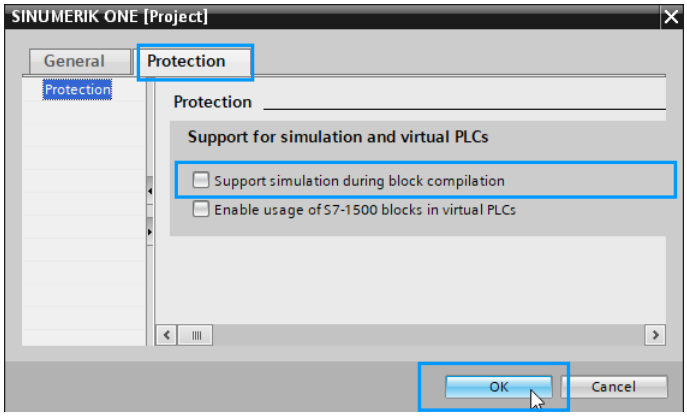
Virtual

Enable simulation with S7-PLCSIM Advanced.



Real

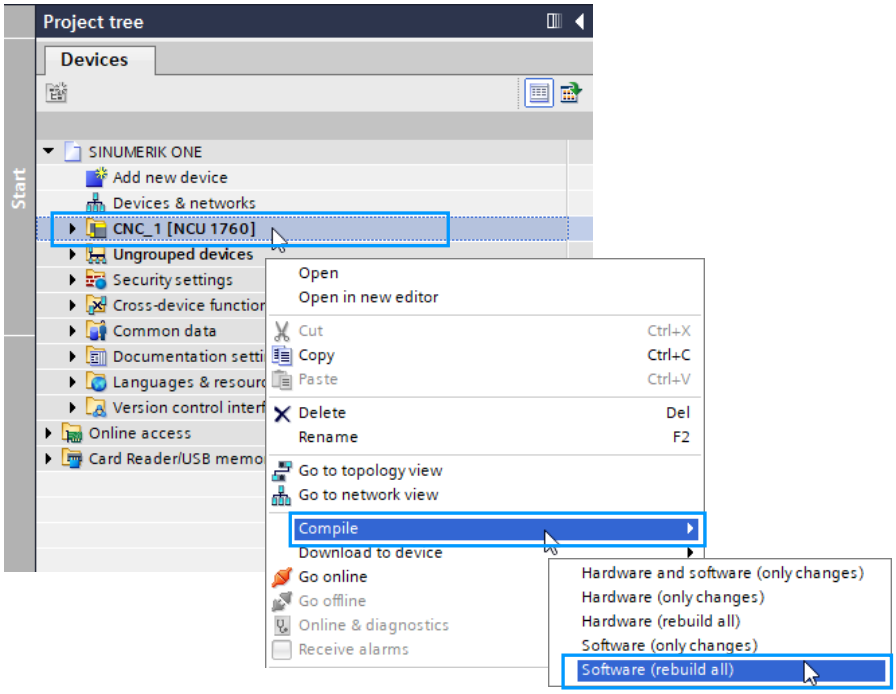
Disable simulation with S7-PLCSIM Advanced.



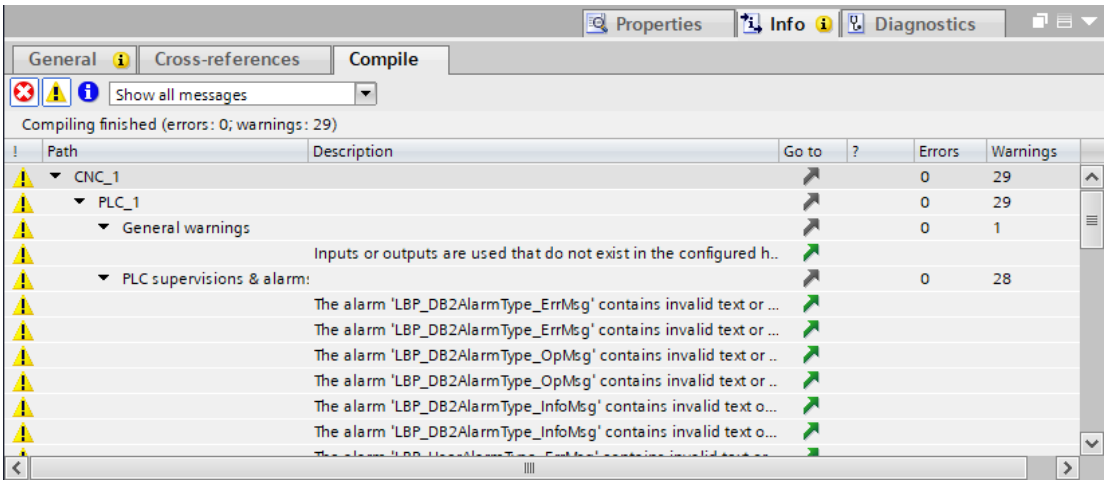
Notes:

Saving and compiling the software

To compile the software, right-click the marked CNC or PLC.



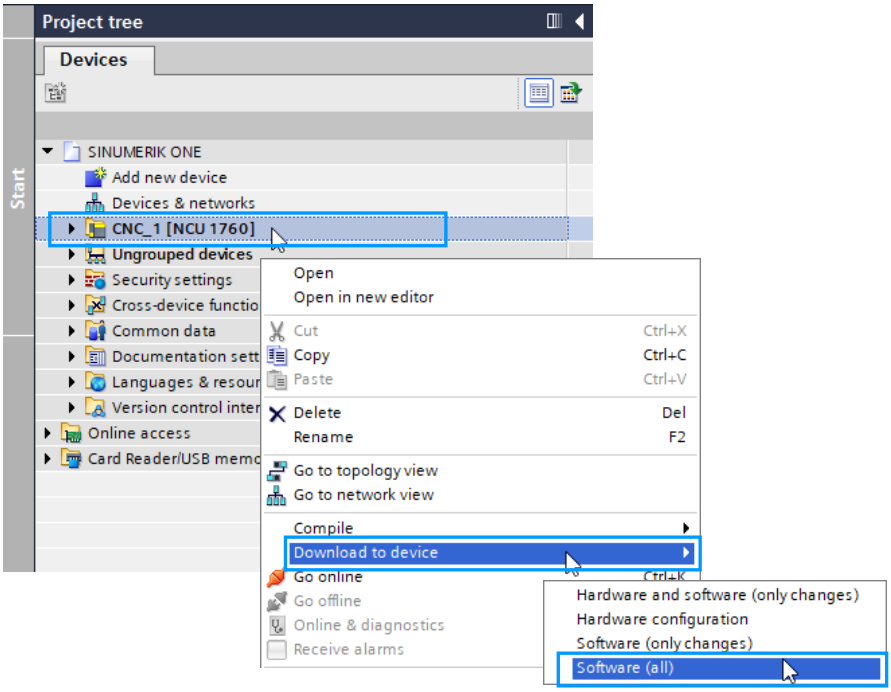
The compilation result is displayed in the Inspector window.



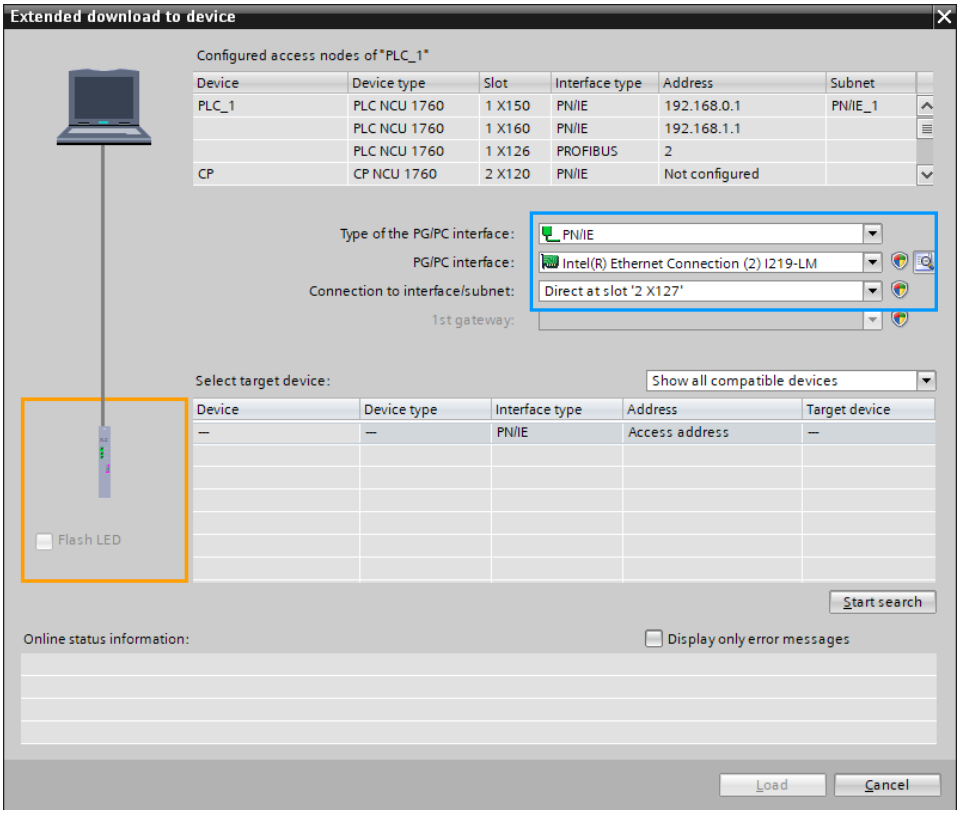
Notes:

Downloading software to the device

To download the software to the device, right-click the marked CNC or PLC.



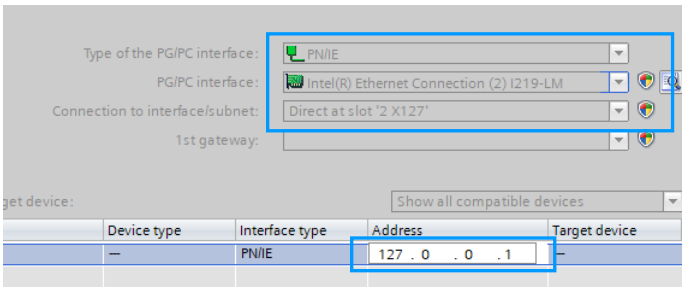
The PG/PC interface is set and the search operation then started. The "Extended download" dialog opens automatically when no connection data exists in the project.



Notes:

The PG/PC interface is set and the search operation then started.

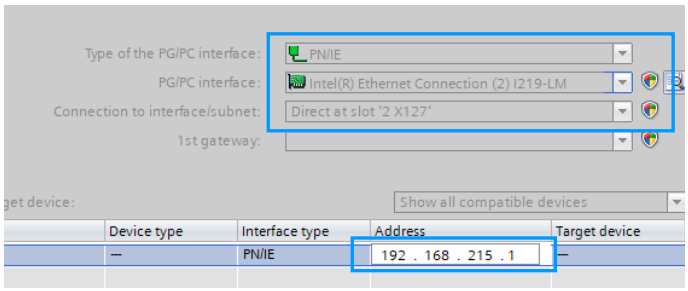
Virtual



Note:

If the Engineering Tools TIA Portal and CMVM are installed on the same PG/PC and no IP address is assigned, 127.0.0.1 can be used for the local host.

Real

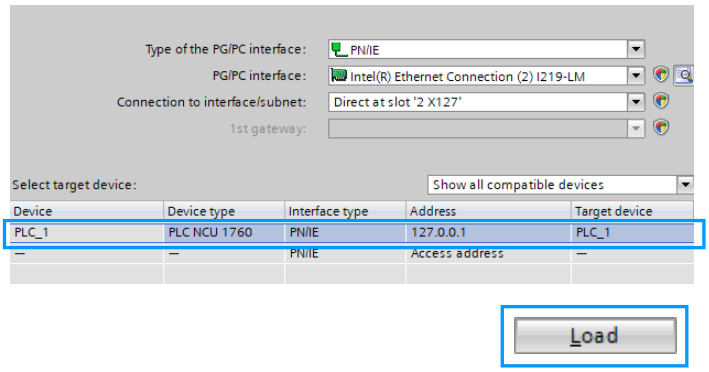


Note:

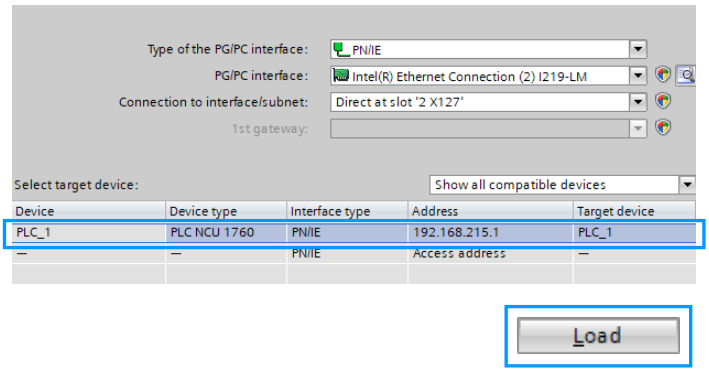
If the PG/PC and NCU are connected via the service interface X127, the IP address 192.168.215.1 can be used.

After the successfully completed scan, the target device is selected and the download started.

Virtual



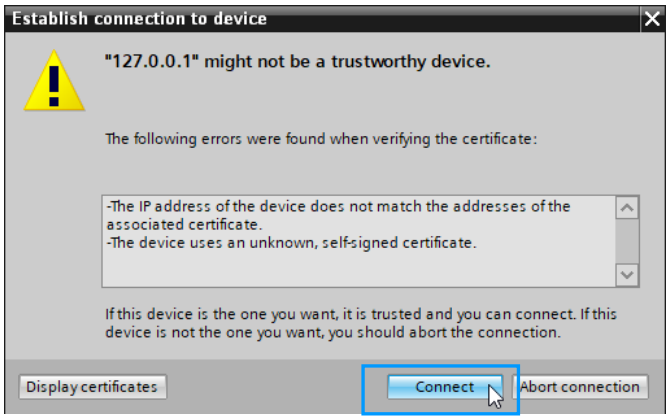
Real



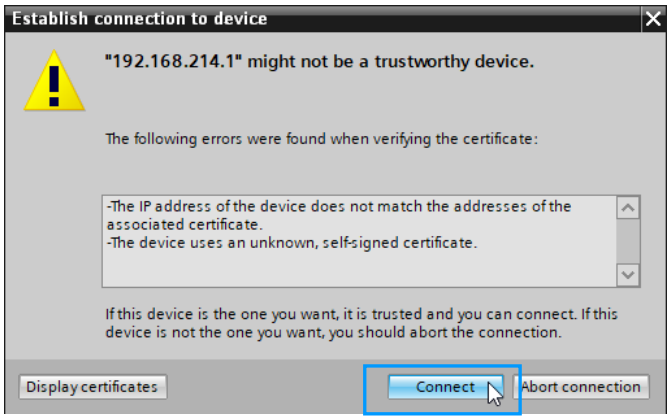
Notes:

If the message below appears select the device as trusted.

Virtual

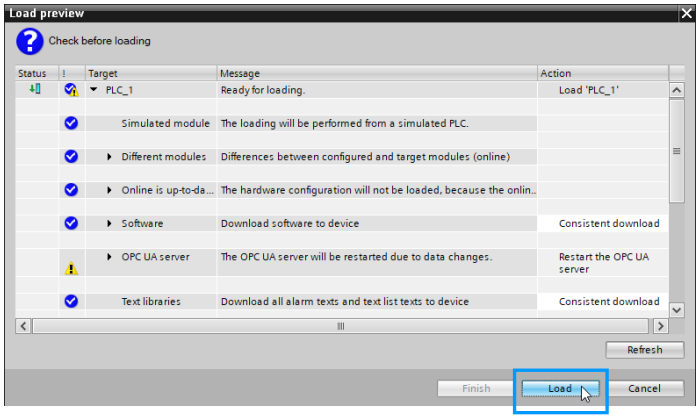


Real

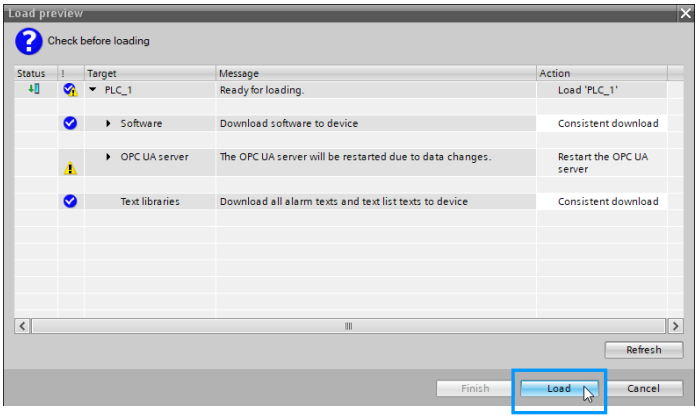


Start the download.

Virtual

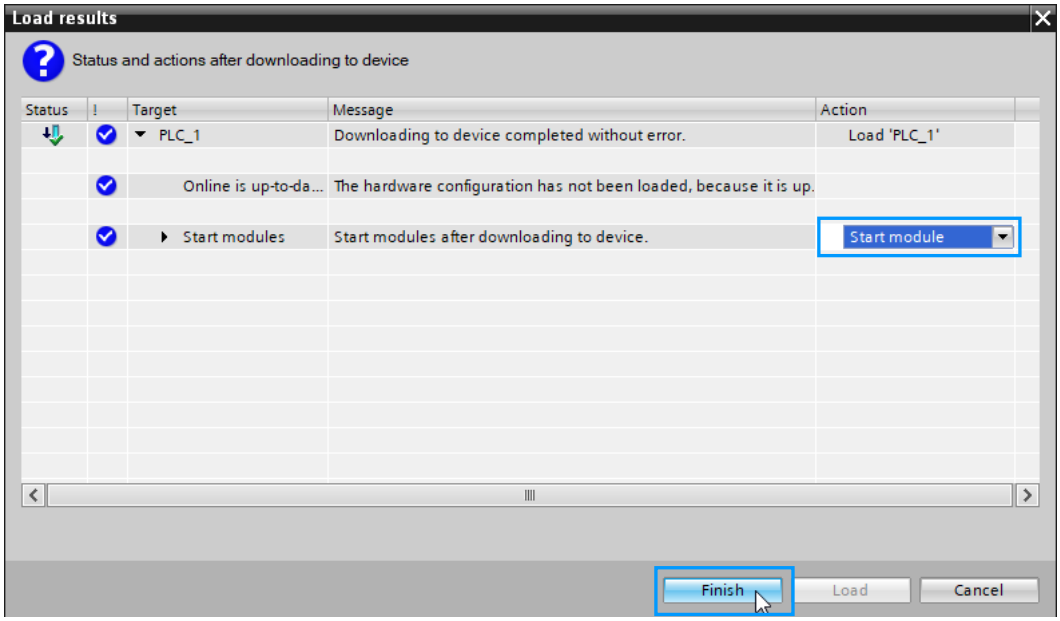


Real

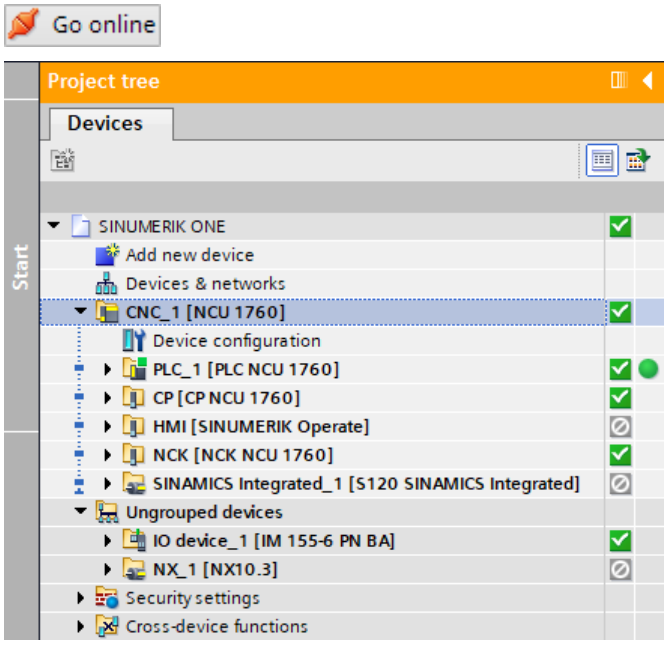


Notes:

Complete the download.



After the successful download, the status can be checked using the online diagnostics.



Notes:

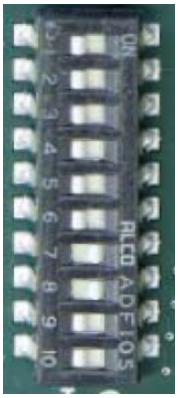
PLC startup parameterization

To complete the example, two parameters must be changed in OB100/FC1 in order to match the associated address and type of the MCP.

MCP1BusAdr :=192
MCP1BusType :=B#16#55

The bus address depends on the DIP switch at the rear of the MCP.
The bus type is 5 that specifies Ethernet.

Switch S1 position	Value when in “ON” state
1	1 = 0
2	2 = 0
3	4 = 0
4	8 = 0
5	16 = 0
6	32 = 0
7	64 = 1
8	128 = 1
9	**PN = OFF**
10	**PN = OFF**

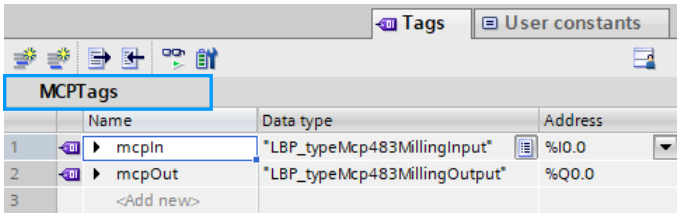


Switch S1

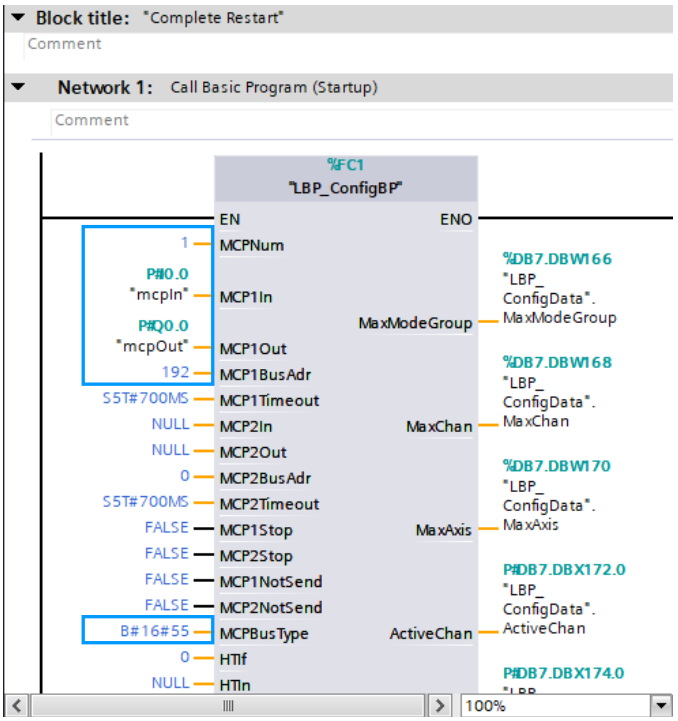
Parameter description of FC1

Signal	Type	For-mat	Value range	Comment
MCPNum	I	INT	Up to 2	Number of active MCP 0: No MCPs available
MCP1In MCP2In	I	POINTER	I0.0-120.0 M0.0-248.0 DBn.DBX0.0-DBXm.0	Start address for input signals of relevant machine control panel
MCP1Out MCP2Out	I	POINTER	Q0.0-120.0 M0.0-248.0 DBn.DBX0.0-DBXm.0	Start address for output signals of relevant machine control panel
MCP1Bus Adr MCP2Bus Adr	I	INT	1 ... 126 192 .. 223	DP slave: PROFI-BUS address Ethernet MCP: DIP-Setting
MCP-BusType	I	BYTE		Righthand half byte (bits 0...3) MCP1 Lefthand half byte (bits 4...7) MCP2 b#16#33: PROFI-BUS b#16#44: PROFI-BUS on the MPI/DP port b#16#55: Ethernet B#16#66: PROFINET

Create a tag table for the MCP signals.



With the following parameters in OB100, the machine control panel via Ethernet (IE) is activated.



Notes:

TCS button (Tool Coordinate System)

To activate the TCS button on the machine control panel, one further parameter must be changed in OB100/FC1.

MCP_IF_TCS :=TRUE

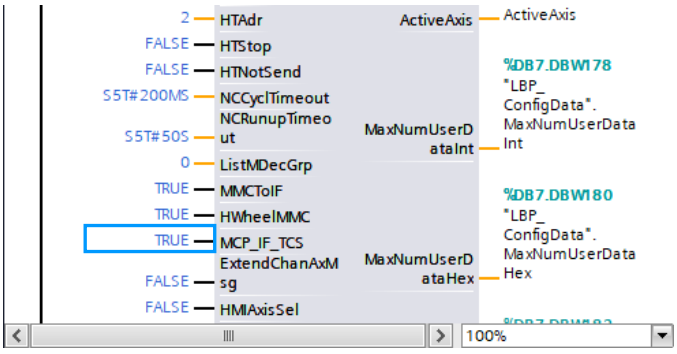
Note:
The parameter can only be used in conjunction with the FC19 that is used for e.g. MCP483, MCP398.



Parameter description of FC1

Signal	Type	Format	Value range	Comment	
MCP_IF_TCS	I	BOOL		FALSE	No evaluation
				TRUE	FC19: Evaluation R11 key acts as manual travel in tool orientation

With the following parameter in OB100, the TCS button on the machine control panel is activated.



Notes:

PLC machine data

In order to assign the PLC machine data to the user DBs, three further parameters must be changed in OB100/FC1.

The PLC machine data are activated via NC-machine data.

MD14504 $MN_MAXNUM_USER_DATA_INT = 512
MD14506 $MN_MAXNUM_USER_DATA_HEX = 512
MD14508 $MN_MAXNUM_USER_DATA_FLOAT = 128

Data type	Number of user data	Value range	User data (NC)	User DB (PLC)
INT	MD14504 $MN_MAXNUM_USER_DATA_INT	0 ...512	MD14510 $MN_USER_DATA_INT [0..511]	"UserData".MD 14510 Array[0..511] of Int
HEX	MD14506 $MN_MAXNUM_USER_DATA_HEX	0 ...512	MD14512 $MN_USER_DATA_HEX [0..511]	"UserData".MD 14512 Array[0..511] of Byte
FLOAT	MD14508 $MN_MAXNUM_USER_DATA_FLOAT	0 ...128	MD14514 $MN_USER_DATA_FLOAT [0..127]	"UserData".MD 14514 Array[0..127] of Real

Parameter description of FC1

Signal	Type	Format	Value range	Comment
MaxNumUserDataInt	Q	INT	0 ...512	Quantity of Integer machine data
MaxNumUserDataHex	Q	INT	0 ...512	Quantity of Hexa-decimal machine data
MaxNumUserDataReal	Q	INT	0 ...128	Quantity of Real machine data
UserDataIntArray	Q	VARIANT		Points to an array of type "INT" or structure in any user DB that holds the Integer machine data
UserDataHexArray	Q	VARIANT		Points to an array of type "BYTE" or structure in any user DB that holds the Hexadecimal machine data
UserDataRealtArray	Q	VARIANT		Points to an array of type "REAL" or structure in any user DB that holds the Real machine data

Create a user DB "UserDataInt [DB220]" that holds the Integer machine data.

UserDataInt

	Name	Data type	Offset	Start value
1	Static			
2	MD14510	Array[0..511] of Int	0.0	

Create a user DB "UserDataHex [DB221]" that holds the Hex machine data.

UserDataHex

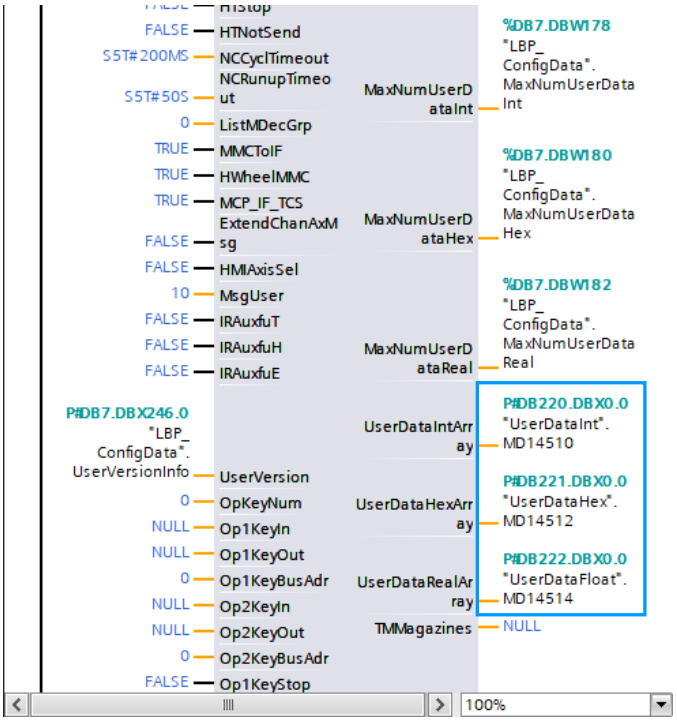
	Name	Data type	Offset	Start value
1	Static			
2	MD14512	Array[0..511] of Byte	0.0	

Create a user DB "UserDataFloat [DB222]" that holds the Real machine data.

UserDataFloat

	Name	Data type	Offset	Start value
1	Static			
2	MD14514	Array[0..127] of Real	0.0	

With the following parameters in OB100, the user DBs are assigned to the PLC machine data.



Notes:

MCP (Machine Control Panel) Transfer

In addition, the signals of the Machine Control Panel (MCP) has to be transferred.

Function Description:

With the following STEP 7 Functions, the Machine Control Panel signals can be transferred from the MCP periphery to the corresponding signal at the PLC --> NCK Interface.
If a standard signal transfer is acceptable then one of the following STEP 7 Functions can be called, dependant upon the MCP type.

FC19 MCP Milling

FC25 MCP Turning

FC24 MCP small

FC26 HT8

The following group signals are transferred:

Mode groups

Axis selections

WCS/MCS switchover commands

Traversing keys

Overrides

Key-operated switch

Prerequisites for the Function:

MCP hardware available

HMI requirement:
None

Drive requirement (SINAMICS):
None

NCK requirement:
None

PLC requirement:
Parameterization of corresponding STEP 7 function.

Steps to achieve running function:

Parameterisation of the corresponding function.

Parameter description of FC19

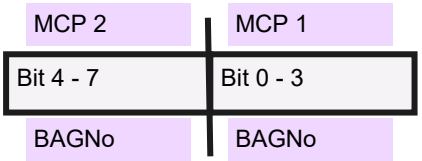
Signal	Type	Format	Range of values	Remark
BAGNo	I	BYTE	0 - b#16#0A and b#16#10 - b#16#1A	No. of mode group to which the mode signals are transferred. BAGNo >= b#16#10 means access to the second machine control panel.
ChanNo	I	BYTE	0 - B#16#0A	Channel no. for the channel signals
SpindleIFNo	I	BYTE	0 - 31 (B#16#1F)	Number of the axis interface declared as a spindle
FeedHold	Q	BOOL		Feed stop from MCP, modal
SpindleHold	Q	BOOL		Spindle stop from MCP, modal

Axis sequence for FC19, FC25, etc

The axis sequence for the FC19 can be found in the interface DB10 beginning at DBB8 for the first axis, a maximum of 9 axes can be configured.
The Machine axis number has to be entered into the Interface byte.
Axis numbers 1 - 9 are entered sequentially as default.

Parameter BAGNo

The BAGNo is a parameter with size BYTE.
This byte can be considered as Left and Right nibble (4 bits).
The left nibble is the BAG No for the 2nd MCP if available, the right nibble being the BAG no for the 1st MCP.



Notes:

MCP communication via Ethernet (IE)

In addition, the operating unit communication has to be established via Industrial Ethernet (IE).

Function Description:

The following STEP 7 Function is used for communication with operating units via Industrial Ethernet (IE).

FB25000 MCP communication via Ethernet (IE)

The block must be called cyclically. The configuration of the connections to the operating units is located in DB7, which is specified in the block call of FB25000. The configuration parameters can be set in the start-up OB (e.g. OB100) via FC1. Further influence or a changed configuration of the connections can be made in the cycle via the parameters in DB7.

The connections to the operating units are automatically established and monitored using the configuration in DB7. Input data received from the operating units is written to the configured pointers by the DB7. The block reads output data sent to the operating units from the configured pointers in DB7.

Prerequisites for the Function:

MCP hardware available

HMI requirement:
None

Drive requirement (SINAMICS):
None

NCK requirement:
None

PLC requirement:
Parameterization of FB25000 in the PLC

Note:

This function is not available in the virtual SINUMERIK.

Steps to achieve running function:

Parameterisation of the corresponding function.

Parameter description of FB25000

Signal	Type	Format	Range of values	Remark
Config	I/O	VARIANT	LBP_ConfigData[DB7]	Configuration DB “LBP_ConfigData [DB7]” of type “LBP_typeConfigData”
Error	Q	BOOL		An error has occurred, see Status
Status	Q	WORD		See Status and Errors

Error codes

State	Meaning
16#0000	No error
16#8001	The variable at the parameter “Config” is not of the type “LBP_typeConfigData”
16#8002	A connection specific error has occurred
Connection specific errors	
16#0000	No error
16#8101	Error when opening or establishing the communication connection
16#8201	Error when sending the identity request to the operating unit
16#8202	Error when receiving the identity response from the operating unit
16#8203	The operating unit has no IPv4 address

Notes:

State	Meaning
16#8301	The parameter data for this operating unit could not be determined
16#8302	The parameter data are invalid
16#8303	Error when sending the parameterization request to the operating unit
16#8304	The parameterization was rejected by the operating unit
16#8305	Timeout when receiving the parameterization response from the operating unit
16#8401	Timeout when receiving data
16#8402	The pointer for the input data of the operating unit is invalid
16#8403	The pointer for the output data of the operating unit is invalid
16#8404	Error when storing input data from the operating unit
16#8405	Error when copying output data to the operating unit
16#8501	Error when disconnecting or closing the connection

Cyclic communication Handwheels

In addition, the handwheel signals has to be transferred.

Function Description:

Die following STEP 7 Function transfers handwheel signals (Impulses signals of the handwheel) from PLC to NCK.

FC25002 Cyclic communication Handwheels

FC25002 uses the instance of the FB25000 block to transfer the handwheel signals and transfer the data as soon as new data is received from an MCP or HT.

For the NCK to process the handwheel signals, the machine data MD11350[i] = 6 must be set for each of the desired handwheels and the assignment of the transfer interface 1 to 6 to NCK to the handwheels has to be made via the machine data MD11352[i].

Prerequisites for the Function:

Handwheels available

MD11350[0] $MN_HANDWHEEL_SEGMENT =6 (IE)
MD11350[1] $MN_HANDWHEEL_SEGMENT =6 (IE)
MD11351[0] $MN_HANDWHEEL_MODULE =1
MD11351[1] $MN_HANDWHEEL_MODULE =1
MD11352[0] $MN_HANDWHEEL_INPUT =1
MD11352[1] $MN_HANDWHEEL_INPUT =2

HMI requirement:
None

Drive requirement (SINAMICS):
None

NCK requirement:
None

PLC requirement:
Parameterization of FC25002 in the PLC

Note:

This function is not available in the virtual SINUMERIK.

Steps to achieve running function:

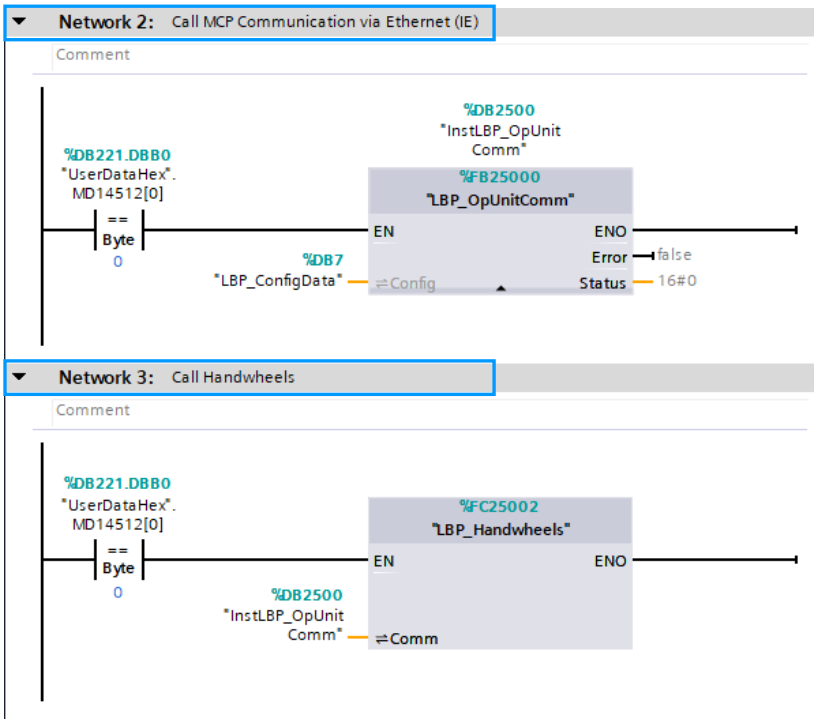
Parameterisation of the corresponding function.

Parameter description of FC25002

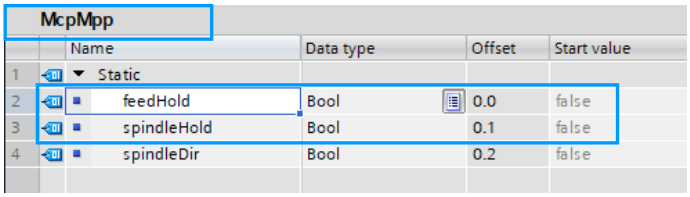
Signal	Type	Format	Range of values	Remark
Comm	I	"LBP_OpUnit Comm"		Instance data of "LBP_OpUnit Comm"

Notes:

With the following logic in OB1, the MCP communication via Industrial Ethernet (IE) is established.



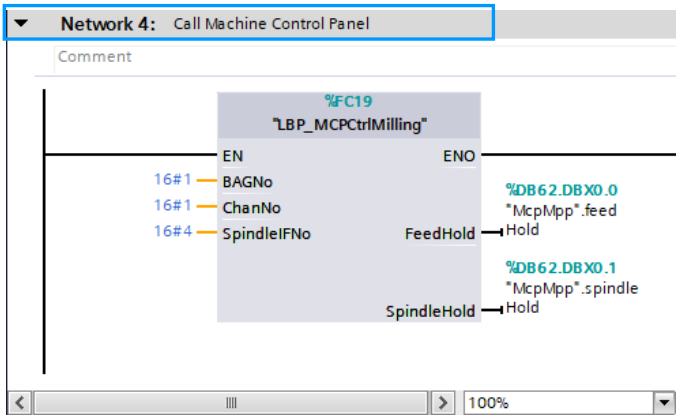
Create a user DB “McpMpp [DB62]” that holds the FC19 output signals.



Note:

The absolute addresses in the user DB are actualized when compiling the software.

With the following logic in OB1, the MCP signals are transferred.



Notes:

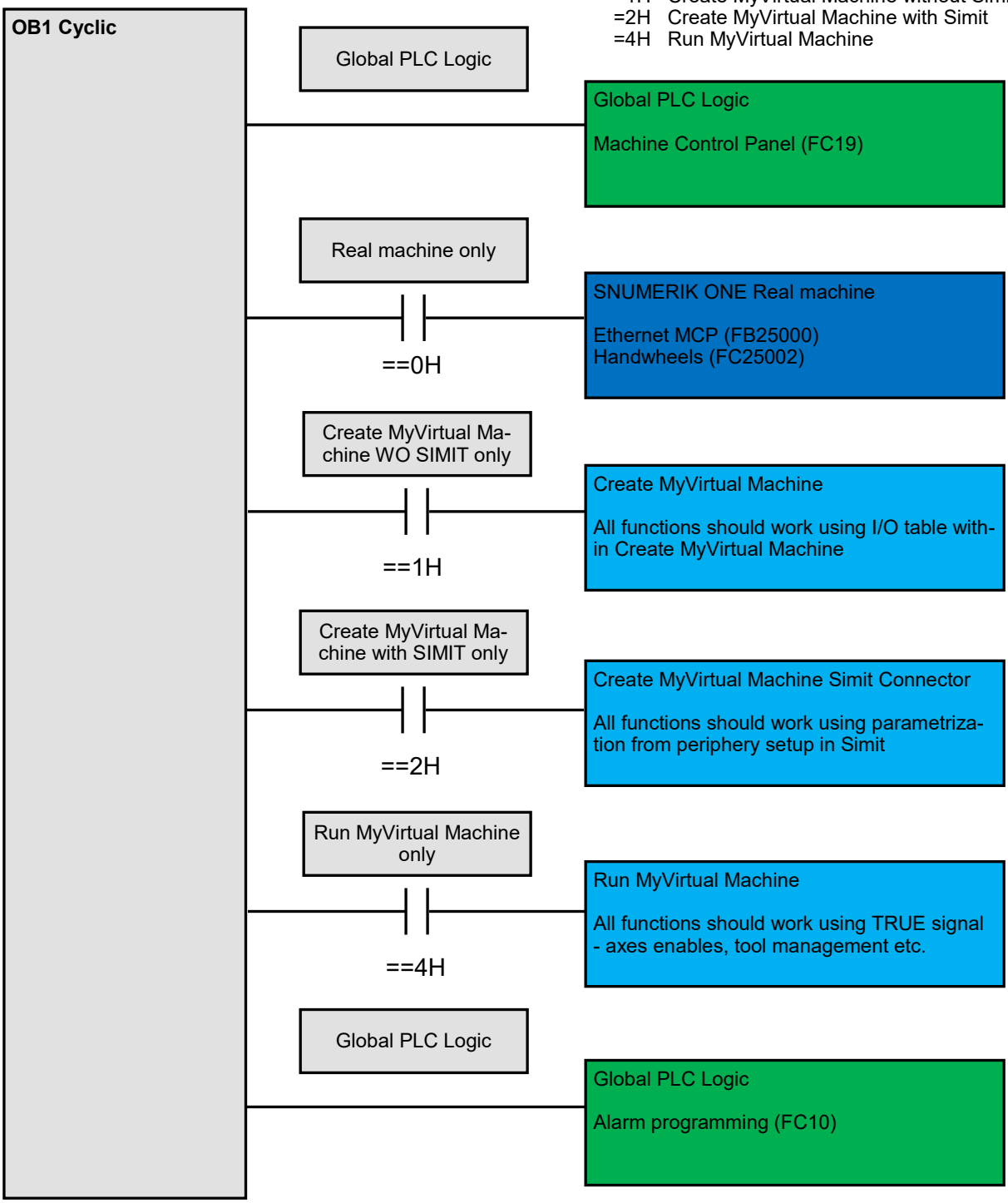
Blocks for OEM signals, enables, etc.

PLC Matrix

At this point of the commissioning procedure the following must be observed:

In the example the PLC machine data are used to build the PLC user program universally, the machine type is defined by a PLC machine data.

MD14512[0] $MN_USER_DAT_HEX
=0H Real Maschine
=1H Create MyVirtual Machine without Simit
=2H Create MyVirtual Machine with Simit
=4H Run MyVirtual Machine



Notes:

MCS/WCS switchover

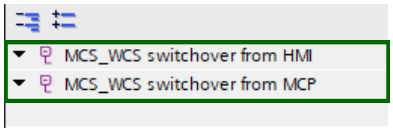
The following interface signals are required.

DB19	Signals to operator panel (PLC → HMI, HMI → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0 PLC → HMI	Actual value in WCS 0=MCS	Save action log	HMI-Adv Shutdown	Delete recall alarms	Delete cancel alarms	Key disable	Darken screen	Brighten screen
DBB20 HMI → PLC	MCS/WCS Changeover	Simulation active	Language 2 switched HMI-Emb.	Recall alarm cleared	Cancel alarm cleared	Cancel key actuated	Screen is dark	

The following MCP signals are required.

Signals from/to Machine Control Panel (M version)								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IBn+5	Axis selection Y R2	Z R3	5th axis R5	MCS/WCS	R11	9th axis R9	8th axis R8	6th axis R6
QBn+3	Axis selection Z R3	5th axis R5	MCS/WCS R12	R11	9th axis R9	8th axis R8	6th axis R6	Direction + R15

Create a user FB “Mode [FB100]” for the MCS/WCS switch-over (SCL).




```
// Example "Mode" [FB100]

// Variables for MCS/WCS switchover

// Static
//    statMcsWcsFlag : Bool;

// Temp
//    tempMcsWcsEdgeP : Bool;

       REGION MCS_WCS switchover from HMI

           // Set Job Start
           IF "LBP_HMI".E_ActWCS THEN
               "LBP_HMI".A_ActWCS := NOT "LBP_HMI".A_ActWCS;
           END_IF;

       END_REGION

       REGION MCS_WCS switchover from MCP

           // Scan rising edge
           #tempMcsWcsEdgeP := "mcpIn".mcsWcs AND NOT #statMcsWcsFlag;
           #statMcsWcsFlag := "mcpIn".mcsWcs;

           // Set Job Start
           IF #tempMcsWcsEdgeP THEN
               "LBP_HMI".A_ActWCS := "mcpOut".mcsWcs;
           END_IF;

       END_REGION
```

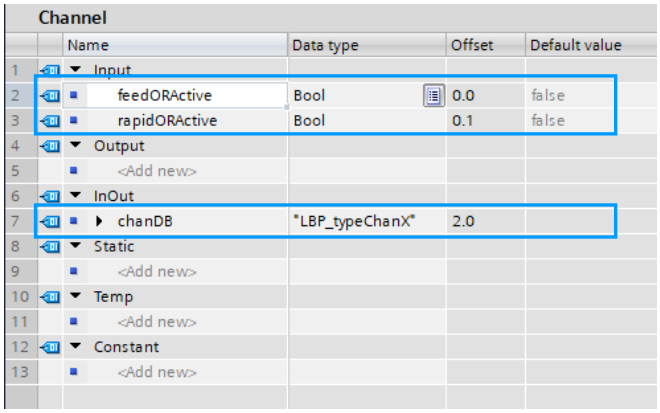
Notes:

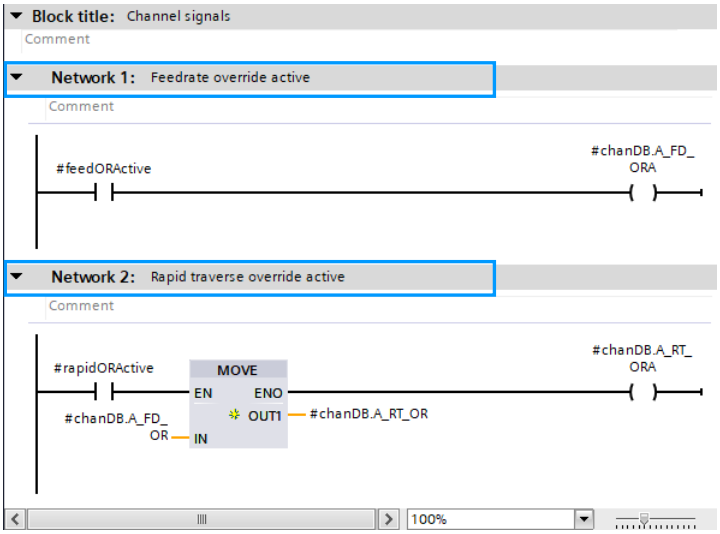
Channel block

The following interface signals are required.

DB21 – DB30	Signals to NC channel (PLC→ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0		Activate dry run feedrate	Activate M01	Activate single block	Activate DRF	RESU: Start retrace support	RESU: Backward / forward	
DBB1	Activate program test	PLC action completed	CLC correction	CLC stop	Time monitoring active (tool management)	Disable all synchronized actions	Enable protection zones	Activate referencing
...								
DBB4	Feedrate override H	G	F	E	D	C	B	A
DBB5	Rapid traverse override H	G	F	E	D	C	B	A
DBB6	Feedrate override active	Rapid traverse override active		Program level abort /Z1-K1/	Delete subroutine no. of passes	Delete distance-to-go	Read-in disable	Feed disable

Create a user FB “Channel [FB121]” that transfers the channel signals.





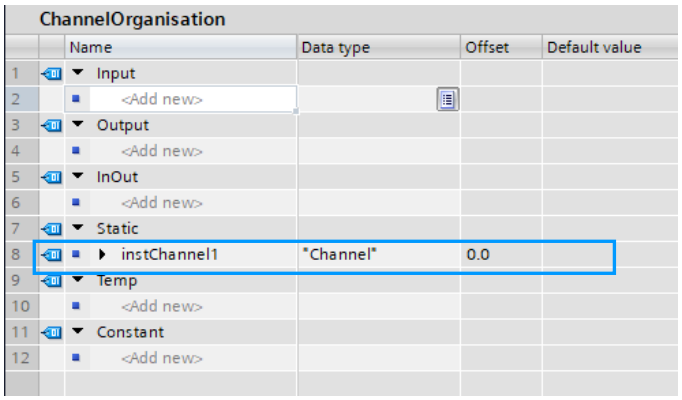
Note:

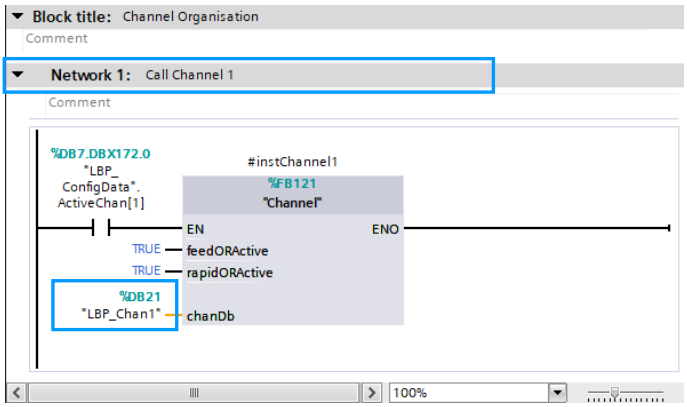
The block should be created “not optimized” because of transfer of ”Classic DB” via parameter.

Notes:

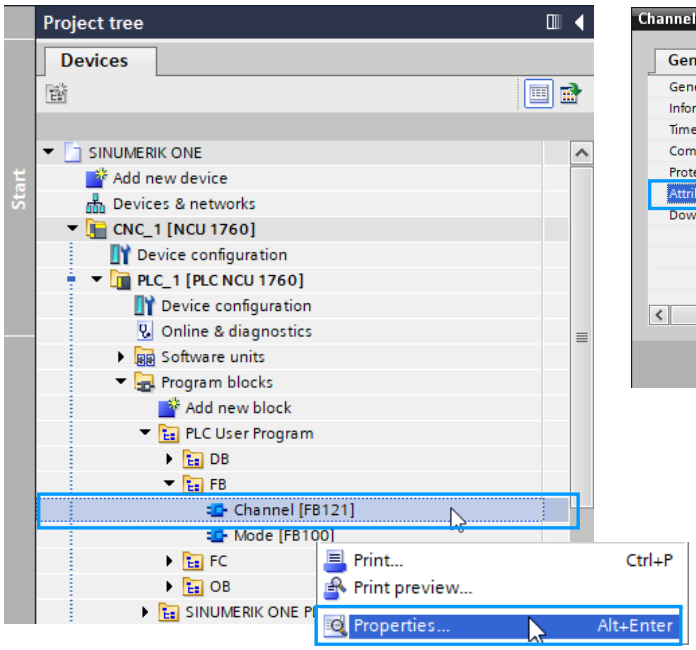
Create a user FB “ChannelOrganisation [FB120]” that calls the individual channels.

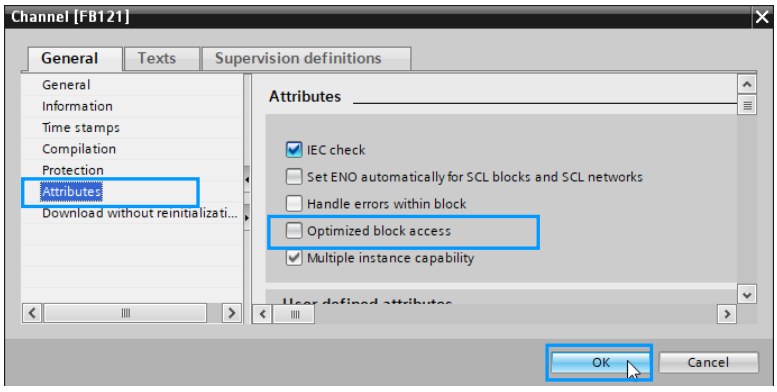
Call channel 1 as a multi instance.





Select non-optimized block access for user FB.





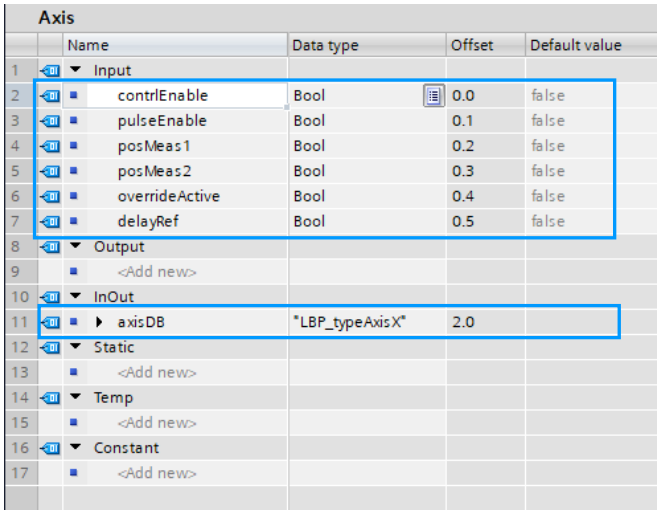
Notes:

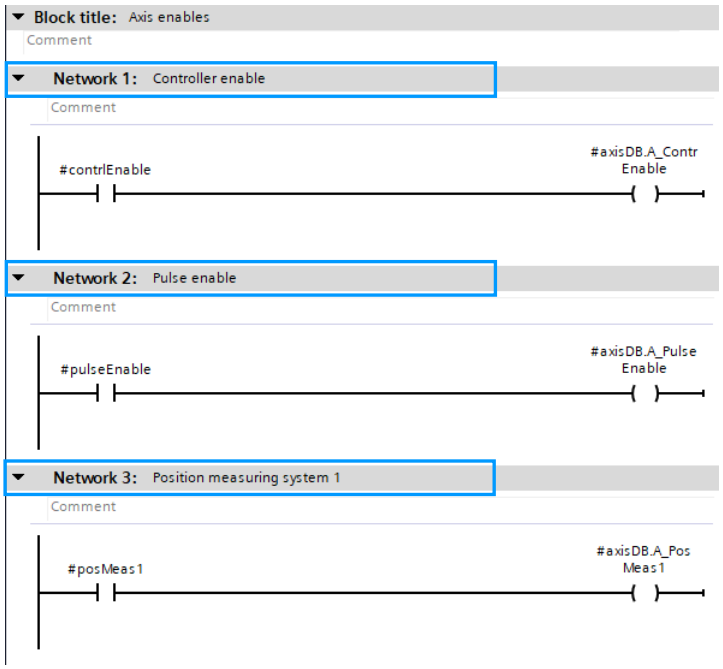
Axis/spindle block

The following interface signals are required.

DB31 – DB61	Signals to axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB1 Axis and spindle	Override active	Position measuring system 2	Position measuring system 1	Follow-up mode	Axis/spindle disable	Sensor fixed stop	Acknowl. fixed stop reached	Acknowl. fixed stop reached
DBB2 Axis and spindle	Reference point value 4	Reference point value 3	Reference point value 2	Reference point value 1	Clamping in progress	Delete dis-tance-to-go/ spindle reset	Controller enable	Cam activa-tion
DBB12 Axis	Delay refer-ence point approach			Modulo rotary axes: Acti-vate travers-ing range limits	2nd software limit switch plus	2nd software limit switch minus	Hardware limit switch plus	Hardware limit switch minus
DBB21 Drive	Pulse enable	Speed con-troller inte-grator disable	Motor selec-tion	Motor selection B	Motor selection A	Drive parameter set selection C	Drive parameter set selection B	Drive parameter set selection A

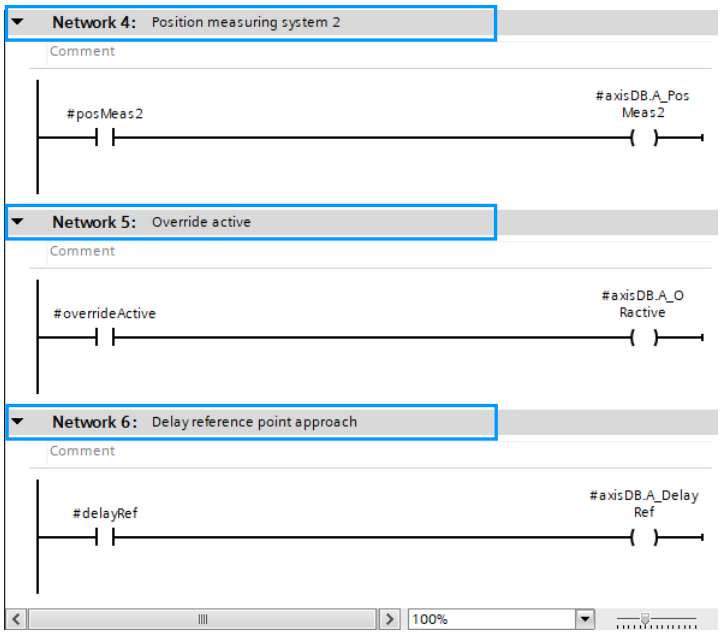
Create a user FB “Axis [FB131]” that transfers the axis/spindle signals.



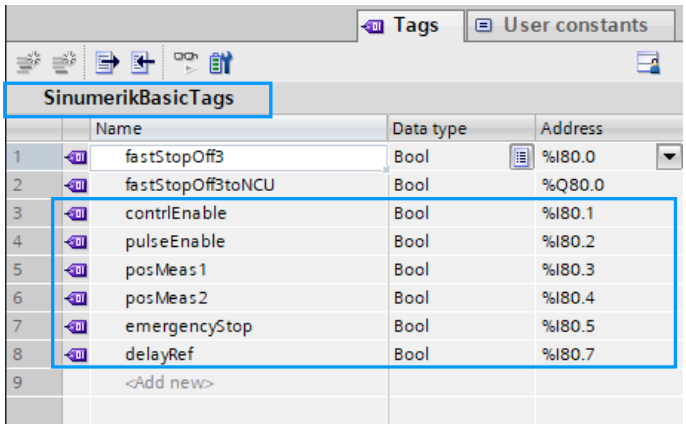


Notes:

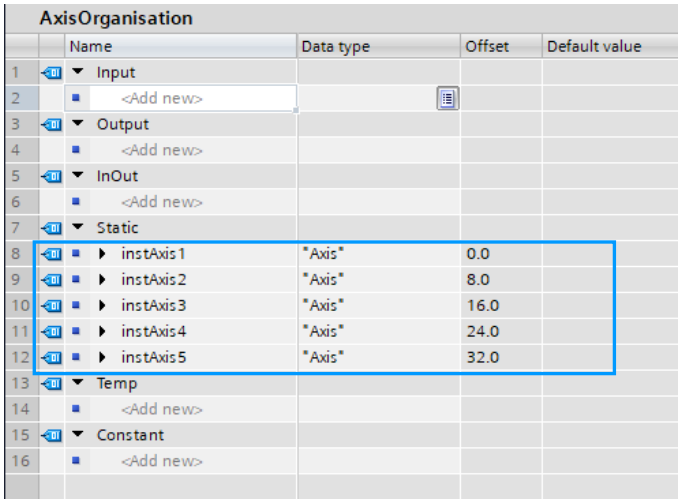
User FB “Axis [FB131]” - continued



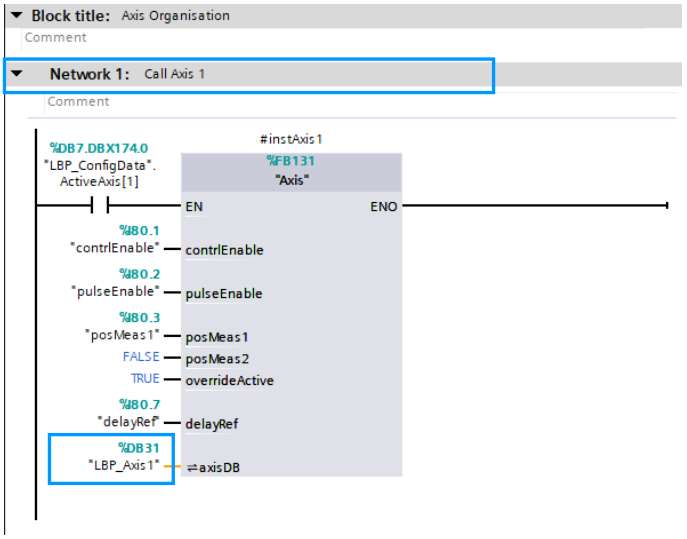
Create a tag table for the basic signals.



Create a user FB “AxisOrganisation [FB130]” that calls the individual axes/spindle.

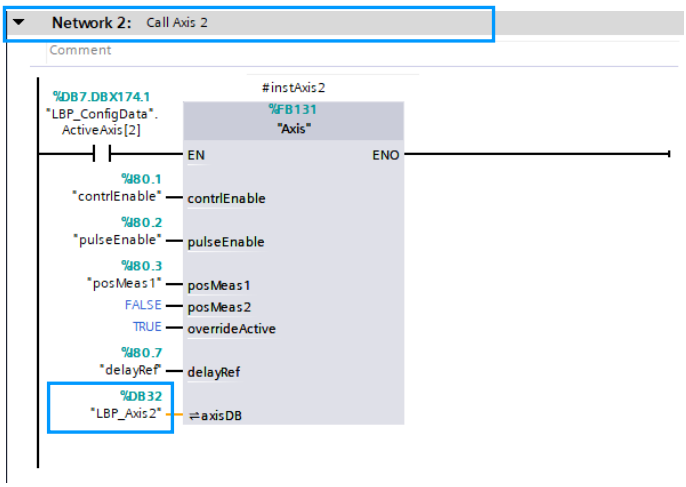


Call axis 1 as a multi instance.

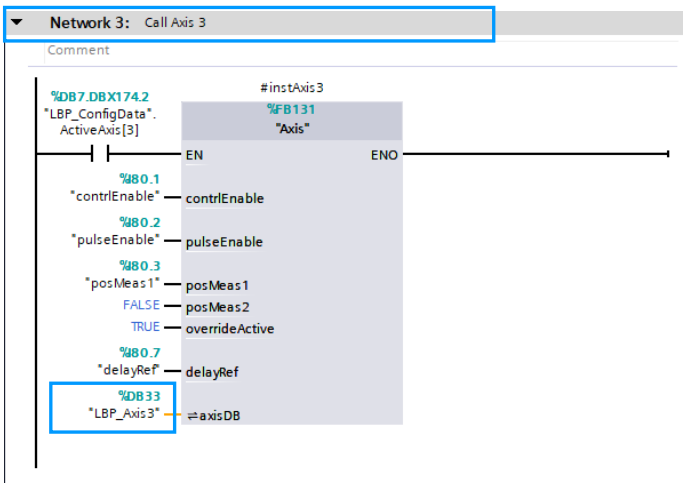


Notes:

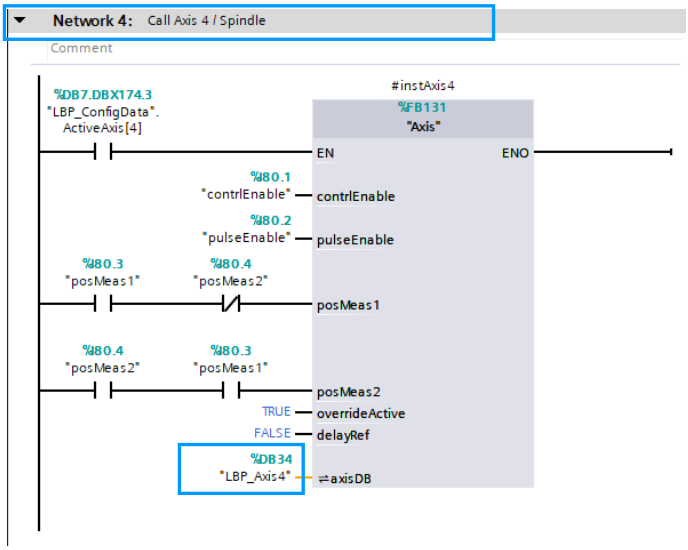
Call axis 2 as a multi instance.



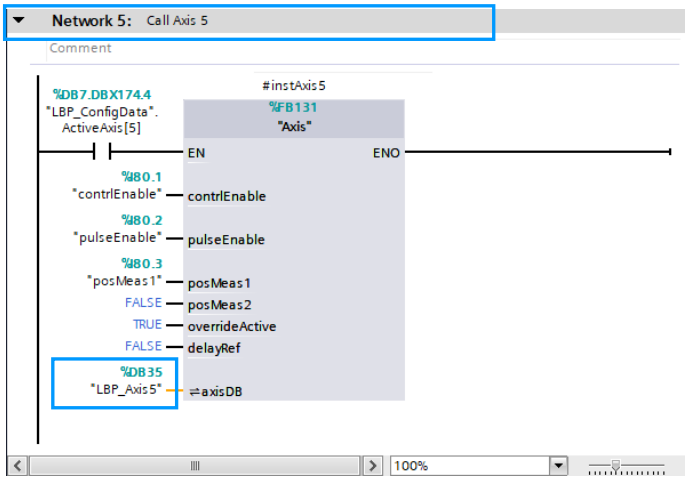
Call axis 3 as a multi instance.



Call axis 4 / spindle as a multi instance.



Call axis 5 as a multi instance.



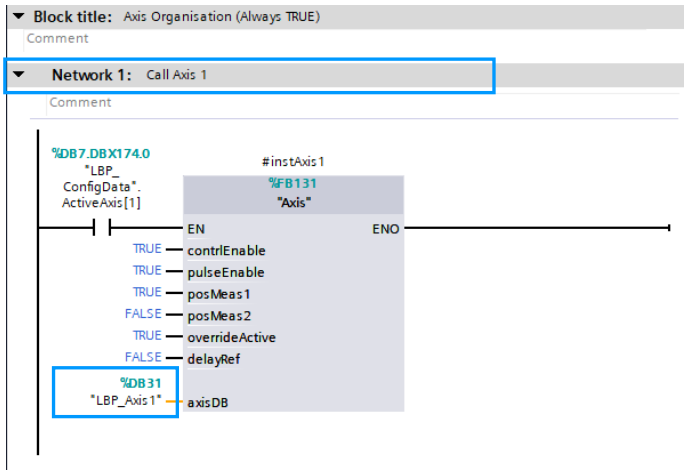
Notes:

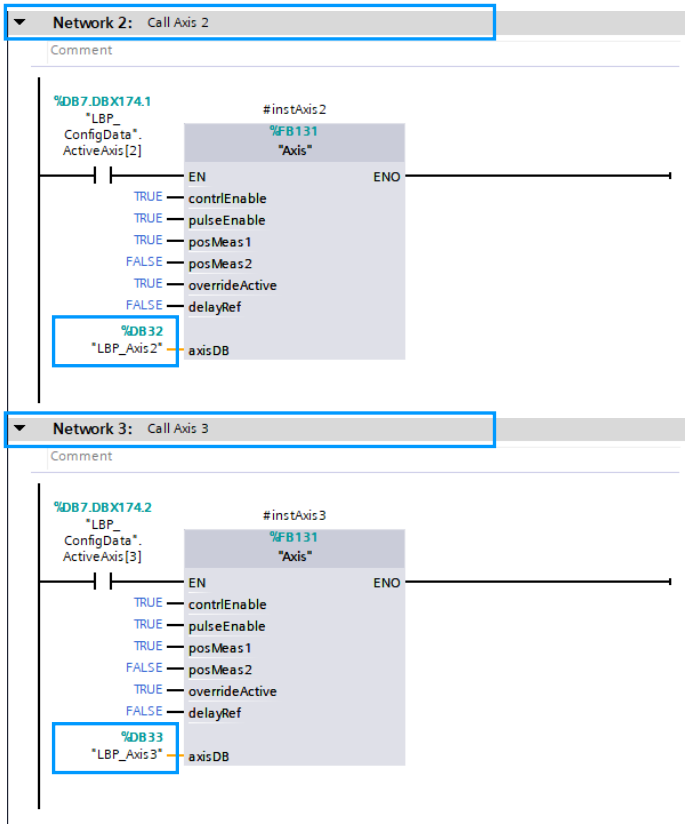
Create a user FB “AxisOrgAlwaysTrue [FB139]” that calls the individual axes/spindles.

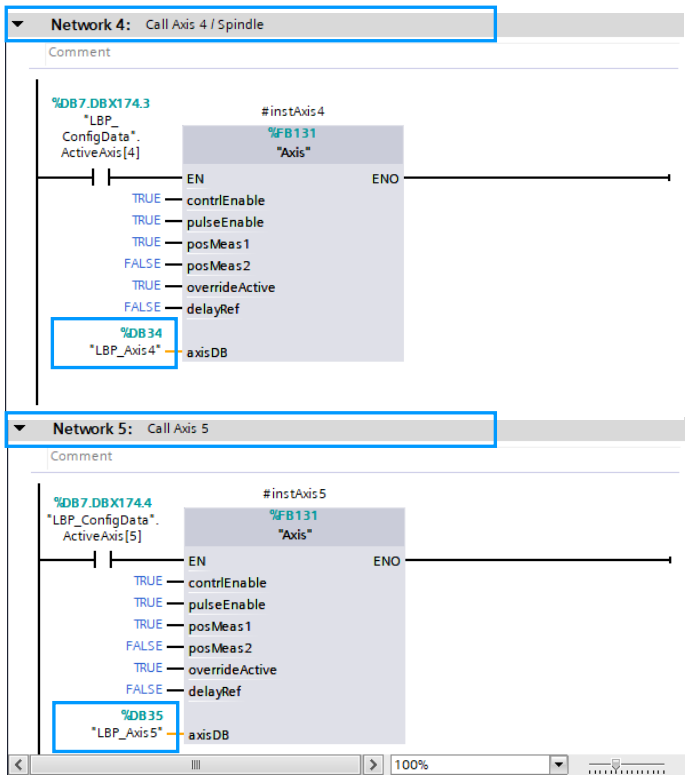
Call axis 1 to 5 as a multi instance.

AxisOrgAlwaysTrue

	Name	Data type	Offset	Default value
1	Input			
2	<Add new>			
3	Output			
4	<Add new>			
5	InOut			
6	<Add new>			
7	Static			
8	instAxis1	"Axis"	0.0	
9	instAxis2	"Axis"	8.0	
10	instAxis3	"Axis"	16.0	
11	instAxis4	"Axis"	24.0	
12	instAxis5	"Axis"	32.0	
13	Temp			
14	<Add new>			
15	Constant			
16	<Add new>			







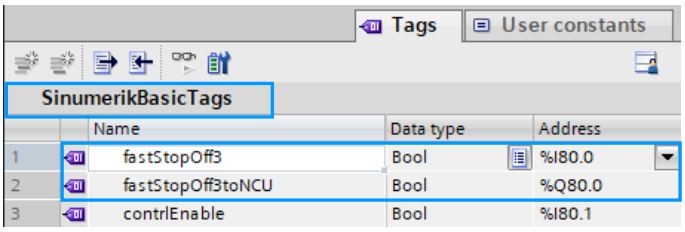
Notes:

Basic signals

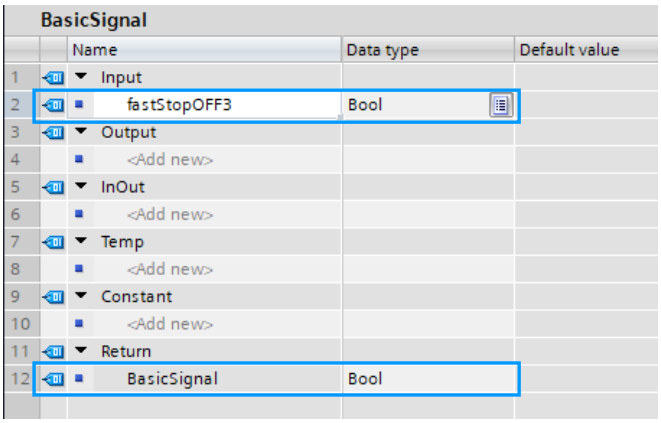
The following basic signals are required.

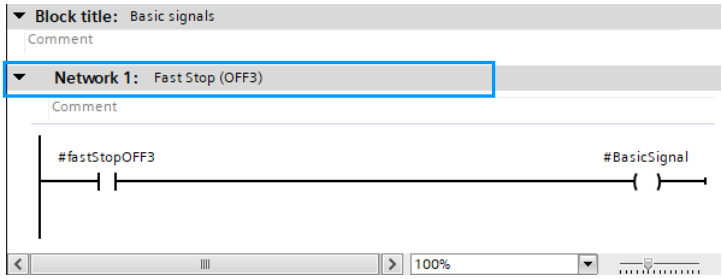
X122	Signal	Function / Default Setting	BICO connection	
			Source	Destination
Pin 1	Sinamics DI 0	ON/OFF1 for DriveCLiQ line modules (ALM, SLM >=16Kw)	CU-r0722.0	Infeed p0840
		Infeed ready for non DriveCLiQ SLM (SLM <16Kw)	SLM X21.1	Drive p0864
Pin 2	Sinamics DI 1	OFF3 drive enable-Fast Stop	CU-r0722.1	Each drive 2nd OFF3, p0849

Tag table with basic signals.



Create a user FC “BasicSignal [FC60]” that transfers the basic signals (OFF3).





Notes:

BAG/NCK block

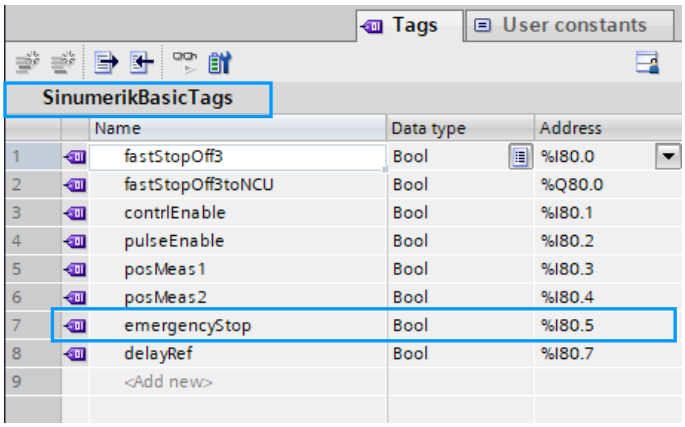
The following interface signals are required.

DB10	Signals to NC (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB56	Keyswitch					Acknowl. Emergency Stop	Emergency Stop	
	Position 3	Position 2	Position 1	Position 0				

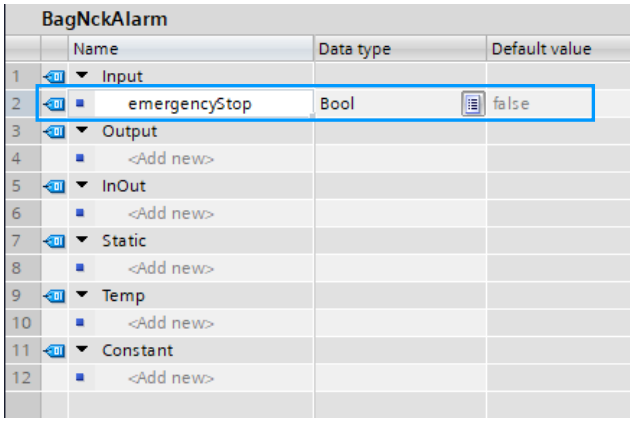
DB11	Signale an BAG1 (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Mode group reset	BAG-Stopp Mode group stop Axes plus spindle	Mode group stop	Mode change disable		Operating mode		
						JOG	MDA	AUTO

DB21 – DB30	Signals to NC channel (PLC→ NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB7	Reset		Suppress Start Lock	NC Stop axes plus spindle	NC Stop	NC Stop to block limit	NC Start	NC Start disable

Tag table with basic signals.

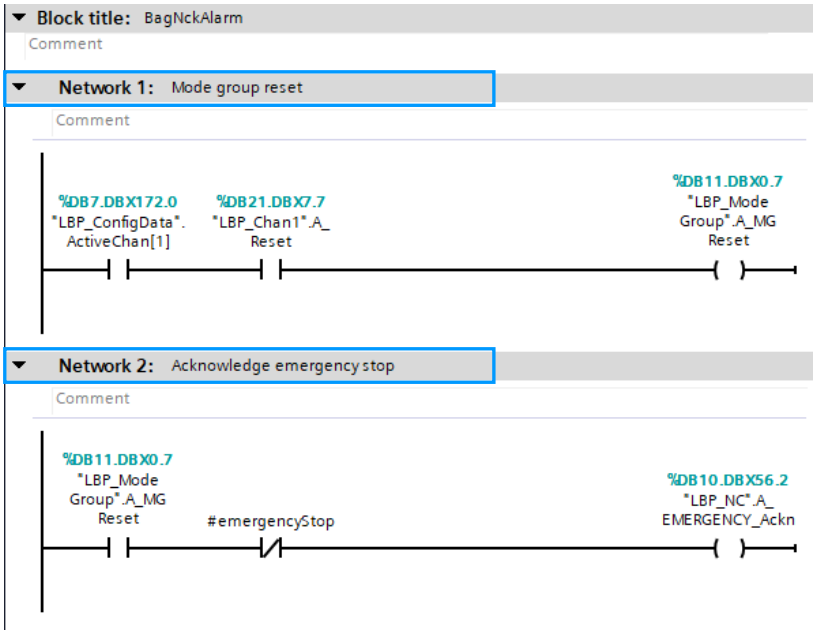


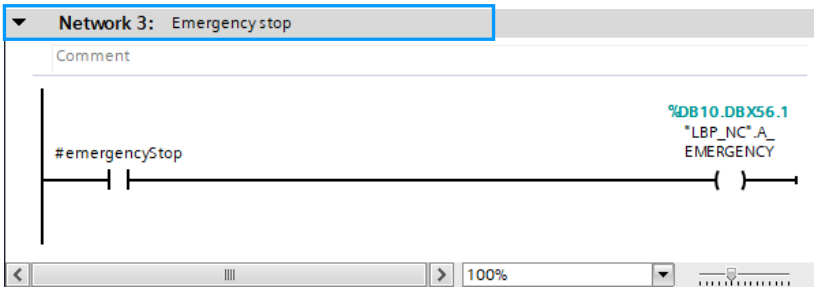
Create a user FB “BagNckAlarm [FB110]” that transfers the BAG/NCK signals.



Notes:

User FB “BagNckAlarm [FB110]” - continued





Notes:

Alarm block

The following interface signals are required.

DB2	Signals for PLC messages (PLC → HMI)							
Byte (Message type))	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Signals WITH display of a fault/operating message (from DBB310)							
	Channel 1							
	Feed disable (alarm no.: 510000-510015)							
310 (FM)	510007	510006	510005	510004	510003	510002	510001	510000
311 (BM)	510015	510014	510013	510012	510011	510010	510009	510008
	Axis/spindle							
	Feed stop/spindle stop for axis/spindle 1 (alarm no.: 600400-600415)							
496 (FM)	600407	600406	600405	600404	600403	600402	600401	600400
497 (BM)	600415	600414	600413	600412	600411	600410	600409	600408
	User areas							
	User area 0: byte 0 (alarm no.: 700000-700015)							
554 (FM)	700007	700006	700005	700004	700003	700002	700001	700000
	User area 0: byte 5 (alarm no.: 700032-700039)							
558 (BM)	700039	700038	700037	700036	700035	700034	700033	700032

The following acknowledgment procedures are implemented for alarm and operator messages.

Operator messages:

Operator messages are intended for the display of normal operating states as information for the operator. Acknowledgment signals are, therefore, not required for this type of message.

Alarm messages:

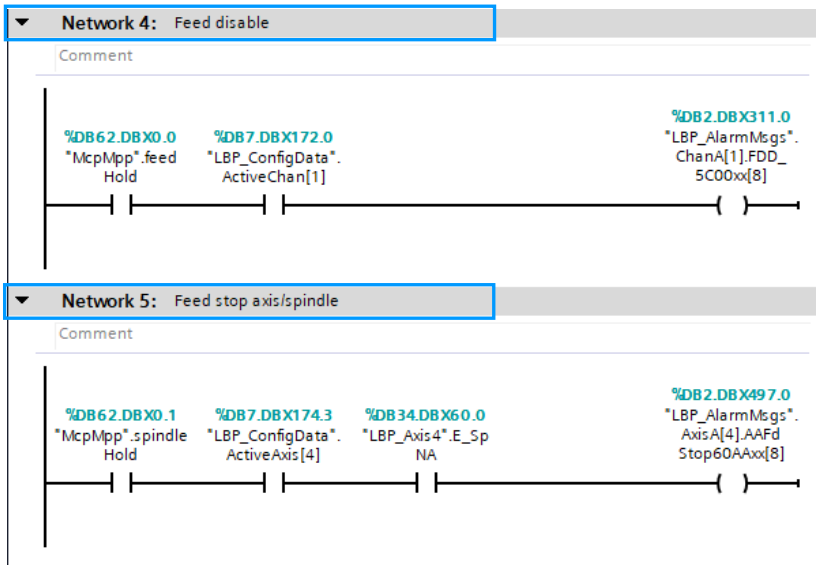
Alarm messages display error states on the machine, which generally lead to a machine stoppage. If the cause of the error disappears, the associated alarm message is only deleted if the user has acknowledged it (e.g. by pressing a key on the MCP).

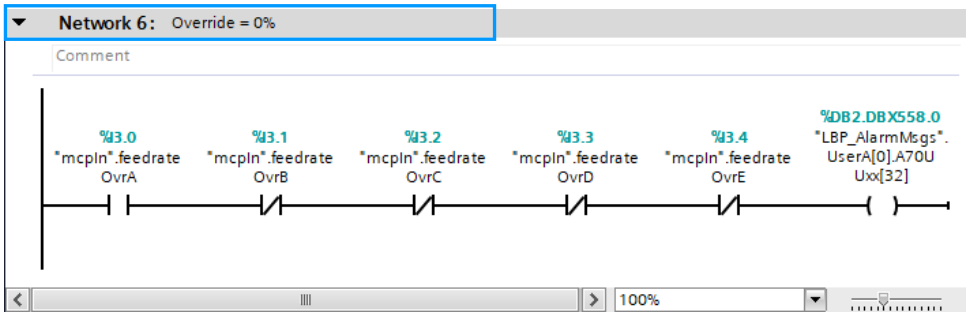
Signal groups of DB2

Area	Data type	Offset	Comment
Chan	Array[1..10] of Struct	0.0	Bit arrays for channel signals without message generation - Feedrate disable - Read-in disable - NC Start disable
Axis	Array[1..31] of Struct	120.0	Bit arrays for axis signals without message generation - Feedrate disable
AddValUserA0	Array[0..63] of INT	182.0	16-bit integer values for alarms/messages in user area 0 (UserA[0] of the messages 700000 to 700063)
ChanA	Array[1..10] of Struct	310.0	Bit arrays for channel signals with message generation - Feedrate disable - Read-in disable - NC Start disable
AxisA	Array[1..31] of Struct	490.0	Bit arrays for axis signals with message generation - Feedrate disable
UserA	Array[0..63] of Struct	554.0	Bit arrays for alarms/ messages in the user area

Notes:

Create a user FB “BagNckAlarm [FB110]” that activates the PLC alarms.





The following alarms are used in the example.

Alarm no	Alarm text
510008	Feed Hold from MCP
600408	Spindle Hold from MCP
700032	Feed Override = 0%

Note:

The creation of the alarm texts is described in module A123 “Alarms and Messages“.

Notes:

Alarms and messages

In addition, the alarms and messages has to be transferred.

Function Description:

Die following STEP 7 Function transfers alarms and messages to the user interface.

FC10 Alarm and operator messages

Prerequisites for the Function:

HMI requirement:
None

Drive requirement (SINAMICS):
None

NCK requirement:
None

PLC requirement:
Parameterization of FC10 in the PLC

Steps to achieve running function:

Parameterisation of the corresponding function.

Parameter description of FC10

Signal	Type	Format	Range of values	Remark
ToUserIF	I	BOOL		Signals transferred to user interface in each cycle
Quit	I	BOOL		Acknowledgement of alarm messages
NumAct-AlarmMsgs	Q	UINT	0...65535	Number of active alarms/messages
Error	Q	BOOL		An error has occurred during the processing.
StatusId	Q	UINT	0...65535	Status identifier
Status	Q	WORD	16#0...16#FFFF	Status information

Error codes

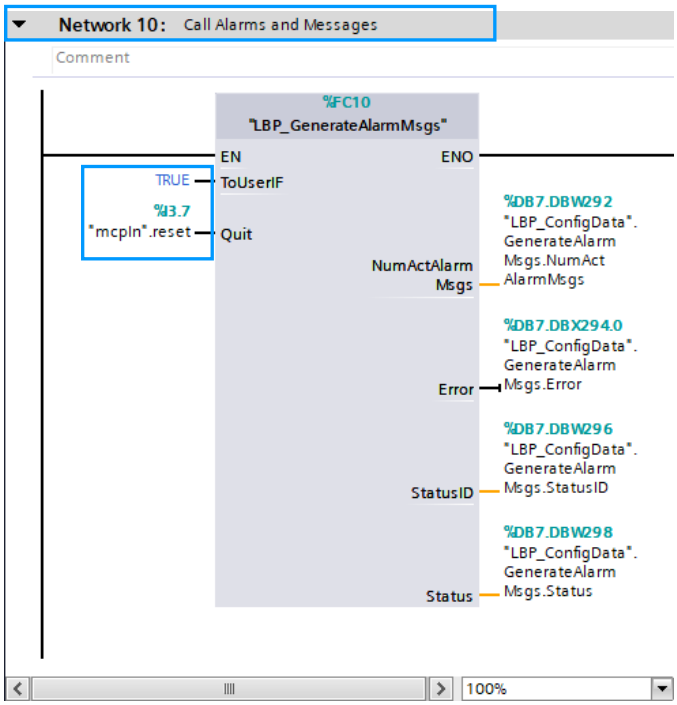
Error	StatusId	State	Meaning
FALSE	1	16#0	No error
FALSE	2	16#7003	The max. number of messages per program cycle have been sent
FALSE	3	16#0085	Message of type “Information only”
FALSE	3	16#70C1	The CPU is currently not generating any messages, because initialization routines are running
FALSE	3	16#70C2	The max. number of messages per time unit have been sent
FALSE	3	16#70C3	All dynamic message instances have been used
FALSE	3	16#70C4	Message is being output and cannot be overwritten
FALSE	4	16#70C1	FC10 is currently not generating any messages, because initialization routines are running
TRUE	1	16#8232	DB is not available: “LBP_Alarms [DB2]“
TRUE	1	16#8233	DB is not available: “LBP_InternalDB3 [DB3]“
TRUE	1	16#8235	DB is not available: “LBP_ParamAlarmMsg [DB5]“
TRUE	1	16#8262	DB access not possible because of attribute “optimized“: “LBP_Alarms [DB2]“
TRUE	1	16#8263	DB access not possible because of attribute “optimized“: “LBP_InternalDB3 [DB3]“
TRUE	1	16#8265	DB access not possible because of attribute “optimized“: “LBP_ParamAlarmMsg [DB5]“

Notes:

Error codes

Error	StatusId	State	Meaning
TRUE	1	16#8242	DB too small: “LBP_Alarms [DB2]“
TRUE	1	16#8243	DB too small: “LBP_InternalDB3 [DB3]“
TRUE	1	16#8245	DB too small: “LBP_ParamAlarmMsg [DB5]“
TRUE	1	16#8252	DB access not possible because of write protection: “LBP_Alarms [DB2]“
TRUE	1	16#8253	DB access not possible because of write protection: “LBP_InternalDB3 [DB3]“
TRUE	1	16#8255	DB access not possible because of write protection: “LBP_ParamAlarmMsg [DB5]“
TRUE	2	16#8267	The value of the parameter “MsgUser“ in DB7 is outside the valid range 0 … 64
TRUE	2	16#8277	The value of the parameter “MaxChan“ in DB7 is outside the valid range 0 … 10
TRUE	2	16#8287	The value of the parameter “MaxAxis“ in DB7 is outside the valid range 0 … 31
TRUE	2	16#8290	“ToUserIF“ = TRUE is not supported on the Hardware used
TRUE	2	16#87FF	System error
TRUE	3	16#87FF	Invalid static message information

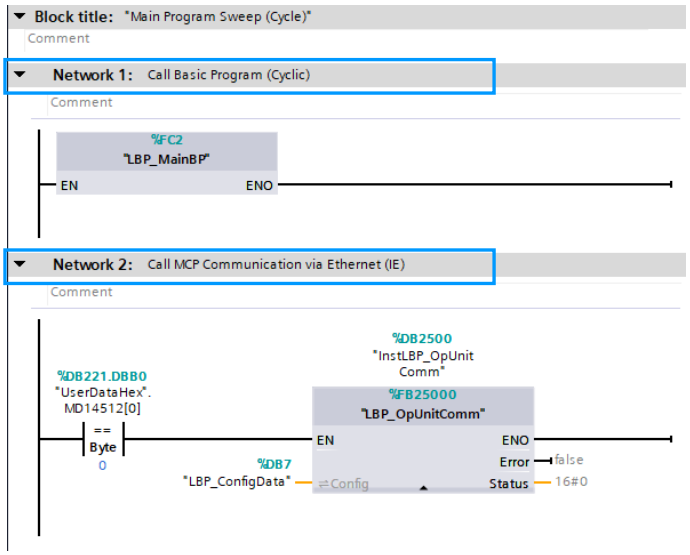
With the following logic in OB1, the alarms and messages are transferred.

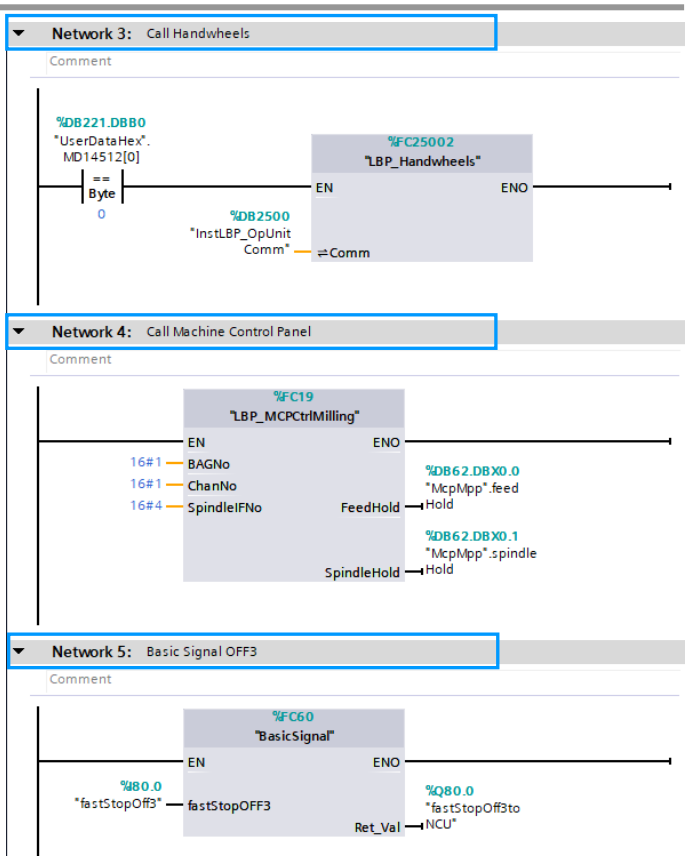


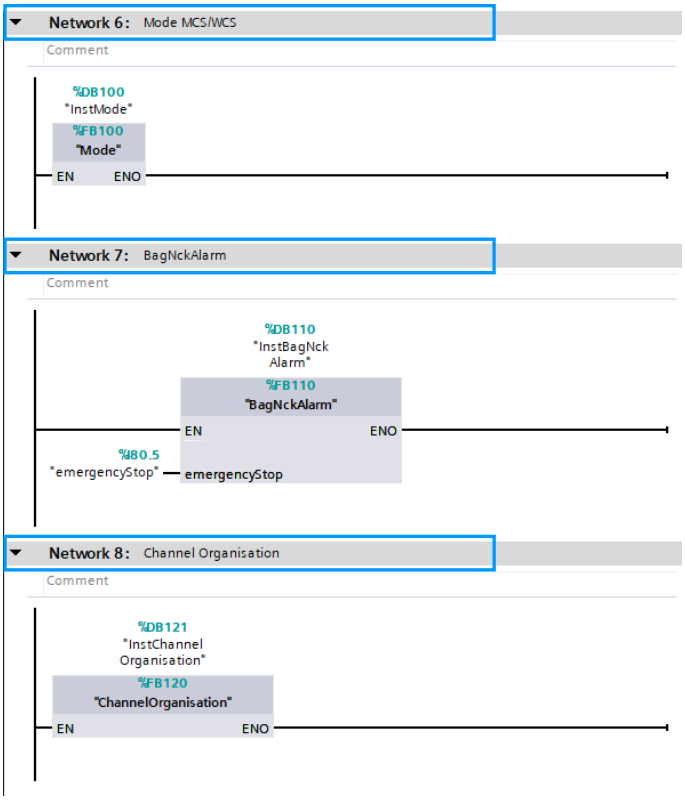
Notes:

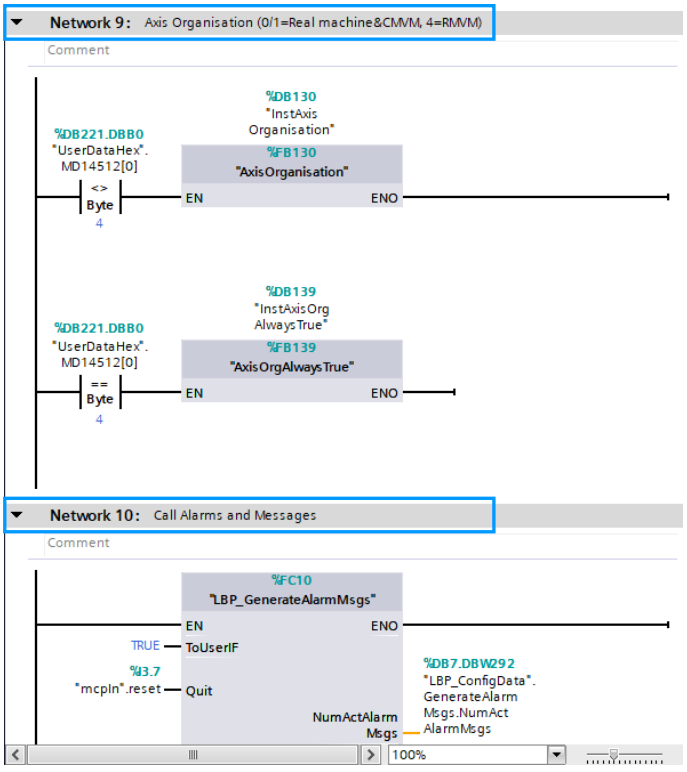
Calling user blocks in OB1

The newly created user blocks must be called in OB1.





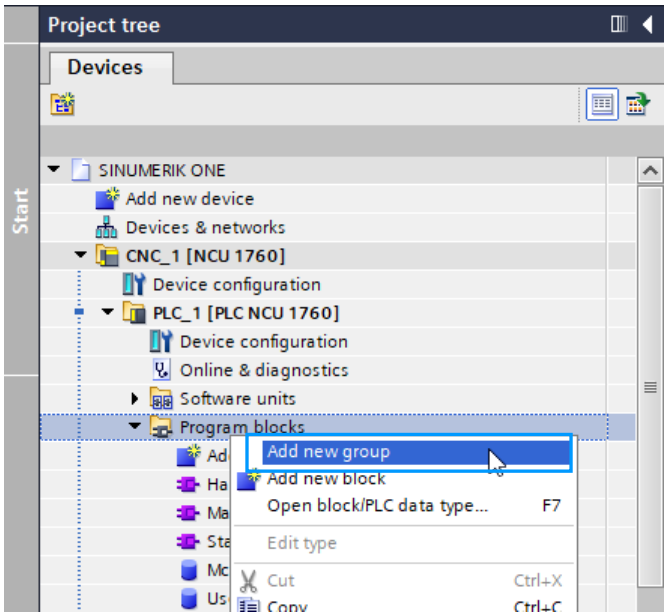


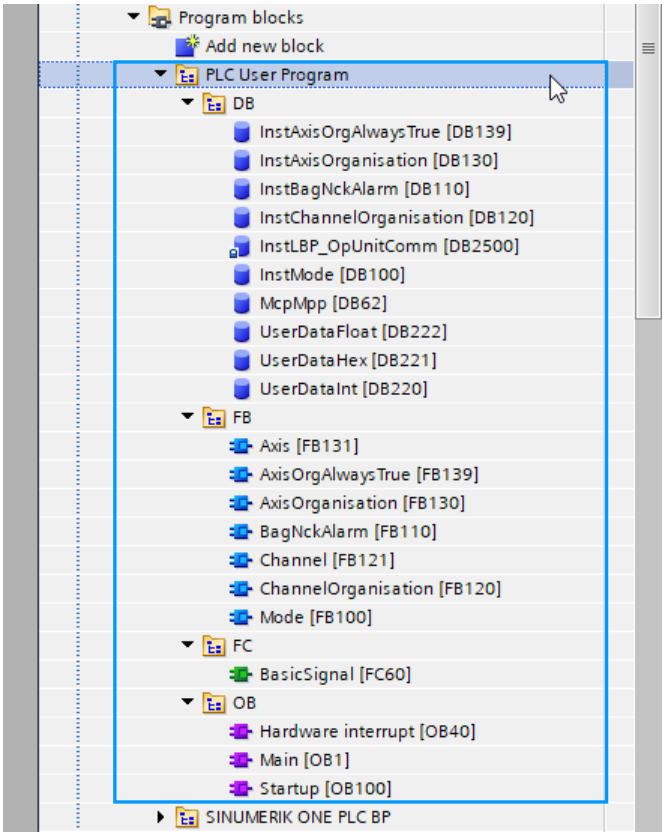


Notes:

Grouping user blocks

Create a new group in the project navigation.





Set machine type

In the example the PLC machine data are used to build the PLC user program universally, the machine type is defined by a PLC machine data.

MD14512[0] $MN_USER_DAT_HEX
=0H Real Maschine
=1H Create MyVirtual Machine without Simit
=2H Create MyVirtual Machine with Simit
=4H Run MyVirtual Machine

Virtual

Set machine type “Create MyVirtual Machine” via PLC machine data 14512[0] =1H.

Real

Set machine type “Real machine” via PLC machine data 14512[0] =0H.

A113: END

Notes:

A123

SINUMERIK ONE

Alarms and Messages

Module Description:

Alarms and messages are essential to inform operators, maintenance personnel etc. of the state of the machine.
Alarms and messages can be configured to have different behaviour on the operation of the machine.
User help files can be created to give a more detailed description of each alarm/message.
This module describes how the alarms and messages are activated.
The effect an alarm has on the system is described and how to acknowledge an alarm.

Module objective:

Upon completion of this module you will be able to perform the following tasks:

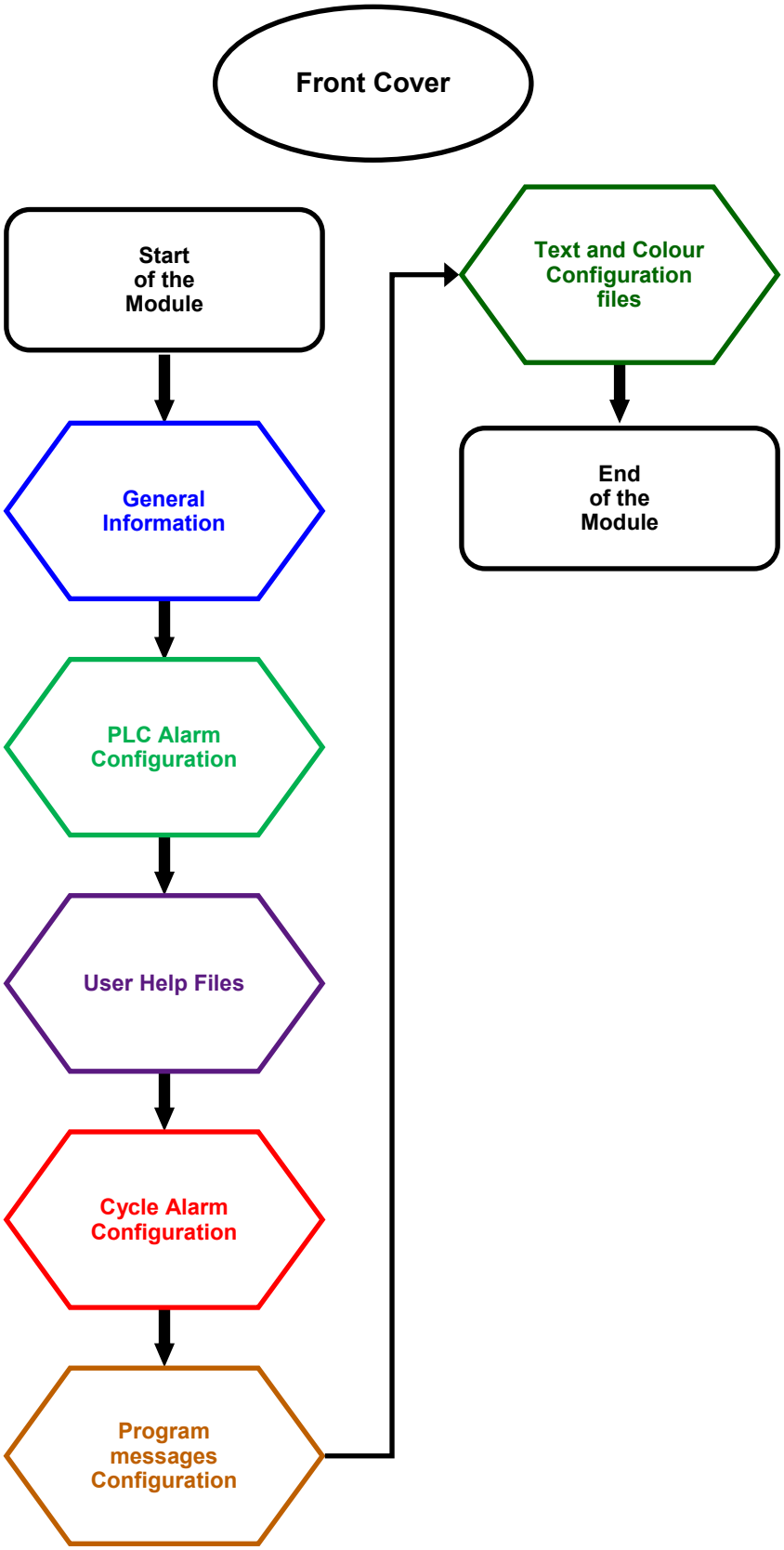
- Configure, activate and set the attributes of messages and alarms and create user help files.

Content:

General Information
PLC Alarm Configuration
User Help Files
Cycle Alarm Configuration
Program Messages Configuration
Text and Colour Configuration files

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
General Information
PLC Alarm Configuration
User Help Files
Cycle Alarm Configuration
Program messages Configuration
Text and Colour Configuration files
End of the Module

General information

The PLC alarms/messages are sub-divided into the following ranges:

Message/Alarm No.	Function
400000 - 499999 *2)	PLC messages in general
500000 - 599999 *1)	PLC messages for channel
600000 - 609999 *1)	PLC messages for axis and spindle
700000 - 709999 *1)	PLC messages for user
800000 – 809999 *2)	PLC messages for sequence chains/ graphics
810000 - 810009 *2)	PLC System messages
900000 - 999999 *2)	HMI Pro alarms

*1)

Messages/alarms within these ranges are activated via the user PLC. Each message/alarm has an interface bit within DB2 to facilitate the activation of each individual alarm. DB2 can be configured using DB5.

The reaction of the machine to alarms within the 500000/600000 range is predefined by the system.

Alarms within the 700000 range do not have any effect on the machine unless they are configured to do so via the user PLC program.

*2)

Messages/alarms within these ranges are system generated and cannot be configured by the user.

Note:

It is possible to import the alarm texts into TIAP? (But then no longer editable in Sinumerik ONE)

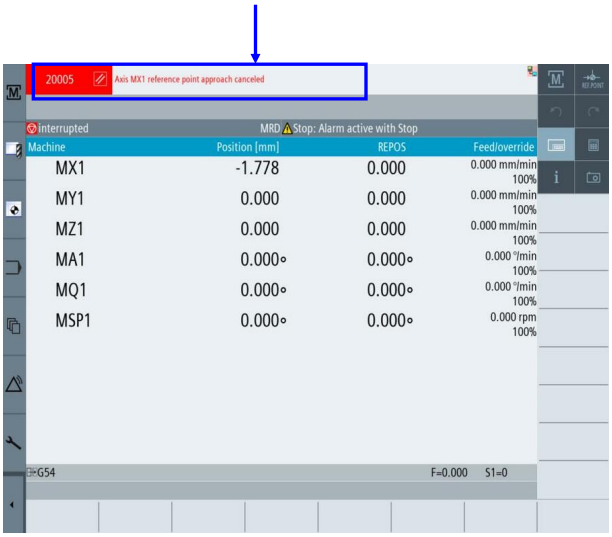
As soon as a fault occurs on the machine, the corresponding alarm will be displayed in the message line at the top of the screen.
The alarm number, acknowledgement method and a short text description will be displayed.

Further alarms are indicated by a downward pointing arrow, to the right of the alarm number.

The alarms/messages can be displayed in red or black. Red is traditionally used to indicate a more serious alarm and black is used for a message.

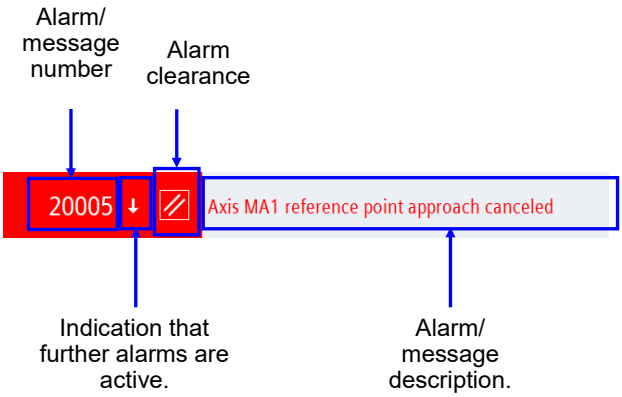
The alarm/message text can be entered via the HMI of the controller or created offline using TiA portal or AMM.

Alarms/messages appear at the top of the screen



Alarm/message details.

Notes:



A list of active alarms/messages can be viewed.



The alarms and messages are categorised and accessed via different soft-keys

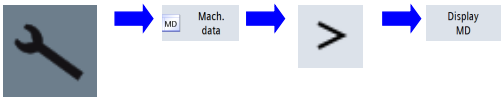


The “Alarm list” screen displays serious alarms



The “Messages” screen displays less serious alarms .

The alarms displayed at the top of the screen can be made to cycle in turn after a set time.
This is controlled by Display Machine Data 9056.



Display machine data			
9009	$MM_KEYBOARD_STATE	2	po
9032	$MM_HMI_MONITOR	""	po
9056	$MM_ALARM_ROTATION_CYCLE	300 ms	po
9057	$MM_ENABLE_CHANNEL_MSG_FILTER	1	po

With values of less than 500 there is no rotation cycle. Values of 500 to 10000 set the time that each alarm/message is displayed for.
The units are milliseconds.
The machine data is activated with an NC reset.



PLC Alarm Configuration

User PLC Requirements

PLC Interface signals (DB2)

DB5 is used for activating the alarms/messages.
Each alarm number has its own interface bit which, when set high will activate the alarm.

Whether an alarm behaves as an Error Message (EM) or Operator Message (OM) is pre-defined for all PLC user alarm in the 500000 and 600000 ranges (but can also be changed in DB5).

Note:
SINUMERIK ONE uses - ExtendALMsg ONLY.
With this setting, the alarm reaction can be configured using DB5

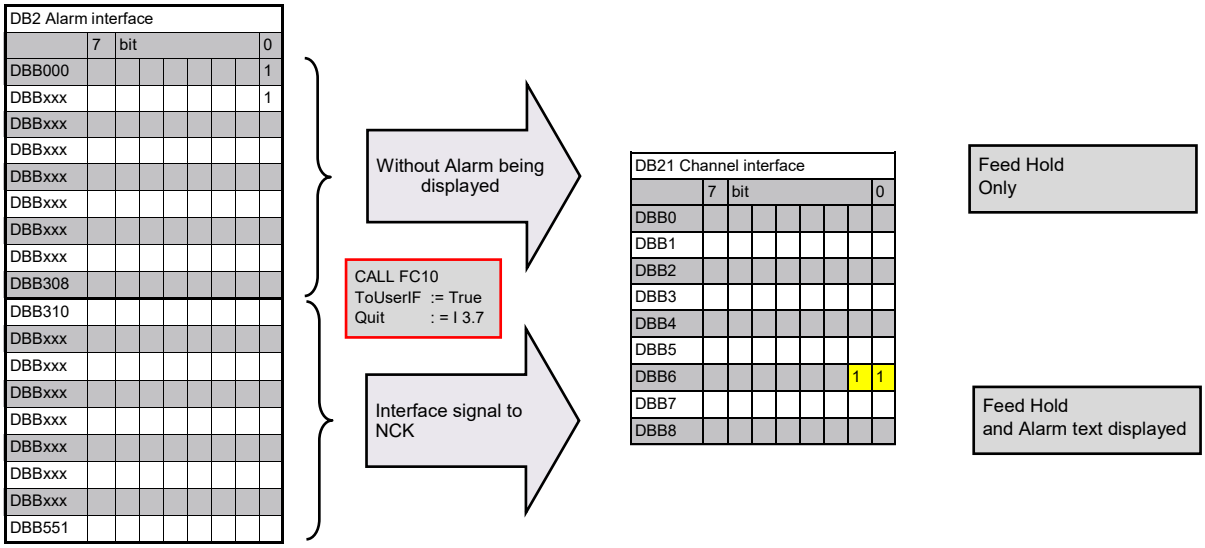
A list of all the interface signals is found in Module Appendix C PLC Interface.

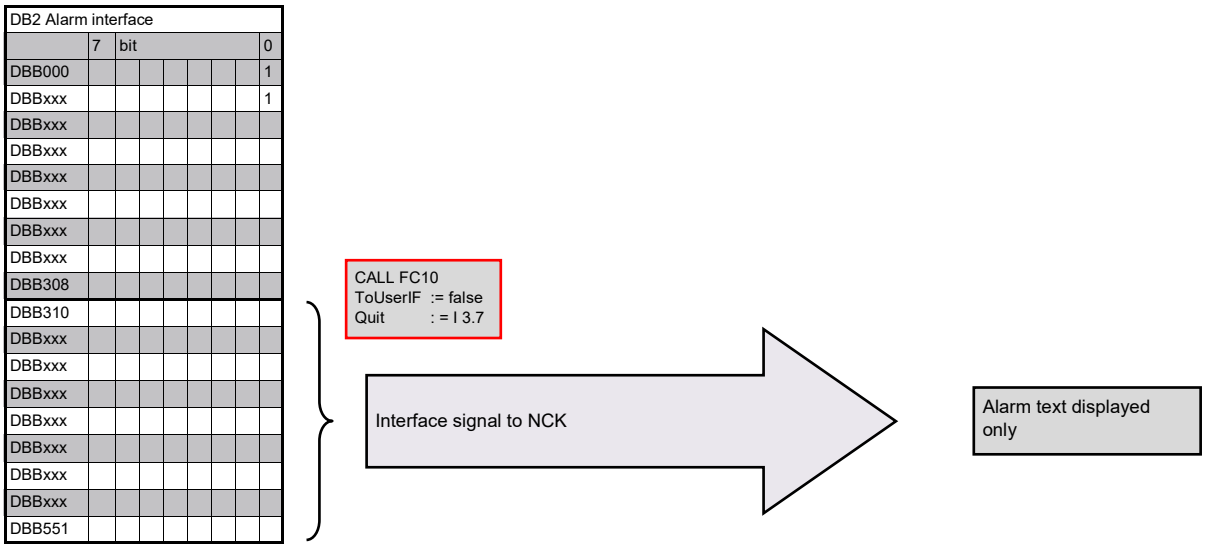
Notes:

Following is the Function diagram based upon FC10 parameter ToUserIF = True and False

These diagrams show alarms without text and with text

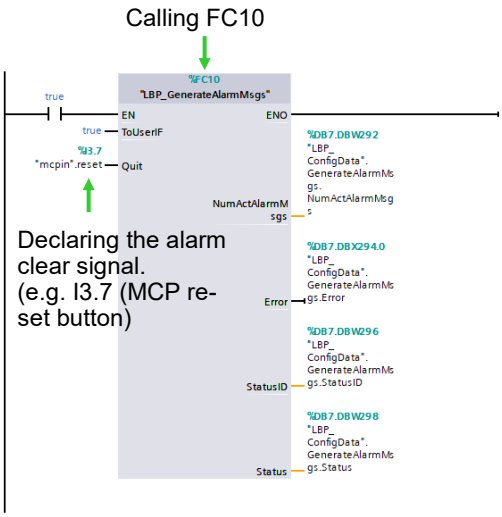
NOTE:
For alarms 510000 to 501315 and 600100 to 602315 only (OEM alarms are unaffected)





Notes:

To make the alarm system operational it is necessary to call and parameterize FC10 in the user PLC program.



FC10 parameter : **ToUserIF:=TRUE**
Error messages/alarms in the 500000 and 600000 ranges (Set via DB2) will be displayed. The controller **will** automatically issue feed-hold, read-in disables etc.

Channel Alarms 510000 to 511315:

The following table is a sample of DB2. This shows the interface signals for alarms 510000 to 511315. These are alarms for channel 1. (See Module Appendix C for additional channel signals)

The **red** text denotes that the alarm is classed as an Error Message. Alarms in the **blue** text classifies them as Operator Messages.

Error messages, when activated, will appear on the “Alarm list” screen and require an acknowledgement to clear. (FC10 “Quit” function e.g. I3.7)

Operator Messages will appear on the “Messages” screen and remain displayed until the interface bit is set to 0.
The effect of a message on the system is also shown for each byte.

DB2 interface signals for alarms 510000 to 591315

DB2 Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
310	510007	510006	510005	510004	510003	510002	510001	510000
	Feed only disable-Error message (EM)							
311	510015	510014	510013	510012	510011	510010	510009	510008
	Feed only disable-Error message (OM)							
312	510107	510106	510105	510104	510103	510102	510101	510100
	Feed and read-in disable-Error message (EM)							
313	510115	510114	510113	510112	510111	510110	510109	510108
	Feed and read-in disable-Error message (EM)							
314	510123	510122	510121	510120	510119	510118	510117	510116
	Feed and read-in disable-Operator message (OM)							
315	510131	510130	510129	510128	510127	510126	510125	510124
	Read-in disable-Operator message (OM)							
316	510207	510206	510205	510204	510203	510202	510201	510200
	Read-in disable-Error message (EM)							
317	510215	510214	510213	510212	510211	510210	510209	510208
	Read-in disable-Error message (EM)							
318	510223	510222	510221	510220	510219	510218	510217	510216
	Read-in disable-Operator message (OM)							

DB2 Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
319	510231	510230	510229	510228	510227	510226	510225	510224
	Read-in disable-Operator message (OM)							
320	510307	510306	510305	510304	510303	510302	510301	510300
	NC start disable-Error message (EM)							
321	510315	510314	510313	510312	510311	510310	510309	510308
	NC start disable-Operator message (OM)							
322	511107	511106	511105	511104	511103	511102	511101	511100
	Feed stop Geometry axis-Error Message (EM)							
323	511115	511114	511113	511112	511111	511110	511109	511108
	Feed stop Geometry axis-Operator message (OM)							
324	511207	511206	511205	511204	511203	511202	511201	511200
	Feed stop Geometry axis-Error Message (EM)							
325	511215	511214	511213	511212	511211	511210	511209	511208
	Feed stop Geometry axis-Operator message (OM)							
326	511307	511306	511305	511304	511303	511302	511301	511300
	Feed stop Geometry axis-Error Message (EM)							
327	511315	511314	511313	511312	511311	511310	511309	511308
	Feed stop Geometry axis-Operator message (OM)							

Notes:

Axis and Spindle Alarms 600100 - 603115

Alarms in the 600100 - 699999 range are used for creating a feed/spindle stop.
Two bytes are reserved for each axis/spindle. The first of which is defined as error messages (EM) while the second byte is defined as operator messages (OM).

DB2 interface signals for alarms 600100 to 603115:

DB2 Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
490	600107	600106	600105	600104	600103	600102	600101	600100
	Feed stop/spindle stop for axis/spindle 1							
491	600115	600114	600113	600112	600111	600110	600109	600108
492	600207	600206	600205	600204	600203	600202	600201	600200
	Feed stop/spindle stop for axis/spindle 2							
493	600215	600214	600213	600212	600211	600210	600209	600208
494	600307	600306	600305	600304	600303	600302	600301	600300
	Feed stop/spindle stop for axis/spindle 3							
495	600315	600314	600313	600312	600311	600310	600309	600308
496	600407	600406	600405	600404	600403	600402	600401	600400
	Feed stop/spindle stop for axis/spindle 4							
497	600415	600414	600413	600412	600411	600410	600409	600408

User PLC Alarms 700000 - 706363

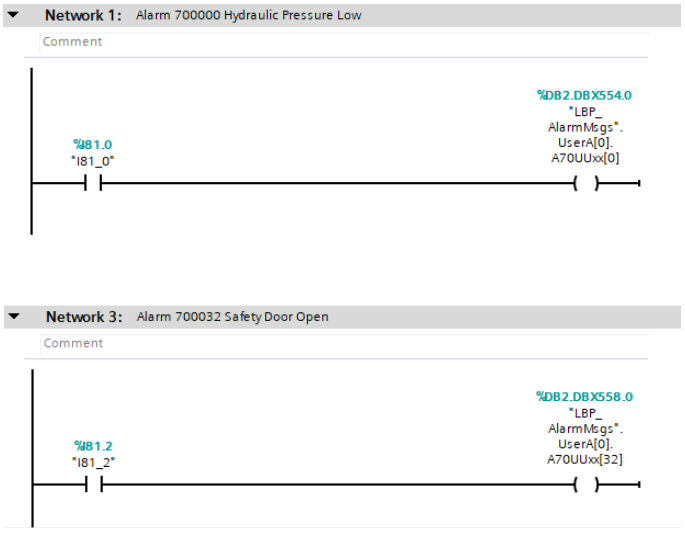
Alarms in the 700000 - 709999 range are free for configuring in the user PLC program.
The effect the alarm has on the system is entirely the responsibility of the machine manufacturer.
The alarms are separated into error messages (EM), and operator messages (OM)

DB2 interface signals for alarms **700000** to **706363**:

DB2 Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
554	700007	700006	700005	700004	700003	700002	700001	**700000**
555	700015	700014	700013	700012	700011	700010	700009	700008
556	700023	700022	700021	700020	700019	700018	700017	700016
557	700031	700030	700029	700028	700027	700026	700025	700024
558	700039	700038	700037	700036	700035	700034	700033	**700032**
559	700047	700046	700045	700044	700043	700042	700041	700040
560	700055	700054	700053	700052	700051	700050	700049	700048
561	700063	700062	700061	700060	700059	700058	700057	700056

Alarm Activation

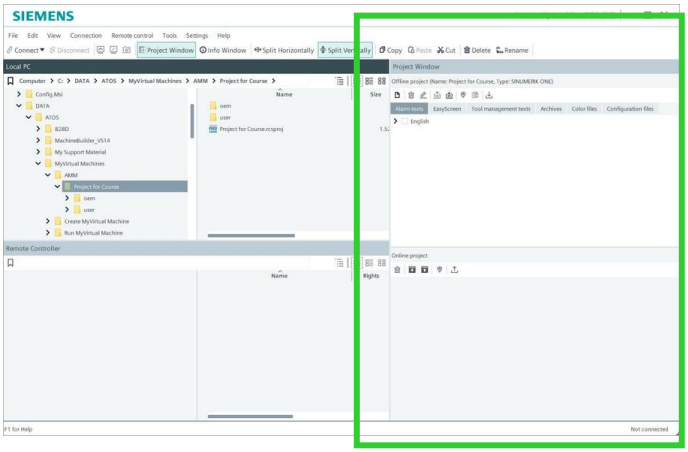
The interface signals have to be set in the user PLC.
The following example shows how alarms 700000 (EM) and 700032 (OM) are activated in TiA portal.



Creating language files using AMM-Project.

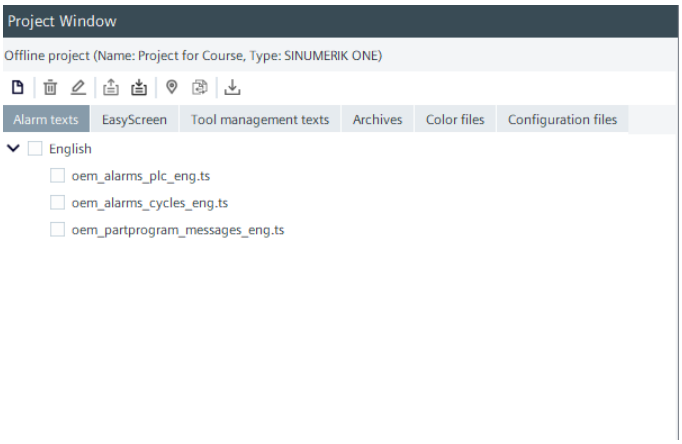
Access MyMachine can be used to the create language specific files. Start the Access MyMachine program.

Make sure the project, created previously in module A101, appears on the right-hand side.

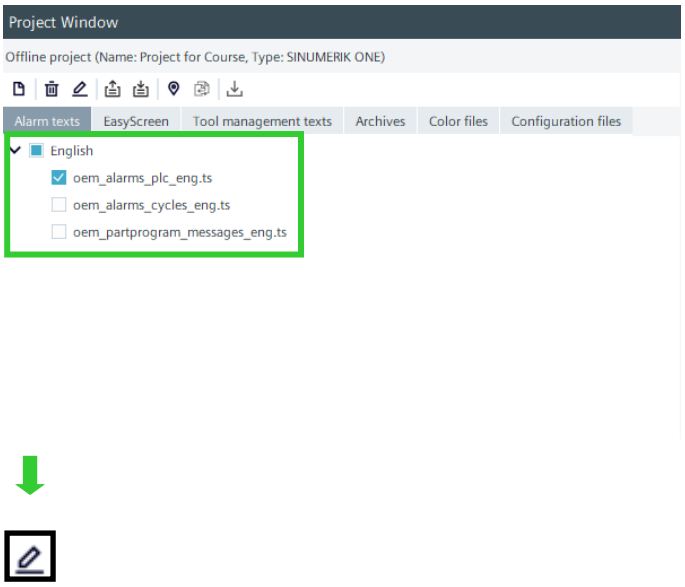


Notes:

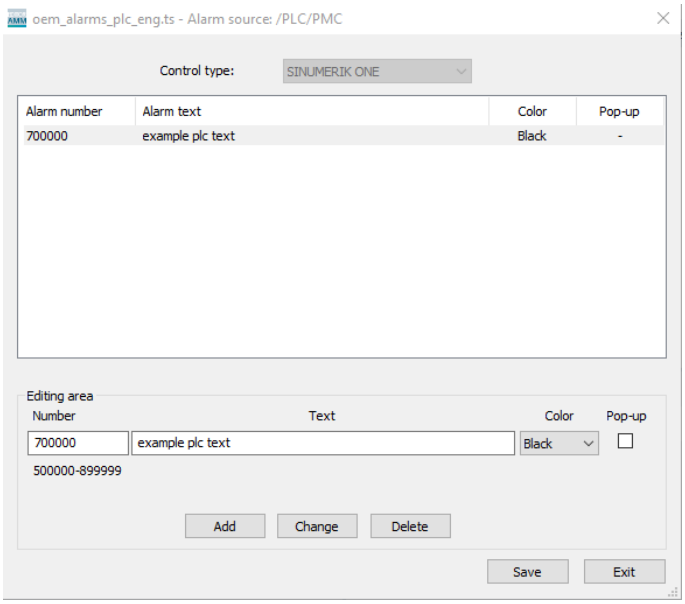
Expand the content of the English in the project window.



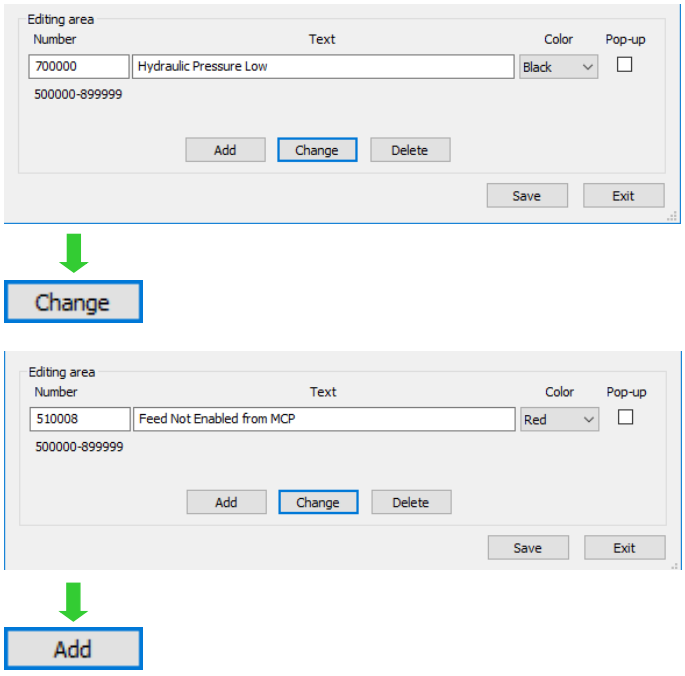
Select the file "oem_alarms_plc_eng.ts. and open it using the "Edit" key.



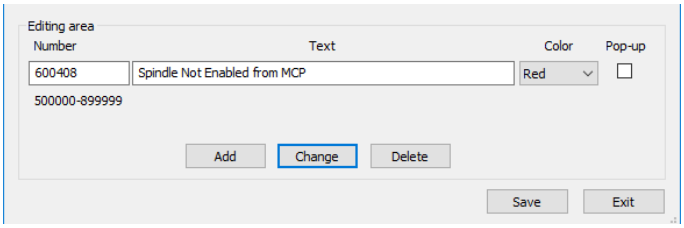
Edit the required text for the first language.



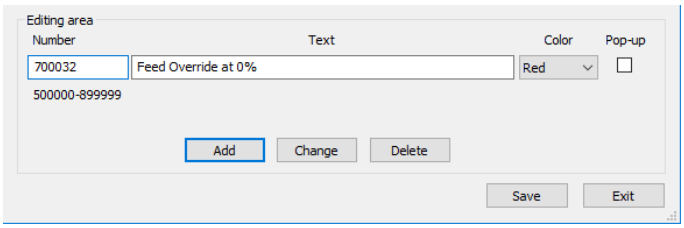
In the Edit window, change the alarm number and required text followed by “Change or “Add”



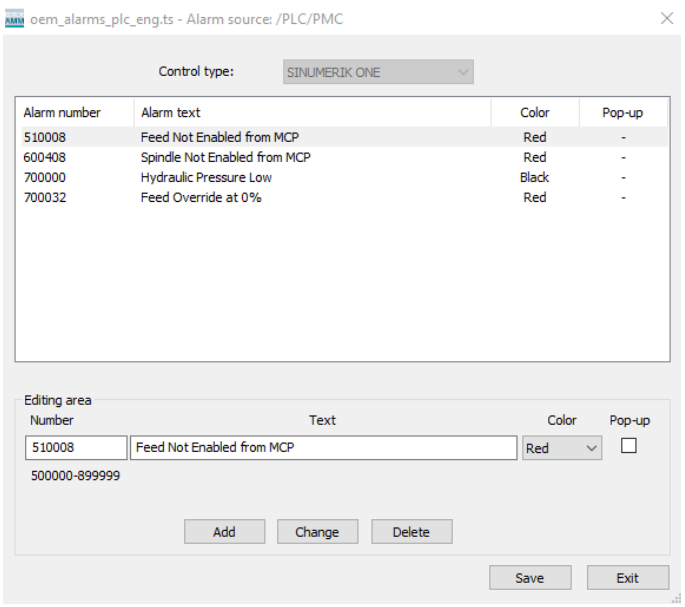
Notes:





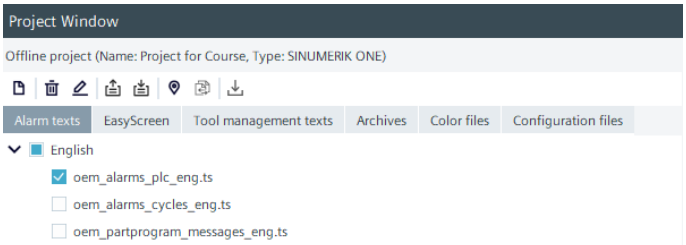




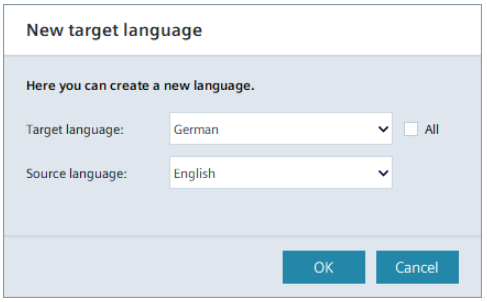




Example Adding additional language specific files for the alarm texts:
To add additional language files for the alarm texts, highlight the "Alarm texts" menu, followed by "New target language"







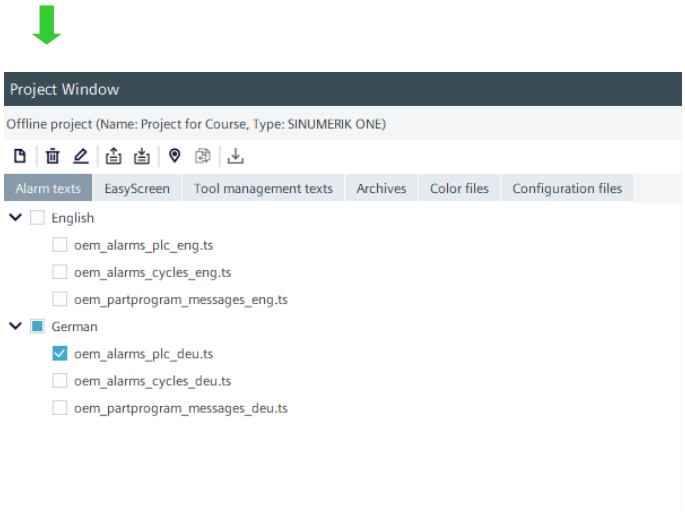
Note:

Target lang:	select from the remaining 8 standard languages
Master copy:	select the master language from the control.
All languages:	Allows you to pick the full remaining 21 languages.

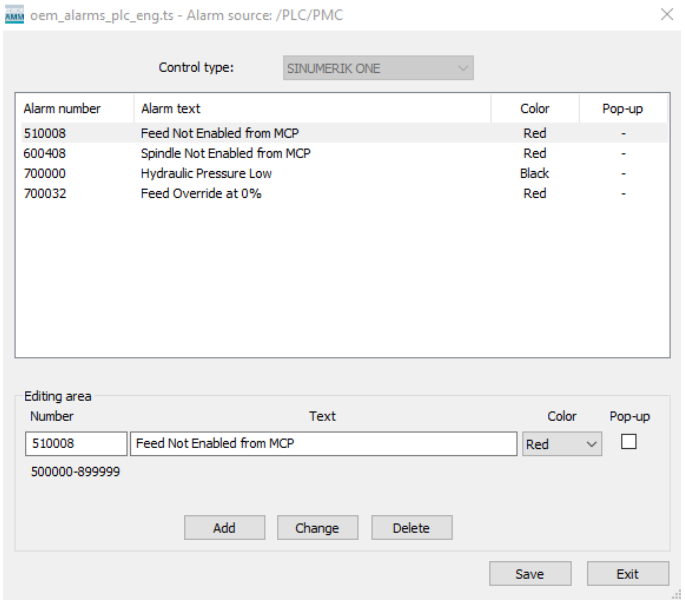


Notes:

Select the file "oem_alarms_plc_deu.ts. and open it using the "Edit" key.

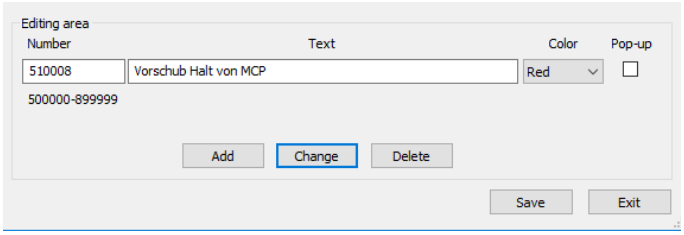


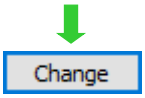


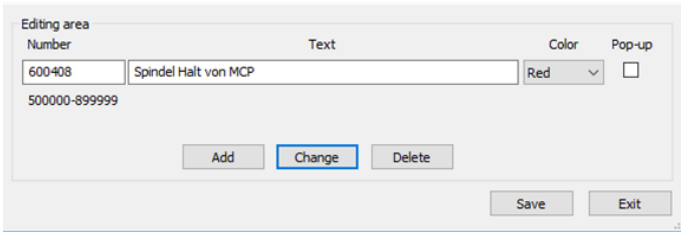


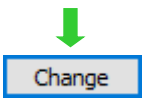
Enter the required text for the second language.

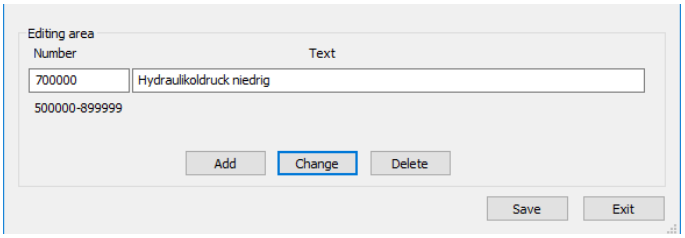
In the Edit window, change the alarm number and required text followed by “Change”

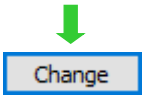


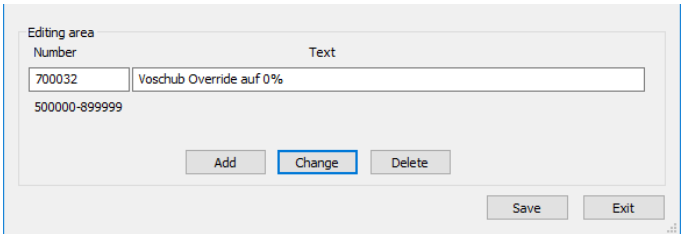


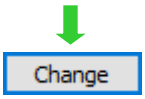








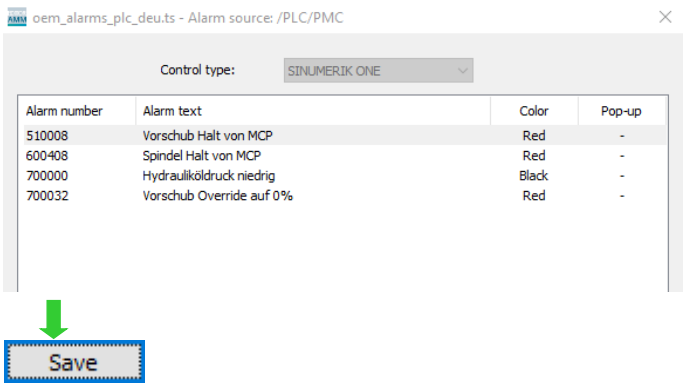




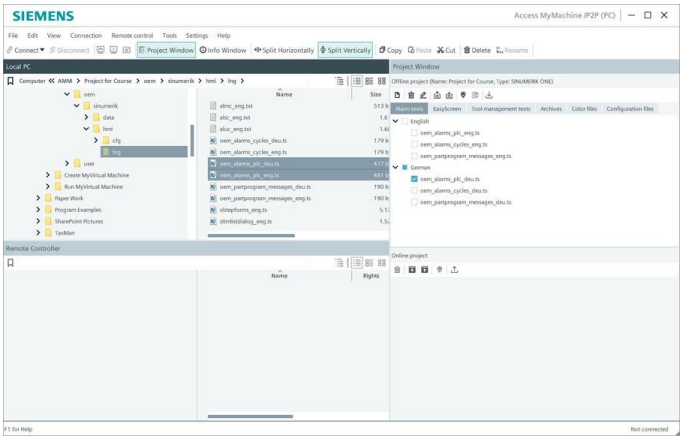


Notes:

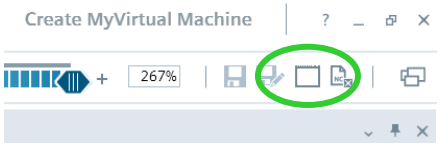
German translation:
510008 Vorschub Halt von MCP
600408 Spindel Halt von MCP
700000 Hydrauliköldruck niedrig
700032 Vorschub Override auf 0%



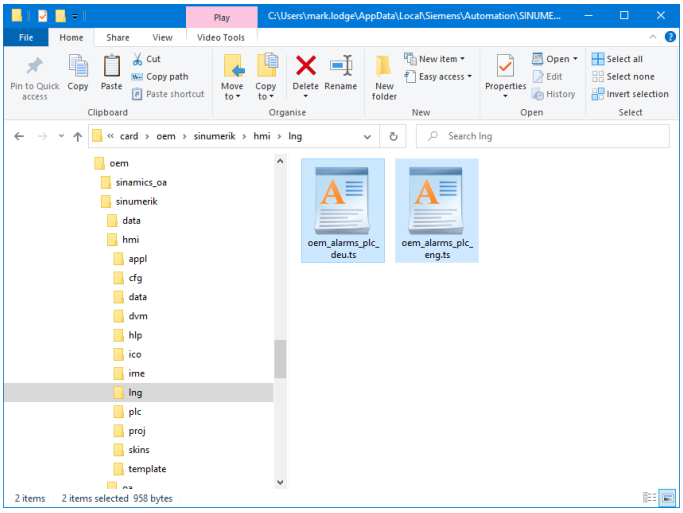
Language files can be copied from AMM into the virtual machine.



Copy and paste this “oem_alarms_plc_deu.ts and oem_alarms_plc_eng.ts” files into the virtual machine.



This will open up windows explorer, where the language file can be located.



When the texts have been completed it is necessary to switch the controller off/on to activate the changes.

The data is now saved to a **ts** file in the following location:
NCU: SINUMERIK data\oem\sinumerik\hmi\lng
IPC: C:\ProgramData\Siemens\MotionControl\ oem\sinumerik\hmi\lng

The file name is:
oem_alarms_plc_eng.ts. The “**eng**” part of the name is the language identifier.

Language identifier examples:
eng = English
deu = German
fra = French
ita = Italian
esp = Spanish
chs = Chinese (Simplified)
cht = Chinese (Traditional)
kor = Korean
ptb = Portuguese (Brazil)

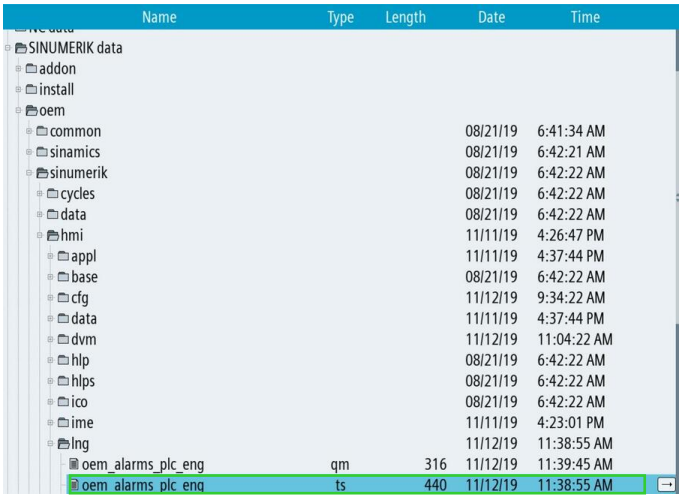
rebooting the HMI, in the virtual machine.



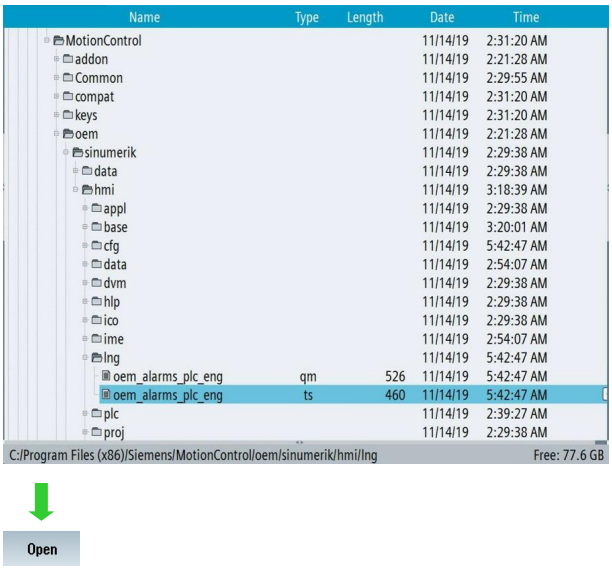
NOTE
Any alarm text created in AMM that is to be used in an IPC can only be transferred using a USB memory stick, as AMM will NOT connect to an IPC.

Notes:

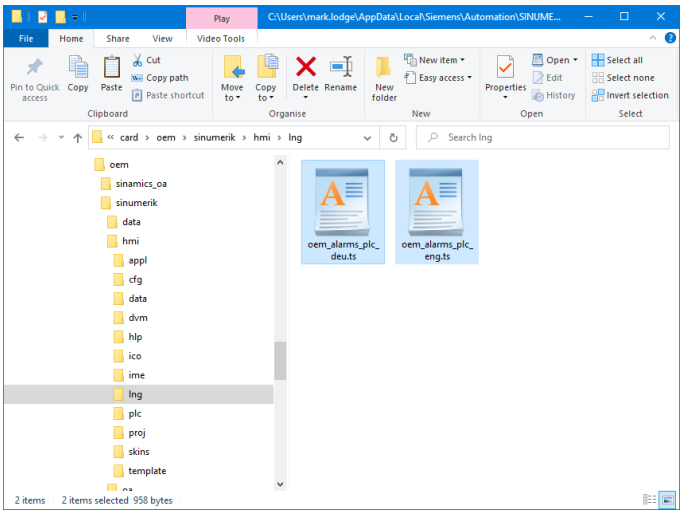
NCU: Location of the alarm text file.



IPC: Location of the alarm text file.

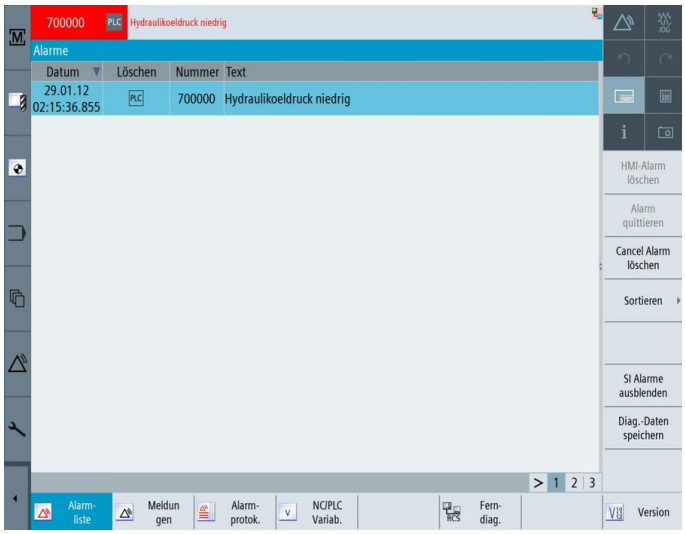


Virtual machine: Location of the alarm text file.

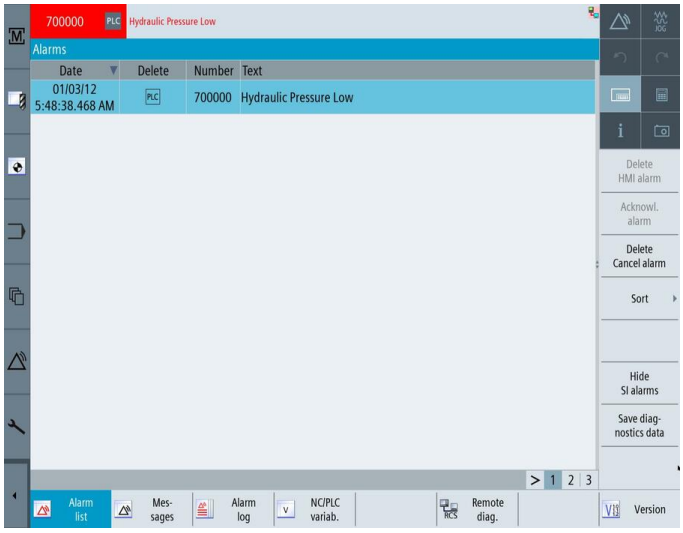


Notes:

German:



English:



Notes:

User Help Files

To further enhance the fault finding procedure it is possible to create a “Help” file.
When the HELP key is pressed the details of the selected alarm will be displayed on the screen.

Help file information and Rules:
The help file is of the type HTML (**H**yper **T**ext **M**ark-up **L**anguage) and needs to be stored in a specific directory.
File location for English language:
NCU: SINUMERIK data\oem\sinumerik\hmi\hlp**eng**\sinumerik_alarm_plc_pmc
IPC: C:\ProgramData\Siemens\MotionControl\oem\sinumerik\hmi\hlp**eng**\sinumerik_alarm_plc_pmc

The italicised text denotes folders that require creating. Separate folders are required for each language where the Help file is to be utilised.
Examples:
English - ***eng****sinumerik_alarm_plc_pmc*
German - ***deu****sinumerik_alarm_plc_pmc*

The HTML file name:
sinumerik_alarm_oem_plc_pmc.html

A separate file needs to be created and stored in each of the language folders.

HTML script language

Once the file is created the HTML code can be entered. HTML uses *tags*, each tag has a specific function.
Tags have to be opened and closed in the correct manner, example:

<html> Opening the “html” tag
Information/more tags entered here etc.
</html> Closing the “html” tag

Table of Common HTML Tags:

TAG	Description
html	Root element-Main tag for the help file, contains all other tags, first to be opened and last to be closed.
head	Contains document information, opened immediately after the HTML tag, closed directly before the BODY tag is opened.
body	The main body of the HTML document, determines what will be displayed on the screen. Opened immediately after the closing the HEAD tag. Closed immediately before the HTML tag.
table	Creates a table for the help file information.
tr	Table row, creates a row within the table.
td	Table data cell. Used within the **tr** tag, contains the text that is to be displayed on the help screen. The tag is used with the **width** attribute to determine the width of the cell and the v**align** attribute to align the text within a cell.
p	Used to begin a new paragraph.
b	Text will be written in **bold** type.
i	Text will be written in *italic* type.
br	Starts a new line.

Notes:

Example help file
Below is an example help file for alarm 700000. The text has been coloured to assist with understanding the contents.

```
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset="UTF-8"/><title></title></head>
<body>
<table>
<tr>
<td width="15%"><b><a name="700000">700000</a></b></td>
<td width="85%"><b>Hydraulic Pressure Low</b></td>
</tr>
<tr>
<td valign="top" width="15%"><b>Parameter:</b></td>
<td width="85%">N/A</td>
</tr>
<tr>
<td valign="top" width="15%"><b>Explanation:</b></td>
<td width="85%">I8.1 is low.</td>
</tr>
<tr>
<td valign="top" width="15%"><b>Reaction:</b></td>
<td width="85%">Emergency Stop Initiated</td>
</tr>
<tr>
<td valign="top" width="15%"><b>Remedy:</b></td>
<td width="85%">The hydraulic unit has indicated a fault. Check oil level. Check pressure switch. <br />Clean filter if necessary.</td>
</tr>
```

```
<tr>
<td valign="top" width="15%"><b>Part number:</b></td>
<td width="85%">Hydraulic oil type: AAA. Pressure switch type: BBB <br /> Filter type CCC</td>
</tr>
<tr>
<td valign="top" width="15%"><b>Program continuation:</b></td>
<td width="85%">Machine restart required</td>
</tr>
</table>
<p></p>
</body>
</html>
```

The **“a”** tag is known as an anchor.
The system is able to cross reference the highlighted alarm with the entry within the “a” tag.

<td width="15%">700000</td>

When an alarm is selected and the help key is pressed the system uses the anchor to locate and display the correct table contents for the selected alarm.

Adding further alarms to the HTML file is simply a case of copying/pasting the existing “table” tag and its contents and editing to suit the individual alarm.

The contents of the “head” tag should not be changed. This determines the content type and text format.

Notes:

Creating the help files/directories on the controller

The folders can be created directly on the controller or by using the Access MyMachine software package.

Creating the help file folders on the controller:

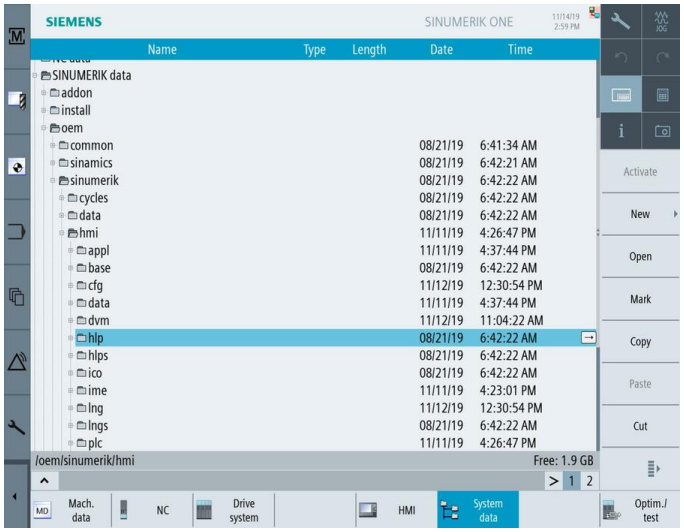
Locate the directory where the folders are to be created.

NCU: SINUMERIK data\oem\sinumerik\hmi\hlp

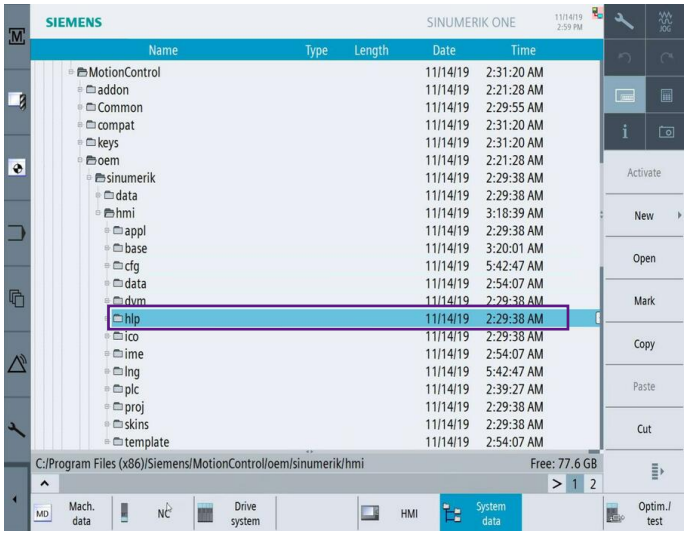
IPC: C:\ProgramData\Siemens\MotionControl\oem\ sinumerik\hmi\hlp



NCU “hlp” folder location:



IPC “hlp” folder location:



Cursor to the “hlp” directory and select the “New” soft-key.

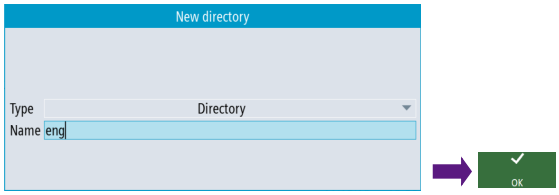


Select the “Directory” soft-key, this is usually already selected by default.



Enter the directory name, this is the language directory. Several will have to be created if help files are to be available for more than one language. See previous page for language specific folder names.

Below: Creating a folder for English help files.



Notes:

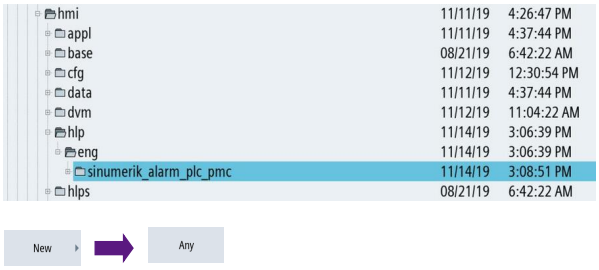
Highlight the “eng” folder and select the “New” soft-key.



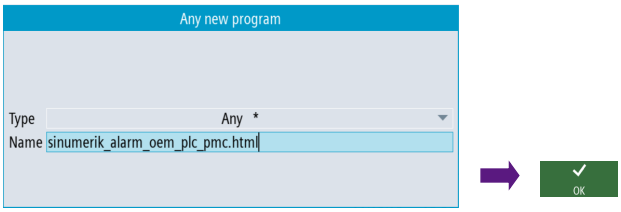
Enter the mandatory name for the new directory:
“**sinumerik_alarm_plc_pmc**”.



Creating the help file:
Cursor to “**sinumerik_alarm_plc_pmc**” directory and select the “New” soft-key followed by the “Any” soft-key.



Type in the help file name
“sinumerik_alarm_oem_plc_pmc.html“.



The file will automatically open in readiness for the HTML code to be entered.

The file is closed with the “exit” soft-key:



Creating the help files/directories on an external PC

It is also possible to create the directories/help files offline using an ASCII editor (e.g. Notepad®).
This is usually a more comfortable method than stood at the controller.
These files have to be transferred to the controller and placed in the correct folders using the USB Interfaces or with the Access MyMachine software.

The file needs to be placed in certain folders on the controller.
The **“sinumerik_alarm_oem_plc_pmc.html”** file has to be placed in one of the following directories:

NCU:
SINUMERIK data\user\sinumerik\hmi\hlp\eng\sinumerik_alarm_plc_pmc
“SINUMERIK data\oem\sinumerik\hmi\hlp\eng\sinumerik_alarm_plc_pmc

IPC:
C:\ProgramData\Siemens\MotionControl\user\sinumerik\hmi\hlp\eng\sinumerik_alarm_plc_pmc
C:\ProgramData\Siemens\MotionControl\oem\sinumerik\hmi\hlp\eng\sinumerik_alarm_plc_pmc

Notes:

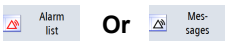
Testing the help file on the controller

To test the Help file the alarm specified in the Help file should be activated.
Alarm 700000 is activated by DB2.DBX554.0

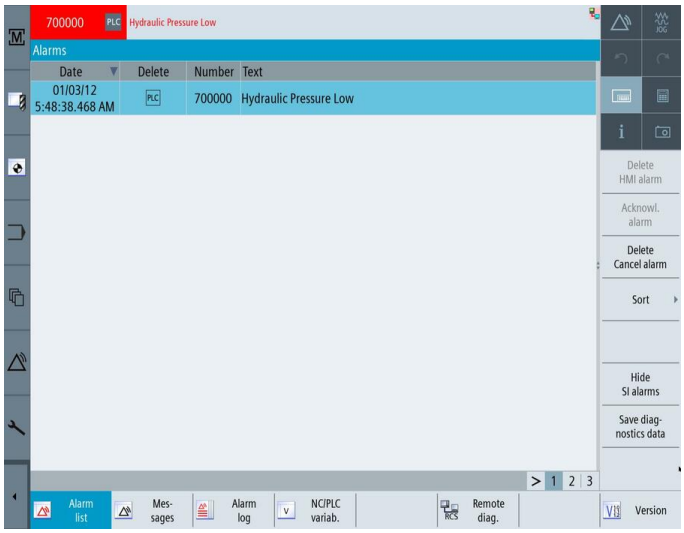
With the alarm active locate the “Alarms/Messages” screen.



The alarm will appear in either the “Alarmlist” or “Messages” screen depending how it has been configured.

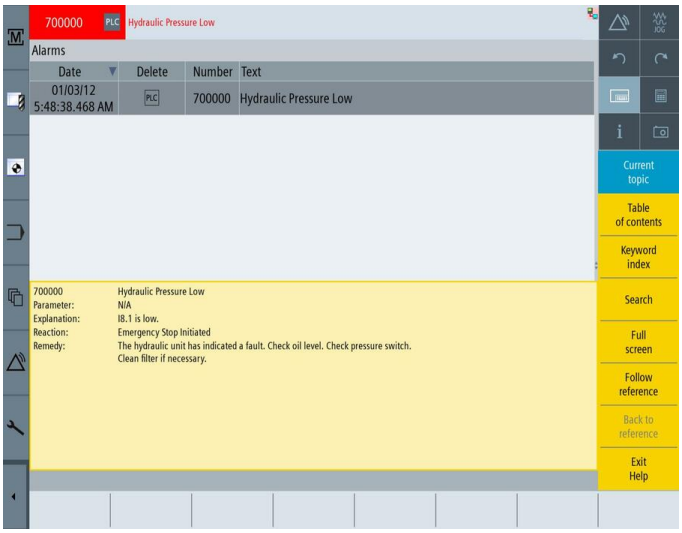


If several alarms exist, cursor to the correct one.



The HELP function is activated by the “HELP” key.







Notes:

Cycle Alarm Configuration

Alarms can be set from within an NC program. Alarms are displayed in alarm window in either RED or BLACK text.
An example is shown using the SETAL command.

…
N10 SETAL(65000)
…

The valid range for alarm numbers lies between 60000 and 69999, 0f which 60000 to 64999 are reserved for SIEMENS cycles and 65000 to 69999 are available to OEM / User.

The alarm text must be entered into the language specific files in the control via AMM or HMI alarm text editor, which is described in this module.

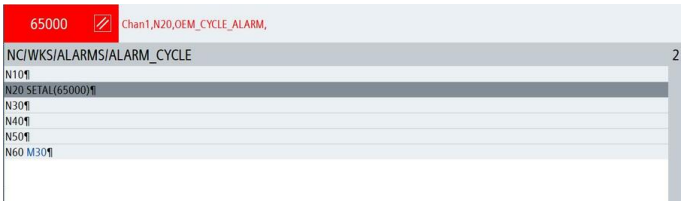
When entering the alarm text string there are 4 parameters that can be used.

Parameter	Meaning
%1	Channel number
%2	Block number
%3	Text index for cycle alarm
%4	Additional alarm parameters

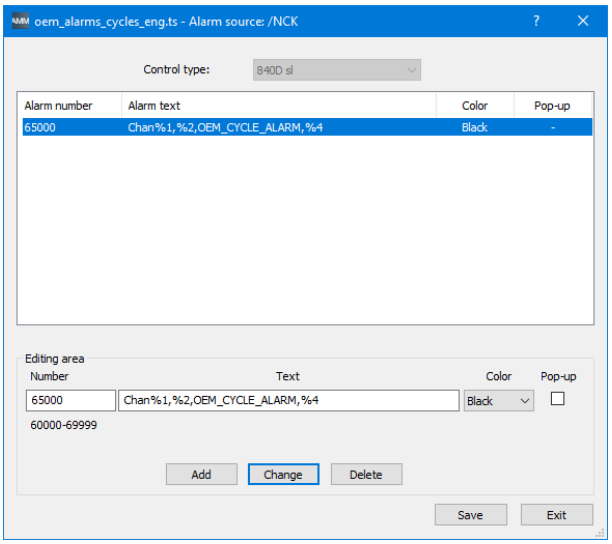
Example text entered into the language file via AMM or HMI alarm text editor:

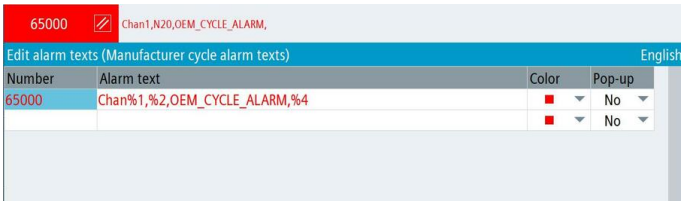
Chan%1,%2,OEM_CYCLE_ALARM,%4

The effect is shown below:



View's of the entered text using AMM and the HMI text editor.





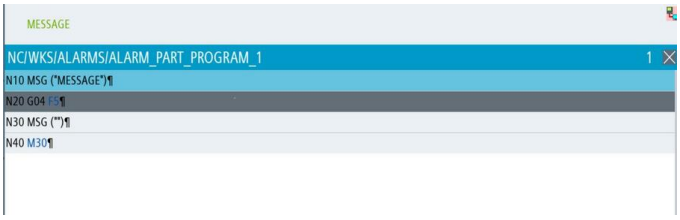
Notes:

Program Messages Configuration

Messages can be set from within an NC program. Messages are displayed in alarm window in GREEN text.
An example is shown using the MSG command.

…
N10 MSG(“MESSAGE”)
…

The effect is shown below:

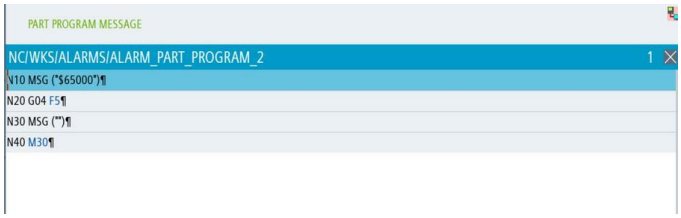


The alarm text can also be entered into the language specific files in the control via AMM or HMI alarm text editor, which is described in this module.

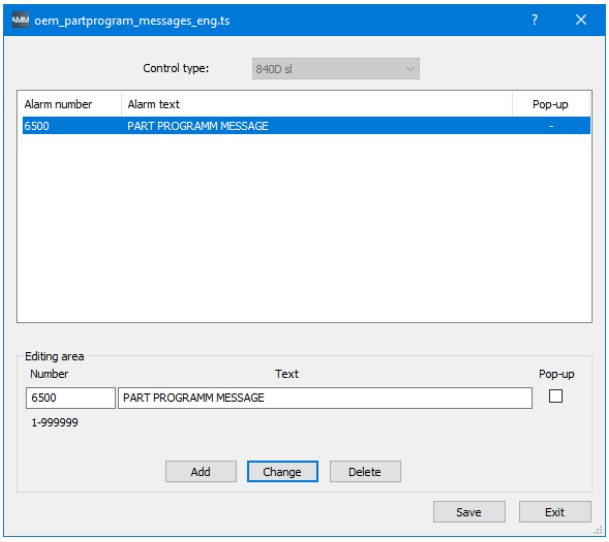
The text in the NC program is entered as a “text identifier”, starting with the character “$”

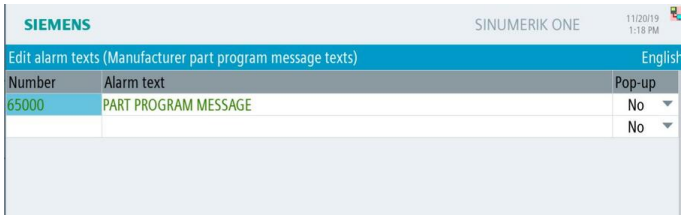
Eg MSG(“$65000”) and text added into language specific file “PART PROGRAM MESSAGE”

The effect is shown below:



View's of the entered text using AMM and the HMI text editor.





Text and Colour Configuration files

The standard colours for the HMI alarms and messages are, RED or BLACK for PLC and CYCLE alarms and GREEN for MESSAGES. These colours can be chosen when using the HMI alarm text editor and in AMM.

There is the ability to register an OEM file that is not restricted by these basic colours.
The colour of :-
Alarm number
Alarm number background
Alarm text
Alarm text background
Can be changed. In the following example a “CYCLE ALARM” background will be changed to “Yellow”

We must first declare a new OEM file in the configuration files for the text and colour. The name of these file is going to be “alarm_example”.

Notes:

First create the text configeration file slaesvcadapconf.xml in following directory: /oem/sinumerik/hmi/cfg, and add the line.

This is the edited content of the file.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CONFIGURATION>
   <AlarmTexts>
      <BaseNames>
         <AlarmTextEditor_oem_alarms_plc type="QString" value="oem_alarms_plc"/>
         <AlarmTextEditor_oem_alarms_cycles type="QString" value="oem_alarms_cycles"/>
         <AlarmTextEditor_oem_partprogram_messages type="QString" value="oem_partprogram_messages"/>
          <AlarmTextEditor_oem_alarm_example type="QString" value="oem_alarm_example"/>
      </BaseNames>
   </AlarmTexts>
</CONFIGURATION>
```

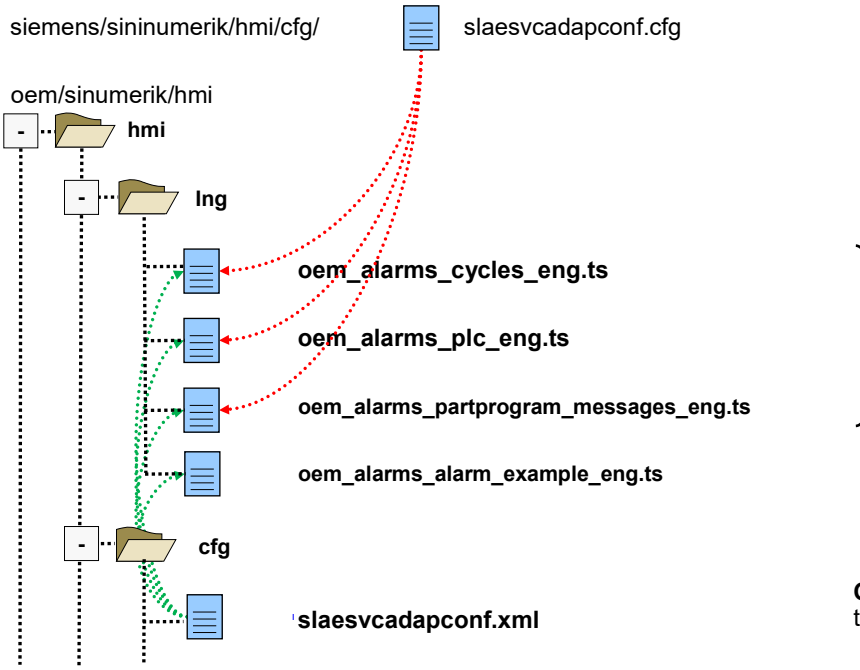
Next create a file oem_alarm_example_eng.ts in the following directory: /oem/sinumerik/hmi/lng,

This is the content of the file.

```
<!DOCTYPE TS>
<TS>
   <context>
         <name>slaeconv</name>
         <message><source>65200/NCK</source>
         <translation>Chan%1,%2,alarm_example,%4</translation>
         </message>
   </context>
</TS>
```

Notes:

OEM Alarm text Configuration file location



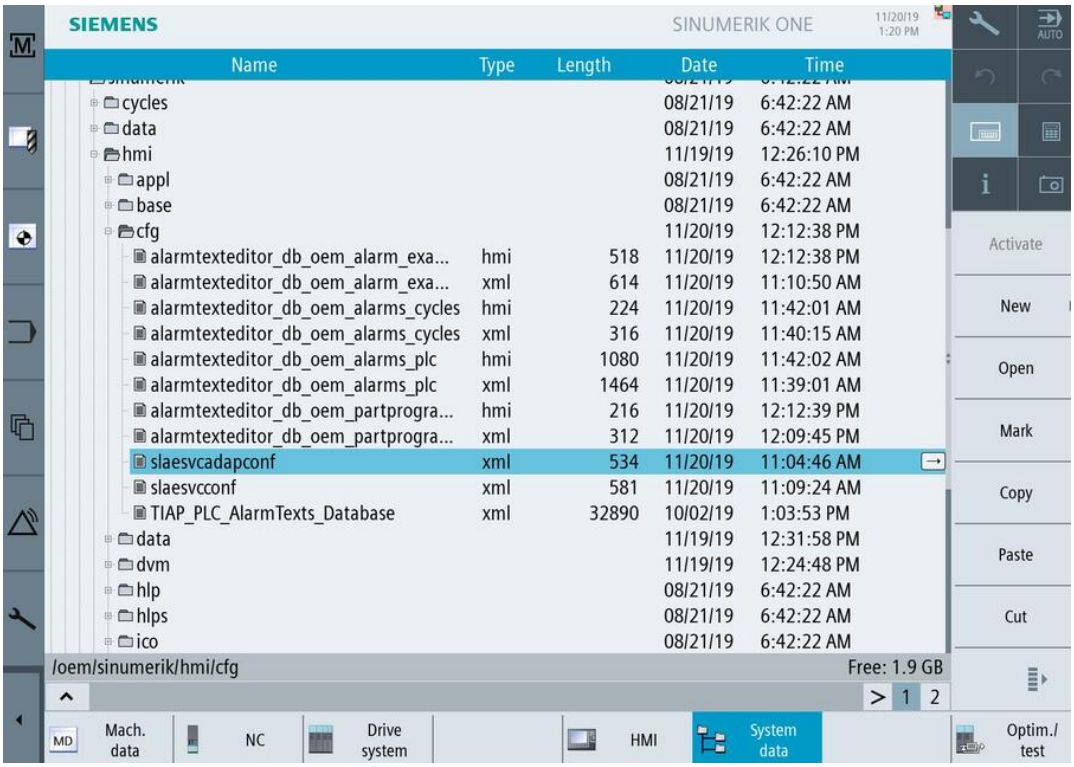
Siemens default configeration file

The standard Alarm text files that are generated when text is entered via the HMI editor or using AMM

oem_alarms_alarm_example_eng.ts is declared in the slaesvcadapconf.xml

OEM_Configuration file that is generated when text is entered via the HMI editor or using AMM

Additional alarm text files must be declared in slaesvcadapconf. xml!
A sample-configuration file "oem_slaesvcadapconf. xml" is in following directory: /siemens/sinumerik/hmi/template/cfg.
But it must be renamed "slaesvcadapconf. xml" when transfered to its destination folder.



Notes:

Create the colour configeration file slaesvcconf.xml in following directory: /oem/sinumerik/hmi/cfg.

This is edited content of the file.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Configuration of the Solutionline Alarm & Event Service -->
<CONFIGURATION>
        <DataBases>
                <OEM_DataBase_01 type="QString" value="oem_slaedatabase"/>
                <OEM_DataBase_02 type="QString" value="alarmtexteditor_db_oem_alarms_cycles"/>
                <OEM_DataBase_03 type="QString" value="alarmtexteditor_db_oem_alarms_plc"/>
                <OEM_DataBase_04 type="QString" value="alarmtexteditor_db_oem_partprogram_messages"/>
                <OEM_DataBase_05 type="QString" value="alarmtexteditor_db_oem_alarm_example"/>
        </DataBases>
</CONFIGURATION>
```

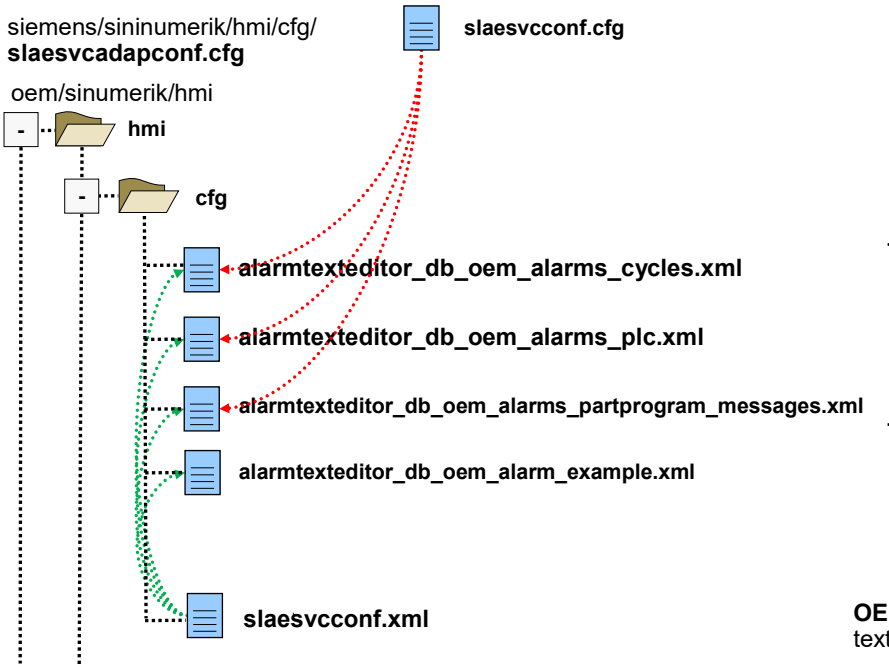
Next create a file alarmtexteditor_db_oem_alarm_example.xml in the following directory: /oem/sinumerik/hmi/cfg,

This is edited content of the file.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE SlAeAlarmAttributes>
<SlAeAlarmAttributes Version="01.00.00.00">
        <Types></Types>
        <Sources>
        <Source CatLink="1" SourceID="0" SourceURL="/NCK">
        <Alarms>
        <Alarm AlarmID="65200"><DISPLOC>0</DISPLOC>
        <MSGTEXT>slaeconv|65200/NCK</MSGTEXT>
        <NUMBERBACKGROUNDCOLOR>#FF0000</NUMBERBACKGROUNDCOLOR>
        <NUMBERCOLOR>#FFFFFF</NUMBERCOLOR>
        <TEXTBACKGROUNDCOLOR>#FFFF00</TEXTBACKGROUNDCOLOR>
        <TEXTCOLOR>#FF0000</TEXTCOLOR>
        </Alarm></Alarms>
        </Source></Sources>
</SlAeAlarmAttributes>
```

Notes:

OEM Color configuration file



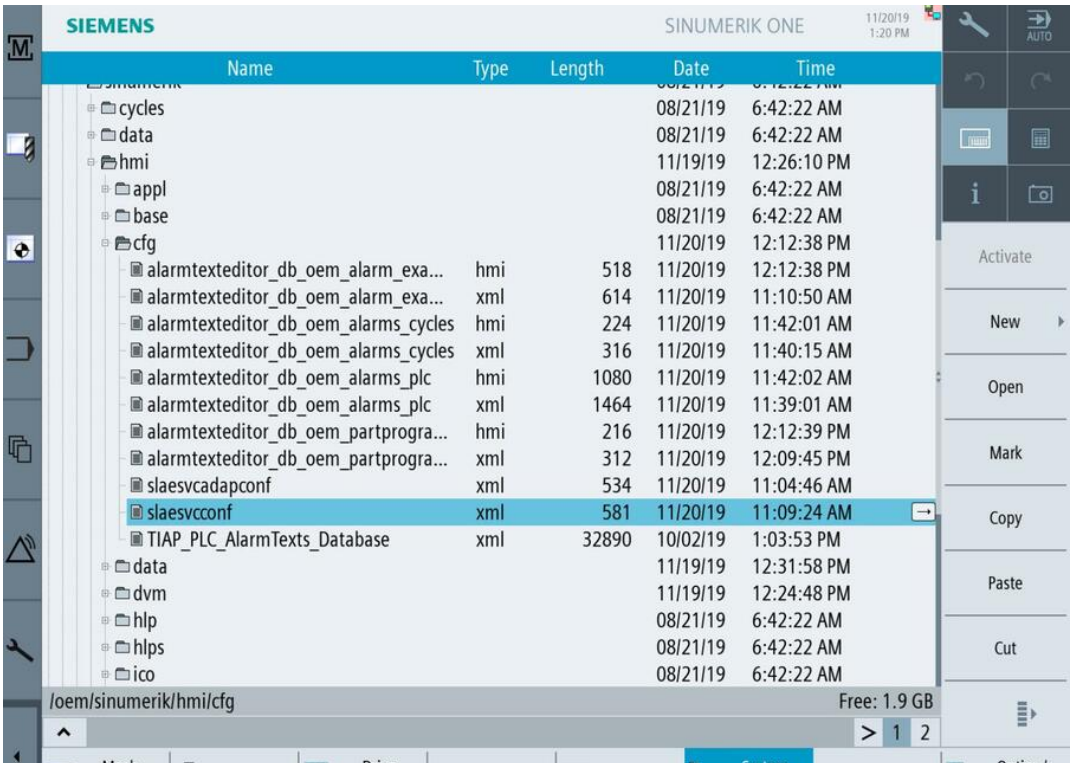
Siemens default configeration file

Color files that are generated when colour is entered via the HMI editor or using AMM

alarmtexteditor_db_oem_alarm_example.xml is created by OEM so that non standard colours can be used

OEM_Configuration file that is generated when text is entered via the HMI editor or using AMM

Additional colour files must be declared in slaesvcconf. xml
A sample-configuration file "oem_slaesvcconf. xml" lies in following directory: /siemens/sinumerik/hmi/template/cfg.
But it must be renamed "slaesvcconf. xml" when transfered to its destination folder.



Notes:

There is a html colour code list which is reprecented by a hexidecimal number.

HTML Codes format:
Each HTML code contains symbol "#" and 6 letters or numbers. These numbers are in hexadecimal numeral system. For example "FF" in hexadecimal represents number 255 in Decimal.

Meaning of symbols:
The first two symbols in HTML color code represents the intensity of **red** color. 00 is the least and FF is the most intense. The third and fourth represents intensity of **green** and fifth and sixth represents the intensity of **blue**. So with combining the intensity of red, green and blue an unlimited colour pallet can be created;)

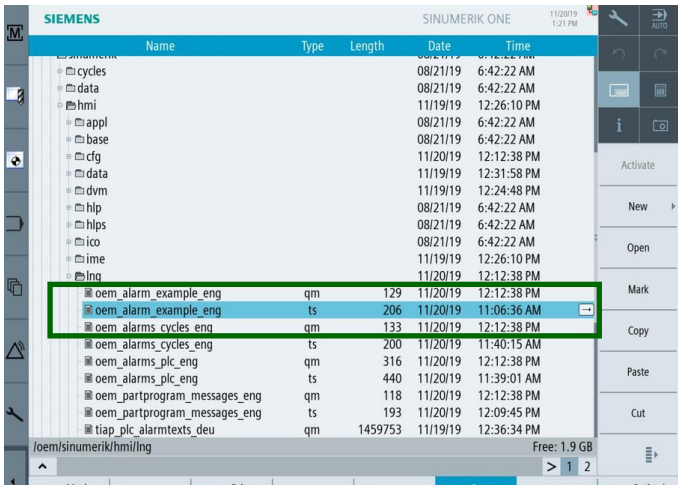
By defining the colour in the alarmtexteditor_db_oem_test.xml we can adjust:
Alarm number
Alarm number background
Alarm text
Alarm text background

Color Name	HEX	Crimson	#DC143C
AliceBlue	#F0F8FF	Cyan	#00FFFF
AntiqueWhite	#FAEBD7	DarkBlue	#00008B
Aqua	#00FFFF	DarkCyan	#008B8B
Aquamarine	#7FFFD4	DarkGoldenRod	#B8860B
Azure	#F0FFFF	DarkGray	#A9A9A9
Beige	#F5F5DC	DarkGreen	#006400
Bisque	#FFE4C4	DarkKhaki	#BDB76B
Black	#000000	DarkMagenta	#8B008B
BlanchedAlmond	#FFEBCD	DarkOliveGreen	#556B2F
Blue	#0000FF	DarkOrange	#FF8C00
BlueViolet	#8A2BE2	DarkOrchid	#9932CC
Brown	#A52A2A	DarkRed	#8B0000
BurlyWood	#DEB887	DarkSalmon	#E9967A
CadetBlue	#5F9EA0	DarkSeaGreen	#8FBC8F
Chartreuse	#7FFF00	DarkSlateBlue	#483D8B
Chocolate	#D2691E	DarkSlateGray	#2F4F4F
Coral	#FF7F50	DarkTurquoise	#00CED1
CornflowerBlue	#6495ED	DarkViolet	#9400D3
Cornsilk	#FFF8DC	DeepPink	#FF1493

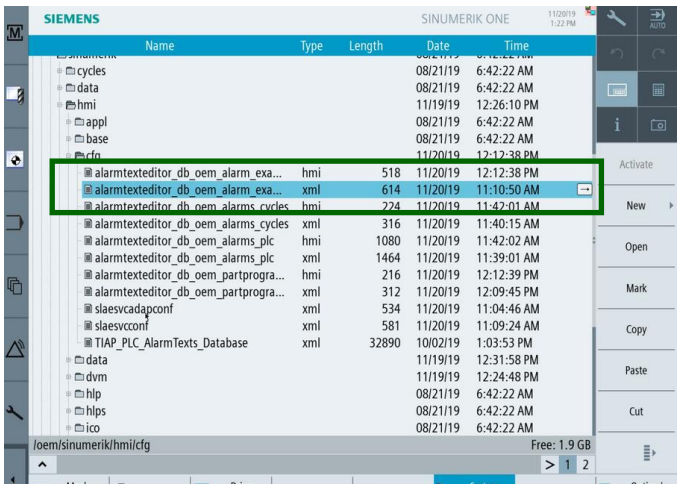
DeepSkyBlue	#00BFFF	Lavender	#E6E6FA	Magenta	#FF00FF
DimGray	#696969	LavenderBlush	#FFF0F5	Maroon	#800000
DodgerBlue	#1E90FF	LawnGreen	#7CFC00	MediumAquaMarine	#66CDAA
FireBrick	#B22222	LemonChiffon	#FFFACD	MediumBlue	#0000CD
FloralWhite	#FFFAF0	LightBlue	#ADD8E6	MediumOrchid	#BA55D3
ForestGreen	#228B22	LightCoral	#F08080	MediumPurple	#9370DB
Fuchsia	#FF00FF	LightCyan	#E0FFFF	MediumSeaGreen	#3CB371
Gainsboro	#DCDCDC	LightGoldenRodYellow	#FAFAD2	MediumSlateBlue	#7B68EE
GhostWhite	#F8F8FF	LightGray	#D3D3D3	MediumSpringGreen	#00FA9A
Gold	#FFD700	LightGreen	#90EE90	MediumTurquoise	#48D1CC
GoldenRod	#DAA520	LightPink	#FFB6C1	MediumVioletRed	#C71585
Gray	#808080	LightSalmon	#FFA07A	MidnightBlue	#191970
Green	#008000	LightSeaGreen	#20B2AA	MintCream	#F5FFFA
GreenYellow	#ADFF2F	LightSkyBlue	#87CEFA	MistyRose	#FFE4E1
HoneyDew	#F0FFF0	LightSlateGray	#778899	Moccasin	#FFE4B5
HotPink	#FF69B4	LightSteelBlue	#B0C4DE	NavajoWhite	#FFDEAD
IndianRed	#CD5C5C	LightYellow	#FFFFE0	Navy	#000080
Indigo	#4B0082	Lime	#00FF00	OldLace	#FDF5E6
Ivory	#FFFFF0	LimeGreen	#32CD32	Olive	#808000
Khaki	#F0E68C	Linen	#FAF0E6	OliveDrab	#6B8E23

Notes:

Orange	#FFA500	SandyBrown	#F4A460	Yellow	#FFFF00
OrangeRed	#FF4500	SeaGreen	#2E8B57	YellowGreen	#9ACD32
Orchid	#DA70D6	SeaShell	#FFF5EE		
PaleGoldenRod	#EEE8AA	Sienna	#A0522D		
PaleGreen	#98FB98	Silver	#C0C0C0		
PaleTurquoise	#AFEEEE	SkyBlue	#87CEEB		
PaleVioletRed	#DB7093	SlateBlue	#6A5ACD		
PapayaWhip	#FFEFD5	SlateGray	#708090		
PeachPuff	#FFDAB9	Snow	#FFFAFA		
Peru	#CD853F	SpringGreen	#00FF7F		
Pink	#FFC0CB	SteelBlue	#4682B4		
Plum	#DDA0DD	Tan	#D2B48C		
PowderBlue	#B0E0E6	Teal	#008080		
Purple	#800080	Thistle	#D8BFD8		
RebeccaPurple	#663399	Tomato	#FF6347		
Red	#FF0000	Turquoise	#40E0D0		
RosyBrown	#BC8F8F	Violet	#EE82EE		
RoyalBlue	#4169E1	Wheat	#F5DEB3		
SaddleBrown	#8B4513	White	#FFFFFF		
Salmon	#FA8072	WhiteSmoke	#F5F5F5		



oem_alarm_example_eng.ts, which states the text for the cycle alarm.This file is declared in slaesvcadapconf.xml configeration file.

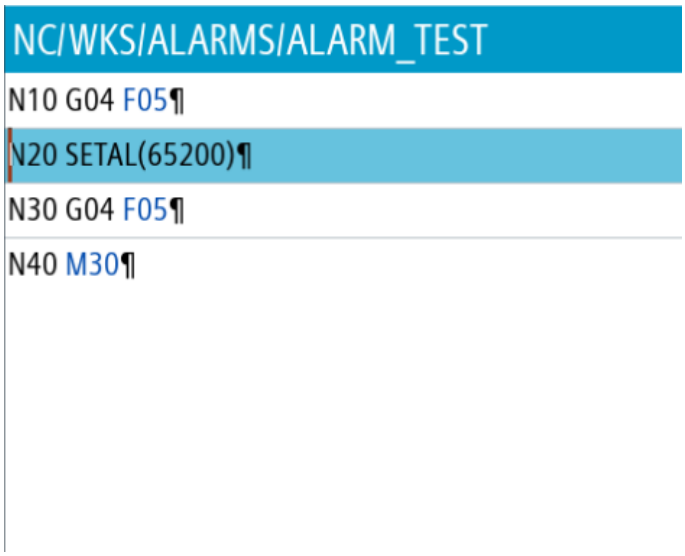


alarmtexteditor_db_oem_alarm_example.xml, which states the text for the cycle alarm.This file is declared in slaesvcconf.xml configeration file.

Notes:

Once all the files have been created and edited, reboot the vitual machine.





This simple program when run will post this alarm.



This method of creating alarm text and backgrounds with different coulours will not be affected by colour changes done by the HMI editor or AMM.

A123: END

Notes:

PLC Functions

Module Description:

The following functions are available in the PLC:

- Reading and writing of NC Variables
- PI Services
- Reading Global User Data
- Starting ASUP
- Axis / Spindle control

Module objective:

Upon completion of this module you will be able to perform the above listed PLC tasks.

Content:

Reading and writing of NC Variables

PI Services

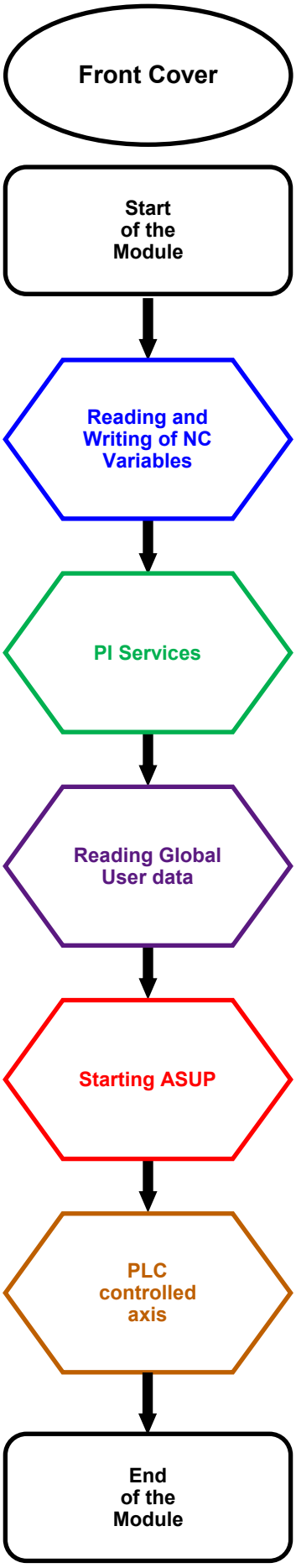
Reading Global User Data

Starting ASUP

Axis / Spindle control

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Reading and Writing of NC Variables
PI Services
Reading Global User data
Starting ASUP
PLC controlled axis
End of the Module

Reading and Writing of NC Variables

Function Description:

During the manufacturing process it may be necessary to have data available to the NC program from the automation system in order to optimize the NC machining process.
This could take the form of a numerical value being transferred to an R Parameter or a machine setting data receiving from the automation system an optimised value.
In the opposite direction it may be necessary to influence the automation system dependent upon the current NC machining process, positional data available in the automation system is an example of this.

Two function blocks are available for the function of:

Reading NC data - FB2
Writing NC data - FB3

In addition to the FBs, a tool is available for the engineering PC, this tool is required for generating the necessary parameter values for the two function blocks.

Prerequisites for the Function:

HMI Requirement:
None

Drive requirement (SINAMICS):
None

NCK Requirement:
None

PLC Requirement:
Parameterization of FB2 or FB3 in the PLC

Steps to achieve function:

To be able to access NC variables in the PLC user program, you must configure objects that are required for this in your project. Use the resources in the TIA Portal for this.

Using the NC variable list, you can select NC variables, put them together in a user-defined way, configure them, and add them to an NC variable data block.

In the programming editor, create the user program with the desired NC variable access.

For each NCU created in the project, there is the folder "Configured NC variables" under the “NCK” entry in the project tree. This folder contains the various NC variable tables.

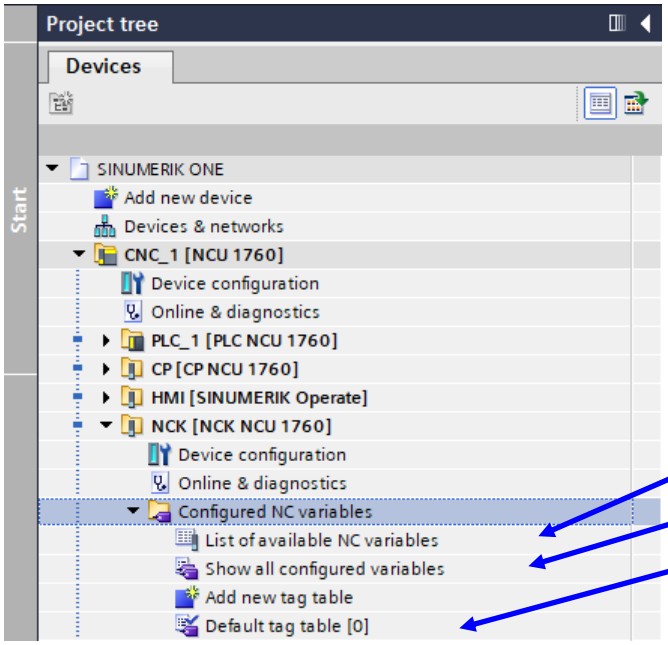


Table “List of available NC variables”

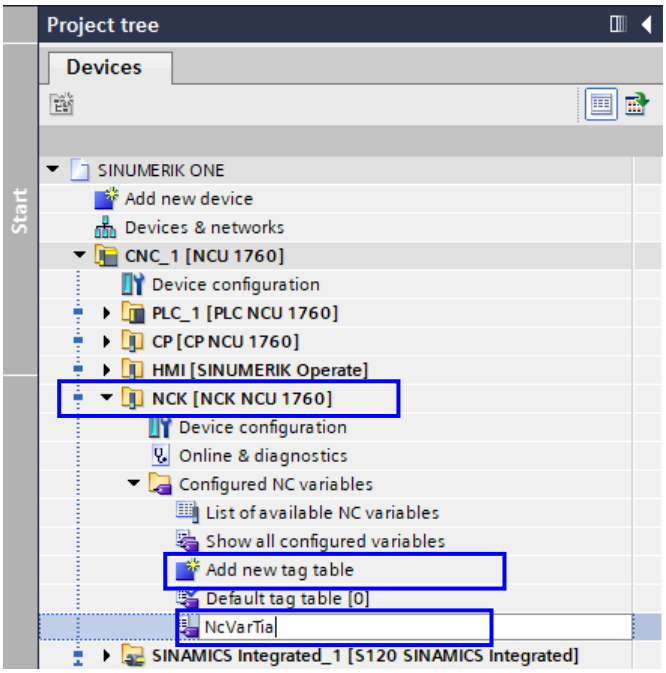
Table “All configured variables”

User-defined variable tables

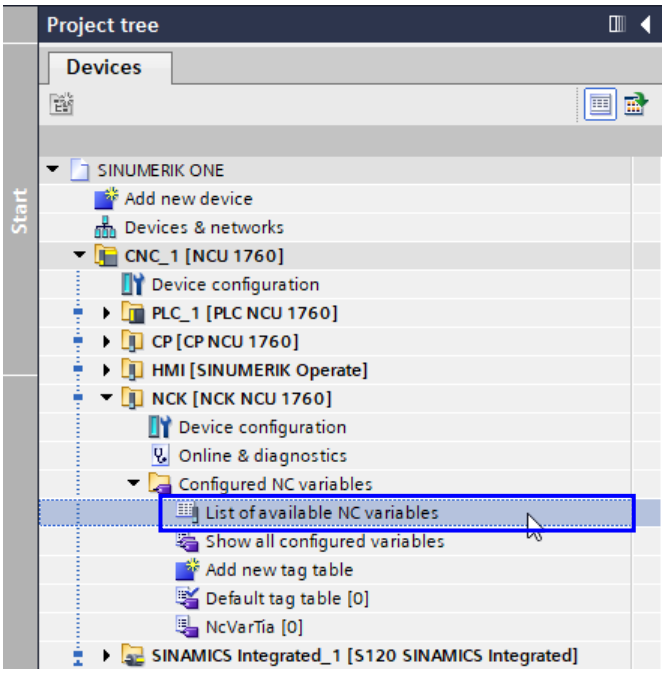
Notes:

Creating VAR-DB with TIA-Portal

Start a new NC tag table.



Select the NC variable database.



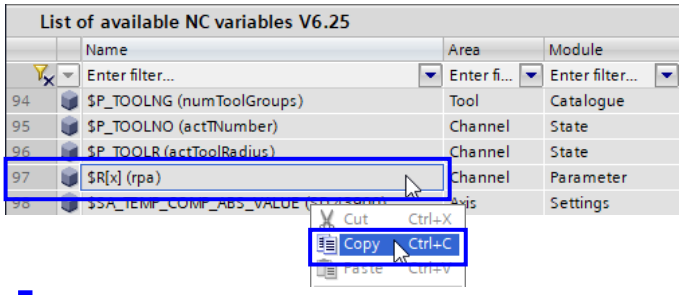
Upon database selection, a complete list of the available NC variables is displayed.

List of available NC variables V6.25

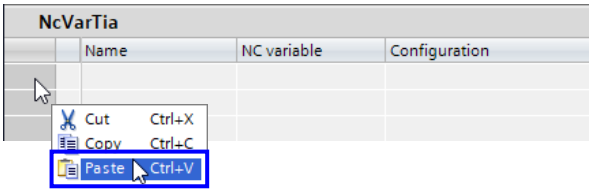
	Name	Area	Module	Block	Description	Data type	Access rights
	Enter filter...	Enter fi...	Enter filter...	Ent...	Enter filter...	Enter filt...	Enter filter...
1	$A_DBB[x] (aDbb)	NCK	State	S	PLC data byte	UINT_16	Read/write
2	$A_DBD[x] (aDbd)	NCK	State	S	PLC data doubleword	INTEGER	Read/write
3	$A_DBR[x] (aDbr)	NCK	State	S	PLC Real data (32 bits)	DOUBLE	Read/write
4	$A_DBW[x] (aDbw)	NCK	State	S	PLC data word (unsigned)	UINT_16	Read/write
5	$A_MYMLN (toolMyPlace)	Tool	Data	TD	Owner magazine location of the tool	UINT_16	Read
6	$A_MYMN (toolMyMag)	Tool	Data	TD	Owner magazine of the tool	UINT_16	Read
7	$A_TOOLADAPT (adaptNo)	Tool	Data	TD	Adapter number at the magazine location of tool	UINT_16	Read
8	$A_TOOLMLN[x] (toolInPlace)	Tool	Data	TD	Current magazine location of the tool	UINT_16	Read
9	$A_TOOLMN[x] (toolInMag)	Tool	Data	TD	Magazine in which the tool is currently located	UINT_16	Read
10	$A_TOOLMTADAPT (adaptNoOnMultitool)	Tool	Data	TD	Adapter number at the multitool loc. of the tool	UINT_16	Read
11	$A_TOOLMTLN[x] (toolInMultitoolPlace)	Tool	Data	TD	Multitool location number of the tool	UINT_16	Read
12	$A_TOOLMTN[x] (toolInMultitool)	Tool	Data	TD	Multitool in which the tool is located.	UINT_16	Read
13	$A_USEDD (resultCuttingEdgeNrUsed)	Tool	Find	TF	D numbers of cutting edges since last count	INTEGER	Read
14	$A_USEDND (resultNrOfCutEdgesUsed)	Tool	Find	TF	Number of cutting edges since last workpiece count	INTEGER	Read
15	$A_USEDT (resultToolNrUsed)	Tool	Find	TF	Tool numbers T since the last count	INTEGER	Read
16	$AA_ETRANS[x] (linShift)	Channel	ExternFrame	FE	Translation of an external frame	DOUBLE	Read/write
17	$AA_IB[x] (cmdToolBasePos)	Channel	MachineAxis	SMA	Pos. setpoint in machine coordinate system	DOUBLE	Read
18	$AA_IBN[x] (actToolBasPosBN)	Channel	GeometricAxis	SEGA	Current setpoint relative to the BZS	DOUBLE	Read
19	$AA_IEN[x] (actToolBasPosEN)	Channel	GeometricAxis	SEGA	Current setpoint in the SZS coordinate system	DOUBLE	Read
20	$AA_IM[x] (actToolBasePos)	Channel	MachineAxis	SMA	Current setpoint of the axis position in the MCS	DOUBLE	Read
21	$AA_IW[x] (actToolEdgeCenterPos)	Channel	GeometricAxis	SGA	Center point of a cutting edge	DOUBLE	Read
22	$AA_LOAD[x] (aaLoad)	Channel	MachineAxis	SEMA	Drive load in %	DOUBLE	Read
23	$AA_POWER[x] (aaPower)	Channel	MachineAxis	SEMA	Drive power in W (611D only)	DOUBLE	Read
24	$AA_S[x] (actSpeed)	Channel	LogicalSpindle	SSP2	Actual value of the spindle speed	DOUBLE	Read

Notes:

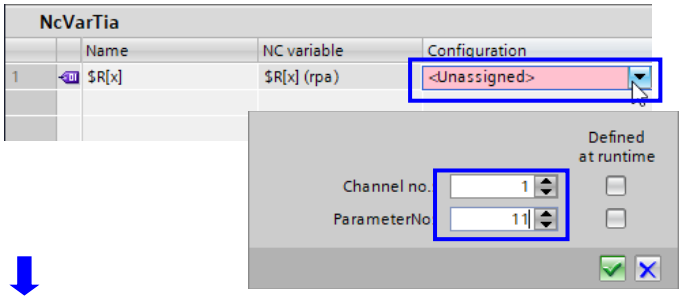
Search the variables by means of the filter selection and copy with the right mouse key.



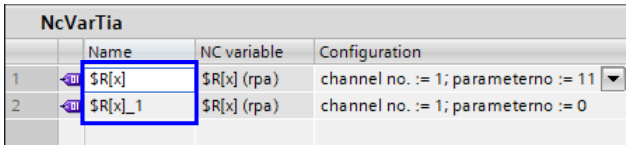
Insert the variables to the NC tag table with the right mouse key.



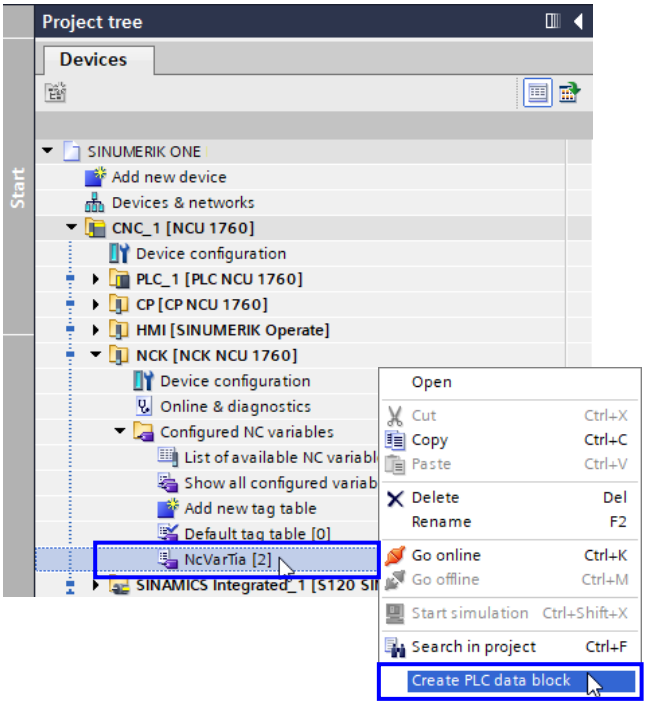
In some cases, an input screen is displayed in which you can make further parameterizations, e.g. channel / axis number, etc. Enter an R parameter with the value 11 and another R parameter with the value 0 in the line field.



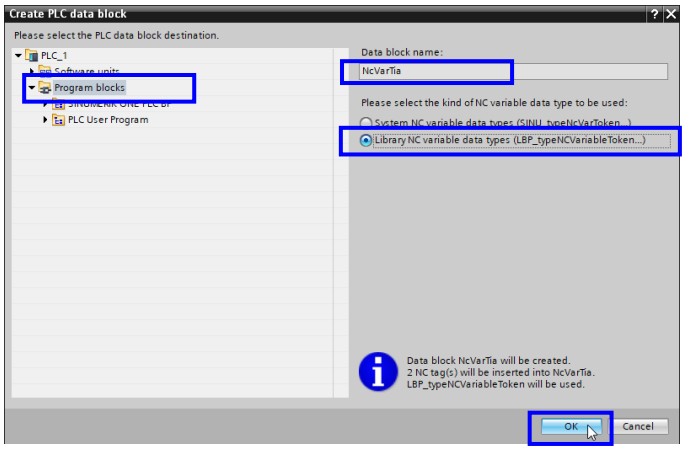
In the following screenshot, two R parameters are used as an example.
The first with fixed parameterization, not parameterizing with zero, and the second using parameterized index zero.
It is therefore possible to manipulate the index of the R parameter in the user PLC program.



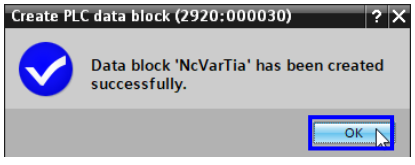
When the user has completed the selection of the necessary variables, the user variable table can be saved and transferred to a user DB.



You can decide here on the destination folder and name for the data block, as well as the NC variable data type to be used.

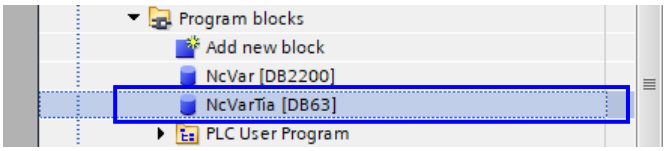


Confirm with OK.

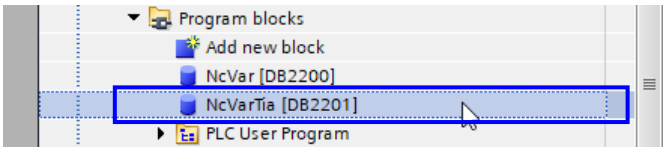


Notes:

The created data block is included in your TIA project.



The block number can be changed in properties.



Content of TIA VAR DB with STRUCT information.

NcVarTia

	Name	Data type	Offset	Start value
1	Static			
2	$R[x]	"LBP_typeNCVar...	0.0	
3	token	"LBP_typeNCVariabl...	0.0	
4	syntaxId	Byte	0.0	16#82
5	areaAndUnit	Byte	1.0	16#41
6	column	Word	2.0	16#1
7	row	Word	4.0	16#B
8	blockType	Byte	6.0	16#15
9	numberOfRows	Byte	7.0	16#1
10	dataTypeCode	Byte	8.0	16#F
11	length	Byte	9.0	16#8
12	value	LReal	10.0	0.0
13	$R[x]_1	"LBP_typeNCVariabl...	18.0	
14	token	"LBP_typeNCVariabl...	18.0	
15	syntaxId	Byte	18.0	16#82
16	areaAndUnit	Byte	19.0	16#41
17	column	Word	20.0	16#1
18	row	Word	22.0	16#0
19	blockType	Byte	24.0	16#15
20	numberOfRows	Byte	25.0	16#1
21	dataTypeCode	Byte	26.0	16#F
22	length	Byte	27.0	16#8
23	value	LReal	28.0	0.0

After being successfully compiled in TIA, the TIA VAR DB can be used as a parameter value for the FB2 or FB3.

Notes:

The call and parameterization of the FB2 and FB3 can now take place according to the following parameter description:

Signal	Type	Format	Range of values	Remark
Req	I	BOOL		Job start with positive signal edge
NumVar	I	INT	1 ... 8	Number of variables to be read: ”Addr1” to ”Addr8”
Addr1 to Addr8	I	VARI-ANT	[DBName]. [S7-Name]	Variable identifiers from **NC Var selector**
Unit1 to Unit8	I	BYTE		Area address, optional for variable addressing
Column1 to Column8	I	WORD		Column address, optional for variable addressing
Line1 to Line8	I	WORD		Line address, optional for variable addressing
Error	Q	BOOL		Negative acknowledgment of job or execution of job impossible
NDR	Q	BOOL		Job successfully executed Data are available
State	Q	WORD		See error identifiers
RD1 to RD8	I/Q	VARI-ANT		Target area for read data

Parameter description of **FB2**

FB2
Error codes

State		Meaning	Note
H-Byte 1)	L-Byte		
1-8	1	Access error	
0	2	Error in job	Incorrect compilation of variables in a job
0	3	Negative acknowledgement, job not executable	Internal error, possible remedy: ● Check job data ● NC-Reset ● fewer string variables in one call
1-8	4	Insufficient local user memory available	Data type of the read variable is greater than specified in "RD1" - "RD8"
1-8	5	Format conversion error	Error on conversion of var. type
0	6	FIFO full	Job must be repeated since queue is full
1-8	10	Error in variable addressing	"Unit" or "Column"/"Line" contains value 0
0	11	Address of variable invalid	Check "Addr" (or variable name), "Area", "Unit"
0	12	NumVar == 0	Check parameter "NumVar"

1) High-Byte > 0 → Number of the variable in which the error occurred

Notes:

Parameter description of **FB3**

Signal	Type	Format	Range of values	Remark
Req	I	BOOL		Job start with positive signal edge
NumVar	I	INT	1 ... 8	Number of variables to be written: "Addr1" to "Addr8"
Addr1 to Addr8	I	VARI-ANT	[DBName]. [S7-Name]	Variable identifiers from **NC Var selector**
Unit 1 to Unit 8	I	BYTE		Area address, optional for variable addressing
Column 1 to Column 8	I	WORD		Column address, optional for variable addressing
Line 1 to Line 8	I	WORD		Line address, optional for variable addressing
Error	Q	BOOL		Negative acknowledgment of job or execution of job impossible
Done	Q	BOOL		Job successfully executed
State	Q	WORD		See error identifiers
SD1 to SD8	I/Q	VARI-ANT		Data to be written

FB3
Error codes

State		Meaning	Note
H-Byte 1)	L-Byte		
1-8	1	Access error	
0	2	Error in job	Incorrect compilation of variables in a job
0	3	Negative acknowledgement, job not executable	Internal error, possible remedy: ● Check job data ● NC-Reset
1-8	4	Data areas or data types do not match or string is empty	Check data in "SD1" - "SD8"
1-8	5	Format conversion error	Error on conversion of var. type
0	6	FIFO full	Job must be repeated since queue is full
1-8	10	Error in variable addressing	"Unit" or "Column"/"Line" contains value 0
0	11	Variable address invalid or variable is read-only	Check "Addr" (or variable name), "Area", "Unit"
0	12	NumVar == 0	Check parameter "NumVar"
1-8	15	User data too long	Remedy: Write fewer variables per job or use shorter string variables
1) High-Byte > 0 → Number of the variable in which the error occurred			

Notes:

Content of VAR DB with STRUCT information.

NcVar

	Name	Data type	Offset	Start value
1	Static			
2	C1_RP_rpa11_1	Struct	0.0	
3	SYNTAX_ID	Byte	0.0	BYTE#16#82
4	Area_Unit	Byte	1.0	BYTE#16#41
5	Column	Word	2.0	WORD#16#0001
6	Line	Word	4.0	WORD#16#000B
7	BlockType	Byte	6.0	BYTE#16#15
8	NumOfLine	Byte	7.0	BYTE#16#01
9	DataType	Byte	8.0	BYTE#16#0F
10	Length	Byte	9.0	BYTE#16#08
11	C1_RP_rpa0_1	Struct	10.0	
12	SYNTAX_ID	Byte	10.0	BYTE#16#82
13	Area_Unit	Byte	11.0	BYTE#16#41
14	Column	Word	12.0	WORD#16#0001
15	Line	Word	14.0	16#0
16	BlockType	Byte	16.0	BYTE#16#15
17	NumOfLine	Byte	17.0	BYTE#16#01
18	DataType	Byte	18.0	BYTE#16#0F
19	Length	Byte	19.0	BYTE#16#08

For comparison, VAR DB created with the “NC VAR selector” tool.

Content of TIA VAR DB with STRUCT information.

NcVarTia

	Name	Data type	Offset	Start value
1	Static			
2	$R[x]	"LBP_typeNCVar...	0.0	
3	token	"LBP_typeNCVariabl...	0.0	
4	syntaxId	Byte	0.0	16#82
5	areaAndUnit	Byte	1.0	16#41
6	column	Word	2.0	16#1
7	row	Word	4.0	16#B
8	blockType	Byte	6.0	16#15
9	numberOfRows	Byte	7.0	16#1
10	dataTypeCode	Byte	8.0	16#F
11	length	Byte	9.0	16#8
12	value	LReal	10.0	0.0
13	$R[x]_1	"LBP_typeNCVariabl...	18.0	
14	token	"LBP_typeNCVariabl...	18.0	
15	syntaxId	Byte	18.0	16#82
16	areaAndUnit	Byte	19.0	16#41
17	column	Word	20.0	16#1
18	row	Word	22.0	16#0
19	blockType	Byte	24.0	16#15
20	numberOfRows	Byte	25.0	16#1
21	dataTypeCode	Byte	26.0	16#F
22	length	Byte	27.0	16#8
23	value	LReal	28.0	0.0

The DATA DB is also created and used for the examples.

NcData

	Name	Data type	Offset	Start value
1	Static			
2	L_IN	String[32]	0.0	'SUNRISE '
3	GUD_GJ	String[8]	34.0	'__GJ'
4	Path	String[32]	44.0	'/_N_CMA_DIR'
5	PName	String[32]	78.0	'_N_PLCASUP1_SPF'
6	RpaValue	LReal	112.0	0.0
7	GudToken	Struct	120.0	
8	SYNTAX_ID	Byte	120.0	16#0
9	Area_Unit	Byte	121.0	16#0
10	Column	Word	122.0	16#0
11	Line	Word	124.0	16#0
12	BlockType	Byte	126.0	16#0
13	NumOfLine	Byte	127.0	16#0
14	DataType	Byte	128.0	16#0
15	Length	Byte	129.0	16#0
16	GudValue	DInt	130.0	0

Note:

To be able to read an R parameter of the type DOUBLE without adapting the format, a variable of the LREAL type must be specified in the PLC.

Example: Reading and writing NC variables.

```
// Example "PLCFunctions" [FB140]

// Variables for "ReadVar" and "WriteVar"

// Input
//    readVar : Bool;
//    writeVar : Bool;

// Static
//    statReadVarFlag : Bool;
//    statReadVarReq : Bool;
//    statWriteVarFlag : Bool;
//    statWriteVarReq : Bool;

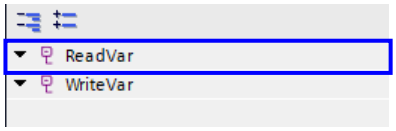
// Multi instances generated with the call
//    instLBP_ReadVar : "LBP_ReadVar";
//    instLBP_WriteVar : "LBP_WriteVar";

// Temp
//    tempReadVarEdgeP : Bool;
//    tempReadVarNDR : Bool;
//    tempReadVarError : Bool;
//    tempReadVarState : Word;
//    tempWriteVarEdgeP : Bool;
//    tempWriteVarDone : Bool;
//    tempWriteVarError : Bool;
//    tempWriteVarState : Word;

// Constant
//    None
```

Notes:

Example: Reading NC variables.




```
REGION ReadVar

    // Scan rising edge
    #tempReadVarEdgeP := #readVar AND NOT #statReadVarFlag;
    #statReadVarFlag := #readVar;

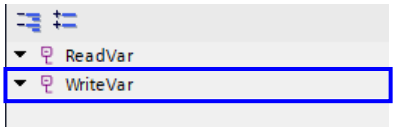
    // Set Job Start
    IF #tempReadVarEdgeP THEN
        #statReadVarReq := TRUE;
    END_IF;

    // Read rpa
    #instLBP_ReadVar(Req := #statReadVarReq,
                     NumVar := 1,
                     Addr1 := "NcVarTia"."$R[x]".token,
                     Addr2 := NULL,
                     Addr3 := NULL,
                     Addr4 := NULL,
                     Addr5 := NULL,
                     Addr6 := NULL,
                     Addr7 := NULL,
                     Addr8 := NULL,
                     Error => #tempReadVarError,
                     NDR => #tempReadVarNDR,
                     State => #tempReadVarState,
                     RD1 := "NcVarTia"."$R[x]".value,
                     RD2 := NULL,
                     RD3 := NULL,
                     RD4 := NULL,
                     RD5 := NULL,
                     RD6 := NULL,
                     RD7 := NULL,
                     RD8 := NULL);

    // Reset Job Start
    IF #tempReadVarNDR OR ("mcpIn".reset AND #tempReadVarError) THEN
        #statReadVarReq := FALSE;
    END_IF;

END_REGION
```

Example: Writing NC variables.




```
REGION WriteVar

    // Scan rising edge
    #tempWriteVarEdgeP := #writeVar AND NOT #statWriteVarFlag;
    #statWriteVarFlag := #writeVar;

    // Set Job Start
    IF #tempWriteVarEdgeP THEN
        #statWriteVarReq := TRUE;
    END_IF;

    // Write rpa
    #instLBP_WriteVar(Req := #statWriteVarReq,
                      NumVar := 1,
                      Addr1 := "NcVarTia"."$R[x]_1".token,
                      Line1 := 16#C,
                      Addr2 := NULL,
                      Addr3 := NULL,
                      Addr4 := NULL,
                      Addr5 := NULL,
                      Addr6 := NULL,
                      Addr7 := NULL,
                      Addr8 := NULL,
                      Error => #tempWriteVarError,
                      Done => #tempWriteVarDone,
                      State => #tempWriteVarState,
                      SD1 := "NcVarTia"."$R[x]".value,
                      SD2 := NULL,
                      SD3 := NULL,
                      SD4 := NULL,
                      SD5 := NULL,
                      SD6 := NULL,
                      SD7 := NULL,
                      SD8 := NULL);

    // Reset Job Start
    IF #tempWriteVarDone OR ("mcpIn".reset AND #tempWriteVarError) THEN
        #statWriteVarReq := FALSE;
    END_IF;

END_REGION
```

PI Services

Function description: Automate an Operator function

The following PI services are available:

PI service	Function
ASUP	Assign interrupt
CANCEL	Execute cancel
CONFIG	Reconfiguration of tagged machine data
DIGION	Digitizing on
DIGIOF	Digitizing off
FINDBL	Activate block search
LOGIN	Activate password
LOGOUT	Reset password
NCRES	Trigger NC-RESET
RETRAC	Select JOG retract mode
SELECT	Select program for processing for one channel
SETUDT	Sets the current user data to active

PI service	Function
SETUFR	Activate user frame
CRCEDN	Create new cutting edge
CREACE	Create cutting edge
CREATO	Generate tool
DELECE	Delete a cutting edge
DELETO	Delete tool
MMCSEM	Semaphores for various PI services
TMCRTO	Create tool
TMFDPL	Empty location search for loading
TMFPBP	Empty location search
TMGETT	T-number for the specified tool identifier with duplo number
TMMVTL	Prepare magazine location for loading, unload tool
TMPOSM	Position magazine location or tool
TMPCIT	Set increment value for workpiece counter
TMRASS	Reset active status
TRESMO	Reset monitoring values

PI service	Function
TSEARC	Complex search using search screen forms
TMCRMT	Create Multitool
TMDLMT	Delete Multitool
POSMT	Position Multitool
FDPLMT	Empty location search for Multitools

Function Description: cont.

PI services are typically mimics of Operator actions which can be carried out from the PLC.

A typical example is the setting and the resetting of the password, this is an action which is often carried out by the operator but can also be automated using the PLC program.

It is therefore possible to automate the deactivation of the password at system start-up,

Prerequisites for the Function:

HMI Requirement:
None

Drive requirement (SINAMICS):
None

NCK Requirement:
Dependant upon function

PLC Requirement:
Parameterization of FB7 in the PLC

Notes:

Steps to achieve function:

Parameter description of **FB7**

Signal	Type	Format	Range of values	Remark
Req	I	BOOL		Job request
PIServic e	I	VARI-ANT	[LBP_PIServices] .[PI-Service]	PI service description
Unit	I	INT	1, 2, 3 … 10	Area number
Addr1 to Addr4	I	VARI-ANT	[DBName]. [VarName]	Reference to strings specification accord-ing to selected PI service
WVar1 to WVar16	I	WORD	-32768...32767 8000H...7FFFH	INTEGER or WORD variables. Specifica-tion according to selected PI service,
Error	Q	BOOL		Negative acknowl-edgment of job or execution of job impossible
Done	Q	BOOL		Job successfully executed
State	Q	WORD		See error identifiers

Error codes

State	Meaning	Note
3	Negative acknowledg-ment, job not executable	Internal error, possible remedy: NC RESET
4	Variable type	Check datatype of the parameters Addr1 to Addr4
5	Copying fault	Check parameter
6	FIFO full	Job must be repeated since queue is full
12	PI Service undefined	String data required has not been parame-terized
14	PI Service is unknown	No valid PI description
15	String too long	Check strings Addr1.. Addr4

In order to use the LOGIN PI service, a string variable has to be available as parameter source.
The string variable is the name of the Password, it is con-tained in a user DB using data type STRING.

STRING[32] := `SUNRISE ` **NOTE 8 Characters - use SPACES**

DATA DB with L_IN string.

NcData

	Name	Data type	Offset	Start value
1	Static			
2	L_IN	String[32]	0.0	'SUNRISE '
3	GUD_GJ	String[8]	34.0	'__GJ'
4	Path	String[32]	44.0	'/_N_CMA_DIR'
5	PName	String[32]	78.0	'_N_PLCASUP1_SPF'
6	RpaValue	LReal	112.0	0.0
7	GudToken	Struct	120.0	
8	GudValue	DInt	130.0	0

Example of Password Login and Logout

```
// Example "PLCFunctions" [FB140] - continued

// Variables for "Logout" and "Login"

// Input
//    logout : Bool;
//    login : Bool;

// Static
//    statLogoutFlag : Bool;
//    statLogoutReq : Bool;
//    statLoginFlag : Bool;
//    statLoginReq : Bool;

// Multi instances generated with the call
//    instLBP_ReqPIServiceLogout : "LBP_ReqPIService";
//    instLBP_ReqPIServiceLogin : "LBP_ReqPIService";

// Temp
//    tempLogoutEdgeP : Bool;
//    tempLogoutDone : Bool;
//    tempLogoutError : Bool;
//    tempLogoutState : Word;
//    tempLoginEdgeP : Bool;
//    tempLoginDone : Bool;
//    tempLoginError : Bool;
//    tempLoginState : Word;

// Constant
//    None
```

Notes:

Example of Password Logout




```
REGION "Logout"

    // Scan rising edge
    #tempLogoutEdgeP := #logout AND NOT #statLogoutFlag;
    #statLogoutFlag := #logout;

    // Set Job Start
    IF #tempLogoutEdgeP THEN
        #statLogoutReq := TRUE;
    END_IF;

    // PI Logout
    #instLBP_ReqPIServiceLogout(PIService := "LBP_PIServices".LOGOUT,
                                Req := #statLogoutReq,
                                Unit := 1,
                                Addr1 := NULL,
                                Addr2 := NULL,
                                Addr3 := NULL,
                                Addr4 := NULL,
                                Error => #tempLogoutError,
                                Done => #tempLogoutDone,
                                State => #tempLogoutState);

    // Reset Job Start
    IF #tempLogoutDone OR ("mcpIn".reset AND #tempLogoutError) THEN
        #statLogoutReq := FALSE;
    END_IF;

END_REGION
```

Example of Password Login




```
REGION "Login"

    // Scan rising edge
    #tempLoginEdgeP := #login AND NOT #statLoginFlag;
    #statLoginFlag := #login;

    // Set Job Start
    IF #tempLoginEdgeP THEN
        #statLoginReq := TRUE;
    END_IF;

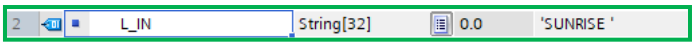
    // PI Login
    #instLBP_ReqPIServiceLogin(PIService := "LBP_PIServices".LOGIN,
                               Req := #statLoginReq,
                               Unit := 1,
                               Addr1 := "NcData".L_IN,
                               Addr2 := NULL,
                               Addr3 := NULL,
                               Addr4 := NULL,
                               Error => #tempLoginError,
                               Done => #tempLoginDone,
                               State => #tempLoginState);

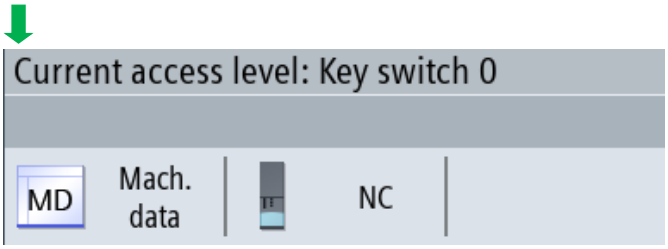
    // Reset Job Start
    IF #tempLoginDone OR ("mcpIn".reset AND #tempLoginError) THEN
        #statLoginReq := FALSE;
    END_IF;

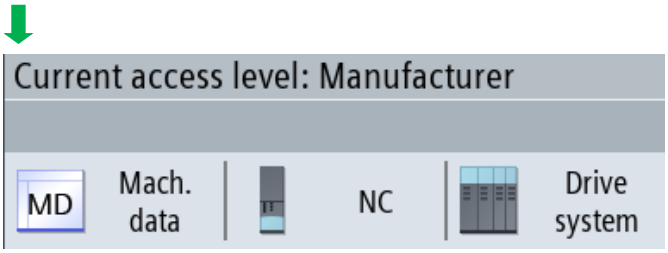
END_REGION
```

Notes:

Parameter part for Addr1 -- PI Login







Reading Global User data

Function Description:

Global User Data can be both read and written to by the automation system (PLC).

Global User Data are often referred to as GUDs . The OEM is able to define his own user variables in this way.
Dependant upon the OEM functionality of the relevant GUD it may be necessary in process to read the current value to the PLC or to write the value from the PLC.

With the function block FB5 you are able to read the value of a GUD.

Because GUDs are generated by the OEM (user), it is not possible to obtain the internal address of the variable with the NC Var selector in order to write the GUD.

By reading the content of the variable with FB5 and with additional parameterization of the FB5 the internal address can be obtained as a result parameter of the FB5, this result / output parameter can be used as a destination parameter with the FB3 allowing an indirect way to write the GUDs.

Even when reading repeatedly, the use of the internal address supplied by FB5 in conjunction with FB2 is useful, as it is more efficient.

Prerequisites for the Function:

HMI Requirement:
None

Drive requirement (SINAMICS):
None

NCK Requirement:
GUD Data available

PLC Requirement:
Parameterization of FB5 in the PLC

Notes:

Steps to achieve function:

Parameter description of **FB5**

Signal	Type	Format	Value range	Comment
Req	I	BOOL		Job start with positive signal edge
Addr	I	VARI-ANT	[DBName]. [VarName]	GUD variable name in a variable of data type STRING
Area	I	BYTE	0, 2	Area address: 0: NCK variables 2: Channel variables
Unit	I	BYTE	1, 2, .. 10	NCK area: Unit:=1 Channel area: Channel no.
Index1	I	INT	0, 1, 2, .. <max. field index>	Field index 1 of variable Variable has the value 0 if no field index is used.
Index2	I	INT	0, 1, 2, .. <max. field index>	Field index 2 of variable Variable has the value 0 if no field index is used.

Signal	Type	Format	Value range	Comment
CnvtToken	I	BOOL		Activate generation of a 10 byte variable token
VarToken	I	VARI-ANT	[DBName]. [S7-Name]	Address to a 10-byte token (see example)
Error	Q	BOOL		Negative acknowledgment of job or execution of job impossible
Done	Q	BOOL		Job successfully executed
State	Q	WORD		See error identifiers
RD	I/Q	VARI-ANT		Target area for read data, e.g. DB area or Marker of the same data type as the variable to be read

FB5
Error codes

State	Meaning	Note
1	Access error	
2	Error in job	Incorrect compilation of variables in a job
3	Negative acknowledgement, job not executable	Internal error, possible remedy. NC-Reset
4	Data areas or data types do not tally	Check data to be read in RD
4	Insufficient local user memory available	read variable is longer than specified in RD
5	Format conversion error	Adr parameter must be transferred as string
6	FIFO full	Job must be repeated since queue is full
10	Error in variable addressing	Unit contains value 0
11	Address of variable invalid	Address check (or variable name), "Area", "Unit"
15	String more than 32 characters	GUD variable name too long

Notes:

In order to read a GUD value, the OEM must first generate and activate the GUD file using the HMI function available in the System explorer.

The name of the GUD name has to be parameterized as a STRING variable to the function block, at parameter; Addr.
If the GUD variable is an array, the array index can be parameterized as well.

According to the description, field index refers to array index.

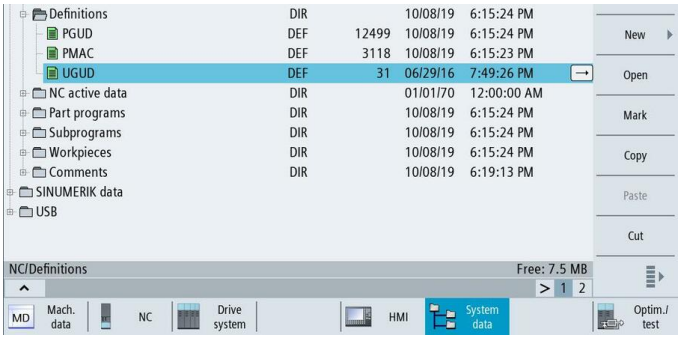
You can additionally use the parameters, to obtain the address of the GUD for parameterization with FB3 PUT

CnvtToken	Activate generation of a 10 byte variable token
VarToken	[DBName].[VarName] Address to a 10-byte token

Syntax in order to generate an NCK GUD array variable with the name __GJ.

```
NC/DEF.DIR/UGUD.DEF
N100 DEF NCK INT __GJ[10]¶
M30¶
```

Create and activate GUD file and variables



The variable in the NC viewed on HMI in “Global GUD”.

Global user variables	UGUD
__GJ[0]	1
__GJ[1]	2
__GJ[2]	3
__GJ[3]	4
__GJ[4]	5
__GJ[5]	0
__GJ[6]	0
__GJ[7]	0
__GJ[8]	0
__GJ[9]	0

DATA DB with GUD string

NcData

	Name	Data type	Offset	Start value
1	Static			
2	L_IN	String[32]	0.0	'SUNRISE '
3	GUD_GJ	String[8]	34.0	'__GJ'
4	Path	String[32]	44.0	'/_N_CMA_DIR'
5	PName	String[32]	78.0	'_N_PLCASUP1_SPF'
6	RpaValue	LReal	112.0	0.0
7	GudToken	Struct	120.0	
8	SYNTAX_ID	Byte	120.0	16#0
9	Area_Unit	Byte	121.0	16#0
10	Column	Word	122.0	16#0
11	Line	Word	124.0	16#0
12	BlockType	Byte	126.0	16#0
13	NumOfLine	Byte	127.0	16#0
14	DataType	Byte	128.0	16#0
15	Length	Byte	129.0	16#0
16	GudValue	DInt	130.0	0

Example: Reading and writing GUD parameter.

```
// Example "PLCFunctions" [FB140] - continued

// Variables for "ReadGud" and "WriteGud"

// Input
//    readGud : Bool;
//    writeGud : Bool;

// Static
//    statReadGudFlag : Bool;
//    statReadGudReq : Bool;
//    statWriteGudFlag : Bool;
//    statWriteGudReq : Bool;

// Multi instances generated with the call
//    instLBP_ReadGud : "LBP_ReadGUD";
//    instLBP_WriteGud : "LBP_WriteVar";

// Temp
//    tempReadGudEdgeP : Bool;
//    tempReadGudDone : Bool;
//    tempReadGudError : Bool;
//    tempReadGudState : Word;
//    tempWriteGudEdgeP : Bool;
//    tempWriteGudDone : Bool;
//    tempWriteGudError : Bool;
//    tempWriteGudState : Word;

// Constant
//    None
```

Notes:

Example: Reading GUD parameter.




```
REGION ReadGud

    // Scan rising edge
    #tempReadGudEdgeP := #readGud AND NOT #statReadGudFlag;
    #statReadGudFlag := #readGud;

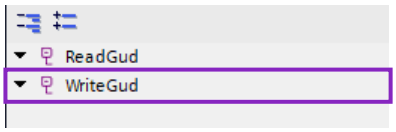
    // Set Job Start
    IF #tempReadGudEdgeP THEN
        #statReadGudReq := TRUE;
    END_IF;

    // Read Gud
    #instLBP_ReadGud(Req := #statReadGudReq,
                     Addr := "NcData".GUD_GJ,
                     Area := 16#0,
                     Unit := 16#1,
                     Index1 := 2, //__GJ[2]
                     Index2 := 0,
                     CnvtToken := TRUE,
                     VarToken := "NcData".GudToken,
                     Error => #tempReadGudError,
                     Done => #tempReadGudDone,
                     State => #tempReadGudState,
                     RD := "NcData".GudValue);

    // Reset Job Start
    IF #tempReadGudDone OR ("mcpIn".reset AND #tempReadGudError) THEN
        #statReadGudReq := FALSE;
    END_IF;

END_REGION
```

Example: Writing GUD parameter.




```
REGION WriteGud

    // Scan rising edge
    #tempWriteGudEdgeP := #writeGud AND NOT #statWriteGudFlag;
    #statWriteGudFlag := #writeGud;

    // Set Job Start
    IF #tempWriteGudEdgeP THEN
        #statWriteGudReq := TRUE;
    END_IF;

    // Write Gud
    #instLBP_WriteGud(Req := #statWriteGudReq,
                      NumVar := 1,
                      Addr1 := "NcData".GudToken,
                      Line1 := 16#4, //__GJ[3]
                      Addr2 := NULL,
                      Addr3 := NULL,
                      Addr4 := NULL,
                      Addr5 := NULL,
                      Addr6 := NULL,
                      Addr7 := NULL,
                      Addr8 := NULL,
                      Error => #tempWriteGudError,
                      Done => #tempWriteGudDone,
                      State => #tempWriteGudState,
                      SD1 := "NcData".GudValue,
                      SD2 := NULL,
                      SD3 := NULL,
                      SD4 := NULL,
                      SD5 := NULL,
                      SD6 := NULL,
                      SD7 := NULL,
                      SD8 := NULL);

    // Reset Job Start
    IF #tempWriteGudDone OR ("mcpIn".reset AND #tempWriteGudError) THEN
        #statWriteGudReq := FALSE;
    END_IF;

END_REGION
```

Notes:

Starting ASUP

Function Description:

An ASUP (Asynchronous Subroutine) is an NC Subroutine written normally by the OEM, this subroutine is started from the automation system (PLC). If a user NC Program is currently active, it will be interrupted according to the parameterization of the PLC user program.

The ASUP function is carried out in two steps:

1) ASUP Preparation
   PI Service ASUP FB7
   Only possible in NCK Reset State

2) ASUP Start
   Parameterization and call of FC9

Prerequisites for the Function:

Software option M43 “Multiple mode actions”
(only necessary for ASUP in JOG mode)

MD11602 $MN_ASUP_START_MASK =1H
MD11604 $MN_ASUP_START_PRIO_LEVEL =100

HMI Requirement:
None

Drive requirement (SINAMICS):
None

NCK Requirement:
ASUP available

PLC Requirement:
Parameterization of FB7 and FC9 in the PLC

Steps to achieve function:

Parameter description of **FC9**

Signal	Type	Format	Value range	Remark
Start	I	BOOL		
ChanNo	I	INT	1 ... 10	No. of the NC channel
IntNo	I	INT	1 ... 8	Interrupt No.
Activ	Q	BOOL		1 = Active
Done	Q	BOOL		1 = ASUP completed
Error	Q	BOOL		1 = Interrupt switched off
StartErr	Q	BOOL		1 = Interrupt number not assigned or deleted
Ref	I/Q	WORD	Global variable (MW, DBW,..)	1 word per FC 9 (for internal use)

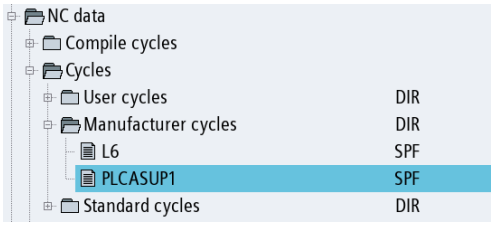
Parameter description of **FB7 “PI“ ASUP**

Signal	Format	Value range	Remark
PIService	VARIANT	"LBP_PIServices".ASUP	Assign interrupt
Unit	INT	1, 2, 3 .. 10	Channel
Addr1 Addr2	VARIANT		Path name Program name
WVar1: WVar2: WVar3: WVar4:	WORD	1, 2, 3 .. 8 1, 2, 3 .. 8 0, 1 0, 1	Interrupt-number Priority LIFTFAST BLSYNC

Prior to calling FC9, the ASUP has to be initialized with the PI service ASUP.
The following example shows the initialization and the call of the ASUP with FC9.
The PI service PI.ASUP requires two strings to be defined in a user DB, the strings being the “Path“ and the name of the ASUP.

Notes:

Create ASUP in the Manufacturer directory.



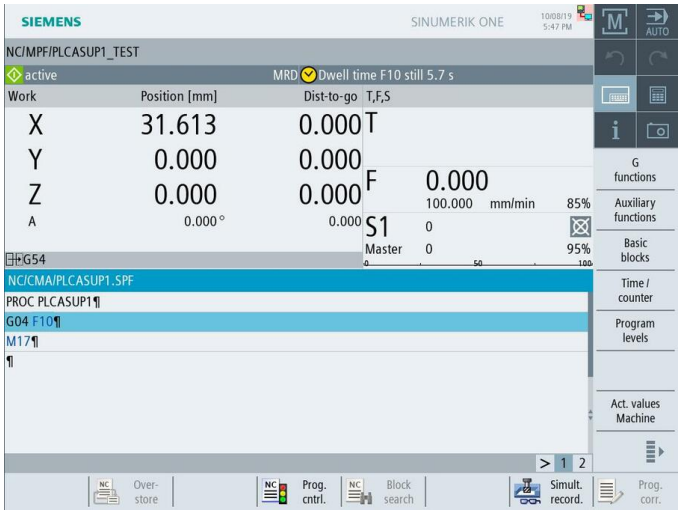
Content of PLC ASUP

```
NC/CMA.DIR/PLCASUP1.SPF
PROC PLCASUP1¶
G04 F10¶
M17¶
```

The following NC Program block will be interrupted:

```
NC/MPF/PLCASUP1_TEST
N10 G01 G91 G94 X100 F100¶
N20 X20¶
N30 M30¶
```

The Interruption can now be seen:



DATA DB with ASUP strings.

NcData

	Name	Data type	Offset	Start value
1	Static			
2	L_IN	String[32]	0.0	'SUNRISE '
3	GUD_GJ	String[8]	34.0	'_GJ'
4	Path	String[32]	44.0	'/_N_CMA_DIR'
5	PName	String[32]	78.0	'_N_PLCASUP1_SPF'
6	RpaValue	LReal	112.0	0.0
7	GudToken	Struct	120.0	
8	GudValue	DInt	130.0	0

Example of initialization and call of ASUP

```
// Example "PLCFunctions" [FB140] - continued

// Variables for "AsupInit" and "AsupStart"

// Input
//    asupInit : Bool;
//    asupStart : Bool;

// Static
//    statAsupInitFlag : Bool;
//    statAsupInitReq : Bool;
//    statAsupFlag : Bool;
//    statAsupReq : Bool;
//    statAsupRef : Word;

// Multi instances generated with the call
//    instLBP_ReqPIServiceAsup : "LBP_ReqPIService";

// Temp
//    tempAsupInitEdgeP : Bool;
//    tempAsupInitDone : Bool;
//    tempAsupInitError : Bool;
//    tempAsupInitState : Word;
//    tempAsupEdgeP : Bool;
//    tempAsupActive : Bool;
//    tempAsupDone : Bool;
//    tempAsupError : Bool;
//    tempAsupStartError : Bool;

// Constant
//    None
```

Notes:

Example of initialization of ASUP




```
REGION AsupInit

    // Scan rising edge
    #tempAsupInitEdgeP := #asupInit AND NOT #statAsupInitFlag;
    #statAsupInitFlag := #asupInit;

    // Set Job Start
    IF #tempAsupInitEdgeP THEN
        #statAsupInitReq := TRUE;
    END_IF;

    // PI Asup
    #instLBP_ReqPIServiceAsup(Req := #statAsupInitReq,
                              PIService := "LBP_PIServices".ASUP,
                              Unit := 1,
                              Addr1 := "NcData".Path,
                              Addr2 := "NcData".PName,
                              Addr3 := NULL,
                              Addr4 := NULL,
                              WVar1 := 16#1,
                              WVar2 := 16#1,
                              Error => #tempAsupInitError,
                              Done => #tempAsupInitDone,
                              State => #tempAsupInitState);

    // Reset Job Start
    IF #tempAsupInitDone OR ("mcpIn".reset AND #tempAsupInitError) THEN
        #statAsupInitReq := FALSE;
    END_IF;

END_REGION
```

Example of call of ASUP




```
REGION AsupStart

    // Scan rising edge
    #tempAsupEdgeP := #asupStart AND NOT #statAsupFlag;
    #statAsupFlag := #asupStart;

    // Set Job Start
    IF #tempAsupEdgeP THEN
        #statAsupReq := TRUE;
    END_IF;

    // Asup Start
    "LBP_ReqASUP"(Start := #statAsupReq,
                  ChanNo := 1,
                  IntNo := 1,
                  Activ => #tempAsupActive,
                  Done => #tempAsupDone,
                  Error => #tempAsupError,
                  StartErr => #tempAsupStartError,
                  "Ref" := #statAsupRef);

    // Reset Job Start
    IF #tempAsupDone OR ("mcpIn".reset AND (#tempAsupError OR
        #tempAsupStartError)) THEN
        #statAsupReq := FALSE;
    END_IF;

END_REGION
```

Notes:

PLC controlled axis

Function Description:

With axis / spindle control it is possible to move an axis or a spindle from the automation system.

The following is an example of what can be carried out with the block:

Position spindle

Rotate spindle

Oscillate spindle

Indexing axes

Positioning axes

Prerequisites for the Function:

The axis has to be defined as a concurrent positioning axis in the NC machine data. MD30460 Bit 5.

HMI Requirement:
None

Drive requirement (SINAMICS):
None

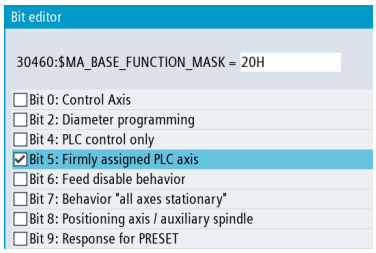
NCK Requirement:
Concurrent positioning axis

PLC Requirement:
Parameterization of the FC18 in the PLC

Function
Prior to commissioning a PLC axis it is necessary to declare the axis as a PLC axis. It is possible to do this permanently via Axis machine data 30460 or using interface signals to switch between PLC/NC control.

A geometry axis cannot be assigned as a PLC axis.
Firmly assigning the axis to the PLC via machine data:

Axis
MD



Parameter description of **FC18**

Signal	Type	Format	Range of values	Remark
Start	I	BOOL		Start the function
Stop	I	BOOL		Stop the functions 2,3,9,A,B
Funct	I	BYTE	1 … 11	1: Position spindle 2: Rotate spindle 3: Oscillate spindle 4: Indexing axis 5: Positioning axis metric 6: Positioning axis inch 7: PosAxis metric with hand-wheel override 8: PosAxis inch with hand-wheel override 9: Rotate spindle with automatic gear stage selection 10: Rotate spindle with constant cutting rate (m/min) 11: Rotate spindle with constant cutting rate (feet/min)

Signal	Type	Format	Range of values	Remark
Mode	I	BYTE	0 ... 5	0: Pos to absolute pos 1: Pos incrementally 2: Pos shortest path 3: Pos absolute, positive approach direction 4: Pos absolute, negative approach direction 5: Rotational direction as for M4
AxisNo	I	INT	1 ... 31	No. of axis/spindle to be traversed
Pos	I	REAL	0,1469368 I -38 to 0,1701412 I +39	Rotary axis: Degrees Indexing axis: Indexing position Linear axis: mm or inches
FRate	I	REAL	0,1469368 I -38 to 0,1701412 I +39	Rotary axis and spindle: rev/min See under table containing info about FRate
InPos	Q	BOOL		1 = Position reached, or function executed
Error	Q	BOOL		1 = error
State	Q	BYTE	0 ... 255	Error code

Notes:

Error codes

State	Meaning
100	Incorrect position programmed for axis/spindle
101	Programmed speed is too high
102	Incorrect value range for constant cutting rate
104	Following spindle: Illegal programming
105	No measuring system available
106	Axis positioning still active
107	Reference mark not found
108	No transition from speed control to position control
109	Reference mark not found
110	Velocity/speed is negative
111	Setpoint speed = 0
112	Invalid gear stage
115	Programmed position has not been reached
117/118	G96/G961 is not/still active in the NC

State	Meaning
120	Axis is not an indexing axis
121	Indexing position error
125	DC (shortest distance) not possible
126/127	Absolute value minus/plus not possible
128	No transverse axis available for diameter programming
130/131	Software limit switch plus/minus
132/133	Working area limit plus/minus
134	Frame not permitted for indexing axis
135	Indexing axis with Hirth joint is active
136	Indexing axis with Hirth joint is active and axis not referenced
137	Spindle operation not possible for transformed spindle/axis
138/139	Axis: Coordinate system-specific working area plus/minus violated
200	System error

Positioning axis

The following example shows how a PLC axis is positioned using an incremental path specified by FC18.

Example of positioning axis

```
// Example "PLCFunctions" [FB140] - continued

// Variables for "PosAxis"

// Input
//    posAx : Bool;

// Output
//    None

// InOut
//    None

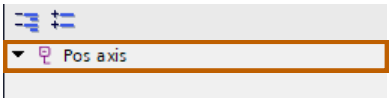
// Static
//    statPosAxFlag : Bool;
//    statPosAxStart : Bool;

// Temp
//    tempPosAxEdgeP : Bool;
//    tempPosAxInPos : Bool;
//    tempPosAxError : Bool;
//    tempPosAxState : Byte;

// Constant
//    None
```

Notes:

Example of positioning axis




```
REGION Pos axis

    // Scan rising edge
    #tempPosAxEdgeP := #posAx AND NOT #statPosAxFlag;
    #statPosAxFlag := #posAx;

    // Set Job Start
    IF #tempPosAxEdgeP THEN
        #statPosAxStart := TRUE;
    END_IF;

    // Pos axis start
    "LBP_CtrlAxisSpindle"(Start := #statPosAxStart,
                          Stop := FALSE,
                          Funct := 16#5,
                          Mode := 16#1,
                          AxisNo := 5,
                          Pos := 30.0,
                          FRate := 1000.0,
                          InPos => #tempPosAxInPos,
                          Error => #tempPosAxError,
                          State => #tempPosAxState);

    // Reset Job Start
    IF #tempPosAxInPos OR ("mcpIn".reset AND #tempPosAxError) THEN
        #statPosAxStart := FALSE;
    END_IF;

END_REGION
```

The PLC axis now moves to the position entered in FC18.
(FC18 Parameter **Pos :=30.0)**



By the changing the value of FC18 it is possible to move the axis to varying positions, For example, the value entered here could be taken from the Tool Management interface and used to position a tool carousel in preparation for a tool change.

Notes:

Positioning using an index table

The number of indexing positions is entered via machine data:

General MD: 10900 $MN_INDEX_AX_LENGTH_POS_TAB_1
10920 $MN_INDEX_AX_LENGTH_POS_TAB_2

There are two index tables (Machine data arrays) available where index positions can be entered:

General MD: 10910 $MN_INDEX_AX_POS_TAB_1
10930 $MN_INDEX_AX_POS_TAB_2

The index array to be used is selected via Axis machine data:

Axis MD: 30500 $MN_INDEX_AX_ASSIGN_POS TAB

The following example shows how to position a PLC axis using the positions entered in an index table found in the “General Machine Data” area.

The 10 positions are divided equally into 36 degree increments

General MD

General machine data			
10900	$MN_INDEX_AX_LENGTH_POS_TAB_1	10	re
10910[0]	$MN_INDEX_AX_POS_TAB_1	0	mm|degr... re
10910[1]	$MN_INDEX_AX_POS_TAB_1	36	mm|degr... re
10910[2]	$MN_INDEX_AX_POS_TAB_1	72	mm|degr... re
10910[3]	$MN_INDEX_AX_POS_TAB_1	108	mm|degr... re
10910[4]	$MN_INDEX_AX_POS_TAB_1	144	mm|degr... re
10910[5]	$MN_INDEX_AX_POS_TAB_1	180	mm|degr... re
10910[6]	$MN_INDEX_AX_POS_TAB_1	216	mm|degr... re
10910[7]	$MN_INDEX_AX_POS_TAB_1	252	mm|degr... re
10910[8]	$MN_INDEX_AX_POS_TAB_1	288	mm|degr... re
10910[9]	$MN_INDEX_AX_POS_TAB_1	324	mm|degr... re

Axis MD

Axis machine data			AX5:MA1
30460	$MA_BASE_FUNCTION_MASK	20H	po
30465	$MA_AXIS_LANG_SUB_MASK	0H	po
30500	$MA_INDEX_AX_ASSIGN_POS_TAB	1	re

Positioning axes using an index table

The following example shows how a PLC axis is positioned using the positions entered in an index table.

Example of positioning axis

```
// Example "PLCFunctions" [FB140] - continued

// Variables for "PosAxis"

// Input
//    posAx : Bool;

// Output
//    None

// InOut
//    None

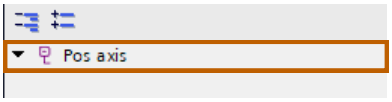
// Static
//    statPosAxFlag : Bool;
//    statPosAxStart : Bool;

// Temp
//    tempPosAxEdgeP : Bool;
//    tempPosAxInPos : Bool;
//    tempPosAxError : Bool;
//    tempPosAxState : Byte;

// Constant
//    None
```

Notes:

Example of positioning axis using an index table




```
REGION Pos axis

    // Scan rising edge
    #tempPosAxEdgeP := #posAx AND NOT #statPosAxFlag;
    #statPosAxFlag := #posAx;

    // Set Job Start
    IF #tempPosAxEdgeP THEN
        #statPosAxStart := TRUE;
    END_IF;

    // Pos axis start
    "LBP_CtrlAxisSpindle"(Start := #statPosAxStart,
                          Stop := FALSE,
                          Funct := 16#4,
                          Mode := 16#2,
                          AxisNo := 5,
                          Pos := 5.0,
                          FRate := 1000.0,
                          InPos => #tempPosAxInPos,
                          Error => #tempPosAxError,
                          State => #tempPosAxState);

    // Reset Job Start
    IF #tempPosAxInPos OR ("mcpIn".reset AND #tempPosAxError) THEN
        #statPosAxStart := FALSE;
    END_IF;

END_REGION
```

The PLC axis now moves to the position entered in MD10910[4].
(FC18 Parameter **Pos :=5.0)**

General machine data

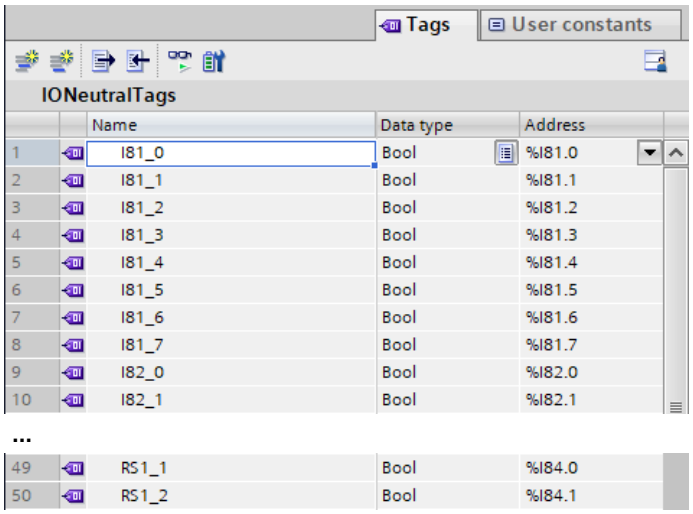
10900	$MN_INDEX_AX_LENGTH_POS_TAB_1	10	re
10910[0]	$MN_INDEX_AX_POS_TAB_1	0 mm|degr...	re
10910[1]	$MN_INDEX_AX_POS_TAB_1	36 mm|degr...	re
10910[2]	$MN_INDEX_AX_POS_TAB_1	72 mm|degr...	re
10910[3]	$MN_INDEX_AX_POS_TAB_1	108 mm|degr...	re
10910[4]	$MN_INDEX_AX_POS_TAB_1	144 mm|degr...	re
10910[5]	$MN_INDEX_AX_POS_TAB_1	180 mm|degr...	re
10910[6]	$MN_INDEX_AX_POS_TAB_1	216 mm|degr...	re
10910[7]	$MN_INDEX_AX_POS_TAB_1	252 mm|degr...	re
10910[8]	$MN_INDEX_AX_POS_TAB_1	288 mm|degr...	re
10910[9]	$MN_INDEX_AX_POS_TAB_1	324 mm|degr...	re

Machine	Position [mm]
MX1	0.000
MY1	0.000
MZ1	0.000
MA1	144.000 °

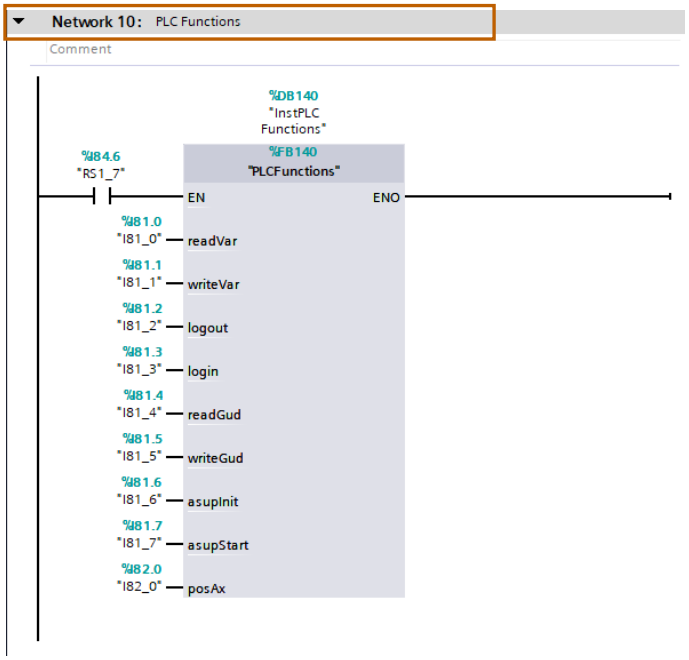
Notes:

Calling PLC functions in OB1

Create a tag table for the PLC functions.



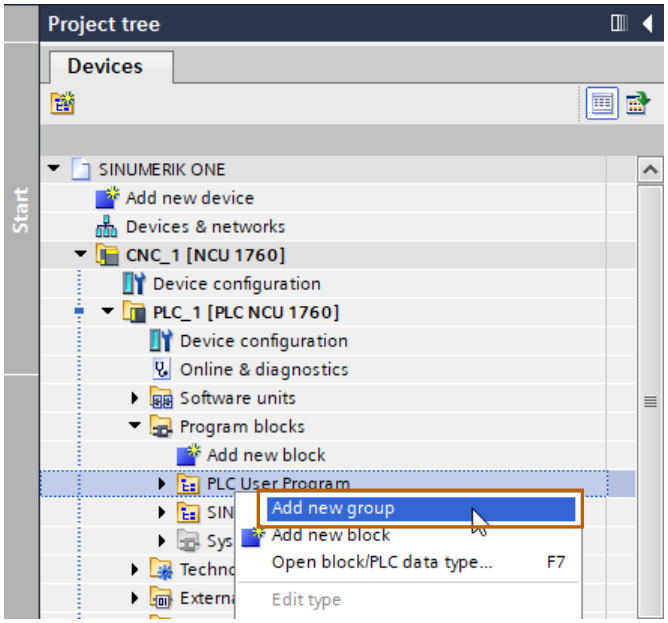
The newly created PLC functions are called in OB1.



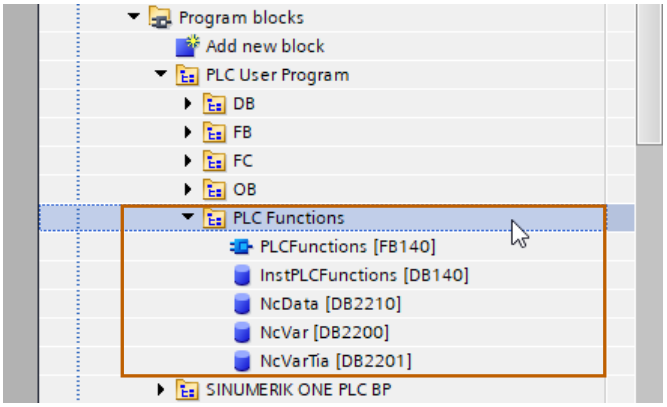
Notes:

Grouping PLC functions

Create a new group in the project navigation.



Insert PLC functions into the new group.



A125: END

Notes:

A119

SINUMERIK ONE

Tool Management

Module Description:

This module explains step by step the installation of Tool Management on a turning machine and on a milling machine.

Module objective:

After working through this module you will be capable of commissioning Tool Management on a turning machine and on a milling machine.

Content:

Fundamentals

User Interface PLC - NCK

Machine data for Tool Management

Magazine Configuration

Tool load / unload

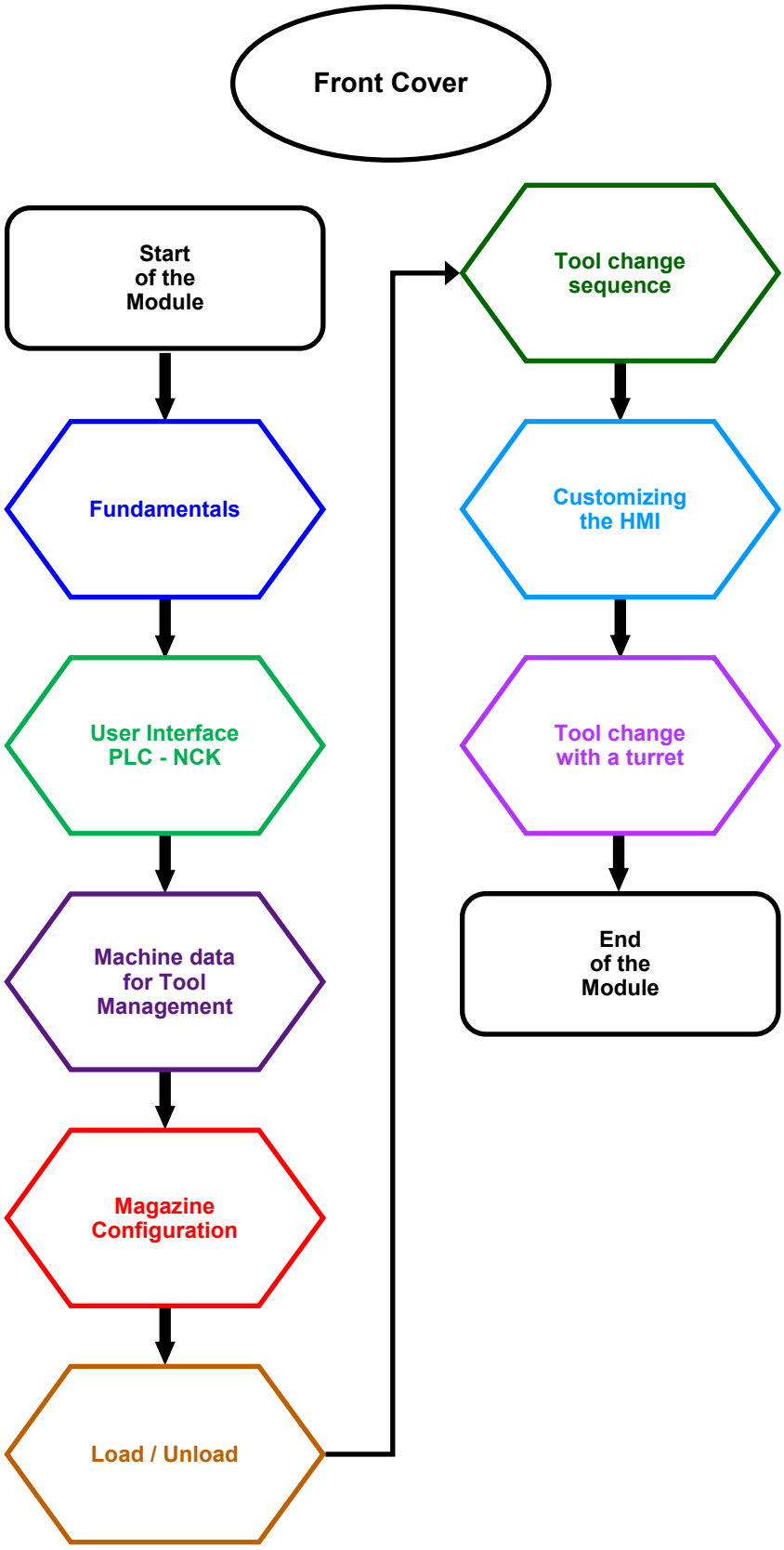
Tool change sequence

Customizing the HMI

Tool change with a turret

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Fundamentals
User Interface PLC - NCK
Machine data for Tool Management
Magazine Configuration
Load / Unload
Tool change sequence
Customizing the HMI
Tool change with a turret
End of the Module

Fundamentals

Requirements

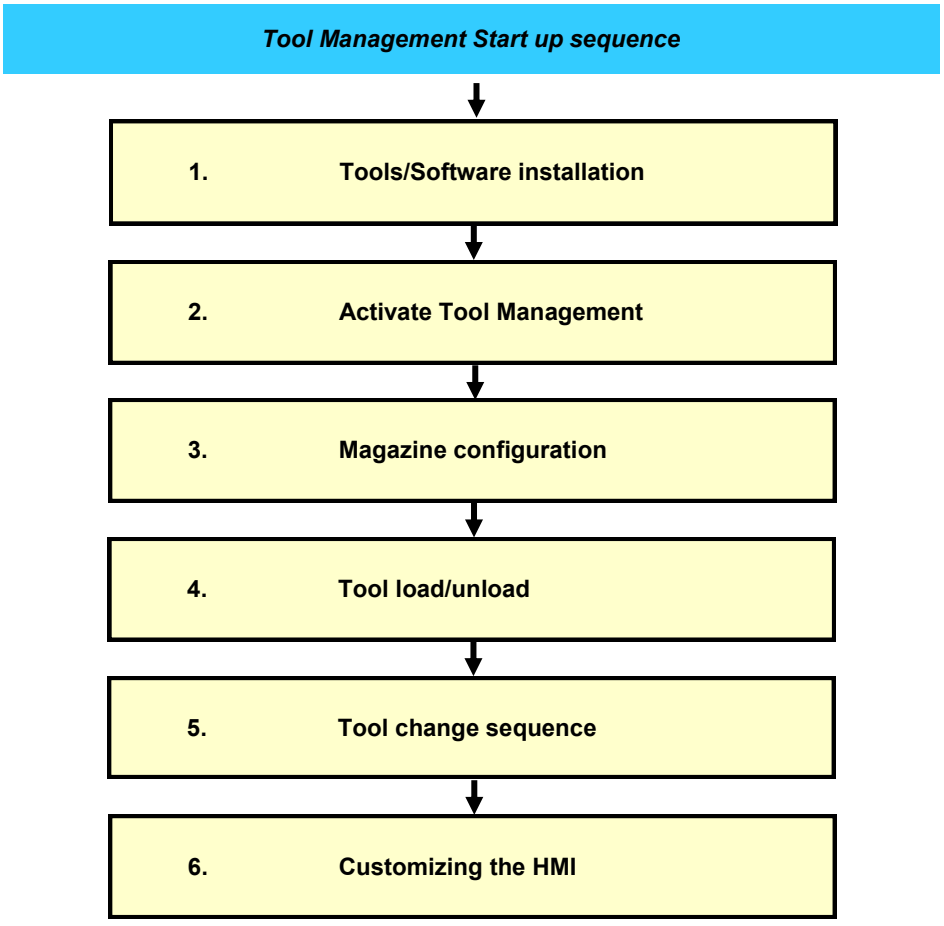
- Sinumerik ONE SW 6.2
- Toolbox ONE SW 6.2

Scope of delivery

Tool management is included in the basic scope of SINUMERIK ONE.

Licencing

- “Tool Management, more than 4 Magazines“
  Order Nr. 6FC5800-0BM88-0YB0



Notes:

Tool Management Checklist

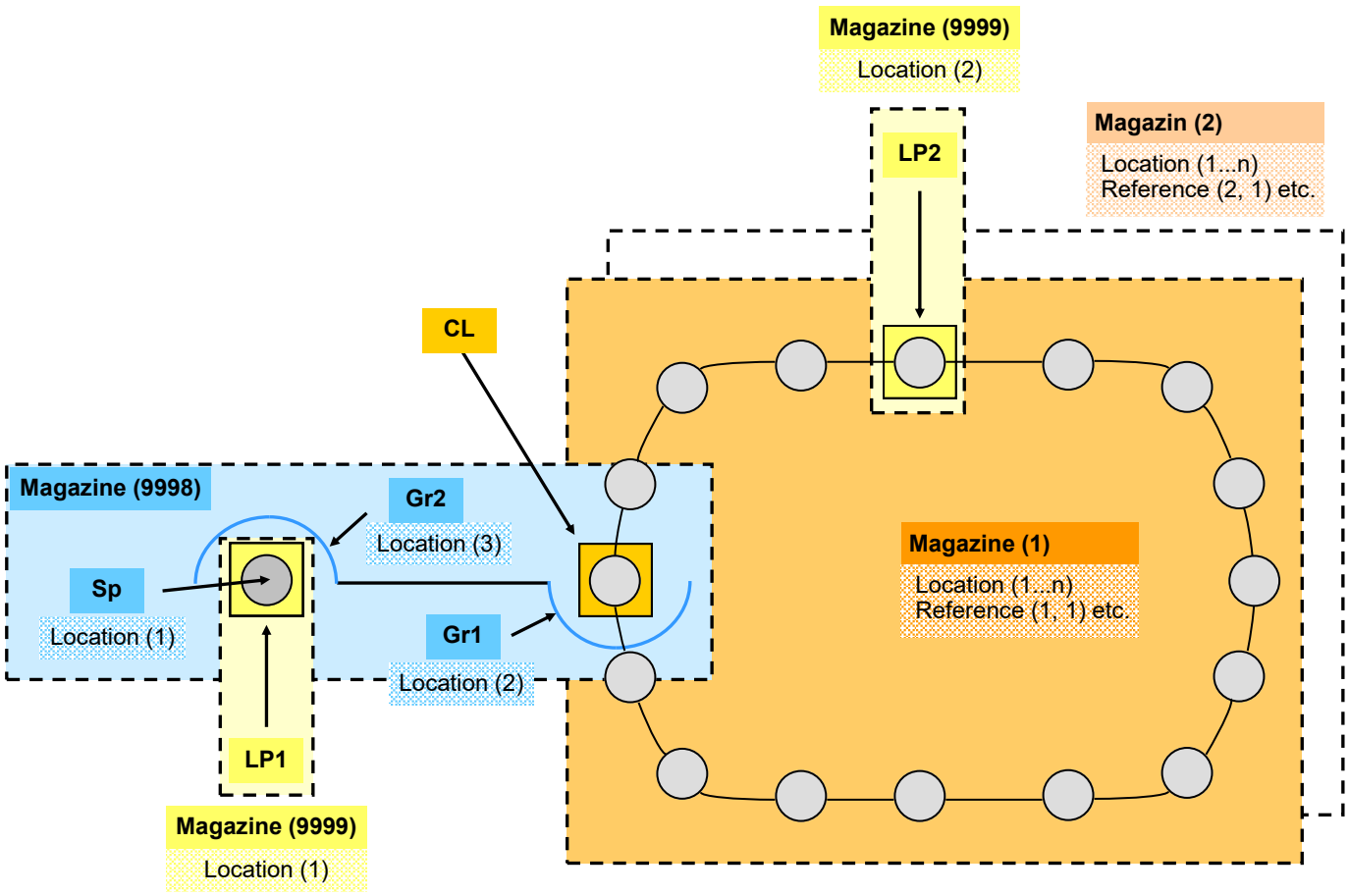
Tool management functions	Remarks
Define new tool /Multitool	
Tool load	
Tool unload	
Tool relocate	
Position Magazine	
Define a new tool directly in the magazine	
Define a new tool directly in the spindle	
Delete a tool in the magazine	
Delete a tool in the spindle	
Relocate the tool in the spindle	
Position a Multitool	
Prepare tool change	
Tool change	
Do tool change with a fixed location tool	
Do tool change with a hand tool	
Do tool change with a Multitool	
Reset during tool change	
Emergency Stop during tool change	
Main switch during tool change	
Tool recovery functions	
Tool change after block search	
Tool change during Simulation	
Tool change in Program Test	
Tool change in Inch und Metric	
Tool change in T,S,M	
Tool change during *Measure Tool* function	

Notes:

Notes:

Tool manangement

Magazine configuration

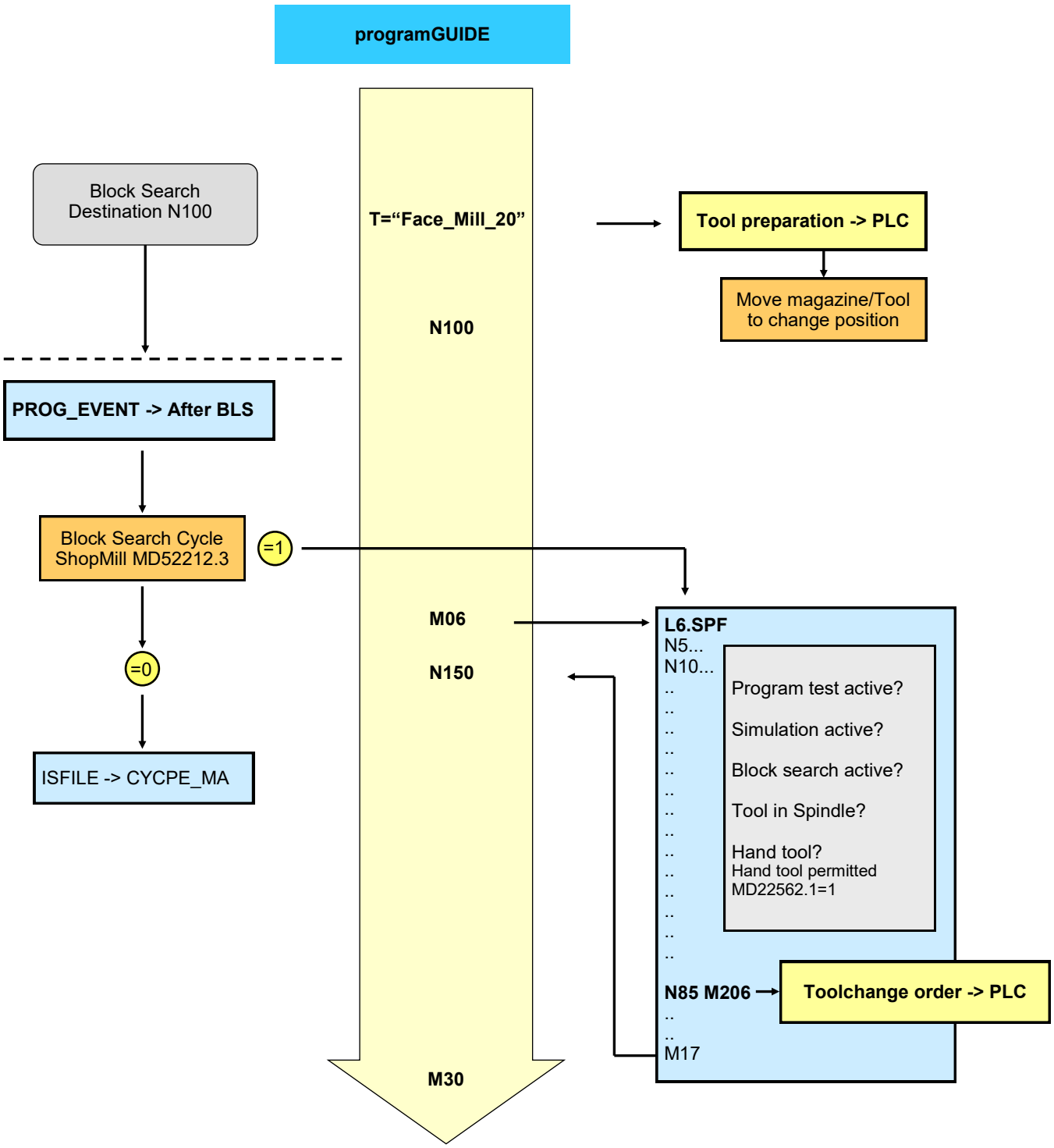


Magazine 1, 2	Real magazines e.g. chain, turret, box magazine
Magazine 9998	Buffer magazine
Magazine 9999	Loading magazine

CL	Change Location
LP1	Loading Position 1 (Spindle/Manual)
LP2	Loading Position 2 (Magazine)
Sp	Spindle
Gr1	Gripper 1
Gr2	Gripper 2

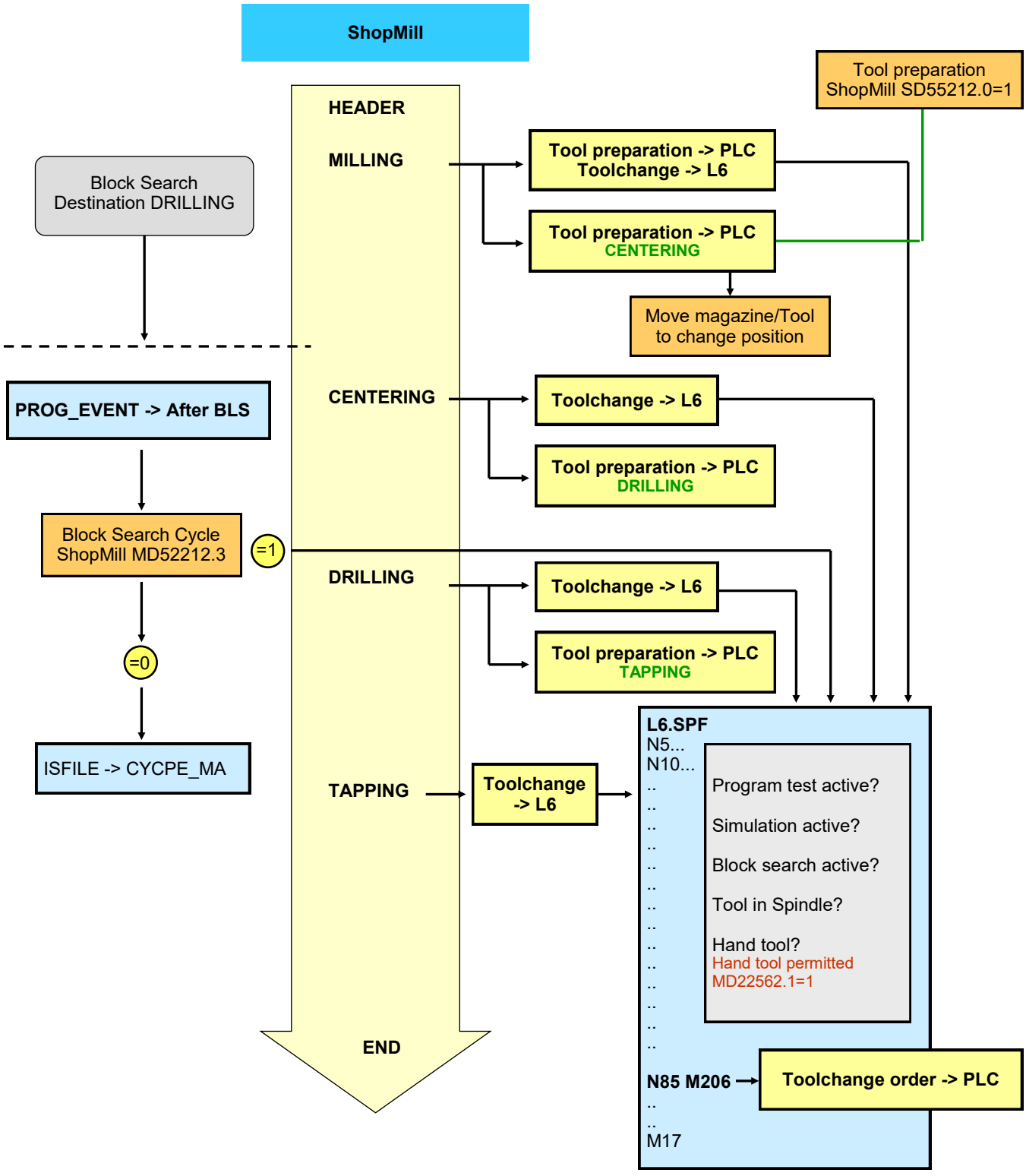
Notes:

Tool change process for milling machines



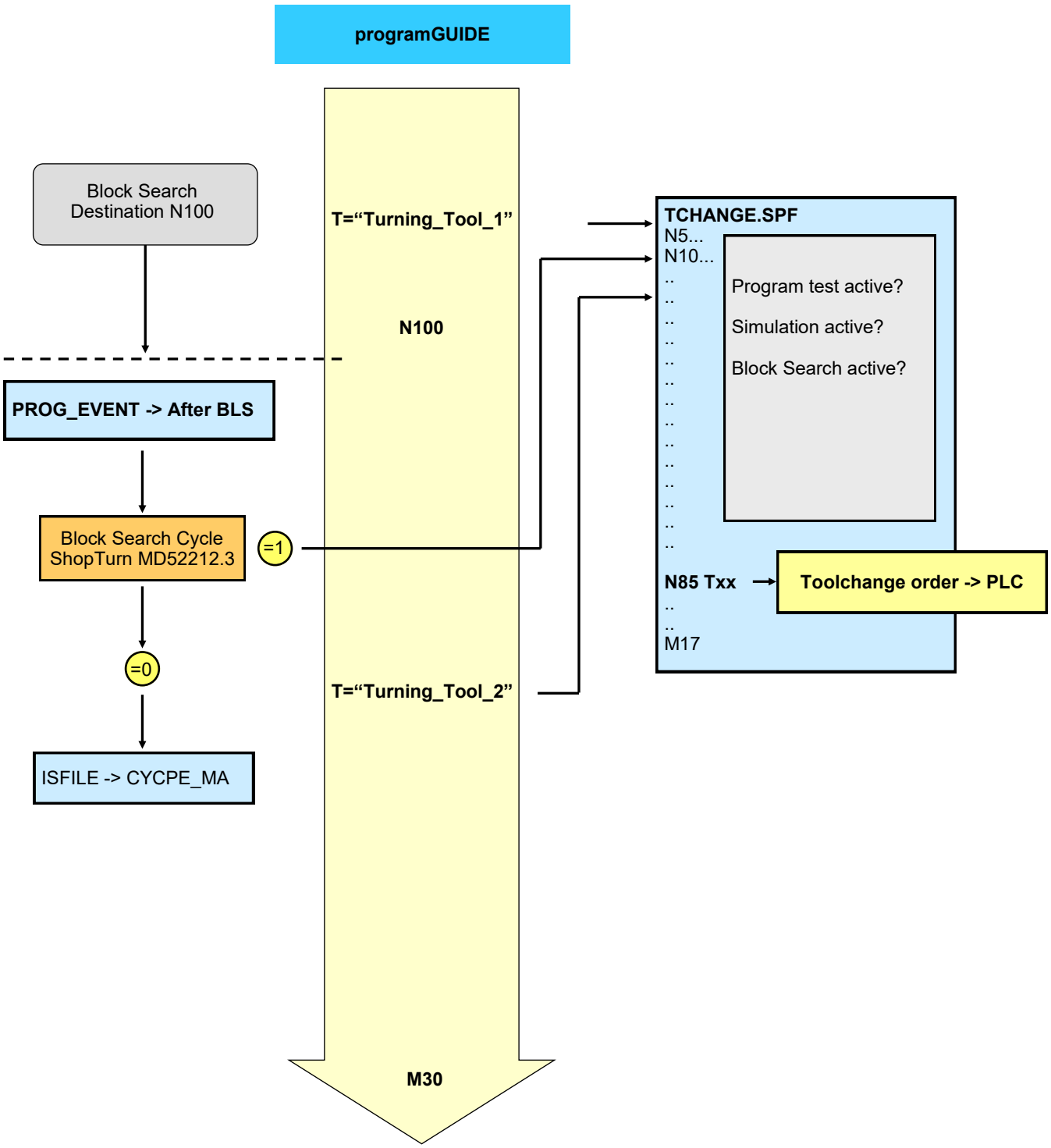
Notes:

Tool change process for milling machines



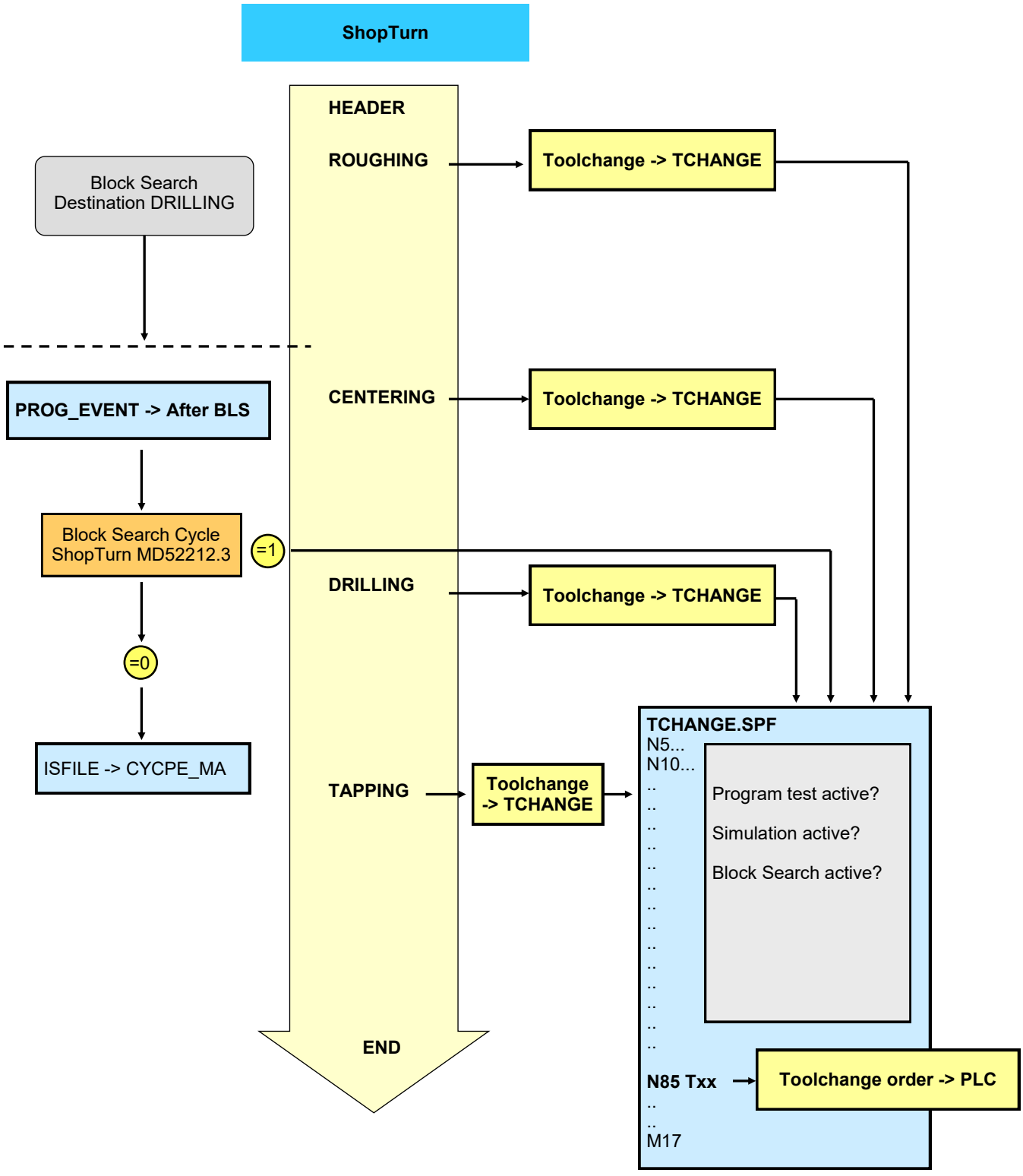
Notes:

Tool change process for turning machines



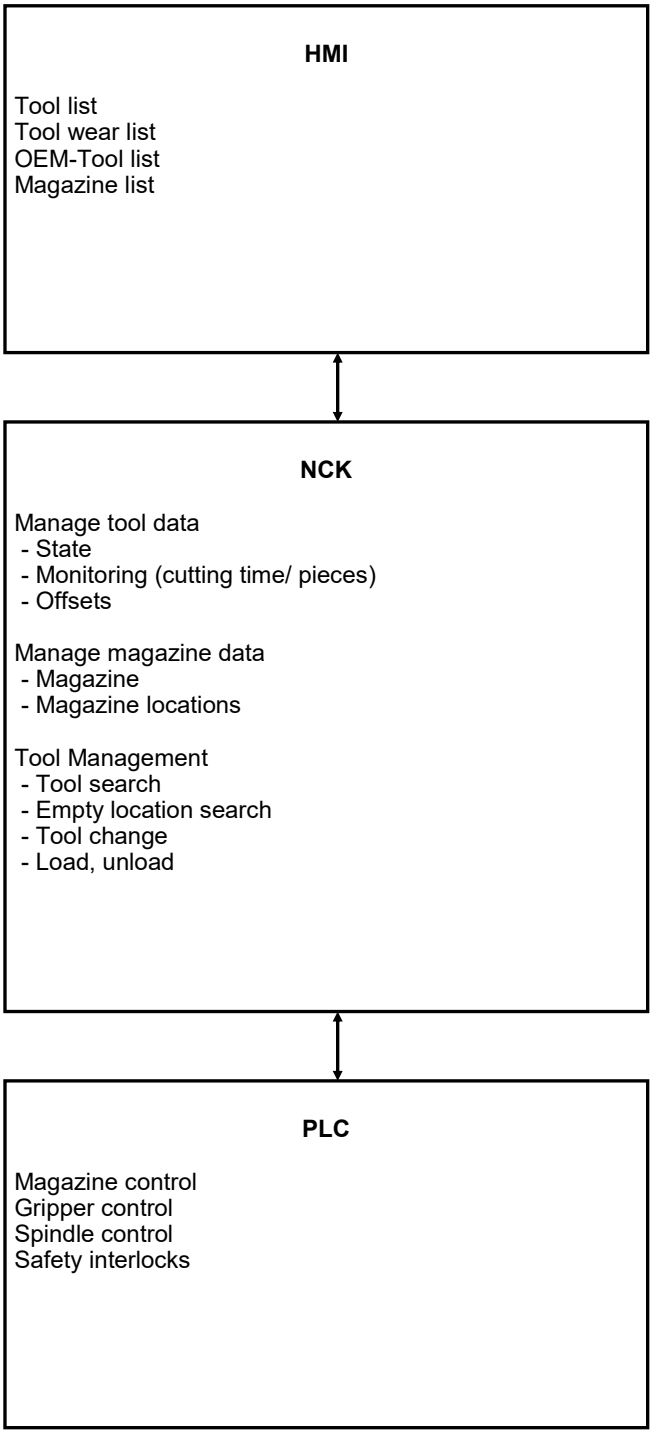
Notes:

Tool change process for turning machines



Notes:

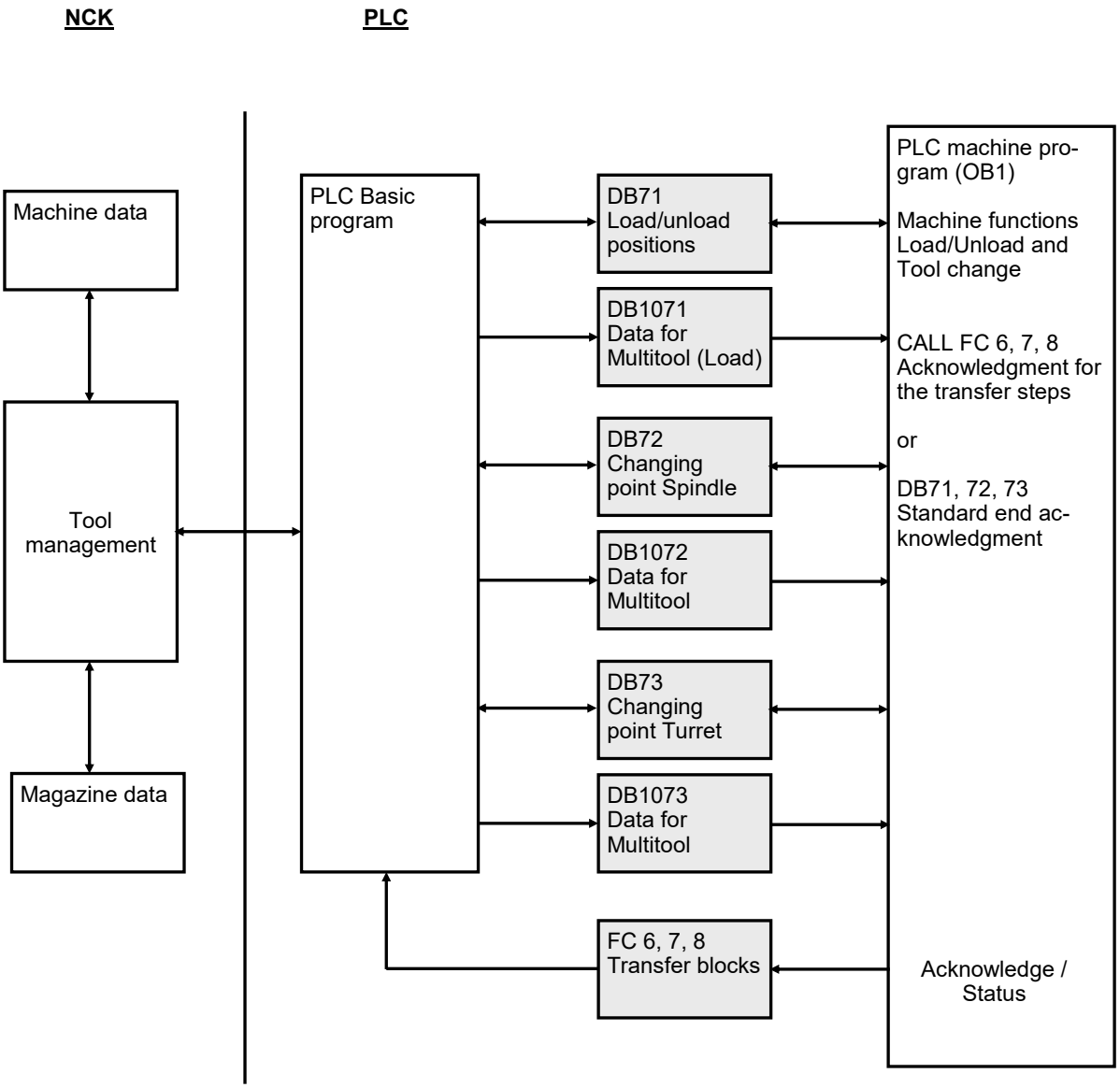
Structure of Tool Management



Notes:

User Interface PLC - NCK

The interface is set up for Tool Management functions: Load/Unload and tool change. The Functions FC6 (Multitool), FC7 (Turret) and FC8 are used to communicate changes in the current positions of tools.



Notes:

Overview of the interface data blocks

The PLC-NCK interface has these data blocks for tool load/unload and for tool change.

Data block no.	Responsible for:
DB71	Interface for loading/unloading points There is an interface in DB71 for each loading point (max. 16) which has been configured for the magazine (Interface 1-16). The interface for loading point 1 is active when loading directly into the Spindle. Interface 1 is also used for relocate functions.
DB72	Interface for spindle as changing position There is an interface in DB72 for each of the available spindles (max. 16) as defined in the magazine configuration, (Interface 1 - 16).
DB73	Interface for turret as change position There is an area in DB73 for each turret (max 16, Interface 1 - 16).
DB1071	Data for Multitool (Load/unloading points) DB1071 applies to Multitool (Load-/Unload) for each of the available loading points.
DB1072	Data for Multitool (Spindle) DB1072 applies to Multitool (Spindle) for each available Spindle.
DB1073	Data for Multitool (Turret) DB1073 applies to Multitool (Turret) for each available turret.

Notes:

Overview of the basic program blocks

The PLC basic program has the following functions to communicate changes in the status or position of tools in the case of tool load/unload and tool change.

Function no.	Responsible for:
FC6	Transfer block for Tool Management and Multitool The function is used in case of position and status changes (Load/ Unload, Tool change) . FC6 is used in the case of “Multitool“ . FC6 has the same functionality as FC8, but it also incorporates the Multitool functionality.
FC7	Transfer block for Tool change with turret FC7 is used for tool change with a turret.
FC8	Transfer block for Tool Management The function is used in case of position and status changes (Load/ Unload, Tool change).

Notes:

Tool management interface

Interface for loading/unloading magazine (DB71)

DB71	Interface for loading/unloading magazine (NCK->PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	I16	I15	I14	I13	I12	I11	I10	I9
DBB2 PLC->NCK	Standard end acknowledgement							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB3 PLC->NCK	I16	I15	I14	I13	I12	I11	I10	I9
DBB n	Reserved	Reserved	Position-ing Multitool	Order from NC-Program	Position-ing Magazine	Relocate	Unload	Load
DBB n+1	Data in extended area (DB1071)	Reserved						Acknowledge Status = 3 PLC->NCK
DBB n+2	Assigned channel (8 bit INT)							
DBB n+3	Tool management no. (8 bit INT)							
DBD n+4	Unassigned parameter 1 (Dword)							
DBD n+8	Unassigned parameter 2 (Dword)							
DBD n+12	Unassigned parameter 3 (Dword)							
DBW n+16	Identification for loading/unloading station (Int), (fixed value 9999)							
DBW n+18	No. of loading station (INT)							
DBW n+20	Magazine no. (source) for unloading/reloading/positioning (INT)							
DBW n+22	Location no. (source) for unloading/reloading/positioning (INT)							
DBW n+24	Magazine no. (target) for loading/reloading/positioning (INT)							
DBW n+26	Location no. (target) for loading/reloading/positioning (INT)							

Notes:

DB71	Interface for loading/unloading magazine (NCK->PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB n+28	Reserved							Loading/ unloading without magazine movement
DBB n+29	Reserved							

Initial addresses of the loading/unloading points (max. 16)
Loading/unloading point 1: n = 4
Loading/unloading point 2: n = 34

Interface for loading / unloading (Multitool) (DB1071)

DB1071	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW n+0	Distance code (1: location number, 2: distance, 3: angle)							
DBW n+2	Number of Multitool locations (number of locations of the Multitool)							
DBD n+4	Multitool location distance (distance of the MT location to be positioned from the reference location, corresponding to the distance coding)							
DBW n+8	Multitool number (internal T number of the Multitool)							
DBW n+10	Multitool location number (location number within the Multitool to which the Multitool positions to)							
DBW n+12	Tool holder (spindle or tool holder number)							
DBW n+14	Reserved							
DBW n+16	Reserved							
DBW n+18	Reserved							

Initial addresses of the loading/unloading points (max. 16)
Loading/unloading point 1: n = 0
Loading/unloading point 2: n = 20

Notes:

Interface for spindle as change position (DB72)

DB72	Signals from spindle (NCK->PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (I) active I8	I7	I6	I5	I4	I3	I2	I1
DBB1	I16	I15	I14	I13	I12	I11	I10	I9
DBB2 PLC->NCK	Standard end acknowledgement I8	I7	I6	I5	I4	I3	I2	I1
DBB3 PLC->NCK	I16	I15	I14	I13	I12	I11	I10	I9
DBB n	Spindle Tool re-mains in spindle	Unload manual tool	Load manual tool	OldT in buffer no (n+42)	T0	Prepare change	Perform change (initiate: M06)	Obligatory change
DBB n+1	Data in extended area (DB1072)	Reserved						Acknowledge Status = 3 PLC->NCK
DBB n+2	Assigned channel (8 bit INT)							
DBB n+3	Tool management no. (8 bit INT)							
DBD n+4	Unassigned parameter 1 (Dword)							
DBD n+8	Unassigned parameter 2 (Dword)							
DBD n+12	Unassigned parameter 3 (Dword)							
DBW n+16	Buffer identification (Int), (fixed value 9998) (corresponds to ”Target position for new tool”)							
DBW n+18	relative location (target or source) in buffer (INT)							
DBW n+20	Magazine no (source) for new tool (INT)							
DBW n+22	Location no. (source) for new tool (INT)							
DBW n+24	Magazine no. (target) for old tool (INT)							
DBW n+26	Location no. (target) for old tool (INT)							
DBW n+28	Tool new: location type (INT)							
DBW n+30	Tool new: size left (INT)							
DBW n+32	Tool new: size right (INT)							
DBW n+34	Tool new: size top (INT)							
DBW n+36	Tool new: size bottom (INT)							
DBB n+38	Tool status for tool new Manual tool	1:1 ex-change		Master tool	to be load-ed	to be un-loaded	ignore disabled	ID for tool in buffer

Notes:

DB72	Signals from spindle (NCK->PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB n+39	Tool status for tool new: Tool was in use	Tool fixed location coded	Tool being changed	Prewarning limit reached	Tool measured	Tool disabled	Tool enabled	Active tool
DBW n+40	Tool new: T no. (INT)							
DBW n+42	If DBX (n+0.4) = 1, then buffer location of old tool is entered here.							
DBW n+44	Original magazine of new tool							
DBW n+46	Original location of new tool							

Initial addresses of the Spindles (max. 16)
Spindle 1: n = 4
Spindle 2: n = 52

Interface for spindle (Multitool) (DB1072)

DB1072	Spindle (NCK -> PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW n+0	Distance code (1: Location number, 2: Distance, 3: Angle)							
DBW n+2	Number of Multitool locations							
DBD n+4	Multitool distance							
DBW n+8	Multitool number (new tool)							
DBW n+10	Multitool location number (new tool)							
DBW n+12	Multitool number (old tool)							
DBW n+14	Multitool location number (old tool)							
DBW n+16	Location type							
DBW n+18	Tool size left							
DBW n+20	Tool size right							
DBW n+22	Tool size above							
DBW n+24	Tool size below							
DBW n+26	Tool status							
DBW n+28	T number of the tool							
DBW n+30	Tool holder							
DBW n+32	Original magazine of the new tool							
DBW n+34	Original location of new tool							
DBW n+36	Reserved							
bis	Reserved							
DBW n+48	Reserved							

Initial addresses of the Spindles (max. 16)
Spindle 1: n = 0
Spindle 2: n = 50

Notes:

Interface for turret (DB73)

DB73	Signals from turret (NCK -> PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (I) active I8	I7	I6	I5	I4	I3	I2	I1
DBB1	I16	I15	I14	I13	I12	I11	I10	I9
DBB2 PLC->NCK	Standard end acknowledgement I8	I7	I6	I5	I4	I3	I2	I1
DBB3 PLC->NCK	I16	I15	I14	I13	I12	I11	I10	I9
DBB n	Reserved	Unload manual tool	Reserved		T0	Reserved	Perform change (initiation: T no.)	Obligatory change
DBB n+1	Data in extended area (DB1073)	Reserved						Acknowledge Status = 3 PLC->NCK
DBB n+2	Assigned channel (8 bit INT)							
DBB n+3	Tool management no. (8 bit INT)							
DBD n+4	Unassigned parameter 1 (Dword)							
DBD n+8	Unassigned parameter 2 (Dword)							
DBD n+12	Unassigned parameter 3 (Dword)							
DBW n+16	Reserved							
DBW n+18	Reserved							
DBW n+20	Magazine no. of the turret (INT)							
DBW n+22	Location no. for new tool (INT)							
DBW n+24	Magazine no. for old tool (INT)							
DBW n+26	Location no. for old tool (INT)							
DBW n+28	Tool new: location type (INT)							
DBW n+30	Tool new: size left (INT)							
DBW n+32	Tool new: size right (INT)							
DBW n+34	Tool new: size top (INT)							
DBW n+36	Tool new: size bottom (INT)							
DBB n+38	Tool status for tool new Manual tool	1:1 exchange		Master tool	to be loaded	to be unloaded	ignore disabled	ID for tool in buffer
DBB n+39	Tool status for tool new Tool was in use	Tool fixed location coded	Tool being changed	Pre-warning limit reached	Tool measured	Tool disabled	Tool enabled	Active tool

Notes:

DB73	Signals from turret (NCK -> PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW n+40	Tool new: T no. (INT)							
DBW n+42	Original location of new tool in this turret magazine							

Initial addresses of the Turrets (max. 16)
Turret 1: n= 4
Turret 2: n = 48

Interface for Turret (Multitool) (DB 1073)

DB1073	Turret (NCK -> PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW n+0	Distance code (1: Location number, 2: Distance, 3: Angle)							
DBW n+2	Number of Multitool locations							
DBD n+4	Multitool distance							
DBW n+8	Multitool number (new tool)							
DBW n+10	Multitool location number (new tool)							
DBW n+12	Multitool number (old tool)							
DBW n+14	Multitool location number (old tool)							
DBW n+16	Location type							
DBW n+18	Tool size left							
DBW n+20	Tool size right							
DBW n+22	Tool size above							
DBW n+24	Tool size below							
DBW n+26	Tool status							
DBW n+28	T number of the tool							
DBW n+30	Tool holder							
DBW n+32	Original magazine of the new tool							
DBW n+34	Original location of new tool							
DBW n+36	reserved							
to	reserved							
DBW n+48	reserved							

Initial addresses of the Turrets (max. 16)
Turret 1: n= 0
Turret 2: n = 50

Notes:

Basic program blocks for tool management

FC6: Transfer block for tool management and multitool

Signal	Type	Type	Value range	Comment
Start	E	BOOL		1 = Start of transfer
TaskIdent	E	BYTE	1 … 5	Interface or tank identifier 1: Loading/unloading location 2: Spindle change position 3: Turret change position 4: Asynchronous transfer 5: Asynchronous transfer with location reservation
TaskIdentNo	E	BYTE	1 …	Number of associated interface or channel number. The upper nibble can specify the interface number for asynchronous transfer (e.g. B#16#12, 1st interface, 2nd channel).
NewToolMag	E	INT	-1, 0 …	Current magazine number of tool to be replaced -1: Tool remains at its location
NewToolLoc	E	INT	0 … max. location number	Current location number of new tool
OldToolMag	E	INT	-1, 0 …	Current magazine number of tool to be replaced -1: Tool remains at its location
OldToolLoc	E	INT	0 … max. location number	Current location number of tool to be replaced
Status	E	INT	1 … 10 103 … 105	Status information about transfer operation
MultitoolLoc	E	INT	0 … max. location number	Multitool location No.
Ready	A	BOOL		1 = Transfer complete
Error	A	INT	0 … 65535	Error feedback 0: No error has occurred 1: Unknown "TaskIdent" 2: Unknown “TaskIdentNo“ 3: Illegal task ("signal "Interface (SS) active" of selected turret = "FALSE") Other values: The number corresponds to the error message of the tool management function in the NCK caused by this transfer.

Notes:

FC7: Transfer block for tool change with turret

Signal	Type	Type	Value range	Comment
Start	E	BOOL		1 = Start of transfer
ChgdRevNo	E	BYTE	1..	Number of turret interface
Ready	A	BOOL		1 = Transfer complete
Error	A	INT	0 … 3	Error feedback 0: No error has occurred 1: No turret present 2: Illegal turret number in parameter "ChgdRevNo" 3: Illegal job ("interface active" signal for Illegal job ("interface active" signal for

FC8: Transfer block for tool management

Signal	Type	Type	Value range	Comment
Start	E	BOOL		1 = Start of transfer
TaskIdent	E	BYTE	1 … 5	Interface or tank identifier 1: Loading/unloading location 2: Spindle change position 3: Turret change position 4: Asynchronous transfer 5: Asynchronous transfer with location reservation
TaskIdentNo	E	BYTE	1 …	Number of associated interface or channel number. The upper nibble can specify the interface number for asynchronous transfer (e.g. B#16#12, 1st interface, 2nd channel).
NewToolMag	E	INT	-1, 0 …	Current magazine number of tool to be replaced -1: Tool remains at its location
NewToolLoc	E	INT	0 … max. location number	Current location number of new tool
OldToolMag	E	INT	-1, 0 …	Current magazine number of tool to be replaced -1: Tool remains at its location
OldToolLoc	E	INT	0 … max. location number	Current location number of tool to be replaced
Status	E	INT	1 … 10 103 … 105	Status information about transfer operation
Ready	A	BOOL		1 = Transfer complete
Error	A	INT		Error feedback 0: No error has occurred 1: Unknown "TaskIdent" 2: Unknown “TaskIdentNo“ 3: Illegal task ("signal "Interface (SS) active" of selected turret = "FALSE") Other values: The number corresponds to the error message of the tool management function in the NCK caused by this transfer.

Notes:

Acknowledgements from tool management

Acknowledg-ment	Status	Explanation
Synchronous Final-Acknowledg-ment	1	**The WZV operation is finished.** In the case of loading/unloading/reloading, the tool has arrived at the required target address. In the case of "Prepare change", the new tool is now available. In the case of "Change" (spindle or revolver), the tools addressed in the interface have now reached the required target addresses. The tool-change operation is thus completed.
	2	**The "new" tool cannot be made available.** This status is only admissible in conjunction with the "Prepare Change" command. When this status is applied, the PLC must be prevented from making a change with the proposed tool. The proposed (new) tool is disabled by the tool manage-ment function in the NCK. A new command is then output by the tool management with a duplo tool.
	3	**An error occurred.** The tool positions must not have been changed. Any changes to the magazine positions which have taken place in the meantime must be notified beforehand.
	4	**It would be better to position the "old" tool in the magazine position speci-fied in parameters "OldToolMag" and "OldToolLoc"..** This status is permissible only in conjunction with preparation for tool change (change into spindle). After this status has been transferred to the tool manage-ment in the NCK, the tool management tries to consider the specified magazine position in the next command. But this is done only when this position is free.
	5	**The operation is complete.** The "new" tool is in the position specified in parameters "NewToolMag", "NewToolLoc". In this case, the specified tool is not really in this position, but is still in the same magazine location. However, this magazine location has been moved to the position set in the parameters (e.g. tool change position).
	6	**The WZV job has been completed.** This status has the same function as status 1, but, in addition, a reservation of the source location is carried out. This status is only permitted when reloading.
	7	**Initiate repetition of the command "Prepare Tool".** This status is only admissible in conjunction with the "Change tool" command. This status is intended for use when the "new" tool has changed its position (e.g. via an asynchronous command of the "new" tool). After "Ready = 1" has been provided by FC 8, the "Prepare Change" command is repeated automatically with the same tool.
	10	**The WZV operation is finished.** This status has the same function as status 1,with the one difference that the mag-azine position ($TC_MAP8) is not changed. This status is only permitted with the Load/ Unload/ Relocate jobs.
Synchronous Intermediate-Acknowledg-ment	103	**The "new" tool can be inserted.** This status is permitted only in the tool change preparation, when the PLC may reject the new tool (e.g. in case of MD20310 Bt 4=1).
	104	**The "new" tool is in the position specified in parameters "NewToolMag", "NewToolLoc".** This status may be used only in conjunction with revolvers, chain-type magazines and disk magazines for the "Tool change preparation" phase. Status enables the tool management to adjust the current position of a magazine and to improve the search strategy for subsequent commands.
	105	**The specified buffer has been reached by all tools involved.** The tools are in the specified tool positions (parameters "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc").

Notes:

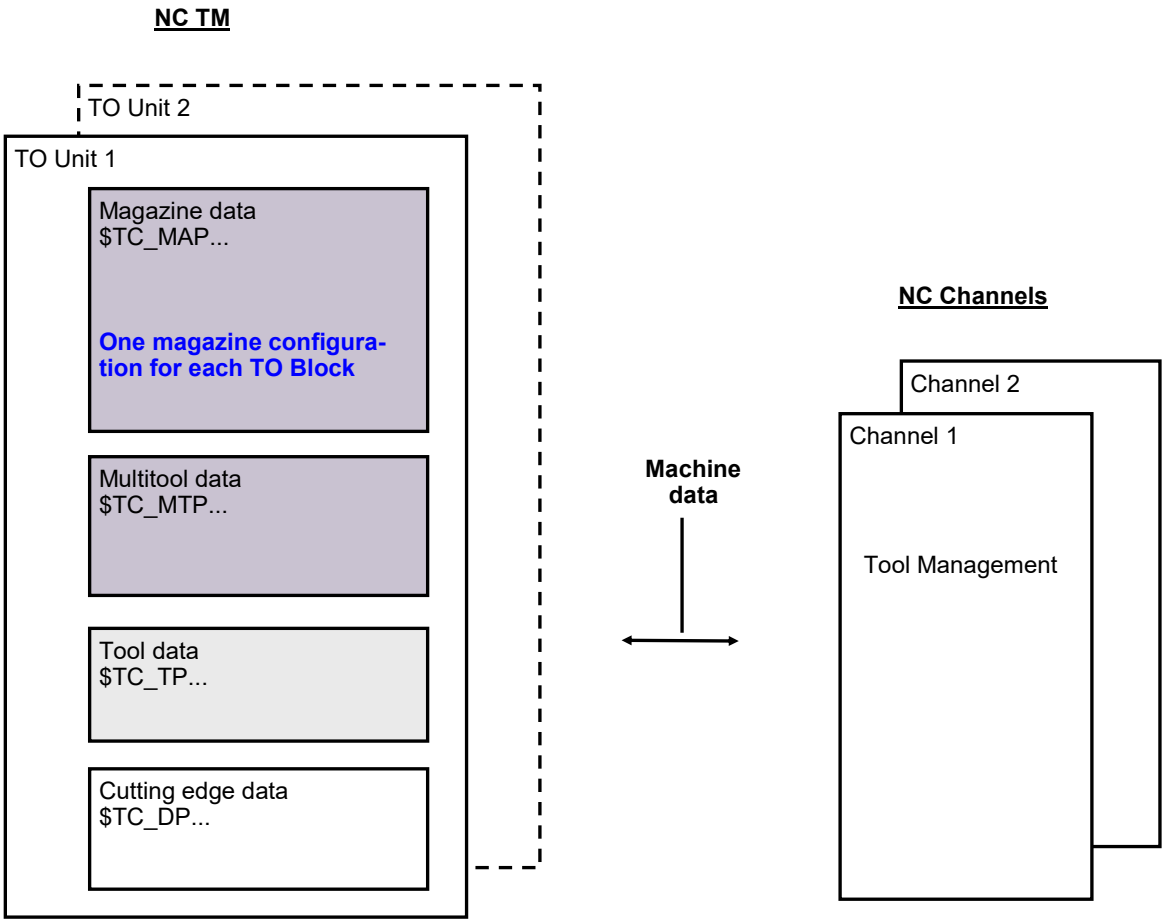
Jobs from tool management

	Jobs	Interface	Acknowledgment	Applications, special features
Jobs from NCK tool management	Loading	DB71	FC6/FC8, TaskIdent =1, TaskIdentNo =InterfaceNo. Status =1	NewToolLoc =Target destination of the tool NewToolMag =Target magazine OldToolLoc =No. of the loading point OldToolMag =9999
	Unloading	DB71	FC6/FC8, TaskIdent =1, TaskIdentNo =InterfaceNo. Status =1	NewToolLoc =No. of the loading point NewToolMag =9999 OldToolLoc =Location of the tool to be unloaded OldToolMag =MagNo of the tool to be unloaded
	Relocating	DB71	FC6/FC8, TaskIdent =1, TaskIdentNo =InterfaceNo. Status =1	NewToolLoc =Target position of the tool NewToolMag =Target magazine OldToolLoc =Source location OldToolMag =Source magazine
	Positioning	DB71	FC6/FC8, TaskIdent =1, TaskIdentNo =InterfaceNo. Status =5	Positioning at the loading magazine corresponding to the interface number. NewToolLoc =Loading point or buffer NewToolMag =9999 or 9998 OldToolLoc =Mag. location to be positioned OldToolMag =Magazine to be positioned
	Prepare tool change	DB72	FC6/FC8, TaskIdent =2, TaskIdentNo =InterfaceNo.	Positioning new tool to the change position, old tool remains in spindle to complete status 1, in order that the change command can be output. OldToolLoc =Buffer (spindle) NewToolLoc =New tool location
	Change in spindle	DB72	FC6/FC8, TaskIdent =2, TaskIdentNo =InterfaceNo.	Old tool is unloaded (Gripper or directly into magazine), new tool is loaded to spindle. Status 1 required, so that part program processing is continued. NewToolLoc =Buffer (spindle) OldToolLoc =Old tool location
	Without NCK Command: Return old Tool to magazine		FC6/FC8, TaskIdent =4, TaskIdentNo =ChannelNo.	Old tool may need to be transferred asynchronously to the location specified in the prepare change command to move the tool from the gripper to the magazine.
	Change with turret	DB73	NormallyFC7 or FC6/ FC8, TaskIdent =3, TaskIdentNo =TurretNo.	When turret has finished swivelling, FC7 is called with turret no. as parameter ChgdRevNo.
Position changes of tools without job from NCK	Asynchronous Transfer		FC6/FC8, TaskIdent =4, TaskIdentNo =ChannelNo. Status =1	General tool transport from the PLC NewToolMag/NewToolLoc =Target OldToolMag/OldToolLoc =Source
	Asynchronous Transfer with location reservation for tool transport		FC6/FC8, TaskIdent =5, TaskIdentNo =ChannelNo. Status =1	Tool transport from PLC NewToolMag/NewToolLoc =Buffer OldToolMag/OldToolLoc =Source in the real magazine
	Asynchronous Transfer without tool movement to align the magazine position		FC6/FC8, TaskIdent =4, TaskIdentNo =ChannelNo. Status =5	Communicate the actual magazine position (e.g. after turret cycles) to the NCK NewToolMag/NewToolLoc =Actual location OldToolMag/OldToolLoc =Change or loading point

Notes:

Machine data for Tool Management

Tool Management requires that changes are made to the machine data for memory configuration. The TO units need to be assigned to the channels, furthermore memory is required in the non-retentive RAM.



Note:

- The cutting edge data are available by default.
- The tool data are only available when tool monitoring is active.
- The magazine and multitool data are only available when the magazine management is active.

Notes:

Overview of the Machine data

MD	Identifier	Settings	Comment	Description
10715	$MN_M_NO_FCT_CYCLE	6		M function to be replaced by a subroutine
10716	$MN_M_NO_FCT_CYCLE_NAME	L6		Subroutine name for M function replacement
10717	$MN_T_NO_FCT_CYCLE_NAME			Name of tool-changing cycle for T function replacement
10718	$MN_M_NO_FCT_CYCLE_PAR	-1		M function replacement with parameters
10719	$MN_T_NO_FCT_CYCLE_MODE	0H		Setting of T function substitution - Bit 0: Substitution of the D or DL number - Bit 1: Call time for block start or block end - Bit 2: Call time for block start and block end
11410	$MN_SUPPRESS_ALARM_MASK	Bit 7 = 1		Mask for support of special alarm outputs - Bit 7: Alarm 22070 (Message alarm “Please change tool to magazine. Repeat data backup.”)
11415	$MN_SUPPRESS_ALARM_MASK_2	Bit 16 = 0		Masking of alarm outputs - Bit 16: Alarm 17212, 17214, 17215, 17216 (Manual tools)
11450	$MN_SEARCH_RUN_MODE	Bit 0 = 1 Bit 1 = 1 Bit 2 = 1		Parameterization for search run Bit 0: Alarm 10208 after action blocks Bit 1: Automatic PROG_EVENT start after block search Bit 2: The output of the spindle functions is suppressed
15710	$MN_TCA_CYCLE_NAME			Name of tool-changing cycle for TCA function replacement
17500	$MN_MAXNUM_REPLACEMENT_TOOLS	0		Maximum number of replacement tools
17504	$MN_MAX_TOOLS_PER_MULTITOOL	6		Maximum number of tool locations per multitool
17530	$MN_TOOL_DATA_CHANGE_COUNTER	1FH		Mark tool data change for HMI - Bit 0: Change tool status - Bit 1: Change residual number of tool - Bit 2: Tool data in update service - Bit 3: Magazine data in update service - Bit 4: ISO data in update service
18080	$MN_MM_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 10 = 1		Memory reservation for tool management (SRAM) - Bit 0: Provide magazine management data - Bit 1: Provide monitoring data - Bit 2: Provide OEM/CC data - Bit 3: Consider adj. location - Bit 4: Complex search for tools in magazine - Bit 5: Wear monitoring - Bit 6: Classification of magazines in wear groupings - Bit 7: Adapter of magazine locations - Bit 8: Additive and/or set-up offsets - Bit 9: Handling a turret - Bit 10: The multitool function is available
18082	$MN_MM_NUM_TOOL	100		Number of tools the NCK can manage (SRAM)
18083	$MN_MM_NUM_MULTITOOL	15		Number of multitools the NCK can manage (SRAM)
18084	$MN_MM_NUM_MAGAZINE	4		Number of magazines the NCK can manage (SRAM)
18085	$MN_MM_NUM_MULTITOOL_LOCATIONS	30		Number of multitool locations the NCK can manage (SRAM)
18086	$MN_MM_NUM_MAGAZINE_LOCATION	35		Number of magazine locations the NCK can manage (SRAM)
18100	$MN_MM_NUM_CUTTING_EDGES_IN_TOA	200		Tool offsets in the TO range (SRAM)
18105	$MN_MM_MAX_CUTTING_EDGE_NO	9		Maximum value of D number
18106	$MN_MM_MAX_CUTTING_EDGE_PERTOOL	9		Maximum number of D numbers per tool

Notes:

Overview of the Machine data

MD	Identifier	Settings	Comment	Description
20110	$MC_RESET_MODE_MASK	Bit 0 = 1 Bit 6 = 1		Definition of basic control settings after reset/part program end - Bit 0: RESET mode - Bit 6: Active tool length compensation
20120	$MC_TOOL_RESET_VALUE	0		Tool length compensation. during run-up (reset/part program end)
20124	$MC_TOOL_MANAGEMENT_TOOLHOLDER	1		Tool holder number
20128	$MC_COLLECT_TOOL_CHANGE	0		Tool change commands to PLC after block search
20130	$MC_CUTTING_EDGE_RESET_VALUE	1		Tool edge length compensation. during run-up (reset/part program end)
20270	$MC_CUTTING_EDGE_DEFAULT	1		Initial position of tool cutting edge without programming
20310	$MC_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 14 = 1 Bit 24 = 1		Activation of tool management functions - Bit 0: Magazine management active - Bit 1: Monitoring functions active - Bit 2: OEM-/CC- functions active - Bit 3: Adjacent location treatment - Bit 4: Requirement Tool change preparation - Bit 5: HL/PLC sync. for main spindle on transport ack. - Bit 6: HL/PLC sync. for second. spindle on transport ack. - Bit 7: HL/PLC Sync. for main spindle during PLC ack. - Bit 8: HL/PLC sync. for second. spindle during PLC ack. - Bit 9: Simulation on PLC acknowledgements - Bit 10: Tool change command only after PLC prep. ack. - Bit 11: Preparation command with same tool - Bit 12: Preparation command when tool already in spindle - Bit 13: Trace of tool sequences - Bit 14: Automatic tool change during RESET and Start - Bit 15: No return transport with multiple preparation cmd. - Bit 16: Programming T="Location number" - Bit 17: Control of time monitoring via PLC - Bit 18: Message to PLC with last replacement tool - Bit 19: Synchronization behaviour (Bits 5-8) - Bit 20: Behaviour during program test - Bit 21: Response to tool state "W" - Bit 22: T="Location number" + tool sub-group ($TC_TP11) - Bit 23: On offset selection no synchronization with HL - Bit 24: Command number 8 and 9 ignore location reservation
20320	$MC_TOOL_TIME_MONITOR_MASK	Bit 0 = 1		Time monitoring for tool in tool holder - Bit 0: Monitoring tool in tool holder 1
22550	$MC_TOOL_CHANGE_MODE	1		New tool offset with M function
22560	$MC_TOOL_CHANGE_M_CODE	206		M function for tool change
22562	$MC_TOOL_CHANGE_ERROR_MODE	Bit 1 = 1		Response to tool change errors - Bit 1: Manual tools permitted
28085	$MC_MM_LINK_TOA_UNIT	1		Assignment of a TO unit to a channel (SRAM)
28450	$MC_MM_TOOL_DATA_CHG_BUFF_SIZE	400		Buffer for tool data changes (DRAM)

Notes:

Overview of the Technology settings

MD	Identifier	Settings	Comment	Description
51200	$MNS_ACCESS_WRITE_TM_GEO	5		Write tool offset geometry data protection level
51201	$MNS_ACCESS_WRITE_TM_WEAR	6		Write tool offset wear data protection level
51202	$MNS_ACCESS_WRITE_TM_WEAR_DELTA	7		Protection level for tool offset restricted writing of wear data
51203	$MNS_ACCESS_WRITE_TM_SC	7		Write tool offset total offsets protection level
51204	$MNS_ACCESS_WRITE_TM_EC	7		Write tool offset use offsets protection level
51205	$MNS_ACCESS_WRITE_TM_SUPVIS	7		Write tool offset monitoring data protection level
51206	$MNS_ACCESS_WRITE_TM_ASSDNO	7		Write tool offset unique D number protection level
51207	$MNS_ACCESS_WRITE_TM_WGROUP	7		Write tool offset wear groups protection level
51208	$MNS_ACCESS_WRITE_TM_ADAPT	7		Write tool offset adapter data protection level
51209	$MNS_ACCESS_WRITE_TM_NAME	4		Write tool offset tool name protection level
51210	$MNS_ACCESS_WRITE_TM_TYPE	4		Write tool offset tool type protection level
51211	$MNS_ACCESS_READ_TM	7		Read tool offset data protection level
51212	$MNS_TM_WRITE_WEAR_ABS_LIMIT	0.999		Maximum tool wear value
51213	$MNS_TM_WRITE_WEAR_DELTA_LIMIT	0		Maximum difference value restricted tool wear input
51214	$MNS_TM_WRITE_LIMIT_MASK	7H		Validity of the restricted tool wear input - Bit 0: Use for cutting edge data, wear - Bit 1: Use for SC data, total offsets - Bit 2: Use for EC data, use offsets
51215	$MNS_ACCESS_WRITE_TM_ALL_PARAM	4		Protection level TM details - write all parameters
51216	$MNS_ACCESS_WRITE_TM_TOOL_CREATE	4		Protection level TM create tool
51217	$MNS_ACCESS_WRITE_TM_TOOL_DELETE	4		Protection level TM delete tool
51218	$MNS_ACCESS_WRITE_TM_TOOL_LOAD	4		Protection level TM load tool
51219	$MNS_ACCESS_WRITE_TM_TOOL_UNLOAD	4		Protection level TM unload tool
51220	$MNS_ACCESS_WRITE_TM_TOOL_MOVE	4		Protection level TM relocate tool
51221	$MNS_ACCESS_WRITE_TM_TOOL_REACTIVATE	4		Protection level TM reactivate tool
51222	$MNS_ACCESS_WRITE_TM_TOOL_MEASURE	4		Protection level TM measure tool
51223	$MNS_ACCESS_WRITE_TM_TOOL_CREATE	4		Protection level TM create tool cutting edge
51224	$MNS_ACCESS_WRITE_TM_TOOL_DELETE	4		Protection level TM delete tool cutting edge
51225	$MNS_ACCESS_WRITE_TM_MAGAZINE_POS	4		Protection level TM position magazine

Notes:

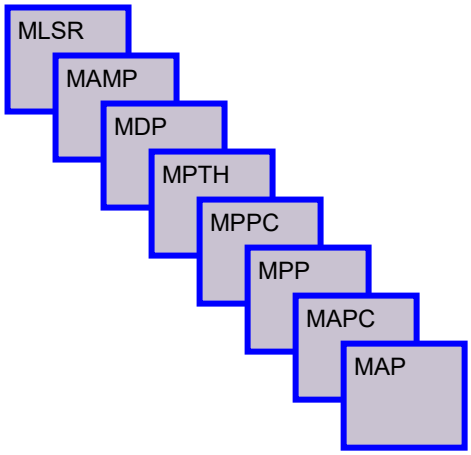
Overview of the Technology settings

MD	Identifier	Settings	Comment	Description
52210	$MCS_FUNCTION_MASK_DISP	Bit 4 = 1		Function mask Display - Bit 4: Show follow-on tool in T,F,S window
52212	$MCS_FUNCTION_MASK_TECH	Bit 3 = 1		Function mask Cross-technology - Bit 3: Call block search cycle for ShopMill/ShopTurn
52229	$MCS_ENABLE_QUICK_M_CODES	0H		Enable fast M functions (for ShopMill/ShopTurn) - Bit 0: Enable fast M functions: coolant OFF - Bit 1: Enable fast M functions: coolant 1 ON - Bit 2: Enable fast M functions: coolant 2 ON - Bit 3: Enable fast M functions: coolant 1 and 2 ON
52230	$MCS_M_CODE_ALL_COOLANTS_OFF	9		M code for all coolants OFF (for ShopMill/ShopTurn)
52231	$MCS_M_CODE_COOLANT_1_ON	8		M code for coolant 1 ON (for ShopMill/ShopTurn)
52232	$MCS_M_CODE_COOLANT_2_ON	7		M code for coolant 2 ON (for ShopMill/ShopTurn)
52233	$MCS_M_CODE_COOLANT_1_AND_2_ON	-1		M code for coolant 1 and 2 ON (for ShopMill/ShopTurn)
52240	$MCS_NAME_TOOL_CHANGE_PROG	M6		Tool change program for G code steps
52270	$MCS_TM_FUNCTION_MASK	0H		Function mask Tool management - Bit 0: Do not permit Create tool on magazine location - Bit 1: Load/unload disable when machine is not in reset - Bit 2: Load/unload disable for emergency stop - Bit 3: Load/unload or relocate tool to/from buffer disabled - Bit 4: Load directly in spindle - Bit 5: Use grinding configuration file - Bit 6: Permit relocating to/from buffer despite disable (Bit3) - Bit 7: Create tool using the T number - Bit 8: Hide Relocate tool - Bit 9: Hide Position magazine - Bit 10: Reactivate tool using Position magazine - Bit 11: Reactivate tool in all monitoring modes - Bit 12: Fade out Reactivate tool
52271	$MCS_TM_MAG_PLACE_DISTANCE	70		Distance between individual magazine locations
52274	$MCS_TM_LOAD_STATION	2		Number of Loading stations
52281	$MCS_TOOL_MCODE_FUNC_ON	-1		M code for tool-specific function ON (for ShopMill)
52282	$MCS_TOOL_MCODE_FUNC_OFF	-1		M code for tool-specific function OFF (for ShopMill)
54215	$SNS_TM_FUNCTION_MASK_SET	0H		Function mask Tool management - Bit 0: Diameter display for rotating tools - Bit 1: Default direction of rotation for all turning tools is M4 - Bit 2: Create tool without name proposal - Bit 3: Input disable for tool name and tool type of loaded tools - Bit 4: Input disable for loaded tools, if the channel is not in Reset - Bit 5: Calculate tool wear inputs additively - Bit 6: Numerical input of the tool ident - Bit 7: Fade out tool monitoring parameter - Bit 8: Diameter display for transverse axis - Geometry - Bit 9: Diameter display for transverse axis - Wear - Bit 10: Enable loading/relocating of tools on buffer locations - Bit 11: Creation of new tools in gripper locations is disabled - Bit 12: Do not unload measuring tools if ”Unload all” function is executed. - Bit 13: Do not delete tool wear when inputting the geometry value - Bit 14: Load/relocate tool. Search without pre-assignment of the magazine
55212	$SCS_FUNCTION_MASK_TECH_SET	Bit 0 = 1		Function mask Cross-technology - Bit 0: Tool preselection active (for ShopMill)

Notes:

Overview of the Magazine data and Tool data

Magazine data

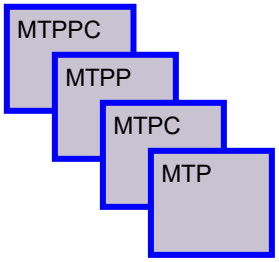


Description:

- MAP: Magazine description data
- MPAC: Magazine user data
- MPP: Magazine location data
- MPPC: Magazine location user data
- MPTH: Magazine location type hierarchy
- MDP: Distance to change location
- MAMP: Magazine block data
- MLSR: Magazine location assignment to spindle

The magazine data are only available when the magazine management is active.

Multitool data

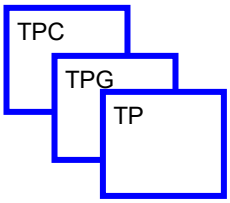


Description:

- MTP: Multitool description data
- MTPC: Multitool user data
- MTPP: Multitool location data
- MTPPC: Multitool location user data

The Multitool data are only available when the magazine management is active.

Tool data

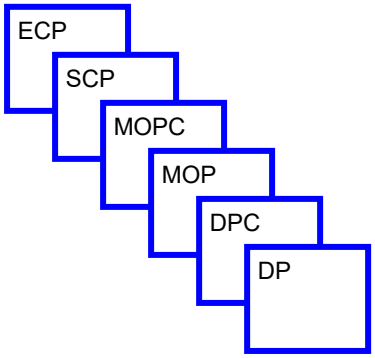


Description:

- TP: Tool related data
- TPG: Tool related grinding data
- TPC: Tool related user data

Note: The Tool data are only available when tool monitoring is active.

Cutting edge data



Description:

- DP: Cutting edge data
- DPC: Cutting edge user data
- MPO: Cutting edge monitoring
- MOPC: User cutting edge monitoring
- SCP: Location dependent offsets fine
- ECP: Location dependent offsets course

Notes:

Description of the Magazine and Tool Data

Magazine data

Parameter	Default	Description
Magazine block data (MAMP)		
$TC_MAMP1	" "	Identifier of the configuration magazine
$TC_MAMP2	0	Type of tool search (Bit 0-7) and Art type of empty location search (Bit 8-15) - Bit 0: active tool/ lowest Duplo - Bit 1: shortest distance - Bit 2: active tool/ min. $TC_TP10 - Bit 3: monitor for lowest actual value - Bit 4: monitor for largest actual value - Bit 5: monitor for $AC_MONMIN - Bit 6: first current magazine (Bit 7=1) - Bit 7: complex search in magazines - Bit 8: first location forward - Bit 9: current location forward - Bit 10: last location backward - Bit 11: current location backward - Bit 12: symmetric actual location - Bit 13: 1:1 Change - Bit 14: Hierarchies - Bit 15: Location type hierarchies
$TC_MAMP3	0	Procedure for tools in a wear group
Magazine description data (MAP)		
$TC_MAP1	0	Type of magazine (1: Chain, 3: Turret, 5: Box magazine, 7: Buffer magazine, 9: Loading magazine)
$TC_MAP2	" "	Identifier of magazine
$TC_MAP3	2	Status of magazine - Bit 0: active magazine - Bit 1: disabled - Bit 2: magazine is at load position - Bit 3: Tool motion is active - Bit 4: Magazine may be moved - Bit 5: Reserved - Bit 6: Magazine is fixed location coded - Bit 7: Reserved - Bit 8: Edge location must not be overlapped left - Bit 9: Edge location must not be overlapped right - Bit 10: Edge location must not be overlapped top - Bit 11: Edge location must not be overlapped bottom
$TC_MAP6	1	Number of tiers (box magazine only)
$TC_MAP7	0	Number of columns
$TC_MAP8	0	Current magazine position relative to change position
$TC_MAP9	0	Current wear group number
$TC_MAP10	0	Current tool search strategies of magazine, see MAMP2.
Magazine user data (MAPC)		
$TC_MAPC1 to $TC_MAPC10	0	Magazine user data 1..10

Notes:

Magazine data

Parameter	Default	Description
Magazine location data (MPP)		
$TC_MPP1	0	Location type (1: Magazine location, 2: Spindle/tool holder, 3: Gripper, 4: Loader, 5: Transfer location, 6: Loading station, 7: Loading point)
$TC_MPP2	9999	Location type (0: every tool fits in this location)
$TC_MPP3	FALSE	Consider adjacent location(1=on, 0=off)
$TC_MPP4	1	Location state - Bit 0: Disabled - Bit 1: Free to accept a tool - Bit 2: Reserved for tool from buffer - Bit 3: Reserved for new tool to be loaded - Bit 4: Occupied in left half location - Bit 5: Occupied in right half location - Bit 6: Occupied in upper half location - Bit 7: Occupied in lower half location - Bit 8: left half location reserved - Bit 9: right half location reserved - Bit 10: top half location reserved - Bit 11: lower half location reserved - Bit 12: Wear group disabled - Bit 13: Overlapping permitted
$TC_MPP5	0	Location type index
$TC_MPP6	0	T No. of tool at this location
$TC_MPP7	0	No. of adapter in magazine location
$TC_MPP_SP	0	For tool speed and acceleration monitoring
$TC_MPP66	0	No. of the tool, for which the identification “reserved for tool in buffer“ is set
Magazine location user data (MPPC)		
$TC_MPPC1 to $TC_MPPC10	0	Magazine location user data 1..10
Magazine location type hierarchy (MPTH)		
$TC_MPTH1 to $TC_MPTH7	9999	Magazine location type hierarchy 1..7
Distance to change location (MDP)		
$TC_MDP1	9999	Distance between tool change position of magazine and location of first internal magazine (loading magazine 9999)
$TC_MDP2	9999	Distance between tool change position of magazine and location of second internal magazine (buffer magazine 9998)
Magazine location assignment to spindle (MLSR)		
$TC_MLSR	9999	System variable for assigning magazine locations of the buffer magazine to the spindle

Notes:

Multitool data

Parameter	Default	Description
Multitool description data (MTP)		
$TC_MTPN	0	Number of locations
$TC_MTP2	“MT-Nr.“	Multitool identifier
$TC_MTP3	1	Size on left
$TC_MTP4	1	Size on right
$TC_MTP5	1	Size on above
$TC_MTP6	1	Size on below
$TC_MTP7	9999	Magazine location type
$TC_MTP8	0	Multitool status - Bit 0: active tool - Bit 1: enabled - Bit 2: disabled - Bit 3: measure - Bit 4: prewarning limit reached - Bit 5: Multitool being changed - Bit 6: fixed location coded - Bit 7: tool was in use - Bit 8: identifier for tools in buffer - Bit 9: ignore disabled state - Bit 10: to be unloaded - Bit 11: to be loaded - Bit 12: master tool - Bit 13: reserved - Bit 14: 1:1 Change - Bit 15: hand tool
$TC_MTP_POS	0	Multitool position
$TC_MTP_KD		Type of distance coding (1: Location number, 2: Distance, 3: Angle)
Multitool user data (MTPC)		
$TC_MTPC1 to $TC_MTPC10	0	Multitool user data 1..10
Multitool location data (MTPP)		
$TC_MTPP2	0	Multitool location type
$TC_MTPP4	0	Multitool location state
$TC_MTPP6	0	T number of tool at this location
$TC_MTPP7	0	Adapter number of adapter at this location
$TC_MTPPL	0	Distance from reference location, distance “length“
$TC_MTPPA	0	Distance from reference location, distance “angle“
Multitool location user data (MTPPC)		
$TC_MTPPC1 to $TC_MTPPC10	0	Multitool location user data 1..10

Notes:

Tool data

Parameter	Default	Description
Tool related data (TP)		
$TC_TP1	T-Nr.	Duplo number
$TC_TP2	“T-Nr.“	Tool identifier
$TC_TP3	1	Size on left
$TC_TP4	1	Size on right
$TC_TP5	1	Size on above
$TC_TP6	1	Size on below
$TC_TP7	9999	Magazine location type
$TC_TP8	0	Tool status - Bit 0: active tool - Bit 1: enabled - Bit 2: disabled - Bit 3: measure - Bit 4: prewarning limit reached - Bit 5: tool being changed - Bit 6: fixed location coded - Bit 7: tool was in use - Bit 8: identifier for tools in buffer - Bit 9: ignore disabled state - Bit 10: to be unloaded - Bit 11: to be loaded - Bit 12: master tool - Bit 13: reserved - Bit 14: 1:1 Change - Bit 15: hand tool
$TC_TP9	0	Tool monitoring method - Bit 0: tool life - Bit 1: workpiece count - Bit 2: wear monitoring active - Bit 3: wear monitoring, location dependent offset fine active
$TC_TP10	0	Replacement change strategy
$TC_TP11	0	Tool information
Tool related user data (TPC)		
$TC_TPC1 to $TC_TC10	0	Tool related user data 1..10

Notes:

Cutting edge data

Parameter	Default	Description
Cutting edge data (DP)		
$TC_DP1	9999	Tool type
$TC_DP2	0	Cutting edge position
$TC_DP3	0	Geometry length 1
$TC_DP4	0	Geometry length 2
$TC_DP5	0	Geometry length 3
$TC_DP6	0	Geometry radius
$TC_DP7	0	Geometry corner radius (tool type 700 slotting saw)
$TC_DP8	0	Geometry length 4 (tool type 700 slotting saw)
$TC_DP9	0	Geometry length 5
$TC_DP10	0	Geometry angle 1
$TC_DP11	0	Geometry angle 2 for conical milling tools
$TC_DP12	0	Wear length 1
$TC_DP13	0	Wear length 2
$TC_DP14	0	Wear length 3
$TC_DP15	0	Wear radius
$TC_DP16	0	Wear slot width/rounding radius
$TC_DP17	0	Wear length 4
$TC_DP18	0	Wear length 5
$TC_DP19	0	Wear angle 1
$TC_DP20	0	Wear angle 2 for conical milling tools
$TC_DP21	0	Adapter length 1
$TC_DP22	0	Adapter length 2
$TC_DP23	0	Adapter length 3
$TC_DP24	0	Clearance angle
$TC_DP25	0	For ShopMill/ShopTurn: A bit-coded value for various states of tools is stored here for ShopMill and ShopTurn (turning direction, coolant 1/2, number of teeth)
Cutting edge user data (DPC)		
$TC_DPC1 to $TC_DPC10	0	Cutting edge user data 1..10

Notes:

Example: Milling machine with chain magazine

The tool change is done at the change location, which is specified in the magazine configuration.

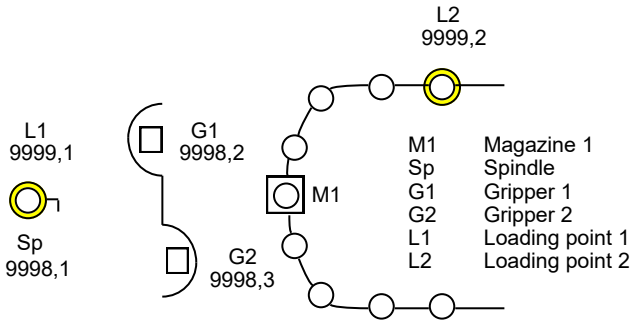
DB72 contains an interface area for each change location which has been configured.

Each interface is set up to handle these following Tool Management situations:

- Prepare tool change
- Execute tool change (triggered via M06)

In order to inform the Tool Management of tool movements the following functions are used in the machine PLC program FC6 (Multitool) and FC8. Standard end acknowledgment via DB72 is also available.

Example: Chain magazine with double gripper



Procedure:

- Setup the machine data for the Tool Management
- Configure the chain magazine
- Tool load / unload
- Generate the tool change program L6
- Tool change sequence
  - Prepare tool change
  - Tool change sequence
  - Hand tools
  - Coordination between hand and magazine tools
  - Interruption to the tool change
  - Recovery functions after an interruption

Notes:

Machine data for Tool management

Overview of the machine data

MD	Identifier	Settings	Comment	Description
10715	$MN_M_NO_FCT_CYCLE	6		M function to be replaced by a subroutine
10716	$MN_M_NO_FCT_CYCLE_NAME	L6		Subroutine name for M function replacement
17500	$MN_MAXNUM_REPLACEMENT_TOOLS	0		Maximum number of replacement tools
17530	$MN_TOOL_DATA_CHANGE_COUNTER	1FH		Mark tool data change for HMI
18080	$MN_MM_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 10 = 1		Memory reservation for tool management (SRAM) - Bit 0: Provide magazine management data - Bit 1: Provide monitoring data - Bit 3: Consider adj. Location - Bit 10: The multitool function is available
18082	$MN_MM_NUM_TOOL	100		Number of tools the NCK can manage (SRAM)
18083	$MN_MM_NUM_MULTITOOL	15		Number of multitools the NCK can manage (SRAM)
18084	$MN_MM_NUM_MAGAZINE	3		Number of magazines the NCK can manage (SRAM)
18085	$MN_MM_NUM_MULTITOOL_LOCATIONS	30		Number of multitool locations the NCK can manage (SRAM)
18086	$MN_MM_NUM_MAGAZINE_LOCATION	35		Number of magazine locations the NCK can manage (SRAM)
18100	$MN_MM_NUM_CUTTING_EDGES_IN_TOA	200		Tool offsets in the TO range (SRAM)
20124	$MC_TOOL_MANAGEMENT_TOOLHOLDER	1		Tool holder number
20310	$MC_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 14 = 1 Bit 24 = 1		Activation of tool management functions - Bit 0: Magazine management active - Bit 1: Monitoring functions active - Bit 3: Adjacent location treatm. - Bit 14: Automatic tool change during RESET and Start - Bit 24: Command No. 8 and 9 ignore location reservation
20320	$MC_TOOL_TIME_MONITOR_MASK	Bit 0 = 1		Time monitoring for tool in tool holder - Bit 0: Monitoring tool in tool holder 1
22550	$MC_TOOL_CHANGE_MODE	1		New tool offset with M function
22560	$MC_TOOL_CHANGE_M_CODE	206		M function for tool change
22562	$MC_TOOL_CHANGE_ERROR_MODE	Bit 1 = 1		Response to tool change errors - Bit 1: Manual tools permitted
28450	$MC_MM_TOOL_DATA_CHG_BUFF_SIZE	400		Buffer for tool data changes (DRAM)

Notes:

Set up the machine data for the tool management

The various Tool Management functions are activated via machine data.

Prerequisite:

- License key for the “Tool Management, more than 4 Magazines“ option is available (if necessary).

Procedure:

- Activate the Option “Tool Management, more than 4 Magazines“ (if necessary)
- Activate Tool Management
- Set the number of magazines and magazine locations
- Set the number of tools and the number of cutting edges
- Set the number of multitools

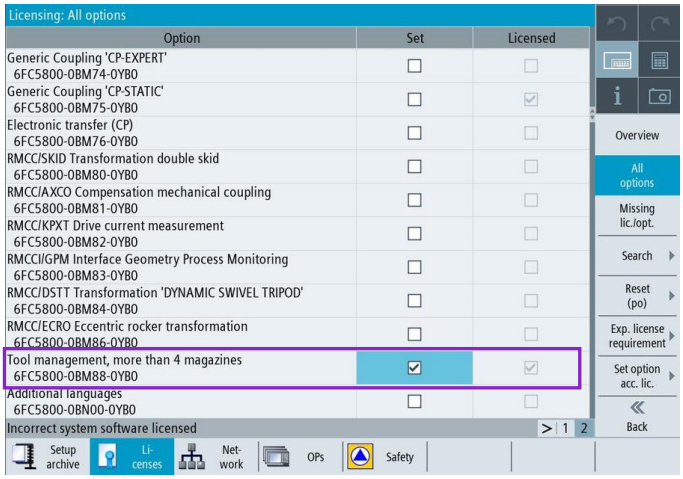
Activate the option “Tool Management, more than 4 magazines“ (if necessary)

The standard function includes 4 magazines. Buffers and loading magazines count as magazines.

Magazine 1, 2
Magazine 3 (Option)
Buffer magazine (9998)
Load magazine (9999)

The option is activated in the menu “Licenses“ in the “Setup“ area . A valid license key is required.

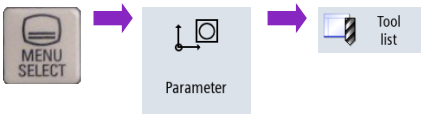


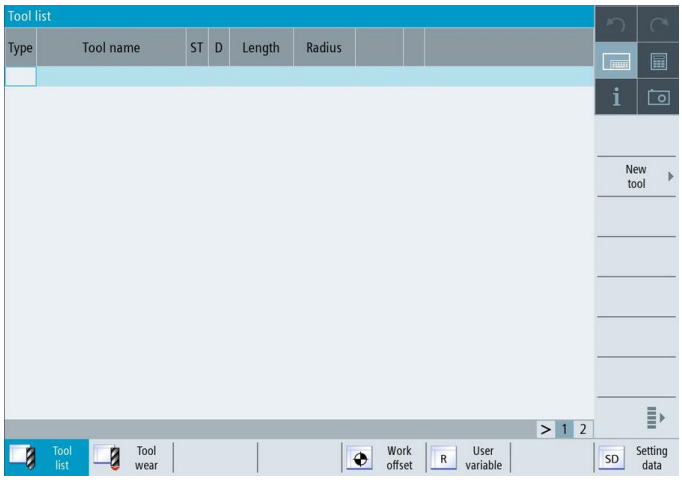


Notes:

Activate Tool Management

By default the Tool Management without magazine management is active, i.e. only tool monitoring is active (MD18080 Bit 1=1, MD20310 Bit 1=1).





Enable Tool Management

Tool Management with magazine management requires these minimum machine data settings.

Reserve memory for Tool Management

MD18080	Memory organisation for Tool Management
Bit 0 = 1	Reserve memory for magazine management
Bit 1 = 1	Reserve memory for tool monitoring
Bit 3 = 1	Reserve memory for adjacent location check
Bit 10 = 1	Multitool function is enabled

Note:
MD18080 and MD20310 Bits 0-3 must always be the same.

Activate Tool Management

MD20310	Activation of Tool Management
Bit 0 = 1	Magazine management active
Bit 1 = 1	Tool monitoring active
Bit 3 = 1	Adjacent location check

Check the HMI settings for changes to tool data

MD17530	Tool data change counter
Bit 0 = 1	Change tool status
Bit 1 = 1	Change residual number of tools
Bit 2 = 1	Tool data in update service
Bit 3 = 1	Magazine data in update service
Bit 4 = 1	ISO data in update service

Buffer for changes to tool data

MD28450 = 400	Number of entries in the buffer for HMI update. This MD concerns how the HMI shows data of tools, whose data is changed via part program instructions (e.g. NEWT).

Activation of the tool cutting time monitor

MD20320	Cutting time monitor for tool in tool holder
Bit 0 = 1	Cutting time monitor for tool in tool holder 1

Assigning the TO Blocks

MD28085 = 1	Assignment of TO blocks to channels The TO area includes all tool and magazine data known to the NCK. The maximum number of TO blocks is equal to the number of channels currently enabled. 1, 1: TO block 1 is assigned to channels 1 and 2. 1, 2: TO block 1 is assigned to channels 1, TO block 2 is assigned to channels 2

Notes:

Define the number of magazines and magazine locations

MD18084 = 3	Number of magazines. Note: Min. 3 magazines incl. buffer and load magazines.
MD18086 = 35	Number of magazine locations. Note: Add together all locations in all magazines.

Example: Chain magazine with double gripper

MD18084 = 3
- Chain magazine (magazine 1)
- Buffer magazine (magazine 9998)
- Load magazine (magazine 9999)

MD18086 = 35
- 30 locations for chain magazine
- 3 locations for buffer magazine (gripper 1/2 and spindle)
- 2 locations for Load magazine (load location at chain magazine and spindle)

Note:
Assigning the magazines to the locations is done in the magazine configuration.

Define the number of Multitools

MD18083 = 15 (Default)	Number of Multitools
MD18085 = 30 (Default)	Number of Multitool locations
MD17504 = 6 (Default)	Max. locations per Multitool



Example: Multitool holder with three tools

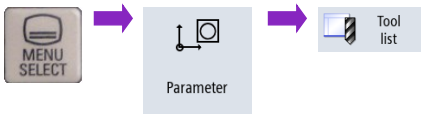
Define the max. number of tools and cutting edges

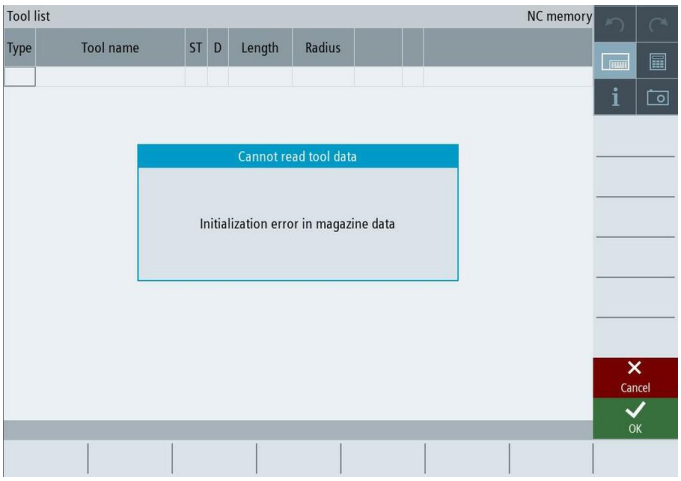
MD18082 = 100	Max number of tools which the NC can handle. Note: Each tool has at least 1 cutting edge.
MD18100 = 200	Number of cutting edges in the TO block

Suggested values for JobShop applications

MD18082 = 100 Number of tools
MD18100 = 200 Number of edges

Once Tool Management has been successfully activated, the magazine data have to be configured. The following alarm indicated this.





Magazine data missing in the NC.

Notes:

Magazine Configuration

The magazine configuration can be accomplished with the operating software “SINUMERIK Operate“.

The magazine configuration contains:

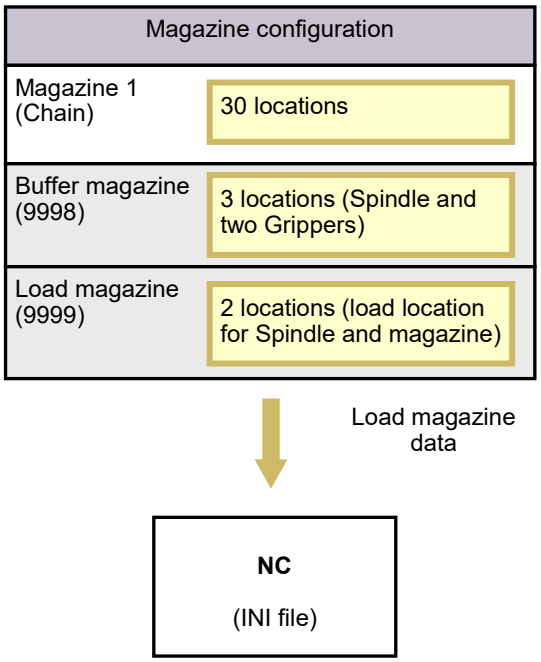
- Definition of the magazines
- Definition of the magazine locations
- Definition of the buffer locations
- Definition of the load locations
- Definition of the search strategy

Procedure:

- Create the magazine configuration
- Load magazine data into the NC

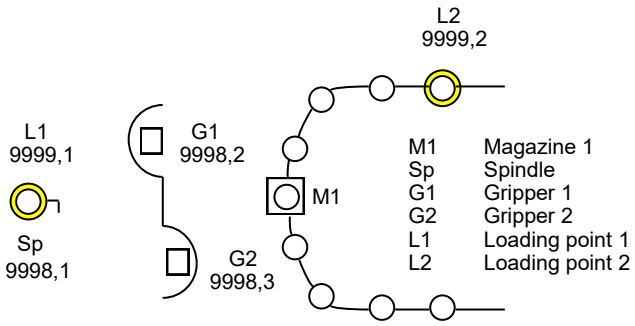
Magazine configuration: how to create, how to load

Example: Chain magazine (30 locations) with double gripper



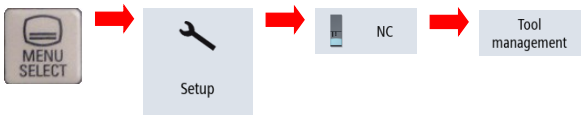
Configure the chain magazine

Example: Chain magazine (30 locations) and double gripper



Notes:

Start the magazine configuration





The Tool management menu contains the following areas:
Active data contains the tool management of the NC
Examples contains examples for the tool management
Network / USB contains the connections

Create a new magazine configuration

Position the cursor on the "Active data" folder and press the "New element" softkey.

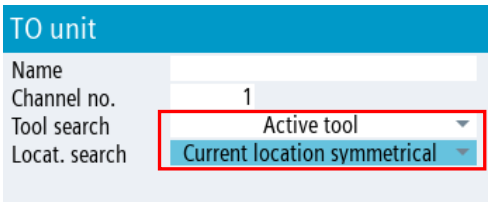




Configure the TO unit

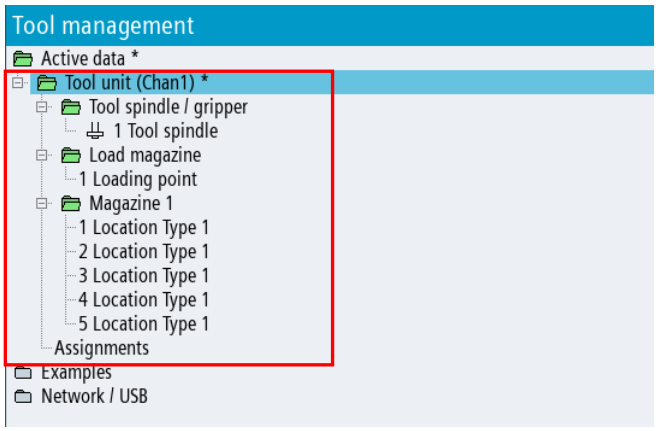
Set the search strategy and accept with "OK".

Channel number input a value of “1”
Tool search select “Active tool”
Location search select “Current location symmetrical”





A magazine configuration is created that consists of a magazine with 5 locations, as well as a tool spindle and a loading point.

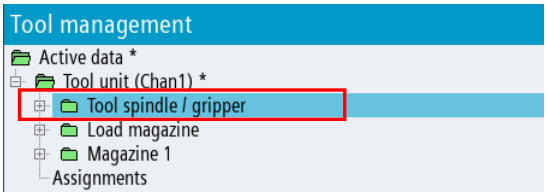


This magazine configuration can now be adapted to the conditions on the machine.

Notes:

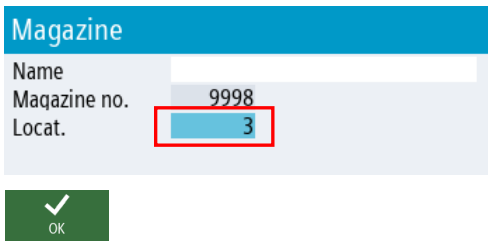
Configure the tool spindle / grippers

Position the cursor on the "Tool spindle / gripper" folder and open the element with the "Input" key.

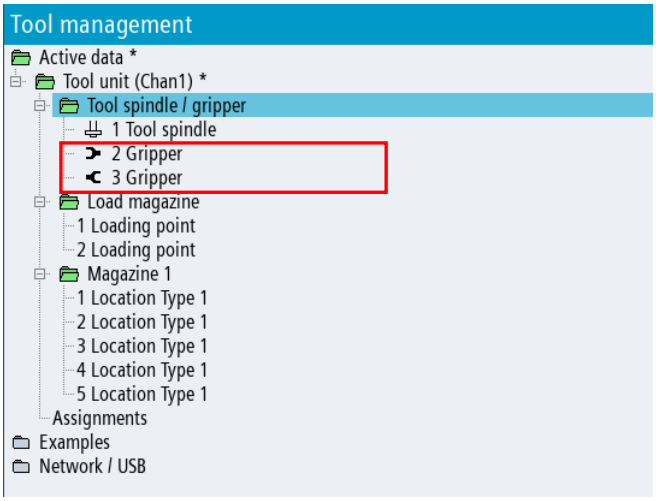


Set the number of locations and accept with "OK".

Name optional
Magazine no. 9998 (not changeable)
Locations input a value of “3”



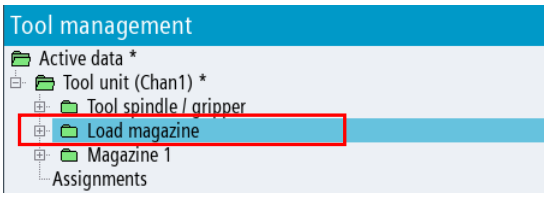
Two new locations of the Gripper type are added to the buffer.



The location type can be defined by the OEM, e.g. to spindle or transfer location. To do this, the cursor must be positioned on the appropriate location and the element opened.

Configure the load magazine

Position the cursor on the "Load magazine" folder and open the element with the "Input" key.

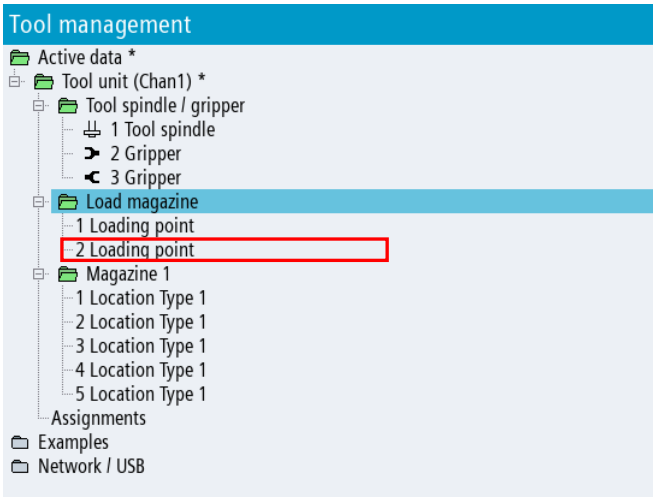


Set the number of locations and accept with "OK".

Name optional
Magazine no. 9999 (not changeable)
Locations input a value of “2”



A new location of the Loading point type is added to the load magazine.

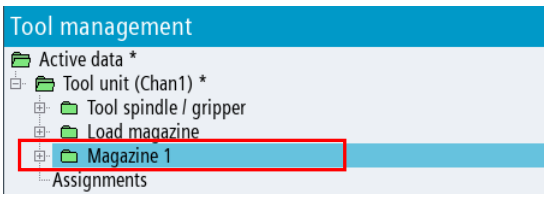


The location type can be defined by the OEM, e.g. to loading station. To do this, the cursor must be positioned on the appropriate location and the element opened.

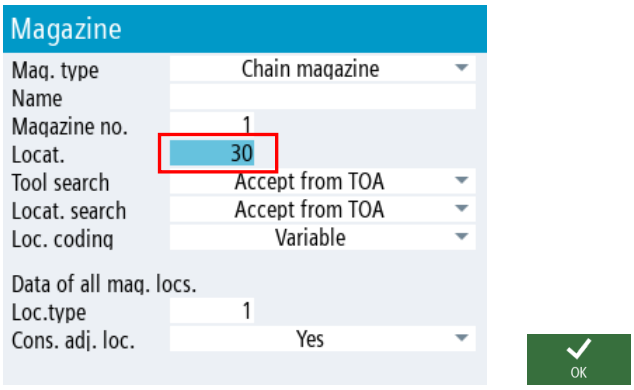
Notes:

Configure Magazine 1

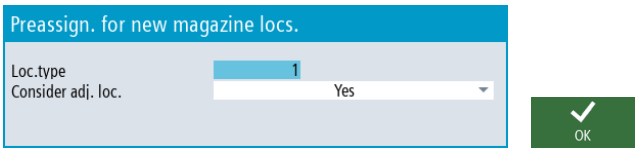
Position the cursor on the "Magazine 1" folder and open the element with the "Input" key.



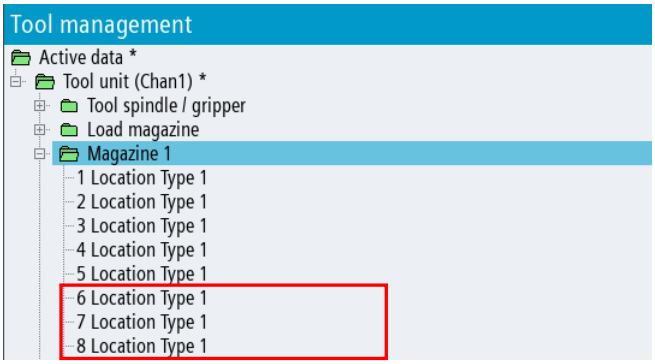
Set the number of locations and accept with "OK".



Accept the preassignments for the new magazine locations for Location type and Consider adjacent location with "OK".



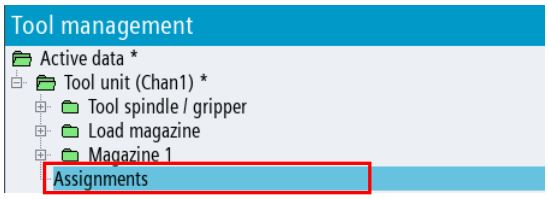
The new locations are added to the magazine.



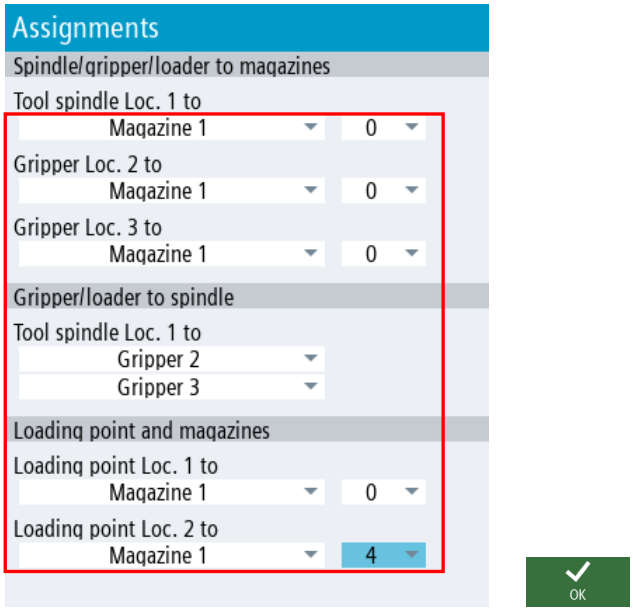
The location type can be defined by the OEM. Therefore, the cursor must be positioned on the appropriate location and the element opened.

Configure the assignments

Position the cursor on the "Assignments" folder and open the element with the "Input" key.



Specify the assignments of the magazine and accept with "OK".



The adaption of the magazine configuration is now completed.

Notes:

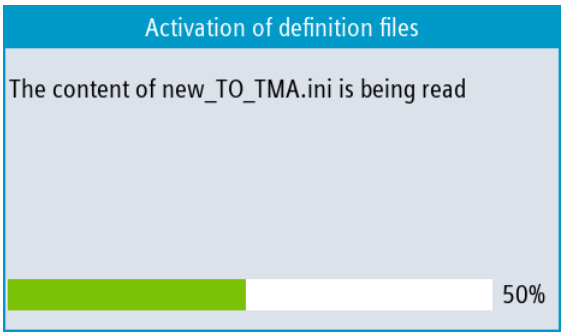
Activate the data

Position the cursor on the "Active data" folder and press the "Activate data" softkey.



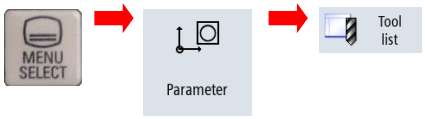


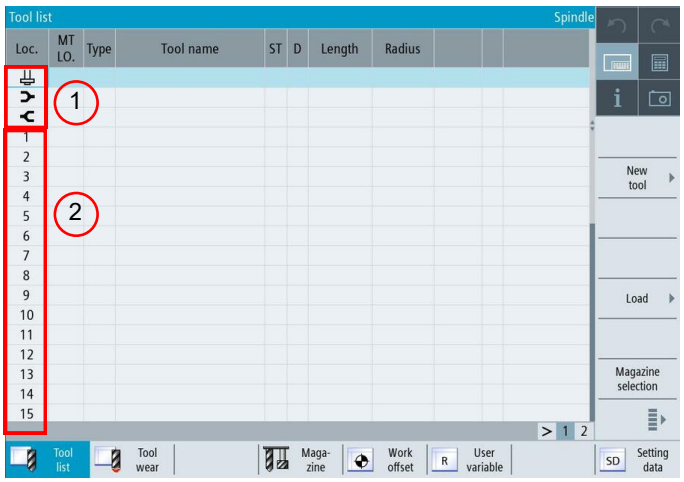
The definition files are read in.



After successfully loading the configuration file, all magazines and buffers in the tool list are displayed in the Parameters area.

Display on the controller



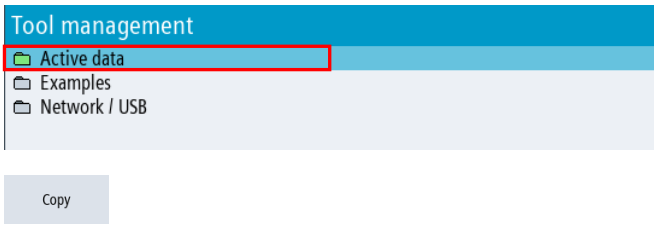


1. Spindle and gripper 1/2 is displayed
2. Chain magazine with 30 locations is displayed

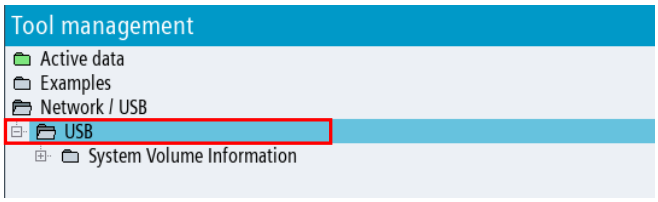
Notes:

Back up the data

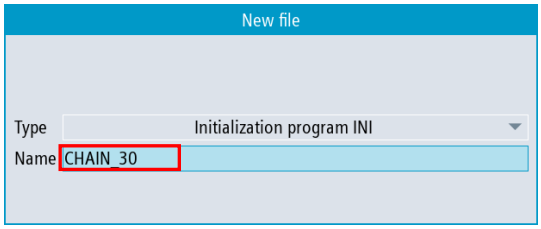
Position the cursor on the "Active data" folder and press the "Copy" softkey.



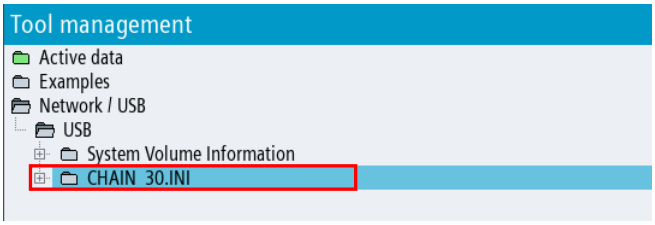
Position the cursor on the "USB" folder and press the "Paste" softkey.



Enter the name for the INI file and accept with "OK".



The INI file is created and saved to the USB stick.



Modifications to the magazine configuration

Activate search strategy 1:1 change

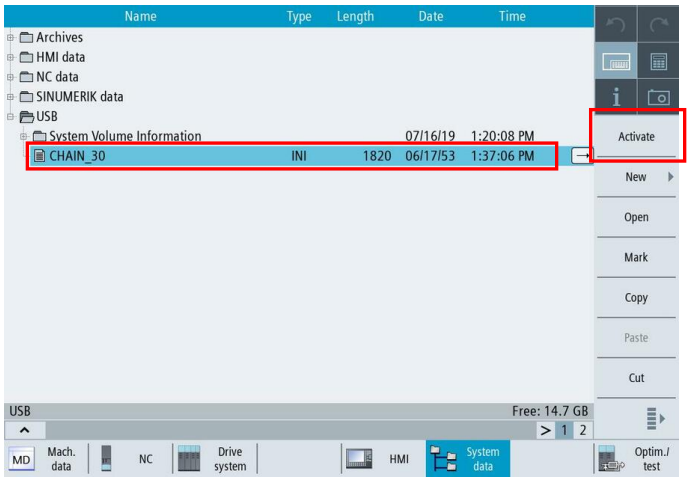
Search strategy 1:1 change can set via $TC_MAMP2 und $TC_MAP10. In each case bit 13=1.
The configuration file (INI file) must be changed with a text editor.

```
...
$TC_MAMP2=4097
...
$TC_MAP10[1]=4097
...
```

```
...
$TC_MAMP2=12289
...
$TC_MAP10[1]=12289
...
```

New search strategy:
- Bit 0=1: active tool/ min. Duplo
- Bit 12=1: symmetric actual location
- **Bit 13=1: 1:1 exchange - old tool /new tool swap**

Activate the modified configuration file (INI file) via the “Activate” softkey.



Note:

The HMI commissioning mask only allows to activate one search strategy, in the example bevor the “Current location symmetrical” location search was activated via the HMI mask, so the 1:1 exchange must be activated manually.

Notes:

Configuration file (INI file)

```
CHANDATA(1)

; delete old data
$TC_MAP1[0]=0
$TC_DP1[0,0]=0

$TC_MAMP1=""
$TC_MAMP2=12289
$TC_MAMP3=0

$TC_MAP1[1]=1
$TC_MAP2[1]=""
$TC_MAP3[1]=16
$TC_MAP6[1]=1
$TC_MAP7[1]=30
$TC_MAP10[1]=12289

$TC_MPP1[1,1]=1
$TC_MPP2[1,1]=1
$TC_MPP3[1,1]=1
$TC_MPP4[1,1]=2
$TC_MPP5[1,1]=0
...
$TC_MPP1[1,30]=1
$TC_MPP2[1,30]=1
$TC_MPP3[1,30]=1
$TC_MPP4[1,30]=2
$TC_MPP5[1,30]=0

$TC_MAP1[9998]=7
$TC_MAP2[9998]=""
$TC_MAP3[9998]=16
$TC_MAP6[9998]=1
$TC_MAP7[9998]=3
$TC_MAP10[9998]=0

$TC_MPP1[9998,1]=2
$TC_MPP2[9998,1]=0
$TC_MPP3[9998,1]=0
$TC_MPP4[9998,1]=2
$TC_MPP5[9998,1]=1
$TC_MPP1[9998,2]=3
$TC_MPP2[9998,2]=0
$TC_MPP3[9998,2]=0
$TC_MPP4[9998,2]=2
$TC_MPP5[9998,2]=1
$TC_MPP1[9998,3]=3
$TC_MPP2[9998,3]=0
$TC_MPP3[9998,3]=0
$TC_MPP4[9998,3]=2
$TC_MPP5[9998,3]=2

$TC_MAP1[9999]=9
$TC_MAP2[9999]=""
$TC_MAP3[9999]=16
$TC_MAP6[9999]=1
$TC_MAP7[9999]=2
$TC_MAP10[9999]=0

$TC_MPP1[9999,1]=7
$TC_MPP2[9999,1]=0
$TC_MPP3[9999,1]=0
$TC_MPP4[9999,1]=2
$TC_MPP5[9999,1]=1
$TC_MPP1[9999,2]=7
$TC_MPP2[9999,2]=0
$TC_MPP3[9999,2]=0
$TC_MPP4[9999,2]=2
$TC_MPP5[9999,2]=2

$TC_MLSR[2,1]=0
$TC_MLSR[3,1]=0

$TC_MDP1[1,1]=0
$TC_MDP1[1,2]=4

$TC_MDP2[1,1]=0
$TC_MDP2[1,2]=0
$TC_MDP2[1,3]=0

M17
```

Delete old data

TO area

Magazine 1

Location 1

to

Location 30

Buffer magazine

Spindle

Gripper 1

Gripper 2

Load magazine

Loading point 1

Loading point 2

Assignments Grippers to spindle

Loading points to magazine

Spindle/grippers to magazine

Variables in the configuration file (INI file)

Parameter	Meaning
$TC_MAMP1	Configuration magazine identifier
$TC_MAMP2	Tool search (bit 0-7) and location search (bit 8-15) - Bit 0=1: Active tool / min. duplo - Bit 12=1: symmetric actual location - Bit 13=1: 1:1 exchange - old tool /new tool swap
$TC_MAMP3	Procedure for tools in a wear group
$TC_MAP1	Magazine type (1=chain, 7=tool buffer magazine, 9=load magazine)
$TC_MAP2	Magazine identifier
$TC_MAP3	Magazine state - Bit 4=1: Magazine can be moved
$TC_MAP6	Number of lines (box magazine only)
$TC_MAP7	Number of columns
$TC_MAP10	Magazine tool search strategies, see MAMP2
$TC_MPP1	Location category (1=magazine location, 2=Spindle /tool holder, 3=gripper, 7=loading point)
$TC_MPP2	Location type (0=compatible with each tool location type)
$TC_MPP3	Consider adjacent location (1=On, 0=Off)
$TC_MPP4	Location state - Bit 1=1: Free to receive a tool
$TC_MPP5	Location type index
$TC_MLSR	Assignment of locations of the buffer magazine to the spindle
$TC_MDP1	Distance from the magazine change position to the location of the first internal magazine (9999)
$TC_MDP2	Distance from the magazine change position to the location of the second internal magazine (9998)

Notes:

Load / Unload

Tool load/unload is done via the loading location defined in the machine configuration.

Each loading location has it own interface area in DB71.

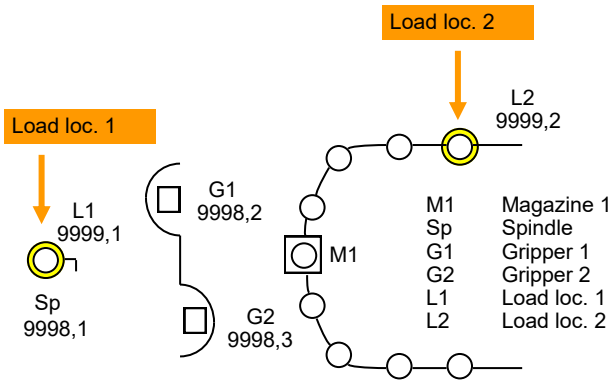
The interfaces are set up to handle the tool management commands: *Tool load*, *unload*, *Relocate* and *position magazine*.

The PLC functions FC6 (Multitool), FC8 and the Standard end acknowledgment with DB71 are used to communicate the current positions of tools.

Procedure:

- Assign loading locations.
- PLC program to handle the acknowledgments for the tool management commands.
  - Load/Unload magazine
  - Load/Unload spindle
  - Relocate tool
  - Position magazine/multitool
  - Interruption to the loading/unloading sequence

Example: Chain magazine with double gripper



Load location 1 is permanently assigned to the Spindle.

Load location 2 is assigned to the magazine via machine data 52274 = 2.

Assign loading locations

Load station Tool management

MD52274 $MCS_TM_TOOL_LOAD_STATION	Load station Tool management
	Change to: 2

Description:

If only one loading station is configured: The setting = 0

= 0: No query window appears on the surface

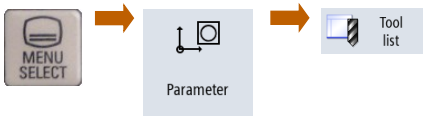
If more than 1 loading station is configured, you can select between

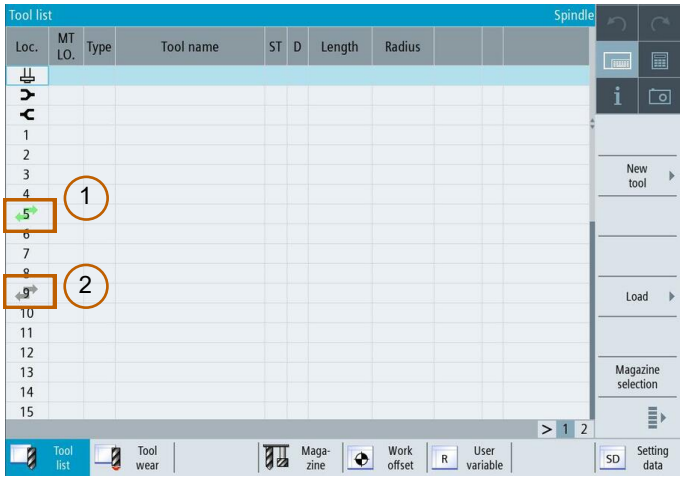
= 0: A query window appears on the surface to select the loading point.

= 1: fixed loading station 1, the window does not appear

= 2: fixed loading station 2, the window does not appear

Display on the controller





1. Magazine location on change position
2. Magazine location on loading point

Notes:

Acknowledgements for Load / Unload

PLC Programming example with acknowledgement using FC6

```
// Example "TMChain" [FB150]

// Variables for "LoadUnload"

// Input
//    loadIF2 : Bool;
//    unloadIF2 : Bool;
//    loadIF1 : Bool;
//    unloadIF1 : Bool;
//    relocateIF1 : Bool;
//    posMagIF2 : Bool;
//    posMtIF1 : Bool;

// Output
//    None

// InOut
//    None

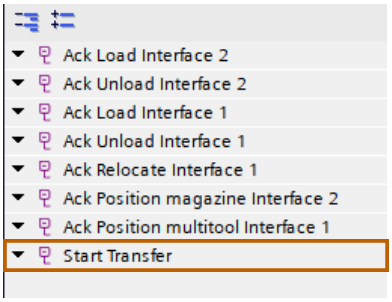
// Static
//    statLoadIF2Flag : Bool;
//    statUnloadIF2Flag : Bool;
//    statLoadIF1Flag : Bool;
//    statUnloadIF1Flag : Bool;
//    statRelocateIF1Flag : Bool;
//    statPosMagIF2Flag : Bool;
//    statPosMtIF1Flag : Bool;
```

```
//    statTask : Byte;
//    statNo : Byte;
//    statMagNew : Int;
//    statLocNew : Int;
//    statMagOld : Int;
//    statLocOld : Int;
//    statStatus : Int;
//    statLocMt : Int;
//    statStart : Bool;

// Temp
//    tempLoadIF2EdgeP : Bool;
//    tempUnloadIF2EdgeP : Bool;
//    tempLoadIF1EdgeP : Bool;
//    tempUnloadIF1EdgeP : Bool;
//    tempRelocateIF1EdgeP : Bool;
//    tempPosMagIF2EdgeP : Bool;
//    tempPosMtIF1EdgeP : Bool;
//    tempReady : Bool;
//    tempError : Int;

// Constant
//    None
```

PLC Programming example with acknowledgement using FC6



Start Transfer

```
REGION Start Transfer

    // Tool management status transfer
    "LBP_TMTransferMT"(Start := #statStart,
                       TaskIdent := #statTask,
                       TaskIdentNo := #statNo,
                       NewToolMag := #statMagNew,
                       NewToolLoc := #statLocNew,
                       OldToolMag := #statMagOld,
                       OldToolLoc := #statLocOld,
                       Status := #statStatus,
                       MultitoolLoc := #statLocMt,
                       Ready => #tempReady,
                       Error => #tempError);

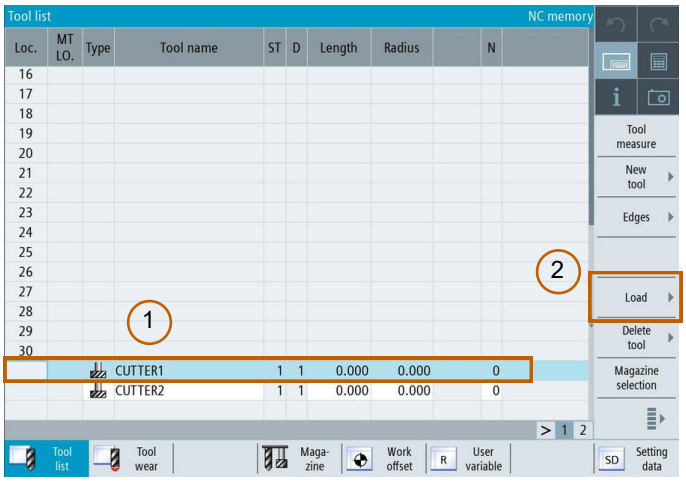
    // Reset Job Start
    IF #tempReady OR ("mcpIn".reset AND (#tempError <> 0)) THEN
        #statStart := FALSE;
    END_IF;

END_REGION
```

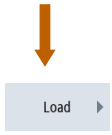
Notes:

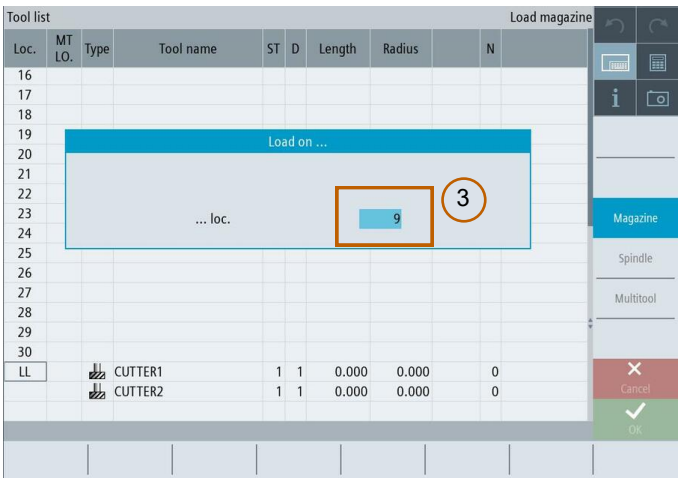
Loading into the magazine

Using the dialogue for “Load”, either a completely new tool can be created and loaded to the magazine or a tool out of the tool list can be transferred to a location in the magazine.



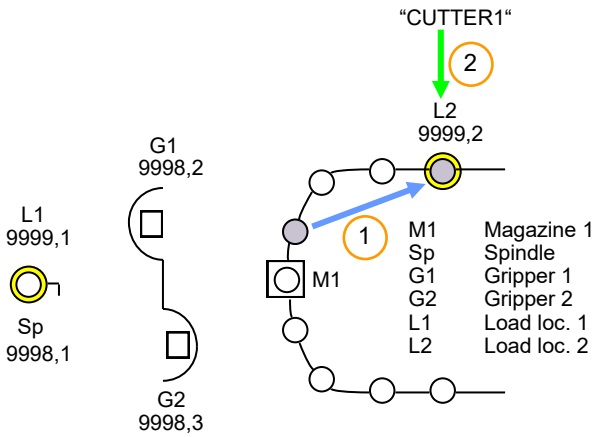
1. Select the tool.
2. Press the softkey “Load“.



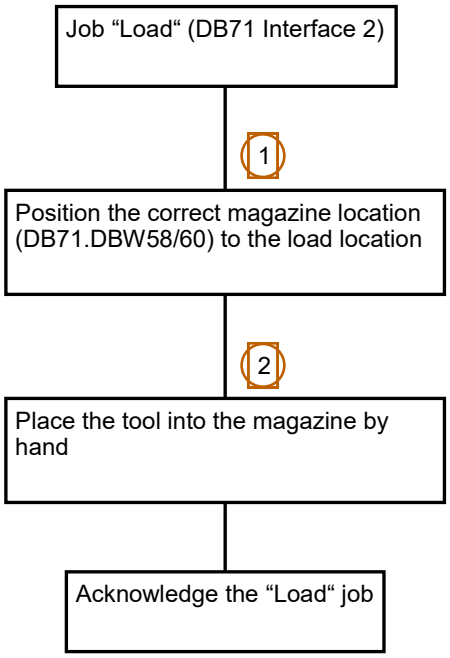


3. Select the desired location and confirm with OK -> Job to PLC (DB71.DBX34.0=1).

Example: Load a tool into the magazine



Sequence load magazine



Notes:

Job: Load magazine DB71

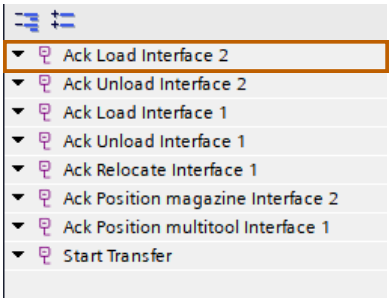
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB34	Re-served	Re-served	Posi-tioning Multi-tool	Order from NC-Pro-gram	Posi-tioning Mag-azine	Relo-cate	Un-load	Load
DBB35	Data in DB1071	Reserved						Ack. sta-tus 3
DBW50	Identification for loading station (fixed value 9999)							
DBW52	No. of loading station							
DBW54	Magazine no. (source) for unloading/reloading/positioning							
DBW56	Location no. (source) for unloading/reloading/positioning							
DBW58	Magazine no. (target) for loading/reloading/positioning							
DBW60	Location no. (target) for loading/reloading/positioning							

Acknowledgement for load the magazine with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer will be started
TaskIdent	1	DB71 Interface
TaskIdentNo	2	Interface 2
NewToolMag	DB71.DBW58	Destination Mag. for load
NewToolLoc	DB71.DBW60	Destination location for load
OldToolMag	0	When loading =0
OldToolLoc	0	When loading =0
Status	1	1: Process finished
MultitoolLoc	-1	When loading =-1

Example: Load tool to magazine location 9
DB71.DBW54 =0
DB71.DBW56 =0
DB71.DBW58 =1
DB71.DBW60 =9

PLC Programming example with acknowledgement using FC6



Load magazine

```
REGION Ack Load Interface 2

    // Scan rising edge
    #tempLoadIF2EdgeP := #loadIF2 AND NOT #statLoadIF2Flag;
    #statLoadIF2Flag := #loadIF2;

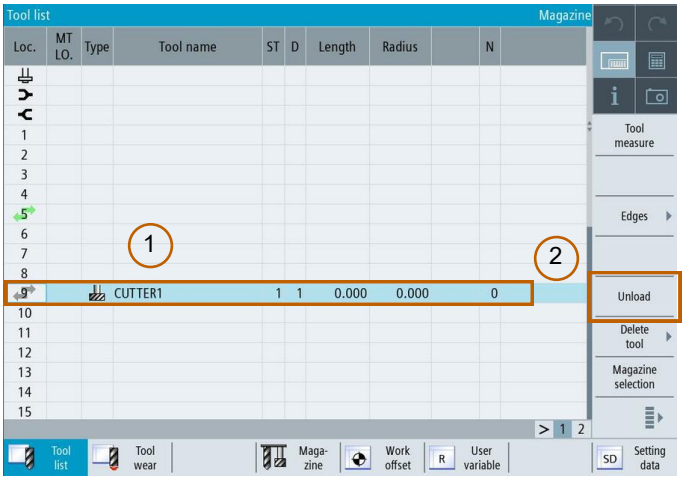
    // Set Job Start
    IF #tempLoadIF2EdgeP AND "LBP_TMLoad".IFNo[2] AND
        "LBP_TMLoad".TIF[2].Loading THEN
        #statTask := 1;                                  //DB71
        #statNo := 2;                                    //Interface 2
        #statMagNew := "LBP_TMLoad".TIF[2].TMag;
        #statLocNew := "LBP_TMLoad".TIF[2].TLoc;
        #statMagOld := 0;
        #statLocOld := 0;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

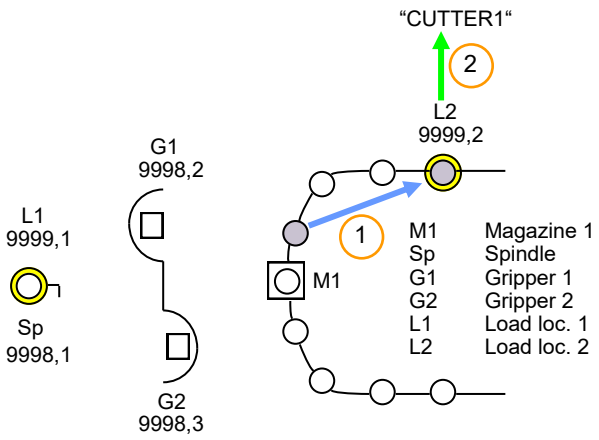
Unloading from the magazine

A tool can be removed from a location in the magazine with the ”Unload“ dialogue.

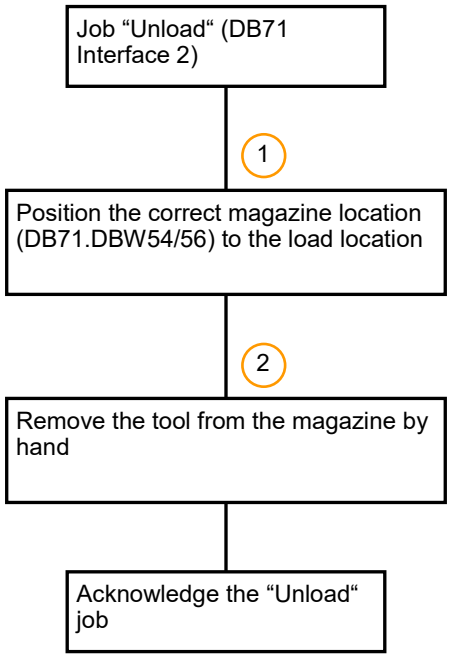


1. Select the magazine location.
2. Press the softkey “Unload“ -> Job to PLC (DB71.DBX34.1=1).

Example: Unload a tool from the magazine



Sequence unload magazine



Notes:

Job: Unload magazine DB71

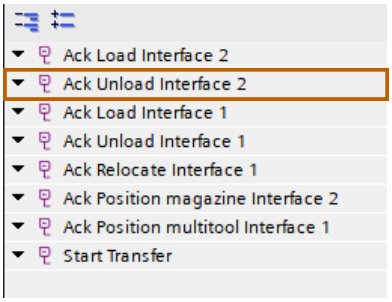
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB34	Re-served	Re-served	Posi-tion-ing Multi-tool	Order from NC-Pro-gram	Posi-tionin g Mag-azine	Relo-cate	Un-load	Load
DBB35	Data in DB10 71	Reserved						Ack Sta-tus 3
DBW50	Identification for loading station (fixed value 9999)							
DBW52	No. of loading station							
DBW54	Magazine no. (source) for unloading/reloading/positioning							
DBW56	Location no. (source) for unloading/reloading/positioning							
DBW58	Magazine no. (target) for loading/reloading/positioning							
DBW60	Location no. (target) for loading/reloading/positioning							

Acknowledgement for unload the magazine with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	1	DB71 interface
TaskIdentNo	2	Interface 2
NewToolMag	0	When unloading =0
NewToolLoc	0	When unloading =0
OldToolMag	DB71.DBW50	Source Magazine for unload
OldToolLoc	DB71.DBW52	Source Location for unload
Status	1	1: Process finished
MultitoolLoc	-1	When unloading =-1

Example: Unload tool from magazine location 9
DB71.DBW50 =9999
DB71.DBW52 =2
DB71.DBW54 =1
DB71.DBW56 =9
DB71.DBW58 =0
DB71.DBW60 =0

PLC Programming example with acknowledgement using FC6



Unload magazine

```
REGION Ack Unload Interface 2

    // Scan rising edge
    #tempUnloadIF2EdgeP := #unloadIF2 AND NOT #statUnloadIF2Flag;
    #statUnloadIF2Flag := #unloadIF2;

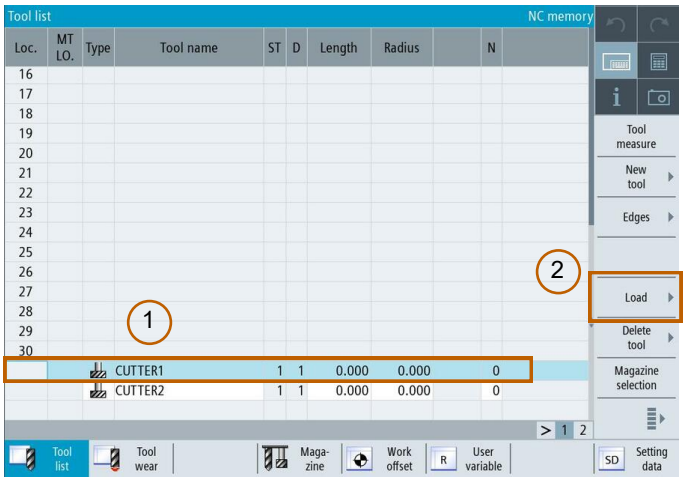
    // Set Job Start
    IF #tempUnloadIF2EdgeP AND "LBP_TMLoad".IFNo[2] AND
        "LBP_TMLoad".TIF[2].Unloading THEN
        #statTask := 1;                                  //DB71
        #statNo := 2;                                    //Interface 2
        #statMagNew := 0;
        #statLocNew := 0;
        #statMagOld := "LBP_TMLoad".TIF[2].LoadIdent;
        #statLocOld := "LBP_TMLoad".TIF[2].LoadNo;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

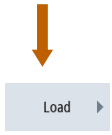
Notes:

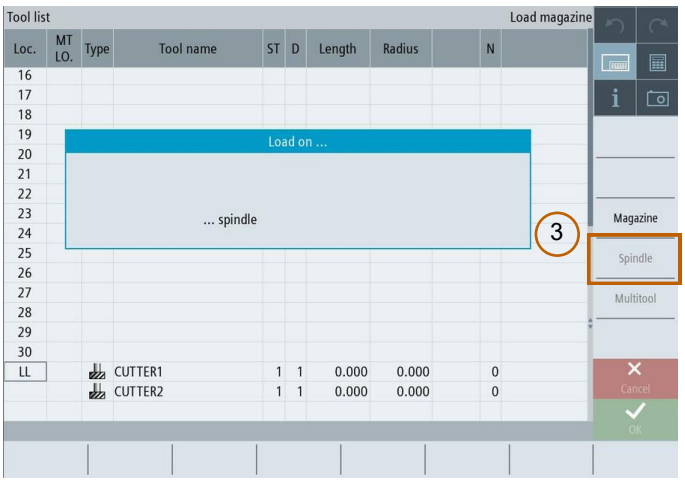
Loading a tool to the spindle

Using the dialogue for ”Load”, either a completely new tool can be created and loaded to the spindle or a tool out of the tool list can be transferred to the spindle.



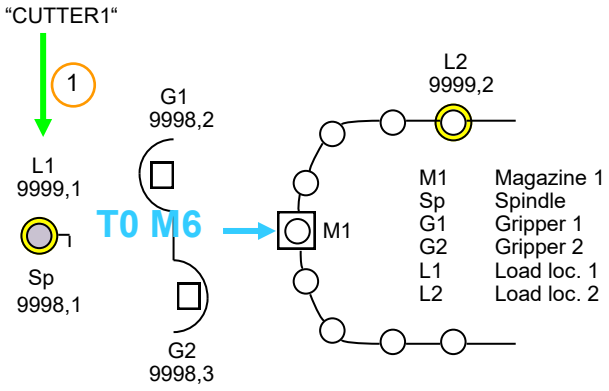
1. Select the tool.
2. Press the softkey “Load“.



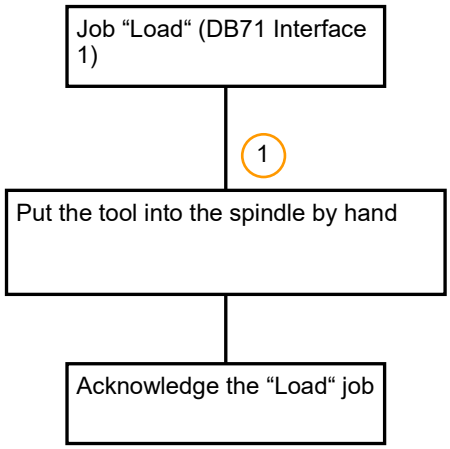


3. Press softkey “Spindle” -> Job to PLC (DB71.DBX4.0=1).

Example: Load a tool to the spindle



Sequence: load to the spindle



Important:

A tool which was loaded into the spindle by hand will always be deposited in the magazine at the next tool change or T0 M6 command.

This will happen even though hand tools are allowed according to MD22562 Bit 1 = 1.

Notes:

Job: Load spindle DB71

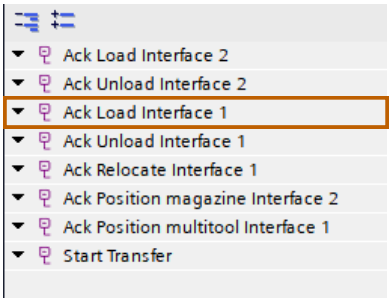
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Re-served	Re-served	Posi-tioning Multi-tool	Order from NC-Pro-gram	Posi-tioning Mag-azine	Relo-cate	Un-load	Load
DBB5	Data in DB1071	Reserved						Ack Sta-tus 3
DBW20	Identification for loading station (fixed value 9999)							
DBW22	No. of loading station							
DBW24	Magazine no. (source) for unloading/reloading/positioning							
DBW26	Location no. (source) for unloading/reloading/positioning							
DBW28	Magazine no. (target) for loading/reloading/positioning							
DBW30	Location no. (target) for loading/reloading/positioning							

Acknowledgement for load the spindle with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer will be started
TaskIdent	1	DB71 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB71.DBW28	Destination Mag. for load
NewToolLoc	DB71.DBW30	Destination Loc for load
OldToolMag	0	When loading =0
OldToolLoc	0	When loading =0
Status	1	Process finished
MultitoolLoc	-1	When loading =-1

Example: Load a tool to the spindle
DB71.DBW24 =0
DB71.DBW26 =0
DB71.DBW28 =9998
DB71.DBW30 =1

PLC Programming example with acknowledgement using FC6



Load spindle

```
REGION Ack Load Interface 1

    // Scan rising edge
    #tempLoadIF1EdgeP := #loadIF1 AND NOT #statLoadIF1Flag;
    #statLoadIF1Flag := #loadIF1;

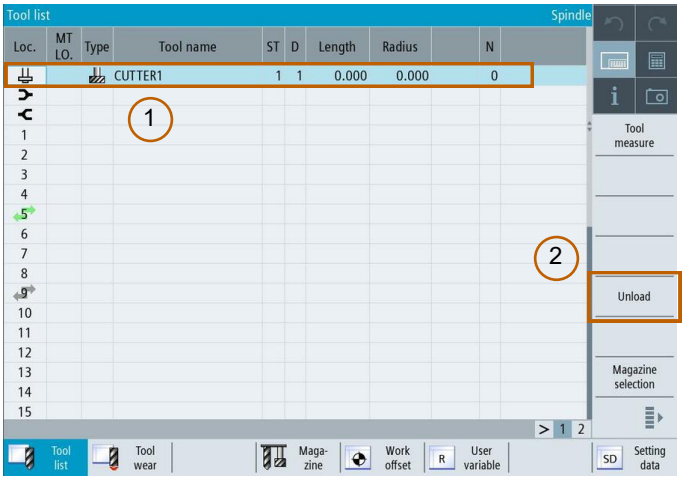
    // Set Job Start
    IF #tempLoadIF1EdgeP AND "LBP_TMLoad".IFNo[1] AND
        "LBP_TMLoad".TIF[1].Loading THEN
        #statTask := 1;                                  //DB71
        #statNo := 1;                                    //Interface 1
        #statMagNew := "LBP_TMLoad".TIF[1].TMag;
        #statLocNew := "LBP_TMLoad".TIF[1].TLoc;
        #statMagOld := 0;
        #statLocOld := 0;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

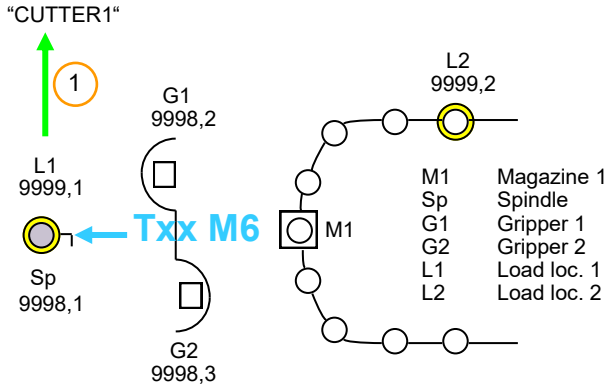
Unload the spindle

The “Unload“ dialogue can be used to unload the tool in the spindle.

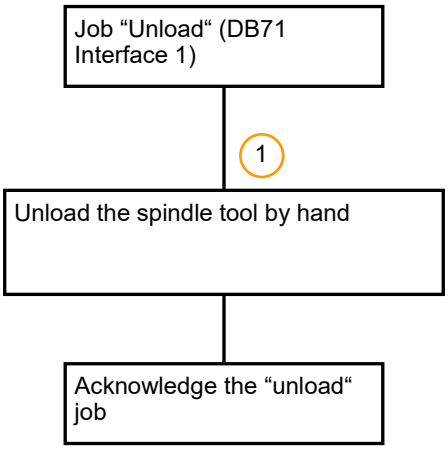


1. Select the spindle location.
2. Press the softkey “Unload“ -> Job to PLC (DB71.DBX4.1=1).

Example: Unload the tool in the spindle



Sequence unload the spindle



Notes:

Job: Unload spindle DB71

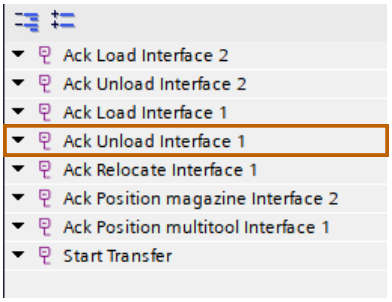
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Re-served	Re-served	Posi-tioning Multi-tool	Order from NC-Pro-gram	Posi-tioning Mag-azine	Relo-cate	Un-load	Load
DBB5	Data in DB10 71	Reserved						Ack. sta-tus 3
DBW20	Identification for loading station (fixed value 9999)							
DBW22	No. of loading station							
DBW24	Magazine no. (source) for unloading/reloading/positioning							
DBW26	Location no. (source) for unloading/reloading/positioning							
DBW28	Magazine no. (target) for loading/reloading/positioning							
DBW30	Location no. (target) for loading/reloading/positioning							

Acknowledgement for unload the spindle with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	1	DB71 Interface
TaskIdentNo	1	Interface 1
NewToolMag	0	When unloading =0
NewToolLoc	0	When unloading =0
OldToolMag	DB71.DBW20	Source Mag. for unload
OldToolLoc	DB71.DBW22	Source Location for unload
Status	1	Process finished
MultitoolLoc	-1	When unloading =-1

Example : Unload tool from spindle
DB71.DBW20 =9999
DB71.DBW22 =1
DB71.DBW24 =9998
DB71.DBW26 =1
DB71.DBW28 =0
DB71.DBW30 =0

PLC-Program example for acknowledgement with FC6



Unload spindle

```
REGION Ack Unload Interface 1

    // Scan rising edge
    #tempUnloadIF1EdgeP := #unloadIF1 AND NOT #statUnloadIF1Flag;
    #statUnloadIF1Flag := #unloadIF1;

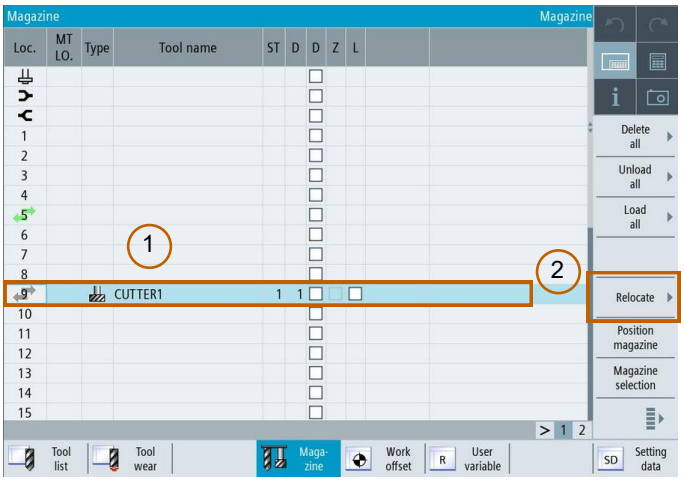
    // Set Job Start
    IF #tempUnloadIF1EdgeP AND "LBP_TMLoad".IFNo[1] AND
        "LBP_TMLoad".TIF[1].Unloading THEN
        #statTask := 1;                                  //DB71
        #statNo := 1;                                    //Interface 1
        #statMagNew := 0;
        #statLocNew := 0;
        #statMagOld := "LBP_TMLoad".TIF[1].LoadIdent;
        #statLocOld := "LBP_TMLoad".TIF[1].LoadNo;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

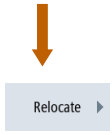
Notes:

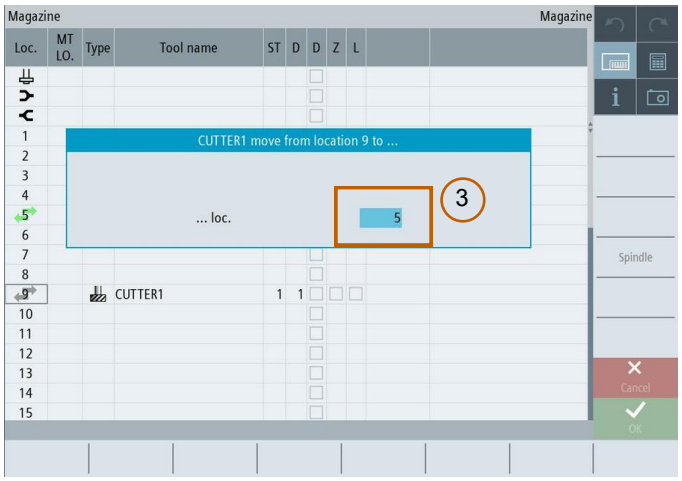
Relocate tool

A tool can be moved from one location to another with the ”Relocate“ dialogue.



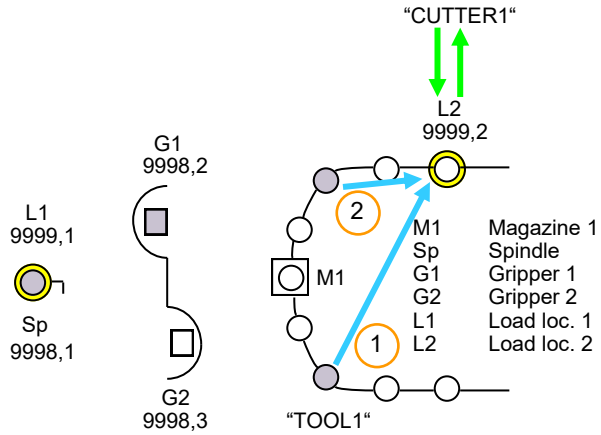
1. Select the tool to be relocated.
2. Press softkey „Relocate“.



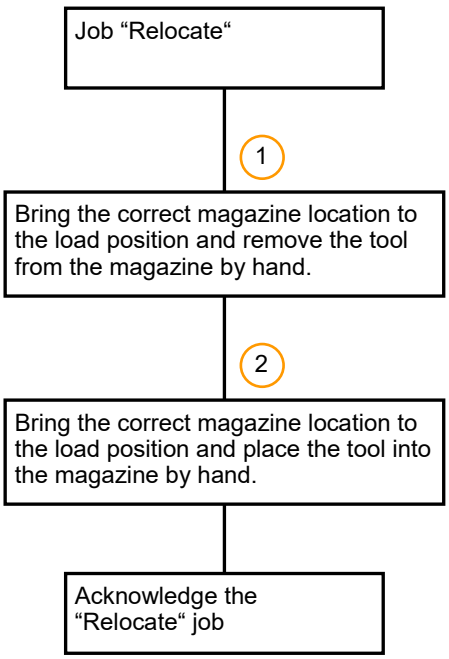


3. Enter the desired location and confirm with OK -> Job to PLC (DB71.DBX4.2=1).

Example: Relocate a tool in the magazine



Sequence for tool relocacte



Notes:

Job: Relocate tool DB71

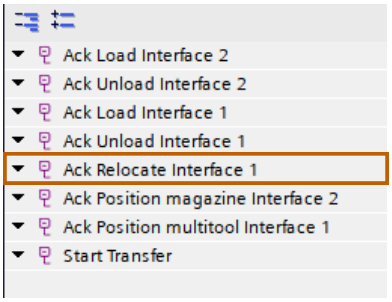
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Re-served	Re-served	Positioning Multi-tool	Order from NC-Program	Positioning Mag-azine	Relo-cate	Un-load	Load
DBB5	Data in DB1071	Reserved						Ack. sta-tus 3
DBW20	Identification for loading station (fixed value 9999)							
DBW22	No. of loading station							
DBW24	Magazine no. (source) for unloading/reloading/positioning							
DBW26	Location no. (source) for unloading/reloading/positioning							
DBW28	Magazine no. (target) for loading/reloading/positioning							
DBW30	Location no. (target) for loading/reloading/positioning							

Acknowledgement for Relocate tool with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	1	DB71 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB71.DBW28	Destination mag. for relocate
NewToolLoc	DB71.DBW30	Destination loc. for relocate
OldToolMag	DB71.DBW24	Source mag. for relocate
OldToolLoc	DB71.DBW26	Source location for relocate
Status	1/6	1: Process finished - relocate within magazine 1 6: Process finished - relocate to magazine 9998 with location reservation.
MultitoolLoc	-1	When relocate =-1

Example: Relocate the tool from location 9 to 5
DB71.DBW24 =1
DB71.DBW26 =9
DB71.DBW28 =1
DB71.DBW30 =5

PLC program example for acknowledgment with FC6



Relocate tool

```
REGION Ack Relocate Interface 1

    // Scan rising edge
    #tempRelocateIF1EdgeP := #relocateIF1 AND NOT #statRelocateIF1Flag;
    #statRelocateIF1Flag := #relocateIF1;

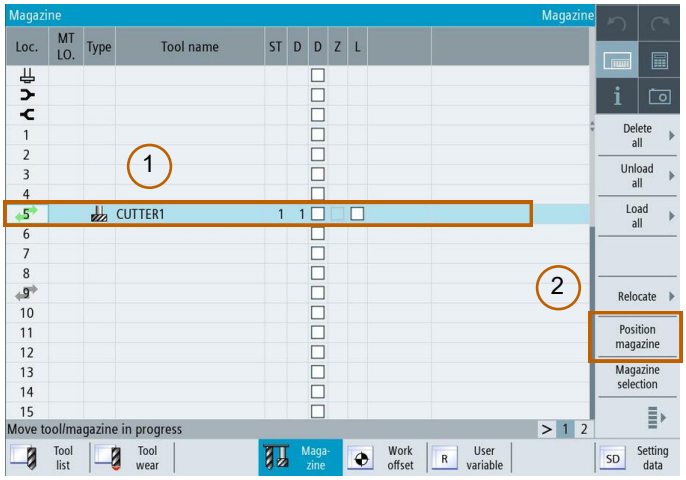
    // Set Job Start
    IF #tempRelocateIF1EdgeP AND "LBP_TMLoad".IFNo[1] AND
        "LBP_TMLoad".TIF[1].Reloading THEN
        #statTask := 1;                                  //DB71
        #statNo := 1;                                    //Interface 1
        #statMagNew := "LBP_TMLoad".TIF[1].TMag;
        #statLocNew := "LBP_TMLoad".TIF[1].TLoc;
        #statMagOld := "LBP_TMLoad".TIF[1].SMag;
        #statLocOld := "LBP_TMLoad".TIF[1].SLoc;
        // Relocate to spindle?
        IF "LBP_TMLoad".TIF[1].TMag = 9998 THEN
            #statStatus := 6;                            //Status 6
        ELSE
            #statStatus := 1;                            //Status 1
        END_IF;
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

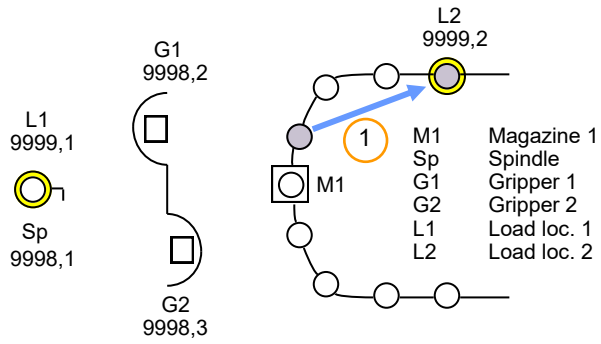
Position magazine

The ”Position magazine“ dialogue is used to position a particular magazine pocket to the load position.

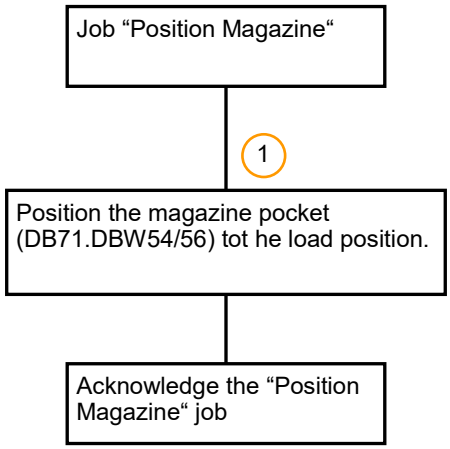


1. Select the magazine pocket.
2. Press the softkey “Position magazine“ -> Job to PLC (DB71.DBX34.3=1).

Example: Position the magazine to the load location



Sequence: Position Magazine



Notes:

Job: Position magazine DB71

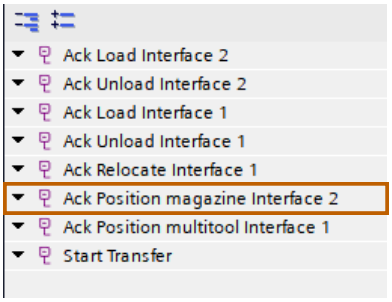
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB34	Re-served	Re-served	Positioning Multi-tool	Order from NC-Program	Positioning Mag-azine	Relo-cate	Un-load	Load
DBB35	Data in DB1071	Reserved						Ack. status 3
DBW50	Identification for loading station (fixed value 9999)							
DBW52	No. of loading station							
DBW54	Magazine no. (source) for unloading/reloading/positioning							
DBW56	Location no. (source) for unloading/reloading/positioning							
DBW58	Magazine no. (target) for loading/reloading/positioning							
DBW60	Location no. (target) for loading/reloading/positioning							

Acknowledgement: Position magazine with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	1	DB71 Interface
TaskIdentNo	2	Interface 2
NewToolMag	DB71.DBW58	Destination mag. for position-ing
NewToolLoc	DB71.DBW60	Destination location for posi-tioning
OldToolMag	0	When positioning =0
OldToolLoc	0	When positioning =0
Status	5	Process finished
MultitoolLoc	-1	When positioning =-1

Example: Bring magazine pocket 5 to the load position
DB71.DBW54 =1
DB71.DBW56 =5
DB71.DBW58 =9999
DB71.DBW60 =2

PLC program example for acknowledgement with FC6



Position mag

```
REGION Ack Position magazine Interface 2

    // Scan rising edge
    #tempPosMagIF2EdgeP := #posMagIF2 AND NOT #statPosMagIF2Flag;
    #statPosMagIF2Flag := #posMagIF2;

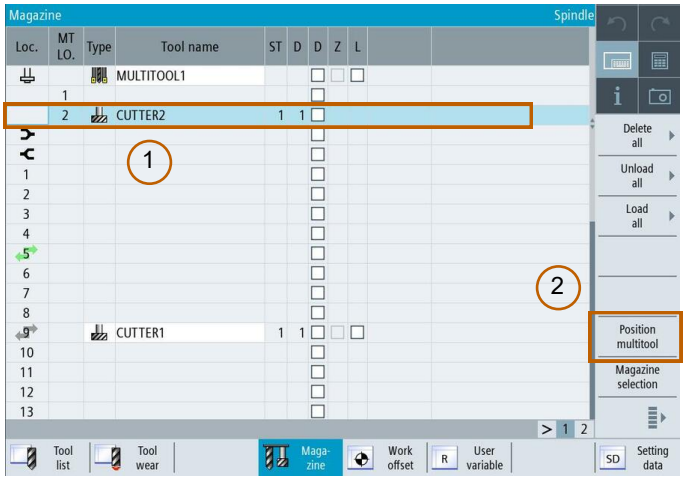
    // Set Job Start
    IF #tempPosMagIF2EdgeP AND "LBP_TMLoad".IFNo[2] AND
        "LBP_TMLoad".TIF[2].Positioning THEN
        #statTask := 1;                                  //DB71
        #statNo := 2;                                    //Interface 2
        #statMagNew := "LBP_TMLoad".TIF[2].TMag;
        #statLocNew := "LBP_TMLoad".TIF[2].TLoc;
        #statMagOld := 0;
        #statLocOld := 0;
        #statStatus := 5;                                //Status 5
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

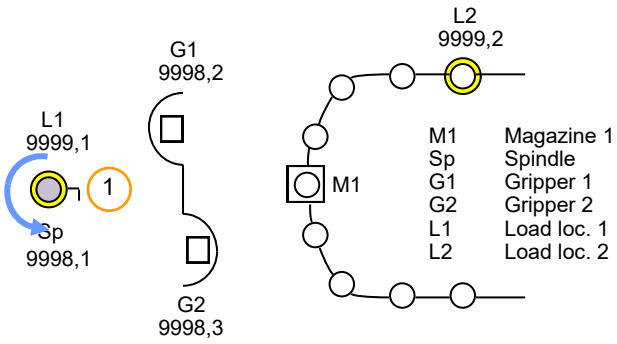
Position Multitool

With the dialogue ”Multitool“, a Multitool already in the spindle can be positioned.

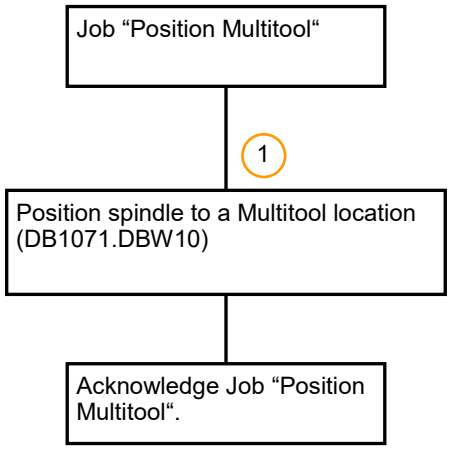


1. Select a Multitool location.
2. Press the softkey “Position multitool“ -> Job to PLC (DB71.DBX4.5=1).

Example: Position Multitool in the spindle



Sequence Position Multitool



Note:
The Multitool can only be positioned after the tool offset has been deselected with D0.

Notes:

Job: Position Multitool DB71

DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Re-served	Re-served	Positioning Multi-tool	Order from NC-Pro-gram	Positioning Mag-azine	Relo-cate	Un-load	Load
DBB5	Data in DB1071	Reserved						Ack. sta-tus 3

Data for Multitool DB1071

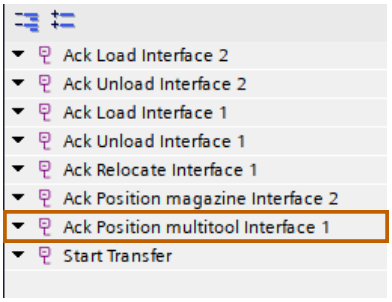
DB1071	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW10	Multitool location number							

Acknowledgement Position Multitool with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	1	DB71 Interface
TaskIdentNo	1	Interface 1
NewToolMag	-1	When positioning MT =-1
NewToolLoc	-1	When positioning MT =-1
OldToolMag	-1	When positioning MT =-1
OldToolLoc	-1	When positioning MT =-1
Status	5	Process finished
MultitoolLoc	DB1071.DBW10	Multitool location

Example: Position Multitool to location 2
DB1071.DBW10 =2

PLC Program example for acknowledgement with FC6



Position MT

```
REGION Ack Position multitool Interface 1

    // Scan rising edge
    #tempPosMtIF1EdgeP := #posMtIF1 AND NOT #statPosMtIF1Flag;
    #statPosMtIF1Flag := #posMtIF1;

    // Set Job Start
    IF #tempPosMtIF1EdgeP AND "LBP_TMLoad".IFNo[1] AND
        "LBP_TMLoad".TIF[1].PositioningMT THEN
        #statTask := 1;                                  //DB71
        #statNo := 1;                                    //Interface 1
        #statMagNew := -1;
        #statLocNew := -1;
        #statMagOld := -1;
        #statLocOld := -1;
        #statStatus := 5;                                //Status 5
        #statLocMt := "LBP_TMLoadMT".MTIF[1].PlaceNo;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Acknowledgements for Reset

PLC Programming example with acknowledgement using FC6

```
// Example "TMChain" [FB150] - continued

// Variables for "Reset"

// Input
//    resetDB71IF1 : Bool;
//    resetDB71IF2 : Bool;
//    resetDB72IF1 : Bool;

// Output
//    None

// InOut
//    None

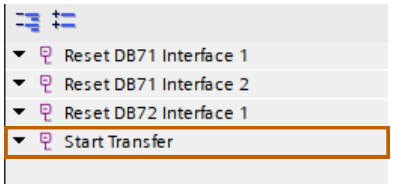
// Static
//    statResetDB71IF1Flag : Bool;
//    statResetDB71IF2Flag : Bool;
//    statResetDB72IF1Flag : Bool;
//    statTask : Byte;
//    statNo : Byte;
//    statMagNew : Int;
//    statLocNew : Int;
//    statMagOld : Int;
//    statLocOld : Int;
```

```
//    statStatus : Int;
//    statLocMt : Int;
//    statStart : Bool;

// Temp
//    tempResetDB71IF1EdgeP : Bool;
//    tempResetDB71IF2EdgeP : Bool;
//    tempResetDB72IF1EdgeP : Bool;
//    tempReady : Bool;
//    tempError : Int;

// Constant
//    None
```

PLC Programming example with acknowledgement using FC6



Start Transfer

```
REGION Start Transfer

    // Tool management status transfer
    "LBP_TMTransferMT"(Start := #statStart,
                       TaskIdent := #statTask,
                       TaskIdentNo := #statNo,
                       NewToolMag := #statMagNew,
                       NewToolLoc := #statLocNew,
                       OldToolMag := #statMagOld,
                       OldToolLoc := #statLocOld,
                       Status := #statStatus,
                       MultitoolLoc := #statLocMt,
                       Ready => #tempReady,
                       Error => #tempError);

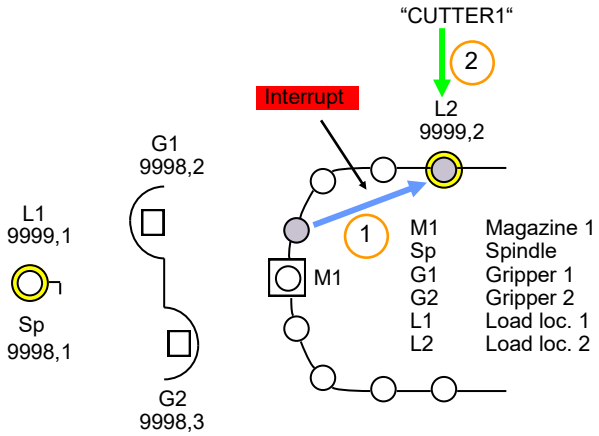
    // Reset Job Start
    IF #tempReady OR ("mcpIn".reset AND (#tempError <> 0)) THEN
        #statStart := FALSE;
    END_IF;

END_REGION
```

Notes:

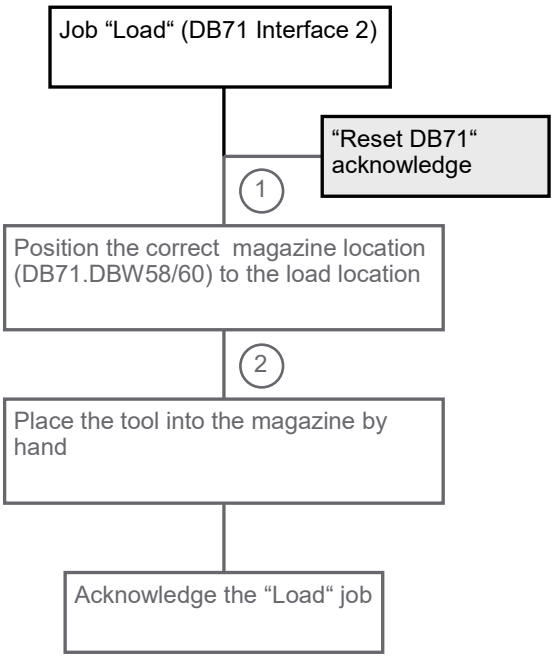
Interruption to the loading/unloading sequence

Example: Load a tool into the magazine



Examples of what can cause the loading sequence to be aborted
- Reset
- E-Stop
- Power failure

Sequence load magazine



Note:

By directly writing to parameter **$TC_MPP6[magazine No.,location No.] = 0**, a tool can be removed from this magazine location (whether it is a real magazine, buffer or loading magazine). All reservations of the removed tool are automatically deleted.
This does not involve an associated command output to the PLC.

Notes:

Job: Load magazine DB71

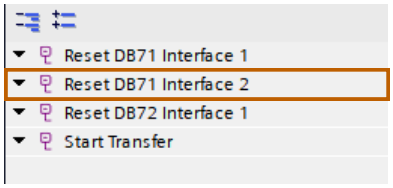
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB34	Re-served	Re-served	Posi-tion-ing Multi-tool	Order from NC-Pro-gram	Posi-tionin g Mag-azine	Relo-cate	Un-load	Load
DBB35	Data in DB10 71	Reserved						Ack Sta-tus 3
DBW50	Identification for loading station (fixed value 9999)							
DBW52	No. of loading station							
DBW54	Magazine no. (source) for unloading/reloading/positioning							
DBW56	Location no. (source) for unloading/reloading/positioning							
DBW58	Magazine no. (target) for loading/reloading/positioning							
DBW60	Location no. (target) for loading/reloading/positioning							

Acknowledgement Reset DB71 with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	1	DB71 interface
TaskIdentNo	2	Interface 2
NewToolMag	-1	When process abort =-1
NewToolLoc	-1	When process abort =-1
OldToolMag	-1	When process abort =-1
OldToolLoc	-1	When process abort =-1
Status	3	Process aborted
MultitoolLoc	-1	When process abort =-1

Example: Load tool to magazine location 9
DB71.DBW54 =0
DB71.DBW56 =0
DB71.DBW58 =1
DB71.DBW60 =9

PLC Programming example with acknowledgement using FC6



Reset DB71 IF2

```
REGION Reset DB71 Interface 2

    // Scan rising edge
    #stempResetDB71IF2EdgeP := #resetDB71IF2 AND NOT #statResetDB71IF2Flag;
    #statResetDB71IF2Flag := #resetDB71IF2;

    // Set Job Start
    IF #stempResetDB71IF2EdgeP AND "LBP_TMLoad".IFNo[2] THEN
        #statTask := 1;                                  //DB71
        #statNo := 2;                                    //Interface 2
        #statMagNew := -1;
        #statLocNew := -1;
        #statMagOld := -1;
        #statLocOld := -1;
        #statStatus := 3;                                //Status 3
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Sequence for Tool Change

The tool change is done at the change location, which is specified in the magazine configuration.

DB72 contains an interface area for each change location which has been configured.

Each interface is set up to handle these following Tool Management situations:

- Prepare tool change
- Execute tool change (triggered via M06)

In order to inform the Tool Management of tool movements the following functions are used in the machine PLC program FC6 (Multitool). Standard end acknowledgment via DB72 is also available.

Procedure:

- Set up the machine data for the tool change
- Generate the tool change program L6
- Write the PLC program for acknowledgment of the tool change commands
  - Prepare tool change
  - Tool change sequence
  - Hand tools
  - Coordination between hand and magazine tools
  - Interruption to the tool change
  - Recovery functions after an interruption

Set up the machine data for the tool change

Tool change with M function

MD22550 = 1	An M function is used to start the „Tool Change“ phase, which M function is specified by MD 22560. The T command starts the „Prepare Tool“ phase.
MD22560 = 206	M Function for Tool Change This M function triggers the start of the tool change process.

Tool change program L6

MD10715 = 6	This is the M function which calls a subprogram to be called. The name of the subprogram is given in MD10716.
MD10716 = L6	The name of the subprogram which is called when the M function specified in MD10715 is programmed.

The tool change program, L6 is called whenever M6 is programmed.

Programming the tool change

```
TEST_PROGRAM.MPF
N10
N20
N30  T="CUTTER1"        → Prepare tool -> PLC
N40
N50
N60  M06                → Tool change program L6 is called
N70
N80
N90
N100 M30 Program end
```

Notes:

Allow hand tools

MD22562	Response to tool change errors Standard behaviour: During the tool change preparation only those tools are recognized whose data are assigned to a magazine.
Bit 1 = 1	Manual tools permitted

Hand tools
The NC part program can also call tools which are not assigned to a magazine. Such tools need to be manually loaded into the machine (e.g. directly into the spindle) for the duration of their use. When the program has finished with the hand tool ,it has to be manually removed from the machine.

Tool Management uses DB72.DBX4.5 and DBX4.6 to inform the PLC program that a hand tool is involved in the change. The machine operator gets one of these alarms 17212, 17214, 17215 ,17216 to warn that manual action is needed to make the tool change. The alarms are reset when the PLC program acknowledges the tool change.

Suppress alarm output for hand tools

MD11415	Suppression of alarm output
Bit 16 = 0	Alarms 17212, 17214, 17215, 17216 for hand tools are not masked out

Bit16=0 No masked out alarms for hand tools
Bit16=1 Masked out alarms for hand tools

- 17212 ”Load hand tool to spindle/tool holder”
- 17214 ”Remove hand tool from spindle/tool holder”
- 17215 ”Remove hand tool from buffer location”
- 17216 ”Remove hand tool from tool holder and load hand tool”

Ignore location reservation with asynchronous Transfer

MD20310	Activation of Tool Management
Bit 24 = 1	Asynchronous transfer command no. 8 and 9: Ignore location reservation

The location reservation of the tool in the magazine will be ignored. Thus it is possible to move a tool to this location (e.g. via asynchronous transfer).

Generate the tool change program L6

The L6, tool change program has to take the machine layout into consideration. The L6 program is stored in the manufacturer cycles directory of the control .

```
PROC L6 SAVE DISPLOF
;---------------------------------------------------------------------
; Example: Tool change cycle for manufacturer
;---------------------------------------------------------------------
DEF INT _WZ_IN_SP,_WZ_VOR
DEF REAL _WWP=649             ; tool change position MCS
DEF REAL _SPP=0               ; spindle position for tool change
;
IF(NOT $P_SEARCH)             ; if no block search active
  _WZ_IN_SP=$TC_MPP6[9998,1] ; tool in spindle
  GETSELT(_WZ_VOR)            ; preselected tool
  ;
  IF(_WZ_IN_SP<>_WZ_VOR)      ; if other tool
    ; Position spindle:
    SPOS=_SPP
    ; Approach tool change position:
    SUPA D0 G0 G90 G40 G60 Z=_WWP
  ENDIF
ELSE
ENDIF
;
; Change tool: Tool management and PLC
M206
;*
M17
;---------------------------------------------------------------------
; END
;---------------------------------------------------------------------
```

Notes:

Sequence for Tool Change

Overview of the acknowledgements

Acknowledgement steps for chain magazine with double gripper and optional hand tools.

Tool change “1:1” T0, no Tool in Spindle
Prepare Change Status 1 **I81.1**
Update Magazine position Status 4 **I81.2**
Tool change Step 1 Stat 105 **I81.3**
Tool change Step 2 Stat 105 **I81.4**
Perform Change Status 1 **I81.5**

Tool change ”not_1:1“	**Tool change ”not_1:1“ T0**	**Tool change ”not_1:1“ no Tool in spindle**	**Tool change „mixed“ Hand tool Out**	**Tool change „mixed“ Hand tool In**
Prepare Change Status 1 **I82.0**	Prepare Change Status 1 **I82.0**	Prepare Change Status 1 **I82.0**	Prepare Change Status 1 **I82.0**	Prepare Change Status 1 **I82.0**
Update Magazine position old Tool Status 4 **I82.1**	Update Magazine position old Tool Status 4 **I82.1**			Update Magazine position old Tool Status 4 **I82.1**
Tool change Step 1 Stat 105 **I82.2**	Tool change Step 1 Stat 105 **I82.2**		Hand tool Out Stat 105 **I83.5**	Tool change Step 1 Stat 105 **I82.2**
Tool change Step 2 Stat 105 **I82.3**	Tool change Step 2 Stat 105 **I82.3**			Tool change Step 2 Stat 105 **I82.3**
Update Magazine position new Tool Status 4 **I82.4**		Update Magazine position new Tool Status 4 **I82.4**	Update Magazine position new Tool Status 4 **I82.4**	
Tool change Step 3 Stat 105 **I82.5**		Tool change Step 3 Stat 105 **I82.5**	Tool change Step 3 Stat 105 **I82.5**	Hand tool In Stat 105 **I83.6**
Tool change Step 4 Stat 105 **I82.6**		Tool change Step 4 Stat 105 **I82.6**	Tool change Step 4 Stat 105 **I82.6**	
Perform Change Status 1 **I82.7**	Perform Change Status 1 **I82.7**	Perform Change Status 1 **I82.7**	Perform Change Status 1 **I82.7**	Perform Change Status 1 **I82.7**

Hand tool T0, no Tool in Spindle
Prepare Change Status 1 **I82.0**
Hand tool In/Out Stat 105 **I83.7**
Perform Change Status 1 **I82.7**

PLC Programming examples

Change1_1	ChangeNot1_1	HandtoolAddon

Notes:

Acknowledgements for tool change with 1:1 exchange

PLC Programming example with acknowledgement using FC6

```
// Example "TMChain" [FB150] - continued

// Variables for "Toolchange 1:1"

// Input
//    tc1_1Prepare : Bool;
//    tc1_1UpdMag : Bool;
//    tc1_1Step1 : Bool;
//    tc1_1Step2 : Bool;
//    tc1_1Perform : Bool;

// Output
//    None

// InOut
//    None

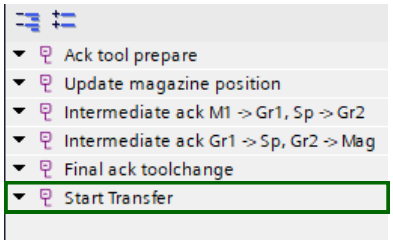
// Static
//    statTc1_1PrepareFlag : Bool;
//    statTc1_1UpdMagFlag : Bool;
//    statTc1_1Step1Flag : Bool;
//    statTc1_1Step2Flag : Bool;
//    statTc1_1PerformFlag : Bool;
```

```
//    statTask : Byte;
//    statNo : Byte;
//    statMagNew : Int;
//    statLocNew : Int;
//    statMagOld : Int;
//    statLocOld : Int;
//    statStatus : Int;
//    statLocMt : Int;
//    statStart : Bool;

// Temp
//    tempTc1_1PrepareEdgeP : Bool;
//    tempTc1_1UpdMagEdgeP : Bool;
//    tempTc1_1Step1EdgeP : Bool;
//    tempTc1_1Step2EdgeP : Bool;
//    tempTc1_1PerformEdgeP : Bool;
//    tempReady : Bool;
//    tempError : Int;

// Constant
//    None
```

PLC Programming example with acknowledgement using FC6



Start Transfer

```
REGION Start Transfer

    // Tool management status transfer
    "LBP_TMTransferMT"(Start := #statStart,
                       TaskIdent := #statTask,
                       TaskIdentNo := #statNo,
                       NewToolMag := #statMagNew,
                       NewToolLoc := #statLocNew,
                       OldToolMag := #statMagOld,
                       OldToolLoc := #statLocOld,
                       Status := #statStatus,
                       MultitoolLoc := #statLocMt,
                       Ready => #tempReady,
                       Error => #tempError);

    // Reset Job Start
    IF #tempReady OR ("mcpIn".reset AND (#tempError <> 0)) THEN
        #statStart := FALSE;
    END_IF;

END_REGION
```

Notes:

Prepare tool change

Programming

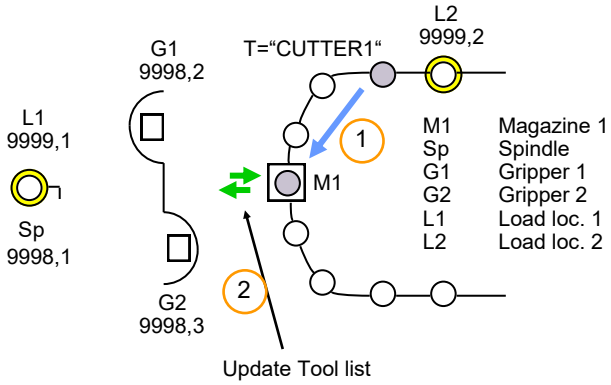
```
TEST_PROGRAM.MPF
N10
N20
N30
N40
N50 T="CUTTER1"
N60
N70
N80
N90
N100 M06
N110
N120
N130
N140
N150
N160 M30 Program end
```

Prepare tool -> PLC

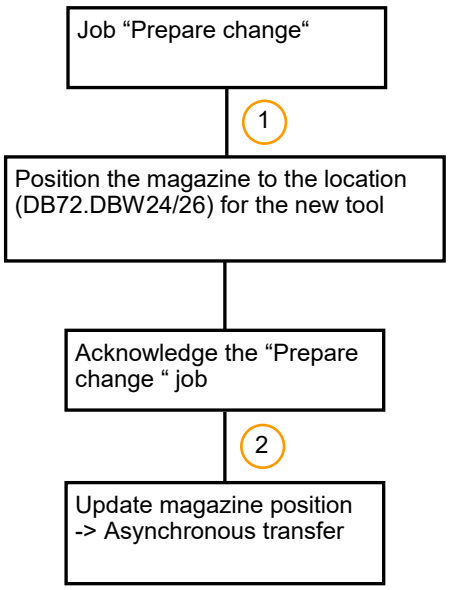
Call L6, do tool change

Prepare for tool change -> Job to PLC (DB72.DBX4.2=1).

Example: Prepare tool change



Sequence for prepare tool change



Note

For a 1:1 exchange the new and old tools must satisfy all these conditions:

- Neither tool is fixed location
- Both tools have the same size
- Both tools have the same location type

Notes:

Job: Prepare tool change

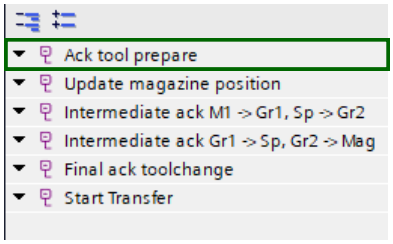
DB72	Signals from spindle (NCK->PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Tool re-mains in spin-dle	Re-move man-ual tool	Load man-ual tool	OldT in buffer no (n -42)	T0	Pre-pare chan ge	Per-form chan ge (init: M06)	Com pulso ry chan ge
DBW20	Buffer identification (fixed value 9998)							
DBW22	Location in buffer							
DBW24	Magazine no. (source) for new tool							
DBW26	Location no. (source) for new tool							
DBW28	Magazine no. (target) for old tool							
DBW30	Location no. (target) for old tool							
DBB42	Tool status for tool new							
	Man-ual tool	1:1 ex-change		Mas-ter tool	to be load-ed	to be un-loaded	disa-bled	ID for tool

Acknowledgement for prepare tool change with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB72.DBW24	Source magazine for prepare tool change
NewToolLoc	DB72.DBW26	Source location for prepare tool change
OldToolMag	DB72.DBW20	Destination magazine for pre-pare tool change
OldToolLoc	DB72.DBW22	Destination location for pre-pare tool change
Status	1	Process finished
MultitoolLoc	-1	When prepare change =-1

Example: Prepare tool change 1:1
DB72.DBW24 =1
DB72.DBW26 =10
DB72.DBW28 =1
DB72.DBW30 =10
DB72.DBX42.6=1

PLC Programming example with acknowledgement using FC6



Prepare change

```
REGION Ack tool prepare

    // Scan rising edge
    #tempTc1_1PrepareEdgeP := #tc1_1Prepare AND NOT #statTc1_1PrepareFlag;
    #statTc1_1PrepareFlag := #tc1_1Prepare;

    // Set Job Start
    IF #tempTc1_1PrepareEdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        NOT "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        #statMagNew := "LBP_TMSpindle".TIF[1].SMag;
        #statLocNew := "LBP_TMSpindle".TIF[1].SLoc;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := "LBP_TMSpindle".TIF[1].BufferIdent;
            #statLocOld := "LBP_TMSpindle".TIF[1].BufferNo;
        END_IF;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

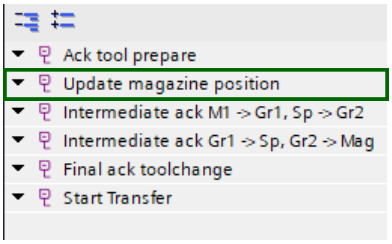
END_REGION
```

Notes:

Asynchronous transfer (Update magazine position)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	4	Asynchronous transfer
TaskIdentNo	1	Channel 1
NewToolMag	DB72.DBW24	Destination magazine for tool
NewToolLoc	DB72.DBW26	Destination location for tool
OldToolMag	9998	Source magazine of tool
OldToolLoc	1	Source location of tool
Status	5	Process finished
MultitoolLoc	-1	Multitool position

PLC Programming example with acknowledgement using FC6



Update mag. pos.

```
REGION Update magazine position

    // Scan rising edge
    #tempTc1_1UpdMagEdgeP := #tc1_1UpdMag AND NOT #statTc1_1UpdMagFlag;
    #statTc1_1UpdMagFlag := #tc1_1UpdMag;

    // Set Job Start
    IF #tempTc1_1UpdMagEdgeP THEN
        #statTask := 4;                                  //Asyn. transfer
        #statNo := 1;                                    //Channel 1
        #statMagOld := 9998;
        #statLocOld := 1;
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := "LBP_TMSpindle".TIF[1].TMag;
            #statLocNew := "LBP_TMSpindle".TIF[1].TLoc;
        ELSE
            #statMagNew := "LBP_TMSpindle".TIF[1].SMag;
            #statLocNew := "LBP_TMSpindle".TIF[1].SLoc;
        END_IF;
        #statStatus := 5;                                //Status 5
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Sequence for tool change

Programming

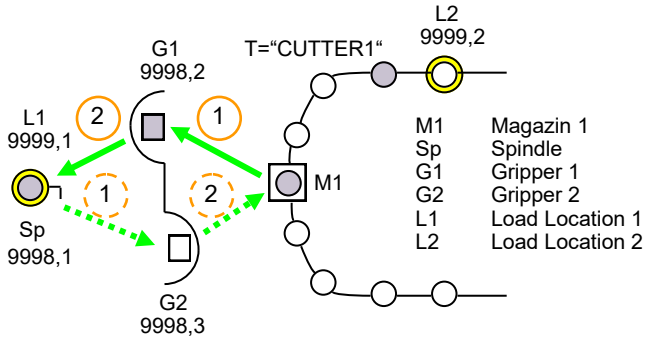
```
TEST_PROGRAM.MPF
N10
N20
N30
N40
N50 T="CUTTER1"
N60
N70
N80
N90
N100 M06
N110
N120
N130
N140
N150
N160 M30 Program end
```

Prepare tool -> PLC

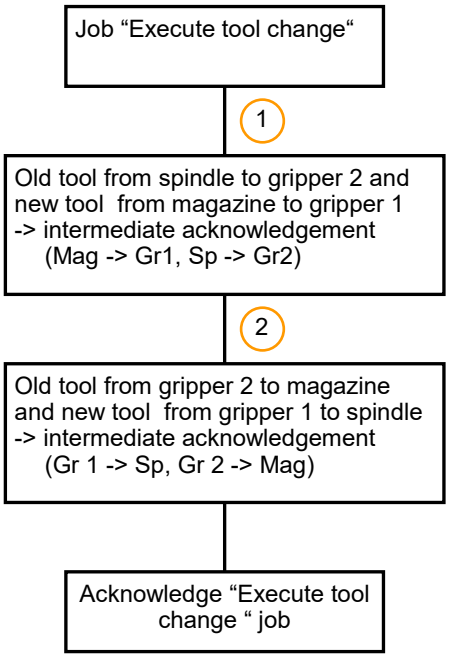
Call L6, do tool change

Tool change -> Job to PLC (DB72.DBX4.1=1).

Example: Tool change with 1:1 exchange



Sequence: Tool change with 1:1 exchange



Note:

For a 1:1 exchange the new and old tools must satisfy all these conditions:

- Neither tool is fixed location
- Both tools have the same size
- Both tools have the same location type

Notes:

Job: Execute tool change

DB72	Signals from spindle (NCK->PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Tool re-mains in spin-dle	Re-move man-ual tool	Load man-ual tool	OldT in buffer no (n -42)	T0	Pre-pare chan ge	Per-form chan ge (init: M06)	Com pulso ry chan ge
DBW20	Buffer identification (fixed value 9998)							
DBW22	Location in buffer							
DBW24	Magazine no. (source) for new tool							
DBW26	Location no. (source) for new tool							
DBW28	Magazine no. (target) for old tool							
DBW30	Location no. (target) for old tool							
DBB42	Tool status for tool new							
	Man-ual tool	1:1 ex-change		Mas-ter tool	to be load-ed	to be un-loaded	disa-bled	ID for tool

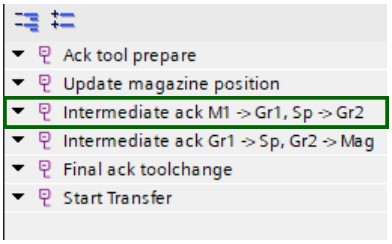
Intermediate acknowledgement (Mag -> Gr 1, Sp -> Gr 2)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	9998	Source magazine of new tool
NewToolLoc	2	Source location of new tool
OldToolMag	9998	Destination magazine for old tool
OldToolLoc	3	Destination location for old tool
Status	105	Intermediate positions reached
MultitoolLoc	-1	Multitool position

Example: Execute tool change 1:1
DB72.DBW24 =1
DB72.DBW26 =10
DB72.DBW28 =1
DB72.DBW30 =10
DB72.DBX42.6=1

The job remains unchanged, the tools are still in their respective starting positions.

PLC Prog example with ack using FC6



Mag -> Gr 1, Sp -> Gr 2

```
REGION Intermediate ack M1 -> Gr1, Sp -> Gr2

    // Scan rising edge
    #tempTc1_1Step1EdgeP := #tc1_1Step1 AND NOT #statTc1_1Step1Flag;
    #statTc1_1Step1Flag := #tc1_1Step1;

    // Set Job Start
    IF #tempTc1_1Step1EdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := 9998;
            #statLocNew := 2;
        END_IF;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := 9998;
            #statLocOld := 3;
        END_IF;
        #statStatus := 105;                              //Status 105
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

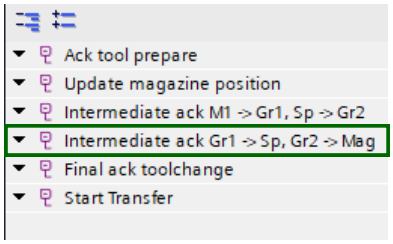
END_REGION
```

Notes:

Intermediate acknowledgement (Gr 1 -> Mag, Gr 2 -> Sp)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB72.DBW20	Source magazine of new tool
NewToolLoc	DB72.DBW22	Source location of new tool
OldToolMag	DB72.DBW28	Destination magazine for old tool
OldToolLoc	DB72.DBW30	Destination location for old tool
Status	105	Intermediate positions reached
MultitoolLoc	-1	Multitool position

PLC Prog example with ack using FC6



Gr 1 -> Sp, Gr 2 -> Mag

```
REGION Intermediate ack Gr1 -> Sp, Gr2 -> Mag

    // Scan rising edge
    #tempTc1_1Step2EdgeP := #tc1_1Step2 AND NOT #statTc1_1Step2Flag;
    #statTc1_1Step2Flag := #tc1_1Step2;

    // Set Job Start
    IF #tempTc1_1Step2EdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                 //DB72
        #statNo := 1;                                   //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := "LBP_TMSpindle".TIF[1].BufferIdent;
            #statLocNew := "LBP_TMSpindle".TIF[1].BufferNo;
        END_IF;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := "LBP_TMSpindle".TIF[1].TMag;
            #statLocOld := "LBP_TMSpindle".TIF[1].TLoc;
        END_IF;
        #statStatus := 105;                             //Status 105
        #statLocMt := -1;
        #statStart := TRUE;                             //Start FC6
    END_IF;

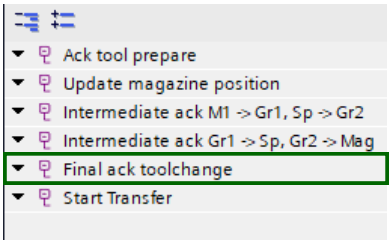
END_REGION
```

Notes:

Final acknowledgement for “Execute Change“ with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB72.DBW20	Source magazine of new tool
NewToolLoc	DB72.DBW22	Source location of new tool
OldToolMag	DB72.DBW28	Destination magazine for old tool
OldToolLoc	DB72.DBW30	Destination location for old tool
Status	1	Process finished
MultitoolLoc	DB1072.DBW10	Multitool position

PLC Prog example with ack using FC6



Perform change

```
REGION Final ack toolchange

    // Scan rising edge
    #tempTc1_1PerformEdgeP := #tc1_1Perform AND NOT #statTc1_1PerformFlag;
    #statTc1_1PerformFlag := #tc1_1Perform;

    // Set Job Start
    IF #tempTc1_1PerformEdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := "LBP_TMSpindle".TIF[1].BufferIdent;
            #statLocNew := "LBP_TMSpindle".TIF[1].BufferNo;
        END_IF;
        #statMagOld := "LBP_TMSpindle".TIF[1].TMag;
        #statLocOld := "LBP_TMSpindle".TIF[1].TLoc;
        #statStatus := 1;                                //Status 1
        // Multitool?
        IF "LBP_TMSpindle".TIF[1].DataInDB1072 THEN
            #statLocMt := "LBP_TMSpindleMT".MTIF[1].NewPlace;
        ELSE
            #statLocMt := -1;
        END_IF;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Acknowledgements for tool change non 1:1 exchange

PLC Programming example with acknowledgement using FC6

```
// Example "TMChain" [FB150] - continued

// Variables for "Toolchange not 1:1"

// Input
//    not1_1Prepare : Bool;
//    not1_1UpdOld : Bool;
//    not1_1Step1 : Bool;
//    not1_1Step2 : Bool;
//    not1_1UpdNew : Bool;
//    not1_1Step3 : Bool;
//    not1_1Step4 : Bool;
//    not1_1Perform : Bool;

// Output
//    None

// InOut
//    None

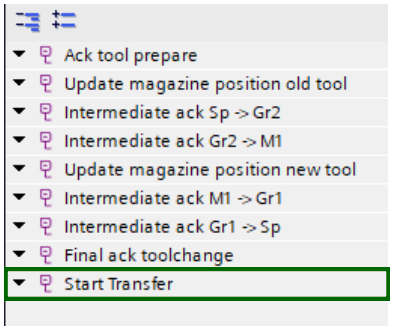
// Static
//    statNot1_1PrepareFlag : Bool;
//    statNot1_1UpdOldFlag : Bool;
//    statNot1_1Step1Flag : Bool;
//    statNot1_1Step2Flag : Bool;
//    statNot1_1UpdNewFlag : Bool;
//    statNot1_1Step3Flag : Bool;
//    statNot1_1Step4Flag : Bool;
//    statNot1_1PerformFlag : Bool;
```

```
//    statTask : Byte;
//    statNo : Byte;
//    statMagNew : Int;
//    statLocNew : Int;
//    statMagOld : Int;
//    statLocOld : Int;
//    statStatus : Int;
//    statLocMt : Int;
//    statStart : Bool;

// Temp
//    tempNot1_1PrepareEdgeP : Bool;
//    tempNot1_1UpdOldEdgeP : Bool;
//    tempNot1_1Step1EdgeP : Bool;
//    tempNot1_1Step2EdgeP : Bool;
//    tempNot1_1UpdNewEdgeP : Bool;
//    tempNot1_1Step3EdgeP : Bool;
//    tempNot1_1Step4EdgeP : Bool;
//    tempNot1_1PerformEdgeP : Bool;
//    tempReady : Bool;
//    tempError : Int;

// Constant
//    None
```

PLC Programming example with acknowledgement using FC6



Start Transfer

```
REGION Start Transfer

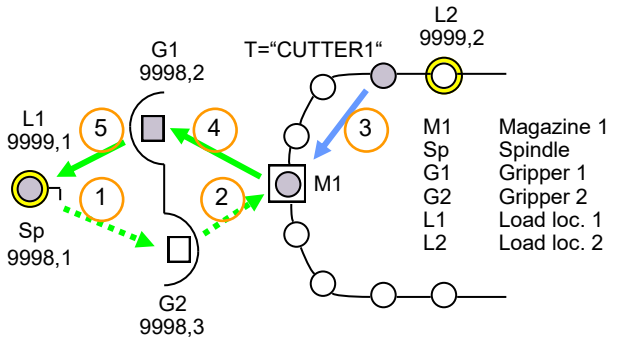
    // Tool management status transfer
    "LBP_TMTransferMT"(Start := #statStart,
                       TaskIdent := #statTask,
                       TaskIdentNo := #statNo,
                       NewToolMag := #statMagNew,
                       NewToolLoc := #statLocNew,
                       OldToolMag := #statMagOld,
                       OldToolLoc := #statLocOld,
                       Status := #statStatus,
                       MultitoolLoc := #statLocMt,
                       Ready => #tempReady,
                       Error => #tempError);

    // Reset Job Start
    IF #tempReady OR ("mcpIn".reset AND (#tempError <> 0)) THEN
        #statStart := FALSE;
    END_IF;

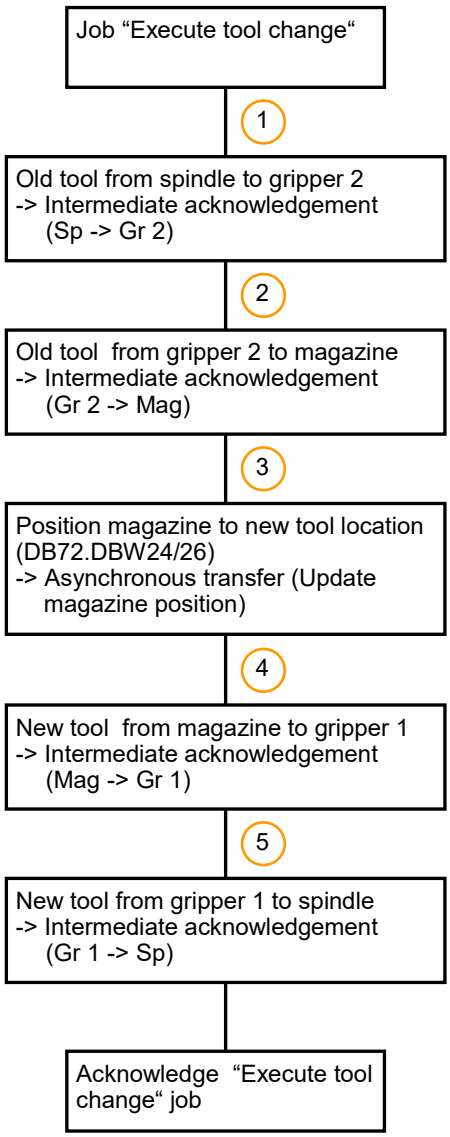
END_REGION
```

Notes:

Example: Tool change non 1:1 exchange



Sequence for non-1:1 exchange



Job: Execute tool change

DB72	Signals from spindle (NCK->PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Tool re-mains in spin-dle	Re-move man-ual tool	Load man-ual tool	OldT in buffer no (n -42)	T0	Pre-pare chan ge	Per-form chan ge (init: M06)	Com pulso ry chan ge
DBW20	Buffer identification (fixed value 9998)							
DBW22	Location in buffer							
DBW24	Magazine no. (source) for new tool							
DBW26	Location no. (source) for new tool							
DBW28	Magazine no. (target) for old tool							
DBW30	Location no. (target) for old tool							
DBB42	Tool status for tool new							
	Man-ual tool	1:1 ex-change		Mas-ter tool	to be load-ed	to be un-loaded	disa-bled	ID for tool

Example: Tool change non 1:1 exchange
DB72.DBW24 =1
DB72.DBW26 =10
DB72.DBW28 =1
DB72.DBW30 =4
DB72.DBX42.6=0

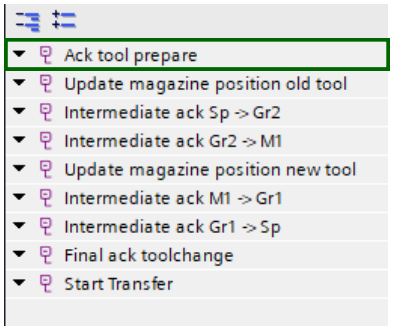
The job remains unchanged , the tools are still in their respective starting positions.

Notes:

Acknowledgement for prepare tool change with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB72.DBW24	Source magazine for prepare tool change
NewToolLoc	DB72.DBW26	Source location for prepare tool change
OldToolMag	DB72.DBW20	Destination magazine for prepare tool change
OldToolLoc	DB72.DBW22	Destination location for prepare tool change
Status	1	Process finished
MultitoolLoc	-1	When prepare change =-1

PLC Programming example with acknowledgement using FC6



Prepare change

```
REGION Ack tool prepare

    // Scan rising edge
    #tempNot1_1PrepareEdgeP := #not1_1Prepare AND NOT #statNot1_1PrepareFlag;
    #statNot1_1PrepareFlag := #not1_1Prepare;

    // Set Job Start
    IF #tempNot1_1PrepareEdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        NOT "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        #statMagNew := "LBP_TMSpindle".TIF[1].SMag;
        #statLocNew := "LBP_TMSpindle".TIF[1].SLoc;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := "LBP_TMSpindle".TIF[1].BufferIdent;
            #statLocOld := "LBP_TMSpindle".TIF[1].BufferNo;
        END_IF;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

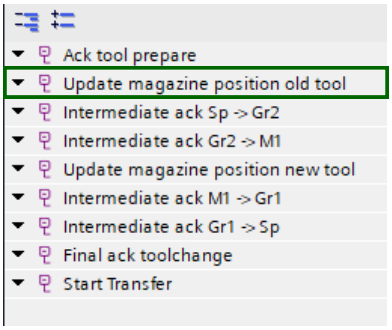
END_REGION
```

Notes:

Asynchronous transfer (Update magazine position old tool)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	4	Asynchronous transfer
TaskIdentNo	1	Channel 1
NewToolMag	DB72.DBW28	Destination magazine for tool
NewToolLoc	DB72.DBW30	Destination location for tool
OldToolMag	9998	Source magazine of tool
OldToolLoc	1	Source location of tool
Status	5	Process finished
MultitoolLoc	-1	Multitool position

PLC Programming example with acknowledgement using FC6



Update mag. Pos. old tool

```
REGION Update magazine position old tool

    // Scan rising edge
    #tempNot1_1UpdOldEdgeP := #not1_1UpdOld AND NOT #statNot1_1UpdOldFlag;
    #statNot1_1UpdOldFlag := #not1_1UpdOld;

    // Set Job Start
    IF #tempNot1_1UpdOldEdgeP THEN
        #statTask := 4;                                  //Asyn. transfer
        #statNo := 1;                                    //Channel 1
        #statMagOld := 9998;
        #statLocOld := 1;
        // Tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag <> 0 THEN
            #statMagNew := "LBP_TMSpindle".TIF[1].TMag;
            #statLocNew := "LBP_TMSpindle".TIF[1].TLoc;
            #statStatus := 5;                            //Status 5
            #statLocMt := -1;
            #statStart := TRUE;                          //Start FC6
        END_IF;
    END_IF;

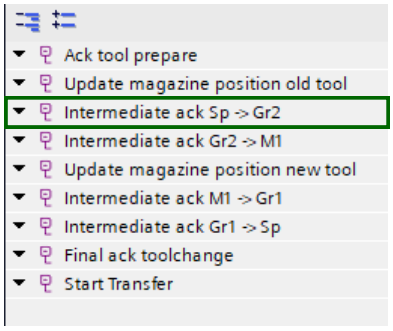
END_REGION
```

Notes:

Intermediate acknowledgement (Sp -> Gr 2)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	0	Source magazine of new tool
NewToolLoc	0	Source location of new tool
OldToolMag	9998	Destination magazine for old tool
OldToolLoc	3	Destination location for old tool
Status	105	Intermediate positions reached
MultitoolLoc	-1	Multitool position

PLC Prog example with ack using FC6



Sp -> Gr 2

```
REGION Intermediate ack Sp -> Gr2

    // Scan rising edge
    #tempNot1_1Step1EdgeP := #not1_1Step1 AND NOT #statNot1_1Step1Flag;
    #statNot1_1Step1Flag := #not1_1Step1;

    // Set Job Start
    IF #tempNot1_1Step1EdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := -1;
            #statLocNew := -1;
        END_IF;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := 9998;
            #statLocOld := 3;
        END_IF;
        #statStatus := 105;                              //Status 105
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

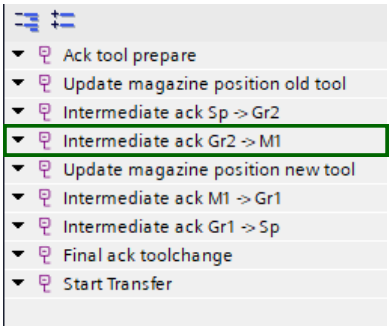
END_REGION
```

Notes:

Intermediate acknowledgement (Gr 2 -> Mag)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	0	Source magazine of new tool
NewToolLoc	0	Source location of new tool
OldToolMag	DB72.DBW28	Destination magazine for old tool
OldToolLoc	DB72.DBW30	Destination location for old tool
Status	105	Intermediate positions reached
MultitoolLoc	-1	Multitool position

PLC Prog example with ack using FC6



Gr 2 -> Mag

```
REGION Intermediate ack Gr2 -> M1

    // Scan rising edge
    #tempNot1_1Step2EdgeP := #not1_1Step2 AND NOT #statNot1_1Step2Flag;
    #statNot1_1Step2Flag := #not1_1Step2;

    // Set Job Start
    IF #tempNot1_1Step2EdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := -1;
            #statLocNew := -1;
        END_IF;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := "LBP_TMSpindle".TIF[1].TMag;
            #statLocOld := "LBP_TMSpindle".TIF[1].TLoc;
        END_IF;
        #statStatus := 105;                              //Status 105
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

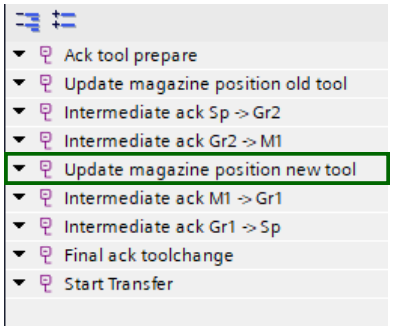
END_REGION
```

Notes:

Asynchronous transfer (Update magazine position new tool)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	4	Asynchronous transfer
TaskIdentNo	1	Channel 1
NewToolMag	DB72.DBW24	Destination magazine for tool
NewToolLoc	DB72.DBW26	Destination location for tool
OldToolMag	9998	Source magazine of tool
OldToolLoc	1	Source location of tool
Status	5	Process finished
MultitoolLoc	-1	Multitool position

PLC Programming example with acknowledgement using FC6



Update mag. Pos. new tool

```
REGION Update magazine position new tool

    // Scan rising edge
    #tempNot1_1UpdNewEdgeP := #not1_1UpdNew AND NOT #statNot1_1UpdNewFlag;
    #statNot1_1UpdNewFlag := #not1_1UpdNew;

    // Set Job Start
    IF #tempNot1_1UpdNewEdgeP THEN
        #statTask := 4;                                  //Asyn. transfer
        #statNo := 1;                                    //Channel 1
        #statMagOld := 9998;
        #statLocOld := 1;
        // No T0?
        IF "LBP_TMSpindle".TIF[1].SMag <> 0 THEN
            #statMagNew := "LBP_TMSpindle".TIF[1].SMag;
            #statLocNew := "LBP_TMSpindle".TIF[1].SLoc;
            #statStatus := 5;                            //Status 5
            #statLocMt := -1;
            #statStart := TRUE;                          //Start FC6
        END_IF;
    END_IF;

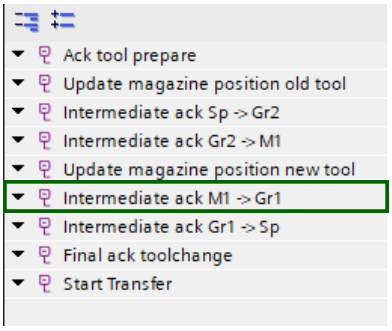
END_REGION
```

Notes:

Intermediate acknowledgement (Mag -> Gr 1)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	9998	Source magazine of new tool
NewToolLoc	2	Source location of new tool
OldToolMag	0	Destination magazine for old tool
OldToolLoc	0	Destination location for old tool
Status	105	Intermediate positions reached
MultitoolLoc	-1	Multitool position

PLC Prog example with ack using FC6



Mag -> Gr 1

```
REGION Intermediate ack M1 -> Gr1

    // Scan rising edge
    #tempNot1_1Step3EdgeP := #not1_1Step3 AND NOT #statNot1_1Step3Flag;
    #statNot1_1Step3Flag := #not1_1Step3;

    // Set Job Start
    IF #tempNot1_1Step3EdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := 9998;
            #statLocNew := 2;
        END_IF;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := -1;
            #statLocOld := -1;
        END_IF;
        #statStatus := 105;                              //Status 105
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

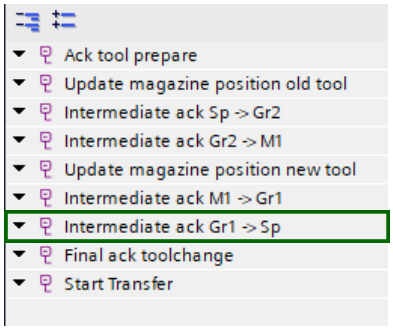
END_REGION
```

Notes:

Intermediate acknowledgement (Gr 1 -> Sp)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB72.DBW20	Source magazine of new tool
NewToolLoc	DB72.DBW22	Source location of new tool
OldToolMag	0	Destination magazine for old tool
OldToolLoc	0	Destination location for old tool
Status	105	Intermediate positions reached
MultitoolLoc	-1	Multitool position

PLC Prog example with ack using FC6



Gr 1 -> Sp

```
REGION Intermediate ack Gr1 -> Sp

    // Scan rising edge
    #tempNot1_1Step4EdgeP := #not1_1Step4 AND NOT #statNot1_1Step4Flag;
    #statNot1_1Step4Flag := #not1_1Step4;

    // Set Job Start
    IF #tempNot1_1Step4EdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                 //DB72
        #statNo := 1;                                   //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := "LBP_TMSpindle".TIF[1].BufferIdent;
            #statLocNew := "LBP_TMSpindle".TIF[1].BufferNo;
        END_IF;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := -1;
            #statLocOld := -1;
        END_IF;
        #statStatus := 105;                             //Status 105
        #statLocMt := -1;
        #statStart := TRUE;                             //Start FC6
    END_IF;

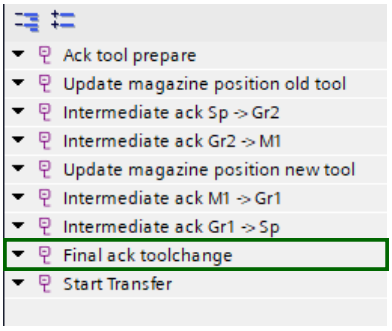
END_REGION
```

Notes:

Final acknowledgement for “Execute Change“ with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB72.DBW20	Source magazine of new tool
NewToolLoc	DB72.DBW22	Source location of new tool
OldToolMag	DB72.DBW28	Destination magazine for old tool
OldToolLoc	DB72.DBW30	Destination location for old tool
Status	1	Process finished
MultitoolLoc	DB1072.DBW10	Multitool position

PLC Prog example with ack using



Perform change

```
REGION Final ack toolchange

    // Scan rising edge
    #tempNot1_1PerformEdgeP := #not1_1Perform AND NOT #statNot1_1PerformFlag;
    #statNot1_1PerformFlag := #not1_1Perform;

    // Set Job Start
    IF #tempNot1_1PerformEdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Prepare AND
        "LBP_TMSpindle".TIF[1].Perform THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := "LBP_TMSpindle".TIF[1].BufferIdent;
            #statLocNew := "LBP_TMSpindle".TIF[1].BufferNo;
        END_IF;
        #statMagOld := "LBP_TMSpindle".TIF[1].TMag;
        #statLocOld := "LBP_TMSpindle".TIF[1].TLoc;
        #statStatus := 1;                                //Status 1
        // Multitool?
        IF "LBP_TMSpindle".TIF[1].DataInDB1072 THEN
            #statLocMt := "LBP_TMSpindleMT".MTIF[1].NewPlace;
        ELSE
            #statLocMt := -1;
        END_IF;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Notes:

Acknowledgements for Hand tools

PLC Programming example with acknowledgement using FC6

```
// Example "TMChain" [FB150] - continued

// Variables for "Handtool Addon"

// Input
//    handTOut : Bool;
//    handTIn : Bool;
//    handTInOut : Bool;

// Output
//    None

// InOut
//    None

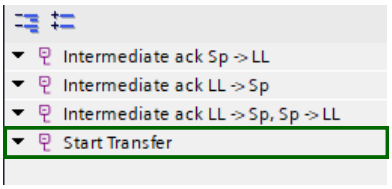
// Static
//    statHandTOutFlag : Bool;
//    statHandTInFlag : Bool;
//    statHandTInOutFlag : Bool;
//    statTask : Byte;
//    statNo : Byte;
//    statMagNew : Int;
//    statLocNew : Int;
//    statMagOld : Int;
//    statLocOld : Int;
```

```
//    statStatus : Int;
//    statLocMt : Int;
//    statStart : Bool;

// Temp
//    tempHandTOutEdgeP : Bool;
//    tempHandTInEdgeP : Bool;
//    tempHandTInOutEdgeP : Bool;
//    tempReady : Bool;
//    tempError : Int;

// Constant
//    None
```

PLC Programming example with acknowledgement using FC6



Start Transfer

```
REGION Start Transfer

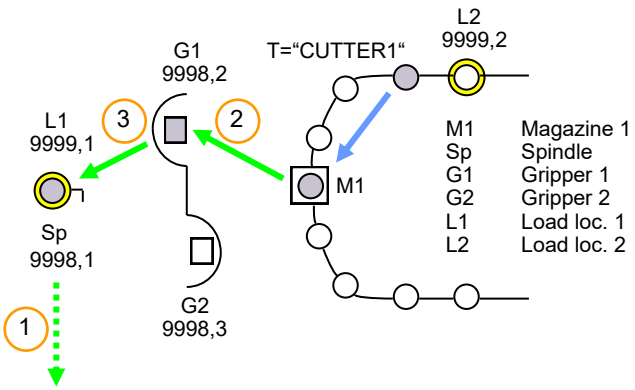
    // Tool management status transfer
    "LBP_TMTransferMT"(Start := #statStart,
                       TaskIdent := #statTask,
                       TaskIdentNo := #statNo,
                       NewToolMag := #statMagNew,
                       NewToolLoc := #statLocNew,
                       OldToolMag := #statMagOld,
                       OldToolLoc := #statLocOld,
                       Status := #statStatus,
                       MultitoolLoc := #statLocMt,
                       Ready => #tempReady,
                       Error => #tempError);

    // Reset Job Start
    IF #tempReady OR ("mcpIn".reset AND (#tempError <> 0)) THEN
        #statStart := FALSE;
    END_IF;

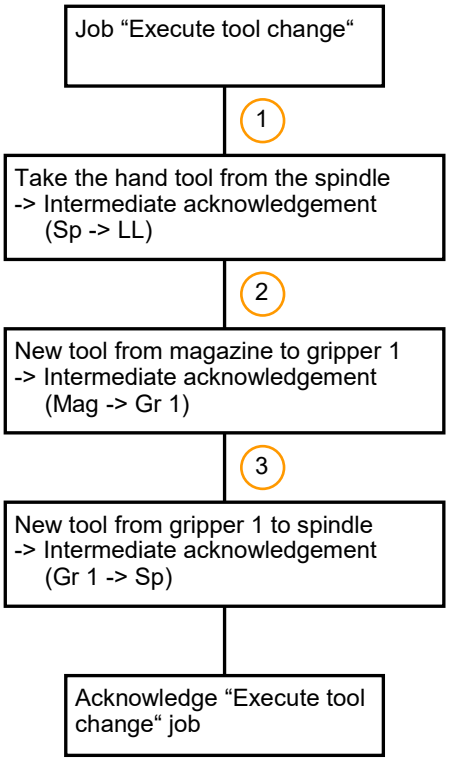
END_REGION
```

Notes:

Example: Old tool hand tool , new tool magazine tool



Sequence for tool change: Hand -> Magazine



Job: Execute tool change DB72

DB72	Signals from spindle (NCK->PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Tool re-main s in spin-dle	Re-move man-ual tool	Load man-ual tool	OldT in buffer no (n -42)	T0	Pre-pare chan ge	Per-form chan ge (init: M06)	Com pulso ry chan ge
DBW20	Buffer identification (fixed value 9998)							
DBW22	Location in buffer							
DBW24	Magazine no. (source) for new tool							
DBW26	Location no. (source) for new tool							
DBW28	Magazine no. (target) for old tool							
DBW30	Location no. (target) for old tool							
DBB42	Tool status for tool new							
	Man-ual tool	1:1 ex-change		Mas-ter tool	to be load-ed	to be un-loaded	disa-bled	ID for tool

Example: Tool change: Hand -> Magazine
DB72.DBW24 =1
DB72.DBW26 =10
DB72.DBW28 =9999
DB72.DBW30 =1

The job remains unchanged , the tools are still in their respective starting positions

Note:

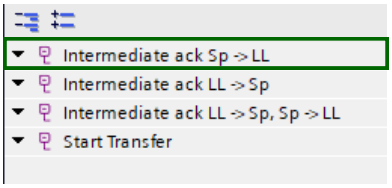
A hand tool only exists as such if it is called out in the part program, and at that time is not loaded in a magazine.

Notes:

Intermediate acknowledgement with FC6 (Sp -> LL)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	0	Source magazine of new tool
NewToolLoc	0	Source location of new tool
OldToolMag	9999	Destination magazine for old tool
OldToolLoc	1	Destination location for old tool
Status	105	Intermediate positions reached
MultitoolLoc	-1	Multitool position

PLC Prog example with ack using FC6



Sp -> LL

```
REGION Intermediate ack Sp -> LL

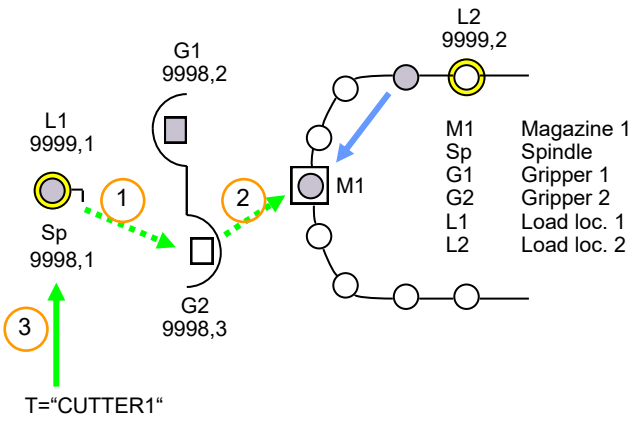
    // Scan rising edge
    #tempHandTOutEdgeP := #handTOut AND NOT #statHandTOutFlag;
    #statHandTOutFlag := #handTOut;

    // Set Job Start
    IF #tempHandTOutEdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Perform AND
        "LBP_TMSpindle".TIF[1].Prepare THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := -1;
            #statLocNew := -1;
        END_IF;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := 9999;
            #statLocOld := 1;
        END_IF;
        #statStatus := 105;                              //Status 105
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

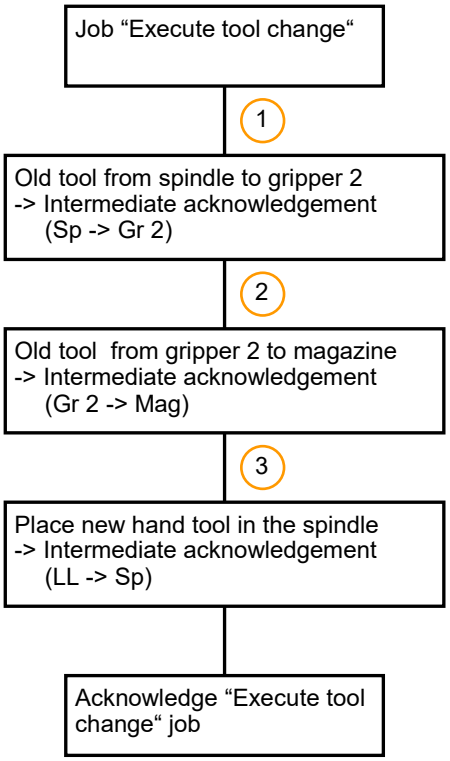
END_REGION
```

Notes:

Example: Old tool magazine tool , new tool hand tool



Sequence for tool change: Magazine -> Hand



Job: Execute tool change DB72

DB72	Signals from spindle (NCK->PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Tool re-main s in spin-dle	Re-move man-ual tool	Load man-ual tool	OldT in buffer no (n -42)	T0	Pre-pare chan ge	Per-form chan ge (init: M06)	Com pulso ry chan ge
DBW20	Buffer identification (fixed value 9998)							
DBW22	Location in buffer							
DBW24	Magazine no. (source) for new tool							
DBW26	Location no. (source) for new tool							
DBW28	Magazine no. (target) for old tool							
DBW30	Location no. (target) for old tool							
DBB42	Tool status for tool new							
	Man-ual tool	1:1 ex-change		Mas-ter tool	to be load-ed	to be un-loaded	disa-bled	ID for tool

Example: Tool change: Magazine -> Hand
DB72.DBW24 =9999
DB72.DBW26 =1
DB72.DBW28 =1
DB72.DBW30 =10

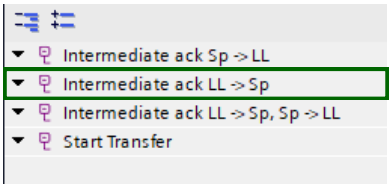
The job remains unchanged , the tools are still in their respective starting positions

Notes:

Intermediate acknowledgement with FC6 (LL -> Sp)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB72.DBW20	Source magazine of new tool
NewToolLoc	DB72.DBW22	Source location of new tool
OldToolMag	0	Destination magazine for old tool
OldToolLoc	0	Destination location for old tool
Status	105	Intermediate positions reached
MultitoolLoc	-1	Multitool position

PLC Prog example with ack using FC6



LL -> Sp

```
REGION Intermediate ack LL -> Sp

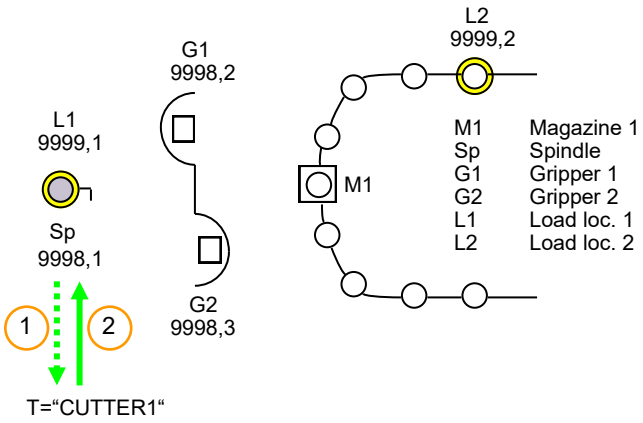
    // Scan rising edge
    #tempHandTInEdgeP := #handTIn AND NOT #statHandTInFlag;
    #statHandTInFlag := #handTIn;

    // Set Job Start
    IF #tempHandTInEdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Perform AND
        "LBP_TMSpindle".TIF[1].Prepare THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := "LBP_TMSpindle".TIF[1].BufferIdent;
            #statLocNew := "LBP_TMSpindle".TIF[1].BufferNo;
        END_IF;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := -1;
            #statLocOld := -1;
        END_IF;
        #statStatus := 105;                              //Status 105
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

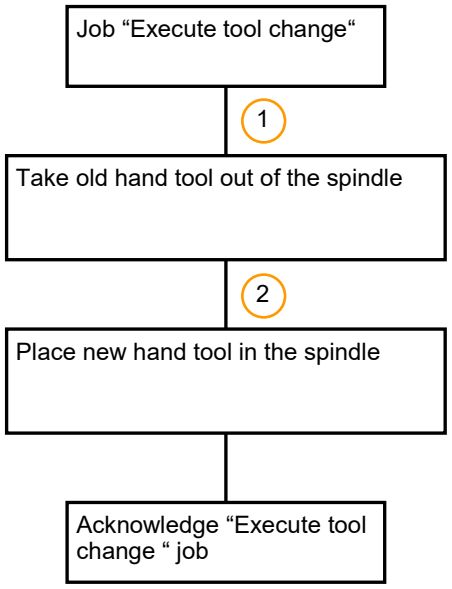
Example: Change hand tool



Job: Execute tool change

DB72	Signals from spindle (NCK->PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Tool re-mains in spin-dle	Re-move man-ual tool	Load man-ual tool	OldT in buffer no (n -42)	T0	Pre-pare chan ge	Per-form chan ge (init: M06)	Com pulso ry chan ge
DBW20	Buffer identification (fixed value 9998)							
DBW22	Location in buffer							
DBW24	Magazine no. (source) for new tool							
DBW26	Location no. (source) for new tool							
DBW28	Magazine no. (target) for old tool							
DBW30	Location no. (target) for old tool							
DBB42	Tool status for tool new							
	Man-ual tool	1:1 ex-change		Mas-ter tool	to be load-ed	to be un-loaded	disa-bled	ID for tool

Sequence for hand tool



Example: Exchange hand tools
DB72.DBW24 =9999
DB72.DBW26 =1
DB72.DBW28 =9999
DB72.DBW30 =1

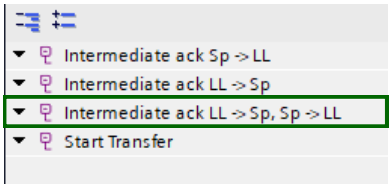
The job remains unchanged , the tools are still in their respective starting positions

Notes:

Intermediate acknowledgement (LL -> Sp, Sp -> LL)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB72.DBW20	Source magazine of new tool
NewToolLoc	DB72.DBW22	Source location of new tool
OldToolMag	DB72.DBW28	Destination magazine for old tool
OldToolLoc	DB72.DBW30	Destination location for old tool
Status	105	Intermediate positions reached
MultitoolLoc	-1	Multitool position

PLC Prog example with ack using FC6



LL -> Sp, Sp -> LL

```
REGION Intermediate ack LL -> Sp, Sp -> LL

    // Scan rising edge
    #tempHandTInOutEdgeP := #handTInOut AND NOT #statHandTInOutFlag;
    #statHandTInOutFlag := #handTInOut;

    // Set Job Start
    IF #tempHandTInOutEdgeP AND "LBP_TMSpindle".IFNo[1] AND
        "LBP_TMSpindle".TIF[1].Perform AND
        "LBP_TMSpindle".TIF[1].Prepare THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMSpindle".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := "LBP_TMSpindle".TIF[1].BufferIdent;
            #statLocNew := "LBP_TMSpindle".TIF[1].BufferNo;
        END_IF;
        // No tool in spindle?
        IF "LBP_TMSpindle".TIF[1].TMag = 0 THEN
            #statMagOld := 0;
            #statLocOld := 0;
        ELSE
            #statMagOld := 9999;
            #statLocOld := 1;
        END_IF;
        #statStatus := 105;                              //Status 105
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Acknowledgements for Reset

PLC Programming example with acknowledgement using FC6

```
// Example "TMChain" [FB150] - continued

// Variables for "Reset"

// Input
//    resetDB71IF1 : Bool;
//    resetDB71IF2 : Bool;
//    resetDB72IF1 : Bool;

// Output
//    None

// InOut
//    None

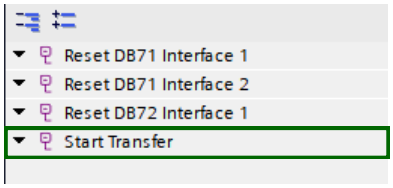
// Static
//    statResetDB71IF1Flag : Bool;
//    statResetDB71IF2Flag : Bool;
//    statResetDB72IF1Flag : Bool;
//    statTask : Byte;
//    statNo : Byte;
//    statMagNew : Int;
//    statLocNew : Int;
//    statMagOld : Int;
//    statLocOld : Int;
```

```
//    statStatus : Int;
//    statLocMt : Int;
//    statStart : Bool;

// Temp
//    tempResetDB71IF1EdgeP : Bool;
//    tempResetDB71IF2EdgeP : Bool;
//    tempResetDB72IF1EdgeP : Bool;
//    tempReady : Bool;
//    tempError : Int;

// Constant
//    None
```

PLC Programming example with acknowledgement using FC6



Start Transfer

```
REGION Start Transfer

    // Tool management status transfer
    "LBP_TMTransferMT"(Start := #statStart,
                       TaskIdent := #statTask,
                       TaskIdentNo := #statNo,
                       NewToolMag := #statMagNew,
                       NewToolLoc := #statLocNew,
                       OldToolMag := #statMagOld,
                       OldToolLoc := #statLocOld,
                       Status := #statStatus,
                       MultitoolLoc := #statLocMt,
                       Ready => #tempReady,
                       Error => #tempError);

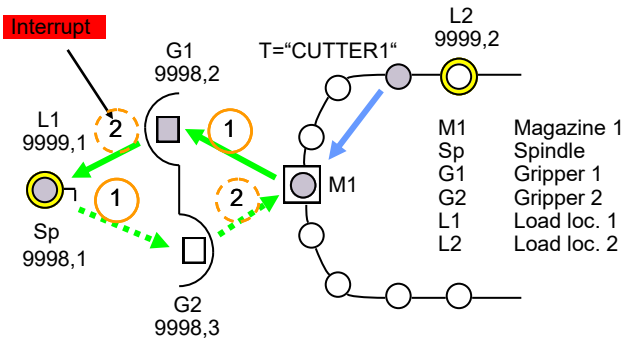
    // Reset Job Start
    IF #tempReady OR ("mcpIn".reset AND (#tempError <> 0)) THEN
        #statStart := FALSE;
    END_IF;

END_REGION
```

Notes:

Interruption to the tool change

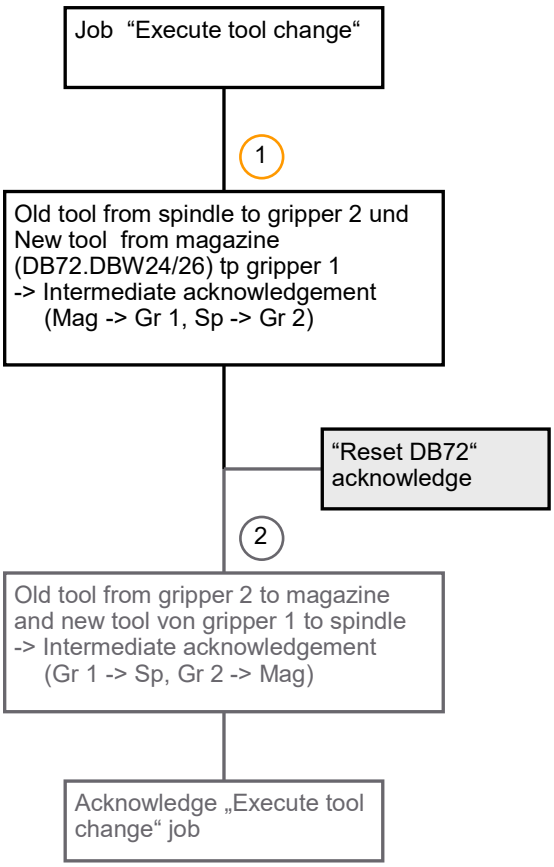
Example: Abort 1:1 tool change



Examples of what can cause the tool change to be aborted

- Reset
- E-Stop
- Power failure

Sequence: Abort tool change



Notes:

Job: “Execute tool change“

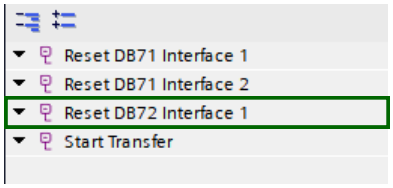
DB72	Signals from spindle (NCK->PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Tool re-mains in spin-dle	Re-move man-ual tool	Load man-ual tool	OldT in buffer no (n -42)	T0	Pre-pare chan ge	Per-form chan ge (initia te: M06)	Com pulso ry chan ge
DBB5	Data in DB10 72	Reserved						Ack. Sta-tus 3
DBW20	Buffer identification (fixed value 9998)							
DBW22	location in buffer (INT)							
DBW24	Location no. (source) for new tool (INT)							
DBW26	Magazine no. (target) for old tool (INT)							
DBW28	Location no. (target) for old tool (INT)							
DBW30	Tool new: location type (INT)							

Acknowledgement Reset DB72 with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	2	DB72 Interface
TaskIdentNo	1	Interface 1
NewToolMag	-1	When process abort =-1
NewToolLoc	-1	When process abort =-1
OldToolMag	-1	When process abort =-1
OldToolLoc	-1	When process abort =-1
Status	3	Process aborted
MultitoolLoc	-1	When process abort =-1

Example: Tool change with 1:1 change
DB72.DBW24 =1
DB72.DBW26 =10
DB72.DBW28 =1
DB72.DBW30 =10

PLC Programming example with acknowledgement using FC6



Reset DB72 IF1

```
REGION Reset DB72 Interface 1

    // Scan rising edge
    #tempResetDB72IF1EdgeP := #resetDB72IF1 AND NOT #statResetDB72IF1Flag;
    #statResetDB72IF1Flag := #resetDB72IF1;

    // Set Job Start
    IF #tempResetDB72IF1EdgeP AND "LBP_TMSpindle".IFNo[1] THEN
        #statTask := 2;                                  //DB72
        #statNo := 1;                                    //Interface 1
        #statMagNew := -1;
        #statLocNew := -1;
        #statMagOld := -1;
        #statLocOld := -1;
        #statStatus := 3;                                //Status 3
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Acknowledgements for Recovery functions

PLC Programming example with acknowledgement using FC6

```
// Example "TMChain" [FB150] - continued

// Variables for "Recovery"

// Input
//    saveData : Bool;
//    asynStep1 : Bool;
//    asynStep2 : Bool;
//    asynStep3 : Bool;
//    asynStep4 : Bool;

// Output
//    None

// InOut
//    None

// Static
//    statSaveDataFlag : Bool;
//    statAsynStep1Flag : Bool;
//    statAsynStep2Flag : Bool;
//    statAsynStep3Flag : Bool;
//    statAsynStep4Flag : Bool;
```

```
//    statTask : Byte;
//    statNo : Byte;
//    statMagNew : Int;
//    statLocNew : Int;
//    statMagOld : Int;
//    statLocOld : Int;
//    statStatus : Int;
//    statLocMt : Int;
//    statStart : Bool;

// Temp
//    tempSaveDataEdgeP : Bool;
//    temptAsynStep1EdgeP : Bool;
//    tempAsynStep2EdgeP : Bool;
//    tempAsynStep3EdgeP : Bool;
//    tempAsynStep4EdgeP : Bool;
//    tempReady : Bool;
//    tempError : Int;

// Constant
//    None
```

PLC Programming example with acknowledgement using FC6



Start Transfer

```
REGION Start Transfer

    // Tool management status transfer
    "LBP_TMTransferMT"(Start := #statStart,
                       TaskIdent := #statTask,
                       TaskIdentNo := #statNo,
                       NewToolMag := #statMagNew,
                       NewToolLoc := #statLocNew,
                       OldToolMag := #statMagOld,
                       OldToolLoc := #statLocOld,
                       Status := #statStatus,
                       MultitoolLoc := #statLocMt,
                       Ready => #tempReady,
                       Error => #tempError);

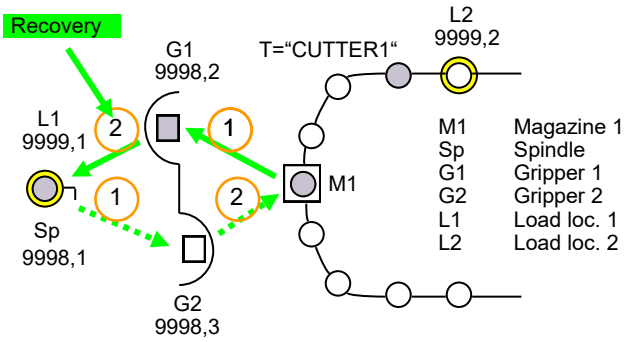
    // Reset Job Start
    IF #tempReady OR ("mcpIn".reset AND (#tempError <> 0)) THEN
        #statStart := FALSE;
    END_IF;

END_REGION
```

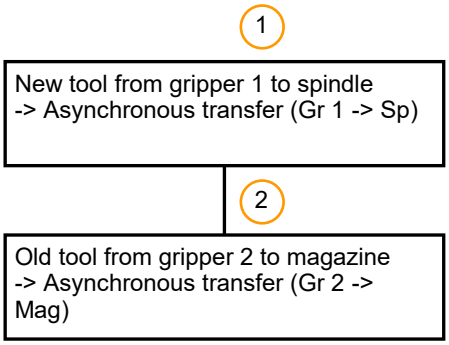
Notes:

Recovery Functions

Example: Tool changer recovery



Sequence: Tool changer recovery



Note:

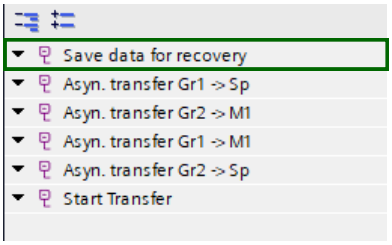
Asynchronous transfers are used to inform the Tool Management of tool movements which have been initiated by the PLC without a specific job request from Tool Management in the NC.

The DATA DB is created and used for the examples.

TMData

	Name	Data type	Offset	Start value
1	Static			
2	RecMagNew	Int	0.0	0
3	RecLocNew	Int	2.0	0
4	RecMagOld	Int	4.0	0
5	RecLocOld	Int	6.0	0

PLC Programming example with acknowledgement using FC6



Initialize data

```
REGION Save data for recovery

    // Scan rising edge
    #tempSaveDataEdgeP := #saveData AND NOT #statSaveDataFlag;
    #statSaveDataFlag := #saveData;

    // Set Job Start
    IF #tempSaveDataEdgeP AND "LBP_TMSpindle".IFNo[1] THEN
        "TMData".RecMagNew := "LBP_TMSpindle".TIF[1].SMag;
        "TMData".RecLocNew := "LBP_TMSpindle".TIF[1].SLoc;
        "TMData".RecMagOld := "LBP_TMSpindle".TIF[1].TMag;
        "TMData".RecLocOld := "LBP_TMSpindle".TIF[1].TLoc;
    END_IF;

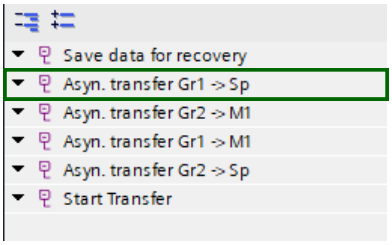
END_REGION
```

Notes:

Asynchronous transfer with FC6 (Gr 1 -> Sp)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	4	Asynchronous transfer
TaskIdentNo	1	Channel 1
NewToolMag	9998	Destination magazine for tool
NewToolLoc	1	Destination location for tool
OldToolMag	9998	Source magazine of tool
OldToolLoc	2	Source location of tool
Status	1	Process finished
MultitoolLoc	-1	Multitool position

PLC Programming example with acknowledgement using FC6



Gr 1 -> Sp

```
REGION Asyn. transfer Gr1 -> Sp

    // Scan rising edge
    #tempAsynStep1EdgeP := #asynStep1 AND NOT #statAsynStep1Flag;
    #statAsynStep1Flag := #asynStep1;

    // Set Job Start
    IF #tempAsynStep1EdgeP THEN
        #statTask := 4;                                  //Asyn. transfer
        #statNo := 1;                                    //Channel 1
        #statMagOld := 9998;
        #statLocOld := 2;
        #statMagNew := 9998;
        #statLocNew := 1;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Asynchronous transfer with FC6 (Gr 2 -> Mag)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	4	Asynchronous transfer
TaskIdentNo	1	Channel 1
NewToolMag	DB151.DBW4	Destination magazine for tool
NewToolLoc	DB151.DBW6	Destination location for tool
OldToolMag	9998	Source magazine of tool
OldToolLoc	3	Source location of tool
Status	1	Process finished
MultitoolLoc	-1	Multitool position

PLC Programming example with acknowledgement using FC6



Gr 2 -> Mag

```
REGION Asyn. transfer Gr2 -> M1

    // Scan rising edge
    #tempAsynStep2EdgeP := #asynStep2 AND NOT #statAsynStep2Flag;
    #statAsynStep2Flag := #asynStep2;

    // Set Job Start
    IF #tempAsynStep2EdgeP THEN
        #statTask := 4;                                  //Asyn. transfer
        #statNo := 1;                                    //Channel 1
        #statMagOld := 9998;
        #statLocOld := 3;
        #statMagNew := "TMData".RecMagOld;
        #statLocNew := "TMData".RecLocOld;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

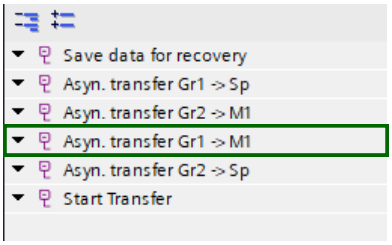
END_REGION
```

Notes:

Asynchronous transfer with FC6 (Gr 1 -> Sp)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	4	Asynchronous transfer
TaskIdentNo	1	Channel 1
NewToolMag	DB151.DBW0	Destination magazine for tool
NewToolLoc	DB151.DBW2	Destination location for tool
OldToolMag	9998	Source magazine of tool
OldToolLoc	2	Source location of tool
Status	1	Process finished
MultitoolLoc	-1	Multitool position

PLC Programming example with acknowledgement using FC6



Gr 1 -> Mag

```
REGION Asyn. transfer Gr1 -> M1

    // Scan rising edge
    #tempAsynStep3EdgeP := #asynStep3 AND NOT #statAsynStep3Flag;
    #statAsynStep3Flag := #asynStep3;

    // Set Job Start
    IF #tempAsynStep3EdgeP THEN
        #statTask := 4;                                  //Asyn. transfer
        #statNo := 1;                                    //Channel 1
        #statMagOld := 9998;
        #statLocOld := 2;
        #statMagNew := "TMData".RecMagNew;
        #statLocNew := "TMData".RecLocNew;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

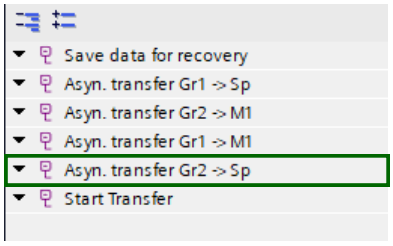
END_REGION
```

Notes:

Asynchronous transfer with FC6 (Gr 2 -> Sp)

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	4	Asynchronous transfer
TaskIdentNo	1	Channel 1
NewToolMag	9998	Destination magazine for tool
NewToolLoc	1	Destination location for tool
OldToolMag	9998	Source magazine of tool
OldToolLoc	3	Source location of tool
Status	1	Process finished
MultitoolLoc	-1	Multitool position

PLC Programming example with acknowledgement using FC6



Gr 2 -> Sp

```
REGION Asyn. transfer Gr2 -> Sp

    // Scan rising edge
    #tempAsynStep4EdgeP := #asynStep4 AND NOT #statAsynStep4Flag;
    #statAsynStep4Flag := #asynStep4;

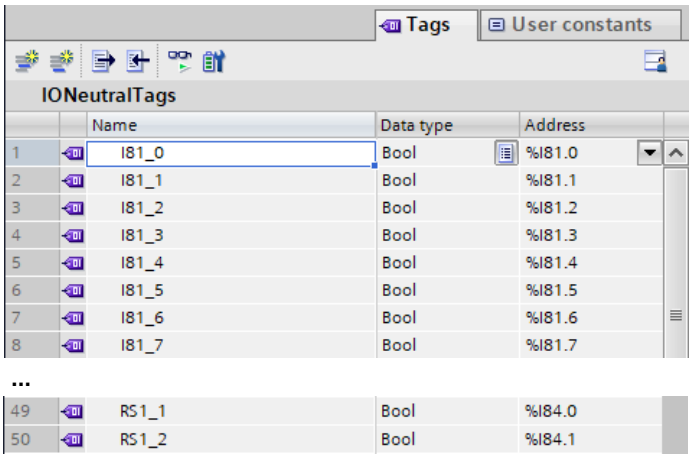
    // Set Job Start
    IF #tempAsynStep4EdgeP THEN
        #statTask := 4;                                  //Asyn. transfer
        #statNo := 1;                                    //Channel 1
        #statMagOld := 9998;
        #statLocOld := 3;
        #statMagNew := 9998;
        #statLocNew := 1;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

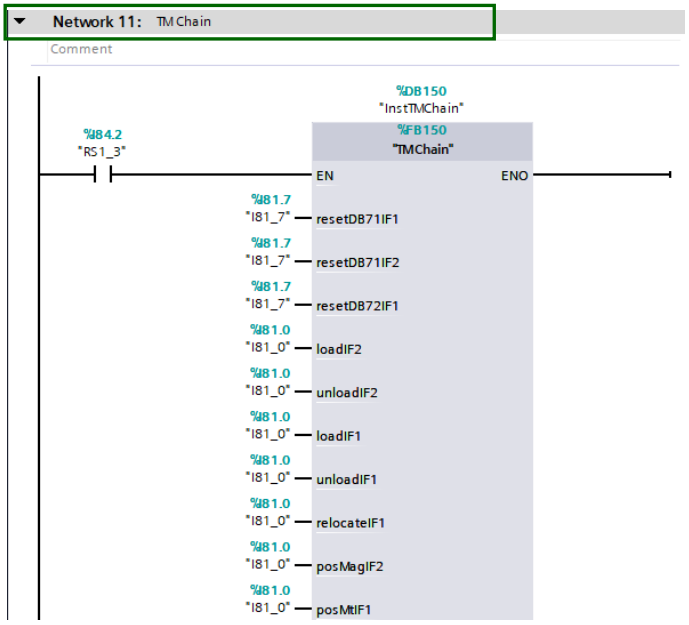
Notes:

Calling TM functions in OB1

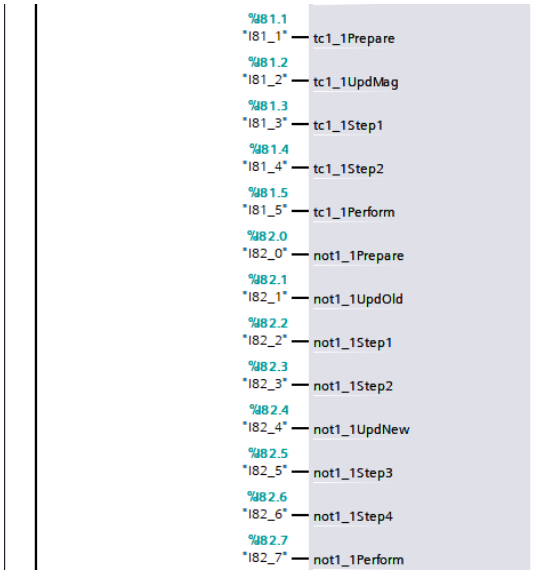
Create a tag table for the TM functions.

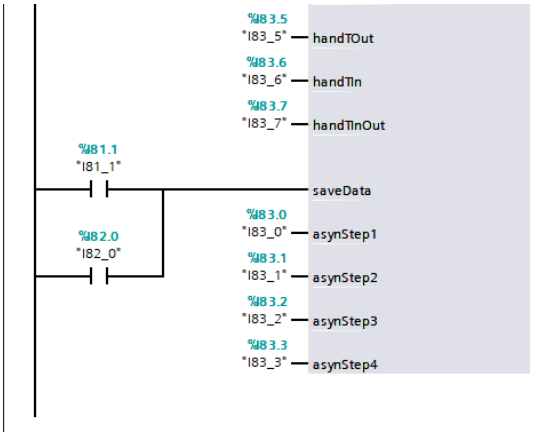


The newly created TM functions are called in OB1.



OB1 - continued.

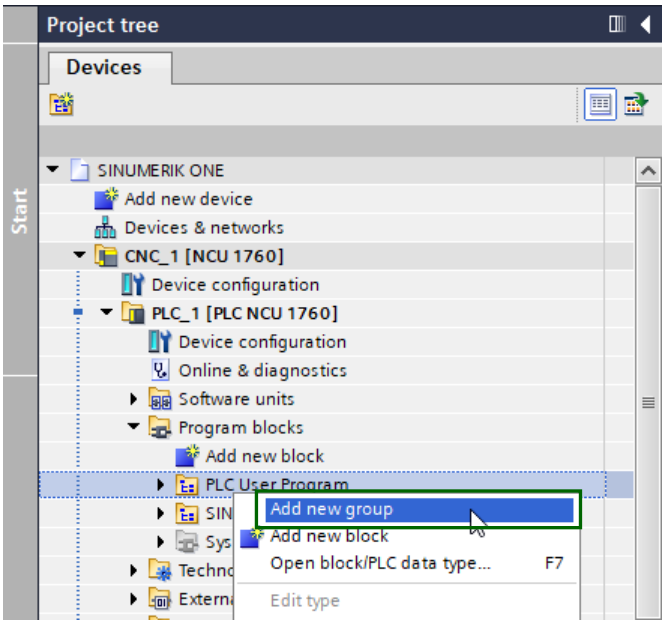




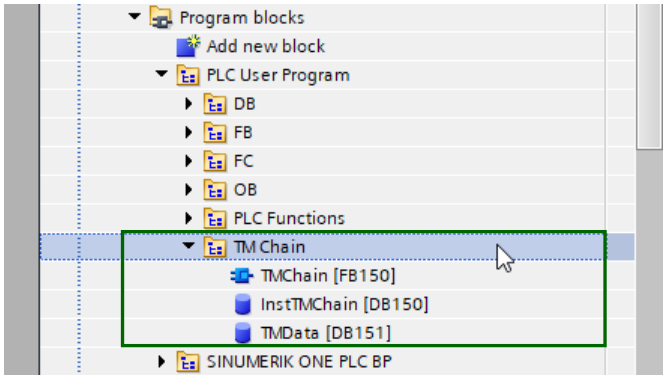
Notes:

Grouping TM functions

Create a new group in the project navigation.



Insert TM functions into the new group.



Notes:

Customizing the HMI

The following screen areas of tool management can be customized to suit particular machine configurations.

- Reconfigure the tool list
- Modify the list parameters
- Reconfigure a screen “Details“
- Reconfigure a screen “New tool - favorites“
- Reconfigure tool types
- Give names to magazine locations
- Generate custom texts

Procedure:

- Activate OEM tool data
- Create the configuration file
- Create the custom text file

The customization of the HMI is accomplished by making changes to these configuration files.

- Technology Milling - **sltmlistconfig.xml**
- Technology Turning - **sltmturninglistconfig.xml**

Template:

There is a template located on the controller in this directory

Configuration file “**oem_sltmlistconfig.xml**“
“System CF Card\siemens\sinumerik\hmi\template\cfg“

Custom text file “**oem_sltmlistdialog_xxx.ts**“
“System CF Card\siemens\sinumerik\hmi\template\lng“

Activate OEM tool data

Example: Activate OEM tool parameter $TC_TPC1

In this example the OEM Tool list will be enabled and the user parameter “ToolUser_1“ will be displayed in an new column. The parameter “ToolUser_1“ refers to the OEM tool parameter $TC_TPC1.

The OEM tool parameters are activated via these NC machine data:

- MD18080 Bit 2=1 OEM-/CC- data available
- MD20310 Bit 2=1 OEM-/CC- functions active
- MD18094 = 2 Number of OEM tool parameters

Notes:

Create the configuration file

Example: OEM tool list

In this example the OEM Tool list will be enabled and the user parameter “ToolUser_1“ will be displayed in an new column. The parameter “ToolUser_1“ refers to the OEM tool parameter $TC_TPC1.

Filename of configuration file:

- **sltmlistconfig.xml** for Technology Milling
- **sltmturninglistconfig.xml** for Technology Turning

Copy the configuration file “sltmlistconfig.xml“ into the user or oem directory.

OEM directory
“System CF Card\oem\sinumerik\hmi\cfg“

User directory
“System CF Card\user\sinumerik\hmi\cfg“

Example: Configuration file “sltmlistconfig.xml“

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<CONFIGURATION>

<!--  ************************************************************ -->
<!--  configuration file of toollist gui in sinumerik operate      -->
<!--  ************************************************************ -->

<!--  ************************************************************ -->
<!--  Enable tool data OEM list                                    -->
<!--  ************************************************************ -->
    <LISTCONFIGURATION>
        <SlTmTooloemForm>
            <Enabled value="true" type="bool" />
        </SlTmTooloemForm>
    </LISTCONFIGURATION>

<!--  ************************************************************ -->
<!--  OEM tool parameter 1 in tool data OEM list                   -->
<!--  ************************************************************ -->
    <LISTCONFIGURATION>
        <SlTmTooloemForm>
            <INSERT_COLUMN>
                <COLUMN6>
                    <TOOLTYPE_ALL>
                        <Item value="ToolUser_1" type="QString" />
                    </TOOLTYPE_ALL>
                </COLUMN6>
            </INSERT_COLUMN>
        </SlTmTooloemForm>
    </LISTCONFIGURATION>

</CONFIGURATION>
```

Enable OEM tool list

Define column for user parameter “ToolUser_1“

Notes:

Create the custom text file

Example: Custom texts for OEM tool list

In this example the custom texts column header, tooltip and tool details are configured for the OEM parameter “ToolUser_1“.

Filename of Custom text file “sltmlistdialog_xxx.ts“ (xxx language code):

- **sltmlistdialog_deu.ts** for german custom texts
- **sltmlistdialog_eng.ts** for english custom texts

Copy the custom text file “sltmlistdialog_xxx.ts“ into the user or oem directory

OEM directory
“System CF Card\oem\sinumerik\hmi\lng“

User directory
“System CF Card\user\sinumerik\hmi\lng“

Example: Custom text file “sltmlistdialog_eng.ts“

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS><TS>

<!--  ************************************************************ -->
<!--  Definition of OEM-Texts                                      -->
<!--  ************************************************************ -->
<context>
    <name>SlTmListForm</name>

<!--  ************************************************************ -->
<!--  Parameter ToolUser_1                                         -->
<!--  ************************************************************ -->
    <message>
        <source>TM_HL_TOOL_USER_1</source>
        <translation>My OEM 1</translation>
        <chars>30</chars>
    </message>
    <message>
        <source>TM_TT_TOOL_USER_1</source>
        <translation>My OEM 1 tooltip</translation>
        <chars>30</chars>
    </message>
    <message>
        <source>TM_HL_TOOL_USER_1_DETAILS</source>
        <translation>My OEM 1 parameter</translation>
        <chars>30</chars>
    </message>

</context>
</TS>
```

Headline

Tooltip

Tool details

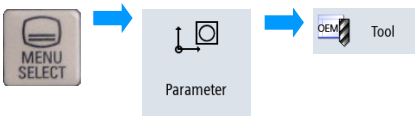
Notes:

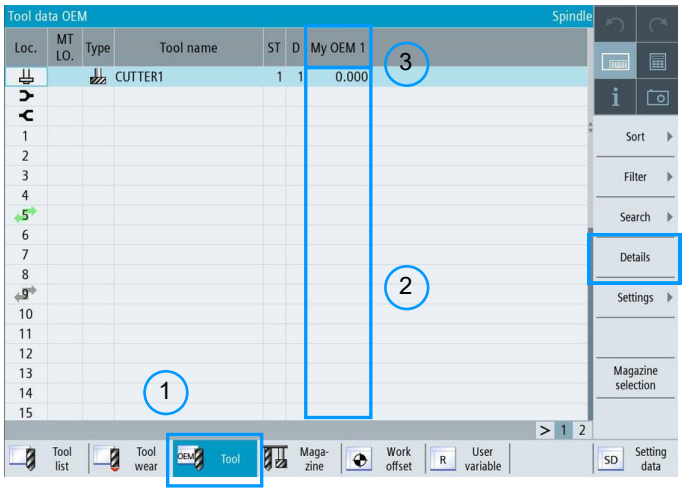
Updating the display on the controller

After all configuration files have been loaded a re-boot is required for the display on the controller. This can be achieved either by means of the Softkey "Reboot HMI“ or by switching the machine OFF and back ON again.

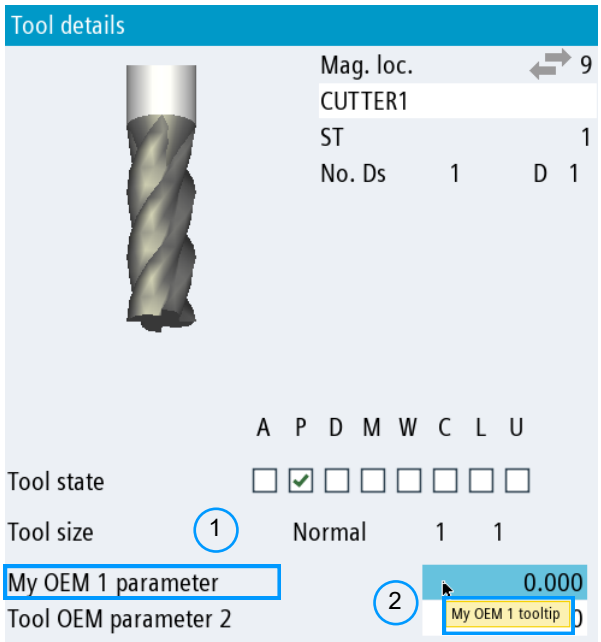


Display on the controller





1. New softkey “OEM Tool“ is displayed
2. New column for OEM parameter 1 is displayed
3. The configured Column heading is displayed



1. The configured descriptive text will be displayed.
2. The configured tool tip text will be displayed.

Notes:

Notes:

Tool change with a turret

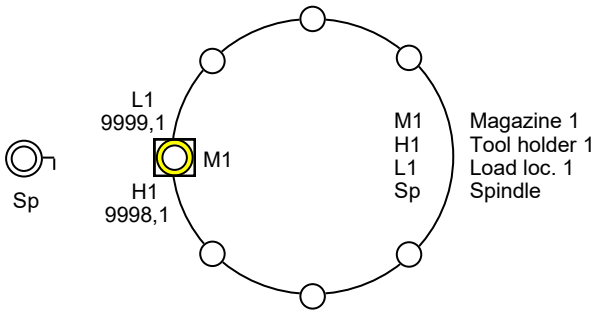
On a lathe, the tool change is handled by the turret which is configured in the magazine configuration.

DB73 contains an interface area for each turret which has been configured.

All interfaces are set up for the „Execute tool change“ command, which is triggered by the NC instruction “T-number“.

In order to inform the Tool Management of tool movements the following functions are used in the machine PLC program FC6 (Multitool), FC7 (Turret) and FC8. Standard end acknowledgment via DB73 is also available.

Example: Turret with 12 locations



Procedure:

- Setup the machine data for the Tool Management
- Configure the turret magazine
- Tool load / unload
- Generate the TCHANGE program
- Tool change with turret

Notes:

Machine data for Tool management

Overview of the machine data

MD	Identifier	Settings	Comment	Description
10717	$MN_T_NO_FCT_CYCLE_NAME	TCHANGE		Name of tool-changing cycle for T function replacement
15710	$MN_TCA_CYCLE_NAME	TCHANGE		Name of tool-changing cycle for TCA function replacement
17500	$MN_MAXNUM_REPLACEMENT_TOOLS	0		Maximum number of replacement tools
17530	$MN_TOOL_DATA_CHANGE_COUNTER	1FH		Mark tool data change for HMI
18080	$MN_MM_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 10 = 1		Memory reservation for tool management (SRAM) - Bit 0: Provide magazine management data - Bit 1: Provide monitoring data - Bit 3: Consider adj. Location - Bit 10: The multitool function is available
18082	$MN_MM_NUM_TOOL	100		Number of tools the NCK can manage (SRAM)
18083	$MN_MM_NUM_MULTITOOL	15		Number of multitools the NCK can manage (SRAM)
18084	$MN_MM_NUM_MAGAZINE	3		Number of magazines the NCK can manage (SRAM)
18085	$MN_MM_NUM_MULTITOOL_LOCATIONS	30		Number of multitool locations the NCK can manage (SRAM)
18086	$MN_MM_NUM_MAGAZINE_LOCATION	14		Number of magazine locations the NCK can manage (SRAM)
18100	$MN_MM_NUM_CUTTING_EDGES_IN_TOA	200		Tool offsets in the TO range (SRAM)
20124	$MC_TOOL_MANAGEMENT_TOOLHOLDER	1		Tool holder number
20310	$MC_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 14 = 1 Bit 16 = 1		Activation of tool management functions - Bit 0: Magazine management active - Bit 1: Monitoring functions active - Bit 3: Adjacent location treatm. - Bit 14: Automatic tool change during RESET and Start - Bit 16: Programming T="Location number"
20320	$MC_TOOL_TIME_MONITOR_MASK	Bit 0 = 1		Time monitoring for tool in tool holder - Bit 0: Monitoring tool in tool holder 1
22550	$MC_TOOL_CHANGE_MODE	0		New tool offset with T function
28450	$MC_MM_TOOL_DATA_CHG_BUFF_SIZE	400		Buffer for tool data changes (DRAM)

Notes:

Set up the machine data for the tool management

The various Tool Management functions are activated via machine data.

Prerequisite:

- License key for the “Tool Management, more than 4 Magazines“ option is available (if necessary).

Procedure:

- Activate the Option “Tool Management, more than 4 Magazines“ (if necessary)
- Activate Turning Technology
- Activate Tool Management
- Set the number of magazines and magazine locations
- Set the number of tools and the number of cutting edges

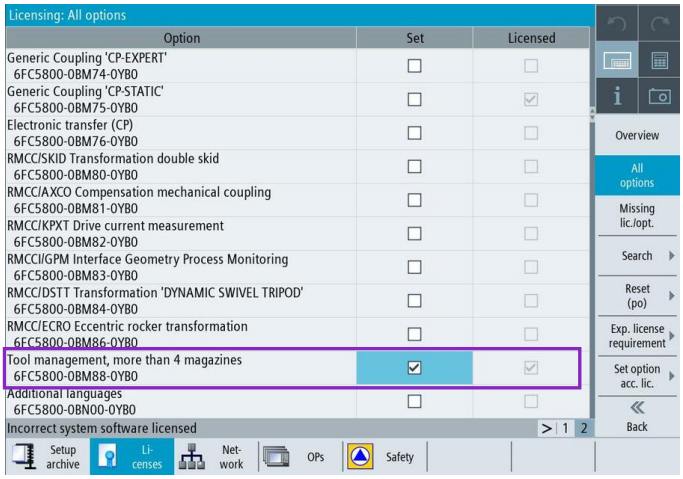
Activate the option “Tool Management, more than 4 magazines“ (if necessary)

The standard function includes 4 magazines. Buffers and loading magazines count as magazines.

Magazine 1, 2
Magazine 3 (Option)
Buffer magazine (9998)
Load magazine (9999)

The option is activated in the menu “Licenses“ in the “Setup“ area . A valid license key is required.

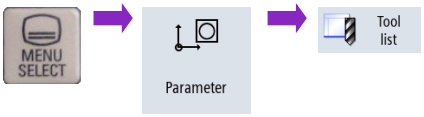


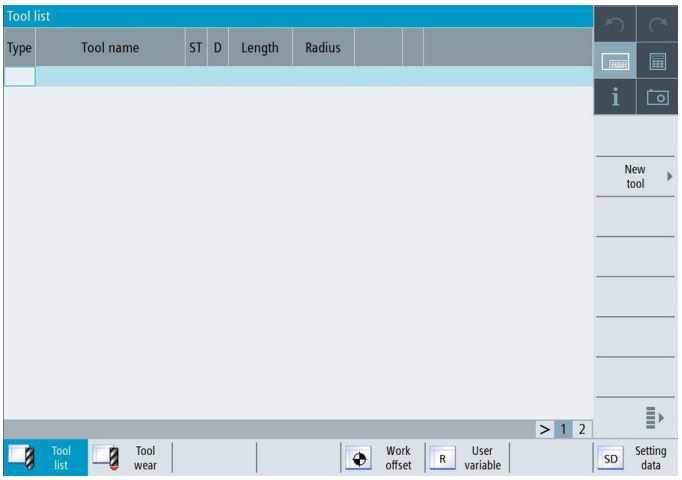


Notes:

Activate Tool Management

By default the Tool Management without magazine management is active, i.e. only tool monitoring is active (MD18080 Bit 1=1, MD20310 Bit 1=1).





Enable Tool Management

Tool Management with magazine management requires these minimum machine data settings.

Reserve memory for Tool Management

MD18080	Memory organisation for Tool Management
Bit 0 = 1	Reserve memory for magazine management
Bit 1 = 1	Reserve memory for tool monitoring
Bit 3 = 1	Reserve memory for adjacent location check
Bit 10 = 1	Multitool function is enabled

Note:
MD18080 and MD20310 Bits 0-3 must always be the same.

Activate Tool Management

MD20310	Activation of Tool Management
Bit 0 = 1	Magazine management active
Bit 1 = 1	Tool monitoring active
Bit 3 = 1	Adjacent location check

Check the HMI settings for changes to tool data

MD17530	Tool data change counter
Bit 0 = 1	Change tool status
Bit 1 = 1	Change residual number of tools
Bit 2 = 1	Tool data in update service
Bit 3 = 1	Magazine data in update service
Bit 4 = 1	ISO data in update service

Buffer for changes to tool data

MD28450 = 400	Number of entries in the buffer for HMI update. This MD concerns how the HMI shows data of tools, whose data is changed via part program instructions (e.g. NEWT).

Activation of the tool cutting time monitor

MD20320	Cutting time monitor for tool in tool holder
Bit 0 = 1	Cutting time monitor for tool in tool holder 1

Assigning the TO Blocks

MD28085 = 1	Assignment of TO blocks to channels The TO area includes all tool and magazine data known to the NCK. The maximum number of TO blocks is equal to the number of channels currently enabled. 1, 1: TO block 1 is assigned to channels 1 and 2. 1, 2: TO block 1 is assigned to channels 1, TO block 2 is assigned to channels 2

Notes:

Define the number of magazines and magazine locations

MD18084 = 3	Number of magazines. Note: Min. 3 magazines incl. buffer and load magazines.
MD18086 = 14	Number of magazine locations. Note: Add together all locations in all magazines.

Example: Turret with 12 locations

MD18084 = 3
- Turret magazine (magazine 1)
- Buffer magazine (magazine 9998)
- Load magazine (magazine 9999)

MD18086 = 14
- 12 locations for turret magazine
- 1 location for buffer magazine (tooholder)
- 1 location for load magazine (load location at turret magazine)

Note:
Assigning the magazines to the locations is done in the magazine configuration.

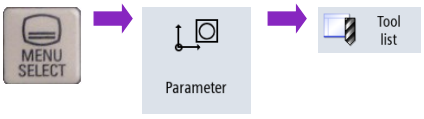
Define the max. number of tools and cutting edges

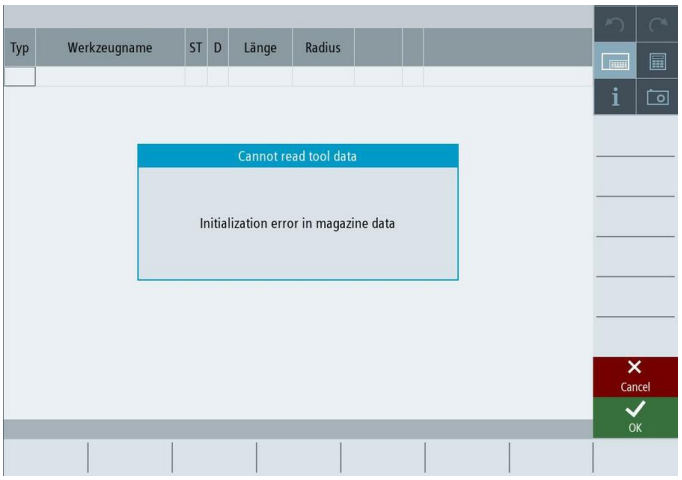
MD18082 = 100	Max number of tools which the NC can handle. Note: Each tool has at least 1 cutting edge.
MD18100 = 200	Number of cutting edges in theTO block

Suggested values for JobShop applications

MD18082 = 100 Number of tools
MD18100 = 200 Number of edges

Once Tool Management has been successfully activated, the magazine data have to be configured. The following alarm indicated this.





Magazine data missing in the NC.

Notes:

Magazine Configuration

The magazine configuration can be accomplished with the operating software “SINUMERIK Operate“.

The magazine configuration contains:

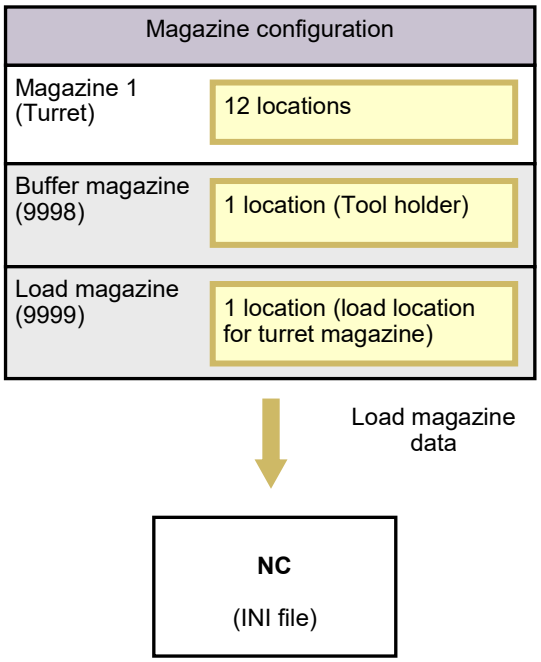
- Definition of the magazines
- Definition of the magazine locations
- Definition of the buffer locations
- Definition of the load locations
- Definition of the search strategy

Procedure:

- Create the magazine configuration
- Load magazine data into the NC

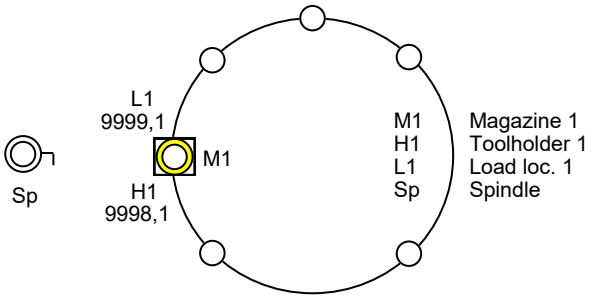
Magazine configuration: how to create, how to load

Example: Turret with 12 locations



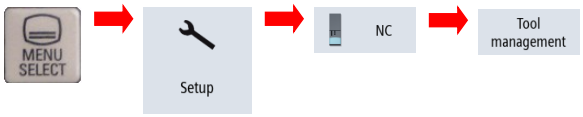
Configure the turret magazine

Example: Turret with 12 locations



Notes:

Start the magazine configuration





The Tool management menu contains the following areas:
Active data contains the tool management of the NC
Examples contains examples for the tool management
Network / USB contains the connections

Create a new magazine configuration

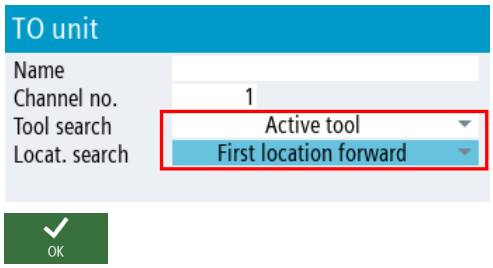
Position the cursor on the "Active data" folder and press the "New element" softkey.



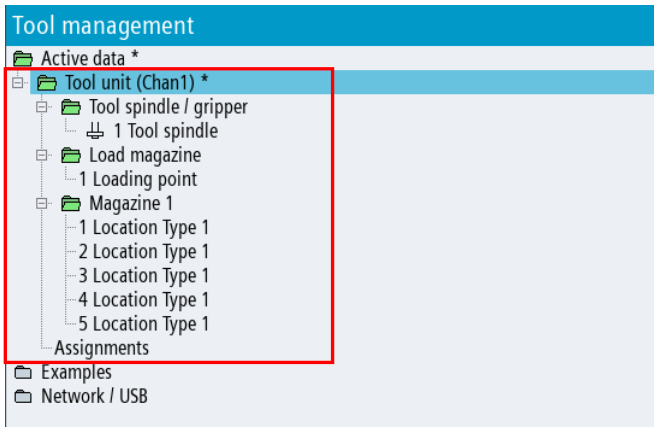
Configure the TO unit

Set the search strategy and accept with "OK".

Channel number input a value of “1”
Tool search select “Active tool”
Location search select “First location forward“



A magazine configuration is created that consists of a magazine with 5 locations, as well as a tool spindle and a loading point.

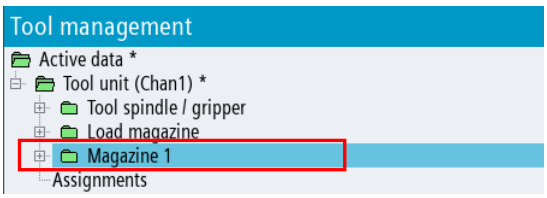


This magazine configuration can now be adapted to the conditions on the machine.

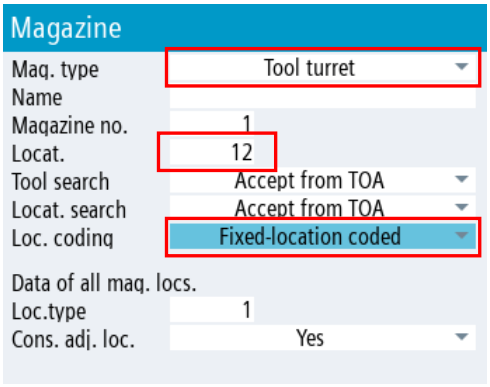
Notes:

Configure Magazine 1

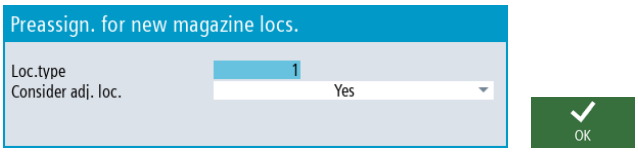
Position the cursor on the "Magazine 1" folder and open the element with the "Input" key.



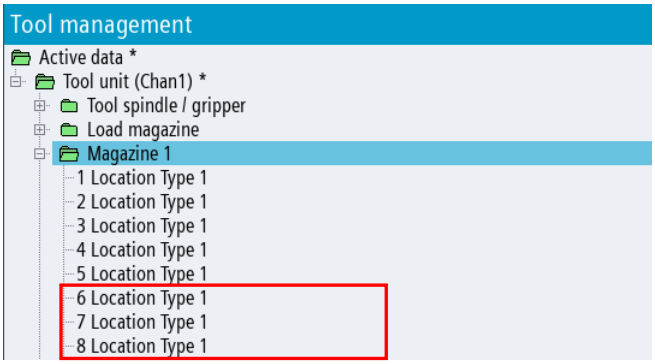
Set the number of locations, the magazine type and type of location coding and accept with "OK".



Accept the preassignments for the new magazine locations for Location type and Consider adjacent location with "OK".



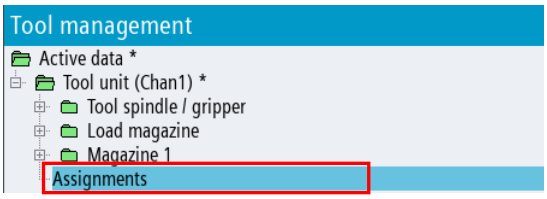
The new locations are added to the magazine.



The location type can be defined by the OEM. To do this, the cursor must be positioned on the appropriate location and the element opened.

Configure the assignments

Position the cursor on the "Assignments" folder and open the element with the "Input" key.



Specify the assignments of the magazine and accept with "OK".



The creation of the magazine configuration is now completed.

Notes:

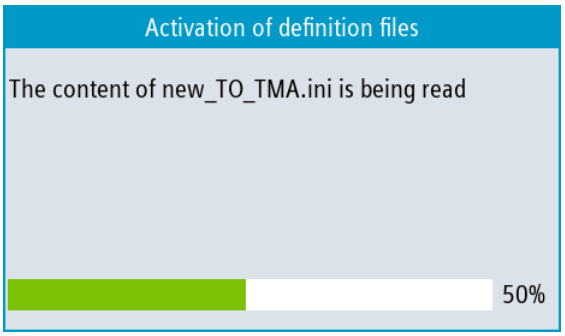
Activate the data

Position the cursor on the "Active data" folder and press the "Activate data" softkey.



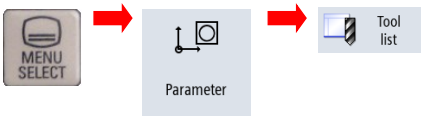


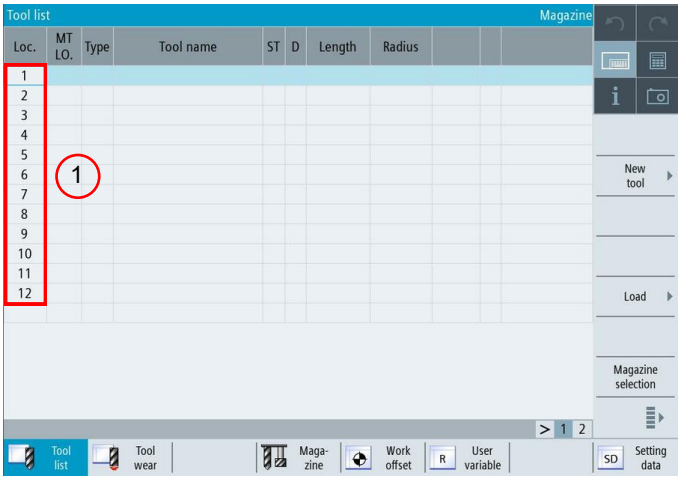
The definition files are read in.



After successfully loading the configuration file, all magazines and buffers in the tool list are displayed in the Parameters area.

Display on the controller





1. Turret with 12 locations is displayed

Notes:

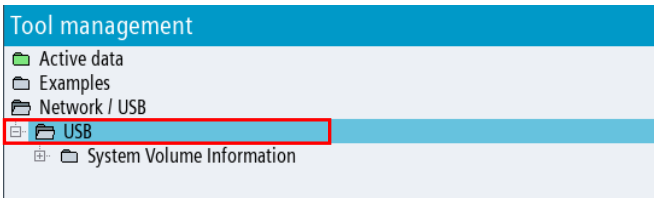
Back up the data

Position the cursor on the "Active data" folder and press the "Copy" softkey.



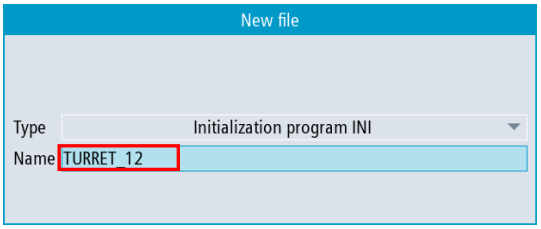
Copy

Position the cursor on the "USB" folder and press the "Paste" softkey.

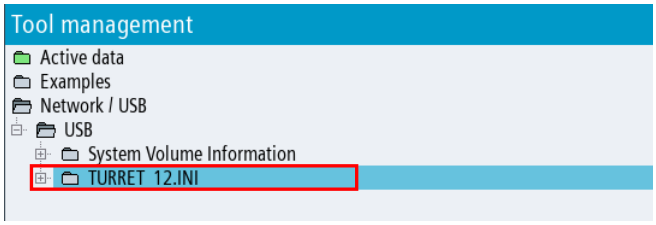


Paste

Enter the name for Initialization program INI and accept with "OK".



The INI file is created and saved to the USB stick.



Notes:

Configuration file (INI file)

```
CHANDATA(1)

; delete old data
$TC_MAP1[0]=0
$TC_DP1[0,0]=0

$TC_MAMP1=""
$TC_MAMP2=257
$TC_MAMP3=0

$TC_MAP1[1]=3
$TC_MAP2[1]=""
$TC_MAP3[1]=80
$TC_MAP6[1]=1
$TC_MAP7[1]=12
$TC_MAP10[1]=257

$TC_MPP1[1,1]=1
$TC_MPP2[1,1]=1
$TC_MPP3[1,1]=1
$TC_MPP4[1,1]=2
$TC_MPP5[1,1]=0
...
$TC_MPP1[1,12]=1
$TC_MPP2[1,12]=1
$TC_MPP3[1,12]=1
$TC_MPP4[1,12]=2
$TC_MPP5[1,12]=0

$TC_MAP1[9998]=7
$TC_MAP2[9998]=""
$TC_MAP3[9998]=16
$TC_MAP6[9998]=1
$TC_MAP7[9998]=1
$TC_MAP10[9998]=0

$TC_MPP1[9998,1]=2
$TC_MPP2[9998,1]=0
$TC_MPP3[9998,1]=0
$TC_MPP4[9998,1]=2
$TC_MPP5[9998,1]=1

$TC_MAP1[9999]=9
$TC_MAP2[9999]=""
$TC_MAP3[9999]=16
$TC_MAP6[9999]=1
$TC_MAP7[9999]=1
$TC_MAP10[9999]=0

$TC_MPP1[9999,1]=7
$TC_MPP2[9999,1]=0
$TC_MPP3[9999,1]=0
$TC_MPP4[9999,1]=2
$TC_MPP5[9999,1]=1


$TC_MDP1[1,1]=0

$TC_MDP2[1,1]=0

M17
```

Delete old data

TO area

Magazine 1

Location 1

to

Location 12

Buffer magazine

Spindle/holder

Load magazine

Loading point 1

Assignments Loading point to magazine
Spindle/holder to magazine

Variables in the configuration file (INI file)

Parameter	Meaning
$TC_MAMP1	Configuration magazine identifier
$TC_MAMP2	Tool search (bit 0-7) and location search (bit 8-15) - Bit 0=1: Active tool / min. duplo - Bit 8=1: First location forward
$TC_MAMP3	Procedure for tools in a wear group
$TC_MAP1	Magazine type (1=chain, 7=tool buffer magazine, 9=load magazine)
$TC_MAP2	Magazine identifier
$TC_MAP3	Magazine state - Bit 4=1: Magazine can be moved - Bit 6=1: Fixed-location coded
$TC_MAP6	Number of lines (box magazine only)
$TC_MAP7	Number of columns
$TC_MAP10	Magazine tool search strategies, see MAMP2
$TC_MPP1	Location category (1=magazine location, 2=Spindle /tool holder, 3=gripper, 7=loading point)
$TC_MPP2	Location type (0=compatible with each tool location type)
$TC_MPP3	Consider adjacent location (1=On, 0=Off)
$TC_MPP4	Location state - Bit 1=1: Free to receive a tool
$TC_MPP5	Location type index
$TC_MDP1	Distance from the magazine change position to the location of the first internal magazine (9999)
$TC_MDP2	Distance from the magazine change position to the location of the second internal magazine (9998)

Notes:

Load / Unload

Tool load/unload is done via the loading location defined in the machine configuration.

Each loading location has it own interface area in DB71.

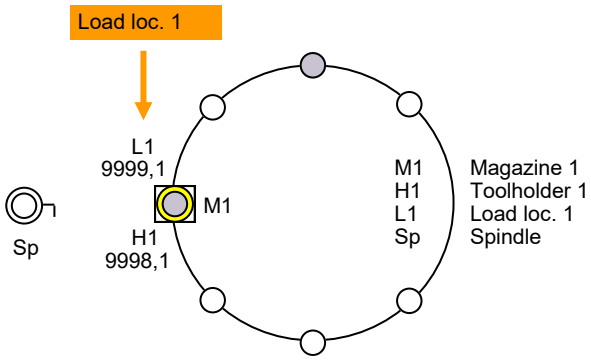
The interfaces are set up to handle the tool management commands: *Tool load*, *unload*, *Relocate* and *position magazine*.

The PLC functions FC6 (Multitool), FC8 and the Standard end acknowledgment with DB71 are used to communicate the current positions of tools.

Procedure:

- Assign loading locations.
- PLC program to handle the acknowledgments for the tool management commands.
  - Load magazine
  - Unload magazine
  - Interruption to the loading/unloading sequence
- Configure Magazine list (hide softkeys for Relocate tool an Position magazine).

Example: Turret with 12 locations



Load location 1 is permanently assigned to the turret magazine.

Assign loading locations

Load station Tool management

MD52274 $MCS_TM_TOOL_LOAD_STATION	Load station Tool management
Default: 0	Change to:

Description:

If only one loading station is configured: The setting = 0

= 0: No query window appears on the surface

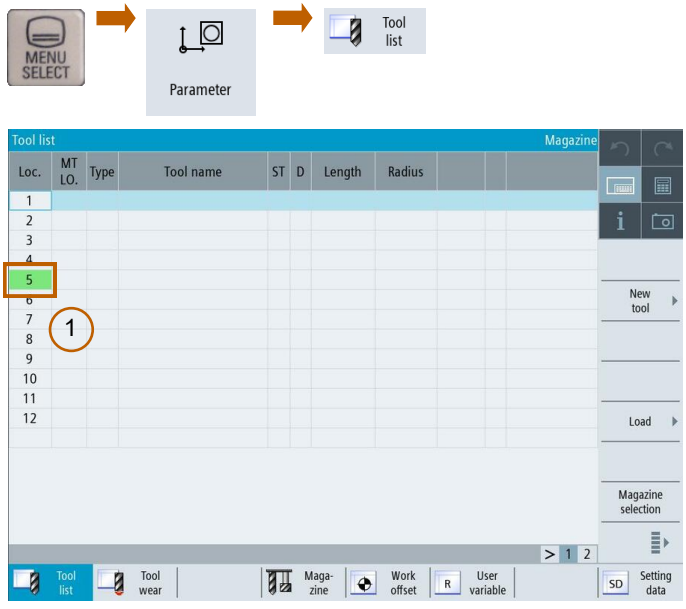
If more than 1 loading station is configured, you can select between

= 0: A query window appears on the surface to select the loading point.

= 1: fixed loading station 1, the window does not appear

= 2: fixed loading station 2, the window does not appear

Display on the controller



1. Magazine location on change position

Notes:

Acknowledgements for Load / Unload

PLC Programming example with acknowledgement using FC6

```
// Example "TMTurret" [FB160]

// Variables for "Load/Unload Turret"

// Input
//    loadIF1 : Bool;
//    unloadIF1 : Bool;
//    relocateIF1 : Bool;
//    posMagIF1 : Bool;

// Output
//    None

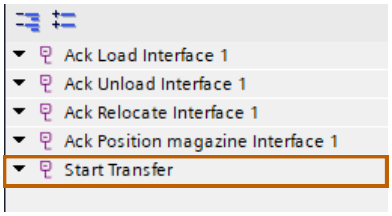
// InOut
//    None

// Static
//    statLoadIF1Flag : Bool;
//    statUnloadIF1Flag : Bool;
//    statRelocateIF1Flag : Bool;
//    statPosMagIF1Flag : Bool;
//    statTask : Byte;
//    statNo : Byte;
//    statMagNew : Int;
//    statLocNew : Int;
//    statMagOld : Int;
//    statLocOld : Int;
//    statStatus : Int;
//    statLocMt : Int;
//    statStart : Bool;

// Temp
//    tempLoadIF1EdgeP : Bool;
//    tempUnloadIF1EdgeP : Bool;
//    tempRelocateIF1EdgeP : Bool;
//    tempPosMagIF1EdgeP : Bool;
//    tempReady : Bool;
//    tempError : Int;

// Constant
//    None
```

PLC Programming example with acknowledgement using FC6



Start Transfer

```
REGION Start Transfer

    // Tool management status transfer
    "LBP_TMTransferMT"(Start := #statStart,
                       TaskIdent := #statTask,
                       TaskIdentNo := #statNo,
                       NewToolMag := #statMagNew,
                       NewToolLoc := #statLocNew,
                       OldToolMag := #statMagOld,
                       OldToolLoc := #statLocOld,
                       Status := #statStatus,
                       MultitoolLoc := #statLocMt,
                       Ready => #tempReady,
                       Error => #tempError);

    // Reset Job Start
    IF #tempReady OR ("mcpIn".reset AND (#tempError <> 0)) THEN
        #statStart := FALSE;
    END_IF;

END_REGION
```

Notes:

Job: Load magazine DB71

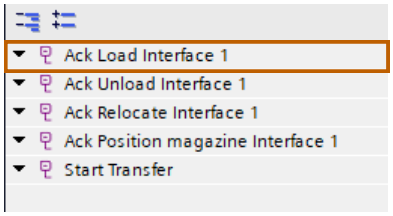
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Re-served	Re-served	Posi-tioning Multi-tool	Order from NC-Pro-gram	Posi-tioning Mag-azine	Relo-cate	Un-load	Load
DBB5	Data in DB1071	Reserved						Ack. sta-tus 3
DBW20	Identification for loading station (fixed value 9999)							
DBW22	No. of loading station							
DBW24	Magazine no. (source) for unloading/reloading/positioning							
DBW26	Location no. (source) for unloading/reloading/positioning							
DBW28	Magazine no. (target) for loading/reloading/positioning							
DBW30	Location no. (target) for loading/reloading/positioning							

Acknowledgement for load the magazine with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer will be started
TaskIdent	1	DB71 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB71.DBW28	Destination Mag. for load
NewToolLoc	DB71.DBW30	Destination location for load
OldToolMag	0	When loading =0
OldToolLoc	0	When loading =0
Status	1/10	1: Process finished 10: Process finished - no change to $TC_MAP8
MultitoolLoc	-1	When loading =-1

Example: Load tool to magazine location 9
DB71.DBW24 =0
DB71.DBW26 =0
DB71.DBW28 =1
DB71.DBW30 =9

PLC Programming example with acknowledgement using FC6



Load magazine

```
REGION Ack Load Interface 1

    // Scan rising edge
    #tempLoadIF1EdgeP := #loadIF1 AND NOT #statLoadIF1Flag;
    #statLoadIF1Flag := #loadIF1;

    // Set Job Start
    IF #tempLoadIF1EdgeP AND "LBP_TMLoad".IFNo[1] AND
        "LBP_TMLoad".TIF[1].Loading THEN
        #statTask := 1;                                  //DB71
        #statNo := 1;                                    //Interface 1
        #statMagNew := "LBP_TMLoad".TIF[1].TMag;
        #statLocNew := "LBP_TMLoad".TIF[1].TLoc;
        #statMagOld := 0;
        #statLocOld := 0;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Job: Unload magazine DB71

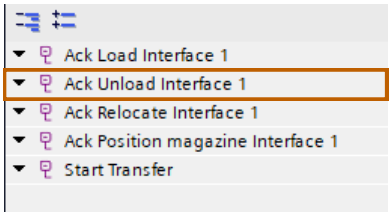
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Re-served	Re-served	Positioning Multi-tool	Order from NC-Program	Positioning Mag-azine	Relo-cate	Un-load	Load
DBB5	Data in DB1071	Reserved						Ack. sta-tus 3
DBW20	Identification for loading station (fixed value 9999)							
DBW22	No. of loading station							
DBW24	Magazine no. (source) for unloading/reloading/positioning							
DBW26	Location no. (source) for unloading/reloading/positioning							
DBW28	Magazine no. (target) for loading/reloading/positioning							
DBW30	Location no. (target) for loading/reloading/positioning							

Acknowledgement for unload the magazine with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer will be started
TaskIdent	1	DB71 Interface
TaskIdentNo	1	Interface 1
NewToolMag	0	When unloading =0
NewToolLoc	0	When unloading =0
OldToolMag	DB71.DBW20	Source Magazine for unload
OldToolLoc	DB71.DBW22	Source Location for unload
Status	1/10	1: Process finished 10: Process finished - no change to $TC_MAP8
MultitoolLoc	-1	When unloading =-1

Example: Unload tool from magazine location 9
DB71.DBW20 =9999
DB71.DBW22 =1
DB71.DBW24 =1
DB71.DBW26 =9
DB71.DBW28 =0
DB71.DBW30 =0

PLC Programming example with acknowledgement using FC6



Unload magazine

```
REGION Ack Unload Interface 1

    // Scan rising edge
    #tempUnloadIF1EdgeP := #unloadIF1 AND NOT #statUnloadIF1Flag;
    #statUnloadIF1Flag := #unloadIF1;

    // Set Job Start
    IF #tempUnloadIF1EdgeP AND "LBP_TMLoad".IFNo[1] AND
        "LBP_TMLoad".TIF[1].Unloading THEN
        #statTask := 1;                                  //DB71
        #statNo := 1;                                    //Interface 1
        #statMagNew := 0;
        #statLocNew := 0;
        #statMagOld := "LBP_TMLoad".TIF[1].LoadIdent;
        #statLocOld := "LBP_TMLoad".TIF[1].LoadNo;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Job: Relocate tool DB71

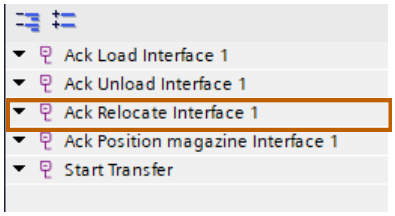
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Re-served	Re-served	Positioning Multi-tool	Order from NC-Program	Positioning Mag-azine	Relo-cate	Un-load	Load
DBB5	Data in DB1071	Reserved						Ack. status 3
DBW20	Identification for loading station (fixed value 9999)							
DBW22	No. of loading station							
DBW24	Magazine no. (source) for unloading/reloading/positioning							
DBW26	Location no. (source) for unloading/reloading/positioning							
DBW28	Magazine no. (target) for loading/reloading/positioning							
DBW30	Location no. (target) for loading/reloading/positioning							

Acknowledgement for Relocate tool with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer will be started
TaskIdent	1	DB71 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB71.DBW28	Destination mag. for relocate
NewToolLoc	DB71.DBW30	Destination loc. for relocate
OldToolMag	DB71.DBW24	Source mag. for relocate
OldToolLoc	DB71.DBW26	Source location for relocate
Status	1/10	1: Process finished 10: Process finished - no change to $TC_MAP8
MultitoolLoc	-1	When relocate =-1

Example: Relocate the tool from location 9 to 5
DB71.DBW24 =1
DB71.DBW26 =9
DB71.DBW28 =1
DB71.DBW30 =5

PLC Programming example with acknowledgement using FC6



Relocate tool

```
REGION Ack Relocate Interface 1

    // Scan rising edge
    #tempRelocateIF1EdgeP := #relocateIF1 AND NOT #statRelocateIF1Flag;
    #statRelocateIF1Flag := #relocateIF1;

    // Set Job Start
    IF #tempRelocateIF1EdgeP AND "LBP_TMLoad".IFNo[1] AND
        "LBP_TMLoad".TIF[1].Reloading THEN
        #statTask := 1;                                  //DB71
        #statNo := 1;                                    //Interface 1
        #statMagNew := "LBP_TMLoad".TIF[1].TMag;
        #statLocNew := "LBP_TMLoad".TIF[1].TLoc;
        #statMagOld := "LBP_TMLoad".TIF[1].SMag;
        #statLocOld := "LBP_TMLoad".TIF[1].SLoc;
        #statStatus := 1;                                //Status 1
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Job: Position magazine DB71

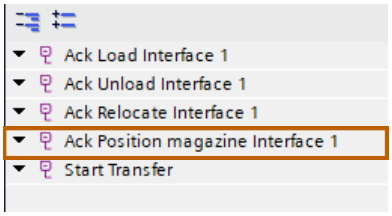
DB71	Loading / unloading points (NCK→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Re-served	Re-served	Positioning Multi-tool	Order from NC-Program	Positioning Mag-azine	Relo-cate	Un-load	Load
DBB5	Data in DB1071	Reserved						Ack. status 3
DBW20	Identification for loading station (fixed value 9999)							
DBW22	No. of loading station							
DBW24	Magazine no. (source) for unloading/reloading/positioning							
DBW26	Location no. (source) for unloading/reloading/positioning							
DBW28	Magazine no. (target) for loading/reloading/positioning							
DBW30	Location no. (target) for loading/reloading/positioning							

Acknowledgement: Position magazine with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer will be started
TaskIdent	1	DB71 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB71.DBW28	Destination mag. for position-ing
NewToolLoc	DB71.DBW30	Destination location for posi-tioning
OldToolMag	0	When positioning =0
OldToolLoc	0	When positioning =0
Status	1	1: Process finished
MultitoolLoc	-1	When positioning =-1

Example: Bring magazine pocket 5 to the load position
DB71.DBW24 =1
DB71.DBW26 =5
DB71.DBW28 =9999
DB71.DBW30 =1

PLC Programming example with acknowledgement using FC6



Position mag

```
REGION Ack Position magazine Interface 1

    // Scan rising edge
    #tempPosMagIF1EdgeP := #posMagIF1 AND NOT #statPosMagIF1Flag;
    #statPosMagIF1Flag := #posMagIF1;

    // Set Job Start
    IF #tempPosMagIF1EdgeP AND "LBP_TMLoad".IFNo[1] AND
        "LBP_TMLoad".TIF[1].Positioning THEN
        #statTask := 1;                                  //DB71
        #statNo := 1;                                    //Interface 1
        #statMagNew := "LBP_TMLoad".TIF[1].TMag;
        #statLocNew := "LBP_TMLoad".TIF[1].TLoc;
        #statMagOld := 0;
        #statLocOld := 0;
        #statStatus := 5;                                //Status 5
        #statLocMt := -1;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

Notes:

Sequence for Tool Change

The tool change is done at the change location, which is specified in the magazine configuration.

DB73 contains an interface area for each change location which has been configured.

Each interface is set up to handle these following Tool Management situations:

- Execute tool change (triggered via T-No)

In order to inform the Tool Management of tool movements the following functions are used in the machine PLC program FC6 (Multitool), FC7 (Turret) and FC8. Standard end acknowledgment via DB73 is also available.

Procedure:

- Set up the machine data for the tool change
- Generate the tool change program TCHANGE
- Write the PLC program for acknowledgment of the tool change commands
  - Tool change sequence
  - Interruption to the tool change sequence

Notes:

Set up the machine data for the tool change

Tool change triggered by T function

MD22550 = 0	The „tool change“ phase is activated by the T function for the new tool.

Tool change program, TCHANGE

MD10717 = TCHANGE	Name of the subprogram which is called when a T command is programmed.
MD15710 = TCHANGE	Name of the subprogram which is called when TCA command is programmed.

The TCHANGE program is called whenever a T command is programmed in the part program.

Programming the tool change

```
TEST_PROGRAM.MPF
N10
N20
N30 T="ROUGHING1"
N40
N50
N60
N70
N80
N90
N100 M30 Program end
```

Tool change program TCHANGE is called

Define tool holder

MD20124 = 1	Tool holder number

If a tool holder number is defined, then the number in $TC_MPP5 is interpreted as the tool holder number, and not as the spindle number. The automatic address extension for T is then also the tool holder number and no longer the spindle number.

The tool holder must be defined if the lathe has more than one spindle .

Activation of “T=Location number“

MD20310	Tool Management Mask
Bit 16 = 1	Programming T=Location number

If “T=Location number“ is active, then tools can be programmed via “T=Location number“.

Notes:

Generate the TCHANGE program

The TCHANGE, tool change program has to take the machine layout into consideration. The TCHANGE program is stored in the manufacturer cycles or the user cycles directory of the control .

Example of the TCHANGE program

```
PROC TCHANGE SAVE SBLOF DISPLOF
;T-Change Cycle (example)
;including TCA-substitution
;
;Machine data to set
;MD10717 $MN_T_NO_FCT_CYCLE_NAME="TCHANGE"
;MD15710 $MN_TCA_CYCLE_NAME="TCHANGE"
;
;------------------------------------------------------------
;--- Output T-Function ---
;------------------------------------------------------------
;
if(($C_T_PROG==1)or($C_TS_PROG==1)or($C_TCA==1))
  if $C_T_PROG==1 ;T=numerisch
    if $C_T==0
      if $C_TE>=1
        T[$C_TE]=0
      else
        T0
      endif
    else
      if $C_TE>=1
        T[$C_TE]=$C_T
      else
        T=$C_T
      endif
    endif
  endif
  ;
  if $C_TS_PROG==1 ;T=string
    if $C_TCA==1
      if $C_DUPLO_PROG==1
        if $C_THNO_PROG>=1
          TCA($C_TS,$C_DUPLO,$C_THNO)
        else
          TCA($C_TS,$C_DUPLO)
        endif
      else
        if $C_THNO_PROG>=1
          TCA($C_TS,,$C_THNO)
        else
          TCA($C_TS)
        endif
      endif
    else
      if $C_TE>=1
        T[$C_TE]=$C_TS
      else
        T=$C_TS
      endif
    endif
  endif
endif
;
;
;------------------------------------------------------------
;--- Continue T-Change (machine functions) ---
;------------------------------------------------------------
;
;
;
m17
```

T=Location number

TCA(“Identifier“)

T=“Identifier“

Variables associated with the TCHANGE program

Variable	Explanation
$C_T_PROG	Bool variable, indicates if the T was programmed with a number, eg T4
$C_TS_PROG	Bool variable, indicates if the T was programmed with a string, e.g. T=“ROUGH“
$C_TCA	Substitution for TCA instruction is active
$C_T	When T is programmed with the number. This returns the number, eg 4 in the case of T4
$C_TE	Address extension for the T instruction
$C_DUPLO_PROG	Bool variable, indicates whether or not the variable $C_DUPLO contains a duplo number
$C_THNO_PROG	Bool variable, indicates whether or not the variable $C_THNO contains a tool holder number
$C_TS	Tool name (String)
$C_DUPLO	Programmed duplo number
$C_THNO	Programmed tool holder or spindle number

Notes:

Tool change with turret

Programming

```
TEST_PROGRAM.MPF
N10
N20
N30
N40
N50 T="ROUGHING1"
N60
N70
N80
N90
N100
N110
N120
N130
N140
N150
N160 M30 Program end
```

Tool change program TCHANGE is called

T="Identifier"
-> **T="ROUGHING1"**

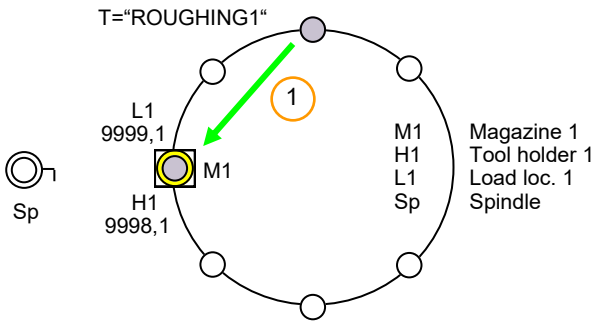
T=Location number
-> **T3 or T=3**

TCA ("Tool identifier", Duplo no., Tool holder no.)
-> **TCA("ROUGHING1", 1, 1)**

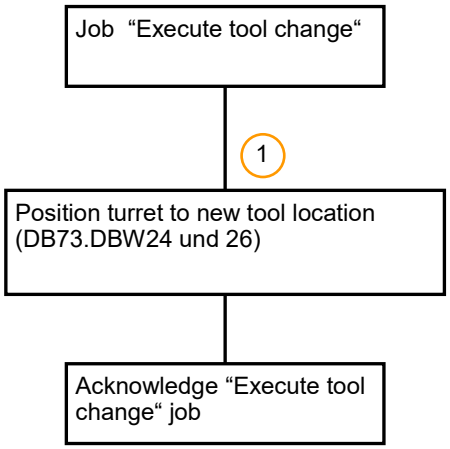
The TCA command forces the tool change to take place, regardless of the status of the tool. It is used ,for example with the "Measure Tool " feature.

Tool change -> Job from PLC (DB73.DBX4.1=1).

Example: Tool change with turret



Sequence: Tool change with turret



Notes:

Acknowledgement for Tool change with T No.

PLC Programming example with acknowledgement using FC6

```
// Example "TMTurret" [FB160] - continued

// Varibles for "Ack Toolchange with T-No"

// Input
//    tcPerform : Bool;

// Output
//    None

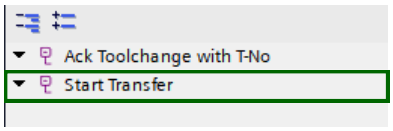
// InOut
//    None

// Static
//    statTcPerformFlag : Bool;
//    statTask : Byte;
//    statNo : Byte;
//    statMagNew : Int;
//    statLocNew : Int;
//    statMagOld : Int;
//    statLocOld : Int;
//    statStatus : Int;
//    statLocMt : Int;
//    statStart : Bool;
```

```
// Temp
//    tempTcPerformEdgeP : Bool;
//    tempReady : Bool;
//    tempError : Int;

// Constant
//    None
```

PLC Programming example with acknowledgement using FC6



Start Transfer

```
REGION Start Transfer

    // Tool management status transfer
    "LBP_TMTransferMT"(Start := #statStart,
                       TaskIdent := #statTask,
                       TaskIdentNo := #statNo,
                       NewToolMag := #statMagNew,
                       NewToolLoc := #statLocNew,
                       OldToolMag := #statMagOld,
                       OldToolLoc := #statLocOld,
                       Status := #statStatus,
                       MultitoolLoc := #statLocMt,
                       Ready => #tempReady,
                       Error => #tempError);

    // Reset Job Start
    IF #tempReady OR ("mcpIn".reset AND (#tempError <> 0)) THEN
        #statStart := FALSE;
    END_IF;

END_REGION
```

Notes:

Job: Execute tool change DB73

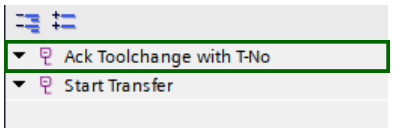
DB73	Signals from turret (NCK -> PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Interface (I) active							
	I8	I7	I6	I5	I4	I3	I2	I1
DBB1	Interface (I) active							
	I16	I15	I14	I13	I12	I11	I10	I9
DBB4	Re-served	Re-move man-ual tool	Re-served	Re-served	T0	Re-served	Per-form chan ge T no	Ob-ligato ry chan ge
DBB5	Data in DB10 73	Reserved						Ack Sta-tus 3
DBW20	Buffer identification (fixed value 9998)							
DBW22	Location in buffer							
DBW24	Magazine no. (source) for new tool							
DBW26	Location no. (source) for new tool							
DBW28	Magazine no. (target) for old tool							
DBW30	Location no. (target) for old tool							

Acknowledgement for “Execute Change“ with FC6

FC6	Parameter	Explanation
Start	TRUE	Transfer is started
TaskIdent	3	DB73 Interface
TaskIdentNo	1	Interface 1
NewToolMag	DB73.DBW20	Source magazine of new tool
NewToolLoc	DB73.DBW22	Source location of new tool
OldToolMag	DB73.DBW28	Destination magazine for old tool
OldToolLoc	DB73.DBW30	Destination location for old tool
Status	1	Process finished
MultitoolLoc	-1	Multitool position

Example: Change to turret position 3
DB73.DBW20 =9998
DB73.DBW22 =1
DB73.DBW24 =1
DB73.DBW26 =3
DB73.DBW28 =0
DB73.DBW30 =0

PLC Programming example with acknowledgement using FC6



Change with T No.

```
REGION Ack Toolchange with T-No

    // Scan rising edge
    #tempTcPerformEdgeP := #tcPerform AND NOT #statTcPerformFlag;
    #statTcPerformFlag := #tcPerform;

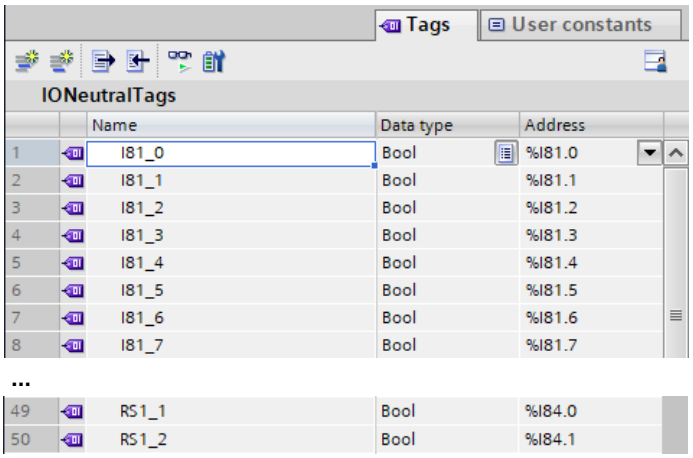
    // Set Job Start
    IF #tempTcPerformEdgeP AND "LBP_TMTurret".IFNo[1] AND
        "LBP_TMTurret".TIF[1].Perform THEN
        #statTask := 3;                                  //DB73
        #statNo := 1;                                    //Interface 1
        // T0?
        IF "LBP_TMTurret".TIF[1].SMag = 0 THEN
            #statMagNew := 0;
            #statLocNew := 0;
        ELSE
            #statMagNew := "LBP_TMTurret".TIF[1].BufferIdent;
            #statLocNew := "LBP_TMTurret".TIF[1].BufferNo;
        END_IF;
        #statMagOld := "LBP_TMTurret".TIF[1].TMag;
        #statLocOld := "LBP_TMTurret".TIF[1].TLoc;
        #statStatus := 1;                                //Status 1
        // Multitool?
        IF "LBP_TMTurret".TIF[1].DataInDB1073 THEN
            #statLocMt := "LBP_TMTurretMT".MTIF[1].NewPlace;
        ELSE
            #statLocMt := -1;
        END_IF;
        #statStart := TRUE;                              //Start FC6
    END_IF;

END_REGION
```

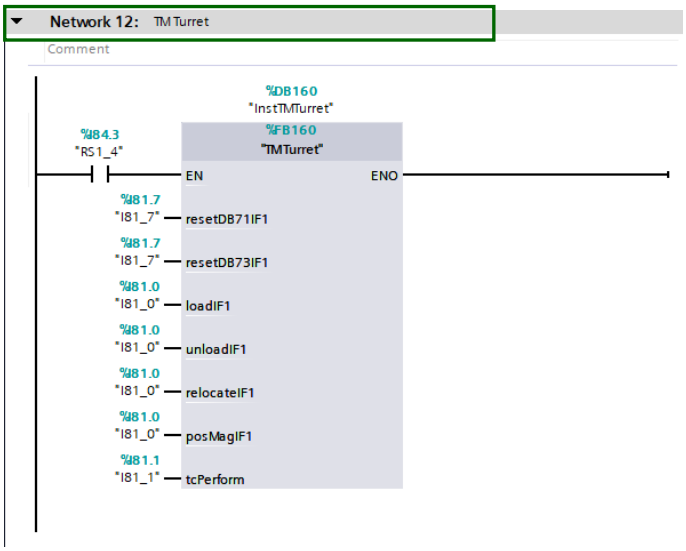
Notes:

Calling Turret functions in OB1

Create a tag table for the turret functions.



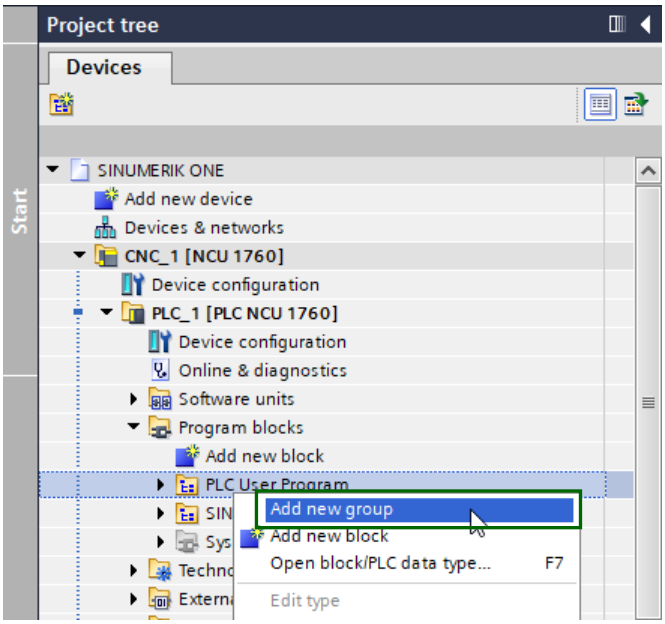
The newly created turret functions are called in OB1.



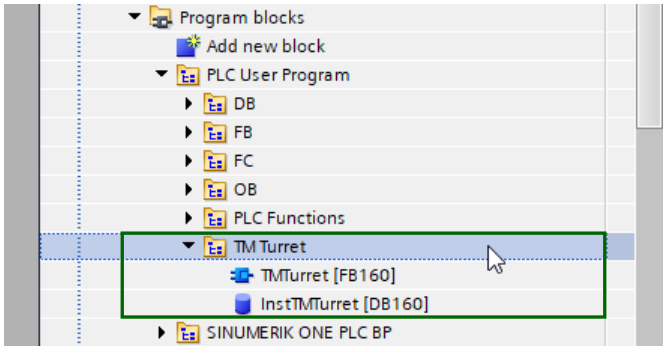
Notes:

Grouping Turret functions

Create a new group in the project navigation.



Insert Turret functions into the new group.



A119: END

Notes:

Notes:

A121

SINUMERIK ONE

Setup Milling

Module Description:

In this module a simple milling machine with the axes X, Y and Z, the rotary axis A and the main spindle SP1 is to be setup for milling. The individual commissioning steps are to be checked from time to time to show the result of the parameterization.
The preparation of the tool list, the cycle selections, the block search and various basic settings are also part of the training scope. There will be a detailed description of the selections to be made for JOG-TSM.

Module objective:

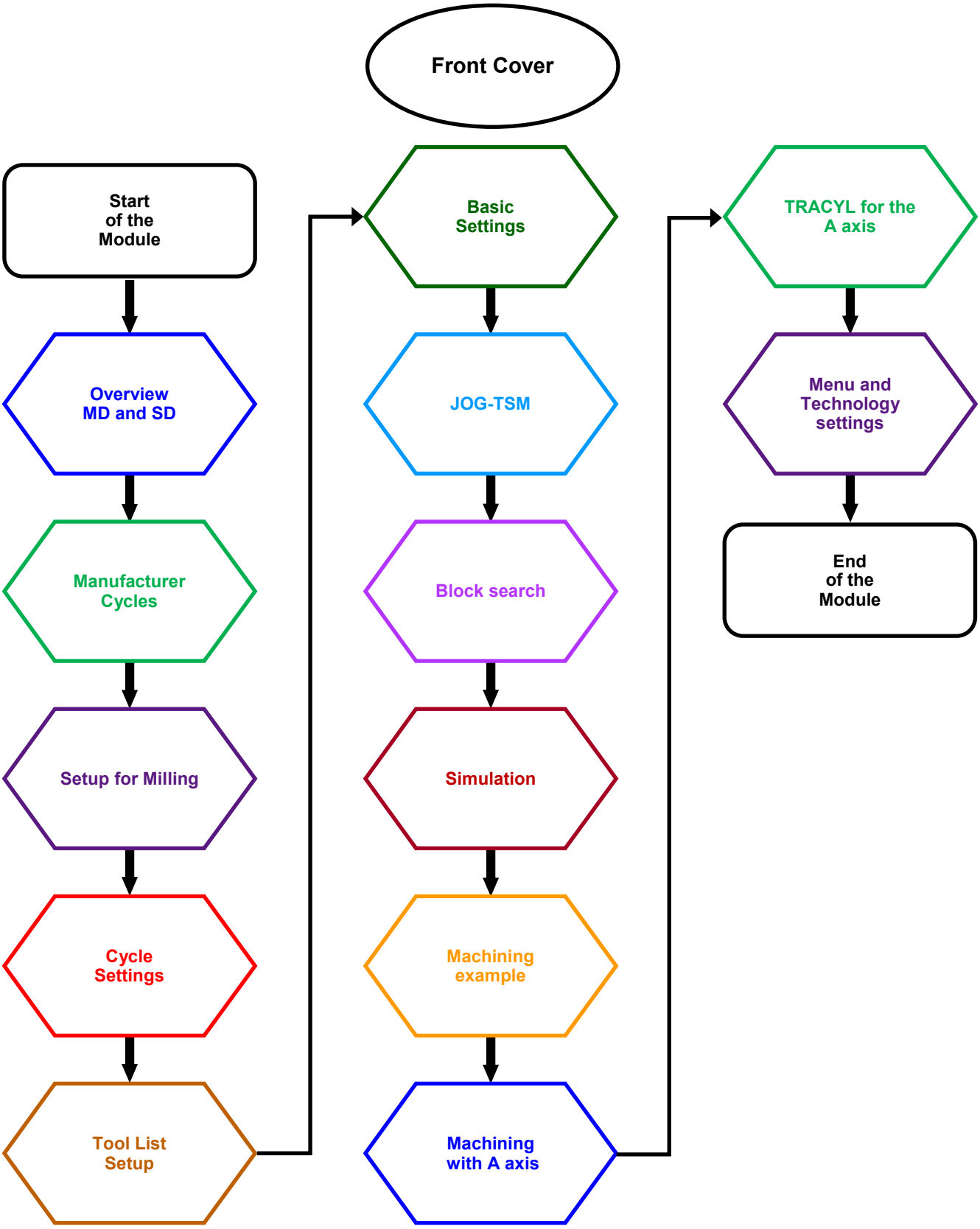
By means of this module you will learn and get to know the meaning of the Machine and Setting data to be used as well as the various means available in order to configure the machine to suit the particular requirements.
The functions will be parameterized in several blocks and subsequently tested to check correct working.

Content:

Sinumerik

Overview MD and SD
Manufacturer Cycles
Setup for Milling
Cycle settings
Tool List Setup
Basic Settings
JOG-TSM
Block Search
Simulation
Machining example
Machining with A axis
TRACYL for the A axis
Menu and Technology settings

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Overview MD and SD
Manufacturer Cycles
Setup for Milling
Cycle Settings
Tool List Setup
Basic Settings
JOG-TSM
Block search
Simulation
Machining example
Machining with A axis
TRACYL for the A axis
Menu and Technology settings
End of the Module

[Overview Machine data and setting data]

Basic settings Machine data/setting data

MD	Identifier	Settings	Comment	Description
10260	$MN_CONVERT_SCALING_SYSTEM	1	Exact	Enable basic system conversion
10602	$MN_FRAME_GEOAX_CHANGE_MODE	1	Exact	Frames when changing geometry axes
10714	$MN_M_NO_FCT_EOP	32	Variable	M function for spindle active after reset
10715	$MN_M_NO_FCT_CYCLE[0]	6	Variable	M function to be replaced by a subroutine
10716	$MN_M_NO_FCT_CYCLE_NAME[0]	L6	Variable	Subroutine name for M function replacement
10722	$MN_AXCHANGE_MASK	Bit 2 = 1	Exact	Parameters for axis replacement behavior
11410	$MN_SUPPRESS_ALARM_MASK	Bit 0 = 1 Bit 1 = 1 Bit 15 =1 Bit 20 =1	Variable	Bit 0: 15110: REORG not possible Bit 1: 10763 The path component of the block in the contour plane is zero Bit 15: 5000 Communication order cannot be executed Bit 20: 2900 Reboot is delayed
11450	$MN_SEARCH_RUN_MODE	Bit 0 = 1 Bit 1 = 1 Bit 2 = 1 Bit 6 = 1	Exact	Bit 0: Machining is stopped with the loading of the last action block after search run Bit 1: Automatic ASUP start after output of action blocks Bit 2: The output of the auxiliary functions in the action blocks is suppressed Bit 6: Block search with simple, computing-time-optimized algorithms
17530	$MN_TOOL_DATA_CHANGE_COUNTER	1FH	Exact	Mark tool data change for HMI
18080	$MN_MM_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1	Exact	Bit 0: Memory for TM-specific data is provided Bit 1: Memory for monitoring data (WZMO) is provided Bit 3: Memory to consider adjacent location is provided
20110	$MC_RESET_MODE_MASK	Bit 0 = 1 Bit 6 = 1 Bit 14 =1	Exact	Bit 0 = 1 G codes acc. to $MC_GCODE_RESET_VALUE Bit 6 = 1 Current setting for active tool length offset is retained after reset/end-of-part-program Bit 14= 1 The current setting of the basic frame is retained
20128	$MC_COLLECT_TOOL_CHANGE	0	Exact	No tool change commands after search run
20130	$MC_CUTTING_EDGE_RESET_VALUE	1	Exact	Tool and cutting edge remain activated
20150	$MC_G_CODE_RESET_VALUE	[15] = 3 [21] = 2 [41] = 1 [51] = 1 [52] = 1	Variable Variable Exact Exact Variable	Group 16: Feedrate override 3 = CFIN Group 22: Tool offset type 2 = CUT2DF Group 42: Toolholder 1 = TCOABS (for swivel) Group 52: Frame rotation in relation to workpiece 1 = PAROTOF (2 = PAROT, for swivel) Group 53: Frame rotation in relation to tool 1 = TOROTOF (for swivel)
20152	$MC_G_CODE_RESET_MODE	[5] = 1 [7] = 1 [12] = 0 [21] = 0	Exact Exact Exact Variable	Group 6: Plane selection remains active after Reset Group 8: Zero point offset remains active after Reset Group 13: Work piece dimensioning inch/metric default after reset Group 22: Tool correction type default after reset
20202	$MC_WAB_MAXNUM_DUMMY_BLOCKS	10	Minimum	maximum number of blocks w/o traversing movement with SAR
20240	$MC_CUTCOM_MAXNUM_CHECK_BLOCKS	4	Minimum	Blocks for look-ahead contour calculation with TRC
20250	$MC_CUTCOM_MAXNUM_DUMMY_BLOCKS	5	Minimum	maximum number of blocks without traversing motion in TRC

Notes:

Basic settings Machine data/setting data

MD	Identifier	Settings	Comment	Description
20310	$MC_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 14 =1	Exact Exact Exact Exact	Bit 0: Tool management active Bit 1: Tool monitoring function active Bit 3: Consider adjacent location active Bit 14: automatic tool change during RESET and Start
20320	$MC_TOOL_TIME_MONITOR_MASK	Bit 0 = 1	Variable	Time monitoring for tool in tool holder 1
20734	$MC_EXTERN_FUNCTION_MASK	Bit 3 = 1	Variable	Errors in ISO scanner are not output, the block is transferred to the Siemens translator.
22550	$MC_TOOL_CHANGE_MODE	1	Exact	Tool change with M function
22560	$MC_TOOL_CHANGE_M_CODE	206	Variable	M function for tool change
24006	$MC_CHSFRAME_RESET_MASK	Bit 0 = 1 Bit 5 = 0	Exact	Active system frames after reset Bit 0 = 1 Actual value setting and scratching remains active Bit 5 = 0 System frame for cycles is deleted
24007	$MC_CHSFRAME_RESET_CLEAR_MASK	Bit 0 = 0 Bit 5 = 1	Exact	Deletion of system frames after reset Bit 0 = 0 Actual value setting and scratching not delete Bit 5 = 1 System frame for cycles delete
24008	$MC_CHSFRAME_POWERON_MASK	Bit 0 = 0 Bit 5 = 0	Exact	Reset channel system frames after power on Bit 0 = 0 Actual value setting and scratching not reset Bit 5 = 0 System frame for cycles not reset
24030	$MC_FRAME_ACS_SET	1	Exact	Adjustment of SZS coordinate system
27860	$MC_PROCESSTIMER_MODE	Bit 0 = 1 Bit 1 = 1 Bit 4 = 1 Bit 5 = 1 Bit 6 = 1 Bit 8 = 1	Variable	Bit 0: Measurement of total operating time is active for all part programs Bit 1: Measurement of current program runtime is active Bit 4: Measurement also with active dry run feed Bit 5: Measurement also with program test Bit 6: $AC_CYCLE_TIME is not deleted on start by ASUP and PROG_EVENT Bit 8: AC_CYCLE_TIME will be deleted with GOTOS
27880	$MC_PART_COUNTER	Bit 0 = 1 Bit 8 = 1 Bit 11 =1	Variable	Bit 0: Counter $AC_REQUIRED_PARTS is activated Bit 8: Counter $AC_ACTUAL_PARTS is active Bit 11: Increment $AC_ACTUAL_PARTS with GOTOS
28000	$MC_MM_REORG_LOG_FILE_MEM	75	Minimum	Memory space for REORG (DRAM)
28010	$MC_MM_NUM_REORG_LUD_MODULES	20	Minimum	Number of blocks for local user variables in REORG (DRAM)
28082	$MC_MM_SYSTEM_FRAME_MASK	Bit 0 = 1 Bit 5 = 1 Bit 6 = 1	Exact	Setting up system frames (SRAM) Bit 0 = 1 System frame for actual value and scratching Bit 5 = 1 System frame for cycles Bit 6 = 1 System frame for transformations
28083	$MC_MM_SYSTEM_DATAFRAME_MASK	Bit 0 = 1 Bit 5 = 0	Exact	Setting up data management system frames (SRAM) Bit 0 = 1 System frame for actual value and scratching Bit 5 = 0 System frame for cycles
28450	$MC_MM_TOOL_DATA_CHG_BUFF_SIZE	400	Exact	Buffer for tool data changes (DRAM)
35040	$MA_SPIND_ACTIV_AFTER_RESET	[AX4] = 2	Exact	Own spindle RESET
42440	$SC_FRAME_OFFSET_INCR_PROG	0	Exact	Zero offsets in frames
42442	$SC_TOOL_OFFSET_INCR_PROG	0	Exact	Tool length offsets
42528	$SC_CUTCOM_DECEL_LIMIT	1	Variable	Feed lowering on circles with tool radius compensation
42940	$SC_TOOL_LENGTH_CONST	0	Exact	Change of tool length components with change of active plane
42950	$SC_TOOL_LENGTH_TYPE	0	Exact	Assignment of tool length compensation independent of tool type

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
51024	$MNS_BLOCK_SEARCH_MODE_MASK_JS	Bit 0 = 1		Bit mask for available search modes (ShopMill/ShopTurn) Bit 0: Block search with calculation without approach Bit 1: Block search with calculation with approach Bit 3: Skip EXTCALL programs Bit 5: Block search with test run with approach Bit 6 Block search with test run without approach
51028	$MNS_BLOCK_SEARCH_MODE_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 4 = 1		Bit mask for available block search modes Bit 0: Block search with calculation without approach Bit 1: Block search with calculation with approach Bit 3: Skip EXTCALL programs Bit 4: Block search without calculation Bit 5: Block search with test run with approach Bit 6 Block search with test run without approach
51038	$MNS_SET_ACT_VALUE	1		Set actual value selection 0 = Set actual value is not offered. 1 = If a user frame (settable work offset e.g. G54) is active, it will be used. In G500 Set actual values is not offered (system frame is no longer used).
51039	$MNS_PROGRAM_CONTROL_MODE_MASK	Bit 0 = 1		Options for machine - program influence: Bit 0: Program test function available
51040	$MNS_SWITCH_TO_MACHINE_MASK	0H		Automatic operating area switchover to machine Bit 0: When selecting a program in the program manager, the operating area does not automatically switch over to Machine. Bit 1: When switching the type of operation via the MCP, the operating area is not automatically switched over to Machine. Bit 2: Do not switch automatically in Programs during program selection. Bit 3: No auto. Start of block search on selection / execution under Programs.
51226	$MNS_FUNCTION_MASK_SIM	Bit 22 = 1	Exact	Function mask Simulation Bit 0: No automatic start on simulation selection Bit 1: Deactivate simulation Bit 4: Ignore frame components in X and Y (compatibility) Bit 5: Turning tool can rotate in the tool spindle Bit 6: Enable handwheel as simulation override Bit 7: Interpretation of handwheel values as absolute values Bit 10: Hide zero point symbol Bit 22: Machine model for simulation from kinematic chain (also without collision avoidance)
51228	$MNS_FUNCTION_MASK_TECH	0H		Function mask Cross-technology Bit 0: G code programming without multi-channel data Bit 1: Enable print function of editor Bit 2: With external program selection, test program

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
52000	$MCS_DISP_COORDINATE_SYSTEM	0		0: Vertical milling machine
52005	$MCS_DISP_PLANE_MILL	0		0: Plane selection on the operator panel 17: always G17 18: always G18 19: always G19
52200	$MCS_TECHNOLOGY	2	Exact	Technology 0: no specific configuration 1: turning 2: milling
52201	$MCS_TECHNOLOGY_EXTENSION	0		Extended technology 0: no specific configuration 1: turning: extended technology Turning has not yet been implemented for Milling 2: milling
52206	$MCS_AXIS_USAGE	[3] = 1		Meaning of the axes 0 = no special meaning; 1 = tool spindle (driven tool); 2 = C axis of the tool spindle (driven tool); 3 = main spindle (turning); 4 = C axis of the main spindle (turning); 5 = counterspindle (turning); 6 = C axis of the counterspindle (turning); 7 = linear axis of the counterspindle (turning); tailstock (turning); 9 = steady rest (turning); 10 = B axis (turning), 12 = B axis of the counter spindle (turning), 13 = Traverse travel X of the counter spindle (turning)
52207	$MCS_AXIS_USAGE_ATTRIB	[4] Bit 0 = 1 Bit 6 = 1 Bit 7 = 1 Bit 8 = 1		Axis attributes Bit 0: Rotates around the 1st geometry axis (applies to rotary axes) Bit 1: Rotates around the 2nd geometry axis (applies to rotary axes) Bit 2: Rotates around the 3rd geometry axis (applies to rotary axes) Bit 3: Positive direction of rotation is counter-clockwise (applies to rotary axes) Bit 4: Displayed direction of rotation for M3 is counter-clockwise (applies to spindles) Bit 5: Direction of rotation M3 corresponds to rotary axis minus (applies to spindles) This bit must be set in the same way as a PLC bit. For ONE / 840D sl: PLC bit DBnn.DBX17.6, nn = 31 + machine axis index For 828D: PLC bit DB38xx.DBX2001.6, xx is machine axis index Bit 6: Show rotary axis as offset target for measuring Bit 7: Offer rotation axis in position pattern Bit 8: Offer rotation axis to accept blank (at milling machines) Bit 9: Spindle is not SPOS-capable Bit 10: Rotary axis rotates around 1st geometry axis (for position pattern only) Bit 11: Rotary axis rotates around 2nd geometry axis (for position pattern only) Bit 12: Rotary axis rotates around 3rd geometry axis (for position pattern only) Bit 13: Position of the axis for blank selection from active WO (for rotary axes) Bit 14: Enable spindle for teach-in in axis mode

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
52210	$MCS_FUNCTION_MASK_DISP	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 4 = 1		Function mask Display Bit 0: Measuring system for programs always in the base system Bit 1: Front view for turning in school coordinate system Bit 2: Hide “T,S,M” soft key in JOG area Bit 3: Generate automatic end-of-program in MDI (with the “Delete blocks” soft key) Bit 4: Show follow-on tool in T,F,S window Bit 5: Hide softkey “Actual machine values“ Bit 6: Hide tool radius/diameter in T,F,S window Bit 7: Hide tool length in T,F,S window Bit 8: Hide tool icon in T,F,S window Bit 9: Display selection ”Simple input” Bit 10: Do not offer channel in job lists Bit 11: For Work activate the display of the logical spindles Bit 12: Hide mold making view for G code Bit 13: Hide channel axes which are currently located in another channel Bit 14: Display overlay of DRF under machine Bit 15: Display overlay of tool ($AA_TOFF) under machine Bit 16: Select JOG axis via HMI Bit 17: Select increment via HMI Bit 18: Offer select “With set stop override to0%” with configured stop
52212	$MCS_FUNCTION_MASK_TECH	Bit 3 = 1	Exact	Function mask Cross-technology Bit 0: Enable swivel plane/align tool with tool carrier Bit 1: No optimized travel along software limit switches (ShopMill/ShopTurn) Bit 2: Special approach logic for step drill (ShopTurn) Bit 3: Call block search cycle for ShopMill/ShopTurn Bit 4: Approach logic through cycle (ShopTurn) Bit 5: Call block search cycle for SERUPRO Bit 6: Work offset ZV cannot be entered as an absolute value (ShopTurn) Bit 7: Detect the expiry of tool lives (ShopMill/ShopTurn) Bit 8: Manual machine (ShopMill/ShopTurn) Bit 9: Selection/deselection of work offset via softkey Bit 10: Enable swivel plane/align tool using kinematic transformation Bit 11: Switch off layer check for drilling and milling tools (ShopTurn) Bit 13: Approach tool change point Y in Machine (ShopTurn) Bit 14: With approach logic only check tool position with cutting edge (ShopTurn) Bit 15: Do not set $MC TOOL_CARRIER_RESET_VALUE after block search Bit 16: Do not automatically execute swivel plane after block search Bit 17: Do not automatically execute align turning tool after block search Bit 18: Enable Determine total load : empty axis + workpiece
52214	$MCS_FUNCTION_MASK_MILL	Bit 0 = 1 Bit 1 = 1 Bit 2 = 1		Function mask Milling Bit 0: Enable cylinder surface transformation (ShopMill) Bit 1: Offer fixed table for blank clamping (at milling machines) Bit 2: Special handling of slide mill/saw for polyhedron Bit 3: Enable inside/rear machining (ShopTurn) Bit 4: Enable spindle clamping (C axis) (ShopTurn) Bit 5: Enable spindle control of tool spindle via surface Bit 6: Enable spindle control of turning spindle via surface Bit 7: Offer fixed table for blank clamping (on boring mills) Bit 8: Enable damping braking (C axis) (ShopTurn)

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
52216	$MCS_FUNCTION_MASK_DRILL	0H		Function mask Drilling Bit 0: CYCLE84 Unhide technology fields Bit 1: CYCLE840 Unhide technology fields Bit 2: Thread tapping also without encoder (ShopMill/ ShopTurn) Bit 3: Constant cutting speed in rel. to diameter of centering (ShopMill/ShopTurn)
52224	$MCS_FUNCTION_MASK_TECHNOLOGY_0	0H		Technology function mask 0 Bit 0: Hide display of tools/tool functions Bit 1: Hide display of spindles/spindle functions Bit 2: Hide “Synchronous actions” softkey Bit 3: Hide “Retract” softkey Bit 4: Hide “Contour” softkey Bit 5: Hide “Diverse” softkey Bit 6: Hide “Simulation” softkey Bit 7: Hide “Simultaneous recording” softkey Bit 8: Hide “Mold making view” softkey
52229	$MCS_ENABLE_QUICK_M_CODES	0H		Enable fast M functions Bit 0: Coolant OFF Bit 1: Coolant 1 ON Bit 2: Coolant 2 ON Bit 3: Coolant 1 and 2 ON
52230	$MCS_M_CODE_ALL_COOLANTS_OFF	9		M-code for all coolants OFF
52231	$MCS_M_CODE_COOLANT_1_ON	8		M-code for coolant 1 ON
52232	$MCS_M_CODE_COOLANT_2_ON	7		M-code for coolant 2 ON
52233	$MCS_M_CODE_COOLANT_1_AND_2_ON	-1		M-code for coolant 1 + 2 ON
52240	$MCS_NAME_TOOL_CHANGE_PROG	M6		Tool change program for G code steps
52270	$MCS_TM_FUNCTION_MASK	0H		Function mask Tool management Bit 0: Do not permit Create tool on magazine location Bit 1: Load/unload disable when machine is not in reset Bit 2: Load/unload disable for emergency stop Bit 3: Load/unload or relocate tool to/from buffer disabled Bit 4: Load directly in spindle Bit 5: Use grinding configuration file Bit 6: Permit relocating to/from buffer despite disable (Bit3) Bit 7: Create tool using the T number Bit 8: Hide Relocate tool Bit 9: Hide Position magazine Bit 10: Reactivate tool using Position magazine Bit 11: Reactivate tool in all monitoring modes Bit 12: Fade out Reactivate tool
52271	$MCS_TM_MAG_PLACE_DISTANCE	70		Distance between individual magazine locations. Is used for graphical display of magazine and tools in tool management.
52274	$MCS_TM_LOAD_STATION	2		Number of load station = 0: If a query window is on the surface or if only one loading station is available there will be no pop-up window = 1: fixed loading station 1 = 2: fixed loading station 2

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
52281	$MCS_TOOL_MCODE_FUNC_ON	-1		ShopMill only: M-code for tool-specific function ON
52282	$MCS_TOOL_MCODE_FUNC_OFF	-1		M-code for tool-specific function OFF If both M commands of a function equal -1, the corresponding field will not be displayed
53230	$MAS_SIM_START_POSITION	X = 0 Y = 0 Z = 649		Axis position at start of simulation Only adjust with geometry axes
54215	$SNS_TM_FUNCTION_MASK_SET	Bit 0 = 1		Function mask Tool management Bit 0: Diameter display for rotating tools Bit 1: Default direction of rotation for all turning tools is M4 Bit 2: Create tool without name suggestion Bit 3: Input disable for tool name and tool type in loaded tools Bit 4: Input disable for loaded tools, if the channel is not in Reset Bit 5: Calculate tool wear inputs additively Bit 6: Numerical input of the tool ident. Bit 7: Fade out tool monitoring parameters Bit 8: Diameter display for transverse axis - Geometry Bit 9: Diameter display for transverse axis - Wear Bit 10: Enable loading/relocating of tools on buffer locations Bit 11: Creating of new tools in gripper locations is disabled Bit 12: Do not unload measuring tools if ”Unload all” function is executed Bit 13: Do not delete tool wear when inputting the geometry value Bit 14: Load/relocate tool. Search without pre-assignment of the magazine
55200	$SCS_MAX_INPUT_FEED_PER_REV	1		Feedrate input upper limit for mm/rev
55201	$SCS_MAX_INPUT_FEED_PER_TIME	10000		Feedrate input upper limit for mm/min
55202	$SCS_MAX_INPUT_FEED_PER_TOOTH	1		Feedrate input upper limit for mm/tooth
55212	$SCS_FUNCTION_MASK_TECH_SET	Bit 0 = 1 Bit 1 = 1 Bit 2 = 1		Function mask Cross-technology Bit 0: Tool preselection active Bit 1: Calculate thread depth from metric thread lead Bit 2: Refer to the table for thread diameter and depth Bit 3: Delete programs generated by contour cycles
55214	$SCS_FUNCTION_MASK_MILL_SET	Bit 0 = 1 Bit 2 = 1		Function mask Milling Bit 0: Default setting - milling cycles with synchronous operation Bit 2: Depth calculation in milling cycles without parameter SC = 0: Depth calculation in milling cycles is carried out between reference plane + safety clearance and depth = 1: Depth calculation in milling cycles is carried out without inclusion of the safety clearance Note: Bit 2 is active with the following milling cycles: CYCLE61, CYCLE71, CYCLE76, CYCLE77, CYCLE79, CYCLE899, LONGHOLE, SLOT1, SLOT2, POCKET3, POCKET4 Bit 3: Deactivation of alarm 61012 different scaling in the planet

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
55216	$SCS_FUNCTION_MASK_DRILL_SET	Bit 3 = 1 Bit 4 = 1		Function mask Drilling Bit 0: Reverse direction of spindle rotation during tapping Bit 1: Consider rotation of the tool plane (CYCLE86) Bit 2: Consider swivelled table kinematics (CYCLE86) Bit 3: Monitoring spindle MD 31050/31060 (CYCLE84) Bit 4: Monitoring spindle MD 31050/31060 (CYCLE840) Bit 5: reserved Bit 6: Adapt spindle position to the tool orientation (CYCLE86) Bit 7: Adapt direction of retraction path in the plane to active mirroring (CYCLE86)
55300	$SCS_EASY_SAFETY_CLEARANCE	1		Simple input: Safety clearance
55301	$SCS_EASY_DWELL_TIME	0.6		Simple input: dwell time
55305	$SCS_EASY_DRILL_DEEP_FD1	90		Simple input: percentage 1st federate deep - hole drilling
55306	$SCS_EASY_DRILL_DEEP_DF	90		Simple input: percentage infeed deep - hole drilling
55307	$SCS_EASY_DRILL_DEEP_V1	1.2		Simple input: min. depth infeed deep - hole drilling
55308	$SCS_EASY_DRILL_DEEP_V2	1.4		Simple input: absolute retraction value deep - hole drilling
55309	$SCS_EASY_THREAD_RETURN_DIST	2		Simple input: return distance thread tapping
55400	$SCS_MILL_ENGRAVE_POINT_RAD	0		Engraving cycle CYCLE60: Circ. path radius for generating the char. ”point”

Notes:

Setup Milling

Pre-requisite:

Sinumerik Operate is already installed, PLC commissioning has been carried out, the tool management set-up has been completed.

Licensing:

In order for the required function to be achieved on your machine, the following software options must be available:

- "ShopMill/ShopTurn"
  Order Nr. 6FC5800-0BP17-0YB0
- "Residual material detection and machining"
  Order Nr. 6FC5800-0BP13-0YB0
- "Simultaneous recording (real-time simulation)"
  Order Nr. 6FC5800-0BP22-0YB0
- "3D simulation 1 (finished part)"
  Order Nr. 6FC5800-0BP25-0YB0
- "TRANSMIT and peripheral surface transformation"
  Order No. 6FC5800-0BM27-0YB0

Configuration of the axes and spindle

Machine axis name

MD10000 $MN_AXCONF_MACHAX_ NAME_TAB	Machine axis name
	Change to:

MD10000[0] = MX1
MD10000[1] = MY1
MD10000[2] = MZ1
MD10000[3] = MSP1
MD10000[4] = MA1

Assignment of geometry axis to channel axis

MD20050 $MC_AXCONF_GEOAX_ ASSIGN_TAB	Assignment of geometry axis to channel axis
	Change to:

MD20050[0] = 1
MD20050[1] = 2
MD20050[2] = 3

Machine axis number valid in channel

MD20070 $MC_AXCONF_MACHAX_ USED	Machine axis number valid in channel
	Change to:

MD20070[0] = 1
MD20070[1] = 2
MD20070[2] = 3
MD20070[3] = 4
MD20070[4] = 5

Geometry axis name in channel

MD20060 $MC_AXCONF_GEOAX_ NAME_TAB	Geometry axis name in channel
	Change to:

MD20060[0] = X
MD20060[1] = Y
MD20060[2] = Z

Channel axis name in channel

MD20080 $MC_AXCONF_CHANAX_ NAME_TAB	Channel axis name in channel
	Change to:

MD20080[0] = X1
MD20080[1] = Y1
MD20080[2] = Z1
MD20080[3] = SP1
MD20080[4] = A

Notes:

Rotary axis / spindle

MD30300 $MA_IS_ROT_AX	Rotary axis / spindle
	Change to:

AX4: MSP1
MD30300 = 1 axis is a rotary axis

AX5: MA1
MD30300 = 1 axis is a rotary axis

AX4: MSP1
MD30310 = 1 Modulo-transformation for rotary axis

AX5: MA1
MD30310 = 1 Modulo-transformation for rotary axis

Modulo 360 degrees displayed for rotary axis or spindle

MD30320 $MA_DISPLAY_IS_MODULO	Modulo 360 degrees displayed for rotary axis or spindle
	Change to:

AX4: MSP1
MD30320 = 1 Actual value display Modulo

AX5: MA1
MD30320 = 1 Actual value display Modulo

Modulo conversion for rotary axis / spindle

MD30310 $MA_ROT_IS_MODULO	Modulo conversion for rotary axis / spindle
	Change to:

Assignment of spindle to machine axis

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX	Assignment of spindle to machine axis
	Change to:

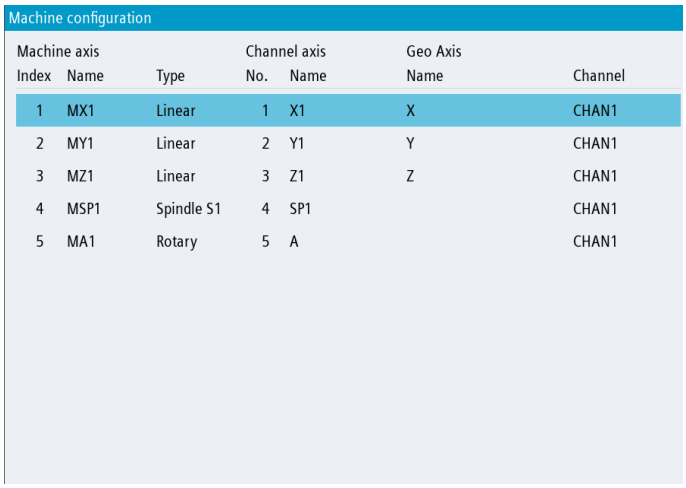
Description:

Definition of the spindle. The spindle is defined if the spindle number is entered into this MD.

AX4: MSP1
MD35000 = 1 Assignment of spindle to the machine axis (spindle S1)

Check of the set axes and spindles:





Notes:

Manufacturer cycles

For the individual matching the following cycles are included in the standard cycle package:

CUST_TECHCYC.SPF	Manufacturer‘s cycle for matching to the technological cycles.
CUST_800.SPF	Manufacturer‘s cycle for matching to the functions swivelling plane and orienting tool (CYCLE800).
CUST_832.SPF	Manufacturer‘s cycle for matching to the function High Speed Settings (CYCLE832).
CUST_CLAMP.SPF	Manufacturer’s cycle for matching to the function multiple workpiece clamping
PROG_EVENT.SPF	Standard cycle for the support of the functionality of: • Block search when milling or turning is activated • Block search and swivelling plane • Block search and tool indexing

Note:

The cycle PROG_EVENT.SPF is a standard Siemens cycle and should not be changed by the manufacturer. For the manufacturer cycles CYCPE1MA.SPF or CYCPE_MA.SPF, the corresponding jump-out markers are prepared at the beginning and at the end of PROG_EVENT.SPF.

At the beginning of the cycle PROG_EVENT
- CYCPE1US in the directory CUS
- CYCPE1MA in the directory CMA

At the end of the cycle PROG_EVENT
- CYCPE_US.SPF in the directory CUS
- CYCPE_MA.SPF in the directory CMA

For these cycles templates are not available. If the CYCPE1MA.SPF and CYCPE_MA.SPF manufacturer cycles are loaded in the NC, the program branches from PROG_EVENT to the manufacturer cycles as appropriate.

Copying the Manufacturer cycles



1. Select the “Setup” mode.



2. Press the Softkey "System data".
3. Open the directory NC-data / Cycles / Standard cycles and select the required cycle.



4. Press the Softkey "Copy" and open the directory NC-data / Cycles / Manufacturer's cycles.

5. Press the Softkey "Paste".
   Now this cycle is available for individual matching

Note:

The cycles intended for the machine matching commencing with CUST_.*.SPF can be copied into the manufacturer or user directory when using the access level "Manufacturer". The cycle "PROG_EVENT.SPF is a standard cycle that may neither be copied nor altered. It is listed here only to indicate the scope for manufacturer-specific matching.

Setting-up for milling

Matching of the coordinate systems to the user interface

Coordinate system position

MD52000 $MCS_DISP_COORDINATE_SYSTEM	Coordinate system position
Default: 0	Change to:

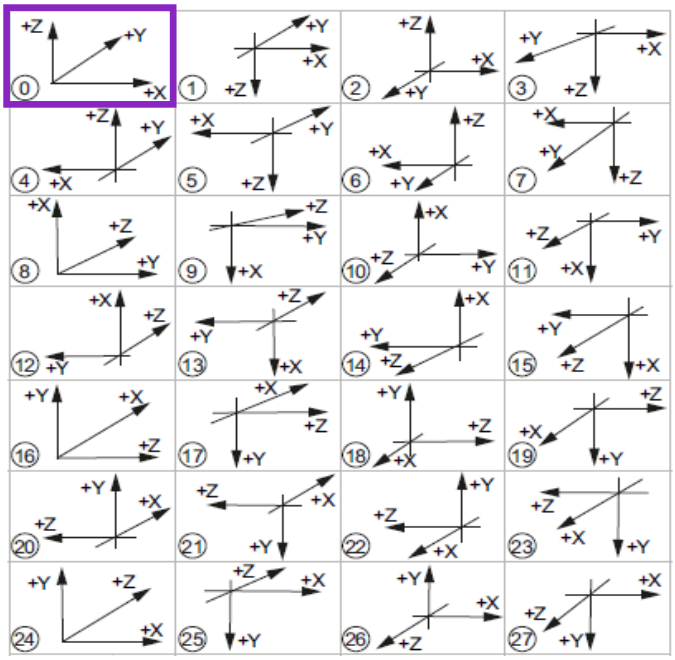
Description:

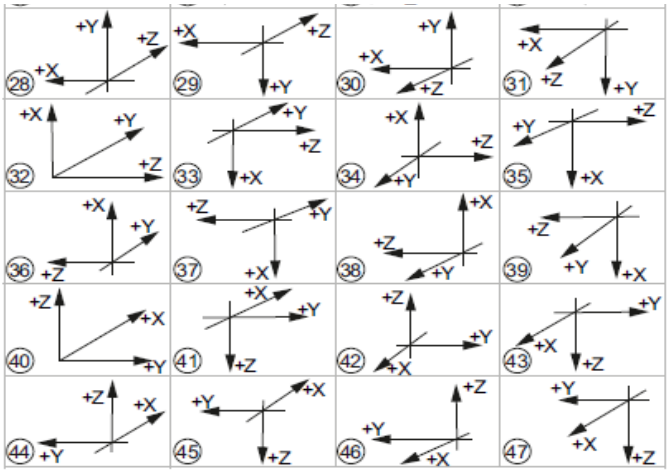
This MD is used to match the coordinate system of the user interface to the coordinate system of the machine. In the user interface all help images, the animated graphics, the simulation as well as the input fields with circular direction indication.
The coordinate system can take on any of the shown positions.

The typical selection for vertical milling machine is 0 = Presetting

Notes:

Selection table of the possible coordinate systems





Plane selection Milling

MD52005 $MCS_DISP_PLANE_MILL	Plane selection Milling
Default: 0	Change to:

Description:

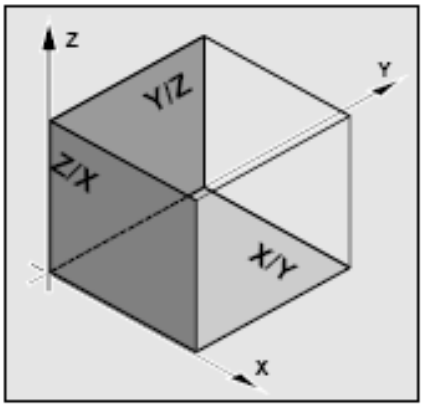
0: Plane selection in the user interface
This setting makes sense only if the machining plane on the machine can be switched, e.g. by the use of angular milling heads.

17: always G17
18: always G18
19: always G19

Note:

Typical vertical function = G17

The machining planes are specified as shown:

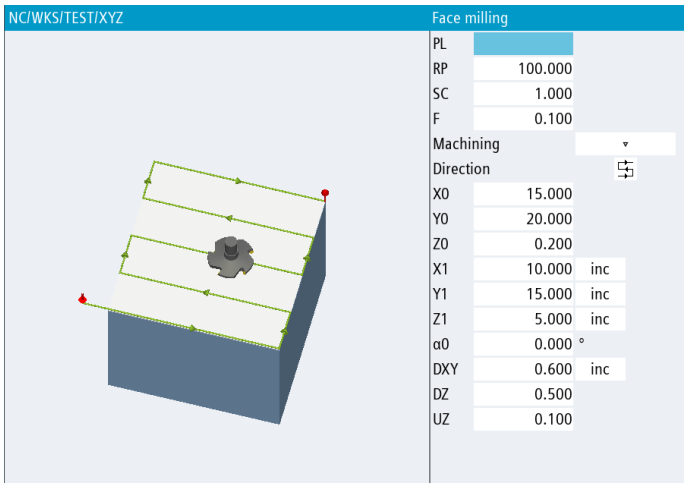


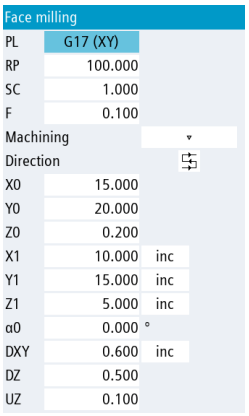
Plane	Tool axis
X/Y	Z G17
Z/X	Y G18
Y/Z	X G19

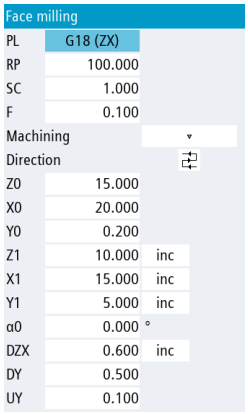
Notes:

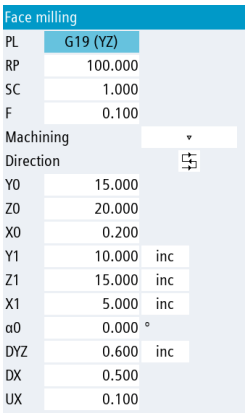
Continuation MD52005

Example planar milling:











Plane selection by means of the toggle key:
G17 (XY), G18 (ZX), G19 (YZ) or no plane

Technology

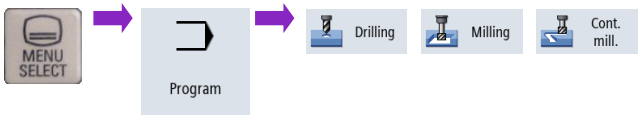
MD52200 $MCS_TECHNOLOGY	Technology
	Change to: 2

Description:

0: No specific configuration
1: Turning
2: Milling

Please also note MD 52201
$MCS_TECHNOLOGY_EXTENSION.

Selection of 2 specifies milling as the main technology. This means, the Softkeys for drilling and milling will be shown along the lower edge of the screen.



Extended technology

MD52201 $MCS_TECHNOLOGY_EXTENSION	Extended technology
Default: 0	Change to:

Meaning of the axes in the channel

MD52206[x] $MCS_AXIS_USAGE	Meaning of the axes in the channel
Default: 0	Change to:

Description:

No matching specification of the machine datum is required for the sample machine being described here.

Note:

If high-speed extension spindles are used, a 2 must be entered into the respective machine datum of the extension spindle. An extension spindle is not included in the axis configuration used here.

Notes:

Axis attributes

MD52207[x] $MCS_AXIS_USAGE_ATTRIB	Axis attributes
Default: 0H	Change to:

Description:

Bit 0: Rotates around the 1st geometry axis (applies to rotary axes)

Bit 1: Rotates around the 2nd geometry axis (applies to rotary axes)

Bit 2: Rotates around the 3rd geometry axis (applies to rotary axes)

Bit 6: Show rotary axis as offset target for measuring

Bit 7: Offer rotation axis in position pattern

Note:

The Bits 3, 4 and 5 have no meaning in conjunction with ShopMill.

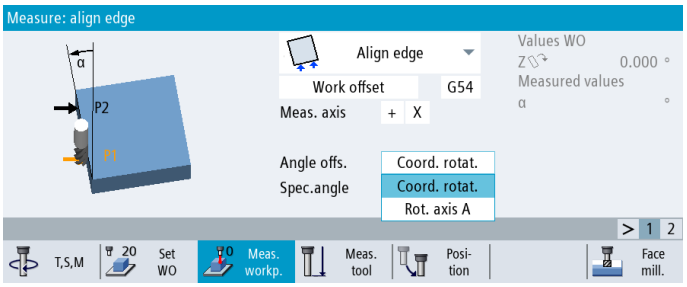
Description:

In case of machines with rotary axes it is possible - in conjunction with the function:
"work piece measuring - edge alignment" - to determine the angular correction

For this the following settings are necessary for the rotary axes:

MD52207[4] A axis
Bit 0 = 1: rotates around the 1st geometry axis
Bit 6 = 1: Show rotary axis as correction destination for measuring

Example: Measuring zero-point work piece - align edge



Number of actual values with large font

MD52010 $MCS_DISP_NUM_AXIS_BIG_FONT	Number of actual values with large font
Default: 3	Change to:

Description:

Number of actual values with large font, Max. value 31.



Only display spindles in actual values window when in axis mode

MD51023 $MNS_ACT_VALUE_SPIND_MODE	Only display spindles in actual values window when in axis mode
Default: 1	Change to:

Description:

The display of the spindles in the actual value windows can be influenced by means of this. If the datum is set to 1 the spindles will be shown in the actual value window only while they are in the axis mode. The axes are displayed below the geometry and auxiliary axes. Axes that are not in the axis mode are shown as gaps.
If set to 0 the spindles will be displayed at all times.

Notes:

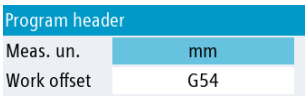
Function mask display

MD52210 $MCS_FUNCTION_MASK_ DISP	Function mask display
Default: Bit 0 = 1	Change to:

Description:

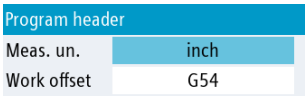
Bit 0: Meas. sys. for programs always in the base system
Bit 1: Face view when turning in the school coordinate system
Bit 2: Hide "T,S,M" softkey in Jog area
Bit 3: Generate automatic end-of-program in MDI (with Softkey "Delete blocks")
Bit 4: Show follow-on tool in T,F,S window

Bit 0 = 0:
In a ShopMill program it is possible to select "mm dimensions" or "inch dimensions" in the program header.





Bit 0 = 1:
The selection field “Unit of meas.” is hidden. The measuring system for programs is always in the base system.



Bit 1: Takes no effect in the Milling technology.

Bit 2=0: The Softkey "T,S,M" is shown in the mode JOG . This setting is in line with the basic setting of ShopMill.



Bit 2=1: The Softkey "T,S,M" is hidden in the mode JOG.



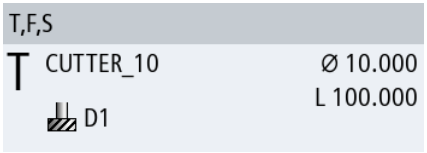
Bit 3=0: Under MDA the command End-of-program will not be generated automatically after "Block delete".



Bit 3=1: Under MDA the M-command entered in MD 10714 $M_NO_FCT_EOP will automatically be generated as end-of-program command after "block delete". In this example configuration it is M32.



Bit 4=0: The follow-up tool is not shown in the T;F;S-window



Bit 4 = 1 The follow-up tool is shown in the T;F;S-window. This setting makes sense only in conjunction with tool prepare on a machine featuring a double gripper.



Note: Also see the setting of SD 55210 $SCS_FUNCTION_MASK_TECH_SET, Bit 0 "Tool pre-selection activated"

Notes:

Function mask Cross-technology

MD52212 $MCS_FUNCTION_MASK_ TECH	Function mask Cross-technology
	Change to: Bit 3 = 1 Bit 5 = 0

Description:

Bit 0: Enable swivel plane/align tool with tool carrier
Bit 1: Optimized travel along software limit switches (ShopMill/ShopTurn)
Bit 2: Special approach logic for step drill (ShopTurn)
Bit 3: Call block search cycle for ShopMill/ShopTurn
Bit 4: Approach logic through cycle (ShopTurn)
Bit 5: Call block search cycle for SERUPRO
Bit 6: Work offset ZV cannot be entered as an absolute value (ShopTurn)

Bit 0 and Bit 1 make sense only in conjunction with swivelling. A more detailed description will be found in Module "Setup Swivelling".

Bit 2: This Bit is not relevant in the technology Milling . In the ShopMill cycles the longest cutting edge is taken into account when retracting (retraction to the safety level).

Bit 3 = 0

The ShopMill cycle E_S_ASUP is not carried out after block search.

Bit 3 = 1

The ShopMill cycle E_S_ASUP will be carried out after block search. Details regarding this setting are contained in the section "Block search"

Bit 5 = 0
Block search cycle for SERUPRO will not be called up.



The setting = 0 is necessary for single-channel ShopMill machines.

Bit 5 = 1
The standard cycle CYCLE207 will be called up after Block search with SERUPRO.

The setting of Bit 3 is not effective in this case. The E_S_ASUP will not be called up.

This setting must be used only for multi-channel machines.

Cycle Settings-Drilling

MD52216 $MCS_FUNCTION_MASK_ DRILL	Function mask Drilling
Default: 0H	Change to:

Description:

Bit 0 = 0 Hide input fields Technology
1 Show input fields Technology

CYCLE84 (without compensating chuck)

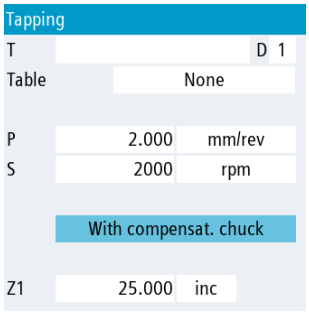
Bit 1 = 0 Hide input fields Technology
1 Show input fields Technology

CYCLE840 (with compensating chuck) valid only for ShopMill.

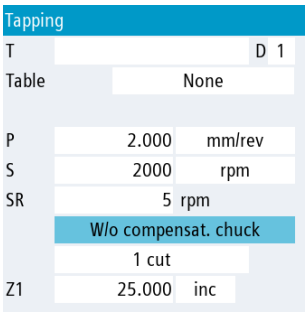
Notes:

Continued MD52216

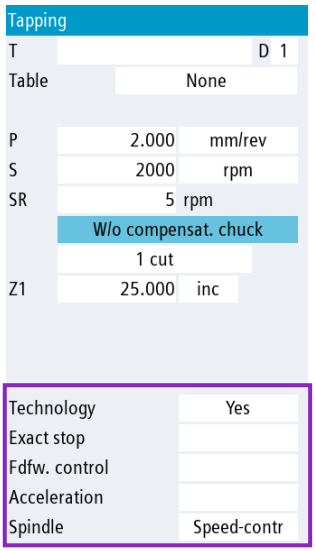
Bit 0 = 0
Technology fields will not be shown in either case:
"with compensating chuck"



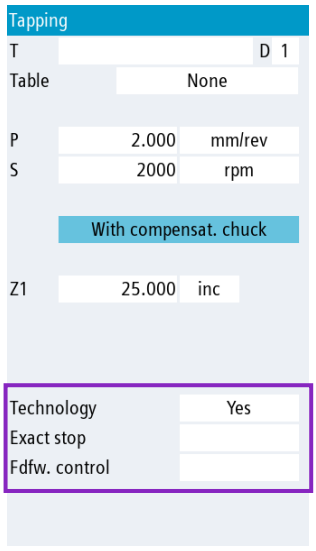
Bit 1 = 0
"without compensating chuck



MD52216 Bit 0 = 1
Technology fields will be shown with the selection "without compensating chuck (Cycle84)



MD52216 Bit 1 = 1
Technology fields will be shown with the selection "with compensating chuck (Cycle840)



Function mask Cross-technology

SD55212 $SCS_FUNCTION_MASK_ TECH_SET	Function mask Cross-technology
	Change to: Bit 0 = 1 Bit 1 = 1 Bit 2 = 1

Description:

Bit 0: Tool pre-selection activated
Bit 1: Evaluate thread depth from metric pitch
Bit 2: Transfer thread diameter and thread depth from table

Bit 0: Tool pre-selection activated
Directly after the exchange has been completed, the tool will be prepared.
The generation of the follow-up tool occurs only with JobShop cycles.

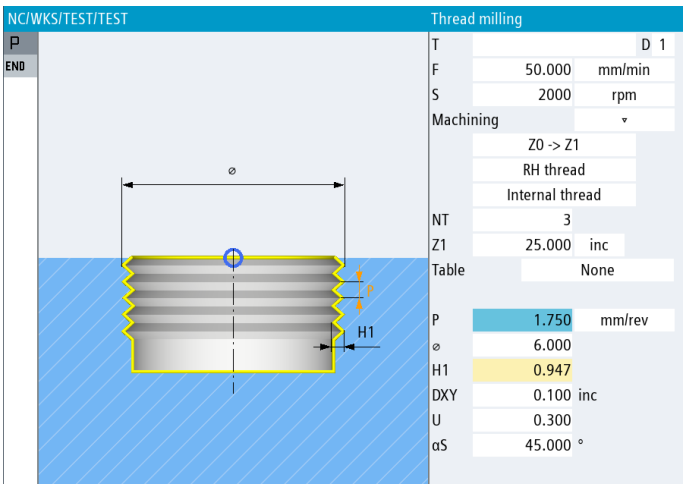
Bit 1: Automatic evaluation of the thread depth for metric threads

Bit 2: Transfer thread diameter and thread depth from table.

Bit 1 = 0 The thread depth will not be evaluated automatically

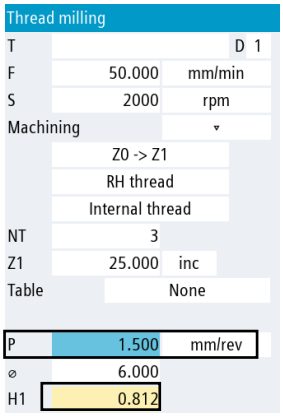
The value for the thread depth "H1" must be entered manually from a table.

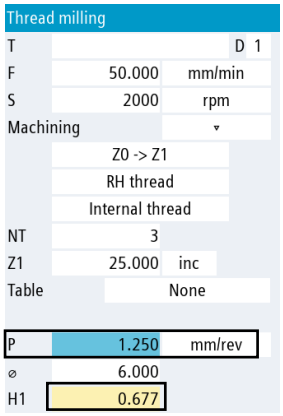
Bit 1 = 1 The thread depth will be evaluated from the metric thread pitch.



Notes:

SD55212 Continued

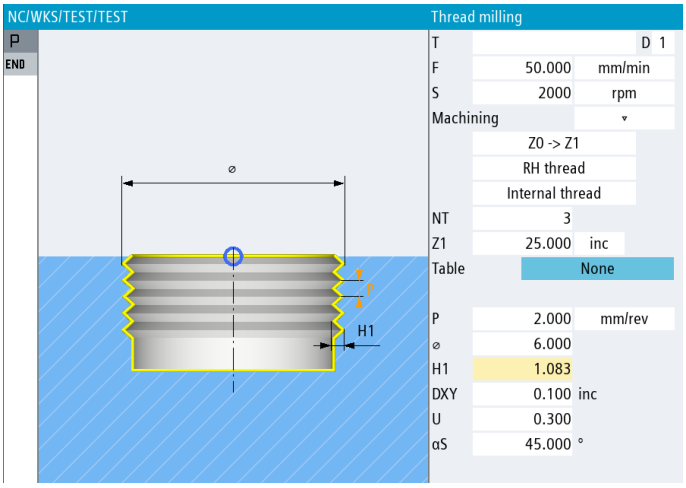


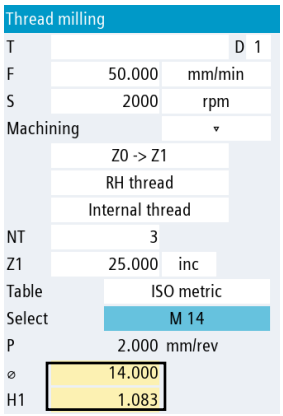


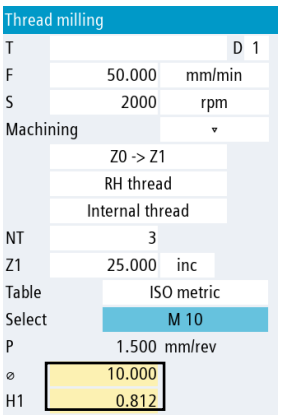
Only if in the table the column "without" is selected and for P the unit "mm/rev" will the thread depth "H1" in the yellow-backed fields be calculated automatically. These automatically obtained values can be overwritten by the operator. If the thread pitch is entered again, the thread depth will be calculated once more automatically.

Different factors are stored for the internal and external threads:
- Internal threads: 0,5413
- External threads: 0,6134

Bit 2 = 1 Thread diameter and depth from a table







As soon as the line "Table ISO metric" is selected, the table is activated for the various threads.

The yellow-backed fields for nominal diameters and thread depth H1 will be filled in with tabular values depending on the chosen thread.

Notes:

Cycle Settings-Milling

SD55214 $SCS_FUNCTION_MASK_ MILL_SET	Function mask Milling
Default: Bit 0 = 1 Bit 2 = 1	Change to:

Description:

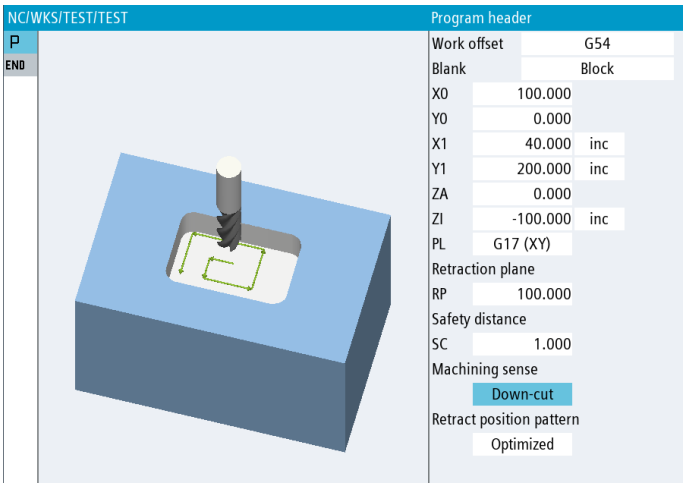
Bit 0: Default setting for milling with synchronous operation

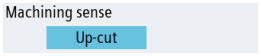
Bit 1: not assigned

Bit 2: Depth calculation in milling cycles without parameter SC

Bit 0 The machining sense (Up-cut/down-cut) can be fixed or selectable via the “toggle” key.

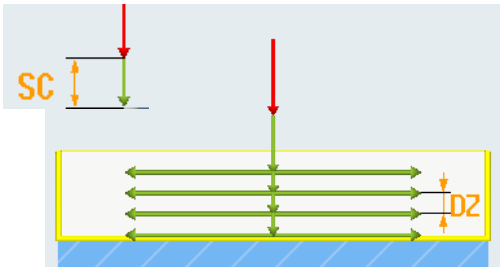
Bit 0 = 1





Bit 2 = 0
The subdivision of cuts in direction of infeed is calculated including the safety gap SC for the path from Z0+SC to Z1

Bit 2 = 1
The subdivision of cuts in direction of infeed is obtained without the safety gap SC for the path from Z0 to Z1



Note:
The safety gap is specified in the program header.

Function mask Drilling

SD55216 $SCS_FUNCTION_MASK_ DRILL_SET	Function mask Drilling
Default: Bit 3 = 1 Bit 4 = 1	Change to:

Description:

Bit 0: Reverse direction of spindle rotation during tapping

Bit 1: Boring CYCLE86: Taking into account the rotation of the work piece plane when positioning the spindle

Bit 2: Boring CYCLE86: Taking into account the indexed table kinematics (tool carrier) when positioning the spindle

Bit 3: Tapping CYCLE84: (without compensating chuck). Monitoring the machine data 31050[x] and 31060[x] of the spindle

Notes:

SD55216 Continued

Bit 4: Tapping CYCLE840: (with compensating chuck). Monitoring the machine data 31050[x] and 31060[x] of the spindle

Bit 5: Reserved

The denominator of the gearbox is contained in MD31050[x] while MD31060[x] contains the numerator. Together the two provide the gear ratio for the spindle. This is required for monitoring the feed obtained from the speed x the pitch.

Description:

Bit 0 = 1: Reverse direction of spindle rotation during tapping

Bit 1 = 1: Taking into account the rotation of the tool plane (CYCLE 86)
In the boring cycle,CYCLE86 the spindle is positioned on the parameter SPOS when reaching the boring depth. If SPOS=0 the tool cutting edge points in the positive direction of the 1st axis of the plane (G17 towards X+).

If the setting is varied by using Bit 1 = 1, a rotation about the tool axis (e.g. ROT Z180) will be taken into account for the spindle positioning.

Bit 2 = 1: Taking into account indexed table kinematics (CYCLE 86)
See description Bit 1
With the selection of Bit 2 = 1 an indexed kinematics will be taken into account when positioning the spindle.

Description:

Bit 3 = 1: Tapping CYCLE84: Monitoring the machine data 31050 $MA_DRIVE_AX_RATIO_DENOM (divider of gearbox ratio) and 31060 $MA_DRIVE_AX_RATIO_NUMERA (numerator of gearbox ratio) of the spindle for the boring axis

The monitoring is activated only if MD35590 $MA_PARAMSET_CHANGE_ENABLE = 2.
This setting permits a parameter block change by the PLC-interface or by the command SCPARA (servo parameter block programming).
If the monitoring is activated the alarm 61127 [Channel %1:] Block %2 (gear ratio of the tapping axis incorrectly defined) will be set:

Bit 4 = 1: Similar to Bit 3, however, for the cycle CYCLE840

Bit 5: Reserved

Cycle Settings

Upper limit feedrate/rev

SD55200 $SCS_MAX_INP_FEED_PER_REV	Upper limit feedrate/rev
Default: 1 mm/U	Change to:

Description:

Upper limit for the selection of the federate/rev

Upper limit feedrate/min

SD55201 $SCS_MAX_INP_FEED_PER_TIME	Upper limit feedrate/min
Default: 10000 mm/min	Change to:

Description:

Upper limit for the input of the feedrate in mm/min

Notes:

Upper limit feedrate/tooth

SD55202 $SCS_MAX_INP_FEED_PER_TOOTH	Upper limit feedrate/tooth
Default: 1 mm	Change to:

Description:

Upper limit for the selection of the feedrate/tooth

Setting up the tool list

These settings are applicable only to JobShop programs.

Enable fast M functions

MD52229 $MCS_ENABLE_QUICK_M_CODES	Enable fast M functions
Default: 0H	Change to:

Description:

Bit 0: Coolant OFF

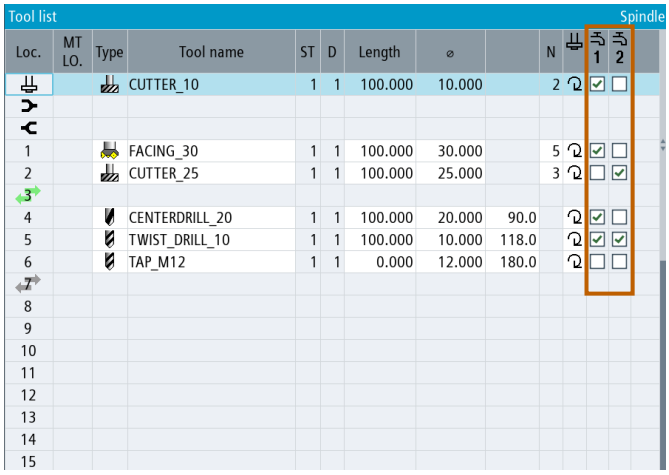
Bit 1: Coolant 1 ON

Bit 2: Coolant 2 ON

Bit 3: Coolant 1 and 2 ON

This MD is used to declare the commands used for the coolant control (see MD52230 - MD52233) as being fast M-commands (M=QU…). In case of fast commands the program progress continues without waiting for the acknowledgement of the M-command by the PLC.

In the tool list there are input fields for coolant 1 and coolant 2 for each tool. The select key can be used to toggle the function "coolant ON/OFF". The respective M-functions are put out if the tool is changed by means of the part-program. The M-commands will not be put out if the tool is exchanged by means of the T,S,M mask.



M-code for all coolants OFF

MD52230 $MCS_M_CODE_ALL_COOLANTS_OFF	M-code for all coolants OFF
Default: 9	Change to:

Description:

M-Code for all coolant OFF

M-code for coolant 1 ON

MD52231 $MCS_M_CODE_COOLANT_1_ON	M-code for coolant 1 ON
Default: 8	Change to:

Description:

M-Code for coolant 1 ON

Notes:

M-code for coolant 2 ON

MD52232 $MCS_M_CODE_COOLANT_2_ON	M-code for coolant 2 ON
Default: 7	Change to:

Description:

M-Code for coolant 2 ON

M-code for both coolants ON

MD52233 $MCS_M_CODE_COOLANT_1_AND_2_ON	M-code for both coolants ON
Default: -1	Change to:

Description:

M-Code for coolant 1 + 2 ON

Already defined: M8 = Coolant 1 ON
M7 = Coolant 2 ON

If in the JobShop tool list both coolants (coolant 1 and coolant 2) have been selected, the following M-command is generated:

-1: M-command for coolant 1 on (setting of MD52231) and M-command for coolant 2 on (setting of MD52232)

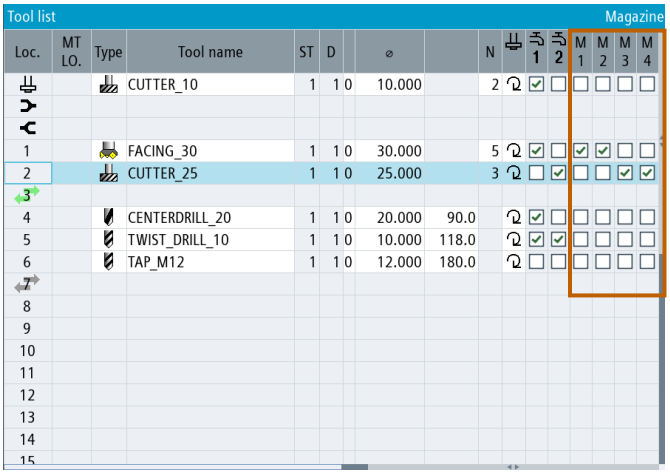
>0: the M-command selected in this MD.
A separate M-command for both coolants is to be used if the M-commands for coolant 1 and coolant 2 are grouped in one M-commands group, however both coolants are to be activated after a block search.

M-code for tool-specific function ON

MD52281[x] $MCS_TOL_MCODE_FUNC_ON	M-code for tool-specific function ON
Default: -1	Change to:

Description:

In MD52281[0] - [3] ON and
in MD52282[0] - [3] OFF up to 4 additional columns for tool-specific functions can be activated in the tool list. These can then be selected in the columns M1- M4 (see screenshot "Tool list" on the following page).



Description:

M-Code for tool-specific function ON
The value -1 means that the M-function will not be put out. If both M-commands of a function = -1, the associated field in the surface will not be shown.

Notes:

M-code for tool specific function OFF

MD52282[x] $MCS_TOOL_MCODE_FUNC_OFF	M-code for tool-specific function OFF
Default: -1	Change to:

Description:

M-Code for tool-specific function ON
The value -1 means that the M-function will not be put out. If both M-commands of a function = -1, the associated field in the surface will not be shown.

Function mask Tool management

MD52270 $MCS_TM_FUNKTION_MASK	Function mask Tool management
Default: 0H	Change to:

Description:

Bit 0: Create tool on a magazine location not permitted.
Tools can only be created outside the magazine.

Bit 1: Loading/unloading is not possible while the machine is not in Reset.
Tools can only be loaded or unloaded if the respective channel is at rest.

Bit 2: Loading/unloading locked during emergency OFF
Tools can only be loaded or unloaded if the Emergency OFF is not activated.

Bit 3: Loading/unloading tools to/from spindle disabled.
Tools cannot be loaded to or unloaded from the spindle.

Bit 4: Loading takes place in the spindle directly.
The tools are only loaded exclusively in the spindle.

Bit 5: Use grinding configuration file for creating the tool list.
Only grinding tools are offered.

Bit 6: Relocating a tool to/from the buffer (spindle/gripper) is permitted desoite a block (see bit3)

Bit 7: Create tool using the tool number. Specify the tool's T-number when creating the tool.
Note: This function is intended for machines with tool management but without magazine configuration. The default setting must not be changed.

Bit 8: Hide 'Re-locate tool'
The function 'Re-locate tool' is suppressed on the user interface.

Bit 9: Hide 'Magazine positioning'
The function 'magazine positioning' is suppressed on the user interface.

Bit 10: Re-activation of tool using magazine positioning
Before re-activation the tool is located on the loading position.

Bit 11: Re-activating the tool in all monitoring modes.
When re-activating a tool, all monitoring modes for this tool released in the NC will also be re-activated; i.e. also those monitoring modes that are not selected for the respective tool, but which are dormant in the background.

Bit 12: Hide 'Tool re-activation'
The function "Tool re-activating" is suppressed on the user interface.

Note:

This MD must be matched to the circumstances of the machine. By hiding those functions on the user interface that are not applicable to the machine, the PLC can be spared unnecessary actions to prevent operation errors.

Notes:

Load station Tool management

MD52274 $MCS_TM_TOOL_LOAD_STATION	Load station Tool management
Default: 0	Change to:

Description:

If only one loading station is configured: The setting = 0

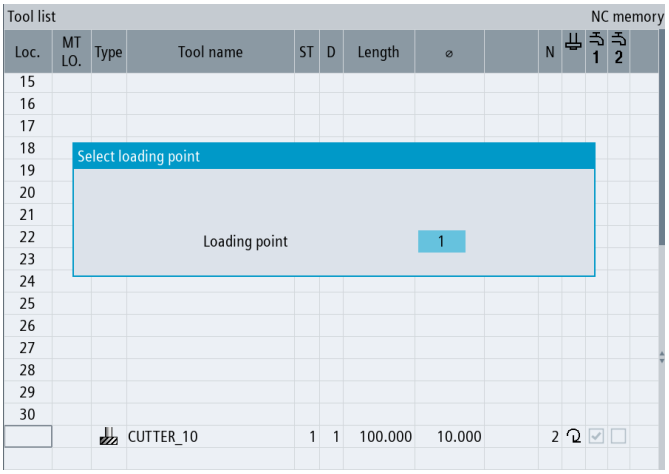
If more than 1 loading station is configured, you can select between

= 0: A query window appears on the surface to select the loading point.

= 1: fixed loading station 1, the window does not appear

= 2: fixed loading station 2, the window does not appear

The following screenshots show a configuration with two loading stations and the setting MD52274 = 0.



Acknowledge the selection made with OK.



Function mask Tool management

SD54215 $SNS_TM_FUNCTION_MASK_SET	Function mask Tool management
Default: 0H	Change to:

Description:

Bit 0: Diameter readout for rotating tools
It is not the radius value, but the diameter value that is shown for rotating tools.

Bit 1: Default direction of rotation for all turning tools is M4
When turning tools are created, the direction of rotation is preset to M4.

Bit 2: Creating tools without assigning a name
When creating a new tool the input field for the tool name remains empty.

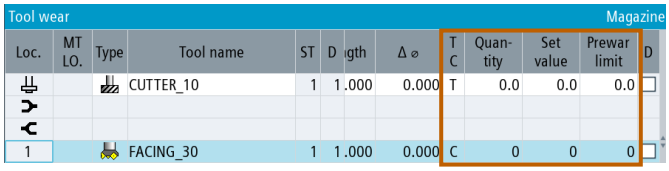
Bit 3: Input interlock for tool name and tool type for loaded tools
Tool name and tool type cannot be changed for loaded tools.

Bit 4: Input disable for loaded tools if the channel is not in reset
With running program the data of the loaded tools cannot be changed.

Bit 5: Evaluation of tool wear inputs additively
The input of wear data will be added to any existing wear values.

Bit 6: Numeric input of the tool identification
No name will be assigned to a new tool being created.
Only numerical characters are permitted .

Bit 7: Hiding the tool monitoring parameter
The tool monitoring parameter will not be shown on the user interface.



Notes:

Continued SD54215

Description:

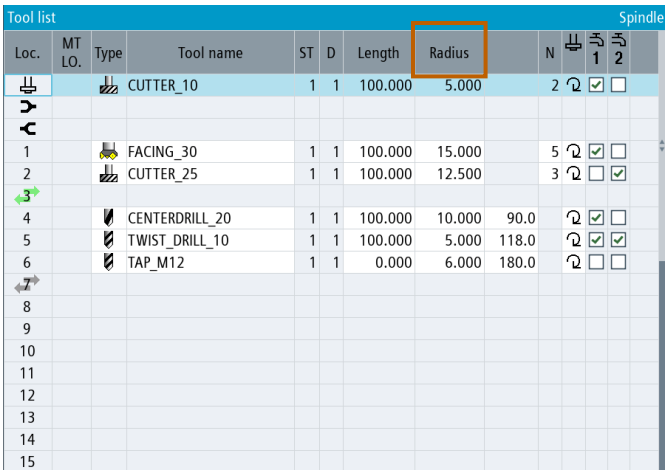
Bit 8: Diameter readout for transverse axis - geometry
The geometry value of the transverse axis is shown as a diameter value.

Bit 9: Diameter readout for transverse axis - wear
The wear value of the transverse axis is shown as diameter value.

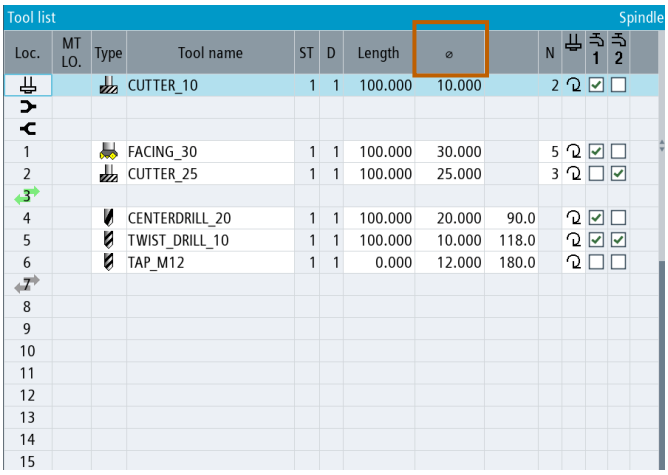
Bit 10: Release of Tool loading / Tool re-location to buffer station
The magazine number can be entered in the loading dialog. This permits access to the buffer station using the magazine number 9998.

Bit 11: The creation of new tools in the gripper locations is blocked.

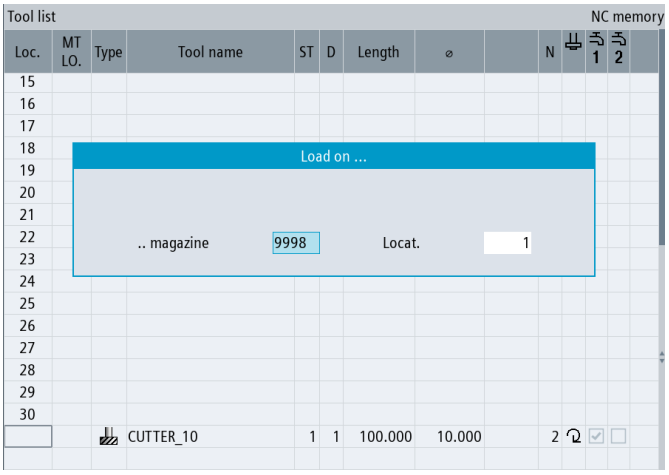
Bit 0 = 0 permanent radius readout for all tools



Bit 0 = 1 Diameter readout for rotating tools



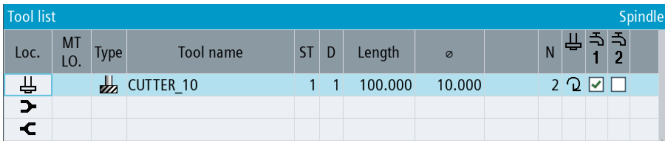
Bit 10 = 1 Release of Tool relocation to buffer station



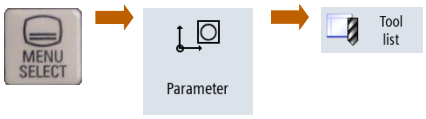
Notes:

Continued SD54215

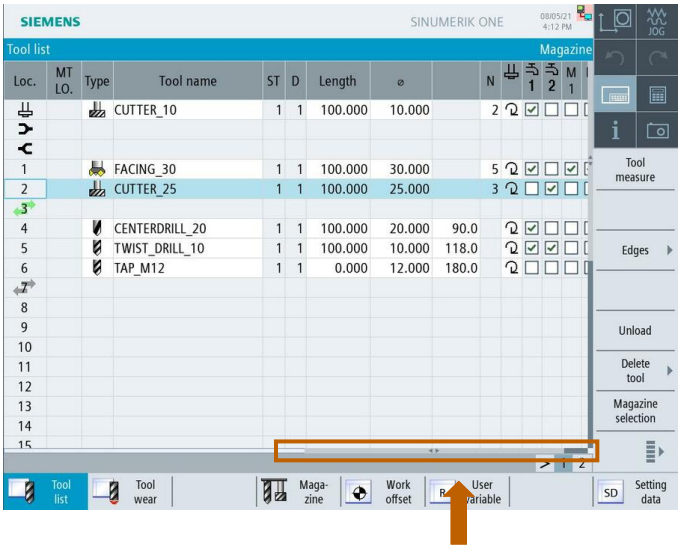
The locations 1 = spindle, 2 = gripper 1 and 3 = gripper 2 can be loaded directly by means of the magazine number 9998.
In the tool list these are the uppermost three lines without numbering.
In the shown example the "TWIST_DRILL_10" is loaded in the spindle.



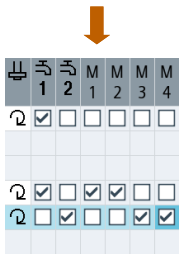
Checking the tool management

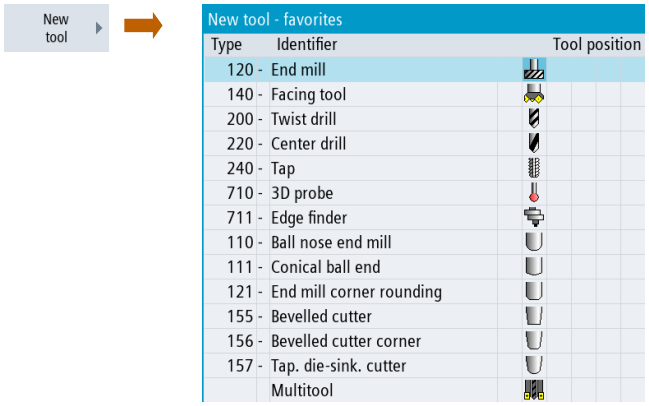


Marked tools can be unloaded. As soon as the cursor is moved to a free magazine location the Softkey "New tool" appears for the creation of a new tool.



By horizontal scrolling the fields for all tool-specific M-functions M1 … M4 appear.





The various types of tools are shown in the selection list.

Notes:

Basic settings

Reset-defiant settings

Definition of basic control settings after reset/PP end

MD20110 $MC_RESET_MODE_MASK	Definition of basic control settings after reset/PP end
	Change to: Bit 0 = 1 Bit 6 = 1 Bit 14 = 1

Description:

Determination of the basic control settings after booting and reset/end-of-part-program with respect to the G-codes (especially the current plane and selectable zero-point offset), tool length and transformation by the setting of the following bits:

Bit 0: Reset mode
Bit 6: Reset response "Active tool length offset"
Bit 14: Reset response "Initial frames"
Bit 16: Reset response "Master spindle"

Description:

The Bits 6 and 16 are only evaluated if Bit 0 = 1.

Meaning of the individual Bits:

Bit 0 = 0:
Initial setting after booting:
- G-Codes as per $MC_GCODE_RESET_VALUES
- Tool length offset not active
- Transformation not active
- no coupled axes groupings active
- no tangential correction active
- no axial rotational feed active
- path rotational feed with master spindle (Default)

Initial setting after Reset or after end-of-part-program :
The current settings will be retained.
With the next part-program Start the following Initial settings will become active:

- G-Codes as per $MC_GCODE_RESET_VALUES
- Tool length offset not active
- Transformation not active
- No coupled-motion groupings active
- No tangential offsets active
- No master value couplings active
- No axial rotational feedrate active
- Path rotational feedrate with master spindle (Default)

Bit 0 = 1:
Initial setting after booting:
- G-Codes as per $MC_GCODE_RESET_VALUES
- Tool length offset active as per $MC_TOOL_RESET_VALUE, $MC_CUTTING_EDGE_RESET_VALUE and $MC_SUMCORR_RESET_VALUE
- Transformation active per $MC_TRAFO_RESET_VALUE
- Geo axis replacement as per $MC_GEOAX_CHANGE_RESET
- No coupled-motion groupings active
- No tangential correction active

Initial setting after reset or after end-of-part-program:
Depending on the setting of $MC_GCODE_RESET_MODE the current settings for the G-groups will be retained or set to the initial values stored in $MC_GCODE_RESET_VALUES.

Description:

Initial setting after Reset or after End-of-part program:
Depending on $MC_RESET_MODE_MASK Bit 6 to 7 the current settings for
- Tool length offset
- Transformation

will either be retained or changed to the initial settings as stored in the MDs.

Depending on Bit 8 and 9 the current settings of coupled motion axes or tangentially corrected axes will either be retained or switched OFF.

Projected synchronous spindle coupling:
The coupling will be un-selected depending on the setting of $MC_COUPLE_RESET_MODE_1.

Non-projected synchronous spindle coupling:
Depending on Bit 10 the coupling will either be switched OFF or retained.
Depending on Bit 14 the initial frame will be retained or un-selected.

Note:
The setting of Bit 0 = 1 must be set for JobShop-uses, such that the settings of the Bits 4 and 16 can be evaluated.

Notes:

Continued MD20110

Bit 6 = 0:
Initial setting for the active tool length offset after Reset/End-of-part program as per $MC_TOOL_RESET_VALUE, $MC_CUTTING_EDGE_RESET_VALUE, $MC_USEKT_RESET_VALUE and $MC_SUMCORR_RESET_VALUE.
Is $MC_TOOL_CHANGE_MODE = 1, the tool specified by $MC_TOOL_PRESEL_RESET_VALUE is additionally pre-selected.
If the tool and/or magazine management is active, it is not the datum $MC_TOOL_RESET_VALUE that is used, instead $MC_TOOL_RESET_NAME will be used.

Bit 6 = 1:
The current setting for the active tool length offset remains in force after reset/End-of-part-program.
If the tool and/or magazine management is active, that tool will be selected which is presently loaded in the master spindle (generally = master tool holder).
If the tool in the master spindle happens to be blocked, the "blocked"-status will be ignored. .

Description:

However, it must be remembered that after the end of program or a program interruption either the last programmed value for the master spindle or master tool holder, or the value specified by $MC_SPIND_DEF_MASTER_SPIND or $MC_TOOL_MANAGEMENT_TOOLHOLDER will determine the master spindle or the master tool holder.
(The selection is made by Bit 16 or Bit 17.)

A special case is $MC_CUTTING_EDGE_DEFAULT = -2:
If a tool has been changed on the spindle, but no new correction D has been programmed, then the previous tool is still active in the NCK.
If an interruption takes place in this state, - e.g. by pressing the reset key, the correction will be specified by the lowest D-number of the tool in the master spindle.

Note:
The setting Bit 6=1 is necessary to ensure that after a reset/End-of-part-program the previously programmed tool and the previously programmed cutting edge remains activated (for instance, if the tool was changed via the T,S,M-mask) and if for example the JobShop-functions in the mode of operation JOG "Zero point work piece" or "Tool measurement" can be carried out with this tool.

Bit 14 = 0:
The basic frame is cancelled

Bit 14 = 1:
The current setting for the basic frame is retained.

Note:
The setting Bit 14 = 1 is necessary to retain the channel-specific basic work offset (WO), which was activated with G500, after a Reset/End-of-part-program.
After activation of the basic WO (JOG-T,S,M-mask - Work offset - basic reference) the values in the Work offset base are transferred to the Total basic WO and then from there to be transferred for activation also to the Total WO.

Description:

Bit 16 = 0:
Initial setting for the master spindle as per $MC_SPIND_DEF_MASTER_SPIND.

Bit 16 = 1:
The current setting of the master spindle (SETMS) is retained. This Bit in case of $MC_TOOL_MANAGEMENT_TOOLHOLDER = 0 has also an effect on the response of Bit 6.

Note:
The setting of Bit 16 is not applicable to ShopMill.

The previously described settings in MD20110 correspond to the further MD-settings in:

MD20120 $MC_TOOL_RESET_VALUE
MD20130 $MC_CUTTING_EDGE_RESET_VALUE
MD20150 $MC_GCODE_RESET_VALUES
MD20152 $MC_GCODE_RESET_MODE
MD20140 $MC_TRAFO_RESET_VALUE
MD20112 $MC_START_MODE_MASK
MD20121 $MC_TOOL_PRESEL_RESET_VALUE
MD20118 $MC_GEOAX_CHANGE_RESET

Notes:

Tool edge with length compensation during runup (reset/ end of part program)

MD20130 $MC_CUTTING_EDGE_ RESET_VALUE	Tool edge with length com- pens. during runup (reset/end of pp)
	Change to: 1

Description:

Specification of the cutting edge, which depending on MD20110 $MC_RESET_MODE_MASK and after part program Start depending on MD20112 $MC_START_MODE_MASK is used to select the tool length offset after booting and after Reset /End-of-part-program.

With active tool management and with Bit 0 and Bit 6 set in MD20110 $MC_RESET_MODE_MASK at selection, the last offset of the tool active at power OFF (as a rule the tool in the spindle) is effective after booting.

Corresponds to:
MD20110 $MC_RESET_MODE_MASK
MD20112 $MC_START_MODE_MASK

Note:

This setting is necessary to ensure that after a reset/End-of-part-program the previously programmed tool and the previously programmed cutting edge remains activated (for instance, if the tool was changed via the T,S,M-mask) and if for example the JobShop-functions in the mode of operation JOG "Zero point work piece" or "Tool measurement" can be carried out with this tool.

Notes:

Initial setting of G groups

MD20150[x] $MC_GCODE_RESET_ VALUES	Initial setting of G groups
	Change to: [15] = 3 [21] = 2 [41] = 1 [51] = 1 [52] = 1

MD20150[15] = 3 (CFIN)
MD20150[21] = 2 (CUT2DF)
MD20150[41] = 1 (TCOABS)
MD20150[51] = 1 (PAROTOF)
MD20150[52] = 1 (TOROTOF)

Description:

Specification of the G-Codes, which after booting and Reset/ End-of-part program are activated depending on MD20110 $MC_RESET_MODE_MASK and MD20152 $MC_GCODE_RESET_MODE as well as after part program Start depending on MD20112 $MC_START_MODE_MASK.

Note:
(exact) means that this setting is necessary in conjunction with JobShop; (variable) means that this should be seen as a suggestion and that another setting is permissible.

MD20150[15] = 3 (variable)
Group 16: Feedrate correction on internal and external curves
Value: 3 = CFIN
This setting provokes a constant feedrate on internal curves and acceleration on external curves

MD20150[21] = 2 (variable)
Group 22: Tool correction type
Value: 2 = CUT2DF
2½-D-tool correction caused by Frame
The tool correction acts relative to the current Frame (oblique plane)

MD20150[41] = 1 (exact)
Group 42: Tool carrier
Value: 1 = TCOABS
Determination of the tool length components from the current tool orientation

MD20150[51] = 1 (exact)
Group 52: Tool-related Frame rotation
Value: 1 = PAROTOF
The tool related Frame rotation is deactivated.
The work piece coordinate system is not aligned to the work piece.

MD20150[52] = 1 (variable)
Group 52: Tool-related Frame rotation
Value: 1 = TOROTOF
The tool related Frame rotation is de-activated.

Notes:

Reset response of G groups

MD20152[x] $MC_GCODE_RESET_MODE	Reset response of G groups
	Change to: [5] = 1 [7] = 1 [12] = 0 [21] = 0

Description:

This machine datum is evaluated only if in MD20110 $MC_RESET_MODE_MASK the Bit 0 is set. For each entry in MD20150 $MN_GCODE_RESET_VALUES (that is for each G group) this MD is used to specify whether

= 0: in case of a Reset/End-of-part-program the setting as per $MC_GCODE_RESET_VALUES is used again
or
= 1: the current setting is retained after Reset/End-of-part-program

Note:
(exact) means that this setting is necessary in conjunction with JobShop; (variable) means that this should be seen as a suggestion and that another setting is permissible.

MD20152[5] = 1 (exact)
Group 6: Plane selection
Value: 1
A plane switch (e.g. G17 for facing) programmed in the part-program remains active after Reset/ End-of-part-program.

MD20152[7] = 1 (exact)
Group 8: Selectable zero point offset
Value: 1
A zero point offset (e.g. G54) programmed in the part program remains active after Reset/End-of-part-program. This permits for example the activation of a zero point offset in the mode-of-operation JOG_T,S,M-mask Parameter zero point offset.

Description:

MD20152[12] = 0 (exact)
Group 13: Work piece dimensioning inch/metric
Value: 0
In the user interface it is possible in the mode-of-operation JOG Basic softkey line ">" Settings per Softkey "Switching Inch" to toggle the measuring system from metric to inch and vice versa.

MD20152[21] = 0 (variable)
Group 22: Tool correction type
Value: 0
After Reset/End-of-part-program the tool correction defined in MD20150[15] is activated once more. CUT2DF in this configuration example.

Maximum number of blocks w/o traversing movement with SAR

MD20202 $MC_WAB_MAXNUM_DUMMY_BLOCKS	Maximum number of blocks w/o traversing movement with SAR
	Change to: 10

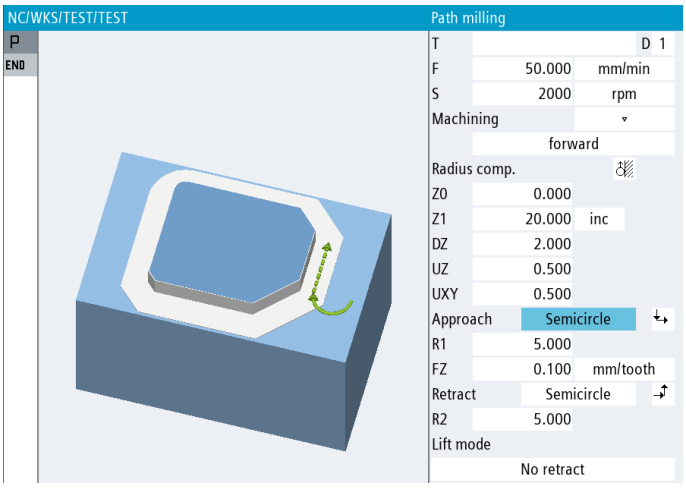
Description:

Maximum number of blocks that can appear between the SAR (soft approach and retraction) block and the traversing block which determines the direction of the approach or retraction tangent.

This setting is required in conjunction with the JobShop-cycles. A typical use is illustrated by the screenshot below taken from a continuous path milling operation on a contour using the approach and leaving strategy "Semi-circle".

Notes:

Continued MD20202



Blocks for look-ahead contour calculation with TRC

MD20240 $MC_CUTCOM_MAXNUM_ CHECK_BLOCKS	Blocks for look-ahead contour calculation with TRC
Default: 4	Change to:

Description:

Indicates the maximum number of blocks with traversing information at the offset plane that are considered simultaneously for collision detection with active TRC (tool radius compensation).

Note:

The Standard setting should not be undercut.

Maximum number of blocks without traversing motion in TRC

MD20250 $MC_CUTCOM_MAXNUM_ DUMMY_BLOCKS	Maximum number of blocks without traversing motion in TRC
	Change to: 5

Note:

The value 5 represents a value based on experience for JobShop installations.

Time monitoring for tool in tool holder

MD20320 $MC_TOOL_TIME_MONITOR _MASK	Time monitoring for tool in tool holder
	Change to: Bit 0 = 1

Description:

The setting Bit 0 = 1 is required for the life-time monitoring of the tool in the tool holder 1. The machine configuration used here only uses one tool holder.

Notes:

Active system frames after reset

MD24006 $MC_CHSFRAME_RESET_MASK	Active system frames after reset
	Change to: Bit 0 = 1 Bit 5 = 0

Description:

Bit-mask for the reset setting of the channel-specific system frames, which will be evaluated with the channel.

Bit 0: System frame active for preset. actual value/ scratching

Bit 5: System frame for cycles

Deletion of system frames after reset

MD24007 $MC_CHSFRAME_RESET_CLEAR_MASK	Deletion of system frames after reset
	Change to: Bit 0 = 0 Bit 5 = 1

Description:

Bit-mask for deleting channel-specific system frames in the data management on Reset..

Bit 0 = 0: The system frame for setting actual values and scratching (basic reference) will not be deleted with Reset/ End-of-part-program.

Bit 5 = 1: The system frame for cycles will be deleted with Reset/End-of-part-program.

Reset channel system frames after power ON

MD24008 $MC_CHSFRAME_POWERON_MASK	Reset channel system frames after power ON
	Change to: Bit 0 = 0 Bit 5 = 0

Description:

This machine data defines whether channel-specific system frames are reset in the data management on Power On. That is offsets and rotations are set to 0, scalings to 1. Mirroring is disabled.

The selection can be made separately for individual system frames.

Bit 0: System frame for set actual value and scratching is deleted after Power On.

Bit 5: System frame for cycles retained after Power On.

Activation and impact of program runtime measurement

MD27860 $MC_PROCESSTIMER_MODE	Activation and impact of program runtime measurement
	Change to: Bit 0 = 1 Bit 1 = 1 Bit 4 = 1 Bit 5 = 1 Bit 6 = 1 Bit 8 = 1

Description:

Under the function program runtime timers are provided as system variables. While the NCK-specific timers are always activated (for timings since the last control unit boot), the channel-specific timers must be started by means of this datum.

Meaning:

Bit 0 = 0
No timing of the total runtime for all part programs

Notes:

Continued MD27860

Bit 0 = 1
The timing of the total runtime for all part programs is active ($AC_OPERATING_TIME)

Bit 1 = 0
No timing of the current program runtime

Bit 1 = 1
The measurement of the current program run-time is active ($AC_CYCLE_TIME)

Bit 2 = 0
No timing of the tool cutting time

Bit 2 = 1
Timing of the tool cutting time is active ($AC_CUTTING_TIME)

Bit 3
Reserved

Bits 4,5 only if Bit 0 and 1= 1:

Bit 4 = 0 No timing during active test run feed motion

Bit 4 = 1 Timing also during active test run feed motion

Description:

Bit 5 = 0 No timing during program test

Bit 5 = 1 Timing also during program test

Bit 6 only if Bit 1 = 1:

Bit 6 = 0
Delete $AC_CYCLE_TIME also after Start by ASUP and PROG_EVENTs.

Bit 6 = 1
$AC_CYCLE_TIME will not be deleted after Start by ASUP and PROG_EVENTs.

Bit 7 only if Bit 2 = 1:

Bit 7 = 0 $AC_CUTTING_TIME counts only with active tool.

Bit 7 = 1 $AC_CUTTING_TIME counts independently of tool.

Bit 8 only if Bit 1 = 1:

Bit 8 = 0
$AC_CYCLE_TIME will not be deleted in case of a jump with GOTOS to the program start.

Description:

Bit 8 = 1
$AC_CYCLE_TIME will be deleted in case of a jump with GOTOS to the program start.

Bit 9 only if Bit 0, 1 = 1:

Bit 9 = 0
$AC_OPERATING_TIME, $AC_CYCLE_TIME: No measurement with override = 0.

Bit 9 = 1
$AC_OPERATING_TIME, $AC_CYCLE_TIME: Measurement also with override = 0.

Bits 10 to 31
Reserved

Work piece counters

A difference is made between the following work piece counters:

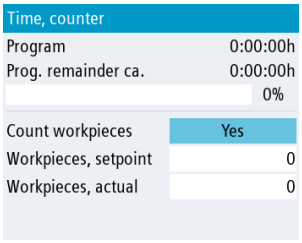
- Number of the command work pieces
  System variable $AC_REQURED_PARTS
  Setting by means of Bit 0 and 1
- Total number of work pieces (actual work pieces) finished since the Start
  System variable $AC_TOTAL_PARTS
  Setting by means of Bit 4, 5 and 7
- Number of all work pieces (actual work pieces) finished since the Start
  System variable $AC_ACTUAL_PARTS
  Setting by means of Bit 8, 9 and 11
- Number of work pieces finished as per the own strategy
  System variable $AC_SPECIAL_PARTS
  Setting by means of Bit 12, 13 and 15

Notes:

Continued Work piece counters

In the user interface of the mode-of-operation AUTO the mask shown alongside can be opened by selecting Time/ Counter





The following assignment applies:
- Parameter "Workpieces, setpoint": $AC_REQUIRED_PARTS
- Parameter "Workpieces, actual": $AC_ACTUAL_PARTS

Activation of workpiece counter

MD27880 $MC_PART_COUNTER	Activation of workpiece counter
	Change to: Bit 0 = 1 Bit 8 = 1 Bit 11 = 1

Bit 0 = 1: Counter $AC_REQUIRED_PARTS is activated
Further meaning of Bit 1-3 only if Bit 0 =1 and $AC_REQUIRED_PARTS > 0:

Further meaning of Bit 1 - 3 only if Bit 0 = 1 and $AC_REQUIRED_PARTS > 0:

Bit 1 = 0: Alarm/VDI-output if $AC_ACTUAL_PARTS corresponds to $AC_REQUIRED_PARTS

Bit 1 = 1: Alarm/VDI-output if $AC_SPECIAL_PARTS corresponds to $AC_REQUIRED_PARTS

Bit 2: reserved

Bit 3: reserved

Bit 4 = 1: Counter $AC_TOTAL_PARTS is active

Further meaning of Bit 5 - 7 only if Bit 4 = 1 and $AC_TOTAL_PARTS > 0:

Bit 5 = 0: Counter $AC_TOTAL_PARTS is increased by the value 1 after a VDI-output of M02/M30.

Bit 5 = 1: Counter $AC_TOTAL_PARTS is increased by the value 1 after the output of the M-command from MD PART_COUNTER_MCODE[0]

Bit 6 : reserved

Bit 7 = 1: Counter $AC_TOTAL_PARTS is increased by the value 1 after a jump back with GOTOS

Bit 8 = 1: Counter $AC_ACTUAL_PARTS is active

Further meaning of Bit 9 - 11 only if Bit 8 = 1 and $AC_REQUIRED_PARTS > 0:

Bit 9 = 0: Counter $AC_ACTUAL_PARTS is increased by the value 1 after a VDI-output of M02/M30

Bit 9 = 1: Counter $AC_ACTUAL_PARTS is increased by the value 1 after the output of the M-command from MD PART_COUNTER_MCODE[1]

Bit 10: reserved

Bit 11 = 1: Counter $AC_ACTUAL_PARTS is increased by the value 1 after a jump back with GOTOS.

Bit 12 = 1: Counter $AC_SPECIAL_PARTS is active

Further meaning of Bit 13-15 only if Bit 12 =1 and $AC_REQUIRED_PARTS > 0:

Bit 13 = 0: Counter $AC_SPECIAL_PARTS is increased by the value 1 after a VDI-output of M02/M30

Bit 13 = 1: Counter $AC_SPECIAL_PARTS is increased by the value 1 after output of the M-command from MD PART_COUNTER_MCODE[2]

Bit 14: reserved

Notes:

Continued MD27880

Bit 15 = 1: Counter $AC_SPECIAL_PARTS is increased by the value 1 after a jump back with GOTOS.

Application example:

MD27880 PART_COUNTER = 101H
Bit 0 = 1: Counter $AC_REQUIRED_PARTS "Work pieces Command" activate
Bit 8 = 1: Counter $AC_ACTUAL_PARTS "Work pieces Actual" activate

MD 27882[1] PART_COUNTER_MCODE = 33

Part program:

In such a JobShop-Program the program repeat can be switched ON in the automatically generated End-of-program block by means of the parameter "Repetition". In this case the command GOTOS will be generated.

In a G code part program the program repetition can be done by:

G-Code-Program_1:	G-Code-Program_2:
…	...
M33; Repeat	GOTOS; Repeat
M30	M30

The number of work pieces is to be set in the mode-of-operation AUTO in the mask “Time/Counter” by the parameter “Workpieces, setpoint”.

Relevant PLC interface signals:

Activating the part program repetition is done by setting the PLC interface signal "Program branches control" to 1.
- For ONE / 840D sl: PLC bit DB21.DBX384.0
- For 828D: PLC bit DB3200.DBX16.0

When the actual number of workpieces matches the number of setpoint the PLC interface signal "Workpiece setpoint reached" is set to 1.
- For ONE / 840D sl: PLC bit DB21.DBX317.1
- For 828D: PLC bit DB3300.DBX4001.1

Memory space for REORG (DRAM)

MD28000 $MC_MM_REORG_LOG_ FILE_MEM	Memory space for REORG (DRAM)
	Change to: 75

Number of blocks for local user variables in REORG (DRAM)

MD28010 $MC_MM_NUM_REORG_ LUD_ MODULES	Number of blocks for local user variables in REORG (DRAM)
	Change to: 20

Notes:

Projecting of channel-specific system frames

MD28082 $MC_MM_SYSTEM_FRAME_MASK	System frames (SRAM)
	Change to: Bit 0 = 1 Bit 5 = 1

Description:

Bit mask for configuring channel-specific system frames included in the channel.

Bit 0: System frame for setting actual value and scratching

Bit 5: System frame for cycles

Note:
The system frames for actual value set and scratching (basic reference) and for cycles are required for JobShop installations.

Projecting of channel-specific system frames in the data management

MD28083 $MC_MM_SYSTEM_DATAFRAME_MASK	System frames (SRAM)
	Change to: Bit 0 = 1 Bit 5 = 0

Description:

Bit mask for configuring channel-specific system frames in the data storage (SRAM).

Bit 0: System frame for setting actual value and scratching

Bit 5: System frame for cycles

Notes:

Work offsets in frames

SD42440 $SC_FRAME_OFFSET_INCR_PROG	Work offsets in frames
	Change to: 0

Note:

This setting is required for JobShop installations.

Tool length compensations

SD42442 $SC_TOOL_OFFSET_INCR_PROG	Tool length compensations
	Change to: 0

Note:

This setting is required for JobShop installations.

Feed reduction on circles with tool radius compensation

SD42528 $SC_CUTCOM_DECEL_LIMIT	Feed reduction on circles with tool radius compensation
	Change to: 1

Note:

This setting is required for JobShop installations. It avoids the "stopping" of the axes in such circular sections if the diameter of the milling cutter is only marginally smaller than the machining radius.

Change of tool length components with change of active plane

SD42940 $SC_TOOL_LENGTH_CONST	Change of tool length components with change of active plane
	Change to: 0

Description:

If this setting data is not equal to 0, the assignment of tool length components (length, wear, base dimensions) to geometry axes is not changed when the machining plane (G17 - G19) is changed.

Assignment of tool length compensation independent of tool type

SD42950 $SC_TOOL_LENGTH_TYPE	Assignment of tool length compensation independent of tool type
	Change to: 0

Description:

This setting data defines the assignment of the tool length components to the geometry axes independently of the tool type. It can assume any value between 0 and 3. Any other value is interpreted as 0.

Notes:

JOG-TSM

In the operation mode JOG the input masks for "T,S,M", "Position" and "Face milling" are available. These permit simple functions to be carried out such:
- T,S,M: Tool changes, spindle functions, work offset selections
- Position: Axes and spindles positioning
- Face milling: Roughing of a workpiece.

After the data have been entered and NC-Start initiated the file MA_JOG_STEP1.MPF will be generated in the directory "Workpieces - TEMP " in the program-manager and executed.

To ensure that such functions as e.g. tool selection and spindle Start will remain activated also after Reset and End-of program, respective machine data are required, which will be described below.

The settings will be described with the T,S,M-Mask; the input masks Position and Face milling do not require special settings.

Tool change

Definition of basic control settings after Reset/PP end

MD20110 $MC_RESET_MODE_MASK	Definition of basic control settings after reset/PP end
	Change to: Bit 6 = 1

Bit 0 = 1 Reset mode
Bit 6 = 1 Active tool length compensation
Bit 14 = 1 Base frame

Description:

Bit 6 = 1:

The current setting for the active tool length correction remains active also after Reset/End-of-part program.
If the tool or magazine management is active, that tool is selected which happens to be in the master spindle (generally referred to as master-tool holder).
If the tool in the master spindle is locked, the "locked" status will be ignored.

It must be borne in mind that after end-of-program, program abort either the last programmed value for the master spindle respectively the master tool holder or else the value specified by $MC_SPIND_DEF_MASTER_SPIND respectively $MC_TOOL_MANAGEMENT_TOOLHOLDER for the master spindle respectively the master-tool holder will be used.

Activation of tool management functions

MD20310 $MC_TOOL_MANAGEMENT_MASK	Activation of tool management functions
	Change to: Bit 14 = 1

Bit 0 - 3 must be set as in MD18080 $MN_MM_TOOL_MANAGEMENT_MASK

Bit 0 = 1 Magazine management active
Bit 1 = 1 Monitoring functions active
Bit 3 = 1 Adjacent location treatment
Bit 14 = 1 autom. tool change during Reset and Start

Description:

Bit 14 = 1: Reset-Mode
Tool and correction selection as per the settings of MD20110 $MC_RESET_MODE_MASK and
MD20112 $MC_START_MODE_MASK.

Notes:

Spindle

These settings are required to enable spindles to be started from the T,S,M mask.

M function for spindle active after reset

MD10714 $MN_M_NO_FCT_EOP	M function for spindle active after reset
	Change to: 32

Suggestion M32

Description:

With Start in the manual mode-of-operation (T,S,M-mask, position) the part-program MA_JOG_STEP1 will be generated and executed.
This part-program will be finished with the M-function in MD10714. The setting of MD35040 = 2 ensures that the spindle remains active also after the end-of-program.

Limitations see MD10715 $MN_M_NO_FCT_CYCLE

Own spindle RESET

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET	Own spindle RESET
	Change to: 2

AX4: MSP1
MD35040 = 2 Own spindle reset

Description:

This machine data defines the response of the spindle after Reset and program end (M2, M30).

This MD is only active in the spindle mode open-loop control mode. In the positioning or oscillation mode, the spindle is always stopped.

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET = 0:

- Spindle stops (with M2/M30 and channel and mode group reset)
- Program is aborted

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET= 1:

- Spindle does not stop
- Program is aborted

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET= 2:

- Spindle does not stop at the M function configured via MD10714 $MN_M_NO_FCT_EOP (e.g. M32).
- However, the spindle stops at channel or mode group reset.

Not relevant to:

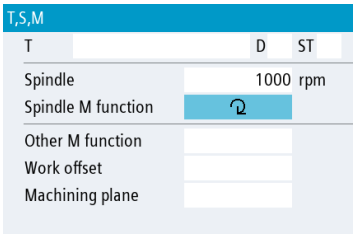
- Spindle modes other than open-loop control mode.

Start main spindle via the T,S,M-mask

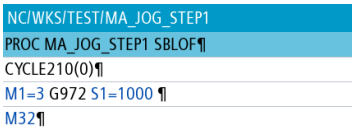
T,S,M-mask with inputs for spindle speed and direction of rotation of the main spindle S1

Enter speed and direction of rotation

and actuate NC-Start.



MA_JOG_STEP1 is generated.



Notes:

Continued Start main spindle via the T,S,M-mask

Description:

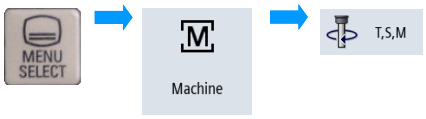
- M1=3 G972 S1=1000
  Starting of the spindle

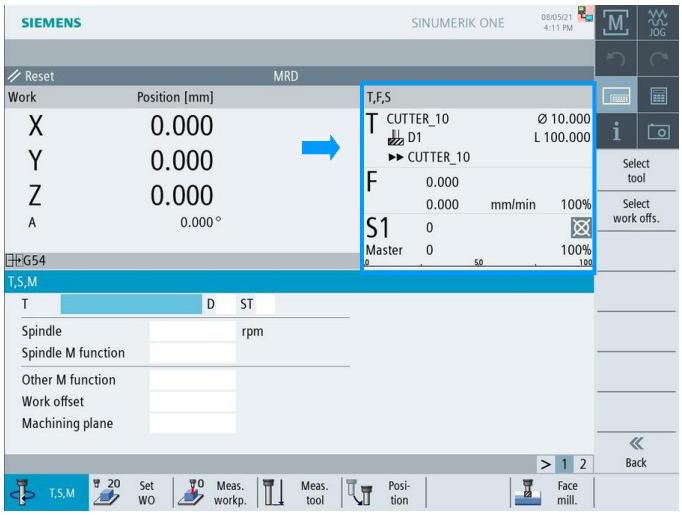
Note:
In case of a ShopMill installation there will be no CUST_TECHCYC call-ups for the spindle control from the T,S,M-mask and during program execution.

Note on spindle direction of rotation:

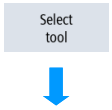
The assignment of the direction arrow for the spindle M-function M3/M4 is effected via the MD52207 $MCS_AXIS_USAGE_ATTRIB.

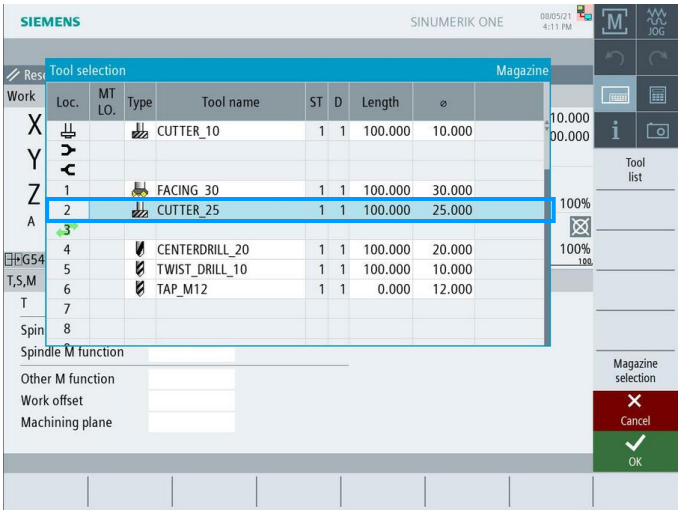
Monitoring the tool change

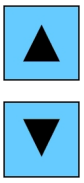




Shown in the T,F,S-field is the tool currently loaded in the spindle.





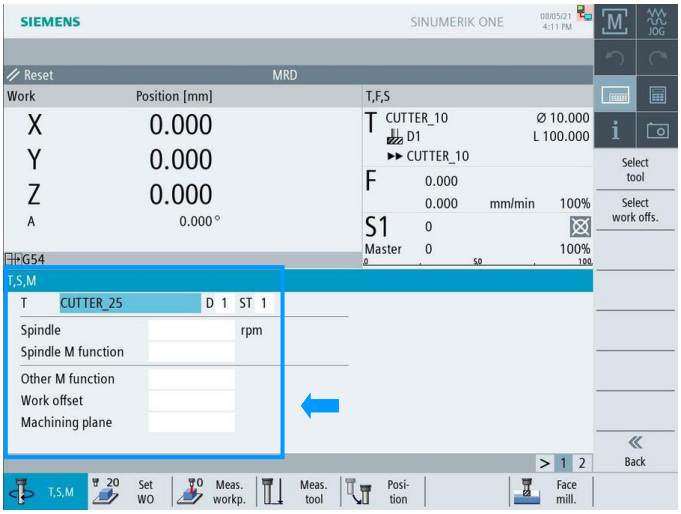


Use the cursor keys to select the required tool and thereafter transfer it by means of the Soft-Key "ÓK" .

Notes:

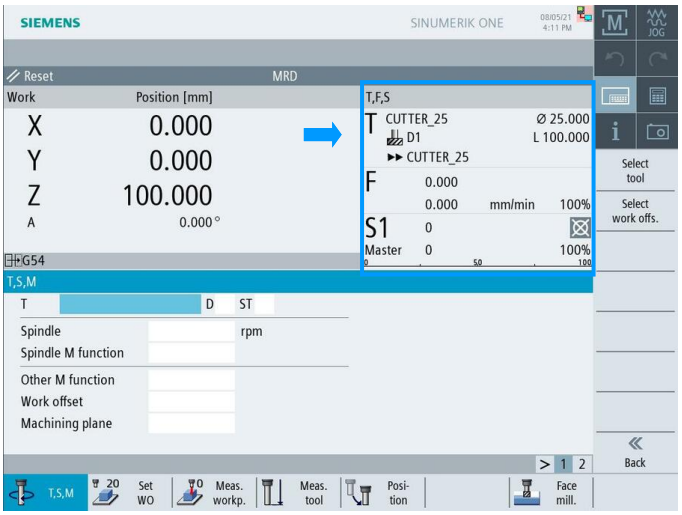
Loading the new tool





The selected tool is now displayed in the T,S,M mask.





The tool will be loaded into the T,F,S mask with Cycle Start. The PLC-tool change functions provided by the machine manufacturer are executed in the background for this process.

Gear step

Parameterize gear stage change

MD35010 $MA_GEAR_STEP_CHANGE_ENABLE	Parameterize gear stage change
	Change to: Bit 0 = 1

Description:

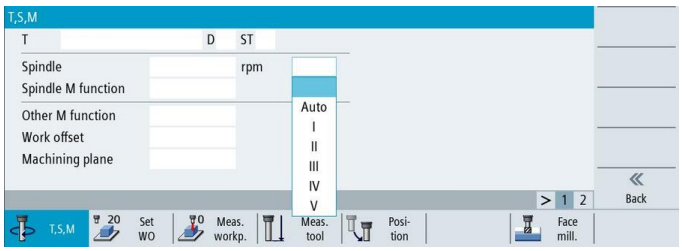
The selection field Gear stage appears only if in MD35010 for the spindle Bit 0 = 1 is set.

Number of gear stages

MD35090 $MA_NUM_GEAR_STEPS	Number of gear stages
Default: 5	Change to:

Description:

Number of prepared gear stages.



With NC-Start the program MA_JOG_STEP1 will be generated - in this example for the selection of the gear stage 2 M1=42.

```
NC/WKS/TEST/MA_JOG_STEP1
PROC MA_JOG_STEP1 SBLOF¶
CYCLE210(0)¶
M1=42 ¶
M32¶
```

Notes:

Inch/Metric-switching

Below once again a summation of the machine data settings required for Inch/metric switching.

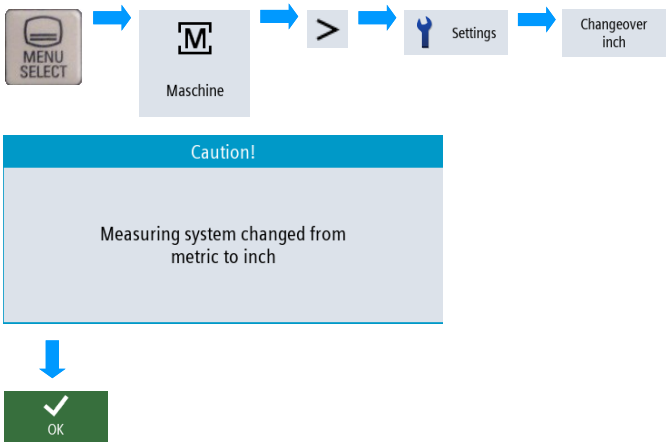
Enable basic system conversion

MD10260 $MN_CONVERT_SCALING_SYSTEM	Enable basic system conversion
	Change to: 1

Description:

Switching the measuring systems with HMI-Softkey
0. Hide Softkey inch/metric switching
1. Display Softkey inch/metric switching

Selection of Inch/Metric-switching:



Note:

MD 20152[12] = 0 must be set for the position readouts to be converted.

Block search

Parameterization for search run

MD11450 $MN_SEARCH_RUN_MODE	Parameterization for search run
	Change to: Bit 0 = 1 Bit 1 = 1 Bit 2 = 1 Bit 6 = 1

Description:

Bit 0: Response to output of alarm 10208

Bit 1: Automatic ASUP start after action blocks

Bit 2: Response to Auxf output (spindle)

Bit 6: Block search with simple, computing-time-optimized algorithms

Function mask Cross-technology

MD52212 $MCS_FUNCTION_MASK_TECH	Function mask Cross-technology
	Change to: Bit 3 = 1

Description:

Bit 3: Call block search cycle for ShopMill/ShopTurn

Note:

The setting of Bit 3=1 ensures that the tool programmed after the block search is duly changed and that the spindle rotates in the tool-assigned direction at the speed programmed in the destination block. For installations without ShopMill this Bit needs not to be set, however, the machine manufacturer must ensure that the tool and spindle functions are gathered and put out during the block search.

Notes:

Tool change commands to PLC after search run

MD20128 $MC_COLLECT_TOOL_ CHANGE	Tool change commands to PLC after search run
	Change to: 0

Description:

This MD is of any meaning only with activated magazine management (MD18080 $MN_MM_TOOL_MANAGEMENT_MASK, MD20310 $MC_TOOL_MANAGEMENT_MASK).
It specifies whether after block search with evaluation tool changing commands, tool preparation commands (generally tool changing commands) are put out to the PLC or not put out to the PLC.

Note:

This setting is necessary for ShopMill/ShopTurn.

1: Tool changing commands and tool preparation commands are gathered and passed on to the PLC with program Start when the search run has attained the destination block.

0: all tool and/or magazine specific commands that were gathered during the block search will not be put out to the PLC with the next program Start! That means that also programmed POSM, TCI, TCA will not be passed on to the PLC.

Note 1:

Without an active magazine management the tool change-M-code will not be gathered, if it is not assigned to an auxiliary function group. With an active magazine management this corresponds to an MD-value = 0.

Note 2:

The value = 0 is sensible for instance if after reaching the search destination the gathered tool change commands are transferred to the PLC in an ASUP-program with the aid of the commands GETSELT, GETEXET.

Simulation

Adjustment of SZS Coordinate system

MD24030 $MC_FRAME_ACS_SET	Adjustment of SZS Coordinate system
	Change to: 1

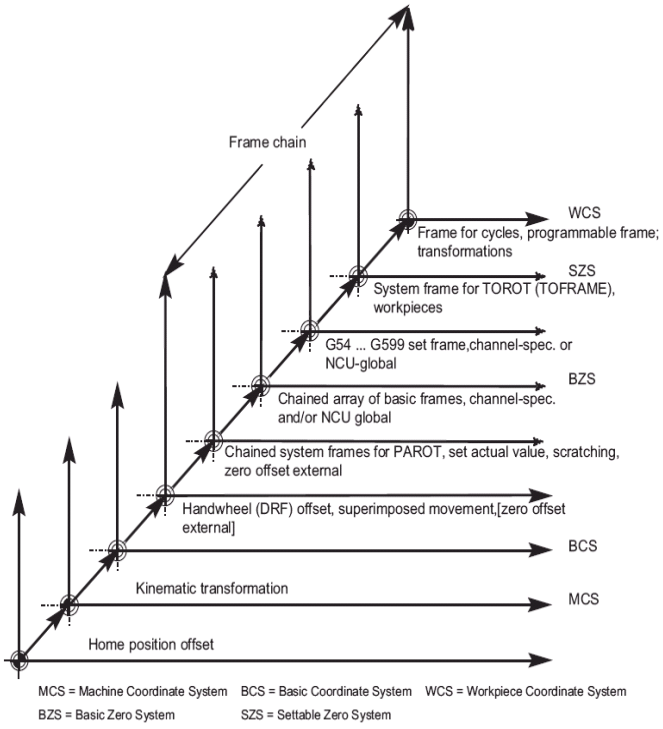
Description:

0: SZS results from the WCS transformed with $P_CYCFRAME and $P_PFRAME.

1: SZS results from the WCS transformed with $P_CYCFRAME.

The position readouts show the SZS-Coordinate system if "WCS" is selected.

Depiction of the Coordinate systems and the Frame chain



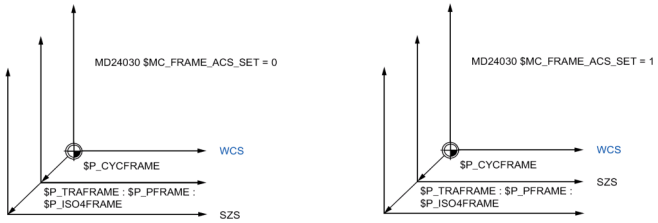
Notes:

Description:

The SZS-Coordinate system is different to the WCS Coordinate system by

- Cycle frame
- Programmable frames (G54, …)
- Transformation (TRANS, ROT, SCALE)

With the setting of MD24030 $MC_FRAME_ACS_SET = 1 the programmable frames are taken into account in the position readouts.



Axis position at start of simulation

MD53230 $MAS_SIM_START_ POSITION	Axis position at start of simulation
	Change to:

AX1: MX1
MD53230 = e.g. 0 mm

AX2: MY1
MD53230 = e.g. 0 mm

AX3: MZ1
MD53230 = e.g. 649 mm

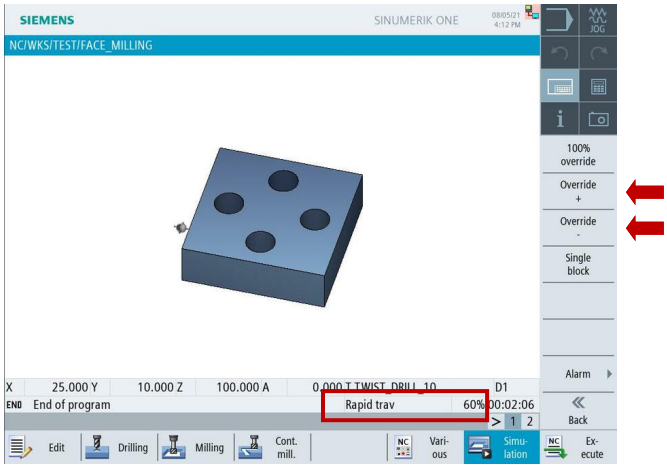
Description:

Axis position at the Start of the simulation.

Note:

For the further axes and spindles this MD should be left on 0.

Simulation description



Tipp:

To permit better observation of the simulation it is advisable to reduce the rapid traverse rate by means of the override switch. The percentage settings do not represent a linear function. A setting of about 60 - 80% are practicable.

Alternatively to the vertical Softkeys the override setting can also be obtained using the following keys:

Override increase: CTRL and ▲

Override reduction: CTRL and ▼

Override maximum: CTRL and M

Notes:

Machining example

Machining example (part program)

The following describes a programming example which demonstrates a subsequent machining operation with planar milling and drilling.

Select tools

Tool list — Spindle

Loc.	MT LO.	Type	Tool name	ST	D	Length	⌀			1	2
1			FACING_30	1	1	100.000	30.000		5		
2			CUTTER_25	1	1	100.000	25.000		3		
3			CUTTER_10	1	1	100.000	10.000		2		
4			CENTERDRILL_20	1	1	100.000	20.000	90.0			
5			TWIST_DRILL_10	1	1	100.000	10.000	118.0			
6			TAP_M12	1	1	0.000	12.000	180.0			

Set work offsets

Work offset - G54 ... G57 [mm]

	X	Y	Z	A	SP1
G54	0.000	0.000	100.000	0.000	0.000
Fine	0.000	0.000	0.000	0.000	0.000

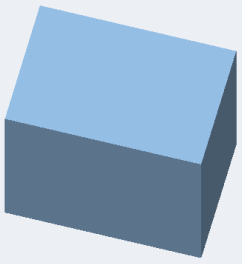
Machining program: Face milling

NC/WKS/TEST/FACE_MILLING 1

P	Program header	G54 Block
	Face milling	T=CUTTER_10 F=0.2/t S=2000rev X0=0 Y0=0 Z0=0 Z1=10inc
	Deep hole drill.1	T=TWIST_DRILL_10 F=100/min S=2000rev Z1=10inc
	001: Posit. circle	Z0=-10 X0=25 Y0=25 R=15 N=4
END	End of program	

Program head with workpiece definition:

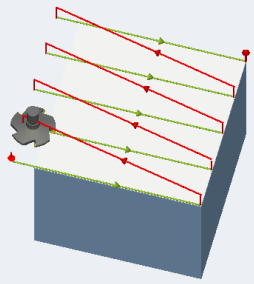
NC/WKS/TEST/FACE_MILLING — Program header



Work offset	G54	
Blank	Block	
X0	0.000	
Y0	0.000	
X1	50.000	abs
Y1	50.000	abs
ZA	0.000	
ZI	-25.000	abs
PL	G17 (XY)	
Retraction plane		
RP	100.000	
Safety distance		
SC	1.000	
Machining sense		
	Down-cut	
Retract position pattern		
	To RP	

Face milling and technology data:

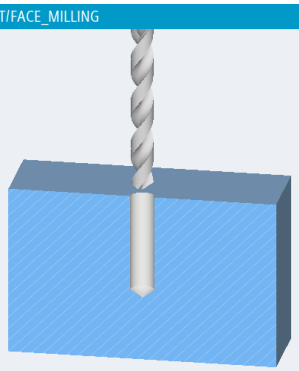
NC/WKS/TEST/FACE_MILLING — Face milling



T	CUTTER_10	D 1
F	0.200	mm/tooth
S	2000	rpm
Machining		
Direction		
X0	0.000	
Y0	0.000	
Z0	0.000	
X1	50.000	abs
Y1	50.000	abs
Z1	10.000	inc
α0	0.000	°
DXY	70.000	%
DZ	5.000	
UZ	0.000	

Deep hole drilling and technology data:

NC/WKS/TEST/FACE_MILLING — Deep hole drilling 1

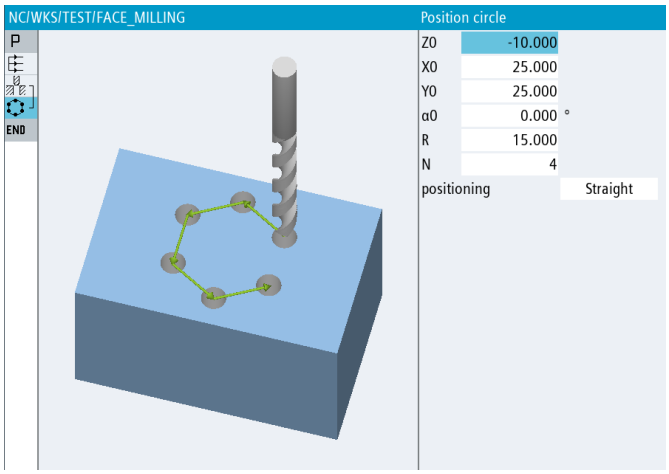


T	TWIST_DRILL_10	D 1
F	100.000	mm/min
S	2000	rpm
	Chipbreaking	
	Shank	
Z1	10.000	inc
FD1	100.000	%
D	10.000	
DF	90.000	%
V1	5.000	
V2	5.000	
DT	0.600	s

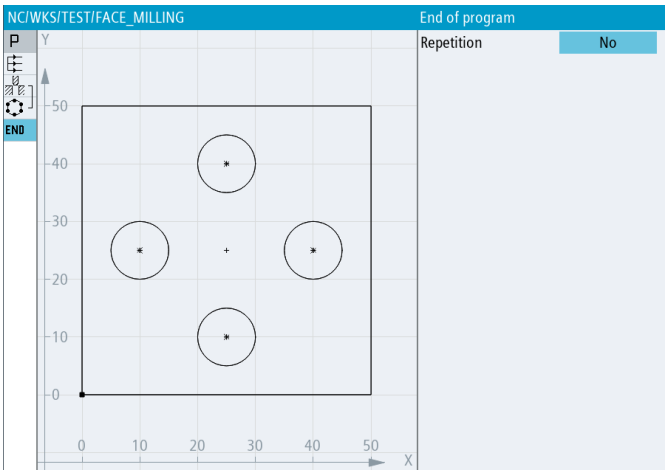
Notes:

Continuation Machining example (part program)

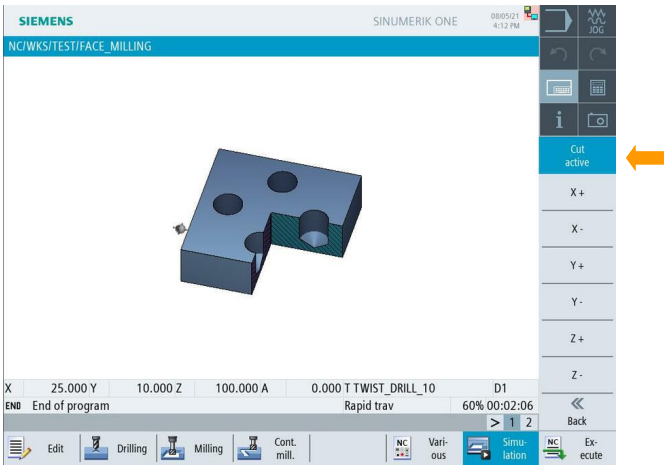
Position circle for deep hole drilling:



Program end and repetition:



Simulation depiction:



Note:

For better representation in the 3D view the function "Section" was used.
Selection of the simulation: Details - Section
Using the softkeys "X +" to "Z -" the section axes can be set.

Under "Details - rotate view" the work piece can be rotated into the required position.

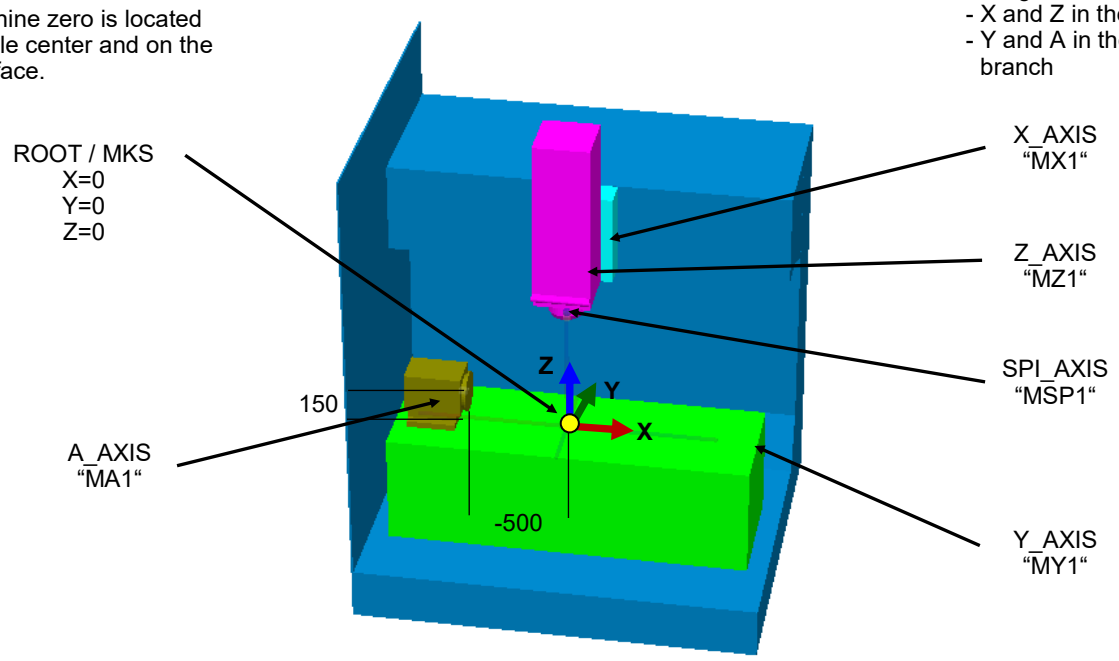
Notes:

Machining with A axis

Example: 3-axis milling machine with dividing unit

The machine zero is located in the table center and on the table surface.

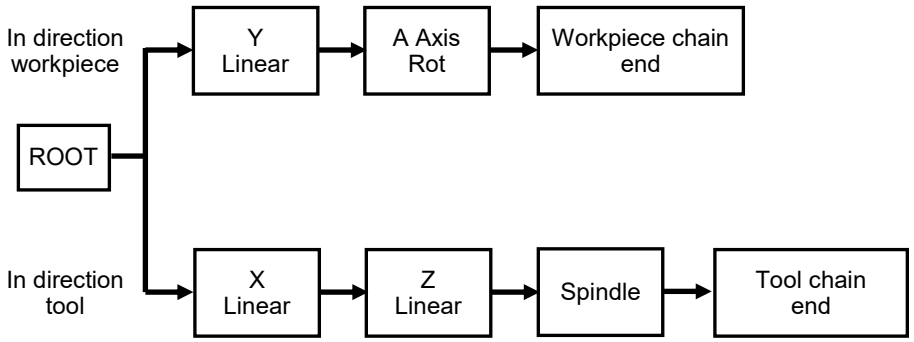
Arrangement of the axes
- X and Z in the tool branch
- Y and A in the workpiece branch



Basic structure of the kinematic chain

The kinematic chain is described based on the machine frame in the direction of the workpiece and in the direction of the tool.

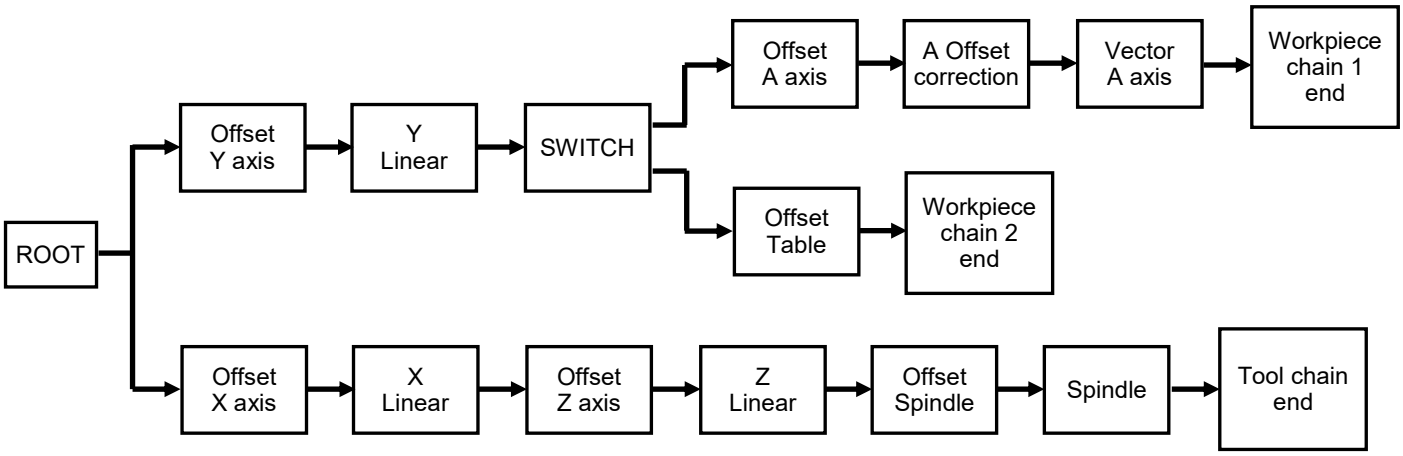
The **ROOT** element represents the machine zero.



Notes:

Detailed representation of the kinematic chain

Each axis or each deployed spindle should consist of an offset element (distance from the previous element) and an axis element (linear axis, rotary axis).



Axis element (Linear axis / rotary axis)

The definition of a linear axis always requires a vector based on the coordinate system currently valid in the chain. The definition of a rotation axis masks an assignment to a rotary axis that turns by the defined vector.

The sign of the vector is defined depending on whether the axis moves the workpiece or the tool.

Sign +: The axis moves the tool
Sign -: The axis moves the workpiece

The assignment of the axis name is based on the machine name as defined in MD10000[n].

Switch (SWITCH)

A switch allows a chain at a defined position to be steered specifically to a different branch. A typical use is for activation of the dividing unit. The switch must be actuated in the NC program by the user or the machine manufacturer.

Offset element

This element can be deployed to represent constant offsets (preferably rotary axis).

For the representation of rotary axis whose position should be corrected later with CYCLE9960, the total offset formed by chaining two individual offsets.

1st offset: X/Y/Z values as specified in the design drawing
2nd offset: Target element for correction by CYCLE9960

Arrangement of correction elements when using CYCLE9960 (kinematic measurement).

The following principle applies:

If the kinematic chain should be processed from the end of the part chain in the tool chain end direction, each of the correction elements must be arranged in this travel direction behind the associated rotary axis.

Notes:

Create a kinematic chain

For creating the kinematic chain on the HMI user interface, an HMI tool is available at the “Commissioning” area.

All required steps can be performed with this tool.

Machine data settings

Maximum number of elements in kinematic chains
MD18880 $MN_MM_MAXNUM_KIN_CHAIN_ELEM = 50

Maximum number of switches in the kinematic chains
MD18882 $MN_MM_MAXNUM_KIN_SWITCHES = 1

Start the configuration

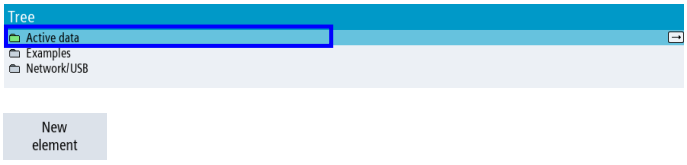




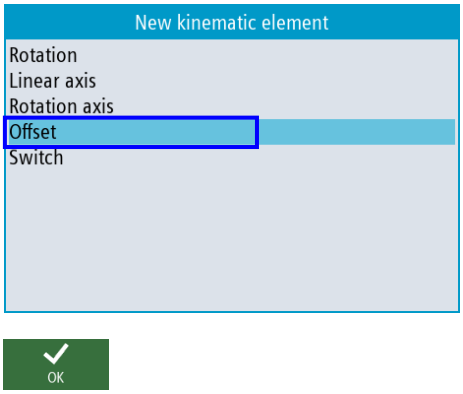
The “Machine model” menu contains the following areas:
Active data contains the kinematic chain of the machine
Examples contains examples for the kinematic chain
Network / USB contains the connections

Create a new kinematic chain

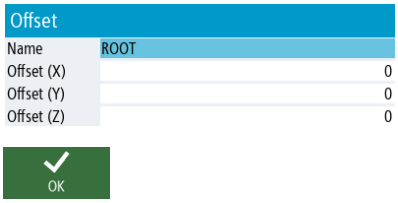
Position the cursor on the "Active data" folder and press the "New element" softkey.



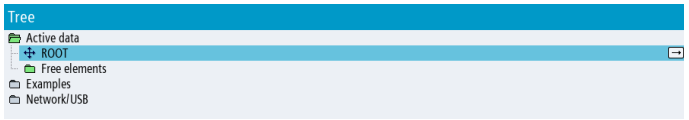
Select an “Offset” element from the input dialog and confirm with the "OK" softkey.



The name of the element is initialized automatically by the NC with ROOT, all the offset values can remain at the value 0.



The ROOT element is now created as first element in the structure tree.

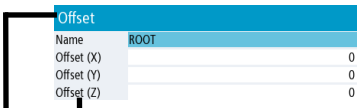


In the next step, the kinematic chain is further extended in the workpiece direction (=workpiece chain), then in the tool direction (=tool chain).

Notes:

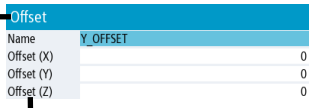
Create a workpiece chain

ROOT element

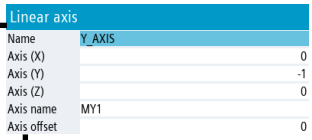


ROOT The name of the element is initialized automatically by the NC; all the offset values can remain at the value 0

Y axis

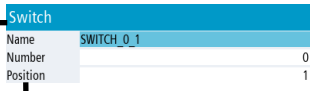


Y_OFFSET The offset vector is defined with (0,0,0)

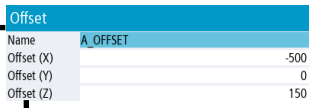


Y_AXIS The linear axis references machine axis "MY1" and has vector (0,-1,0)

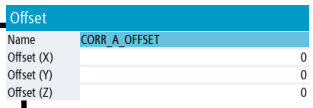
A axis (dividing unit) on the table



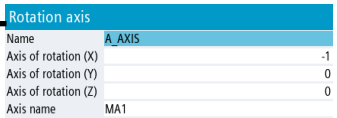
SWITCH_0_1 The branch is valid for $NK_SWITCH[0] =1



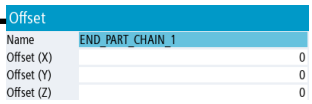
A OFFSET The offset vector is defined as (-500,0,150) - the coarse offset of the A axis, 150 mm over the table surface



CORR_A_OFFSET The vector is entered with (0,0,0) - it is used later by CYCLE9960

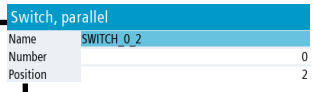


A_AXIS The rotation axis references machine axis "MA1" and has vector (-1,0,0)

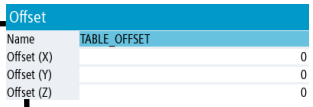


END_PART_CHAIN_1 The offset vector is initialized with (0,0,0)

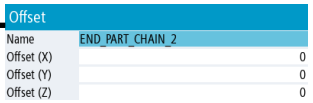
Machine without A axis (Next parallel element)



SWITCH_0_2 The branch is valid for $NK_SWITCH[0] =2



TABLE_OFFSET The offset is defined with the vector (0,0,0) and so points to the table surface



END_PART_CHAIN_2 The offset vector is initialized with (0,0,0)

To the tool chain...

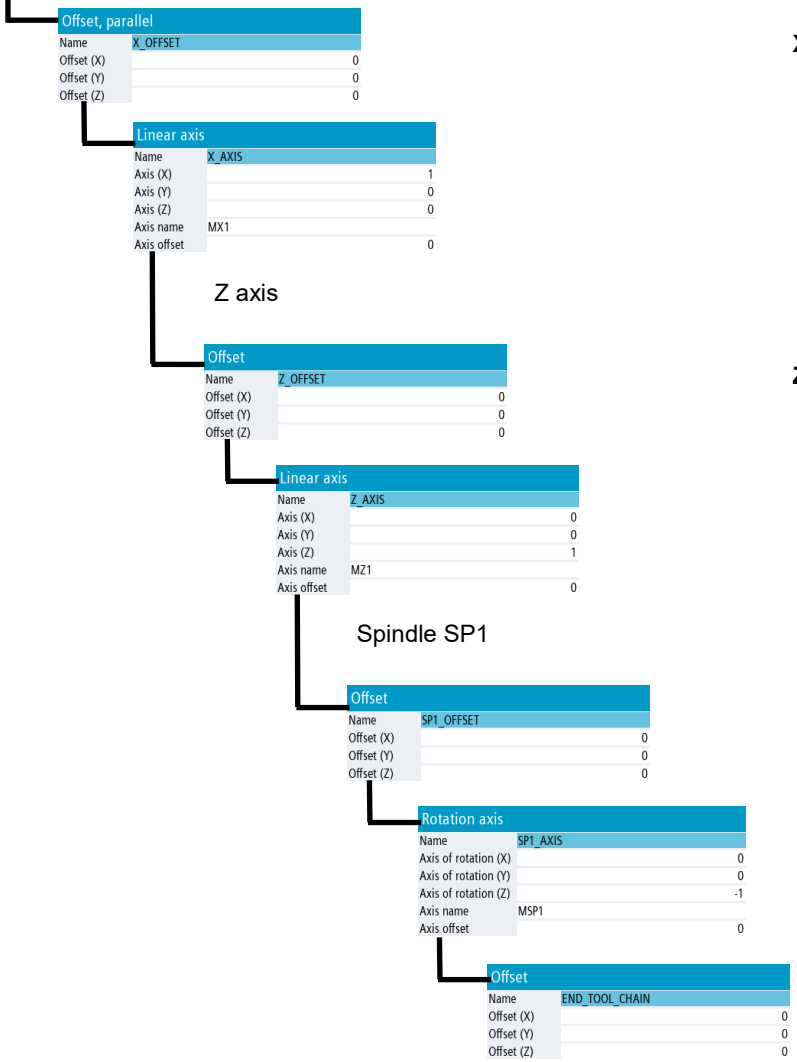
Notes:

Create a tool chain

ROOT element

Next parallel element

X axis



X_OFFSET The offset vector is defined with (0,0,0)

X_AXIS The linear axis references machine axis "MX1" and has vector (1,0,0)

Z_OFFSET The offset vector is defined with (0,0,0)

Z_AXIS The linear axis references machine axis "MZ1" and has vector (0,0,1)

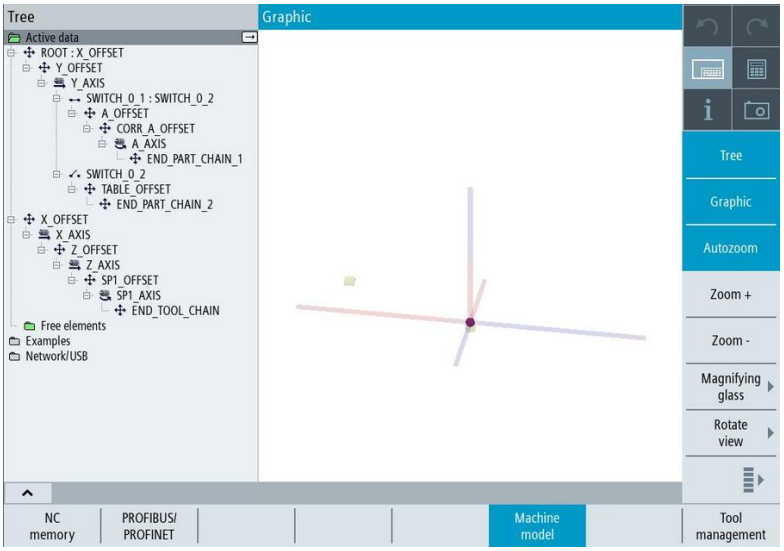
SP1_OFFSET The offset vector is defined with (0,0,0)

SP1_AXIS The rotation axis references spindle “MSP1” has vector (0,0,-1); the direction -1 in the Z component is considered for turning tools or the “L sylus” tool type

END_TOOL_CHAIN Because no offset is used currently in the complete tool chain, the offset values can remain at the value (0,0,0)

Notes:

View of the completed structure tree



Updating the display on the controller

After the kinematic chain have been created a reboot is required for the display on the controller. This can be achieved either by means of the Softkey "Reboot HMI“ or by switching the machine OFF and back ON again.



Position the machine at MCS-0, e.g. via MDA, the axes are displayed in the initial position.

T0 M6
G0 SUPA X0 Y0 Z0
M30

Actuate the SWITCH[0], e.g. via MDA

$NK_SWITCH[0] =1
M30

The software limit switches have the values:

X axis
MD36100 $MA_POS_LIMIT_MINUS[MX1] = -750
MD36110 $MA_POS_LIMIT_PLUS[MX1] = 750

Y axis
MD36100 $MA_POS_LIMIT_MINUS[MY1] = -300
MD36110 $MA_POS_LIMIT_PLUS[MY1] = 300

Z axis
MD36100 $MA_POS_LIMIT_MINUS[MZ1] = 0
MD36110 $MA_POS_LIMIT_PLUS[MZ1] = 650

Notes:

Parameter MD52207 must be set for all kinematics.

MD52207[x] $MCS_AXIS_USAGE_ATTRIB	Attributes of the axes
	Change to:

The following settings are necessary for the rotary axes A:

MD52207[4] A axis
Bit 0 = 1: Rotates around the 1st geometry axis
Bit 6 = 1: Show rotary axis as offset target for measuring
Bit 7 = 1: Offer rotary axis in position pattern
Bit 8 = 1: Offer rotary axis for blank clamping

Description:

Bit 0: Rotates around the 1st geo axis (for rotary axes)
Bit 1: Rotates around the 2nd geo axis (for rotary axes)
Bit 2: Rotates around the 3rd geo axis (for rotary axes)
Bit 6: Show rotary axis as offset target for measuring
Bit 7: Offer rotary axis in position pattern
The orientation of the axis is defined using bits 0 - 2.
If one of the bits 10 - 12 is set, these bits specify the orientation of the axis for position patterns.
Bit 8: Offer rotary axis for blank clamping (on milling machines)
The orientation of the axis is defined using bits 0 - 2.
Bit 10: Rotary axis rotates around 1st geometry axis (only for position pattern)
Bit 11: Rotary axis rotates around 2nd geometry axis (only for position pattern)
Bit 12: Rotary axis rotates around 3rd geometry axis (only for position pattern)

Note:

For bits 0 - 2 and 10 - 12, only the 1st set bit is active in each case.

Note on position pattern with rotary axes:

If one of the bits 10 - 12 is set, then these bits take precedence over bits 0 - 2.

For the programming of the A Axis the following MD must be taken into account:

MD52214 $MCS_FUNCTION_MASK_MILL	Function mask Milling
	Change to: Bit 1 = 1

Description:

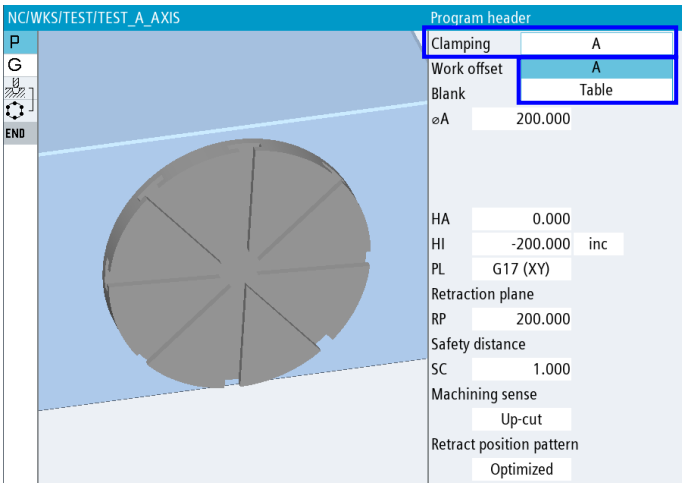
Bit 1: Offer fixed table for blank clamping (at milling machines)

Note:

With the setting Bit 1=1 in the Program header of a NC program the selection field “Clamping“ is displayed.

Thereby, beside the rotary axis, also the table can be selected for the blank clamping.

Select Clamping in the Program header



Notes:

Program example for the A axis

The following describes a programming example which demonstrates a subsequent machining operation with a rotary axis A.

Select tools

Tool list — Spindle

Loc.	MT LO.	Type	Tool name	ST	D	Length	⌀			1	2
1			FACING_30	1	1	100.000	30.000		5		
2			CUTTER_25	1	1	100.000	25.000		3		
3			CUTTER_10	1	1	100.000	10.000		2		
4			CENTERDRILL_20	1	1	100.000	20.000	90.0			
5			TWIST_DRILL_10	1	1	100.000	10.000	118.0			
6			TAP_M12	1	1	0.000	12.000	180.0			
7											

Set work offsets

Work offset - G54 ... G57 [mm]

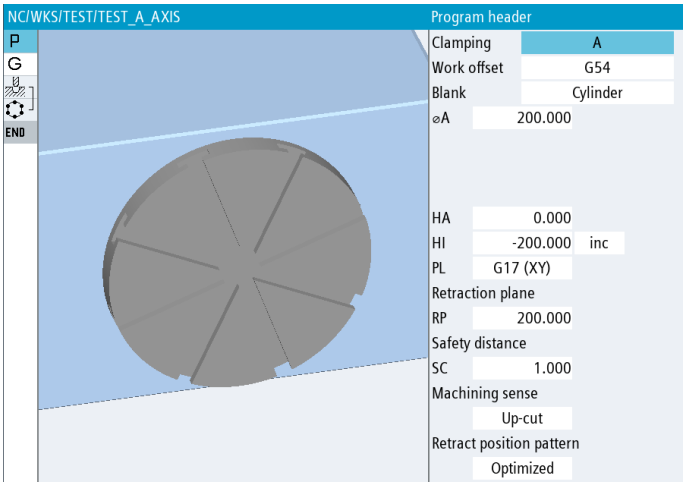
	X	Y	Z	A	SP1
G54	-200.000	0.000	150.000	0.000	0.000
Fine	0.000	0.000	0.000	0.000	0.000

Program example in the editor

NC/WKS/TEST/TEST_A_AXIS 1

P	Program header	A: G54 Cylinder
G	G0 G90 Y0¶	
	Drilling	T=TWIST_DRILL_10 F=50/min S=2000rev Z1=25inc
	001: Posit.circle XA	Z0=100 X0=-20 A0=30 N=8
END	End of program	

Program head with workpiece definition:



Program steps

Drilling and technology data:

Drilling

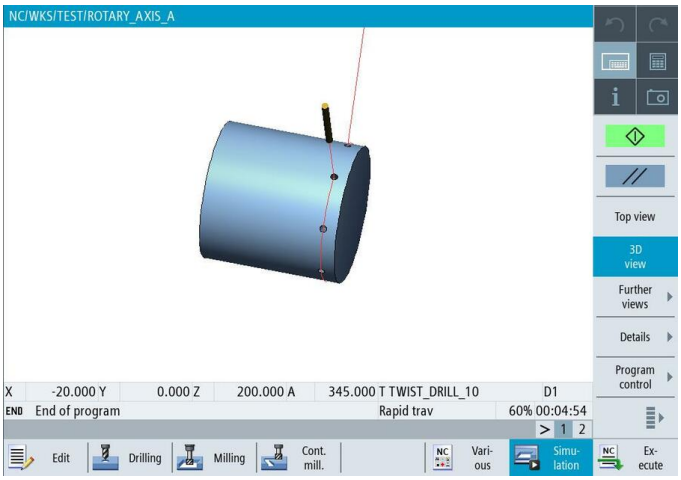
T	TWIST_DRILL_10	D 1
F	50.000	mm/min
S	2000	rpm
	Shank	
Z1	25.000	inc
Spot drill	No	
Thrd. drill	No	
DT	0.600	s

Position circle for drilling:

Position circle

Axes	XA
Z0	100.000
X0	-20.000
A0	30.000 °
N	8

Program example in the simulation



Notes:

TRACYL with the A axis

For creating the transformation on the HMI user interface, an HMI tool is available at the “Commissioning” area.

All required steps can be performed with this tool.

Software option

The following software option is required to use this function:

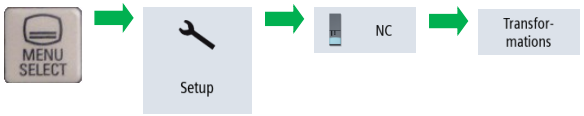
"TRANSMIT and peripheral surface transformation"
6FC5800-0BM27-0YB0

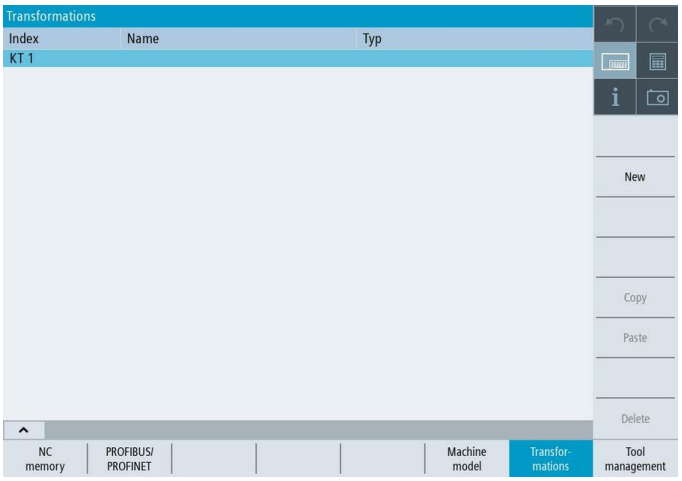
The peripheral surface transformation (TRACYL) function machines the peripheral surface of a workpiece.

Machine data settings

Maximum number of transformations that can be defined by kinematic chains
MD18866 $MN_MM_NUM_KIN_TRAFOS = 1

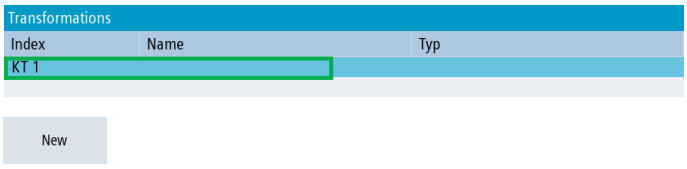
Start the configuration



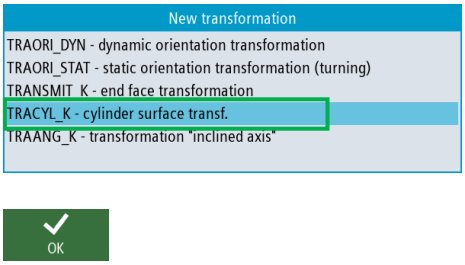


Create a new transformation

Position the cursor on the "KT 1" line and press the "New" softkey.

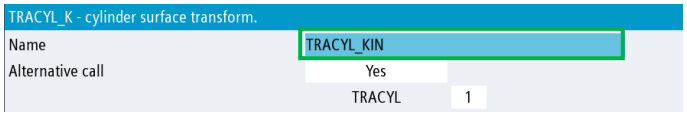


Select the TRACYL_K transformation from the input dialog and confirm with the "OK" softkey.



Specify the name for the transformation

TRACYL_KIN is specified as name for the transformation, the selection field for the alternative call is changed to “Yes” and the associated TRACYL selection block is set to 1.



The part program can activate or deactivate the new transformation with the TRACYL or TRAFOOF command respectively.

If no alternative call should be used, the transformation can be activated later with the TRAFOON(“TRACYL_KIN“) command.

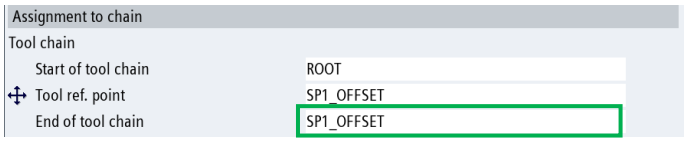
Notes:

Parameterization of the assignment to chain

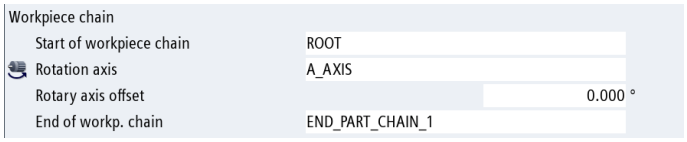
All assignments to the chain are made via the “Select element“ softkey.



SP1_OFFSET is selected for the tool chain as last element from the kinematic chain.



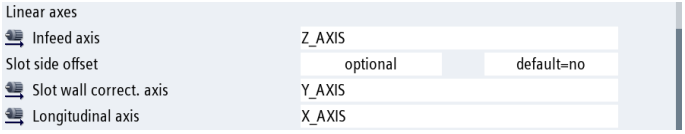
For the workpiece chain, the A_AXIS element is selected for the “Rotation axis” selection and the END_PART_CHAIN_1 element is selected for the end of the workpiece chain.



For the infeed axis, the Z_AXIS element is selected (=the axis that is positioned perpendicular to the rotation axis of the cylinder).

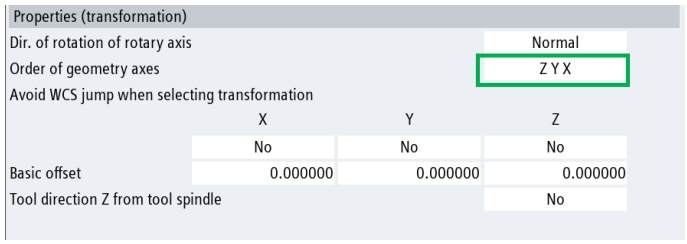
The “Slot side offset“ selection is switched to “no“. For the slot wall correction axis the Y_AXIS element is specified.

For the longitudinal axis the X_AXIS element is specified. (=the axis that runs parallel to the rotation axis of the cylinder).



Parameterization of the transformation properties

- The direction of rotation of the rotary axis should not be inverted.
- The order of the geometry axes should be “ZYX“.
- A WCS jump of the actual value should not be suppressed for any axis when the transformation is selected.
- The base offsets are initialized with value 0.
- The tool direction Z should not be specified explicitly (because already specified for this machine).



Confirm all transformation data with the "OK" softkey.

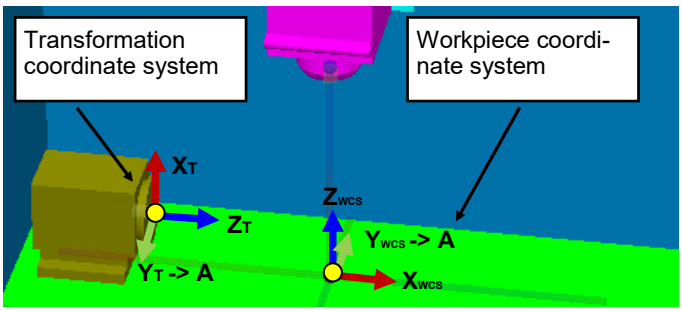


It must be considered that a coordinate system is created internally in the controller (transformation coordinate system) whose Z axis coincides with the rotation axis of the cylinder and whose X axis coincides with the infeed axis. The Y axis of this coordinate system is simulated for the active TRACYL by the rotary axis.

The sequence of the geometry axes is subject of the references:

Which axis of the transformation coordinate system should be moved if
- X is programmed -> Z
- Y is programmed -> Y (Y is mapped to the A axis)
- Z is programmed -> X

The setting to be selected is ZYX.



Notes:

For the programming of the TRACYL transformation in ShopMill the following MD must be taken into account:

MD52214 $MCS_FUNCTION_MASK_MILL	Function mask Milling
	Change to: Bit 0 = 1

Description:

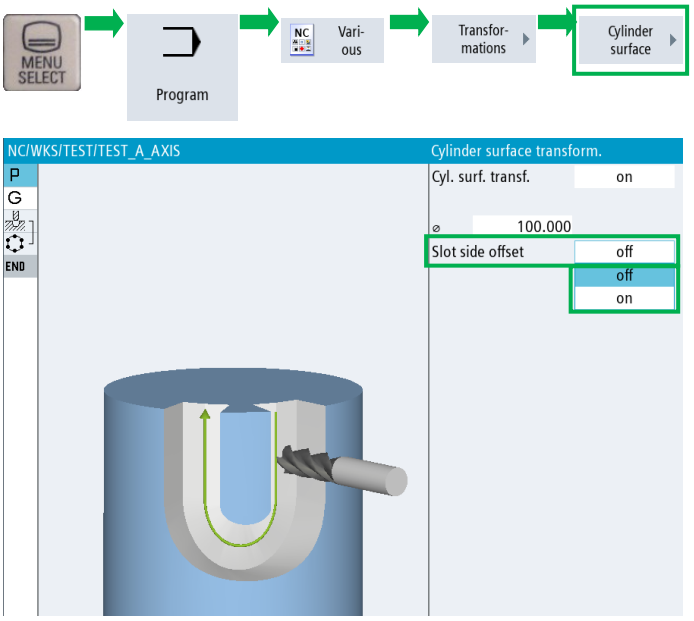
Bit 0: Enable cylinder surface transformation (ShopMill)

Note:

With the setting Bit 0=1 in the ShopMill editor under “Various“ -> “Transformations“ the softkey “Cylinder surface“ is displayed.

Thereby in a ShopMill program the cylinder surface transformation can switched on and off via program step.

Switch on Cylinder surface transformation in ShopMill editor



Notes:

Program example for TRACYL

The following describes a programming example which demonstrates a subsequent machining operation with TRACYL.

Select tools

Tool list — Spindle

Loc.	MT LO.	Type	Tool name	ST	D	Length	⌀			1	2
1			FACING_30	1	1	100.000	30.000		5		
2			CUTTER_25	1	1	100.000	25.000		3		
3			CUTTER_10	1	1	100.000	10.000		2		
4			CENTERDRILL_20	1	1	100.000	20.000	90.0			
5			TWIST_DRILL_10	1	1	100.000	10.000	118.0			
6			TAP_M12	1	1	0.000	12.000	180.0			

Set work offsets

Work offset - G54 ... G57 [mm]

	X	Y	Z	A	SP1
G54	-200.000	0.000	150.000	0.000	0.000
Fine	0.000	0.000	0.000	0.000	0.000

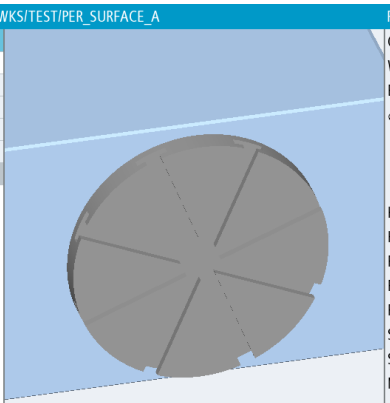
Program example in the editor

NC/WKS/TEST/PER_SURFACE_A

P	Program header	A: G54 Cylinder
G	G0 G90 Y0¶	
	Cylinder surface	on ⌀200
	Drilling	T=TWIST_DRILL_10 F=50/min S=2000rev Z1=25inc
	001: Posit.circle XA	Z0=100 X0=-20 A0=30 N=8
	Cylinder surface	off
END	End of program	

Program head with workpiece definition:

NC/WKS/TEST/PER_SURFACE_A



Program header

Clamping	A
Work offset	G54
Blank	Cylinder
⌀A	200.000
HA	0.000
HI	-200.000 inc
PL	G17 (XY)
Retraction plane	
RP	200.000
Safety distance	
SC	1.000
Machining sense	
	Up-cut
Retract position pattern	
	Optimized

Program steps

Cylinder surface transformation ON:

Cylinder surface transform.

Cyl. surf. transf.	on
⌀	200.000
Slot side offset	off

Drilling and technology data:

Drilling

T	TWIST_DRILL_10	D 1
F	50.000	mm/min
S	2000	rpm
	Shank	
Z1	25.000	inc
Spot drill	No	
Thrd. drill	No	
DT	0.600	s

Position circle for drilling:

Position circle

Axes	XA
Z0	100.000
X0	-20.000
A0	30.000 °
N	8

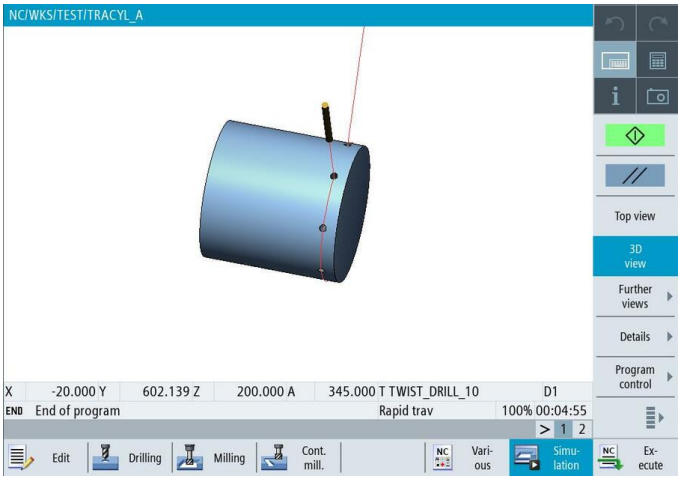
Cylinder surface transformation OFF:

Cylinder surface transform.

Cyl. surf. transf.	off

Program example in the simulation

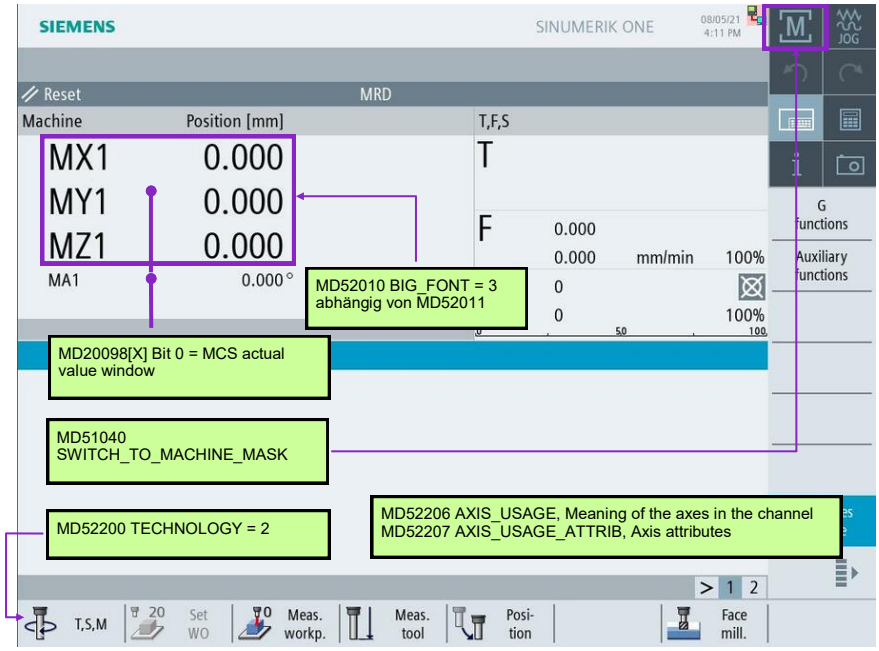
NC/WKS/TEST/TRACYL_A

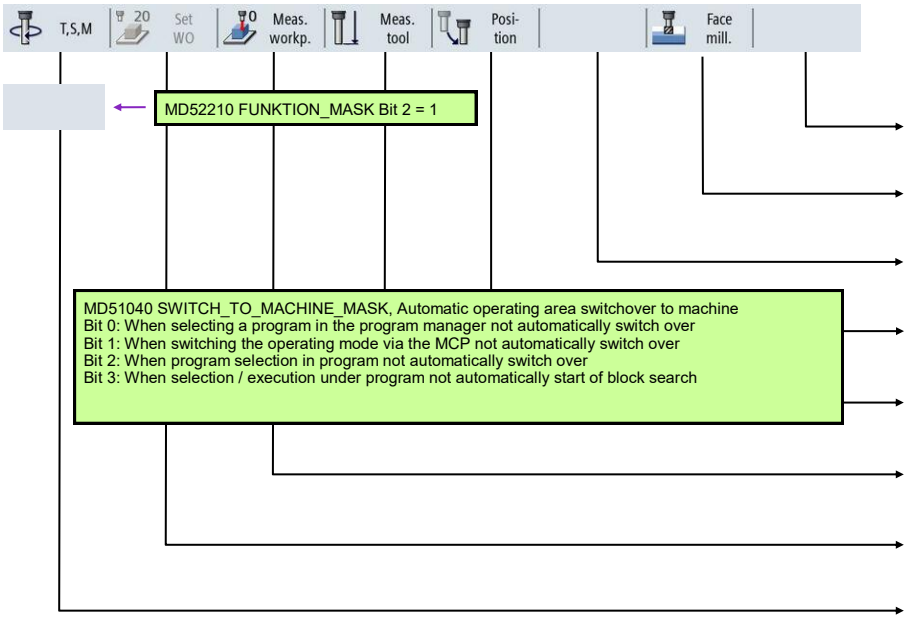


Notes:

Menu and Technology settings

Machine —> JOG

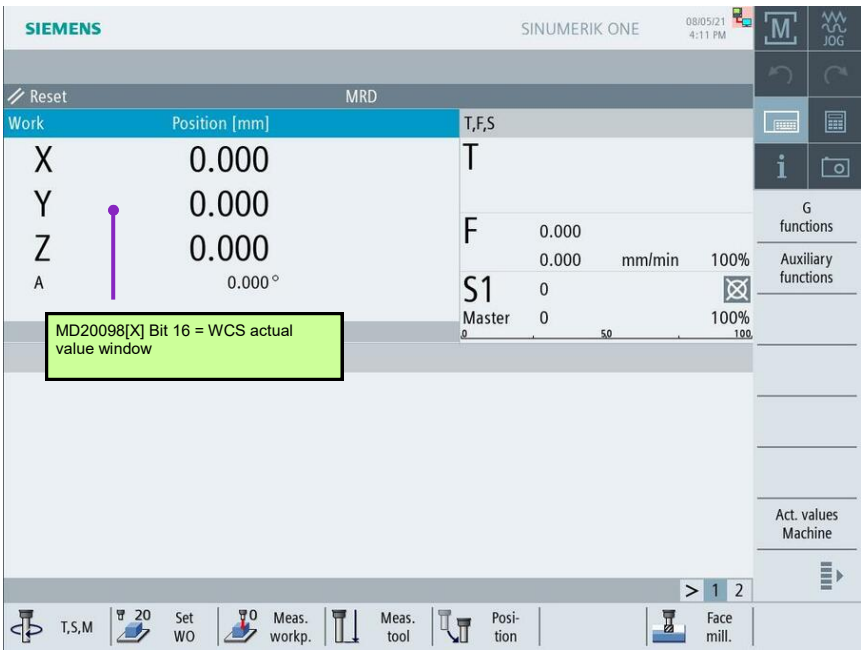




Softkey selection	Description
MD52200 = 2	=2: Technology Milling
MD52200 = 2	=2: Technology Milling
MD52200 = 2	=2: Technology Milling
MD52210 Bit 2 = 0	Bit 2 = 0 Show Softkey “T,S,M“ in JOG area

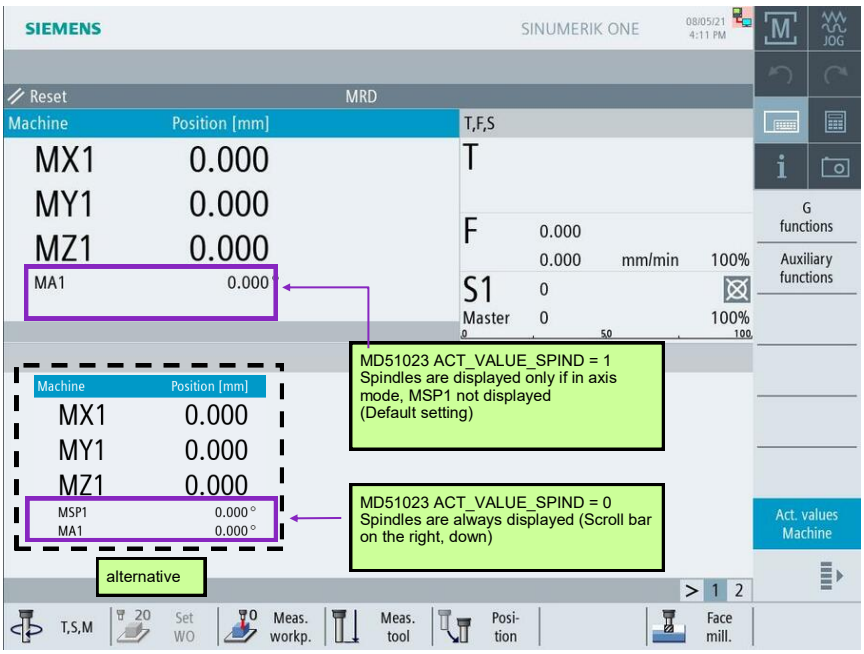
Notes:

Machine —> JOG —> WCS



Softkey selection	Description
Switchover MCS/WCS	**Switchover Softkey from MCS to WCS via PLC**

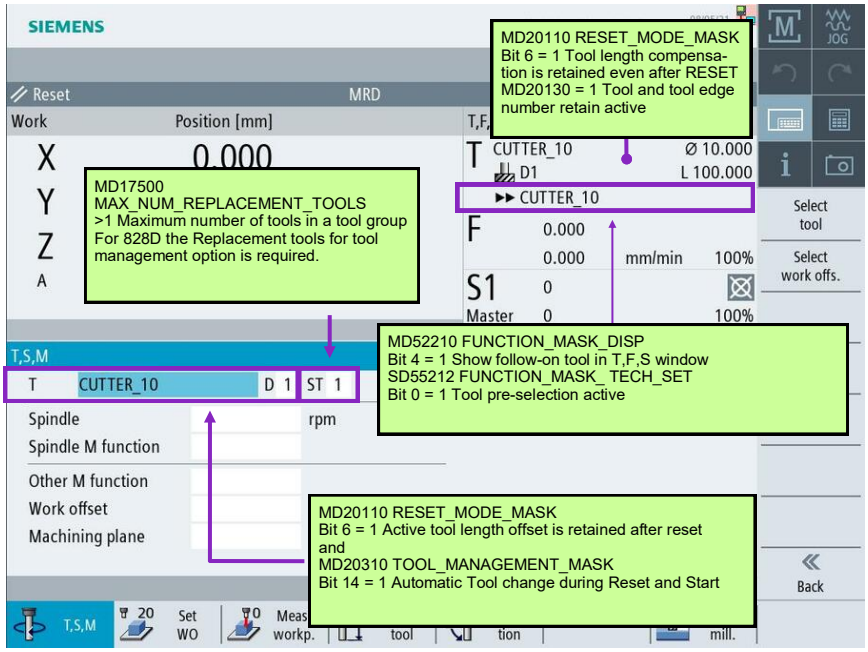
Machine —> JOG —>MCS



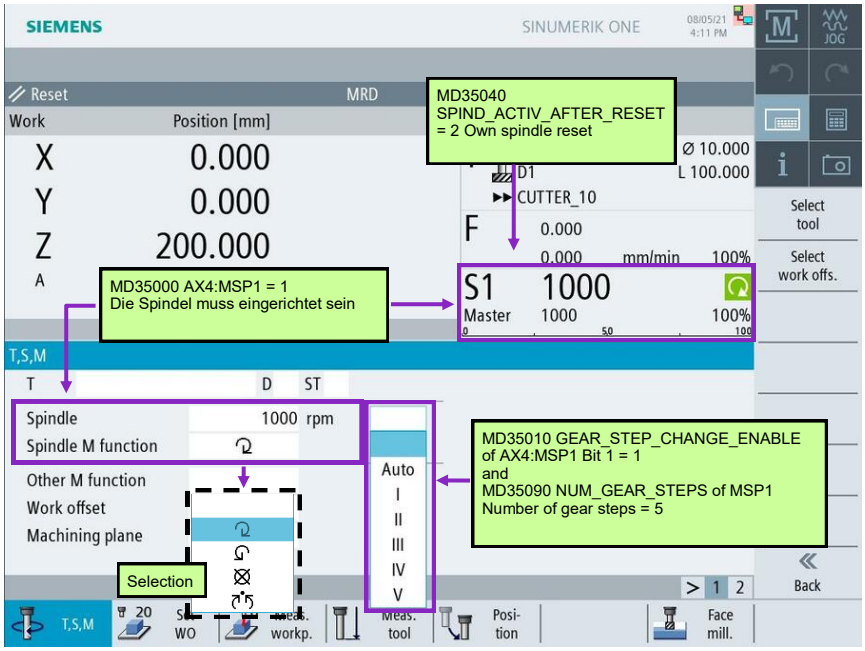
Softkey selection	Description
Switchover MCS/WCS	**Switchover Softkey from MCS to WCS via PLC**

Notes:

Machine —> JOG —> T,S,M

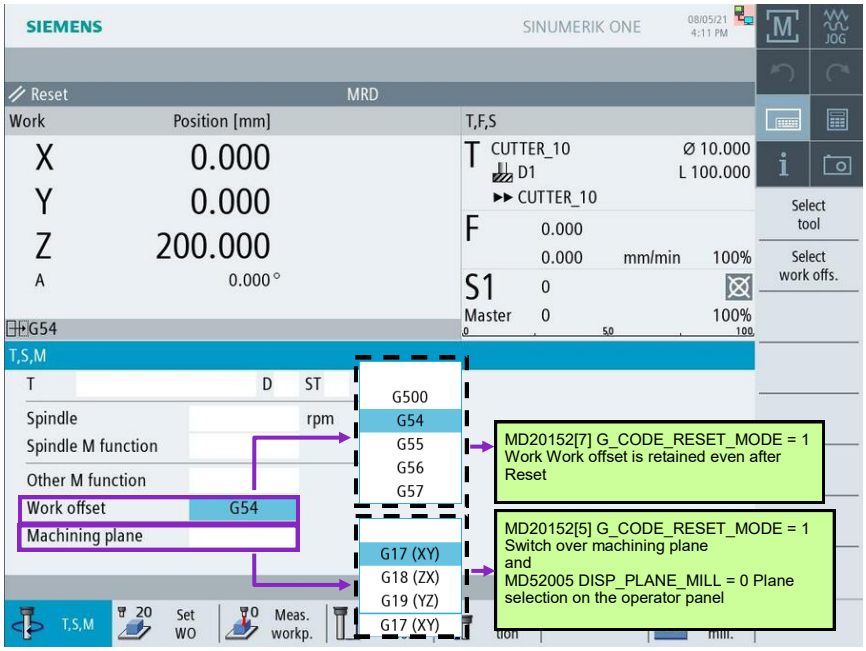


Machine —> JOG —> T,S,M

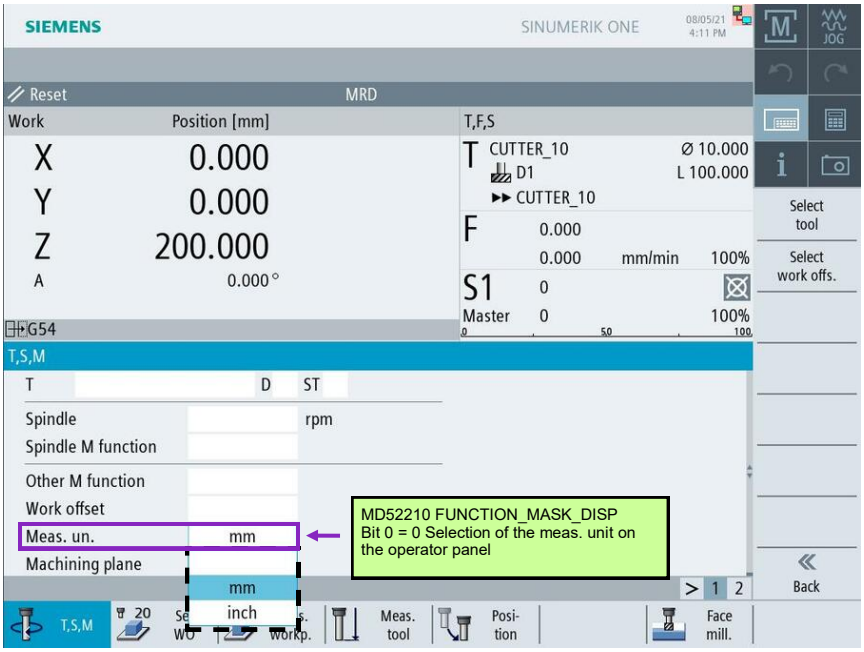


Notes:

Machine —> JOG —> T,S,M

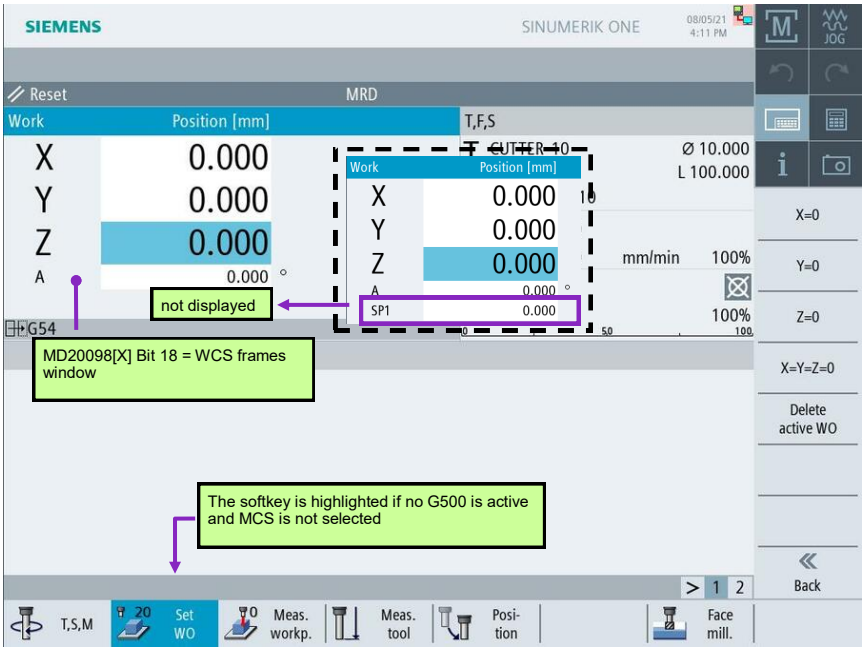


Machine —> JOG —> T,S,M

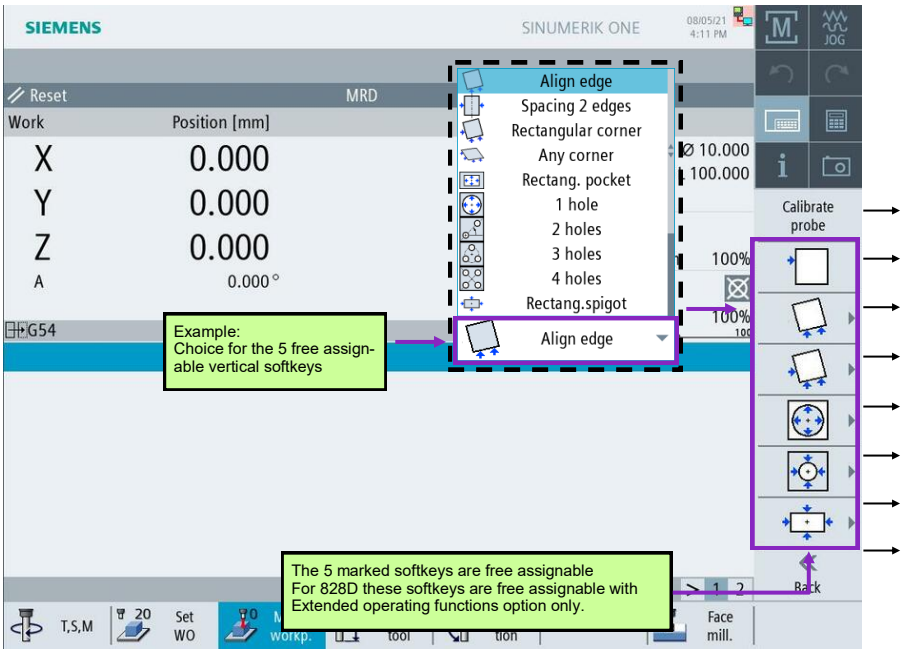


Notes:

Machine —> JOG —> Set WO



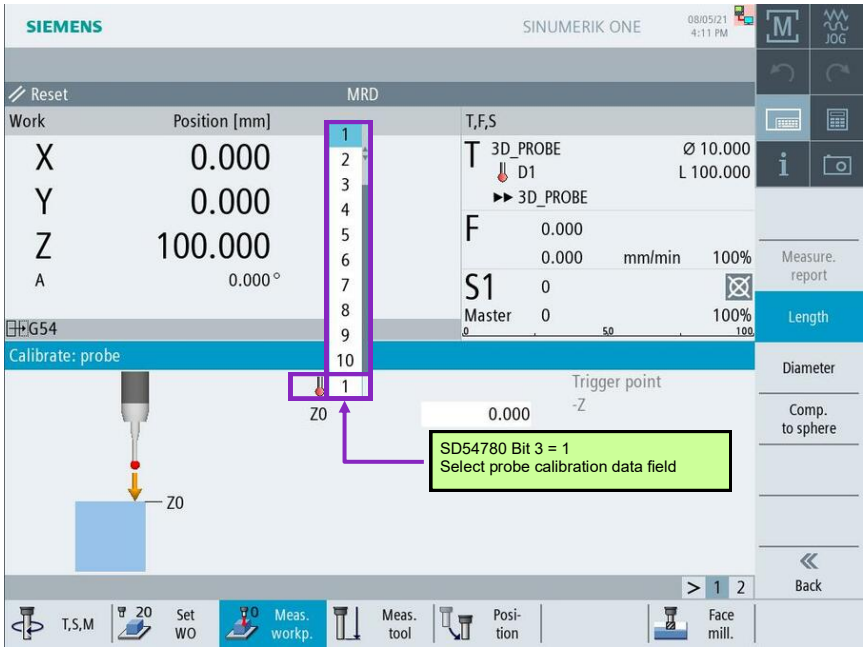
Machine —> JOG —> Measure workpiece



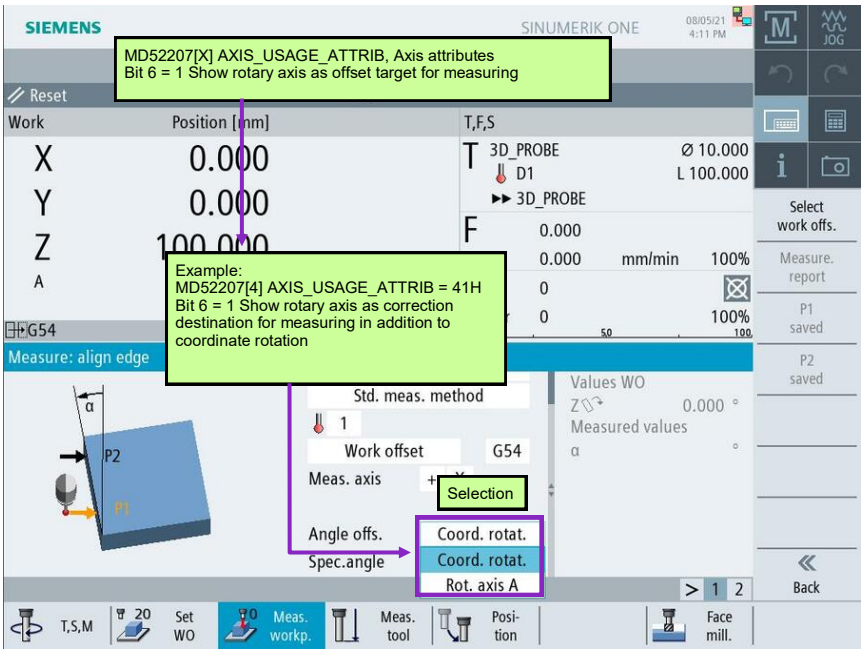
Softkey selection	Description
SD54780 Bit 2 = 1	Bit 2 = 1 Activate calibration for electronic workpiece probe

Notes:

Machine —> JOG —> Measure workpiece

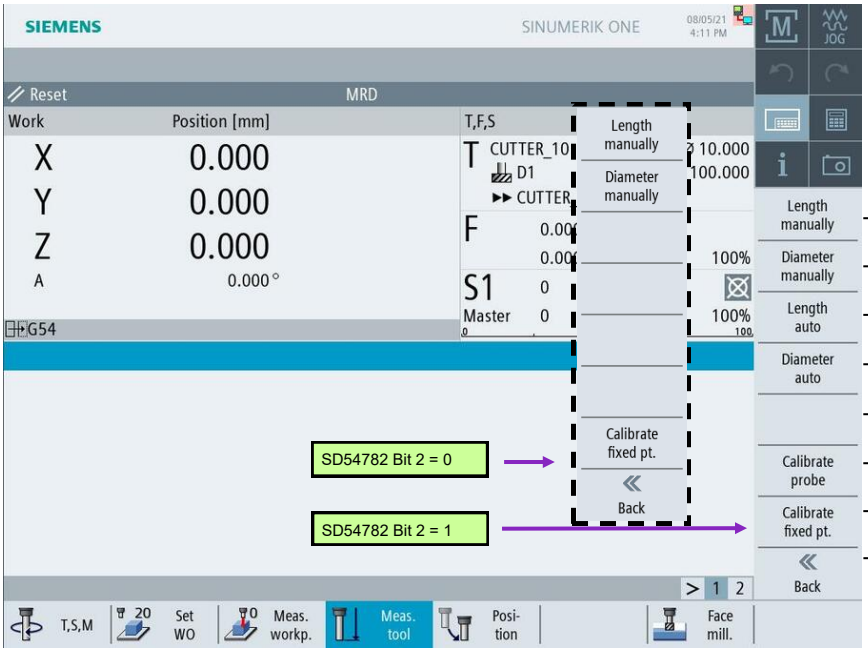


Machine —> JOG —> Measure workpiece



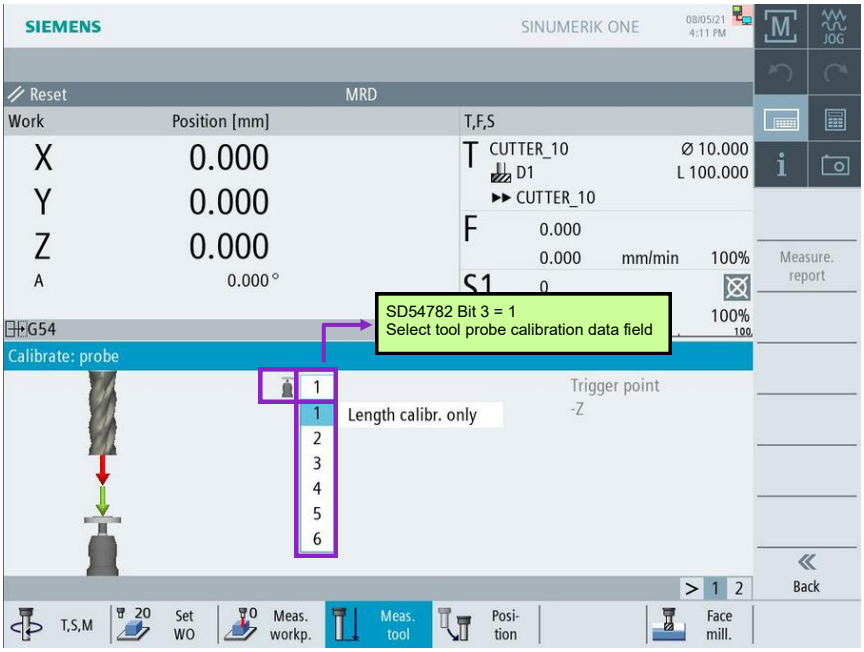
Notes:

Machine —> JOG —> Measure tool



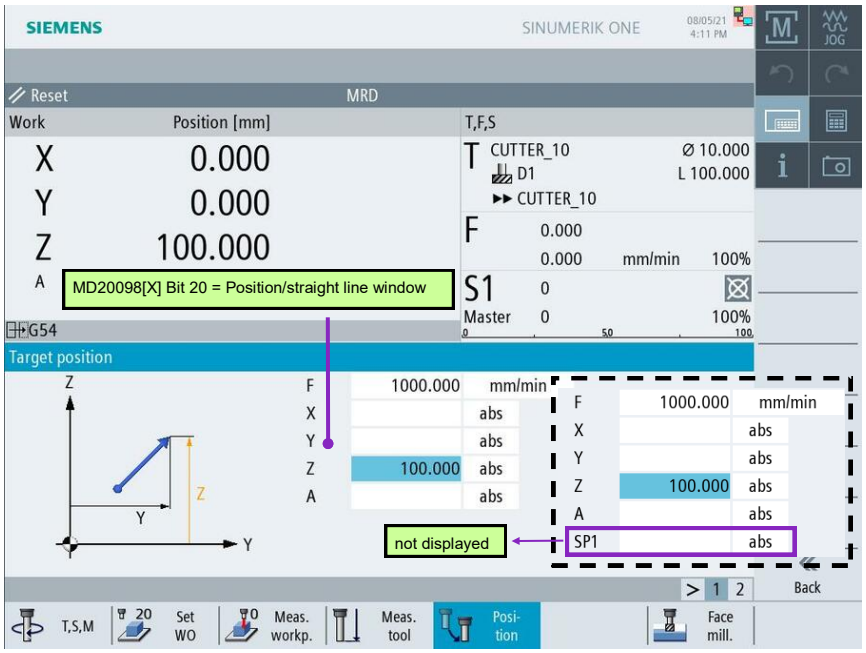
Softkey selection	Description
SD54782	Bit 2 = 1 Enable automatic tool measurement
SD54782	Bit 2 = 1 Enable automatic tool measurement
SD54782	Bit 2 = 1 Enable automatic tool measurement

Machine —> JOG —> Measure tool

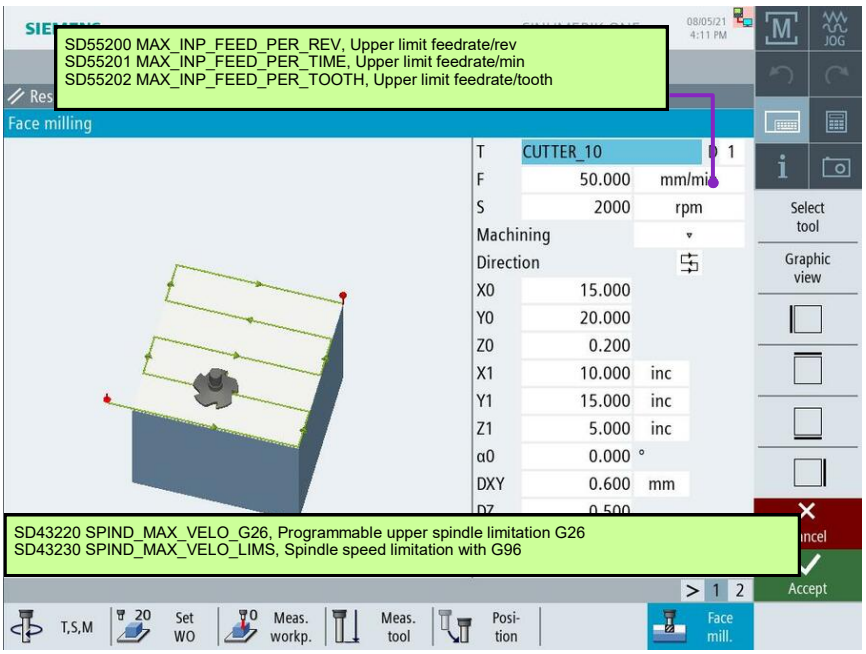


Notes:

Machine —> JOG —> Position

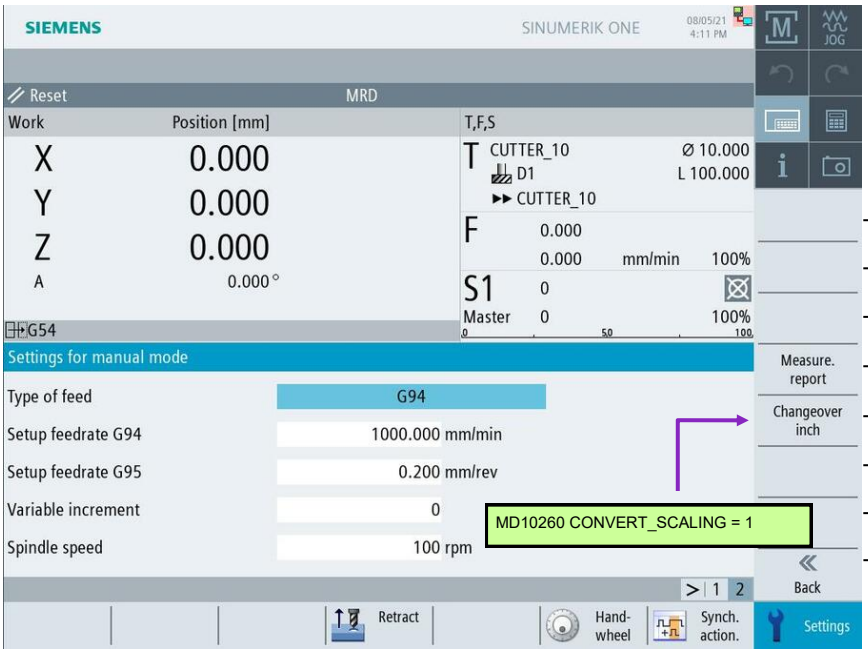


Machine —> JOG —> Face milling

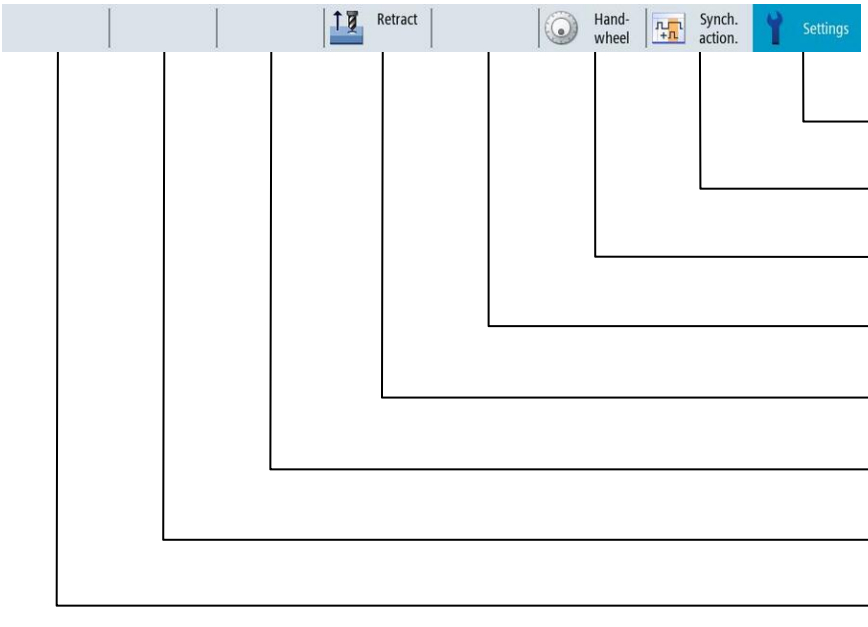


Notes:

Machine —> JOG —> Settings



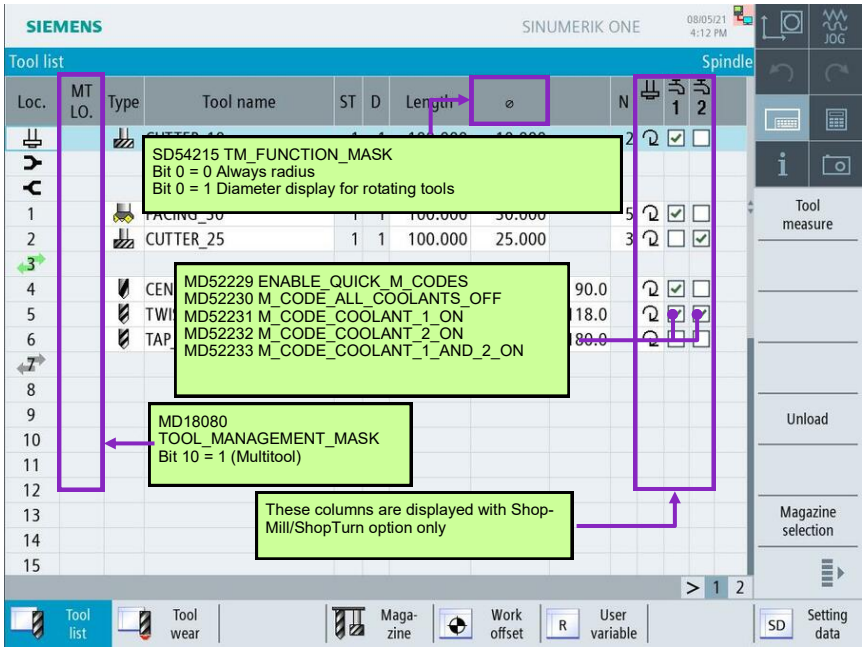
Softkey selection	Description
MD10260 = 1	=1: Enable basic system conversion

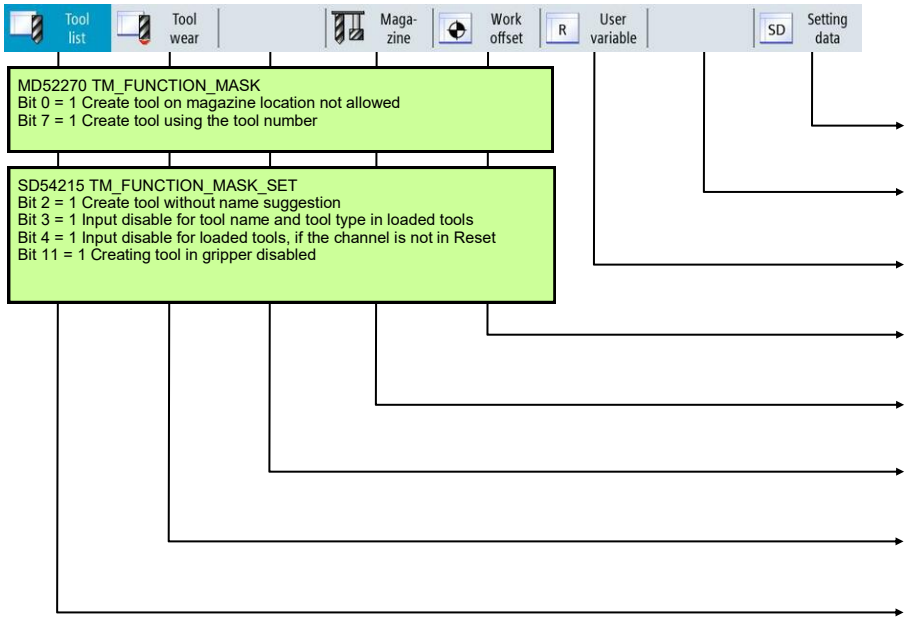


Softkey selection	Description
Option (828D only)	Extended operating functions
MD51067=1	=1: Show handwheel window (Handwheel must be configured in MD11350,11351,11352)

Notes:

Parameter —> Tool list

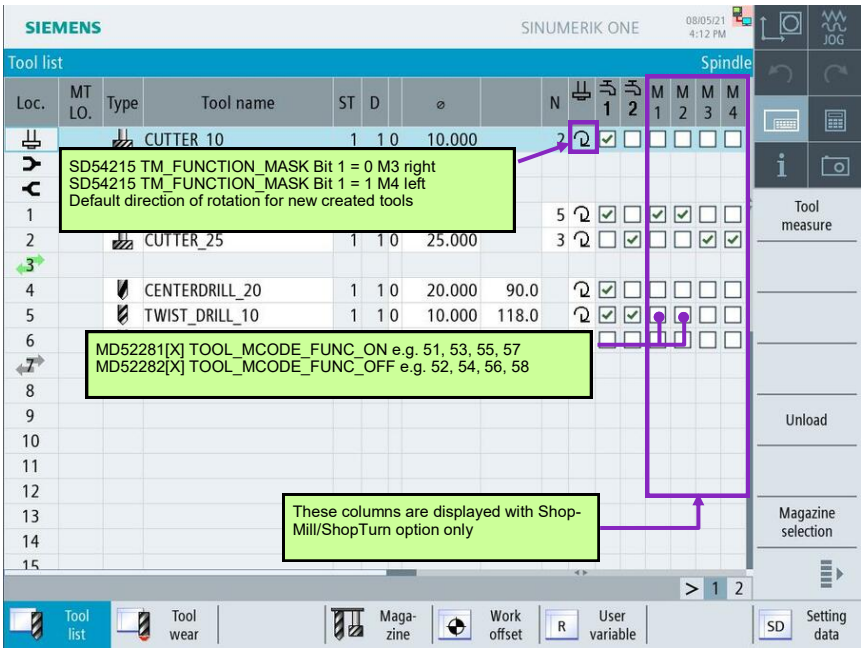




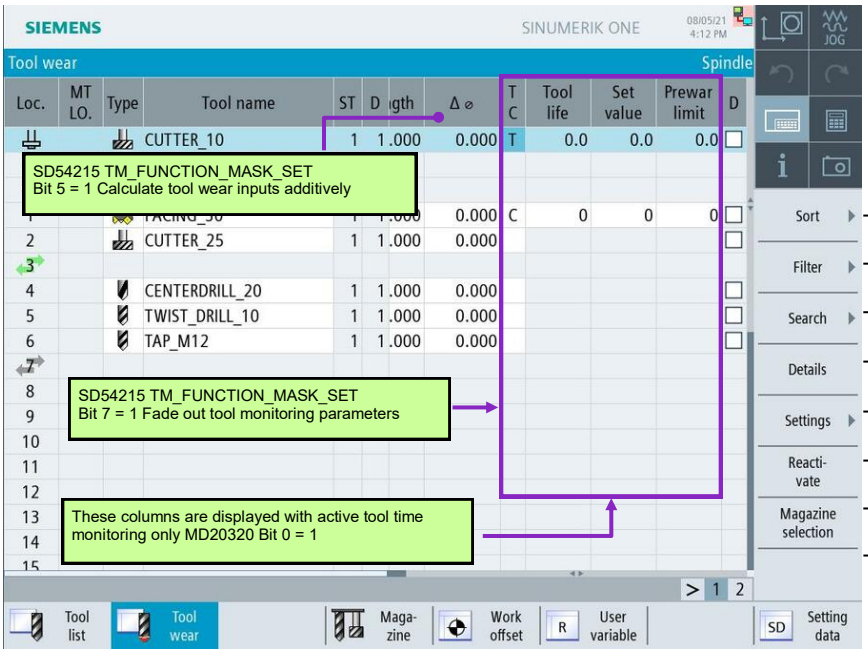
Softkey selection	Description

Notes:

Parameter —> Tool list



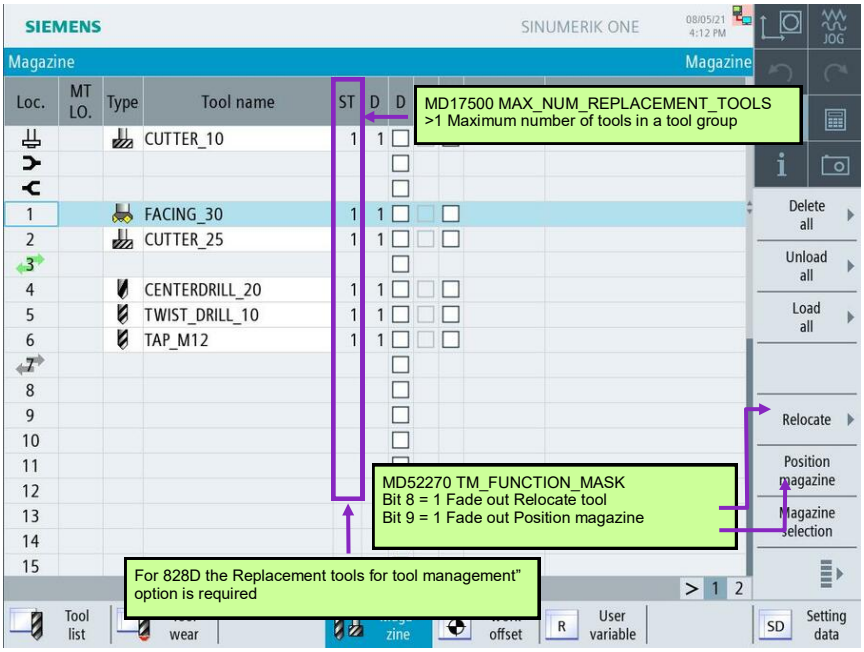
Parameter —> Tool wear



Softkey selection	Description
MD52270	Bit 12 = 0 Enable Reactivate tool

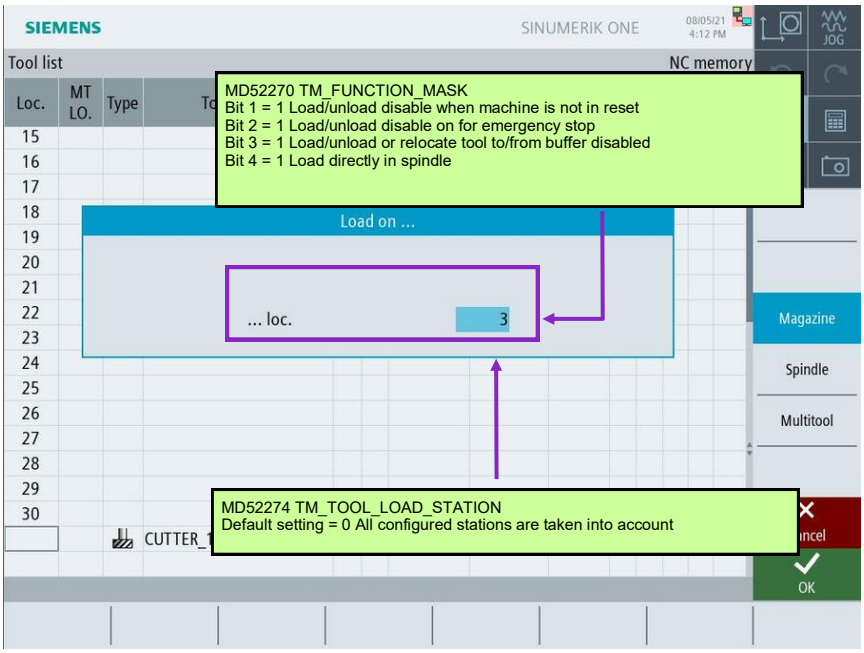
Notes:

Parameter —> Magazine



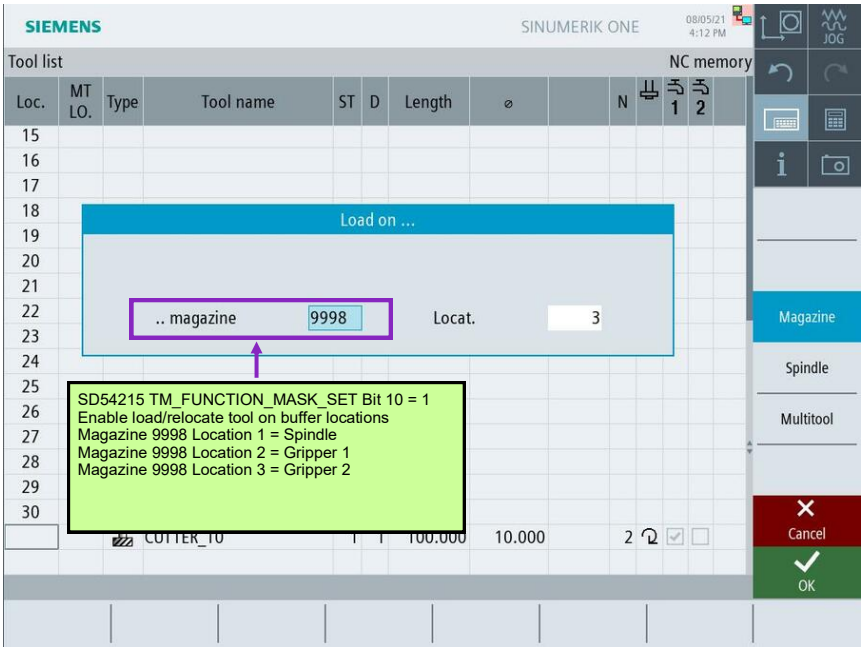
Softkey selection	Description
MD52270	Bit 8 = 0 Enable Relocate tool
MD52270	Bit 9 = 0 Enable Position magazine

Parameter —> Load tool

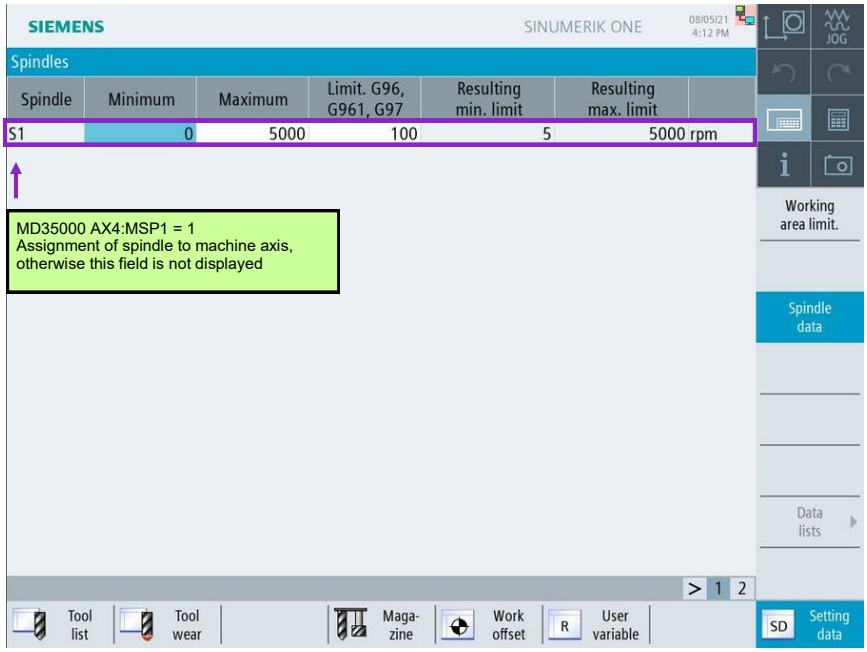


Notes:

Parameter —> Load tool

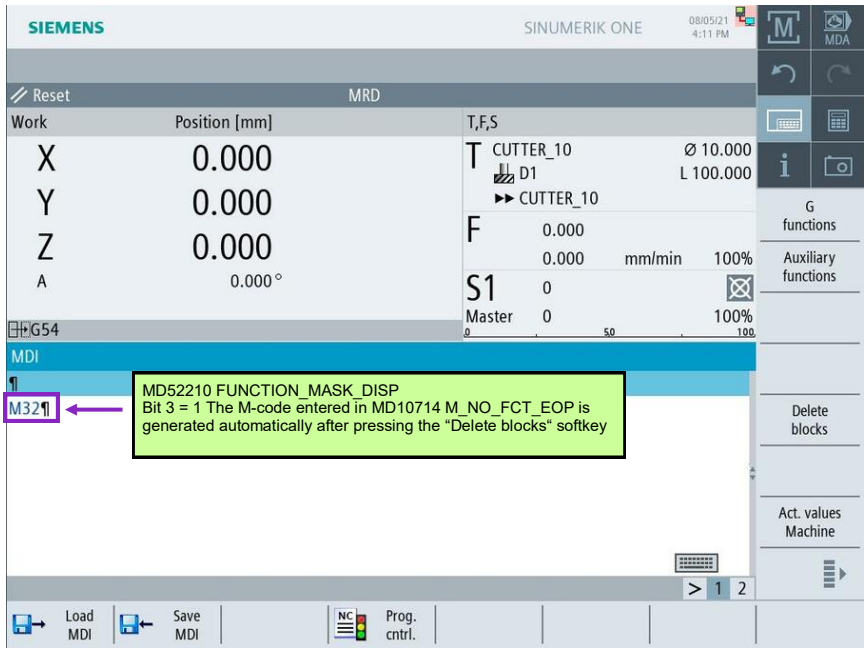


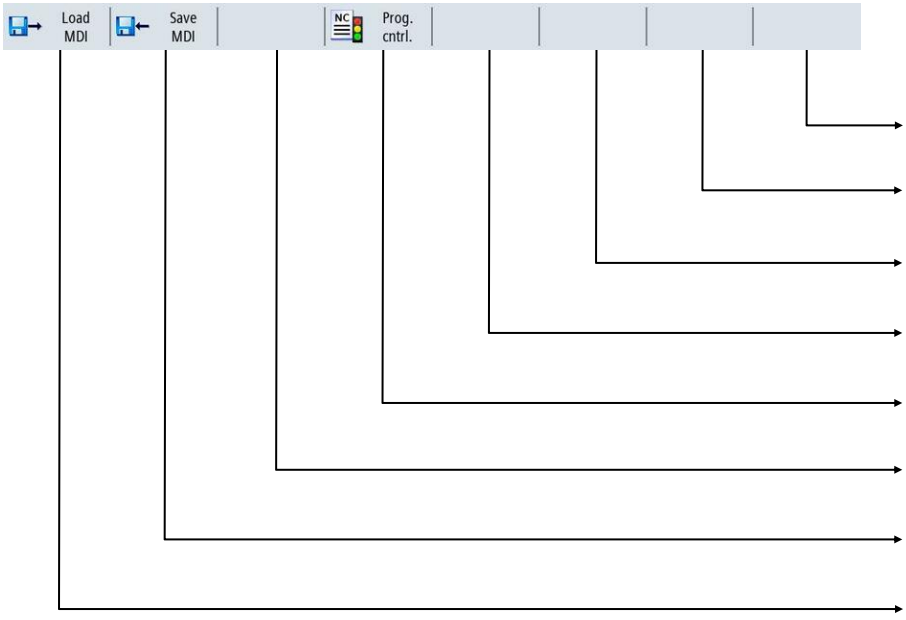
Parameter —> Setting data



Notes:

Machine —> MDA

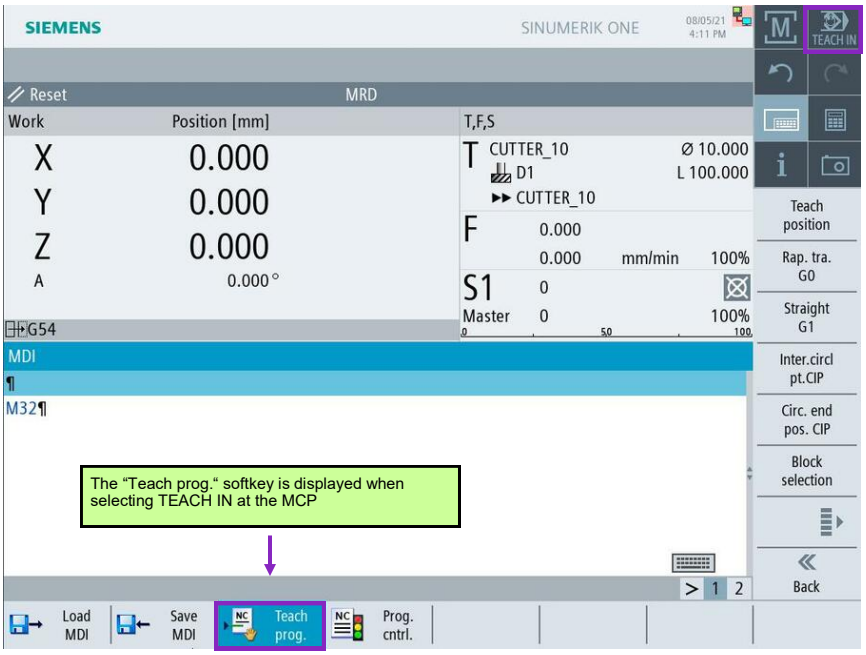


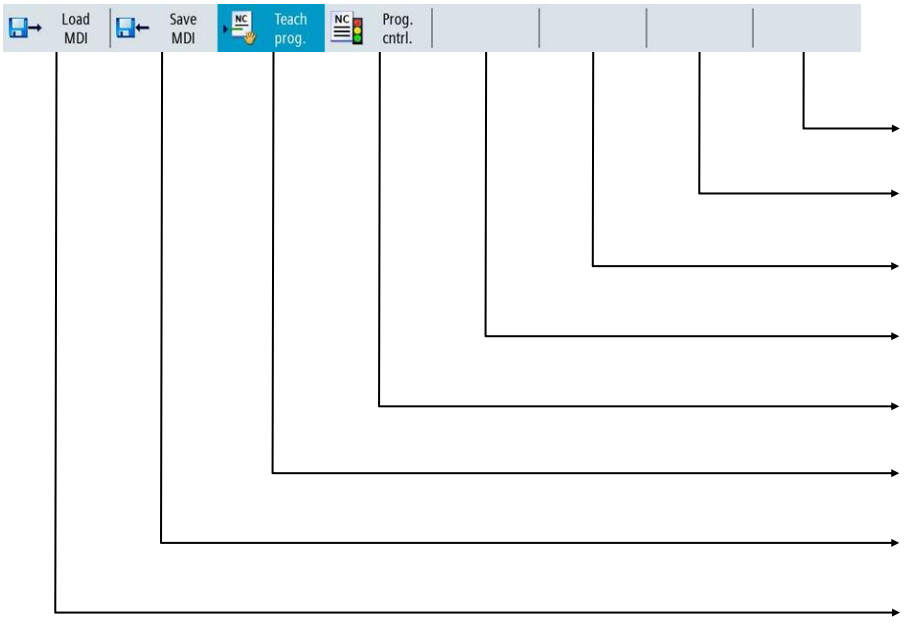


Softkey selection	Description
Option (828D only)	Extended operating functions
Option (828D only)	Extended operating functions

Notes:

Machine —> TEACH IN —> Teach program

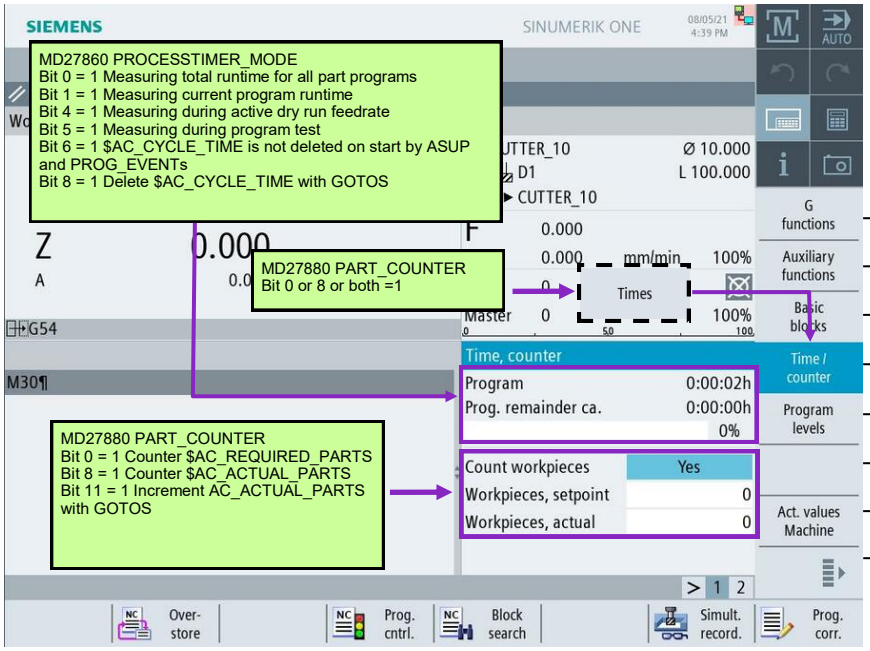




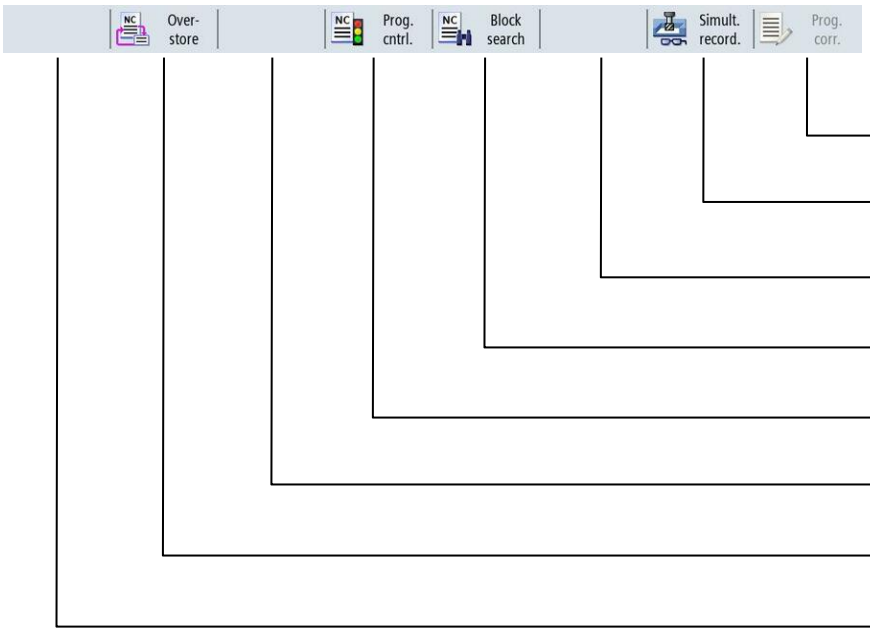
Softkey selection	Description
Option (828D only)	Extended operating functions
Option (828D only)	Extended operating functions
Option (828D only)	Extended operating functions

Notes:

Machine —> AUTO —> Time / Counter



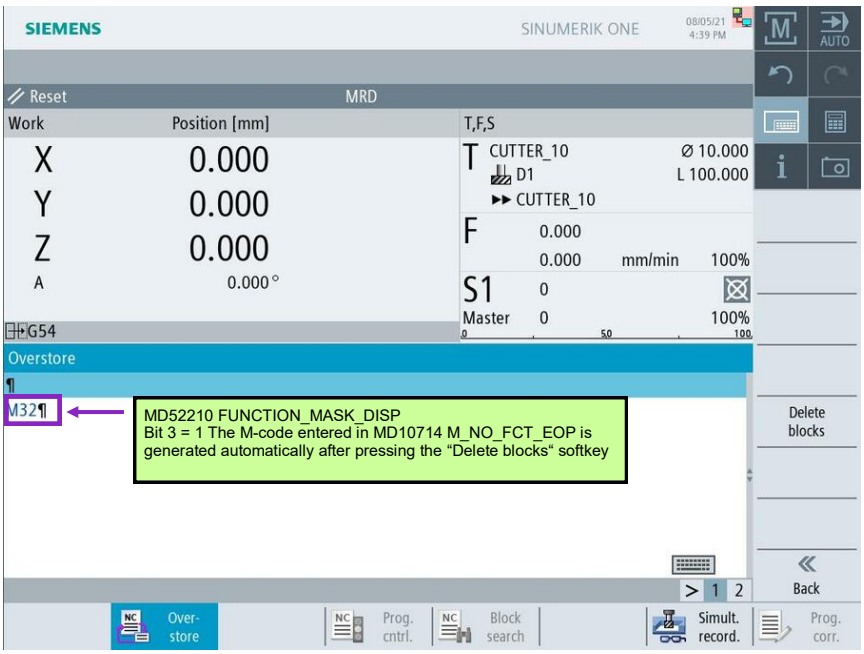
Softkey selection	Description
MD27880 Bit 0 or Bit 8 = 1	**=1: Softkey Time/Counter =0: Softkey Times (Default setting)**



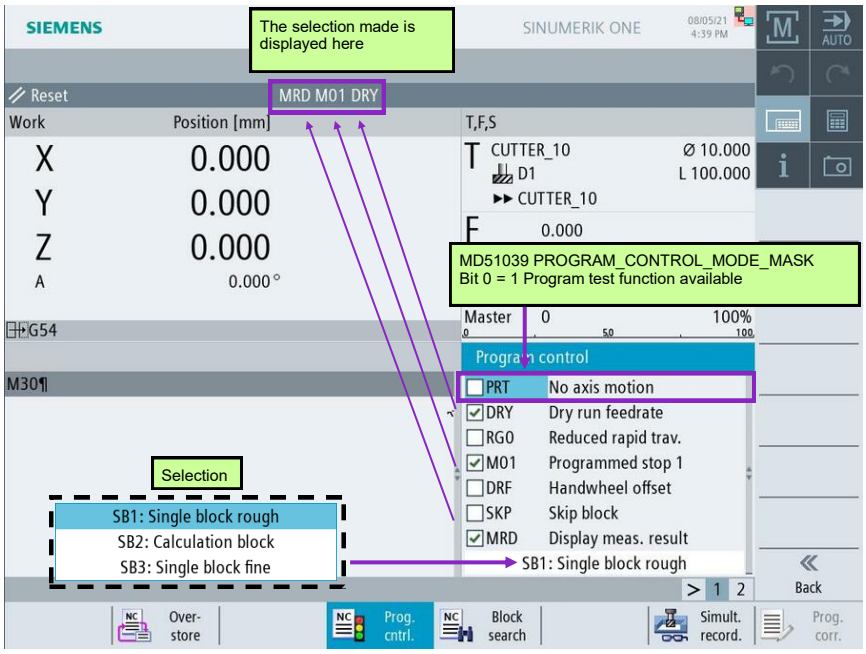
Softkey selection	Description
Option	**Simultan. recording (real-time simulation)**
Option (828D only)	**Extended operating functions**

Notes:

Machine —> AUTO —> Overstore

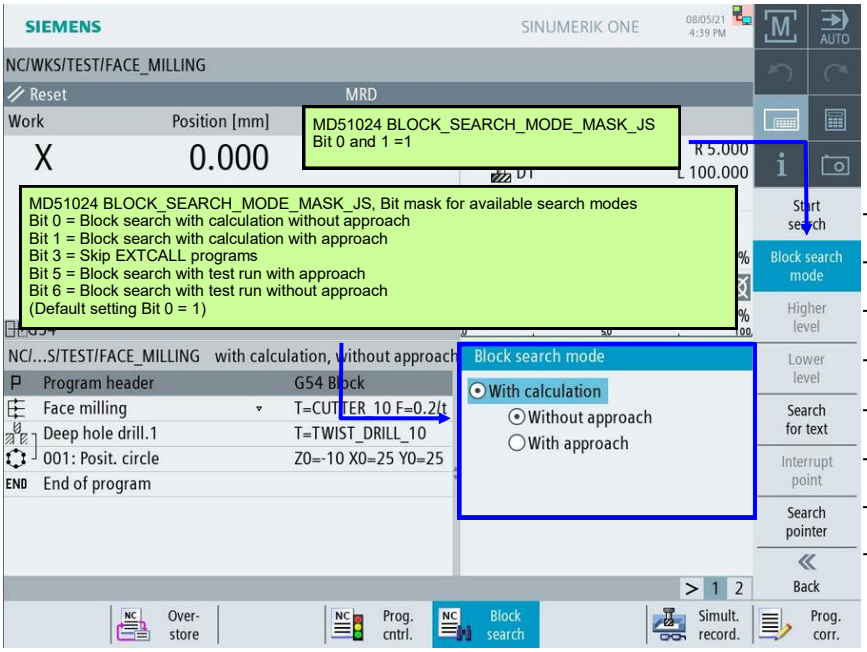


Machine —> AUTO —> Program control



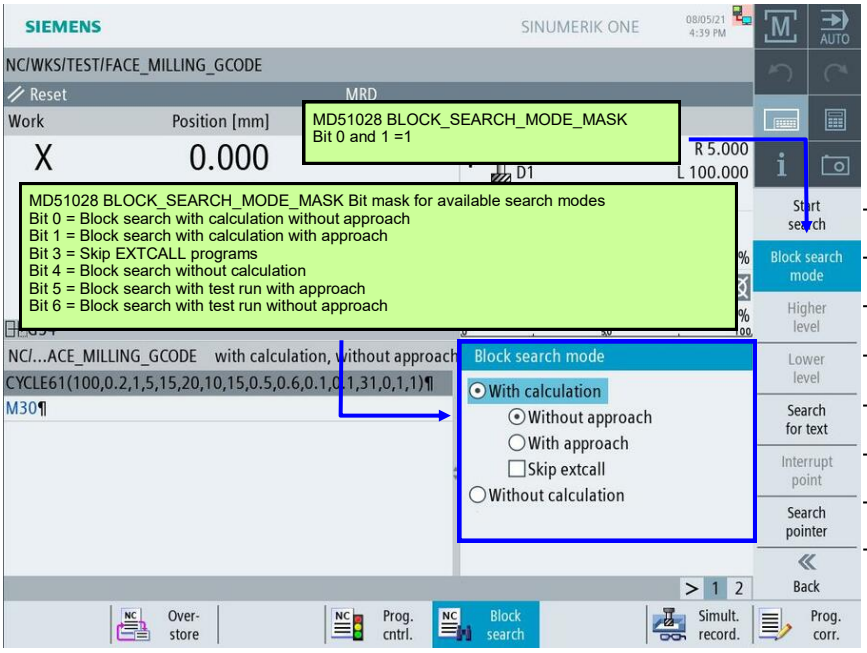
Notes:

Machine —> AUTO —> Block search (ShopMill)



Softkey selection	Description
MD51024 Bit 0 and Bit 1 = 1	Bit mask for available search modes in ShopMill program
Option (828D only)	Extended operating functions

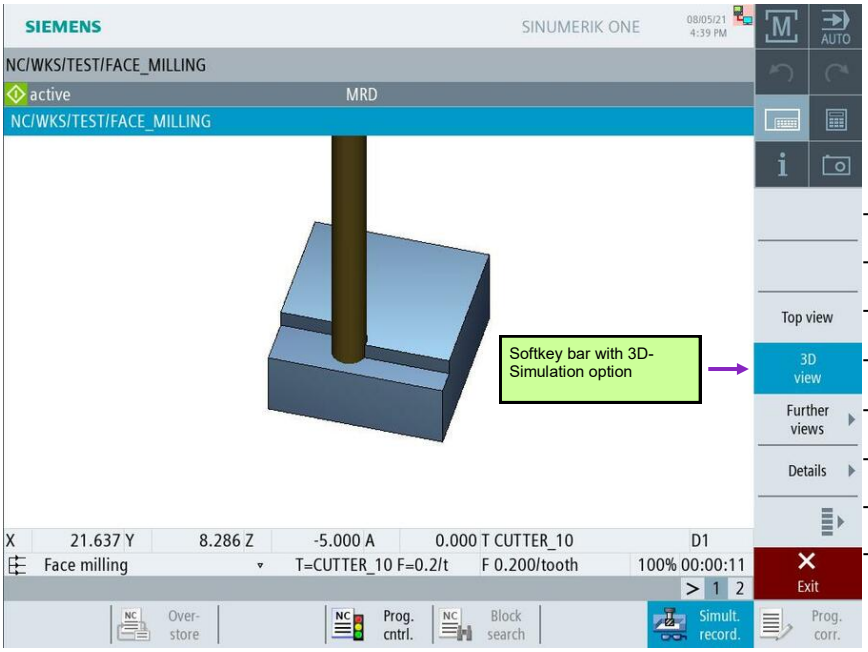
Machine —> AUTO —> Block search (G Code program)



Softkey selection	Description
MD51028 Bit 0 and Bit 1 = 1	Bit mask for available search modes in G code program
Option (828D only)	Extended operating functions

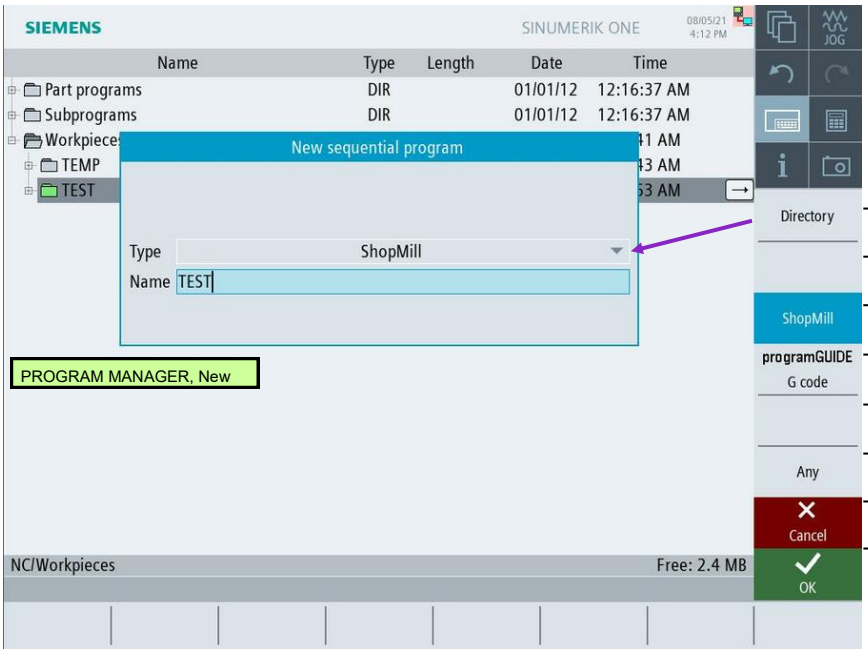
Notes:

Machine —> AUTO —> Simultaneous recording



Softkey selection	Description
Option	**3D simulation 1 (finished part)**

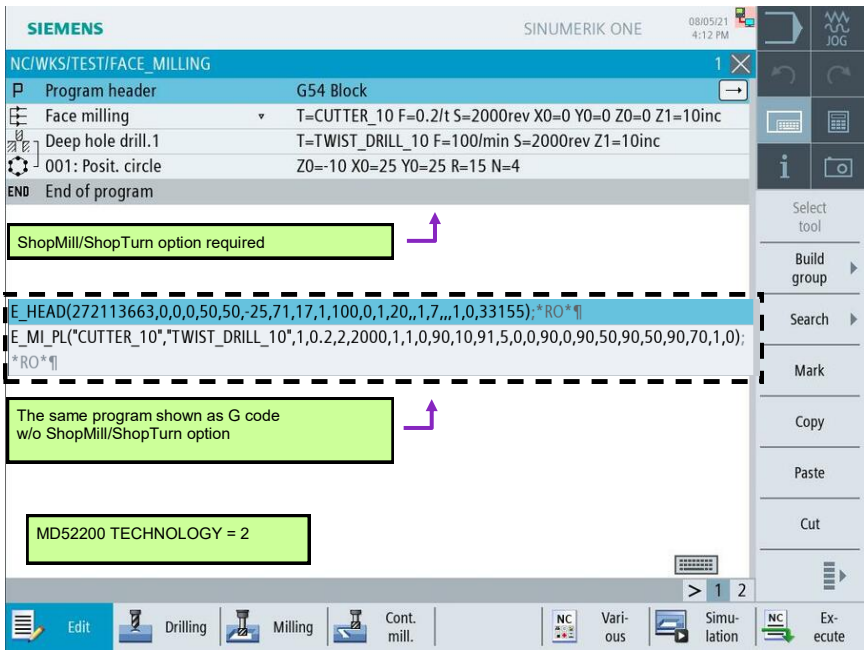
Program manager —> New program

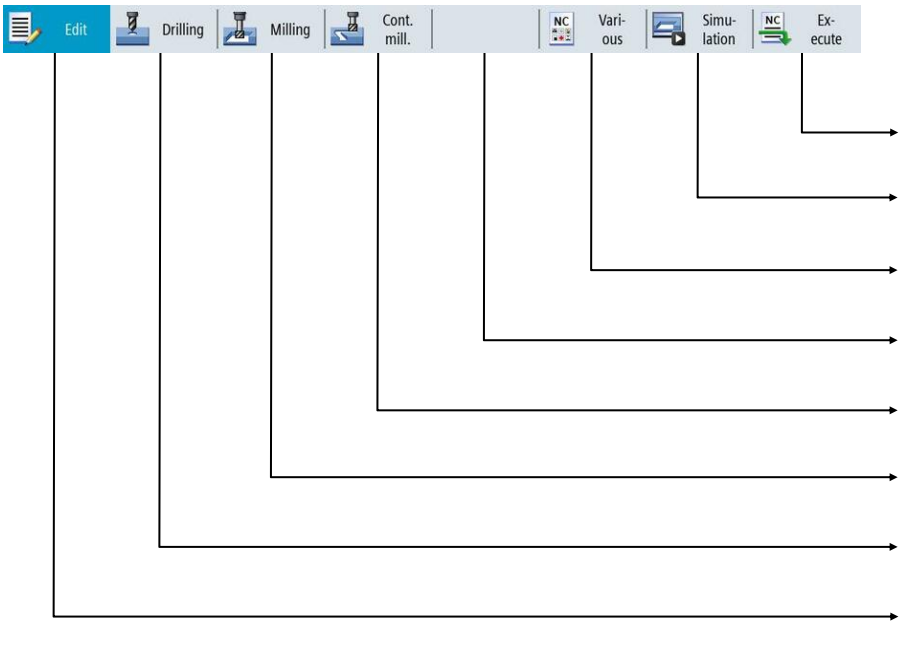


Softkey selection	Description
Option	**ShopMill/ShopTurn**

Notes:

Program —> Edit

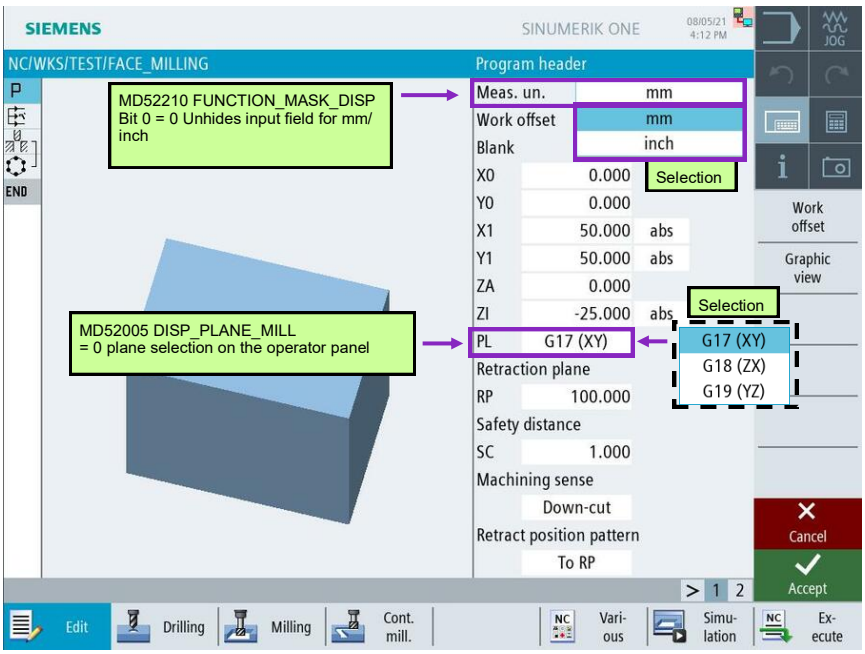




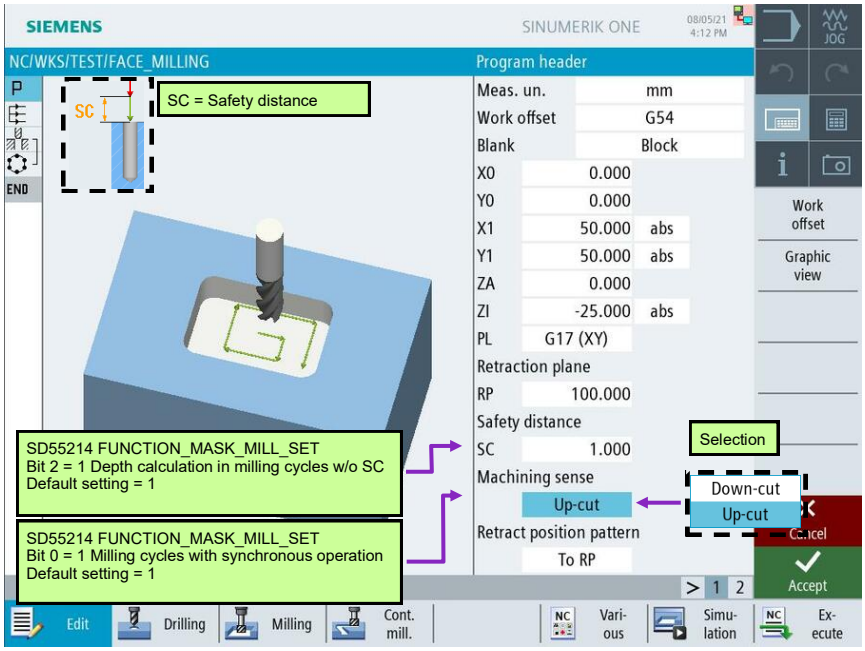
Softkey selection	Description
MD52200 = 2	=2: Technology Milling
MD52200 = 2	=2: Technology Milling
MD52200 = 2	=2: Technology Milling
MD52200 = 2	=2: Technology Milling

Notes:

Program —> Edit —> Program header

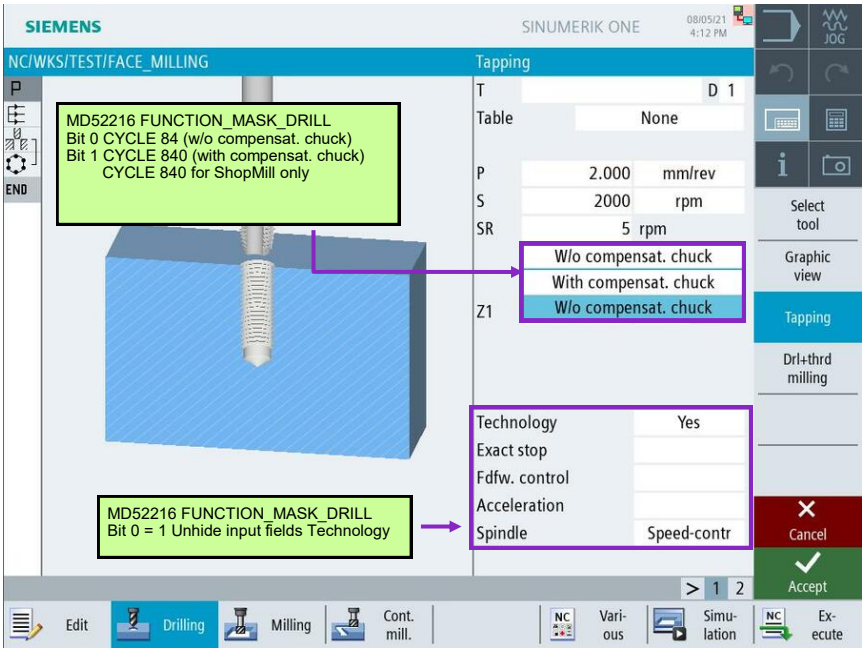


Program —> Edit —> Program header

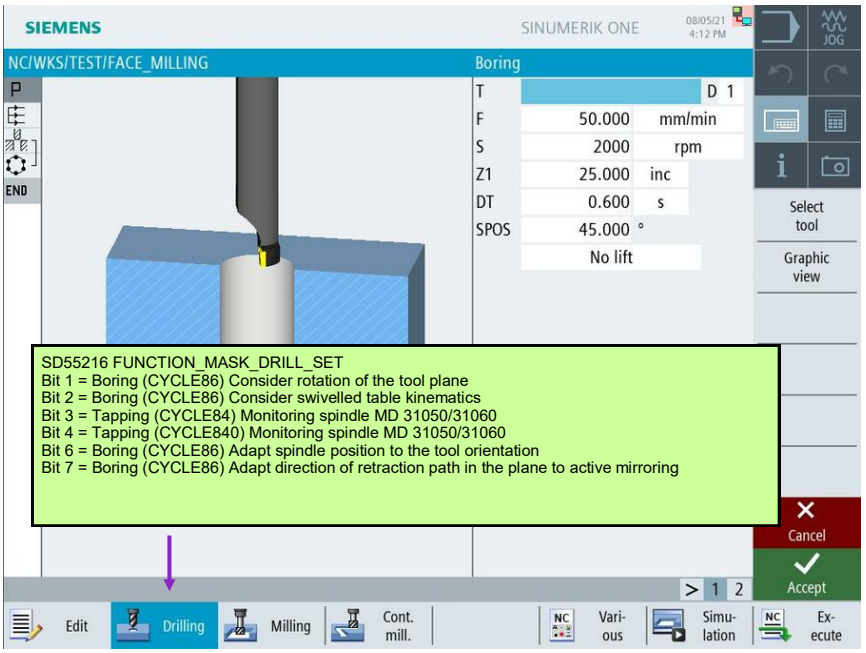


Notes:

Program —> Drilling

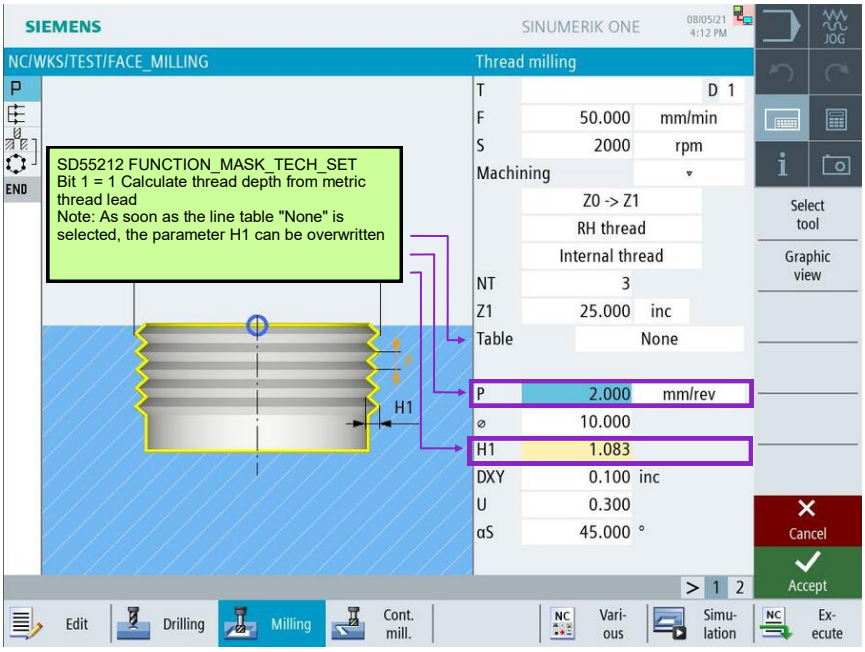


Program —> Drilling

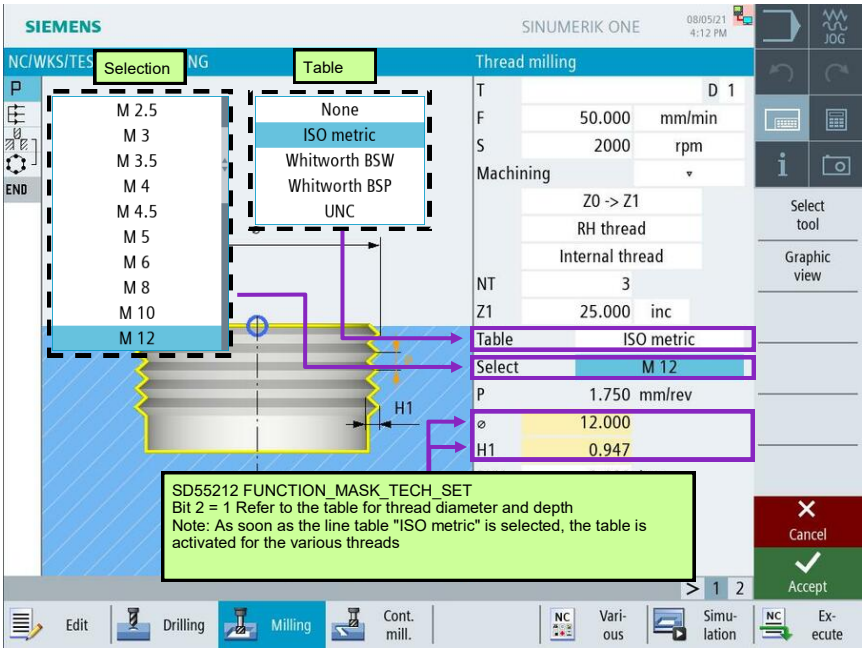


Notes:

Program —> Milling

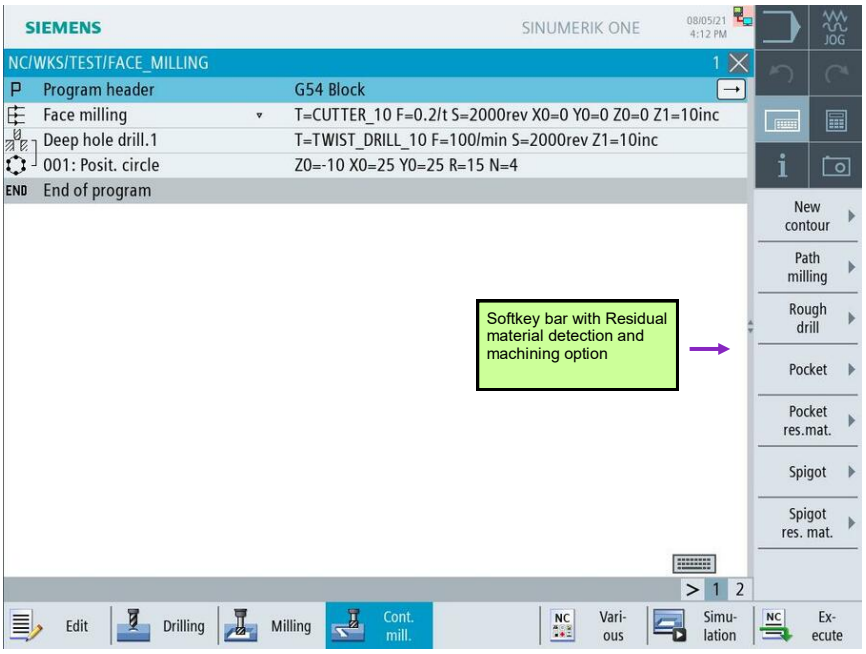


Program —> Milling



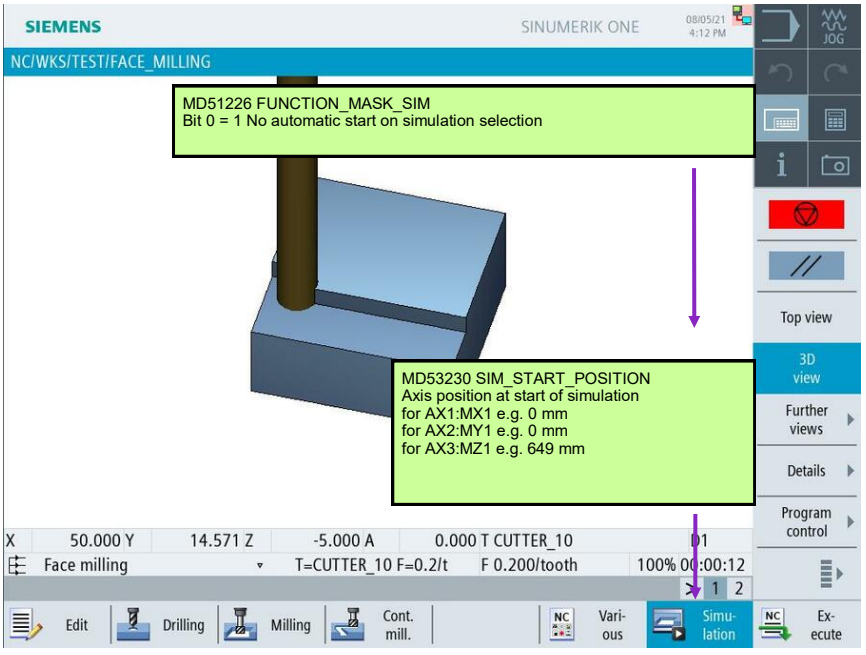
Notes:

Program —> Contour milling



Softkey selection	Description
Option	Residual material detection and machining
Option	Residual material detection and machining

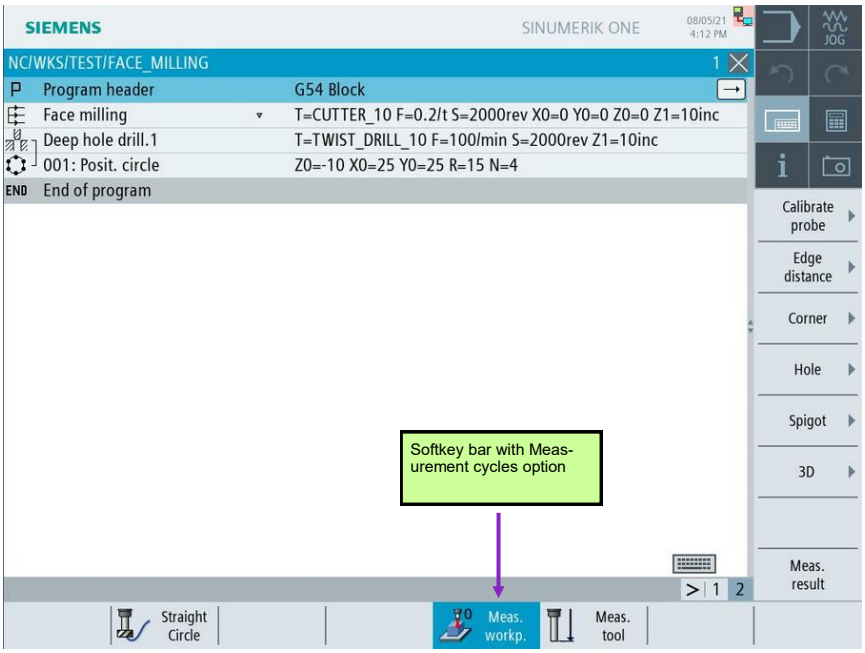
Program —> Simulation



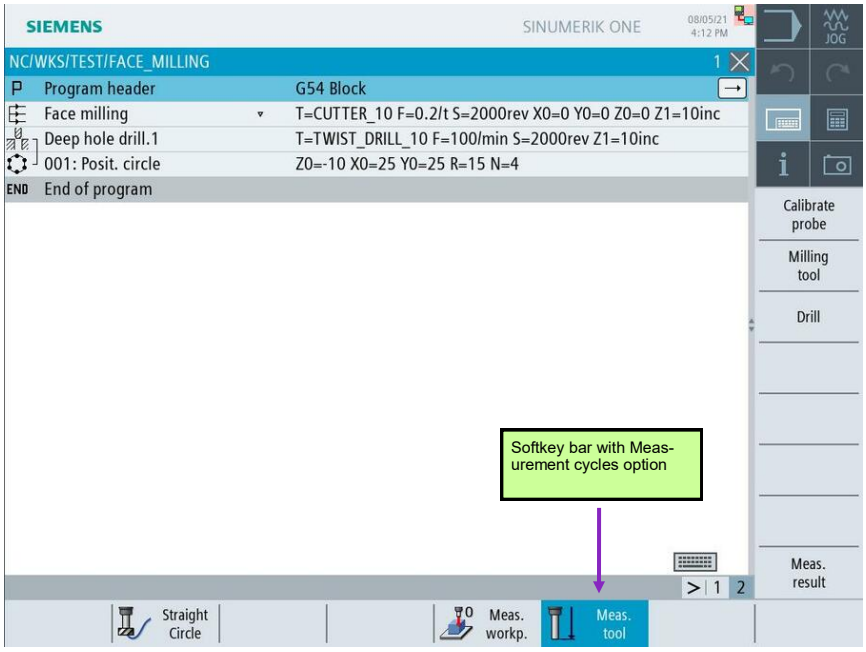
Softkey selection	Description
Option	3D simulation 1 (finished part)

Notes:

Program —> Measure workpiece



Program —> Measure tool



A121: END

Notes:

Notes:

A127

SINUMERIK ONE

Program handling using external storage with USB, SD card, IPC, and networked PC

Module Description:

It is often a requirement to store machining programs externally to the controller.

The external storage can be on the System SD card/SSD IPC, USB memory, or a networked PC.

Examples reason for storing programs externally:

- The size of the program.
- The number of programs on the controller is at the limit.
- Centrally stored programs can be accessed by many networked machines.
- Ensures that the programs can be recovered in the event of data loss.

The EES (Execute from External Storage) feature can then be used to execute the programs, from the external source, as if they were stored in the NC memory.

Module objective:

Upon completion of this module you will will be able to perform the following tasks:

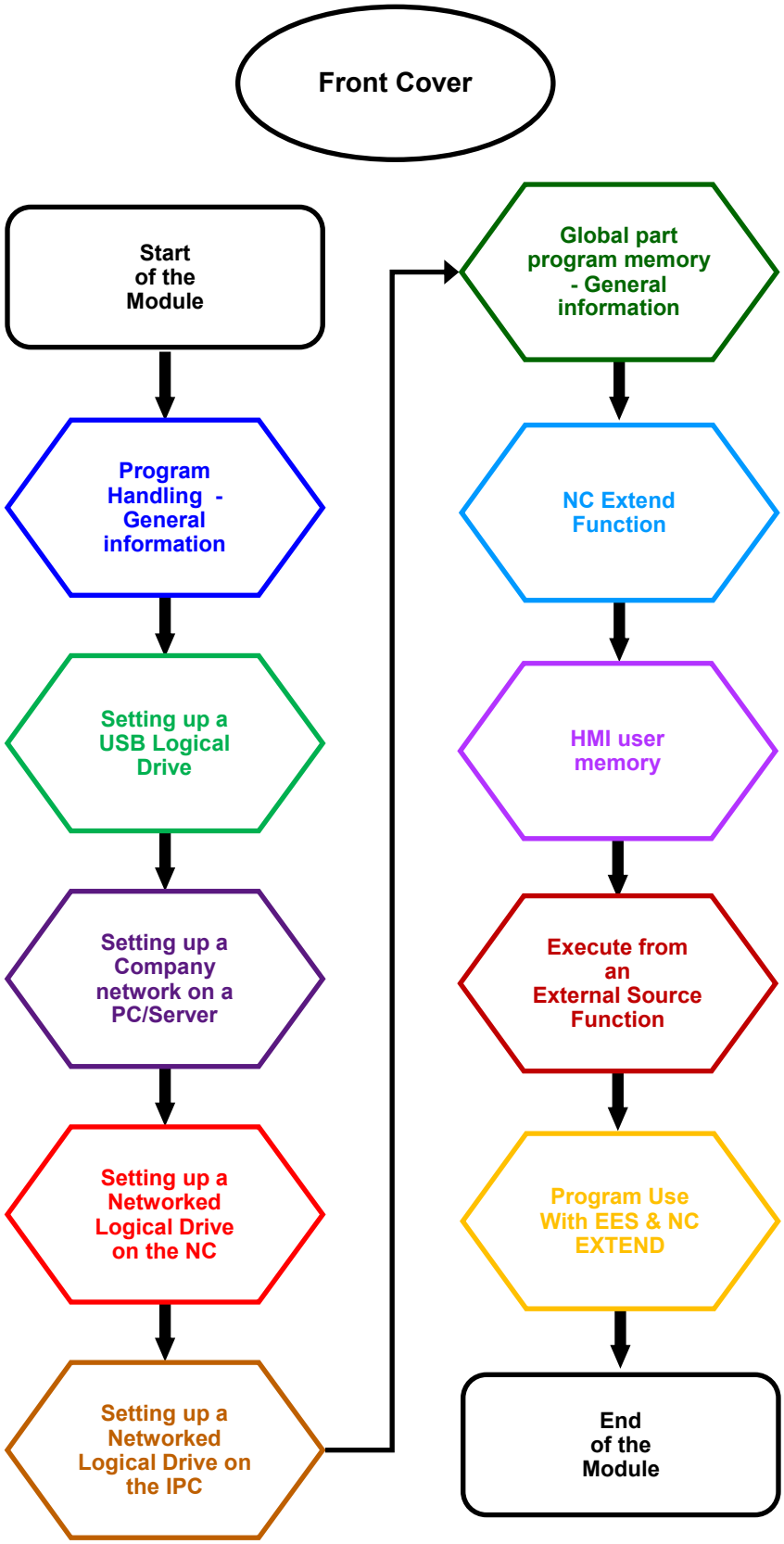
- Setup a logical drive including USB and network drive.
- Activate the NC Extend memory.
- Activate the HMI user memory.
- Activate the Execute from an External Source command.
- Use main programs and subprograms with the above features.

Content:

Program handling - General information
Setting up a USB Logical Drive
Setting up a Company network on a PC/Server
Setting up a Networked Logical Drive on the NC
Setting up a Networked Logical Drive on the IPC
Global part program memory - General information.
NC Extend Function
HMI user memory Function
Execute from an External Source Function
Program Use With EES & NC EXTEND

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Program Handling - General information
Setting up a USB Logical Drive
Setting up a Company network on a PC/Server
Setting up a Networked Logical Drive on the NC
Setting up a Networked Logical Drive on the IPC
Global part program memory - General information
NC Extend Function
HMI user memory
Execute from an External Source Function
Program Use With EES & NC EXTEND
End of the Module

Program Handling - General information.

Logical Drives

The logical drive feature allows storage of machining programs in addition to the standard NC memory.

.

The additional storage can be:

- On the controller (System SD card).
- A USB device.
- A networked computer.
- IPC

By default, the USB features are already active.

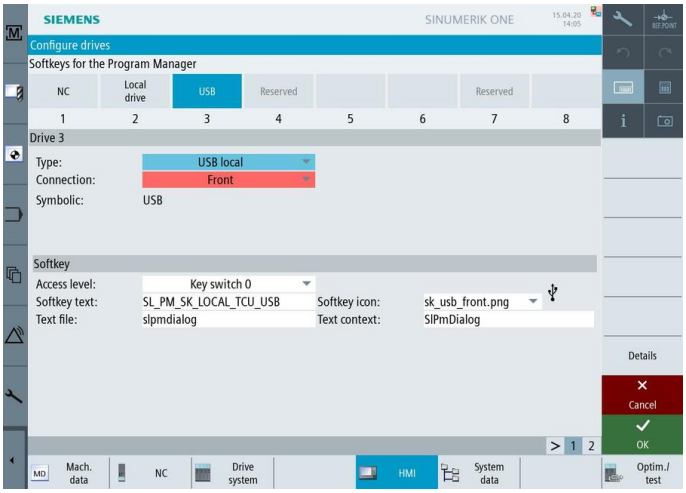
Programs can be easily copied and pasted between the controller and a logical drive.

Setting up a USB Logical Drive

Locate the logical drives set-up area:



Default Logical drives screen for USB:



When the CNC System software is first loaded onto the SSD card, the USB type is set to “USB local” and Connection is set to “Front”, these are the default settings.

Depending on the OP that is connected to the system, changes may have to be made.

By default when an ITC display is fitted to the system, the first accessible USB port is active, in this case X61.

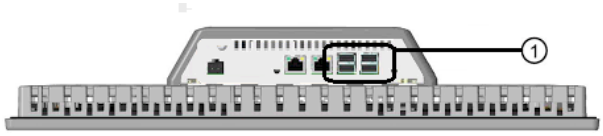
A simple USB extension may be used to extend the USB socket from the OP USB port to the machine operating panel exterior.



Notes:

Simatic ITC.
There are four USB ports available:

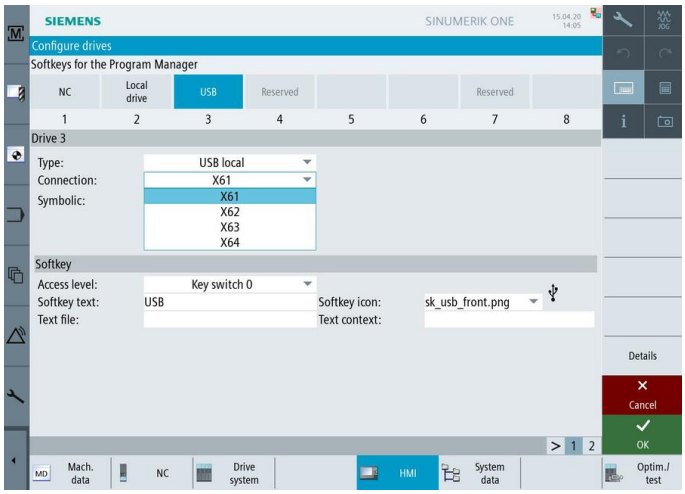
1 - X61 (USB Hi-Speed type A socket).
1 - X62 (USB Hi-Speed type A socket).
1 - X63 (USB Hi-Speed type A socket).
1 - X64 (USB Hi-Speed type A socket).



The required USB ports are defined in the logical drives set-up area. The pull-down menu shows the available ports.

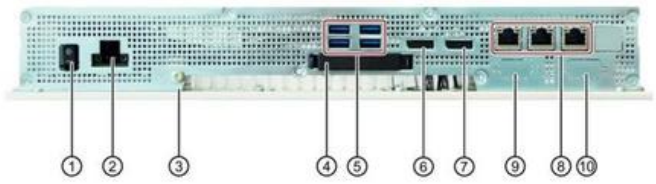
Below shows the four ports of the Simatic ITC that the system has recognized.

Select the port that the USB stick will be plugged into.



Simatic IPC 427E.
There are four USB ports available:

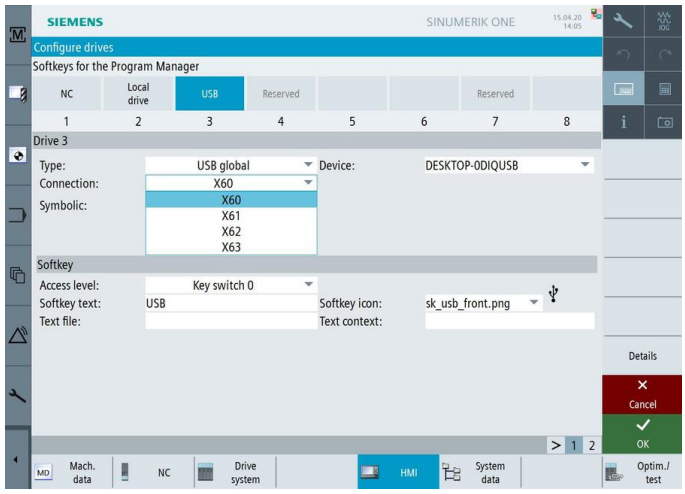
5 - X60 (USB Hi-Speed type A socket).
5 - X61 (USB Hi-Speed type A socket).
5 - X62 (USB Hi-Speed type A socket).
5 - X63 (USB Hi-Speed type A socket).



The required USB ports are defined in the logical drives set-up area. The pull-down menu shows the available ports.

Below shows the four ports on the IPC that the system has recognized.

Select the port that the USB stick will be plugged into.



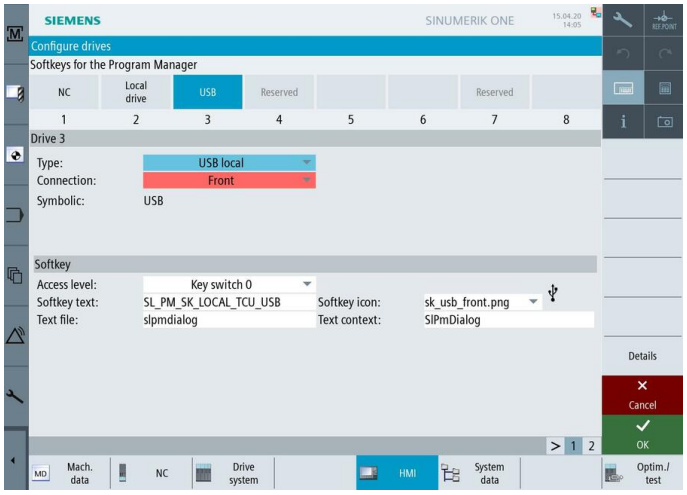
Notes:

Setting up the USB Logical Drive with GLOBAL settings:

Locate the logical drives set-up area:



Default Logical drives screen for USB:

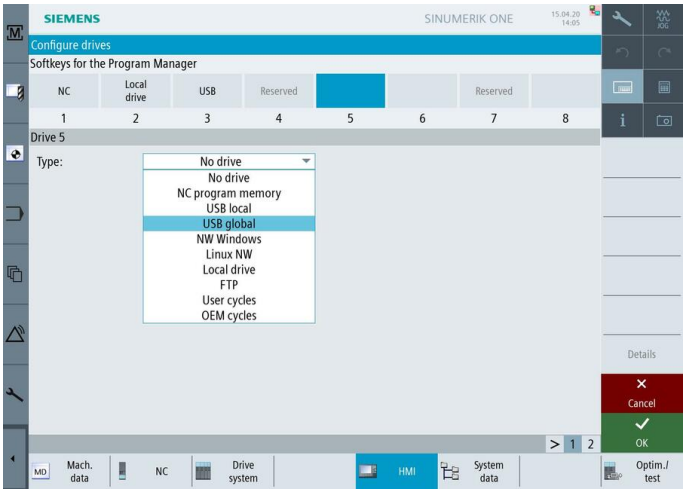


In this example a second OP screen has been added.

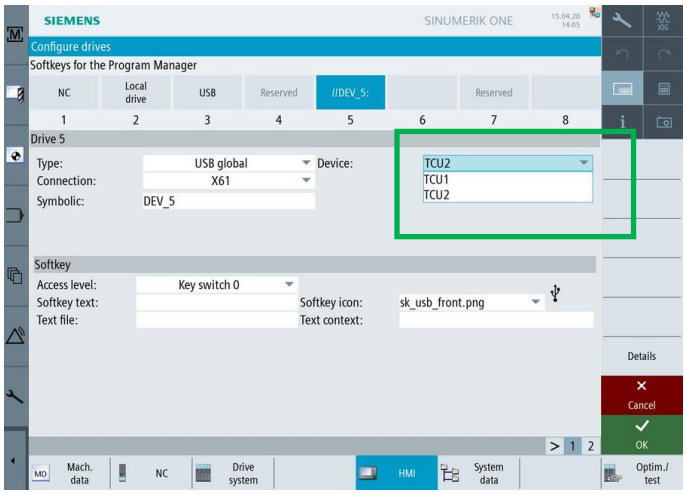
So there will be a doubling up of USB sockets:

1 - X61.
2 - X62.
3 - X63.
4 - X64.

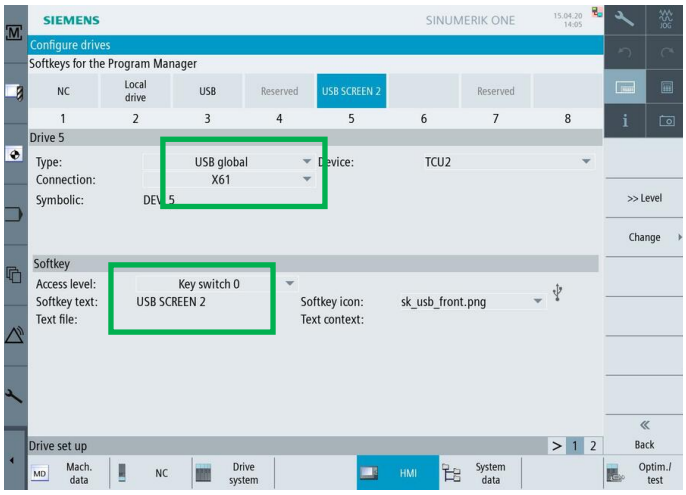
Horizontal position 5 is chosen, and USB global is selected from the menu.



This will allow a “Device” menu to appear allowing the selection of which device is required for the USB sockets.

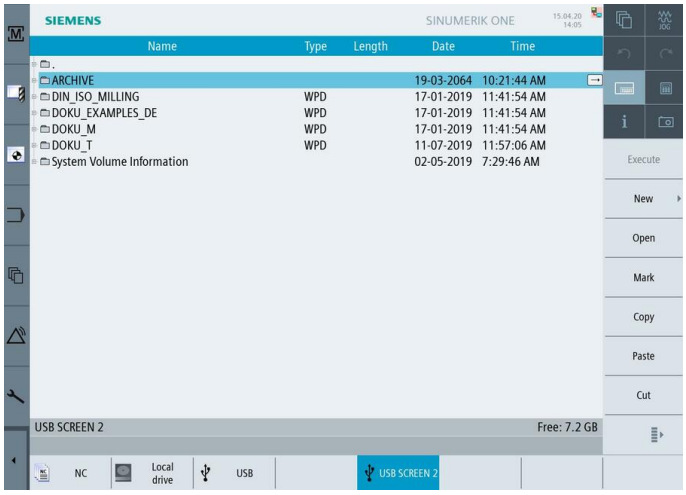


Select which socket is to be used from the “Connection” window, followed by giving the softkey in position 5 text.
To activate press the OK softkey.



Notes:

Located in the “Program Manager” area, there is now the ability to select HSK 5 as a USB connection from a second screen.



Setting up a Company network on a PC/Server

The following example shows how to connect to a Windows10® computer via the **X130** (Factory network) interface of the controller (Direct connection without DHCP)
The computer will be configured with an IP address of **192.168.100.2**
Both will have the subnet mask of **255.255.255.0**

Note:
There are certain IP addresses that must be avoided as they are reserved for use by the system.
IP addresses in the **192.168.214.xxx** range must be avoided.
IP addresses in the **192.168.215.1 to 31** range must also be avoided.

Connect a suitable Ethernet cable between the computer and the X130 interface of the controller.



Overview of file sharing using the SMB protocol in Windows 10/11/Server

The Server Message Block (SMB) protocol is a network file sharing protocol that allows applications on a computer to read and write to files and to request services from server programs in a computer network.
The SMB protocol can be used on top of its TCP/IP protocol or other network protocols. Using the SMB protocol, an application (or the user of an application) can access files or other resources at a remote server. This allows applications to read, create, and update files on the remote server. SMB can also communicate with any server program that is set up to receive an SMB client request. .

Notes:

All the variants of the SMB protocol

SMBv1 was released in 1984 by IBM for file sharing in DOS. Microsoft modified and updated it in 1990.

CIFS was released in 1996 with more features and support for larger file sizes. It came together with the new Windows 95.

SMBv2 debuted in Windows Vista in 2006. It featured a notable boost in performance because of increased efficiency — fewer commands and subcommands meant better speeds.

SMBv2.1 came with Windows 7, bringing improved performance.

SMBv3 was introduced with Windows 8 with many updates. Most notable of which is enhanced security — the protocol started supporting end-to-end encryption.

SMBv3.02 came together with Windows 8.1. It offered the ability to increase security and performance by completely disabling SMBv1.

SMBv3.1.1 was released in 2015 with Windows 10. It added more security elements to the protocol, like AES-128 encryption, protection from man-in-the-middle attacks, and session verification.

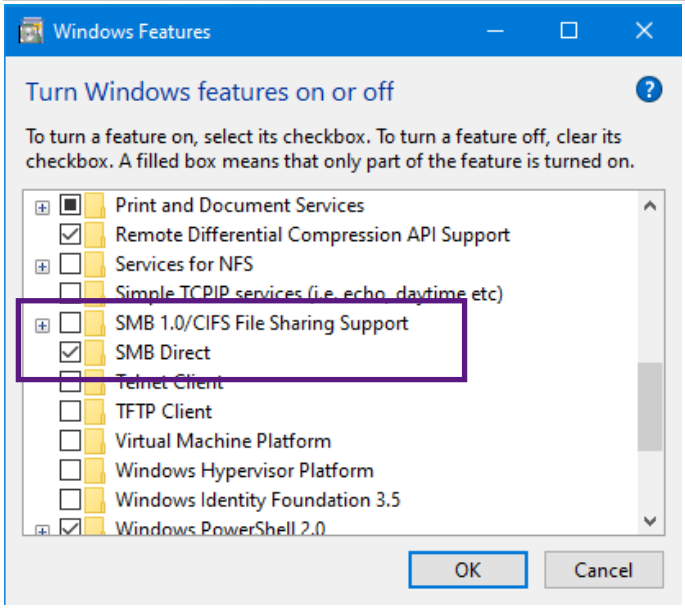
To check these protocols within Windows 10/11/Server are active:

Go to:
Settings
Apps and features
Program s and features
“Turn Windows features on or off”

SMB 1.0/CIFS File Sharing Support
Is NOT active by default
(Suport for the SMB 1.0/CIFS File sharing protocol and the Computer Browser protocol)

SMB Direct
Is active by default
(Remote Direct Memory Access (RDMA) support for the SMB 3.x file sharing protocol)

These features turns on file sharing between Windows and Linux.using SMB protocol for there respective versions



Select the required SMB protocol

To make active restart the PC/Server

Notes:

Configuring an SMB client

To connect the NCU and the PC system in the network to an external Microsoft Windows 10/11/server, the server protocol SMB (Server Message Block) is used. Microsoft Windows 10 supports SMB servers with protocol versions SMBv1, SMBv2 and SMBv3. Depending on the configuration, different servers require a different protocol version to enable a connection with the NCU and the PC system.

The SMB protocol versions are compatible with SINUMERIK ONE v6.2x:

SMB protocol version

V1.0, V2.0, V3.0.2, V3.1.1

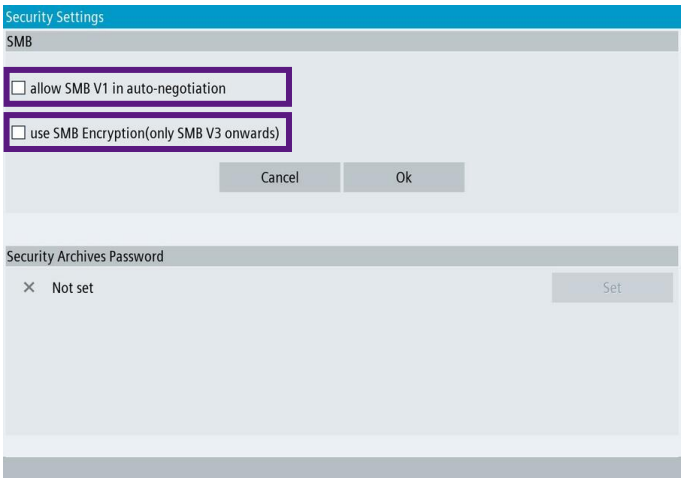
Configuring an SMB on the HMI

Locate the security settings screen:





Modify



A tick can then be applied to correct SMB level required for the factory PC system that the controller is being connected to.

Once selected apply the new setting with the “Ok” softkey.

Ok

To make active, restart the system with a power off/on.

Notes:

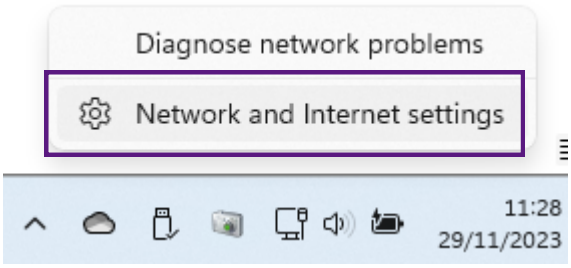
PC/PG setup of the computer network.

The computer must now be given an IP address that is in the same subnet mask.
The IP address for this example is: **192.168.100.2** with a sub-net mask of **255.255.255.0.**

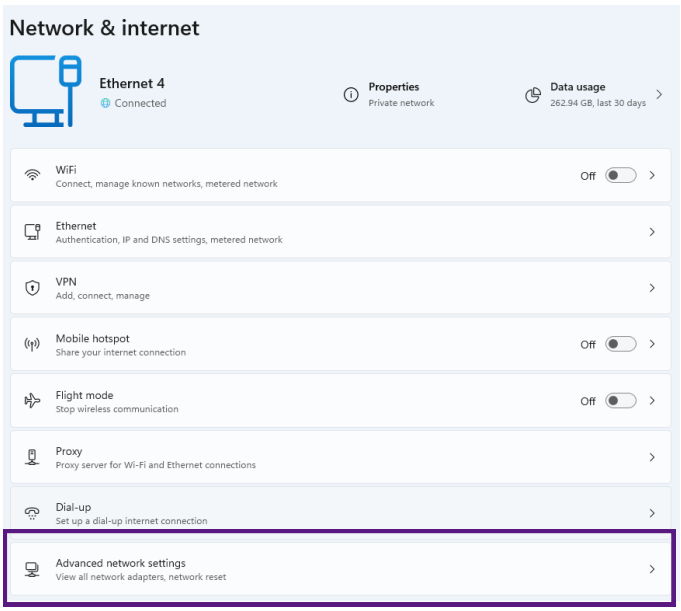
Locate and open the **Network and Internet settings** of the computer to be connected to the controller.



Right click the network icon.



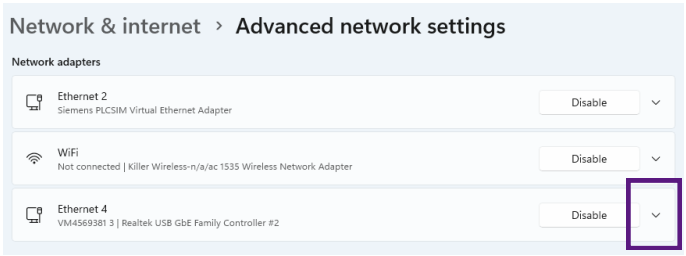
Select "Network and Internet settings"



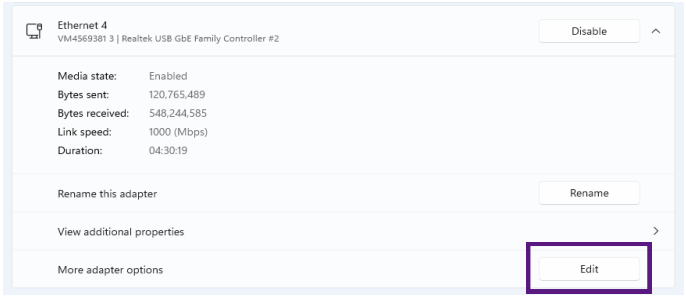
Select "Advanced network settings"

Note the name of the network adapter associated with the Local Area Connection.
In the **example** below the network adapter is of the type: **Realtek USB GbE Family Controller #2**.

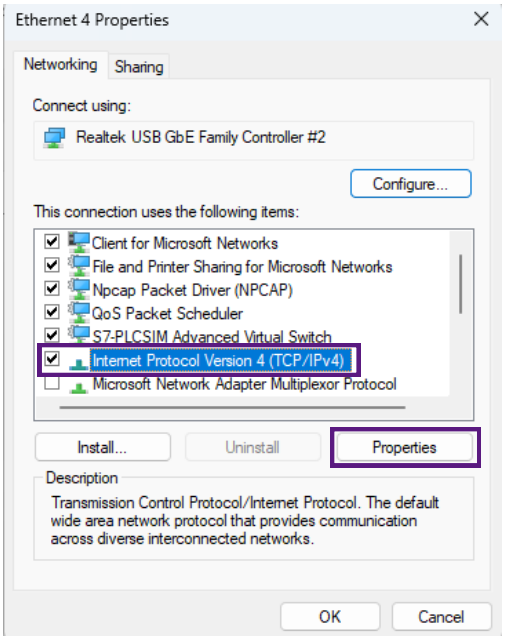
Note: Network adapter may be different between PC makes and types.



Click the expand icon, then on the "Edit" icon:



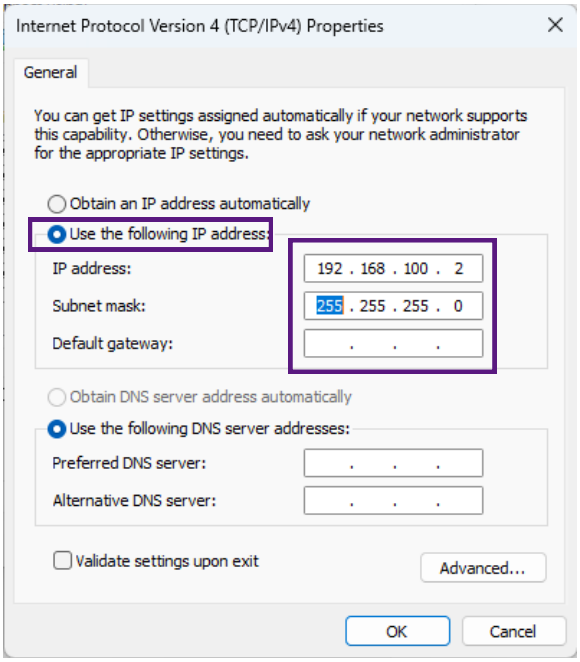
Select the option "**Internet Protocol Version 4 (TCP/IPv4)**" followed by the "Properties" button:

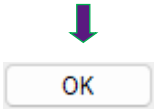


Properties

Notes:

The IP address and the subnet mask can now be entered:





Computer- Shared Folder

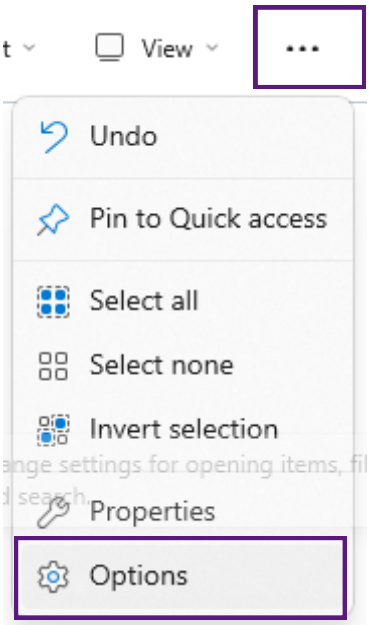
A shared folder allows it to be accessed via the network. The folder is required for the storage of programs which will be accessible from the controller.

For reasons of security it is possible to restrict access to the shared folder by setting permissions.

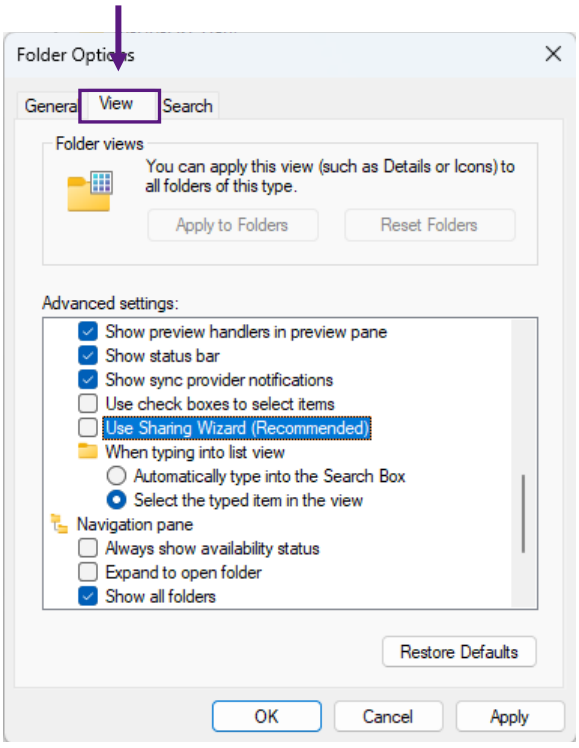
The following example will show how to create a folder on the computer and how to set up sharing and permissions.

The option - ***Use sharing Wizard*** - can be deselected, via the ***Tools*** menu in Windows® Explorer, to ease setup:

From Windows® Explorer, select the ***...*** Menu (See more), followed by ***Options:***

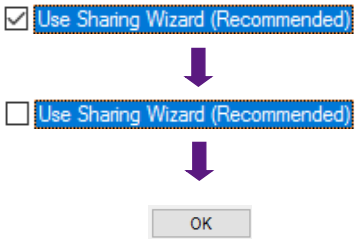


Select ***View:***



Notes:

Deselect the ***Use Sharing Wizard*** option:

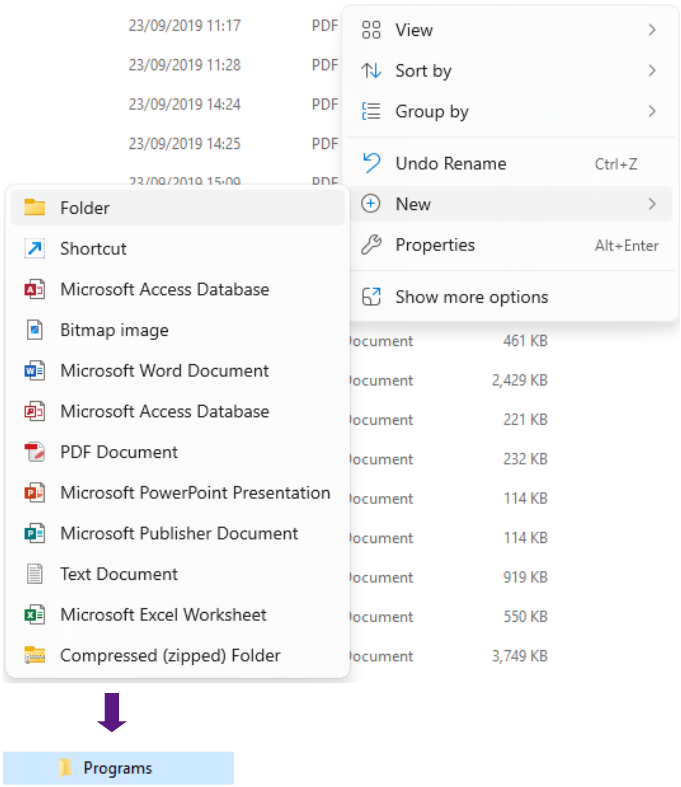


Creating a New Folder

Example:

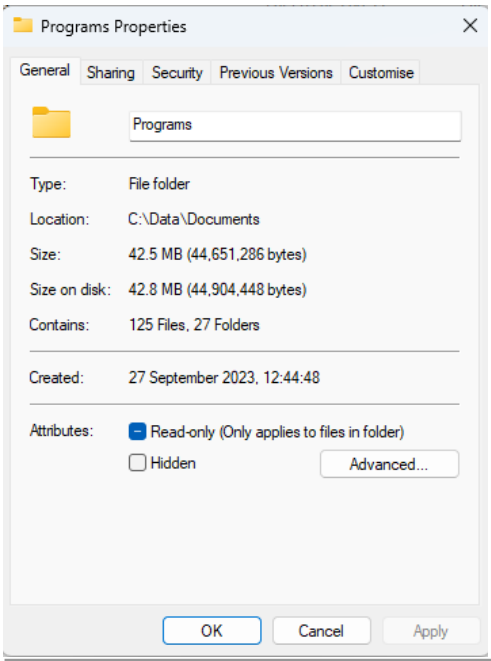
In Windows® Explorer:

- Select a suitable location
- With the mouse pointer In the right hand window, right click and create a new folder.
- Give a suitable name to the folder.
  Example ***Programs.***

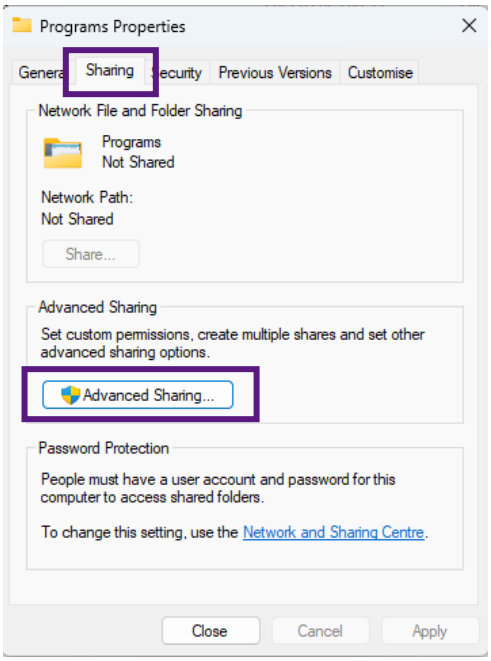


Sharing the New Folder

- Right click with the mouse pointer on the ***Programs*** folder.
- Select the ***Properties*** option from the list

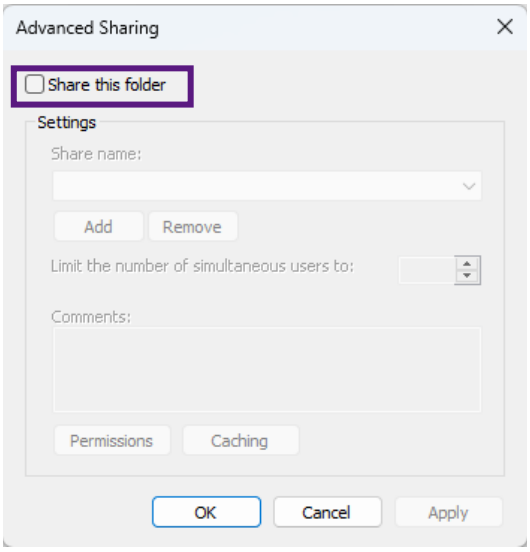


- Select the ***Sharing*** tab.
- Select the ***Advanced sharing*** tab.

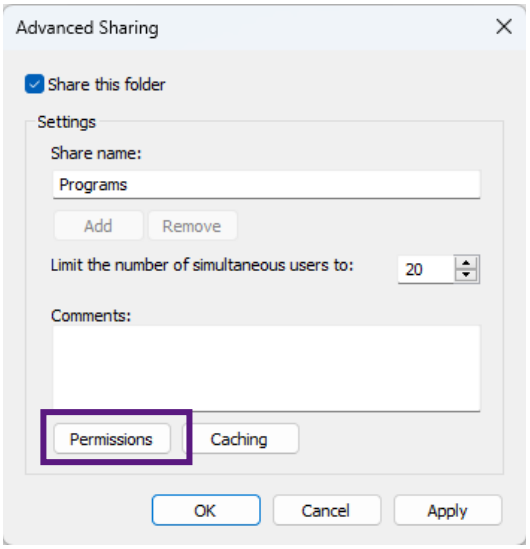


Notes:

- Select ***Share this folder***.

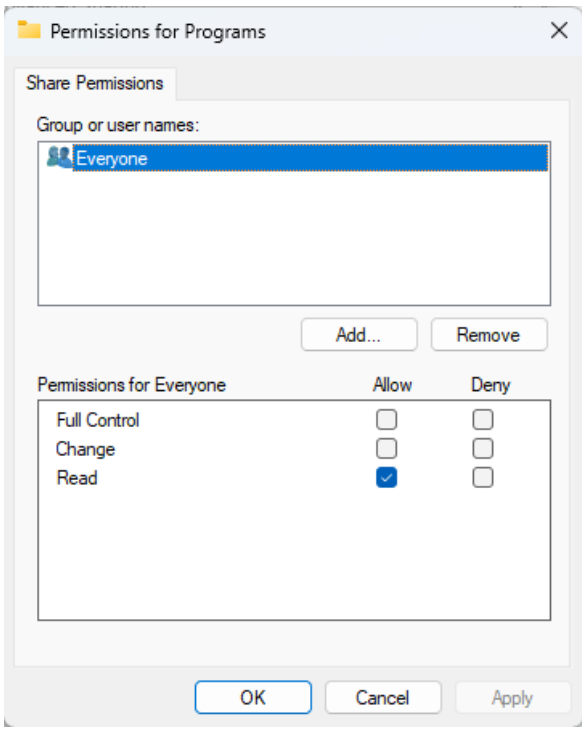


- Select ***Permissions***.

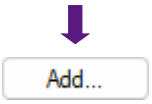




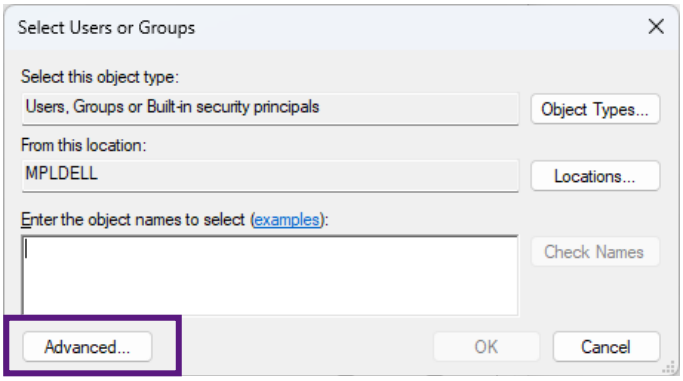
The default setting allows everyone to have read only access to the folder.



It is possible to restrict access to the shared folder by other users of the PC.
The default setting allows everyone to have read only access to the shared folder.
If restrictions are required, the ***Everyone*** user can be removed or the permissions deselected.
Users can than be added via the ***Add*** button.



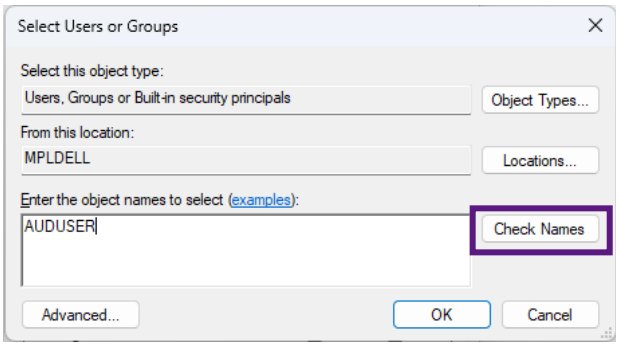
The required users of the computer can be searched for in the ***Advanced...*** field.



Notes:

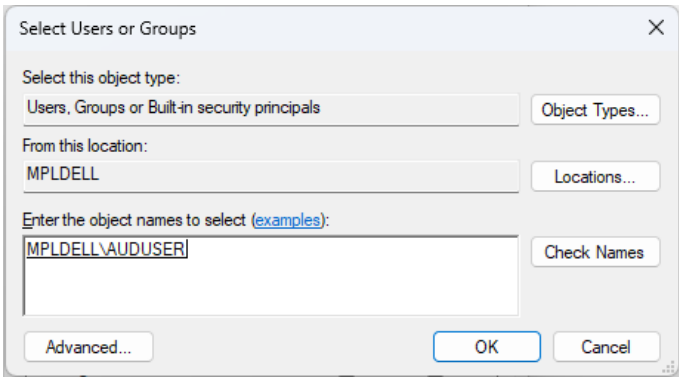
In the following example, the user ***AUDUSER*** is going to be given permission to access the shared folder.

Enter ***AUDUSER*** in the **Check Names** field, then select the ***Check Names*** button.



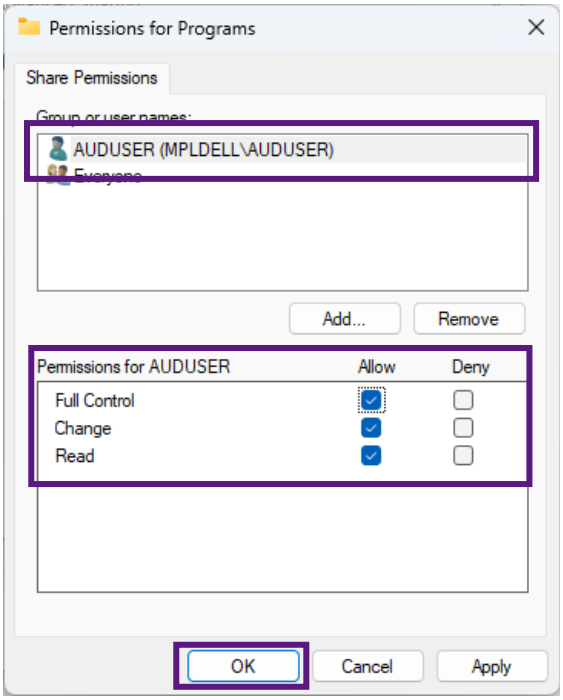
Check Names

If the user exists, it will be displayed.



OK

Example: AUDUSER has been granted full access to the shared “Programs” folder.
The Share permission for Everyone has been removed.



Notes:

Setting up a Networked Logical Drive on the NC

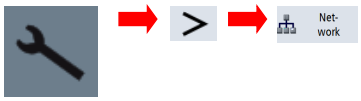
The following example shows how to connect to a Windows10® computer via the **X130** (Factory network) interface of the controller (Direct connection without DHCP)
X130 will be configured with an IP address of **192.168.100.1**.
Both will have the subnet mask of **255.255.255.0**

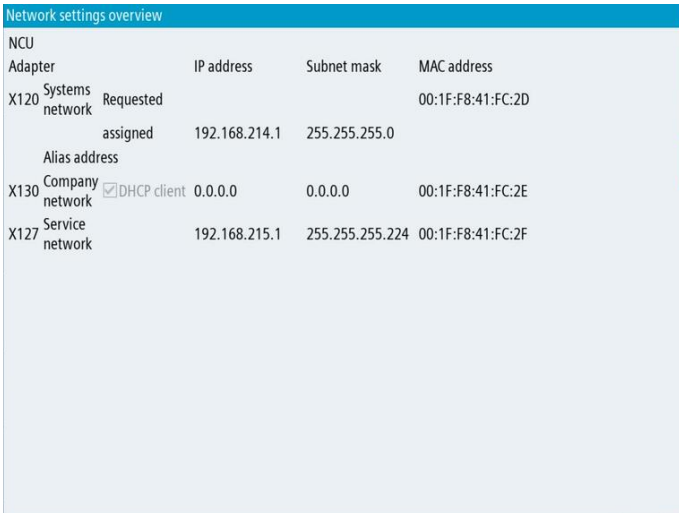
Configuring Interface X130:

Connect a suitable Ethernet cable between the computer and the X130 interface of the controller.



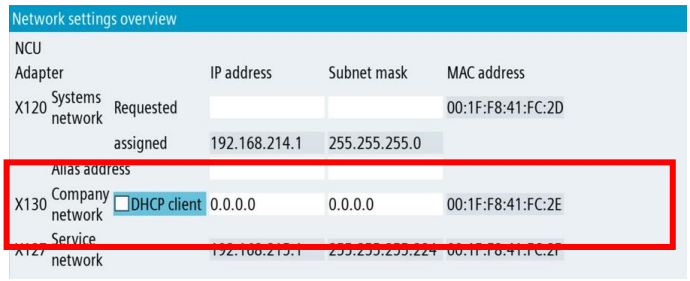
Locate the network settings screen:



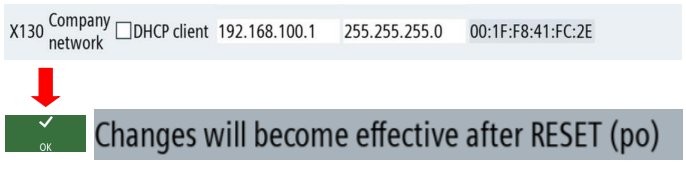


Setting the X130 IP Address:





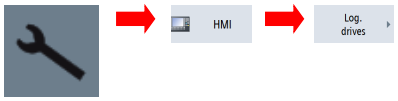
Deselect the DHCP client option and enter the required IP address and subnet mask (192.168.100.1/255.255.255.0):



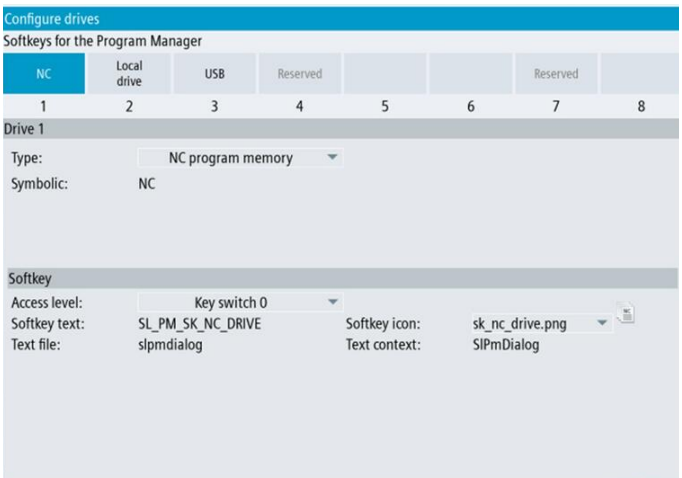
Note: The control needs to be switched off/on to activate the new settings

Setting up the Logical Drive:

Locate the logical drives set-up area:



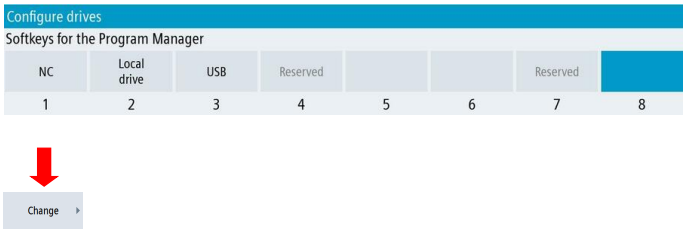
Default Logical drives screen:



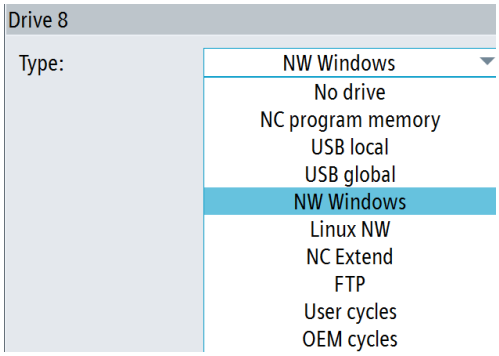
Notes:

Select a free soft-key. Further soft-keys, 24 in total, are accessed via the ***>>Level*** soft-key.

Example - Soft-key 8:



Choose the ***NW Windows*** option:



For this example the following data is required (the external computer settings will be changed later):

Computer name: 192.168.100.2
(IP address of the PC)

Share name: Programs
(Name of the shared folder on the PC).

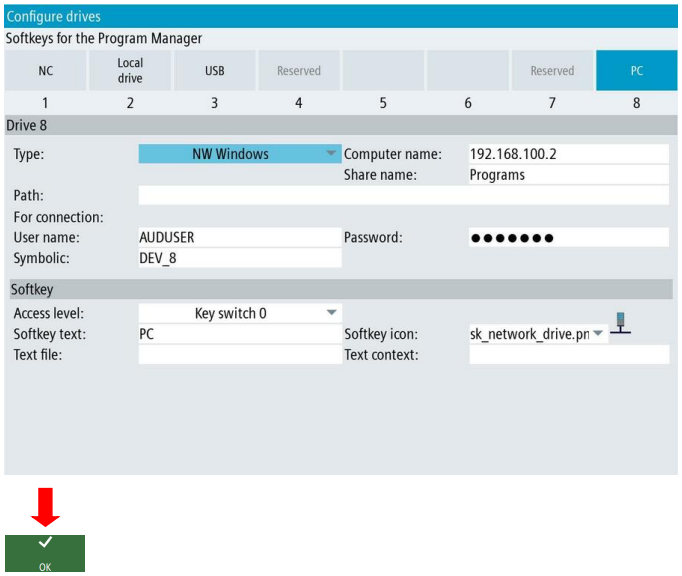
User name: AUDUSER
(Login name of the PC)

Password: SUNRISE
(Login password of the PC, for example)

Soft-key text: Programs
(the text that will appear on the soft-key in the Program manager area).

The remaining parameters are to be left unchanged for this example.

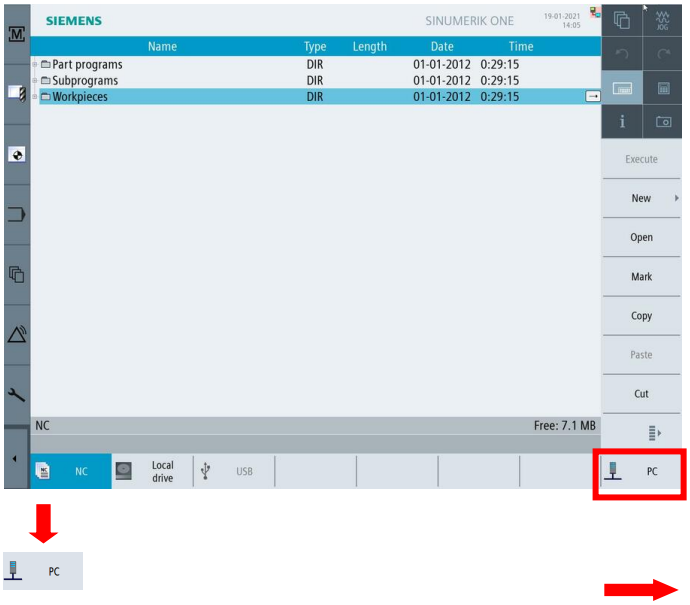
The relevant data has to be entered:





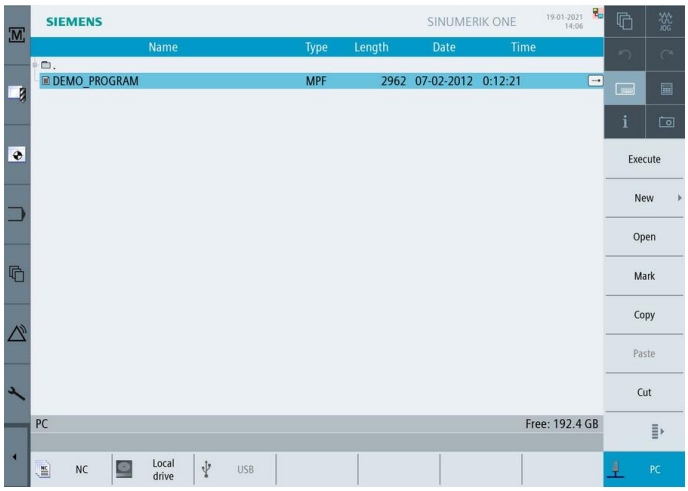
It should now be possible to access the shared folder from the ***Program manager*** area of the controller.





Notes:

The shared folder being accessed from the controller.



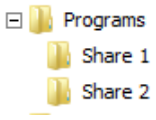
Note:
If the Soft-key for accessing the shared folder is grey, this indicates a problem with the network set up or possibly the security of the computer hosting the shared folder.
Firewall settings should be checked and temporarily disabled to see if this is causing problems.

<u>Accessing individual sub-folders.</u>

If a folder is created within the shared folder it will inherit the share properties of the host folder.

It is possible to directly access a subfolder by entering a path on the Logical drive set-up screen.

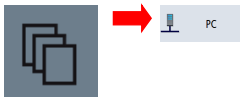
Example:
The ***Programs*** folder has been shared.
Two folders, ***Share 1*** & ***Share 2***, have been added within the shared folder.
There is no need to share these new folders.

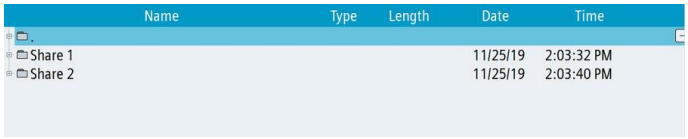


Access the ***Program manager*** screens:



The new folders can be seen inside the original shared folder:

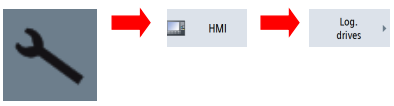


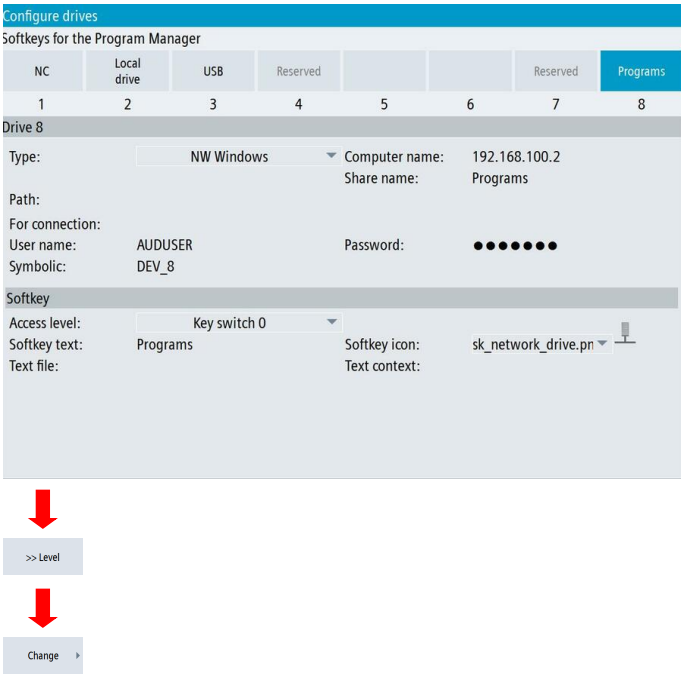


Accessing individual folders.

By entering a path in the logical drives set-up screen it is possible to access individual folders contained within the main shared folder.

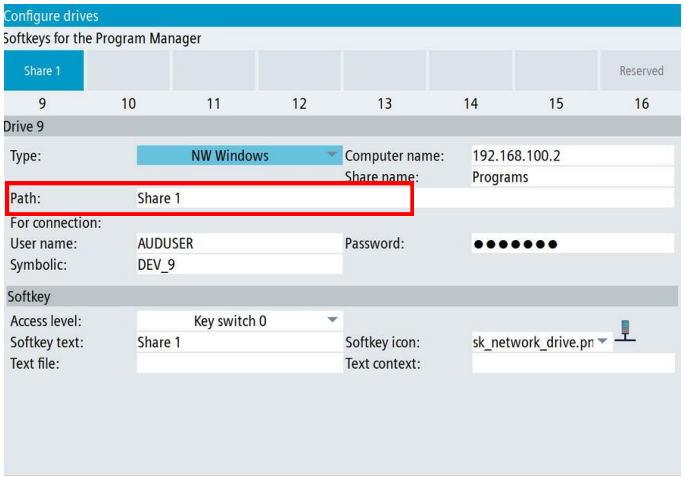
Example:
The following example shows two additional soft-keys which have been set-up to give access to one each of the new folders.



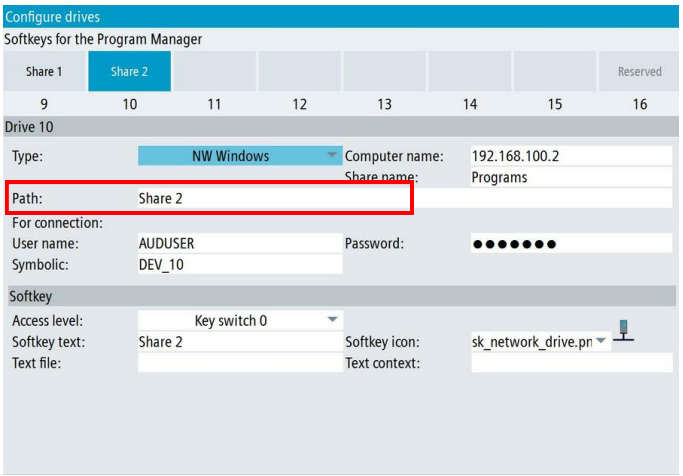


Notes:

The path to **Share 1** has been included.

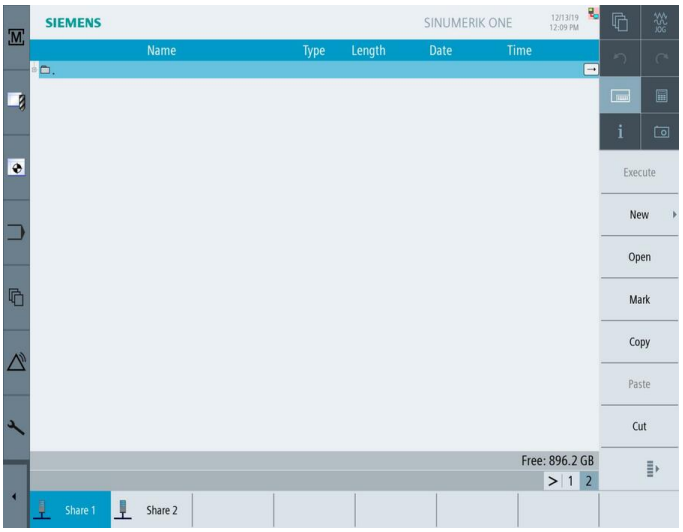


The path to **Share 2** has been included.

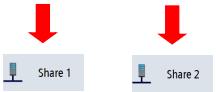


Access soft-keys 9 & 10 in the ***Program manager*** area:





The two sub-folders can now be accessed:



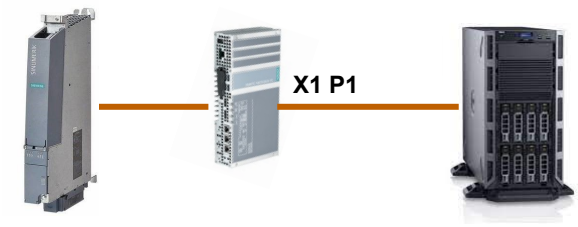
Notes:

Setting up a Networked Logical Drive on the IPC

The following example shows how to connect to a Windows10® computer via the **Eth1 (X1)** (Factory network) interface of the controller (Direct connection without DHCP)
Eth1 will be configured with an IP address of **192.168.100.1**.
Both will have the subnet mask of **255.255.255.0**

Configuring Interface Eth1:

Connect a suitable Ethernet cable between the computer and the Eth1 interface of the controller.

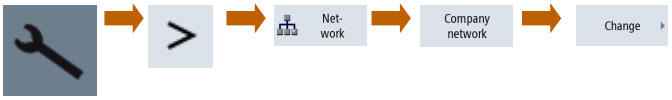


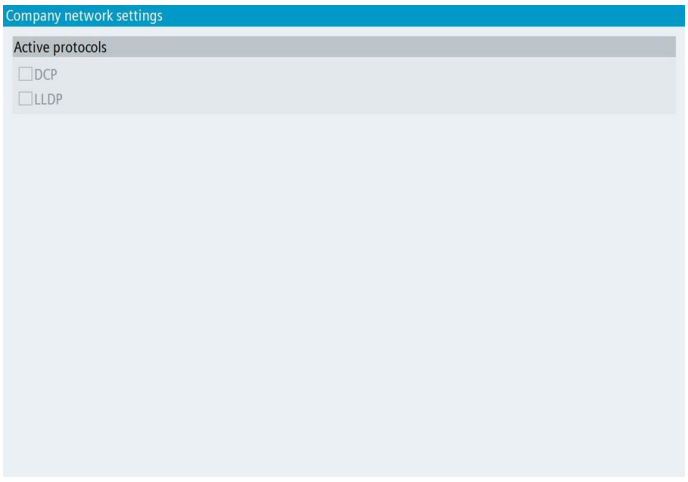
Configuring Interface Eth1

Ethernet interface Eth1 is configurable, the port can be assigned a fixed IP address or be set to automatically acquire an IP address.
Interface Eth1 is primarily for connecting to the company network. When configuring for connection to a company network, certain information from the company network administrator is required.

Firewall: Eth1 has a firewall which can inhibit the online connection of certain software tools. It is possible to open ports to allow connections to be made.

Change the firewall with the following key selection:





Activation of “protocols” on the “Company Network” on the IPC (HMI) if required:

DCP: (Discovery and Configuration Protocol)

The Discovery and Basic Configuration Protocol is a protocol definition within the PROFINET context. It is a link layer based protocol to configure station names and IP addresses. It is restricted to one subnet and mainly used in small and medium applications without an installed DHCP server.

LLCP: (Link Layer Discovery Protocol)

The Link Layer Discovery Protocol is a vendor-neutral link layer protocol used by network devices for advertising their identity, capabilities, and neighbours on a local area network based on IEEE 802 technology, principally wired Ethernet.

These two protocols are primarily used when DHCP is being used along with server / PC name, instead of fixed IP address.

Notes:

Eth1 (X1) Controller Set-up

The controller settings for Eth1 (X1) can now be entered.

Any existing Eth1 (X1) settings should be recorded before any changes are made.

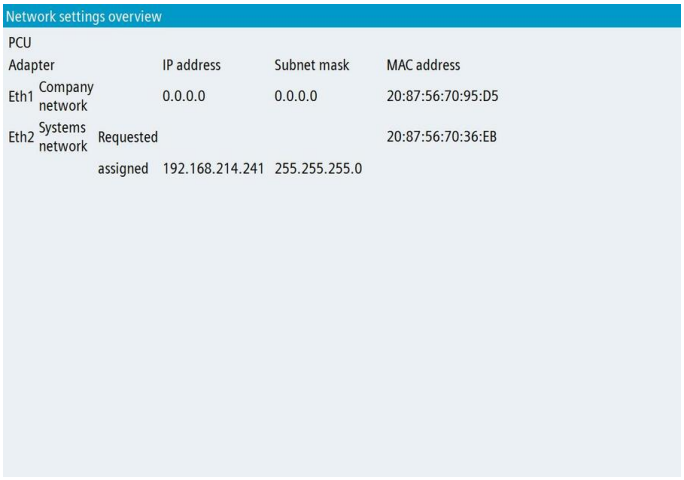
IP address restrictions- IP addresses **192.168.215.xxx** & **192.168.214.xxx**. are reserved for use by the system. Addresses within these ranges should not be used for Eth1 (X1).

The following example shows how to manually assign an IP address to Eth1 (X1).

Push the SK

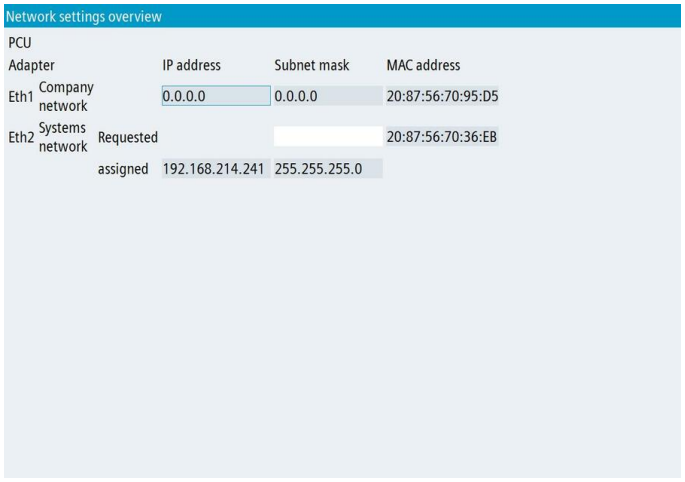


An overview of the Eth1 (X1) and Eth2 (X2) interfaces is displayed. Eth2 is a DHCP server with a fixed IP address of **192.168.214.241** and subnet mask of 255.255.255.224. This interface cannot be configured. Eth1 is the company network and is configurable.





It can be seen that Eth1 is grayed out, hence IP address and subnet mask can not be changed at the moment in the Setup“ Area, use the Diagnostics Area.

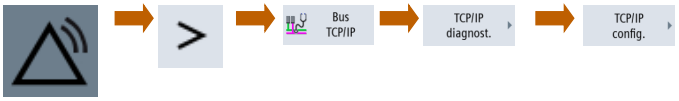


Diagnostic network

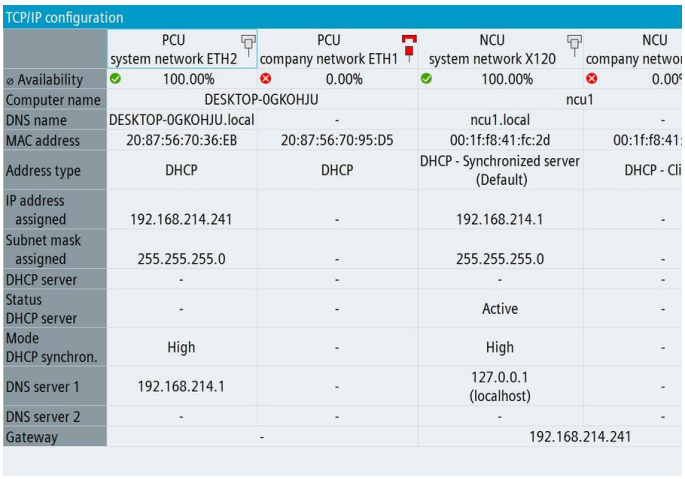
The following example shows how to manually assign an IP address to Eth1.

To change the IP address and subnet mask.

following key selection:



Notes:

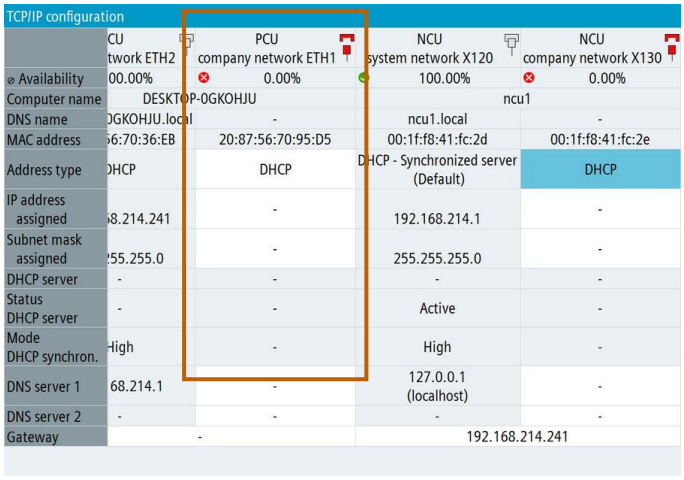


The screen shows all connections over the system.

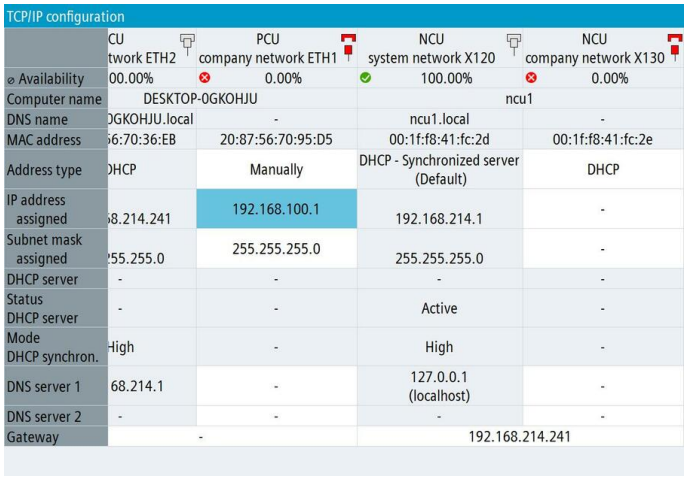
To enter an IP address:

Press “Change“ SK





Change the company network setting for the Eth1.

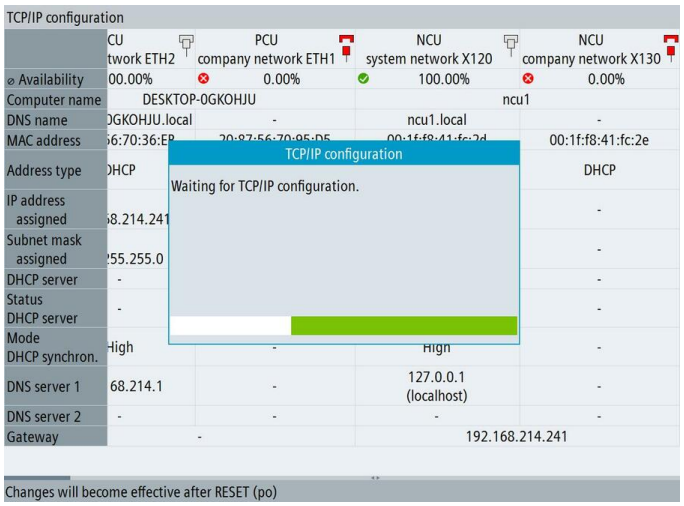


Eth1 (X1) **Controller Set-up**

The controller settings for Eth1 can now be entered.

Any existing Eth1 settings should be recorded before any changes are made.
IP address restrictions- IP addresses **192.168.215.xxx** & **192.168.214.xxx**. are reserved for use by the system. Addresses within these ranges should not be used for Eth1.

When the settings have been entered select the “OK” soft-key”



Allow the controller to update the TCP/IP configuration.

The setting will only become active after RESET (po).



Switch the control off/on to activate the settings.

Notes:

TCP/IP configuration

	PCU system network ETH2	PCU company network ETH1	NCU system network X120	NCU company networ
Availability	100.00%	100.00%	100.00%	0.00%
Computer name	DESKTOP-0GKOHJU		ncu1	
DNS name	DESKTOP-0GKOHJU.local	-	ncu1.local	-
MAC address	20:87:56:70:36:EB	20:87:56:70:95:D5	00:1f:f8:41:fc:2d	00:1f:f8:41:
Address type	DHCP	Manually	DHCP - Synchronized server (Default)	DHCP - Cli
IP address assigned	192.168.214.241	192.168.100.1	192.168.214.1	-
Subnet mask assigned	255.255.255.0	255.255.255.0	255.255.255.0	-
DHCP server	-	-	-	-
Status DHCP server	-	-	Active	-
Mode DHCP synchron.	High	-	High	-
DNS server 1	192.168.214.1	fec0:0:0:ffff::1%1	127.0.0.1 (localhost)	-
DNS server 2	-	fec0:0:0:ffff::2%1	-	-
Gateway	-		192.168.214.241	

When the controller has powered up, navigate to the TCP/IP screen.
Following key selection:



The correct IP address and subnet mask will be shown, as above.

The “Availability” of the network connection is shown. In the example below Eth2 (X2) has a green/white tick symbol with an availability of 100%, this indicates that the connection has hardware attached and is healthy.
If Eth1 (X1) shows a red / white cross symbol, with an availability of 0%. Then it is due to the fact that no hardware connection exists to Eth1 or this is disturbed.
The Display is not dynamic! it can be updated with the SK ”Display new“ .

Eth2: Connection in use

TCP/IP configuration

	PCU system network ETH2	PCU company network ETH1	NCU system network X120	NCU company networ
Availability	100.00%	0.00%	100.00%	0.00%
Computer name	DESKTOP-0GKOHJU		ncu1	
DNS name	DESKTOP-0GKOHJU.local		ncu1.local	-
MAC address	20:87:56:70:36:EB	20:87:56:70:95:D5	00:1f:f8:41:fc:2d	00:1f:f8:41:
Address type	DHCP	Manually	DHCP - Synchronized server (Default)	DHCP - Cli
IP address assigned	192.168.214.241	192.168.100.1	192.168.214.1	-
Subnet mask assigned	255.255.255.0	255.255.255.0	255.255.255.0	-
DHCP server	-	-	-	-
Status DHCP server	-	-	Active	-
Mode DHCP synchron.	High	-	High	-
DNS server 1	192.168.214.1	-	127.0.0.1 (localhost)	-
DNS server 2	-	-	-	-
Gateway	-		192.168.214.241	

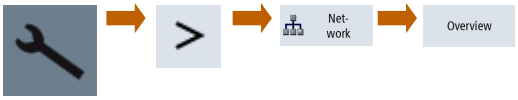
Eth1: Connection not in use

Network settings overview

PCU

Adapter		IP address	Subnet mask	MAC address
Eth1	Company network	192.168.100.1	255.255.255.0	20:87:56:70:95:D5
Eth2	Systems network	Requested		20:87:56:70:36:EB
		assigned 192.168.214.241	255.255.255.0	

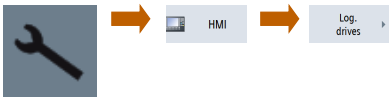
The IP and subnet mask can also be checked in the SETUP AREA.
Following key selection:



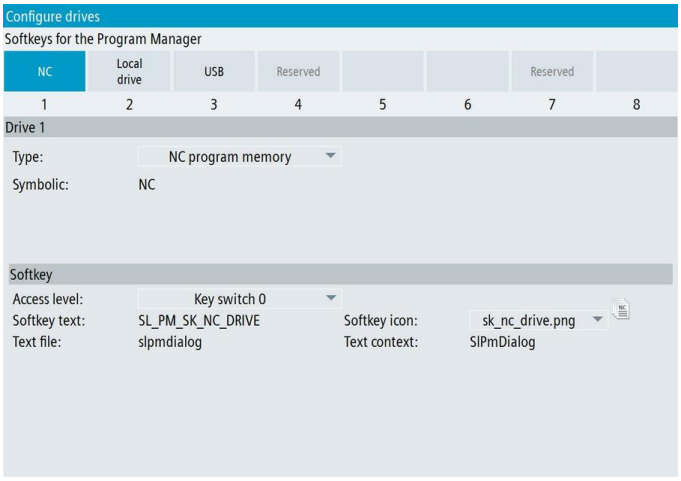
The correct IP address and subnet mask will be shown, as shown above.

Notes:

Setting up a logical drive on the IPC.
Locate the logical drives set-up area:

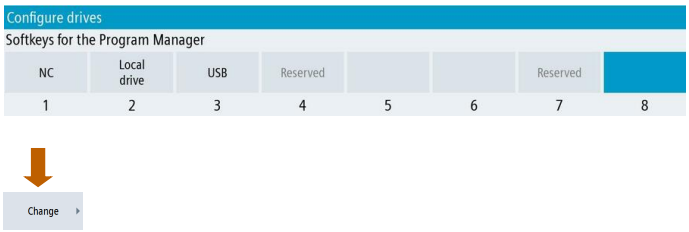


Default Logical drives screen:

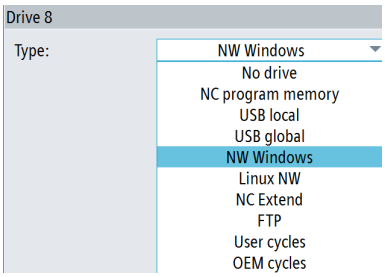


Select a free soft-key. Further soft-keys, 24 in total, are accessed via the ***>>Level*** soft-key.

Example - Soft-key 8:



Choose the ***NW Windows*** option:



For this example the following data is required (the external computer settings will be changed later):

Computer name: 192.168.100.2
(IP address of the PC)

Enabling name: Programs
(Name of the shared folder on the PC).

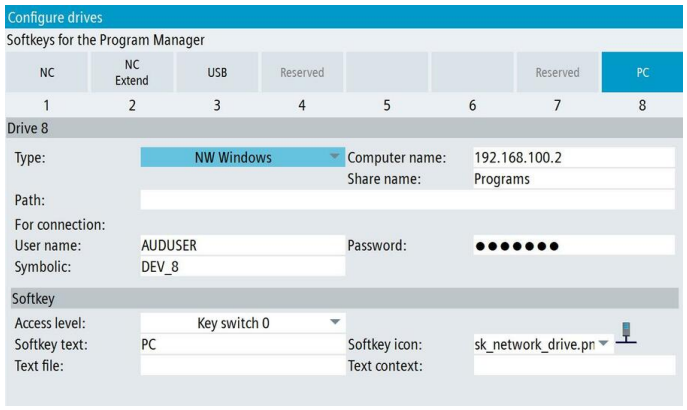
User name: AUDUSER
(Login name of the PC)

Password: SUNRISE
(Login password of the PC, for example)

Soft-key text: PC
(the text that will appear on the soft-key in the Program manager area).

The remaining parameters are to be left unchanged for this example.

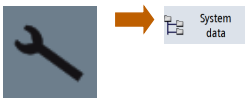
The relevant data has to be entered:

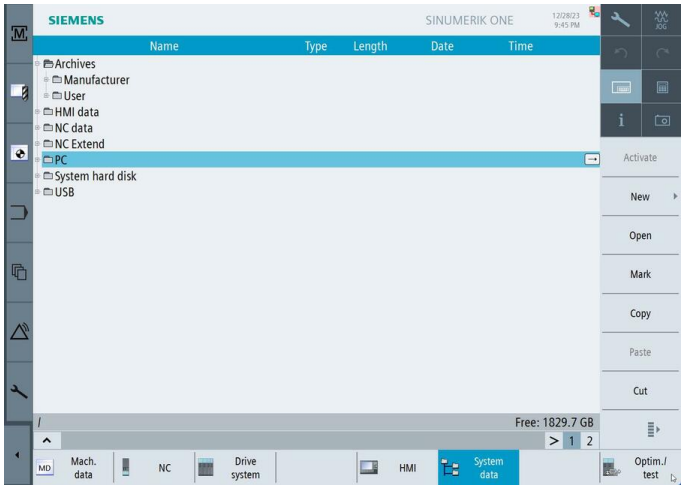




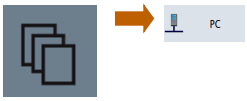
Notes:

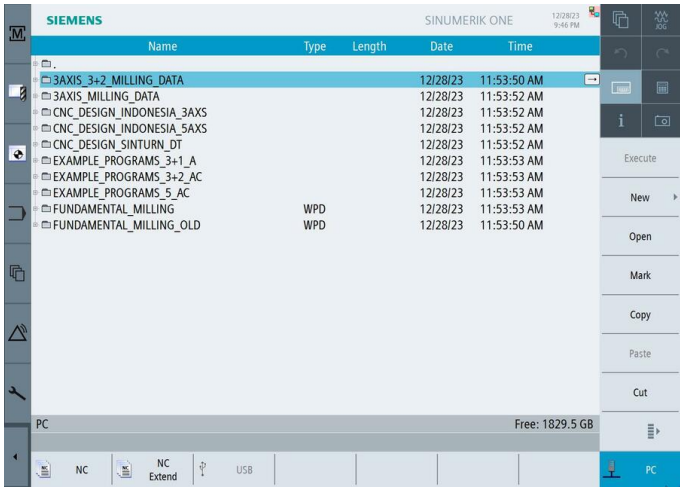
It should now be possible to access the shared folder from the ***System data*** area of the controller.





And the **Program manager** area of the controller.





Global part program memory - General information.

It is also possible to execute a program from a logical drive however, there are limitations with program jumps and block search, for example.

NC Extend, EES, and Global Part Program Memory.

Options are available which allow full program functionality for programs stored externally on the controller.

NC extend will give “Global Part Program Memory” with access to released 100MB storage.

EES will give “Global Part Program Memory” with access to the storage devices of USB, HMI user memory, and networked computer/server.

ONE Hardware	NC Memory 10MB	NC Extend 100MB	HMI user 6GB	EES
PPU 1740	✓	✓	✓	✓
NCU 1740	✓	✓	✓	✓
NCU 1750	✓	✓	✓	✓
NCU 1760	✓	✓	✓	✓

ADD. HMI user memory card of NCU:
Releases unused free memory of System SD card memory which can be used for storage of programs..
(Option number: **6FC5800-0BP12-0YB0**)

CNC user memory extend:
Releases 100MB of System SD card memory.
(Option number: **6FC5800-0BP77-0YB0**)

Execute from External Source (EES):
Suitable for connecting via Ethernet to an external PC for unlimited storage capacity.
(Option number: **6FC5800-0BP75-0YB0**)

Global Part Program Memory:
The Global Part Program Memory feature behaves like standard NC memory and has no restrictions on program jumps, block search etc.
The feature can be activated for ***one*** of the following memory types:

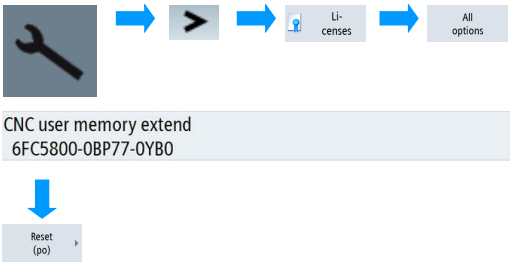
- System SD card (NC Extend).
- USB (EES).
- Networked computer (EES).
- IPC (EES).

Notes:

NC Extend Function.

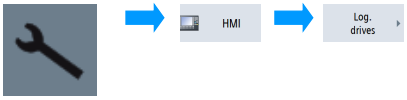
The ***NC Extend*** function releases 100MB of System SD card memory which, when used with the **Global part program memory** feature, behaves exactly like standard NC memory. i.e. there are no restrictions on block search, long program jumps etc. that are found with other methods of executing programs from external storage.

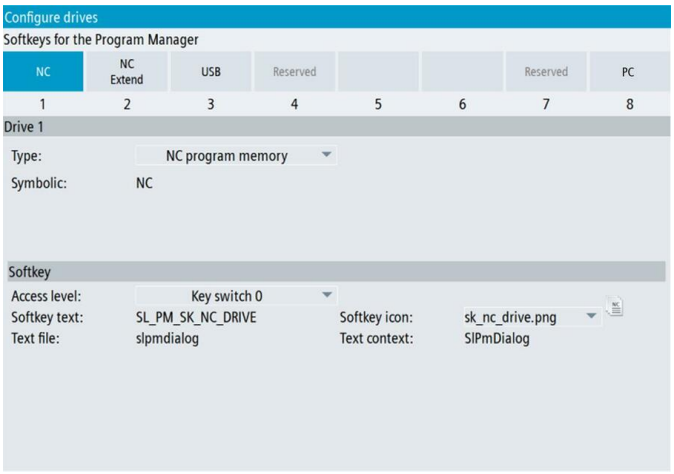
The option - **6FC5800-0BP77-0YB0 -** needs to be set.



Activate the option with an NCK reset:
After the reset it will be possible to select the ***NC Extend*** soft-key in the Program Manager area

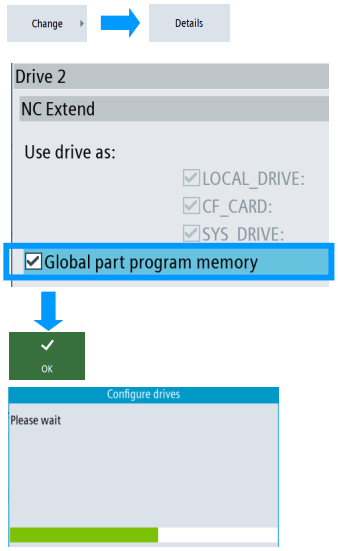
The settings for the NC Extend can be viewed in the Logical drives area.





The default settings, e.g. soft-key text can be changed if required.

Setting the Global part program memory:

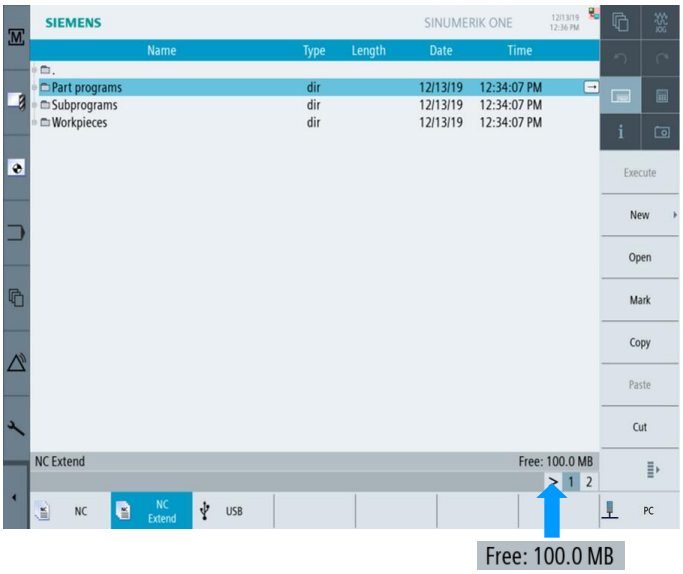


Only one Global Part Program Memory can be active, therefore, If it was already active on another logical drive it will be automatically deactivated for that drive.

The result of the Global part program memory being set is the creation of the folder structure on the NC Extend drive:



The ***NC Extend*** area has 100MB of additional memory for storage of part programs/sub-programs/workpieces.



The available space, in this case on the System SD card, can be seen at the bottom right of the screen.

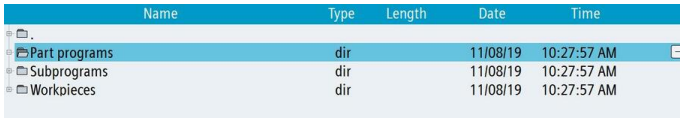
Notes:

The result of activating the Global part program memory is the creation of three folders which match the default NC memory folders.

Navigate to the Program manager screen:

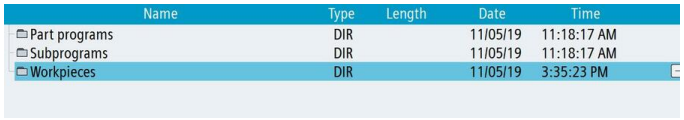


The folder structure that was created by the Global part program memory.



This is the same as the folder structure in the NC memory area.



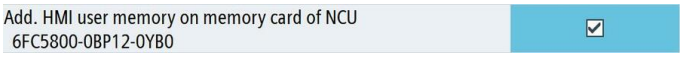


HMI User memory Function.

The ***HMI User memory*** function releases reserved memory of System SD card memory which can be used for storage of programs.

The option - **6FC5800-0BP12-0YB0 -** needs to be set.



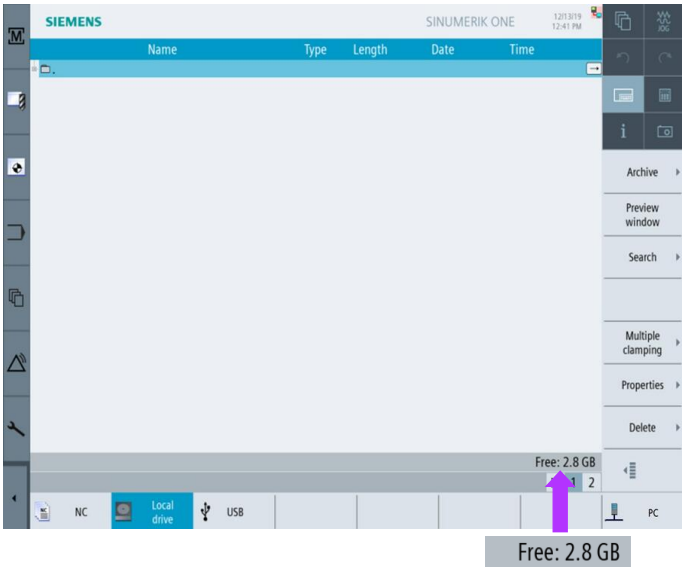




Activate the option with an NCK reset:
After the reset, the ***Local drive*** soft-key is displayed in the Program Manager area.



The ***Local drive*** area has 2.8GB of additional memory for storage of programs.



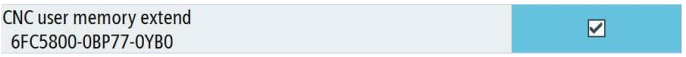
The available space, in this case on the System SD card, can be seen at the bottom right of the screen.

If the NC Extend function is used with the HMI user memory function, when used in conjunction with the **Global part program memory** feature, it will behave exactly like standard NC memory.

i.e. there are no restrictions on block search, long program jumps etc. that are found with other methods of executing programs from external storage.

The option - **6FC5800-0BP77-0YB0 -** needs to be set.



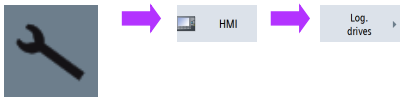


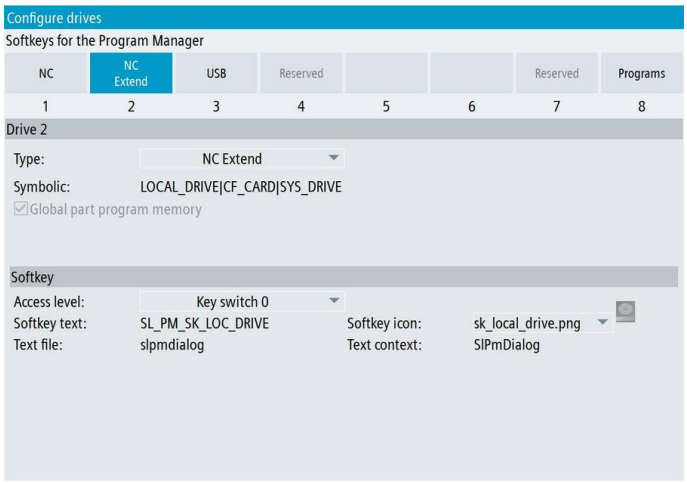


Activate the option with an NCK reset:
After the reset it will be possible to see the ***NC Extend*** soft-key in the Program Manager area

Notes:

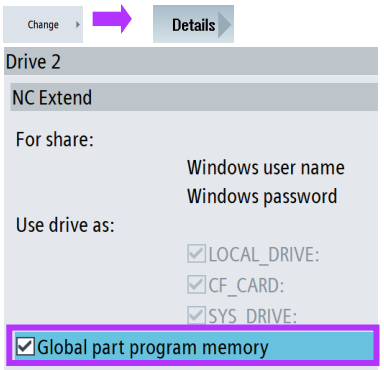
Example - setting the Global part program memory for the NC Extend memory.

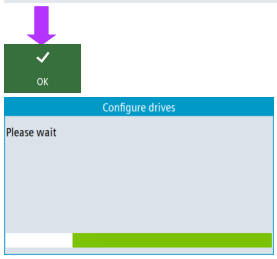




The default settings, e.g. soft-key can be changed if required.

Setting the Global part program memory:



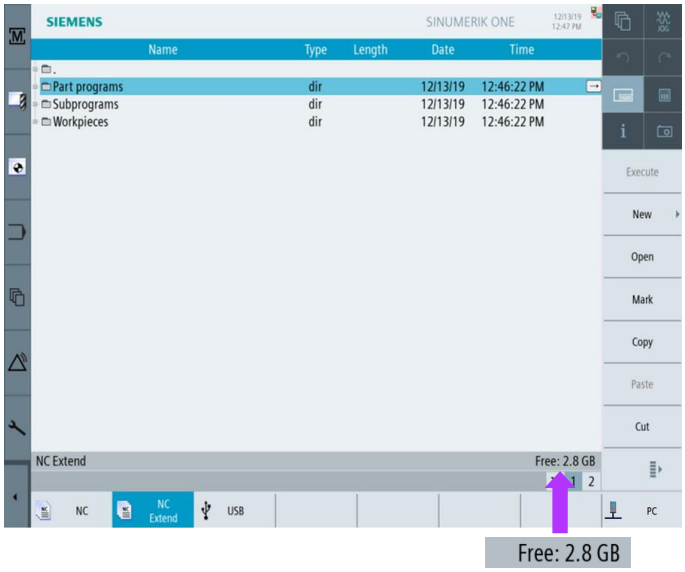


Only one Global Part Program Memory can be active, therefore it needs adding on another logical drive it will be automatically deactivated for that drive.

The result of the Global part program memory being set is the creation of the folder structure on the NC Extend drive:



The ***Local drive*** area has 2.8GB of additional memory for storage of programs.



The available space, in this case on the System SD card, can be seen at the bottom right of the screen.

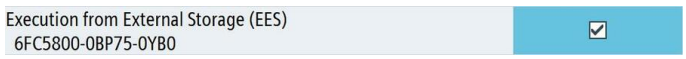
Execute from an External Source Function

The EES (Execute from External Source) function allows a networked logical drive with the Global part program memory setting. The size of available memory is limited only by the free space available on the SD card, IPC SSD or free space available in the shared folder in the networked PC.
It is also possible to use the EES function on a USB memory device.

When the option is used in conjunction with the Global part program memory setting, it behaves exactly like standard NC memory. i.e. there are no restrictions on block search, long program jumps etc., that are found with other methods of executing programs from an external source. It is also possible to edit the program in NC stop.

Activating the EES Function:

The option, 6FC5800-0BP75-0YB0, needs to be set.



Activate the option with an NCK Reset.



Notes:

Once the option is set and activated it can be decided which logical drive is to be used.
This can be:

- IPC
- USB memory device
- Networked computer.

Setting the Global part program memory

Locate the Logical drives area:

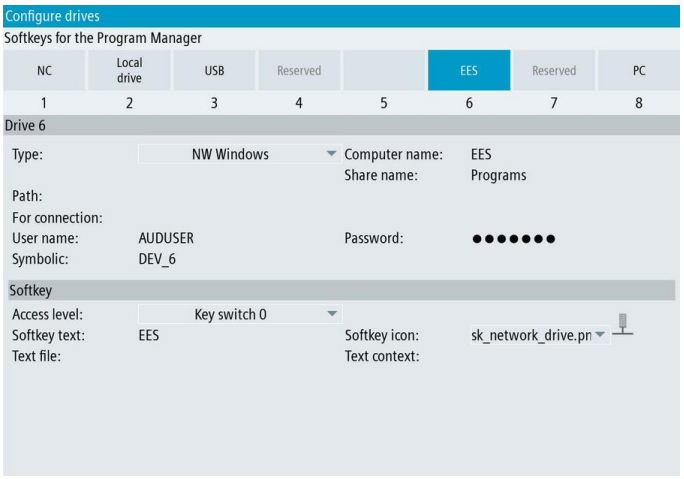


The global part program memory setting can be applied to either:

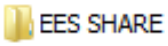
- IPC
- A USB device
- A networked computer.

Example - Setting the Global part program memory for a networked computer.

For the example a new networked logical drive (***EES SHARE***) has been assigned to soft-key 6:



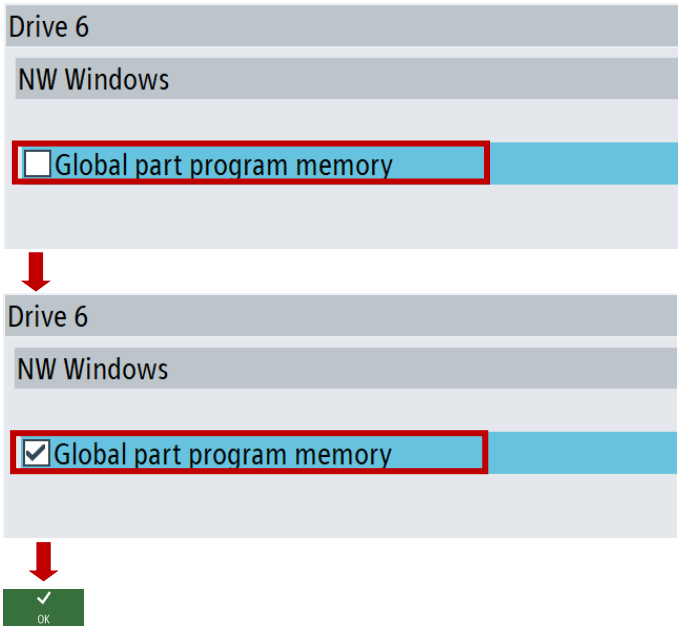
The ***EES SHARE*** folder has been created and shared on the networked computer:



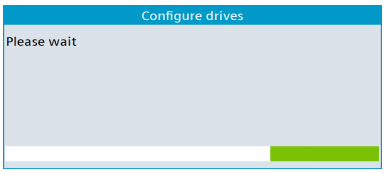
From within the Logical drives area, select the ***EE***S logical drive, followed by the ***Change*** and then ***Details*** soft-keys.



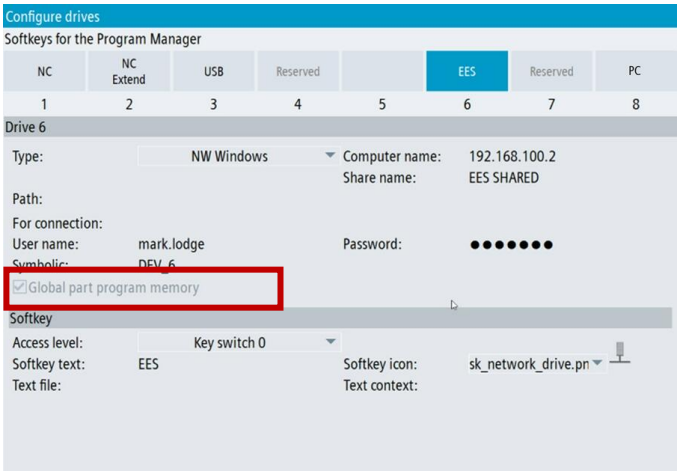
The Global part program memory can now be set.



The EES SHARE Logical drive will now be activated for Global part program memory.

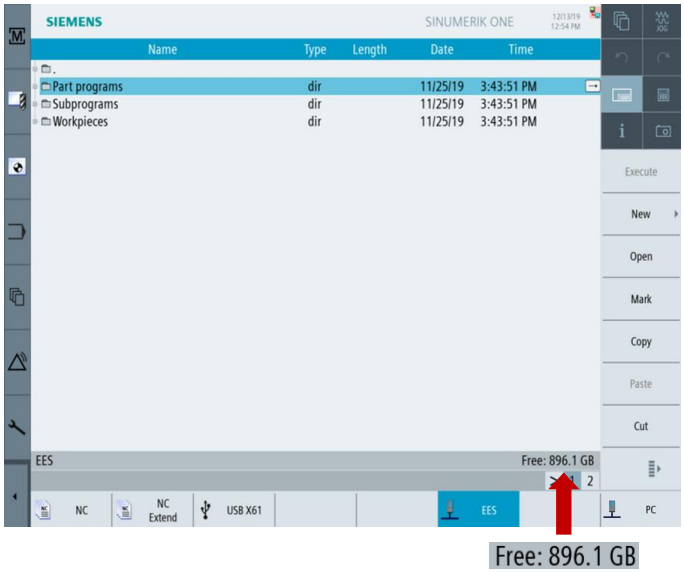


Confirmation the Global part program memory is active is seen on the Logical drive page for that particular logical drive:



Notes:





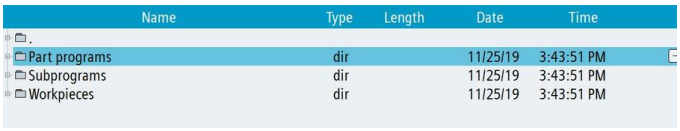
The available space, in this case the networked computer's hard disk, can be seen at the bottom right of the screen.

The result of activating the Global part program memory is the creation of three folders which match the default NC memory folders, also whether this drive is a local drive, USB or a networked drive will become the "MASTER NC MEMORY".

Navigate to the Program manager screen:

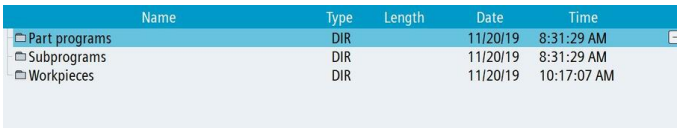


The folder structure that was created by the Global part program memory.



This is the same as the folder structure in the NC memory area.

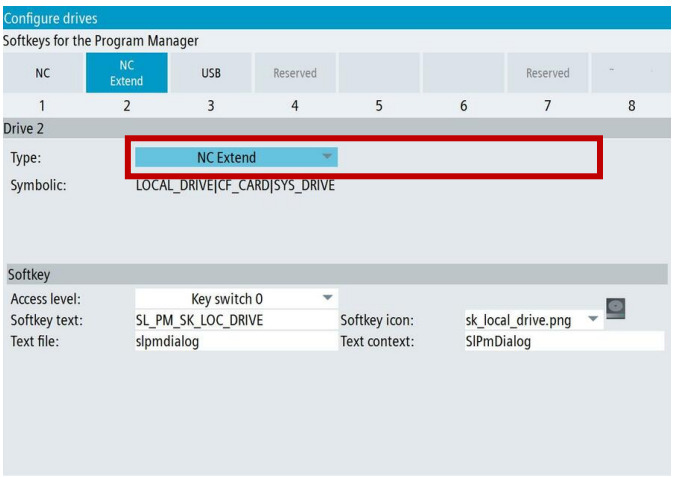




Example - Setting the Global part program memory for an IPC.

For the example a logical drive (***NC Extend***) has been assigned to soft-key 2:

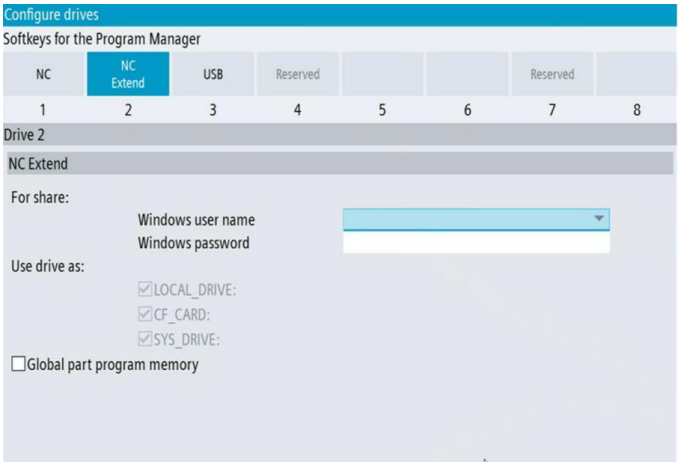




Make sure "NC Extend" is selected.

From within the Logical drives area, select the ***NC Extend*** logical drive, followed by the ***Change*** and then ***Details*** soft-keys.





The "windows user name" can now be set.

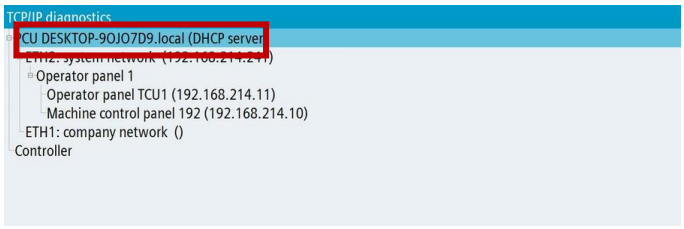


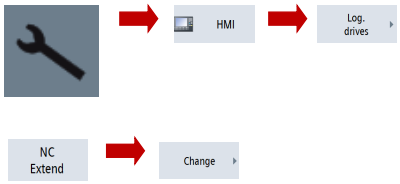
Notes:

In the case an IPC has been installed, the name of the windows account (WINDOWS-xxxxxxx/auduser) is required to setup Global part program memory.

This information is shown in the Bus TCP/IP window:





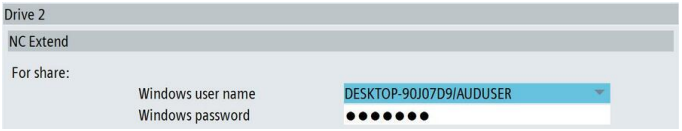


The “windows user name” can be entered.

In this example “DESKTOP-90J07D9”

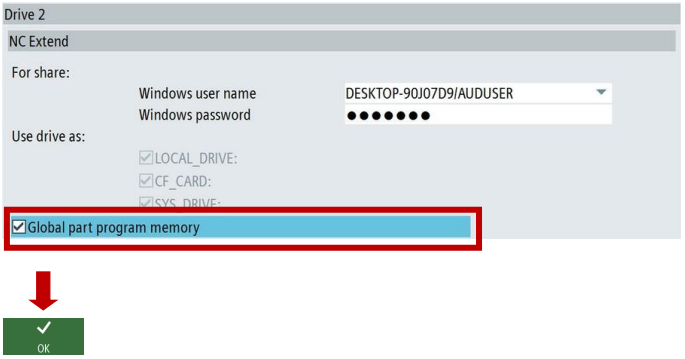
Followed by the “windows password”.

SUNRISE

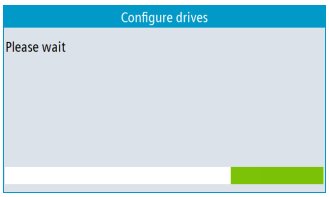


The Global part program memory can now be set.

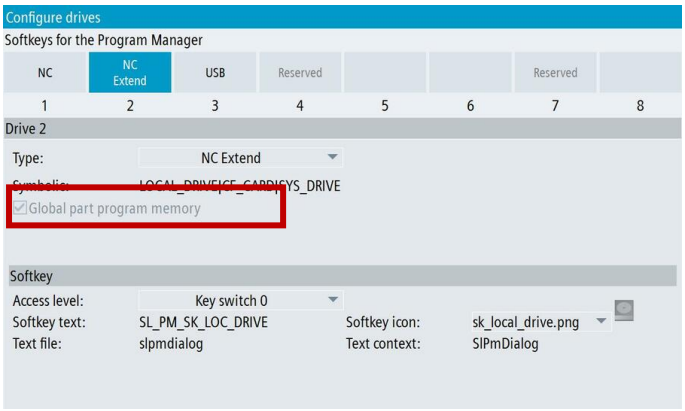




The NC Extend Logical drive will now be activated for Global part program memory.

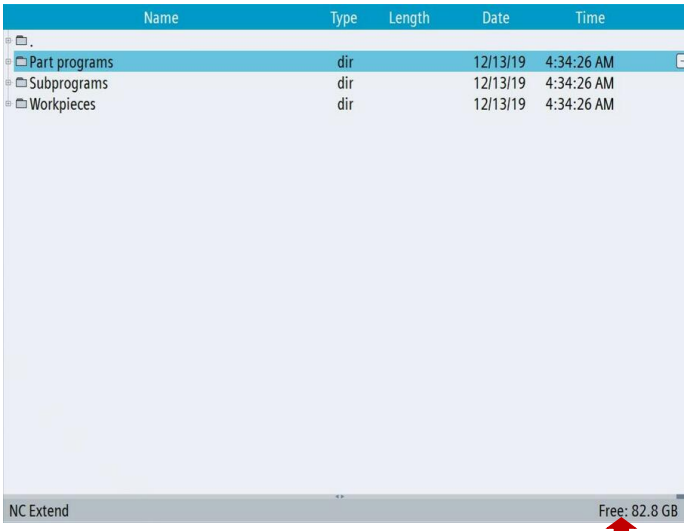


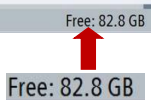
Confirmation the Global part program memory is active is seen on the Logical drive page for that particular logical drive:



Notes:





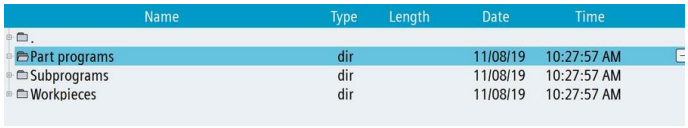


The available space, in this case the networked computer's hard disk, can be seen at the bottom right of the screen.
Note: path to the folders
C:/program files/Siemens/MotionControl/User/Sinumerik/data/Prog

Navigate to the Program manager screen:

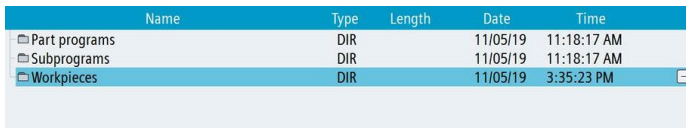


The folder structure that was created by the Global part program memory.



This is the same as the folder structure in the NC memory area.

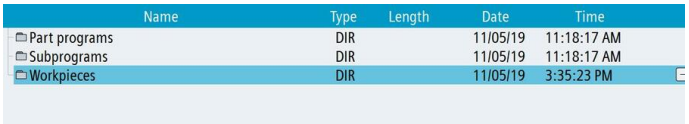




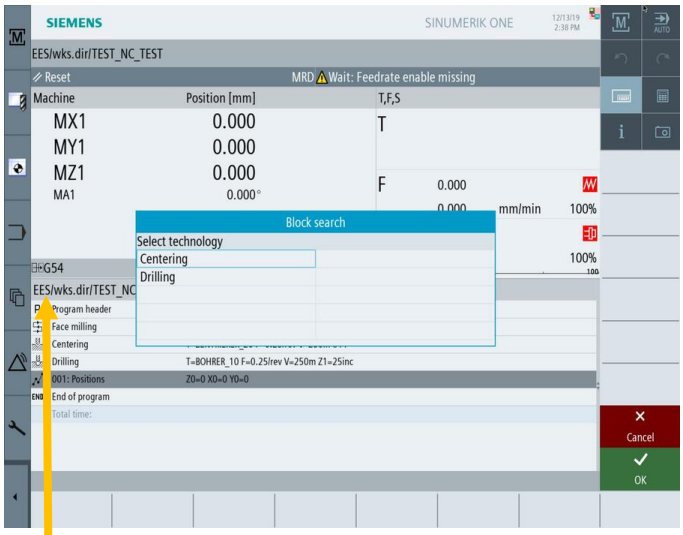
Program Use With EES & NC EXTEND

Part programs, subprograms etc. can now be stored on the external source or on the NC Extended memory and used in the same way as the NC memory.

Activating the Global part program memory results in the creation of three folders (Part program, Subprogram, Workpieces) which match the default NC memory. This is now classed as the MASTER NC MEMORY and applies whether an IPC, local drive, USB or a net-worked drive is declared as the Global part program memory.
The result of activating NC Extend is the unrestricted ability to "Block Search" programs stored on the SD card/IPC/external PC/server.



Three folders which match the default NC memory folders.

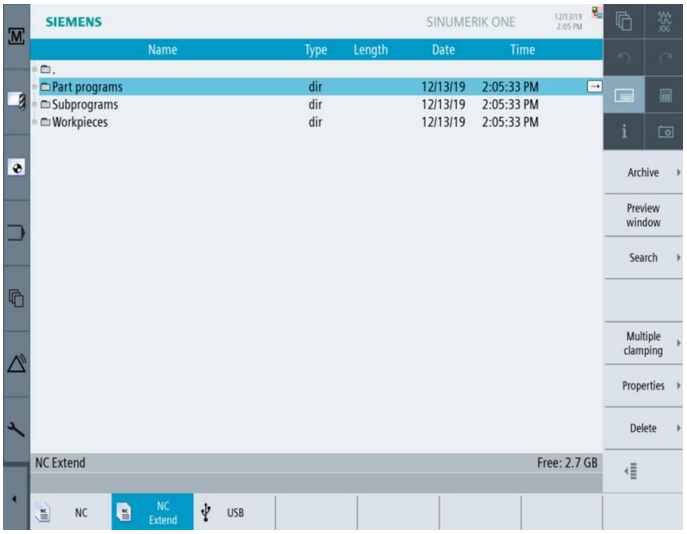


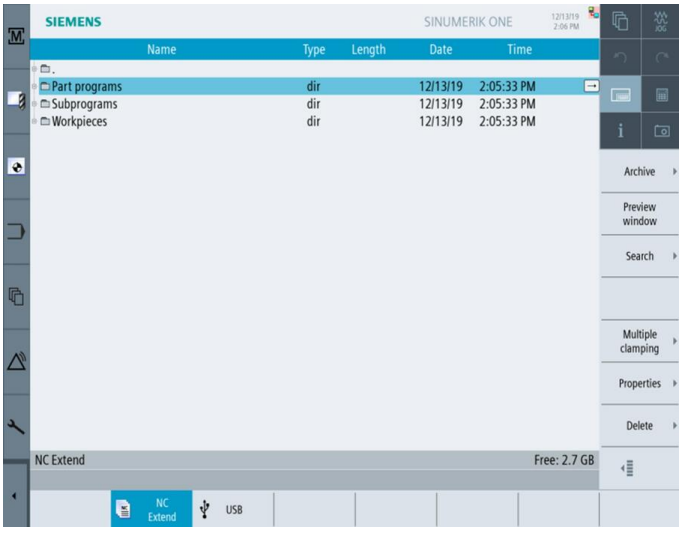
Block search being performed from an external drive, in this case a directory called EES on a PC.

Siemens would recommend that the "NC Memory" softkey needs to be removed, as shown in the examples.
Note: remove soft-key in the AREA: Setup -> HMI -> Log.drives

Notes:

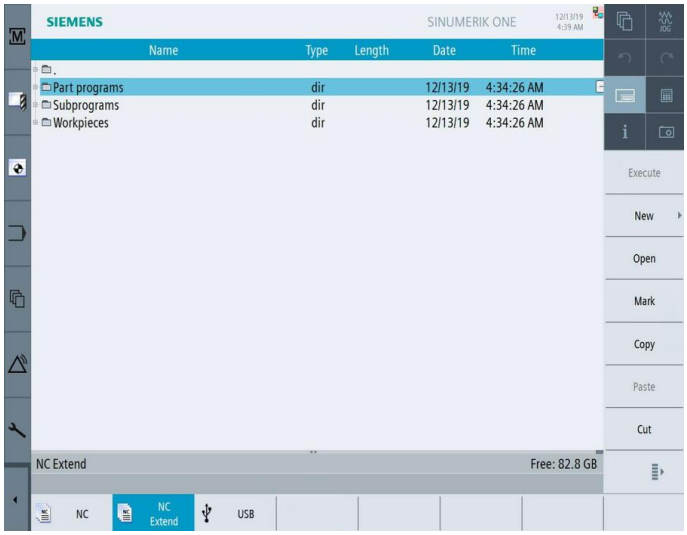
NCU configuration

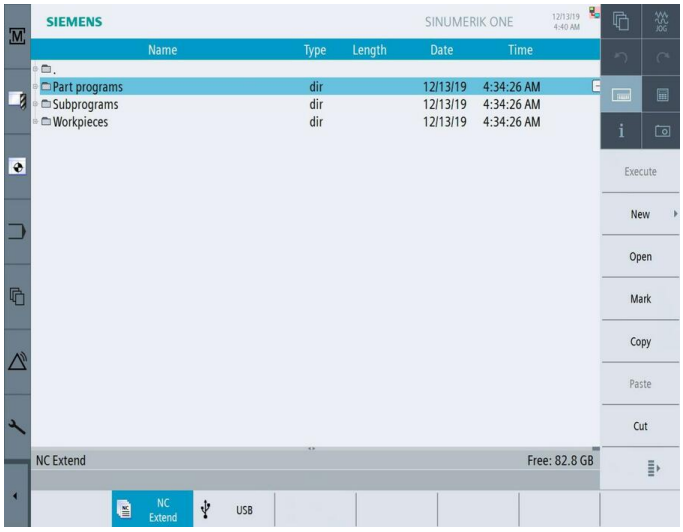




"NC" softkey removed so that there is only ONE "MASTER NC MEMORY".

IPC configuration

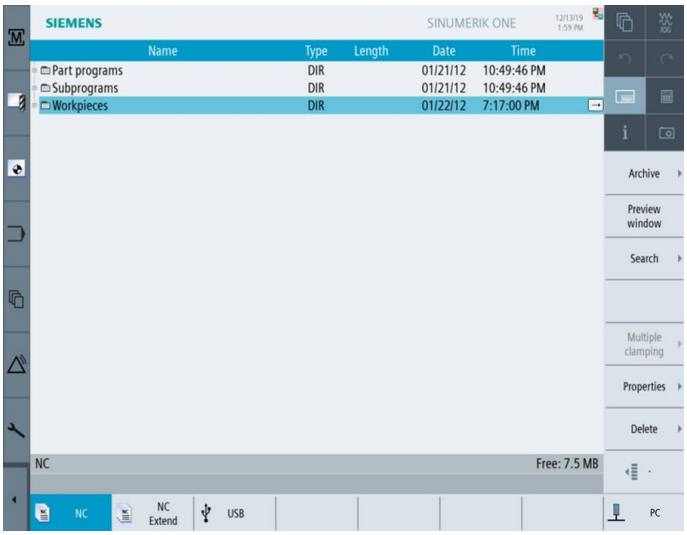


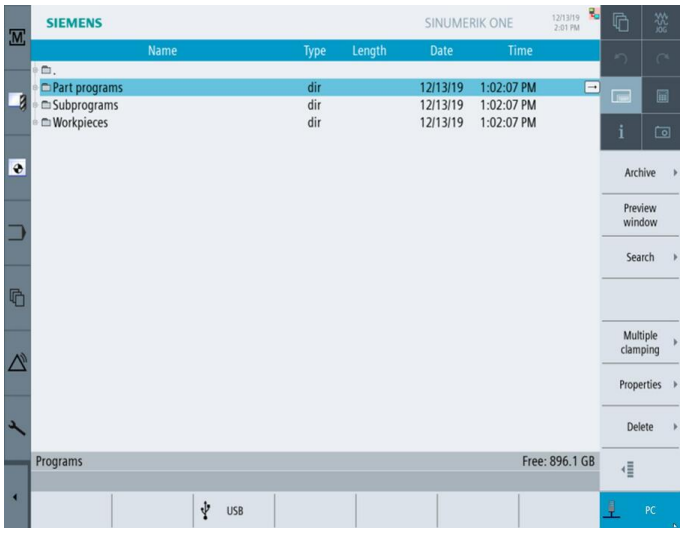


"NC" softkey removed so that there is only ONE "MASTER NC MEMORY".

Notes:

External PC/server configuration





"NC" and "NC Extend" softkey removed so that there is only ONE "MASTER NC MEMORY".

A127: END

Notes:

Notes:

129

SINUMERIK

EasyXML and SideScreens

Module Description:

EasyXML is a function of the SINUMERIK which allows the manufacturer to write custom screens. The OEM screens are intended as an interface to the specific functions offered by the OEM.

Sidescreen is a function of the SINUMERIK with a Multitouch panel, which allows the manufacturer to activate an alpha/numeric keyboard or write an OEM control panel for the Sidescreen.
The OEM Sidescreens are intended as an interface to specific functions offered by the OEM.

Module objective:

Upon completion of this module you will be able to perform the following tasks:

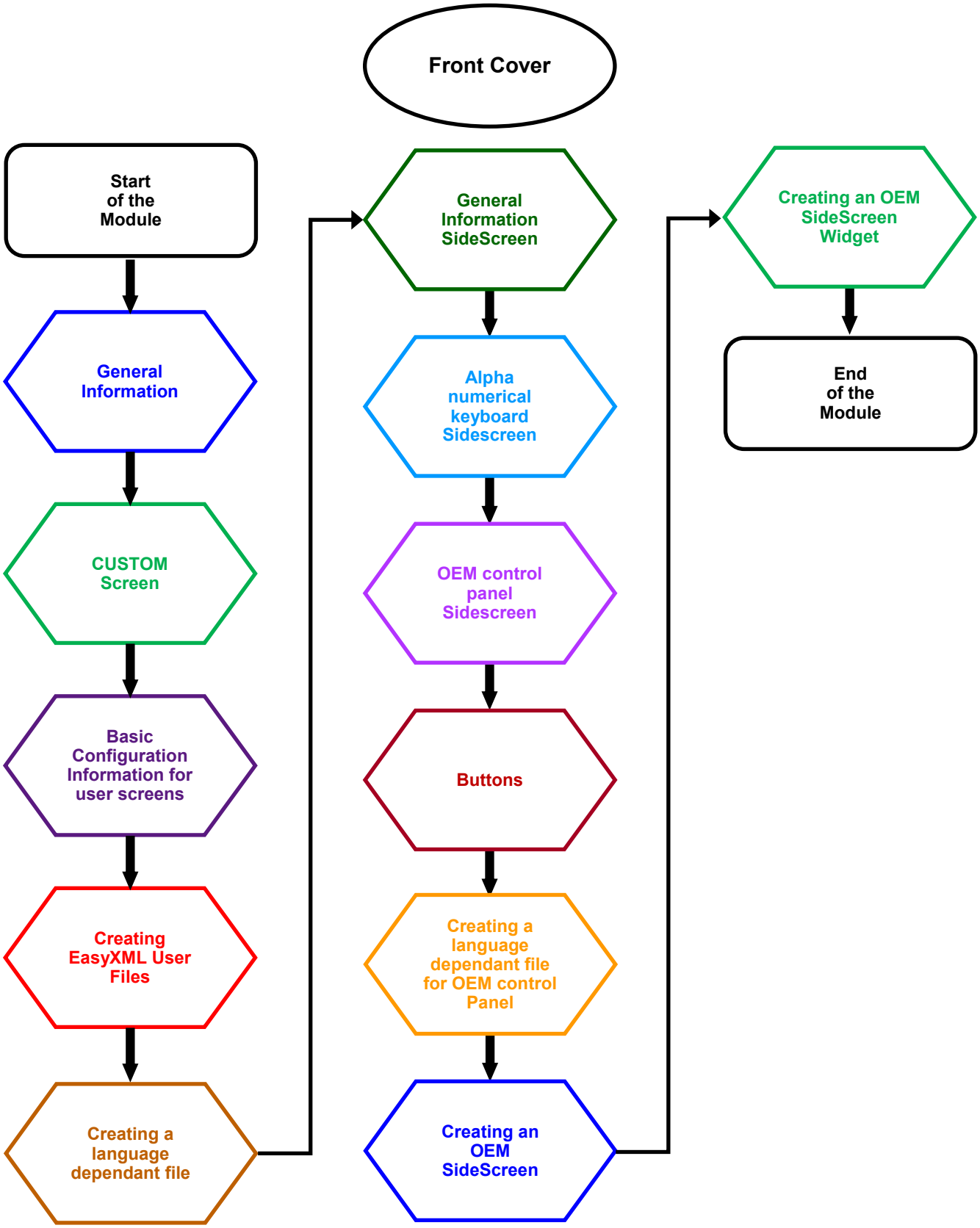
- Create additional display screens using EasyXML.
- Create a Sidescreen using EasyXML.
- Create Language files

Content:

Sinumerik

General Information
Custom screen
Basic Configuration Information
Creating EasyXML User Files
Language Dependent Texts
General Information Sidescreens
Alpha numerical keyboard Sidescreen
OEM control panel Sidescreen
Buttons
Creating a language dependant file for OEM control Panel
Creating an OEM SideScreen
Creating an OEM SideScreen Widget

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
General Information
CUSTOM Screen
Basic Configuration Information for user screens
Creating EasyXML User Files
Creating a language dependant file
General Information SideScreen
Alpha numerical keyboard Sidescreen
OEM control panel Sidescreen
Buttons
Creating a language dependant file for OEM control Panel
Creating an OEM SideScreen
Creating an OEM SideScreen Widget
End of the Module

General Information

The EasyXML function is configured using *.XML files. These files contain the descriptions of the user interface.

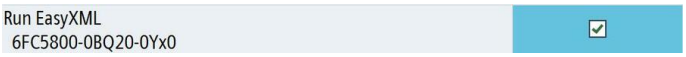
In order to use images in the screens, a graphic programme is required, e.g. Paint Shop Pro® and Paint® are programmes which are compatible with the SINUMERIK controller.

EasyXML can be used for the following functions:

1. Display of dialogs and provision of:
   - Soft-keys
   - Variable tables (this function will be available with a later level of implementation)
   - Text and help text
   - Graphic displays and help displays
2. Calling up dialogs by means of:
   - The (entry) soft-keys
   - PLC selection
3. Dynamic re-structuring of dialogs:
   - Changing or deleting soft-keys
   - Defining and designing variable fields Showing, replacing and deleting (language-dependent/independent) display texts
4. Initiating actions when:
   - Showing dialogs
   - Entering values (variables)
   - Pressing soft-keys
   - Closing dialogues
5.. Exchange of data between dialogs.
6. Variables
   - Reading (NC, PLC, user variables)
   - Writing (NC, PLC, user variables)
   - Linking with mathematical, comparative or logic operators
7. Executing functions:
   - Subroutines
   - File functions
   - PI Services
8. Taking into account levels of protection defined for user groups.

Activating the "**Run EasyXML**" Function:

The option, **6FC5800-0BQ20-0Yx0**, needs to be set.



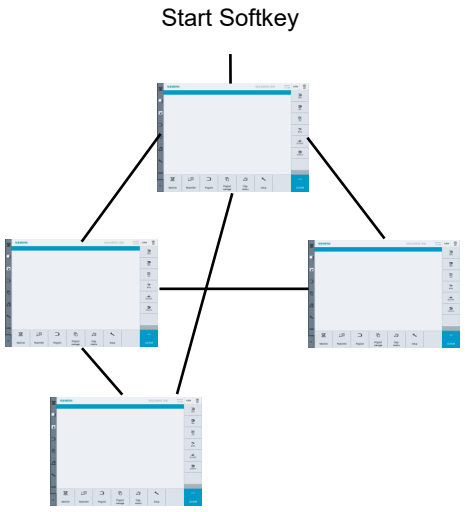
Activate the option with an NCK Reset.



Structure of the menu tree

Menu tree principle:

Several interlinked dialogs create a menu tree. A link exists if you can switch from one dialog to another. You can use the newly defined horizontal/vertical softkeys in this dialog to call the preceding or any other dialog.
A menu tree can be created behind each start softkey:

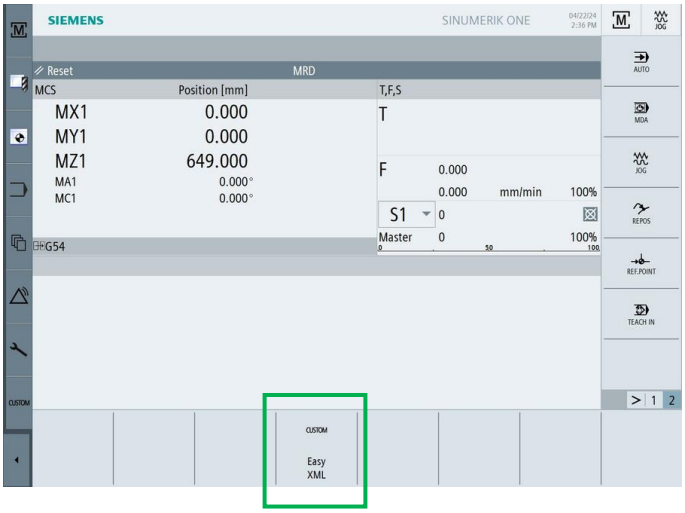


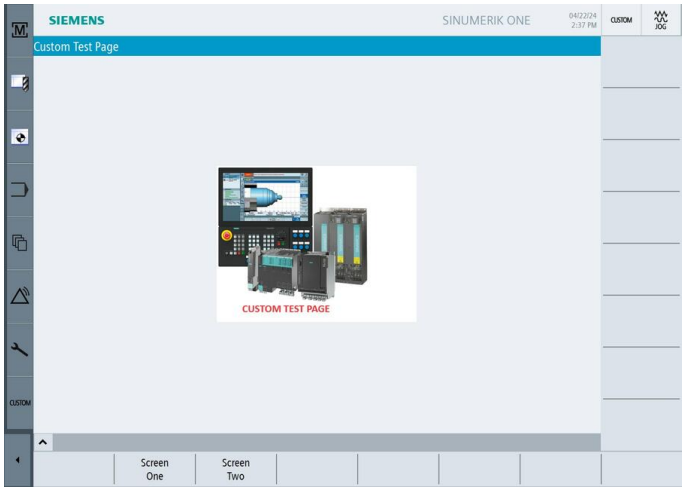
Notes:

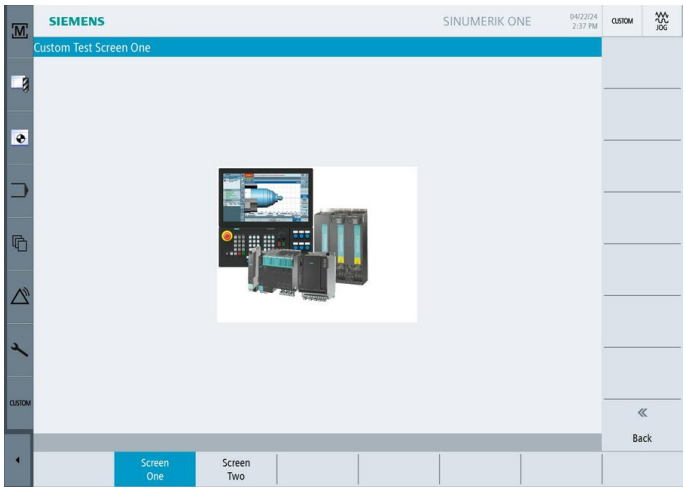
CUSTOM Screen

The “CUSTOM softkey is an extra “AREA” softkey so that the OEM can create there own dialogs or additional softkey bars and can be loaded and used for executing additional actions.

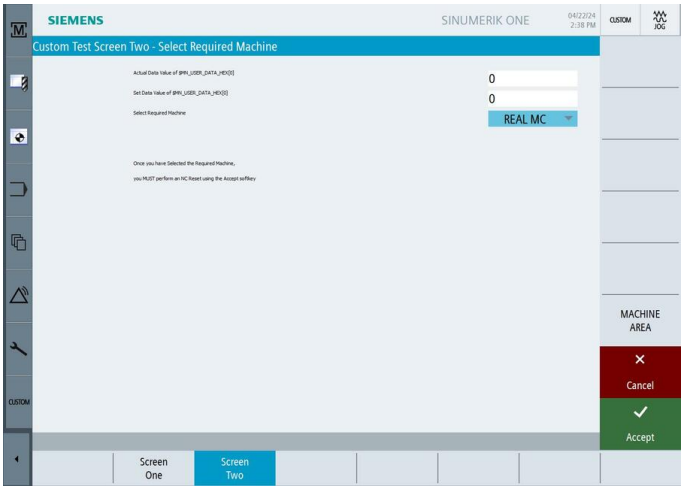
This pictures below shows the objective of this section.





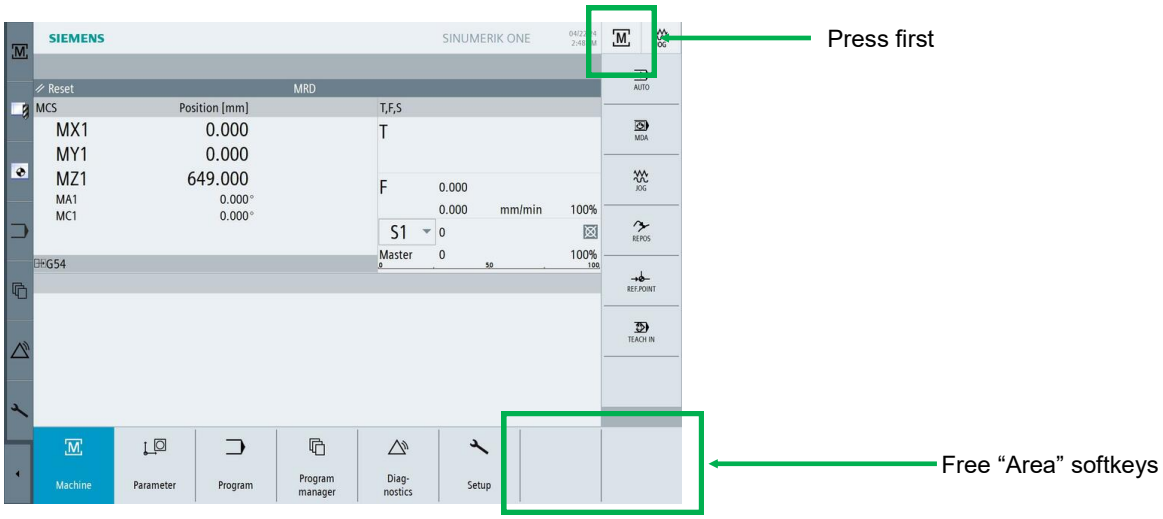


This example applicable to “SINUMRIK ONE” ONLY.



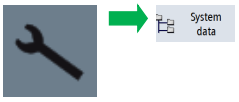
Notes:

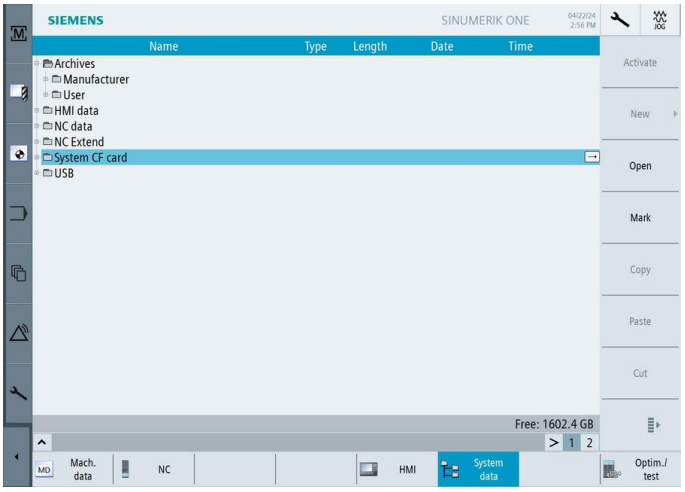
The start point for the “CUSTOM” screens is the “menu select “ (main) area as shown below.



The "xmldial.xml" file is pre-named by the system and is automatically called when the softkey is activated, this is used to initiate the OEM operating sequences.
The initial menu is defined by the form name "main" in the "xmldial.xml" file. You can use the "main" menu to link additional dialogs or additional softkey bars so that they can be loaded and used for executing additional actions.

A configuration file required for the EasyXML function, needs to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate in the “System data” area.





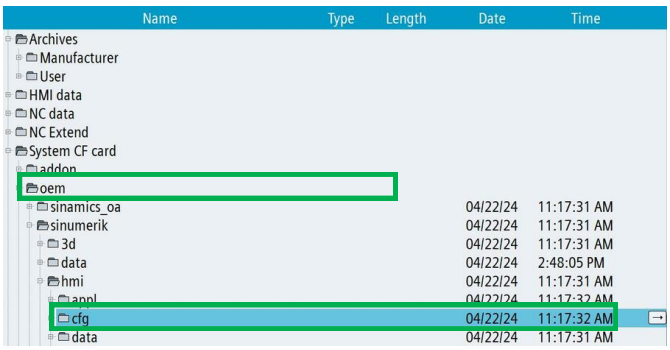
Creating the “slamconfig.ini” file on the controller

This file activates the softkey in the “Area” and activates the xmldial.xml file which is to be used to create the user screens, it also determine which operating areas these files are available in. The file has to be located in one of the following paths/ directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/cfg

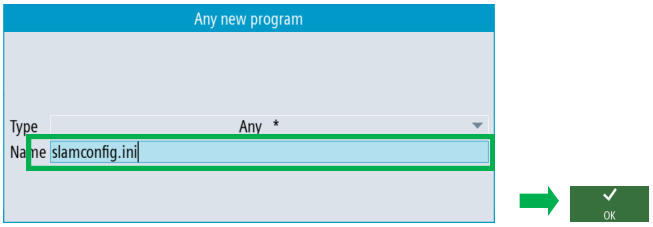
Use the cursor keys to locate and open this directory.



Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing, enter the text below.

Creating the User Files on an external PC

As mentioned previously it is possible to create the EasyXML files directly on the controller or offline using an ASCII editor (e.g. Notepad®).
These files have to be transferred to the controller and placed in the correct folders using the USB/CF/SD Interfaces or with Access MyMachine software.

Example: **slamconfig.ini** file.

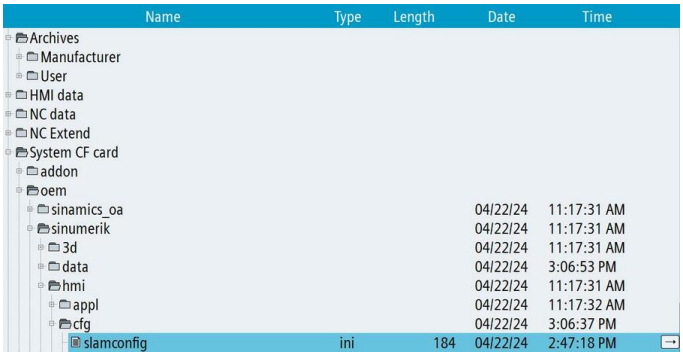
It is necessary to specify a soft-key to be used as the entry point to the “MENU” area.

The following example will show content of the “**slamconfig.ini** ” file for the “OEM Custom Area of the controller.

```
[CustomXML]
textid=Easy%nXML
picture=hr_custom.png
softkeyposition = 12
visible=true
```

The example defines the 12th horizontal soft-key as the entry point, the soft-key will have the text “Easy XML”.
The contents of the “FORM” can be created using the file **xmldial.xml.**





Creating the xmldial.xml file

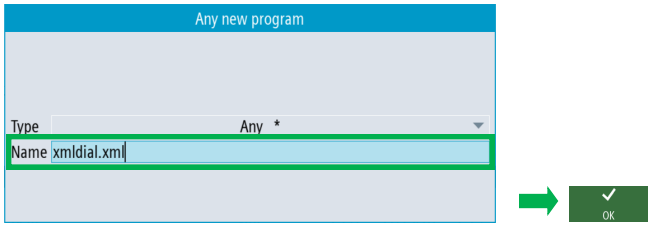
The file needs to be created and placed in one of the following folders:

SINUMERIK Data - CF Card/oem/sinumerik/hmi\appl

IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/appl

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.



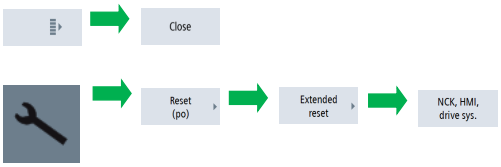


Notes:

Example: xmldial.xml file
The following example will show content of the "**xmldial.xml** " file for the "CUSTOM" screen in the controller.

```
<dialoggui>
<menu name = "main">
<open_form name = "custom_main" />
</menu>
<form name = "custom_main">
<init>
<caption>Custom Test Page</caption>
</init>
<paint>
<img name = "f:/ico1024/custom1.png" xpos="181" ypos="129" height="155" width="154" aspectratiomode="keepbyexpanding" />
</paint>
</form>
```

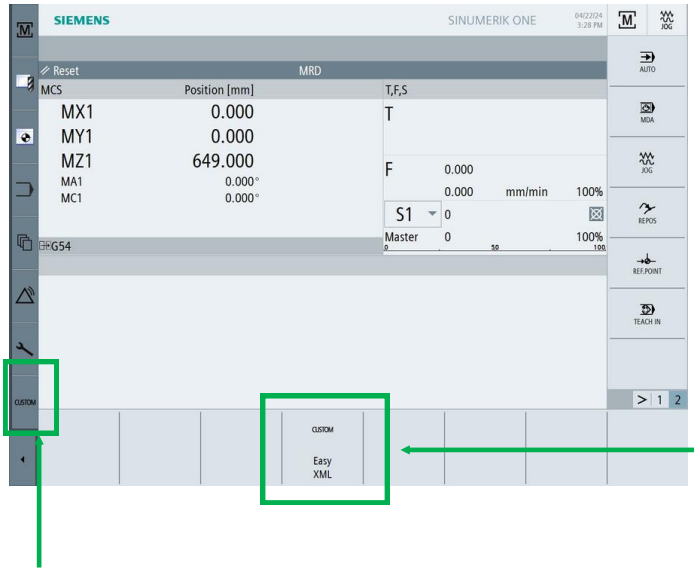
To activate the files, perform:



To see the result of the example the "CUSTOM" Area softkey has to be located on the control.



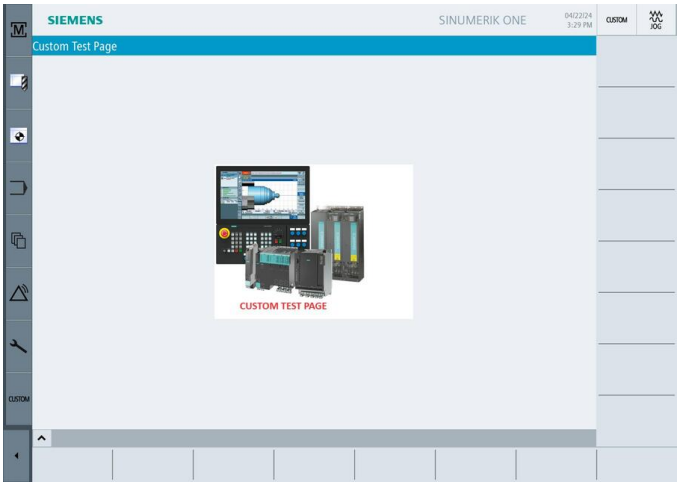
The 12th horizontal soft-key contains the text specified in the "**slamconfig.ini**" file: `HS12="Easy%nXML".`



The new "CUSTOM" softkey MAIN area

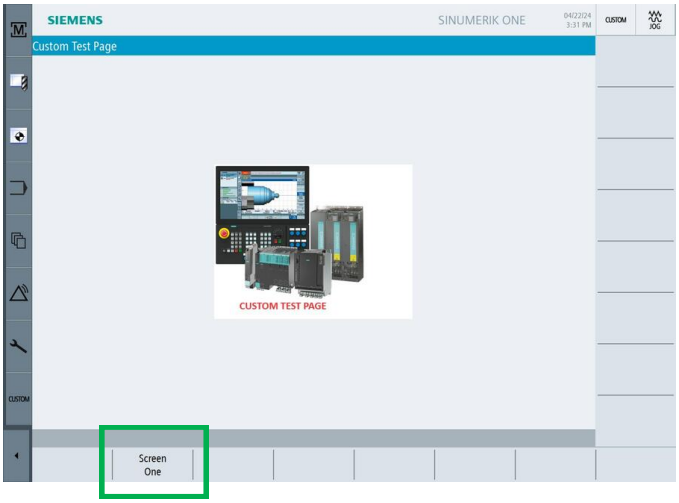
The new "CUSTOM" softkey placed on the sidescreen.

Notes:

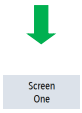


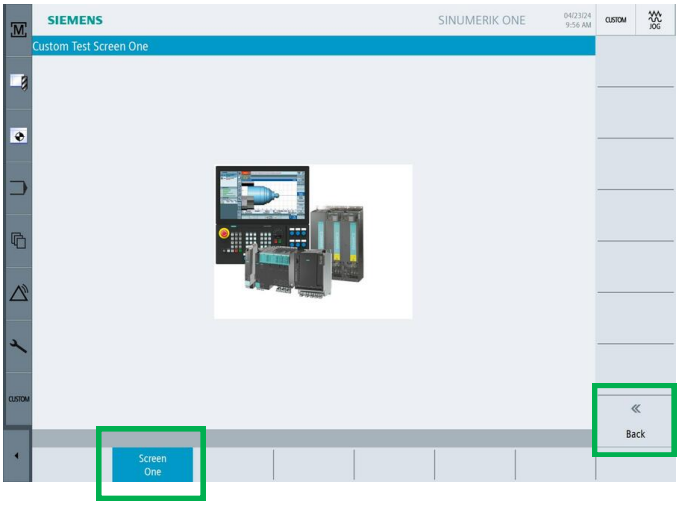
When the “Easy XML” softkey is pressed the XML code in the file xmldial.xml generates the required displayed.

Code contained within the **xmldial.xml** file is used to create soft keys and display texts and graphics etc.



In this example, a horizontal softkey when pressed will show a different screen, the horizontal softkey will change appearance to “blue” to show that is has been pressed, along with a different image





The **xmldial.xml** file will be updated with new XML code as shown below.



The horizontal softkey will change appearance to “blue” to show that is has been pressed.



The vertical softkey when pressed, will go back to the “Custom Test Page”.

Notes:

```
<dialoggui>

<menu name = "main">
<open_form name = "custom_main" />

<let name="curr_softkey_state" type="string">notpressed</let>

<softkey position="2" type="toggle_softkey" refvar="curr_softkey_state">
<caption>Toggle%nSK</caption>
<navigation>main1</navigation>
</softkey>

</menu>

<form name = "custom_main">

<init>
<caption>Custom Test Page</caption>
</init>

<paint>
<img name = "f:/ico1024/custom1.png" xpos="181" ypos="129"  height="155" width="154" aspectratiomode="keepbyexpanding"/>

</paint>
</form>

<menu name = "main1">
<open_form name= "mask1" />

<let name="curr_softkey_state" type="string">pressed</let>

<softkey position="2" type="toggle_softkey" refvar="curr_softkey_state">
<caption>Toggle%nSK</caption>
</softkey>
```

```
<softkey_back>
<navigation>main</navigation>
</softkey_back>

</menu>

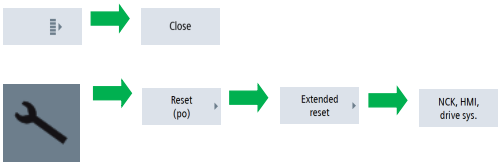
<form name = "mask1">

<init>
<caption>Custom Test Screen One</caption>
</init>

<paint>
<img name = "f:\ico1024\picture1.png" xpos="181" ypos="129"  height="155" width="154" aspectratiomode="keepbyexpanding"/>
</paint>
</form>

</dialoggui>
```

To activate the files, perform:



Notes:

An explanation of the code used:

The file contains the code which determines the content of the “Custom” screen.
Discription:-

```
<menu name = "main">
<open_form name = "custom_main" />

<let name="curr_softkey_state" type="string">notpressed</let>

   <softkey position="2" type="toggle_softkey" refvar="curr_softkey_state">
   <caption>Toggle%nSK</caption>
<navigation>main1</navigation>
</softkey>

</menu>

<form name = "custom_main">

<init>
<caption>Custom Test Page</caption>
</init>

<paint>
<img name = "f:/ico1024/custom1.png" xpos="181" ypos="129"  height="155" width="154" aspectratiomode="keepbyexpanding"/>

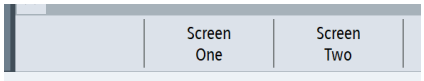
</paint>
</form>
```

```
<menu name="main">

. . . . . .

</menu>
```

The “menu” tag defines a menu containing the softkey description and the dialog to be opened.




```
<let name="curr_softkey_state" type="string">notpressed</let>

   <softkey position="2" type="toggle_softkey" refvar="curr_softkey_state">
   <caption>Screen%nOne</caption>
<navigation>main1</navigation>
</softkey>
```

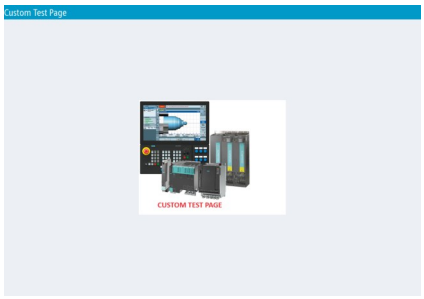
The code “NOTPRESSED” - “PRESSED” allows the toggle of the colour of the softkey.



Notes:

```
<form name = "custom_main">
. . . . . .
</form>
```

The “form” tag defines a menu containing the description of a user dialog, including any text or pictures.



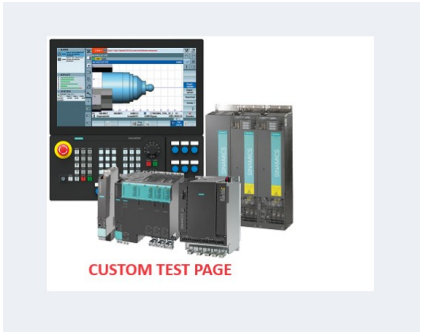

```
<caption>Custom Test Page</caption>
```

Is the text shown on the form/Screen title/name




```
<paint>
<img name = "f:/ico1024/custom1.png" xpos="181" ypos="129" height="155" width="154" aspectratiomode="keepbyexpanding" />
</paint>
```

An image named “custom1.png”, will be displayed when **HS7** is pressed. The image will be positioned **181** pixels from the left and **129** pixels from the top of the screen. Picture size will be 155 x 154 (Pixel aspect).



Notes:

Note

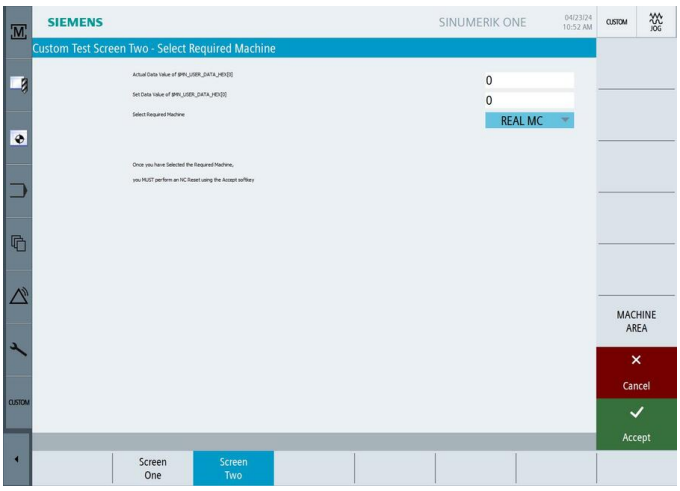
This example applicable to “SINUMRIK ONE” ONLY.

The second example will allow the setting of machine data (MD) and then through the creation of softkeys in the vertical plane, to have the ability perform an NC “Reset (po)”, cancel, and quit the “CUSTOM” screen area.
The machine data that is going to be used is:

14512 $MN_USER_DATA_HEX[0]

And will set either bit 0, or bit 1, or bit 2, or bit 3, of this machine data.

As shown below.



The following code can be written into **xmldial.xml**

```
<dialoggui>

<textfile>oem_xml_screens</textfile>
<let name = "checkoptionbit0" type = "int">0</let>
<let name = "useroptionbit0" type = "int">0</let>

<menu name = "main">
<open_form name = "Custom_main" />

<let name="curr_softkey_state1" type="string">notpressed</let>

   <softkey position="2" type="toggle_softkey" refvar="curr_softkey_state1">
   <caption>Screen%nOne</caption>
<navigation>main1</navigation>
</softkey>

<let name="curr_softkey_state2" type="string">notpressed</let>

   <softkey position="3" type="toggle_softkey" refvar="curr_softkey_state2">
   <caption>Screen%nTwo</caption>
<navigation>main2</navigation>
</softkey>

</menu>

<form name = "custom_main">

<init>
<caption>Custom Test Page</caption>
</init>

<paint>
<img name = "f:/ico1024/custom1.png" xpos="181" ypos="129"  height="155" width="154" aspectratiomode="keepbyexpanding"/>

</paint>
</form>
```

Notes:

```
<menu name = "main1">
<open_form name= "Mask1" />

<let name="curr_softkey_state1" type="string">PRESSED</let>

<softkey POSITION="2" type="toggle_softkey" refvar="curr_softkey_state1">
<caption>Screen%nOne</caption>
</softkey>

<let name="curr_softkey_state2" type="string">NOTPRESSED</let>

<softkey POSITION="3" type="toggle_softkey" refvar="curr_softkey_state2">
<caption>Screen%nTwo</caption>
<navigation>main2</navigation>
</softkey>

<softkey_back>
<navigation>main</navigation>
</softkey_back>

</menu>

<form name = "mask1">

<init>
<caption>Custom Test Screen One</caption>
</init>

<paint>
<img name = "f:\ico1024\picture1.png" xpos="181" ypos="129"  height="155" width="154" aspectratiomode="keepbyexpanding"/>
</paint>
</form>

<menu name = "main2">
<open_form name= "mask2" />
```

```
<let name="curr_softkey_state1" type="string">notpressed</let>

<softkey position="2" type="toggle_softkey" refvar="curr_softkey_state1">
<caption>Screen%nOne</caption>
<navigation>main1</navigation>
</softkey>

<let name="curr_softkey_state2" type="string">pressed</let>

<softkey position="3" type="toggle_softkey" refvar="curr_softkey_state2">
<caption>Screen%nTwo</caption>
</softkey>

<softkey_back>
<navigation>main</navigation>
</softkey_back>

<softkey position = "14">
<caption>MACHINE%nAREA</caption>
<switchtoarea name="areamachine" dialog="simachine"/>
</softkey>

<softkey_cancel>
<OP> UserOptionBit0 = checkoptionbit0 </OP>
<navigation>main</navigation>
</softkey_cancel>

<softkey_accept>
<OP> "$MN_USER_DATA_HEX[0]" = UserOptionBit0 </OP>
<control_reset resetnc="true" />
<switchtoarea name="areamachine" dialog="simachine"/>
</softkey_accept>

</menu>
```

Notes:

```
<menu name = "main2">
<open_form name= "mask2" />

<let name="curr_softkey_state1" type="string">notpressed</let>

<softkey position="2" type="toggle_softkey" refvar="curr_softkey_state1">
<caption>Screen%nOne</caption>
<navigation>main1</navigation>
</softkey>

<let name="curr_softkey_state2" type="string">pressed</let>

<softkey POSITION="3" type="toggle_softkey" refvar="curr_softkey_state2">
<caption>Screen%nTwo</caption>
</softkey>

<softkey_back>
<navigation>main</navigation>
</softkey_back>

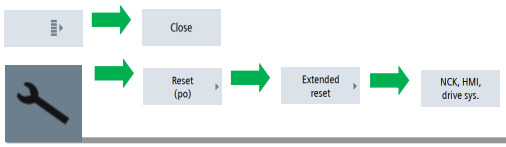
<softkey position = "14">
<caption>MACHINE%nAREA</caption>
<switchtoarea name="areamachine" dialog="simachine"/>
</softkey>

<softkey_cancel>
<op> useroptionbit0 = checkoptionbit0 </op>
<navigation>main</navigation>
</softkey_cancel>

<softkey_accept>
<op> "$MN_USER_DATA_HEX[0]" = useroptionbit0 </op>
<control_reset resetnc="true" />
<switchtoarea name="areamachine" dialog="simachine"/>
</softkey_accept>
```

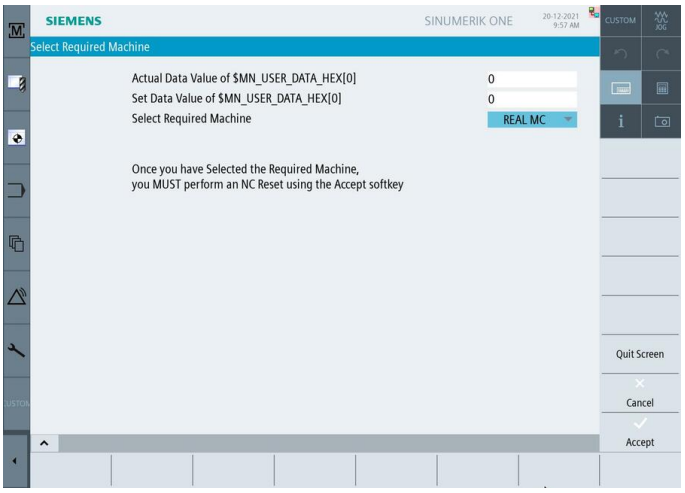
```
</menu>
<form name = "mask2">
<init>
<caption>Custom Test Screen Two - Select Required Machine</caption>
<op> checkoptionbit0 = "$MN_USER_DATA_HEX[0]"</OP>
<op> useroptionbit0 = checkoptionbit0</OP>
<data_access type="true" />
<control name = "check0" xpos = "450" ypos = "34" refvar="checkoptionbit0" hotlink="true" width="90" fieldtype="readonly"  font = "1"
time="normal" color_bk="#ffffff"/>
<control name = "Check1" xpos = "450" ypos = "54" refvar="useroptionbit0" hotlink="true" width="90" fieldtype="readonly"  font = "1"
time="normal" color_bk="#ffffff"/>
<control name = "Option" xpos = "450" ypos = "74" refvar="useroptionBit0" hotlink="true" width="90" fieldtype="combobox" font= "1"
time="normal" color_bk="#ffffff">
<item value="0">REAL MC</item>
<item value="1">CMVM IQ</item>
<item value="2">CMVM SIM</item>
<item value="4">RMVM</item>
</control>
</init>
<paint>
<text xpos = "100" ypos = "34">"Actual Data Value of $MN_USER_DATA_HEX[0]</text>
<text xpos = "100" ypos = "54">"Set Data Value of $MN_USER_DATA_HEX[0]</text>
<text xpos = "100" ypos = "74">"Select Required Machine"</text>
<text xpos = "100" ypos = "124">"Once you have Selected the Required Machine,"</text>
<text xpos = "100" ypos = "139">"you MUST perform an NC Reset using the Accept softkey"</text>
</paint>
</form>
</dialoggui>
```

To activate the files, perform:

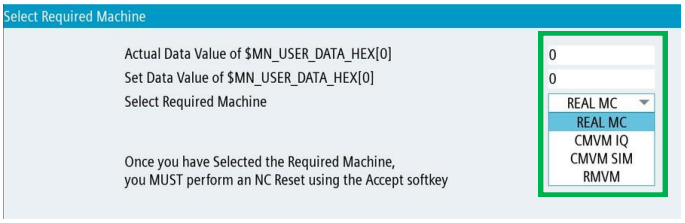


Notes:

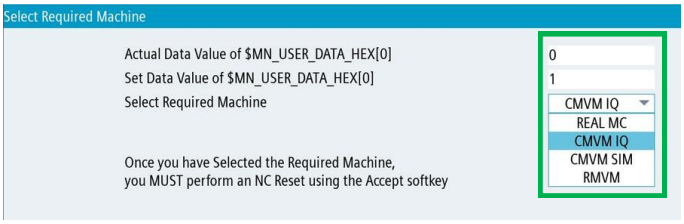


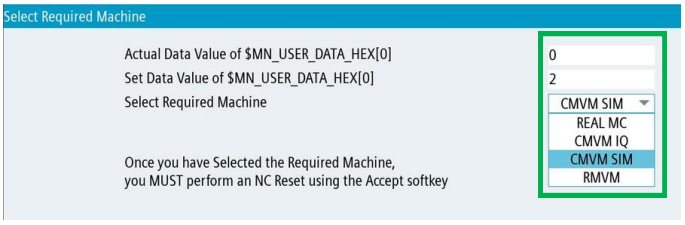


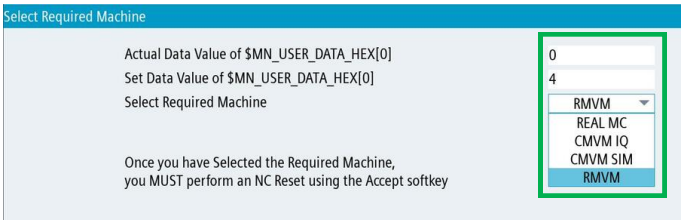
Above shows the new screen.



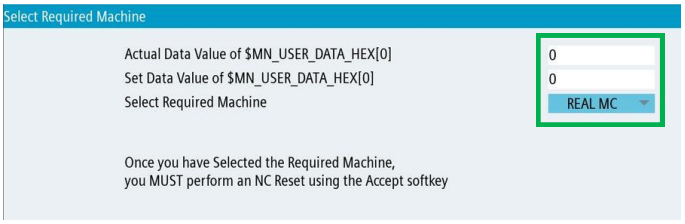
The pull down menu allows the selection of “text”, this will enter a numerical value into the window above the pull down menu.







By pressing the “Accept” softkey an NC “Reset (po)” is performed.
The screen below shows the outcome after an NC “Reset (po)”.
The Actual value and Set value are equal.



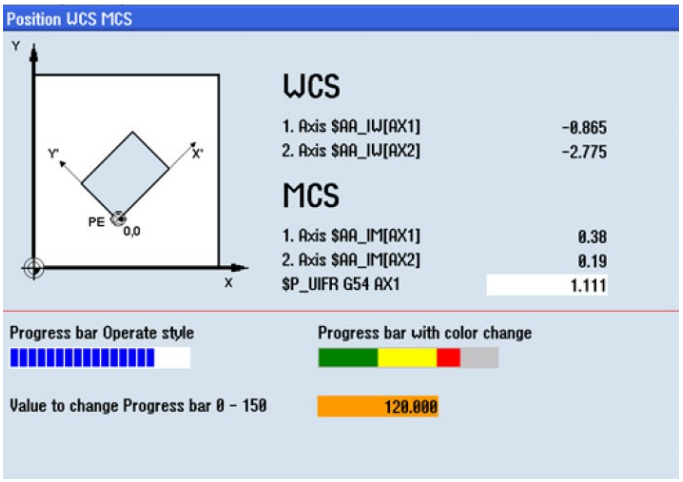
The “Quit Screen” softkey takes you back to the Machine page area.

“Cancel” softkey will put back the screen with data that was in the screen when it was first opened.

Notes:

Basic Application Information for user screens

Application files are used to create user screens.
These application files are automatically interpreted by the controller.
The files are not available as standard and have to be created by the user using an ASCII editor (e.g. Notepad®, Notepad++® or directly on the controller).



Operating areas overview

The "*_oem.xml" file is a master file available for each area which is required to specify code and other files that are to be used to create a user screen.
These files contain the information which determine what is displayed on a particular screen.
The screen is split into different areas:

To enable user screens in these areas of the controller, the correct "*_oem.xml" file has to be modified.
There are eight areas in total:-

Machine Area-Sub divided into:-
Jog
MDA
Auto

Parameter Area

Program Edit Area

Program Manager Area

Diagnosis Area

Start-up Area

Each area can have a "xml" file specified.

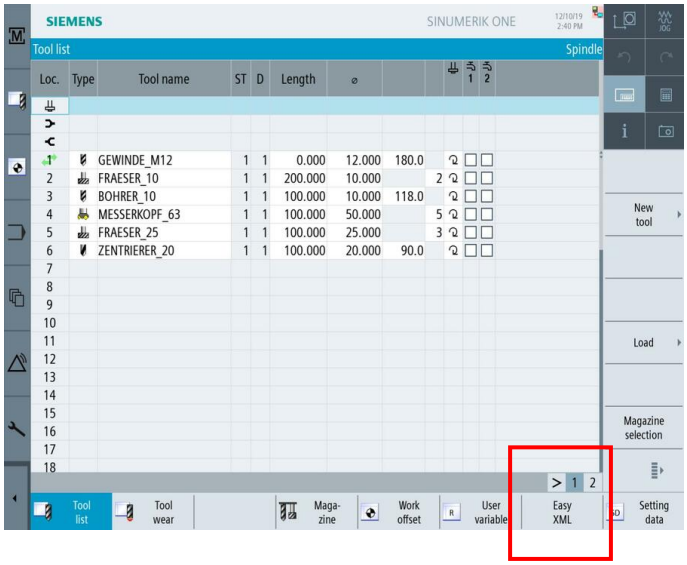
Notes:

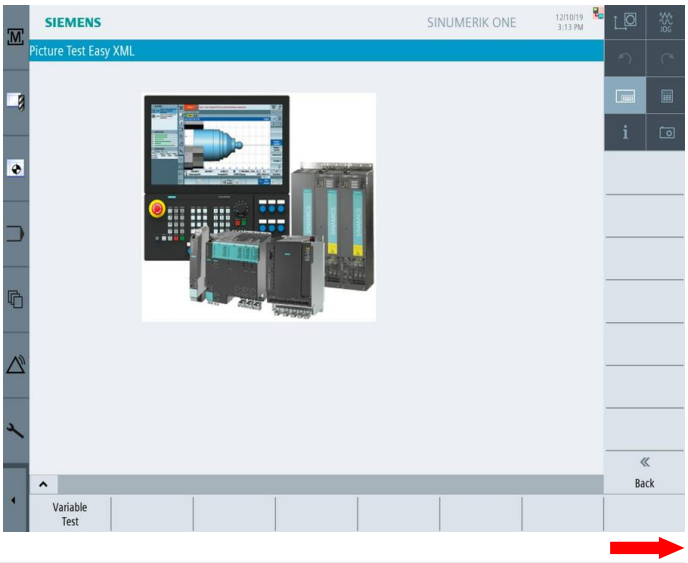
Operating area	Permissible position for start soft-keys	File name	Area name
Machine JOG	HSK6	slmachine_oem.xml	AreaMachine
Machine MDA	HSK6	slmachine_oem.xml	AreaMachine
Machine AUTO	HSK6	slmachine_oem.xml	AreaMachine
Parameter	HSK7	slparameter_oem.xml	AreaParameter
Program	HSK6	slprogramedit_oem.xml	AreaProgramEdit
Program-Manager	HSK2-8, 12-16, if not used for networked drives	slpmdialog_oem.xml	AreaProgramManager
Diagnostics	HSK7	sldiagnose_oem.xml	AreaDiagnosis
Setup	HSK7	slsudiaglog_oem.xml	AreaStartup

Creating Easy XML User Files

The files required for the Easy XML function need to be created, this in turn well create new softkeys and screens where required in different areas as stated above.
This example will create a NEW softkey in the PARAMETER AREA and when pressed a new screen will appear.

This picture below shows the objective of this section.

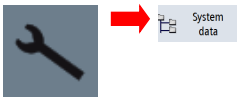


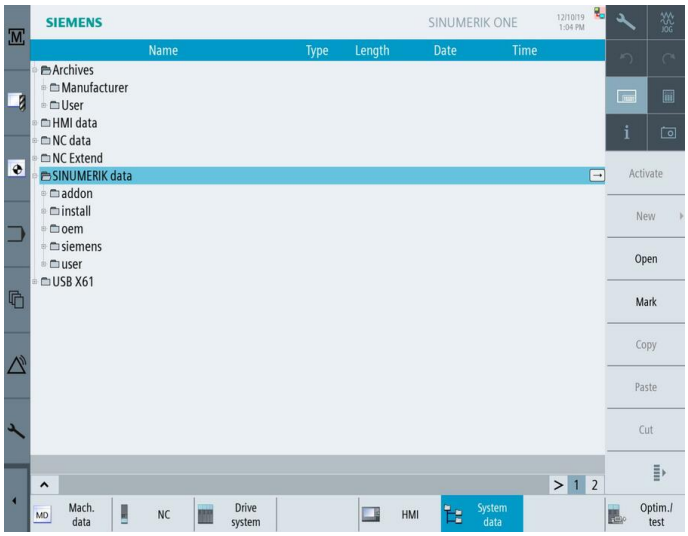


Notes:

These files can be created directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.

These examples are applicable for hardware NCU and IPC.





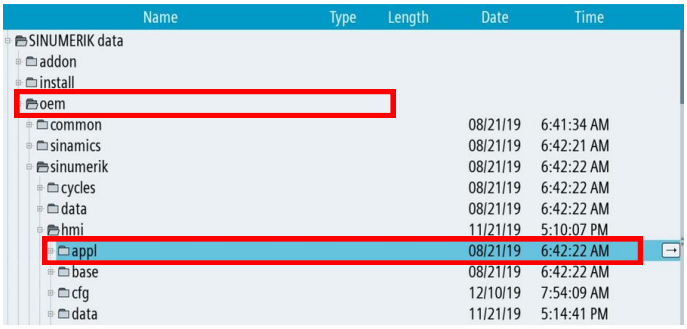
Creating the “*_oem.xml” file on the controller

These files set up the links to the code which are to be used to create the user screens, they also determine which operating areas these files are available in. The file has to be located in one of the following paths/directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/appl

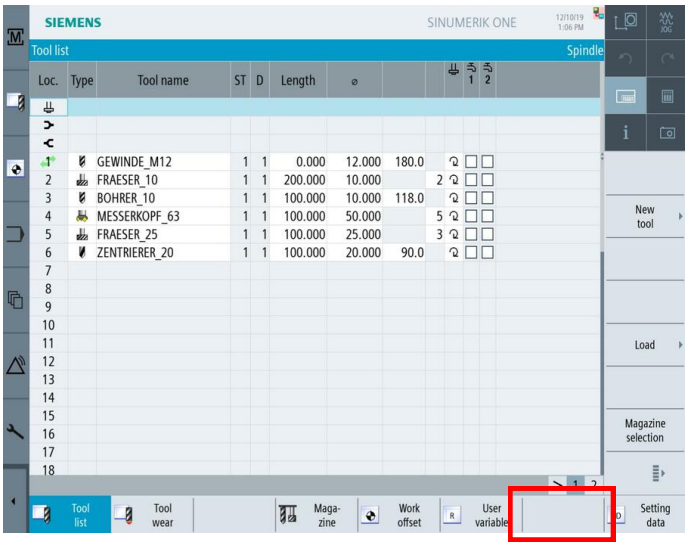
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/appl

Use the cursor keys to locate and open this directory.



This example will create a “Break out” softkey in the “Parameter Area” that then can be used to create further screens. The unused softkey HS 7 position will be used.

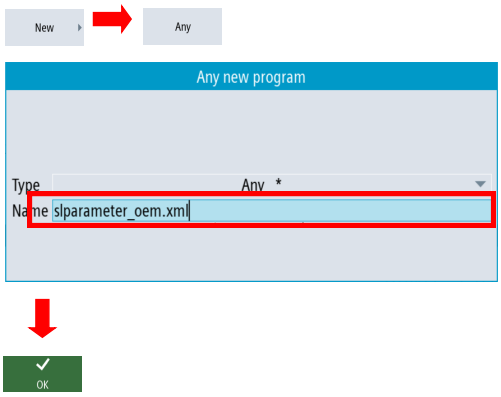




softkey HS7 position

Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.
The “Any” soft-key allows the file type to be determined by the user. In the dialogue box, type in the name of the file and confirm with the “OK” soft-key, for this example use file name slparameter_oem.xml.
The file name must be in lowercase due to the LINUX operating system.

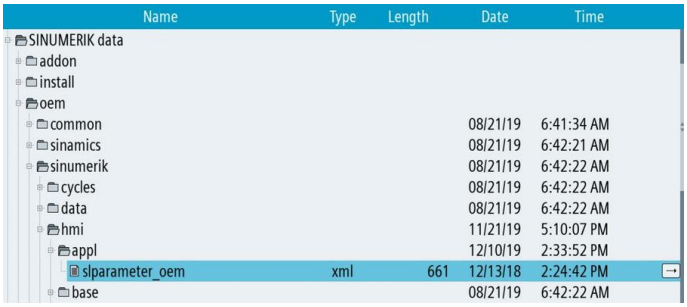


The newly created file will open automatically in preparation for editing.

Creating the User Files on an external PC

As mentioned previously it is possible to create the EasyXML files directly on the controller or offline using an ASCII editor (e.g. Notepad® or Notepad++®).
These files have to be transferred to the controller and placed in the correct folders using the USB Interface or with “Access MyMachine“ software.

Below: The **slparameter_oem.xml** file located in the **appl** directory



Example: slparameter_oem.xml file.
It is necessary to specify a soft-key to be used as the entry point to the parameter area.
Further soft-keys can be assigned within another file, in this case “para_oem.xml” will be used for this.

The following example will show content of the “**slparameter_oem.xml** ” file for the “Parameter Area” of the controller.

```
<!DOCTYPE DIALOG>
<DIALOG>
<TEXTFILE>oem_xml_screens</TEXTFILE>

<MENU name="SlPaMenuHU">
<ETCLEVEL id="0">
<SOFTKEYGROUP name="SlPaSoftkeyGroup" toggleselectedsoftkey="true" >
<SOFTKEY position="7">
<PROPERTY name="textID" type="QString">Easy%nXML</PROPERTY>
<PROPERTY name="translationContext" type="QString">easy_xml</PROPERTY>
<FUNCTION args="-area CustomXML -dialog SlEECustomDialog -mainModule para_oem.xml -entry main"
name="switchToDialog"/>
</SOFTKEY>
</SOFTKEYGROUP>
</ETCLEVEL>
</MENU>
</DIALOG>
```

The example defines the 7th horizontal soft-key as the entry point, the soft-key will have the text “Easy XML” (on 2 lines).

```
<SOFTKEY position="7">
<PROPERTY name="textID" type="QString">Easy%nXML</PROPERTY>
```

When the soft-key is pressed the file **para_oem.xml** will be loaded:

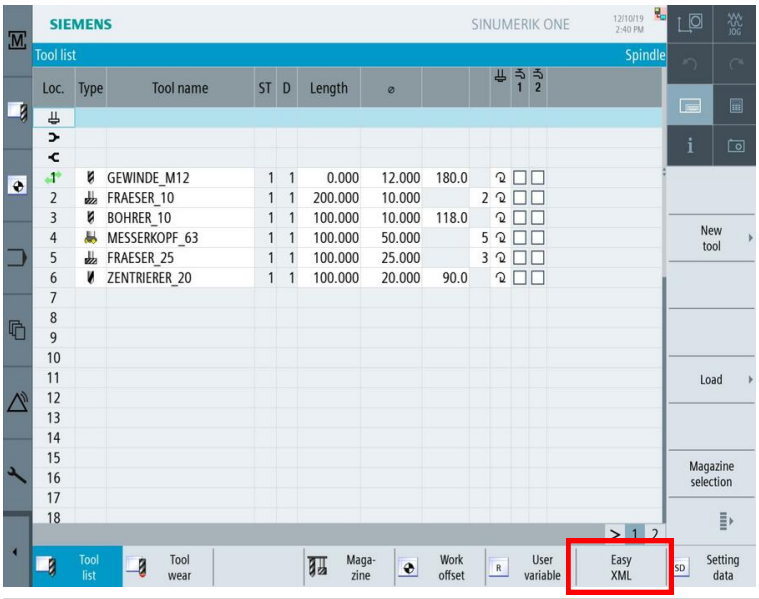
```
<FUNCTION args="-area CustomXML -dialog SlEECustomDialog -mainModule para_oem.xml -entry main"
name="switchToDialog"/>
```

Notes:

To activate the files, perform:







softkey HS7 position with text

Creating the “para_oem.xml” file

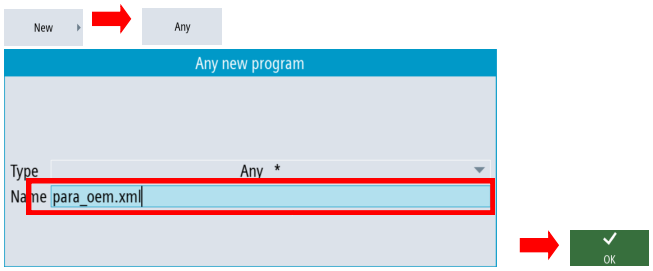
The **para_oem.xml** file has been declared in the “slparameter_oem.xml” file and now needs to be created.

Code contained within the **para_oem.xml** file is used to create soft keys and display texts and graphics etc.

The file needs to be created and placed in one of the following folders:

SINUMERIK Data - CF Card/oem/sinumerik/hmi/appl

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/appl



Notes:

Example: para_oem.xml file

The following example shows content of the "**para_oem.xml** " file for the "Parameter Area" in the controller.

```
<DialogGui>
<menu name = "MAIN">
<open_form name = "MENU_DISPLAY" />
</menu>
<form name = "MENU_DISPLAY">
<init>
<caption>Picture Test</caption>
</init>
<paint>
<img xpos = "23" ypos = "48" height="320" width = "343" name = "f:/ico1024/picture1.png" />
</paint>
</form>
```

The file contains the code which determines the content of the "Run EasyXML" screen.
Discription:-

`<menu name = "MAIN">`
`</menu>`
Is the internal name of the screen.

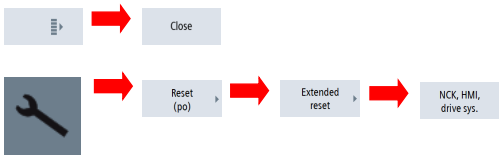
`<open_form name = "MENU_DISPLAY" />`
Opens the form named "MENU_DISPLAY".

`<form name = "MENU_DISPLAY">`
This is the start of the form to be displayed

`<caption>Picture Test</caption>`
Is the text shown on the form/Screen title/name

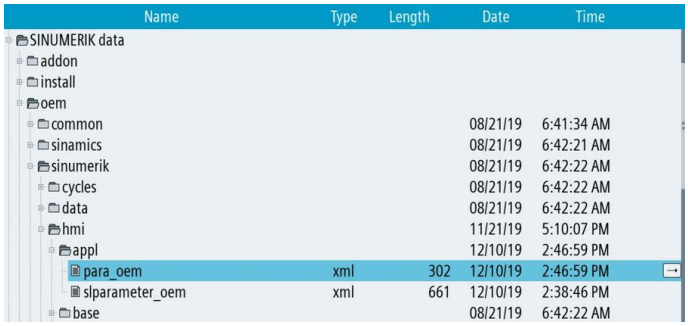
`<img xpos = "23" ypos = "23" height="320" width = "343" name = "f:/ico1024/picture1.png" />`
An image named "picture1.png", will be displayed when **HS7** is pressed. The image will be positioned **23** pixels from the left and **23** pixels from the top of the screen. Picture size will be 320 x 343 (Pixel aspect).

To activate the files, perform:



Notes:

Below: The para_oem.xml file located in the appl directory



To see the results of the example, locate the “parameter” area of the control.



The 7th horizontal soft-key contains the text specified in the “slparameter_oem.xml” file: `HS7 softkey="Easy XML".`

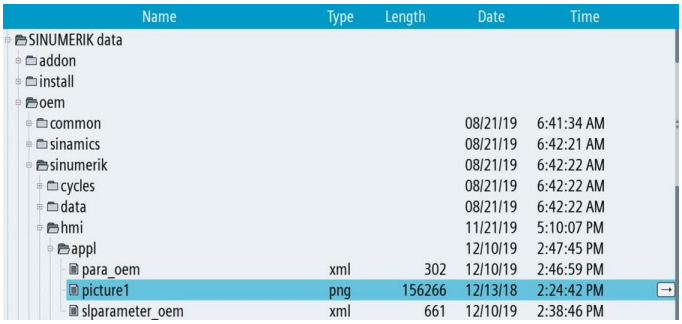
Storing of pictures

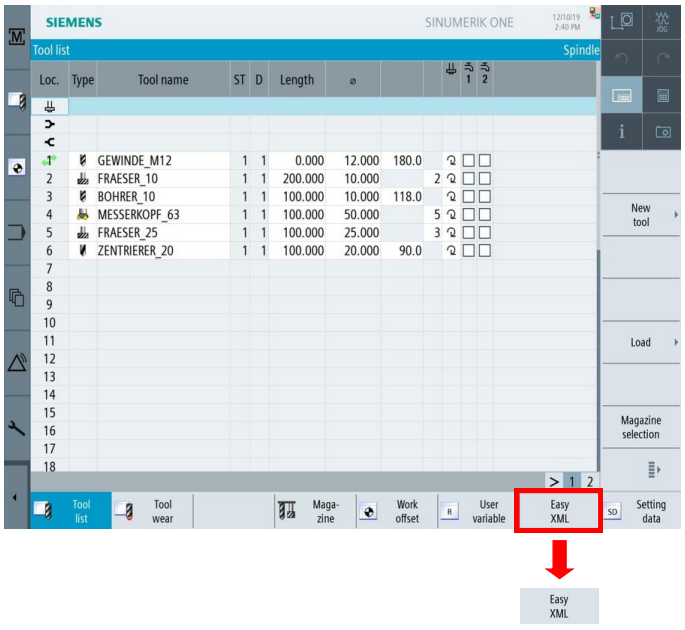
You can create your own icon files under one of the following paths:

SINUMERIK Data - CF Card/oem/sinumerik/hmi/iappl

IPC: C:/ProgramData/Siemens/MotionControl//oem/ sinumerik/hmi/appl

In the example, one icons are used:
picture1.png





Screen title/caption picture1.png image



Notes:

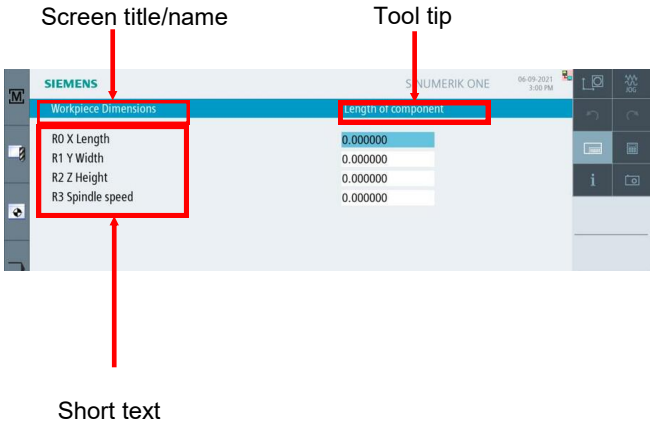
Adding additional Dialogue Elements:

Dialogue Elements refer to short texts, graphics texts, unit texts and I/O fields. Additional soft-keys can also be added.

The dialogue elements are added in the “para_oem.xml file.

In the following example a soft-key with the label “Variable Test” is added. When the soft-key is pressed the screen will display a list of four variables. The four variables are defined under a new form (Workpiece Dimensions).

A detailed description of the Dialogue Element parameters can be found in chapter ”Parameters and Variable Descriptions”.



Example: para_oem.xml file

The following example shows content of the “**para_oem.xml** ” file for the “Parameter Area” in the controller. (This example the text can over write the text of the previous example in the “**para_oem.xml** ” file)

```
<DialogGui>
<menu name = "MAIN">
<OPEN_FORM name = "MAIN_DISPLAY" />
<softkey POSITION="1">
<caption>Variable Test</caption>
<navigation>MENU_R_PARAMETER</navigation>
</softkey>
<softkey_back POSITION="16">
<switchToArea name="AreaParameter" dialog="SlParameter" args="-screen SlTmToollistScreen"/>
</softkey_back>
</menu>

<form name = "MAIN_DISPLAY">
<init>
<caption>Picture Test Easy XML</caption>
</init>

<paint>
<img xpos = "110" ypos = "48" height="208" width = "232" name = "f:/ico1024/picture1.png" />
</paint>
</form>
<!-- ************************************************************Menu R- Parameter-->
<menu name ="MENU_R_PARAMETER">
     <OPEN_FORM name = "R-Parameter" />
          <softkey_back POSITION="16">
          <navigation>MAIN</navigation>
```

Notes:

```
          </softkey_back>
</menu>
<!-- cursor text -->
<let name="R-Parameter_cursor_texts" dim="4" type="string">
{"Length of component"},
{"Width of component"},
{"Height of component"},
{"Main spindle speed"},
</let>
<let name="R_Parameter_cursor">-1</let>

<form name = "R-Parameter">

<init>
<DATA_ACCESS type="true" />

<caption define_sections="2">
<property value="0" length="20" position="2" alignment="left" />
<property value="1" length="30" position="69" alignment="right" />
</caption>

<caption index ="0">Workpiece Dimensions 2</caption>

<op>
R_Parameter_cursor = -1;
</op>

<control name = "c1" xpos = "322" ypos = "34" refvar="nck/Channel/Parameter/R[0]" >
<property item_data="1000" />
<property tooltip="Length of component" />
</control>
```

```
<control name = "c2" xpos = "322" ypos = "54" refvar="nck/Channel/Parameter/R[1]" >
<property item_data="1001" />
<property tooltip="Width of component" />
</control>
<control name = "c3" xpos = "322" ypos = "74" refvar="nck/Channel/Parameter/R[2]" >
<property item_data="1002" />
<property tooltip="Height of component" />
</control>
<control name = "c4" xpos = "322" ypos = "94" refvar="nck/Channel/Parameter/R[3]" >
<property item_data="1003" />
<property tooltip="Main spindle speed" />
</control>
</init>

<paint>
<text xpos = "23" ypos = "34">R0 X Length</text>
<text xpos = "23" ypos = "54">R1 Y Width</text>
<text xpos = "23" ypos = "74">R2 Z Height</text>
<text xpos = "23" ypos = "94">R3 Spindle speed</text>
</paint>

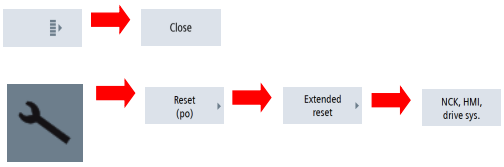
<focus_in>
<op>R_Parameter_cursor = $focus_item_data - 1000</op>
<caption index ="1">$$$R-Parameter_cursor_texts[$R_Parameter_cursor]</caption>
</focus_in>

</form>
```

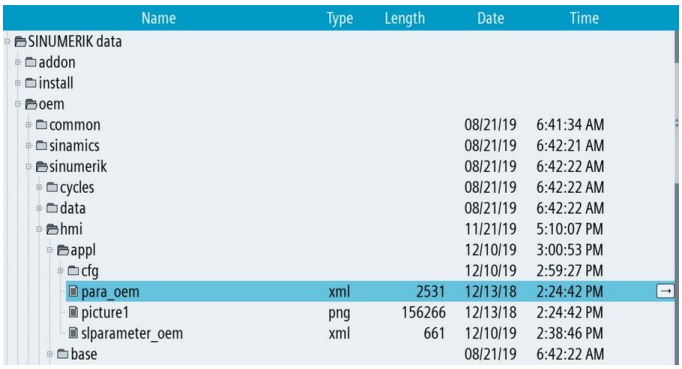


Notes:

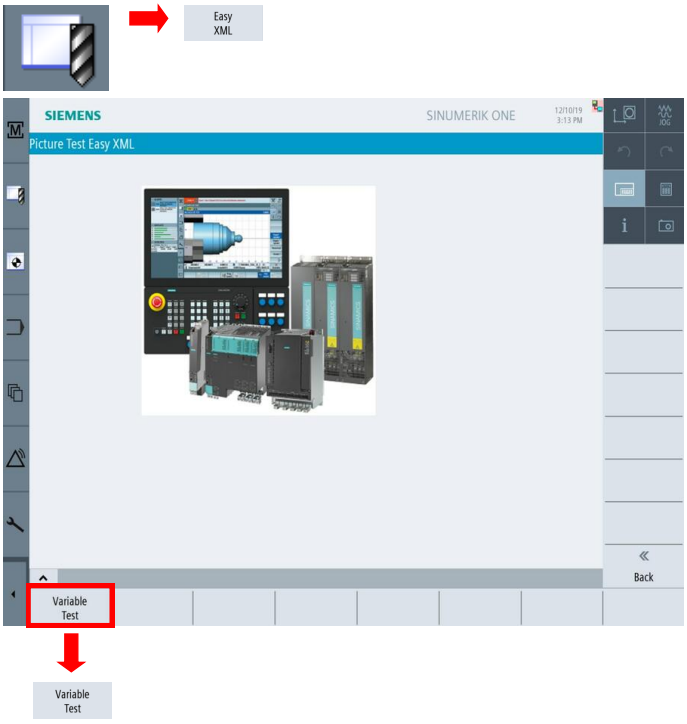
To activate the file, perform:



Below: The para_oem.xml file located in the appl directory



The additional soft-key can now be seen.



When the soft-key is pressed the defined variable will be displayed.



Values entered in the input field will be transferred to the relevant variable.

Notes:

Detailed Analysis:

`<caption>Workpiece Dimensions</caption>`
Name of form:



`<caption>Variable%nTest</caption>`
Determines the text for horizontal soft-key 1 as “Variable Test”:



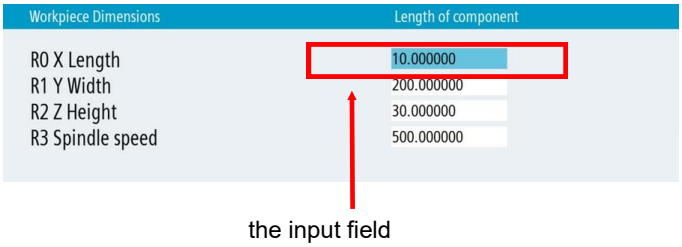
When the soft-key is pressed mask2 will be loaded.

`<navigation>MENU_R_PARAMETER</navigation>`
Goes to
`<menu name ="MENU_R_PARAMETER">`

Which in turn loads code between

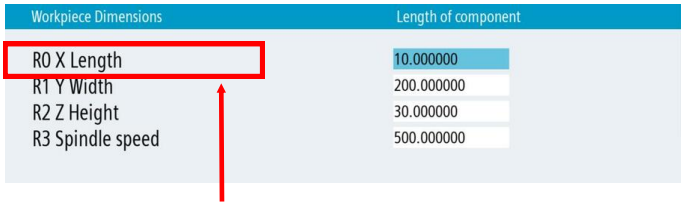
`<form name = "R_PARAMETER">`
And
`</form>`

`<control name = "c1" xpos = "322" ypos = "34" refvar="nck/Channel/Parameter/R[0]" >`
Creates the input field, with an X and Y position.



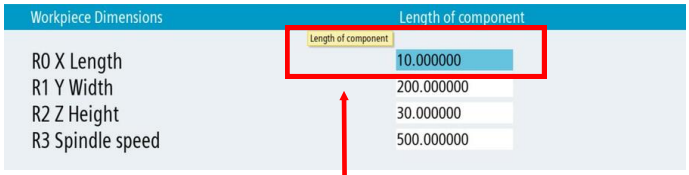
the input field

`<text xpos = "23" ypos = "34">R0 X Length</text>`
Creates the text, with an X and Y position.
R0 X Length



the input field text

`<property tooltip="Length of component" />`
Creates the “tooltip” text.



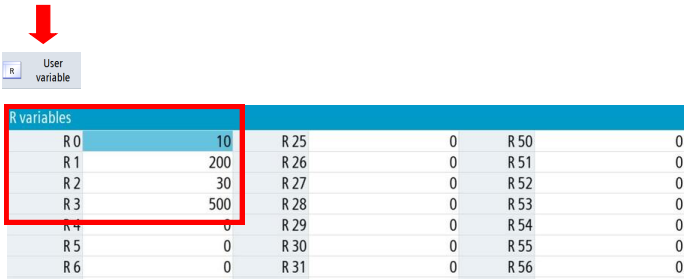
Tooltip text

`VS8=("back")` takes you back to the previous page



Checking the entered values

Values can be entered directly on the OEM screen. These values are transferred directly to the variables.
They can be checked by viewing the variables on the standard screen by pressing the “User variable” soft-key.



Notes:

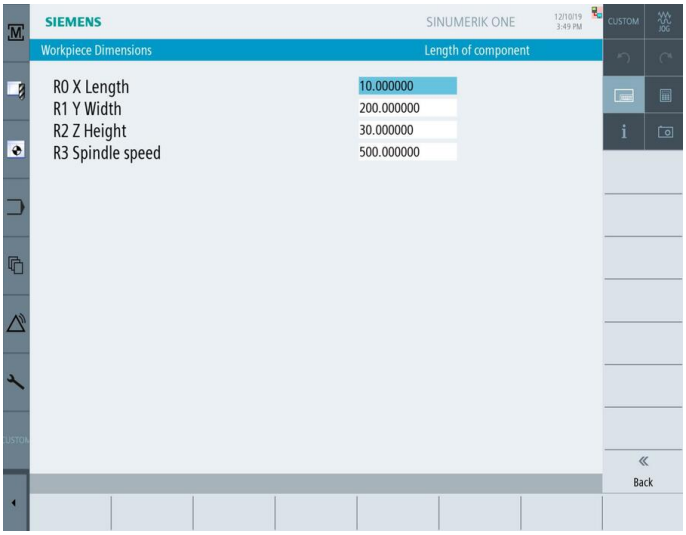
Creating a language dependant file

The files required for the Easy XML function need to be created, this in turn well create language depend screens, depending on the system language selected.

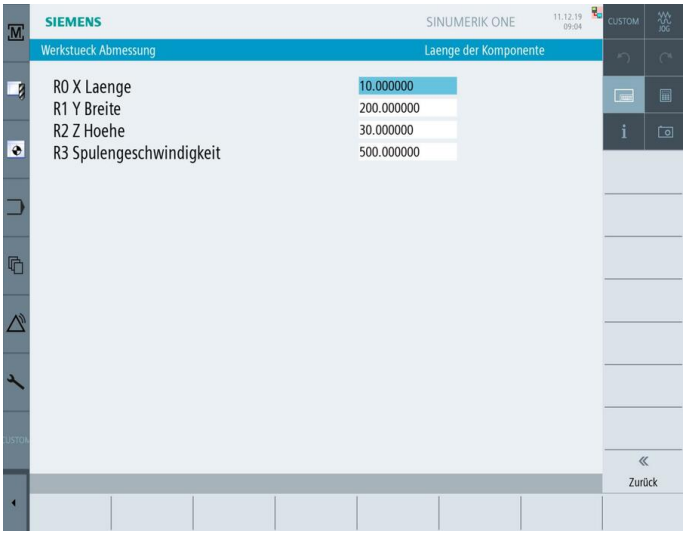
This example will create language dependant screens, for the newly created softkey, in the PARAMETER AREA and when pressed.

This picture below shows the objective of this section.

English:



German:



Language dependant *.ts files are used for :

Softkey labels text
Header text
Help text
And any other text

These file are usually language dependant, so that when a language is selected for the system the appropriate text file is selected and used for the xml files.

The file name always ends with the language code of the particular language and always has the ".ts" extension. For example "oem_xml_screens_deu.ts" for German or "oem_xml_screens_eng.ts" for English.

In this example the **"para_oem.xml"** will be modified, so that any text in this file will be replaced with a generic text, this generic text will then be translated in the appropriate language files English or German.

In this example all language dependant text will have a generic text eg. "$$PARAXML_n"

The $$ requests the system to look for the "oem_xml_screens_eng.ts" for English or "oem_xml_screens_deu.ts" for German language, depending on which system language has been selected.

Notes:

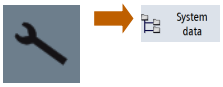
```
<DialogGui>
<menu name = "MAIN">
<OPEN_FORM name = "MAIN_DISPLAY" />
<softkey POSITION="1">
<caption>$$85101</caption>
<navigation>MENU_R_PARAMETER</navigation>
</softkey>
<softkey_back POSITION="16">
<switchToArea name="AreaParameter" dialog="SlParameter" args="-screen SlTmToollistScreen"/>
</softkey_back>
</menu>
<form name = "MAIN_DISPLAY">
<init>
<caption>$$85102</caption>
</init>
<paint>
<img xpos = "110" ypos = "48" height="208" width = "232" name = "f:/appl/picture1.png" />
</paint>
</form>
<!-- menu r - parameter -->
<menu name = "MENU_R_PARAMETER">
<OPEN_FORM name= "R-Parameter" />
<softkey_back POSITION="16">
<navigation>MAIN</navigation>
</softkey_back>
```

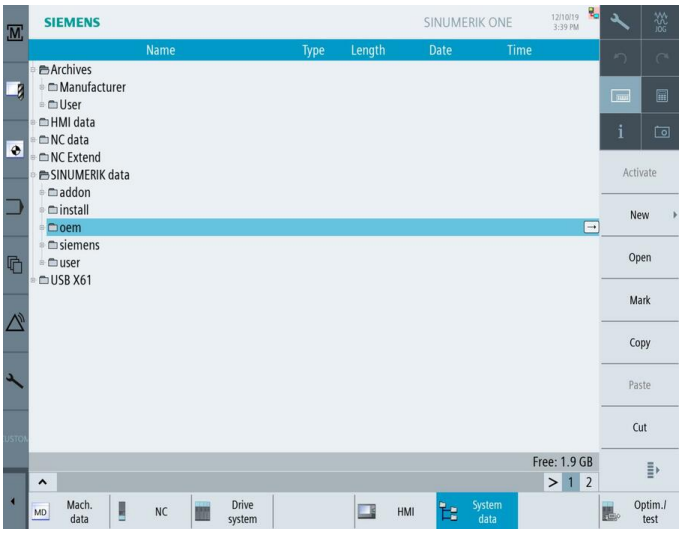
```
</menu>
<!-- cursor text -->
<let name="R-Parameter_cursor_texts" dim="4" type="string">
{"$$85103"},
{"$$85104"},
{"$$85105"},
{"$$85106"},
</let>
<let name="R_Parameter_cursor">-1</let>
<form name = "R-Parameter">
<init>
<DATA_ACCESS type="true" />
<caption define_sections="2">
<property value="0" length="40" position="4" alignment="left" />
<property value="1" length="30" position="58" alignment="right" />
</caption>
<caption index ="0">$$85107</caption>
<op>
R_Parameter_cursor = -1;
</op>
<control name = "c1" xpos = "322" ypos = "34" refvar="nck/Channel/Parameter/R[0]" >
<property item_data="1000" />
<property tooltip="$$85103" />
</control>
<control name = "c2" xpos = "322" ypos = "54" refvar="nck/Channel/Parameter/R[1]" >
<property item_data="1001" />
```

Notes:

```
<property tooltip="$$85104" />
</control>
<control name = "c3" xpos = "322" ypos = "74" refvar="nck/Channel/Parameter/R[2]" >
<property item_data="1002" />
<property tooltip="$$85105" />
</control>
<control name = "c4" xpos = "322" ypos = "94" refvar="nck/Channel/Parameter/R[3]" >
<property item_data="1003" />
<property tooltip="$$85106" />
</control>
</init>
<paint>
<text xpos = "23" ypos = "34">$$85108</text>
<text xpos = "23" ypos = "54">$$85109</text>
<text xpos = "23" ypos = "74">$$85110</text>
<text xpos = "23" ypos = "94">$$85111</text>
</paint>
<focus_in>
<op>R_Parameter_cursor = $focus_item_data - 1000</op>
<caption index ="1">$$$R-Parameter_cursor_texts[$R_Parameter_cursor]</caption>
</focus_in>
</form>
```

The files required for the EasyXML language in the screen function need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





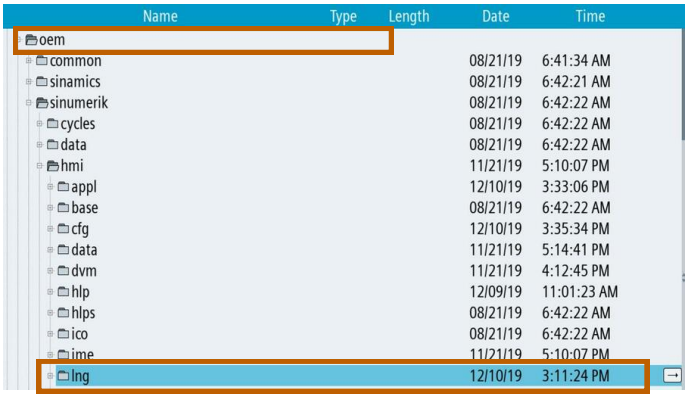
Creating the “oem_xml_screens_eng.ts” file on the controller

This file sets up the link for the Easy XML in the screen. The file has to be located in one of the following paths/directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/lng

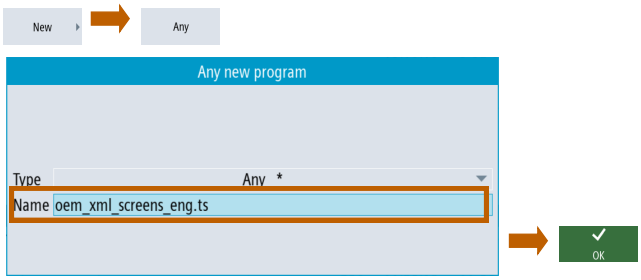
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/lng

Use the cursor keys to locate and open this directory.



Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.



The newly created file will be automatically opened in preparation for editing.

Example: oem_xml_screens_eng.ts file
The following example will show how the “**oem_xml_screens_eng.ts**” file is used to specify the “language” text in the PARAMETER brake out area of the controller.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS>
<TS>
<context>
<name>easy_xml</name>
<!-- My XML parameter screen  -->
<message>
<source>85100</source>
<translation>Easy%nXML</translation>
<chars>16</chars>
</message>
```

```
<message>
<source>85101</source>
<translation>Variable%nTest</translation>
<chars>16</chars>
</message>
<message>
<source>85102</source>
<translation>Picture Test Easy XML</translation>
<chars>16</chars>
</message>
<message>
<source>85103</source>
<translation>Length of component</translation>
<chars>16</chars>
</message>
<message>
<source>85104</source>
<translation>Width of component</translation>
<chars>16</chars>
</message>
<message>
<source>85105</source>
<translation>Height of component</translation>
<chars>16</chars>
</message>
<message>
<source>85106</source>
<translation>Main spindle speed</translation>
<chars>16</chars>
</message>
```

Notes:

```
<message>
<source>85107</source>
<translation>Workpiece Dimensions</translation>
<chars>16</chars>
</message>
<message>
<source>85108</source>
<translation>R0 X Length</translation>
<chars>16</chars>
</message>
<message>
<source>85109</source>
<translation>R1 Y Width</translation>
<chars>16</chars>
</message>
<message>
<source>85110</source>
<translation>R2 Z Height</translation>
<chars>16</chars>
</message>
<message>
<source>85111</source>
<translation>R3 Spindle speed</translation>
<chars>16</chars>
</message>

</context>
</TS>
```



The following example will show how the "**oem_xml_screens_deu.ts**" file is used to specify the "language" text in the PARAMETER brake-out area of the controller.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS>
<TS>
<context>
<name>easy_xml</name>

<!-- My XML parameter screen  -->

<message>
<source>85100</source>
<translation>Easy%nXML</translation>
<chars>16</chars>
</message>
<message>
<source>85101</source>
<translation>Variabler%nTest</translation>
<chars>16</chars>
</message>
<message>
<source>85102</source>
<translation>Testbild Easy XML</translation>
<chars>16</chars>
</message>
<message>
<source>85103</source>
<translation>Laenge der Komponente</translation>
<chars>16</chars>
</message>
```

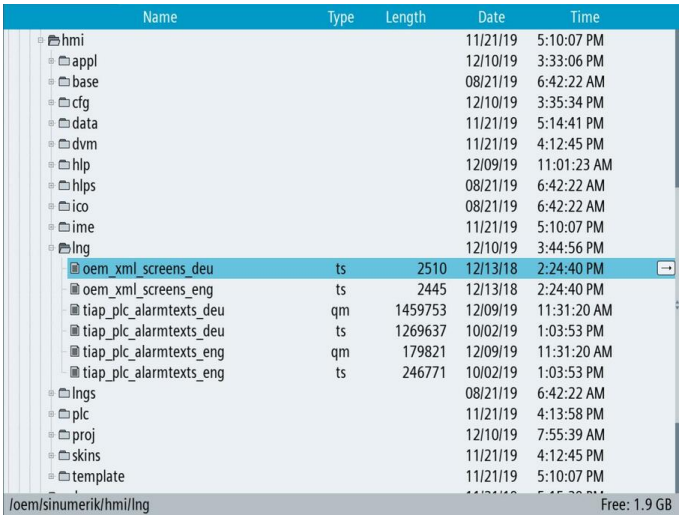
Notes:

```
<message>
<source>85104</source>
<translation>Breite der Komponente</translation>
<chars>16</chars>
</message>
<message>
<source>85105</source>
<translation>Hoehe der Komponente</translation>
<chars>16</chars>
</message>
<message>
<source>85106</source>
<translation>Hauptspindel Geschwindigkeit</translation>
<chars>16</chars>
</message>
<message>
<source>85107</source>
<translation>Werkstueck Abmessungen</translation>
<chars>16</chars>
</message>
<message>
<source>85108</source>
<translation>R0 X Laenge</translation>
<chars>16</chars>
</message>
<message>
<source>85109</source>
<translation>R1 Y Breite</translation>
<chars>16</chars>
</message>
```

```
<message>
<source>85110</source>
<translation>R2 Z Hoehe</translation>
<chars>16</chars>
</message>
<message>
<source>85111</source>
<translation>R3 Spulengeschwindigkeit</translation>
<chars>16</chars>
</message>

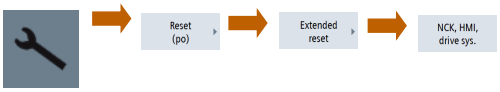
</context>
</TS>
```

The **oem_xml_screens_deu.ts and oem_xml_screens_eng.ts** files located in the **ing** directory



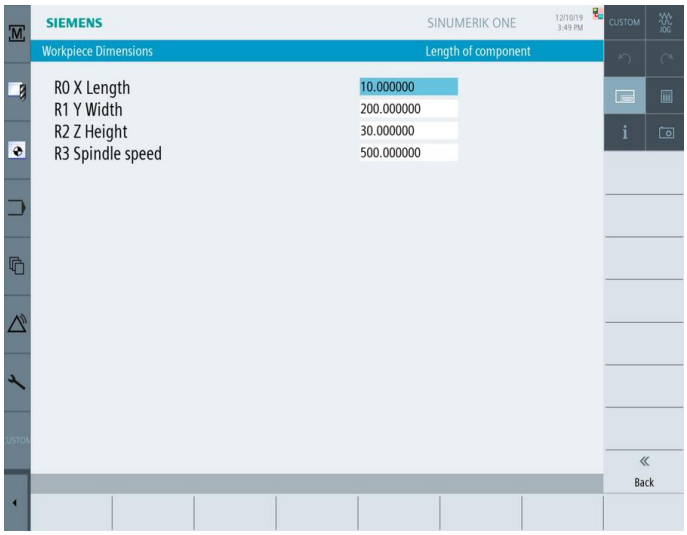
Notes:

To make active perform.

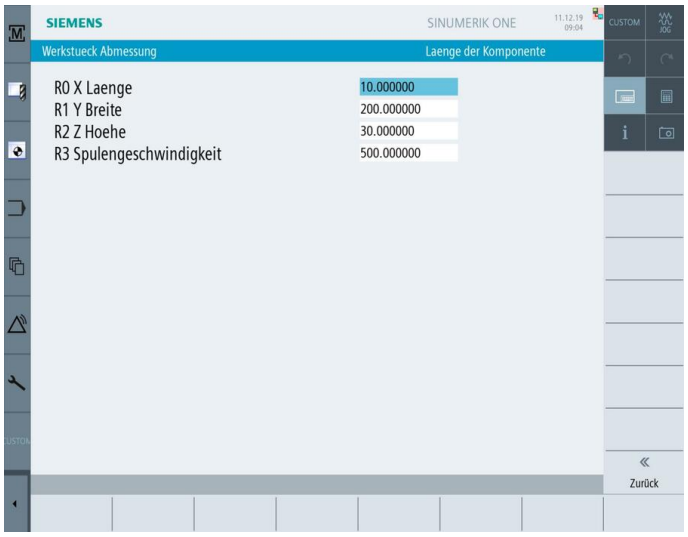


The result is shown below.

English:



German:



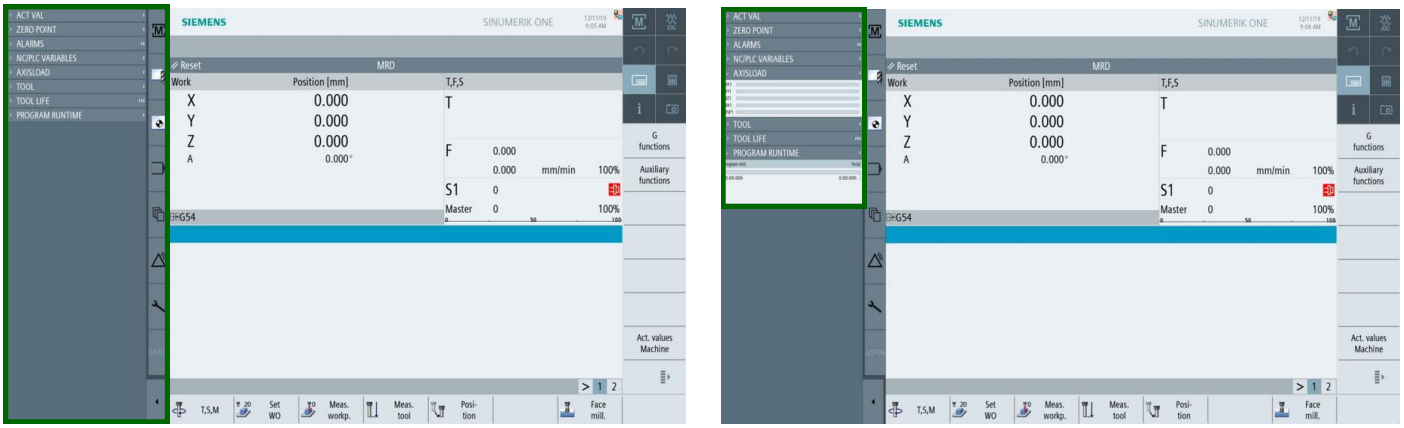
Notes:

General Information SideScreen

These examples are applicable for hardware PPU 290, NCU, and IPC

The Easy XML scripting language can also be used to create dialogs in the Sidescreen area. The “Sidescreen area” Page and “Sidescreen area Widgets” are available for displaying OEM dialogs. The connection is made via the slsidescreen.ini file. Sidescreen areas are automatically started when the operating software starts up and are only terminated when the system shuts down. This means that the system loads the assigned scripts once when the Sidescreen component is activated for the first time.

This picture below shows the objective of this section.



To activate “Sidescreen” machine data must be set, which is located in the Setup area of the controller:



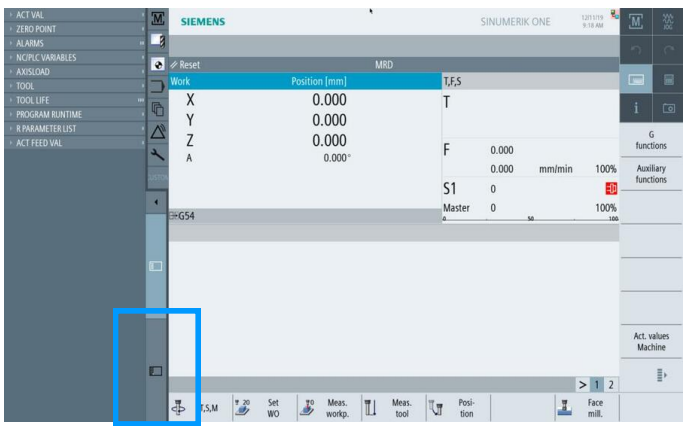
Set “Display MD” 9114 $MM_SIDESCREEN = 1 (Sidescreen activated)

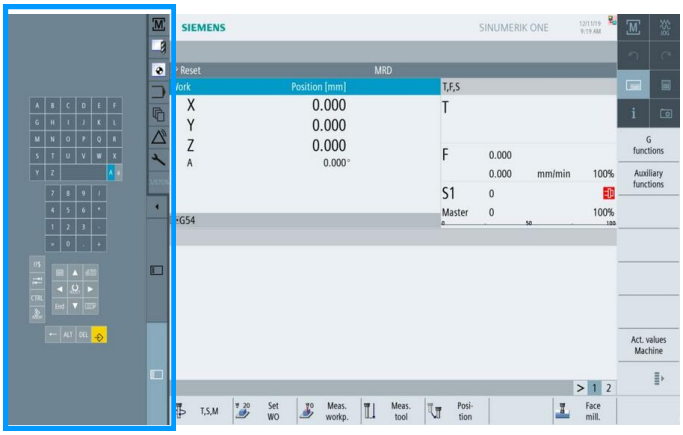
Notes:

Alpha numerical keyboard Sidescreen

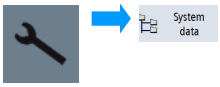
This example will create a keyboard on the side screens.

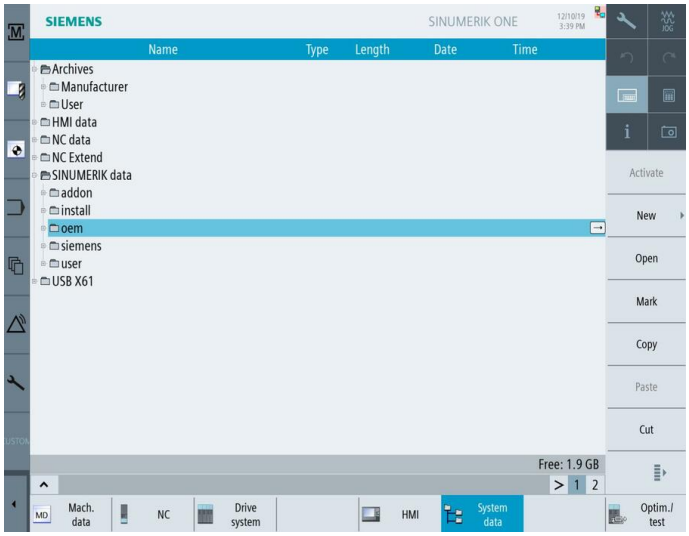
This picture below shows the objective of this section.





The files required for the Alpha numerical keyboard in the Sidescreen area need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





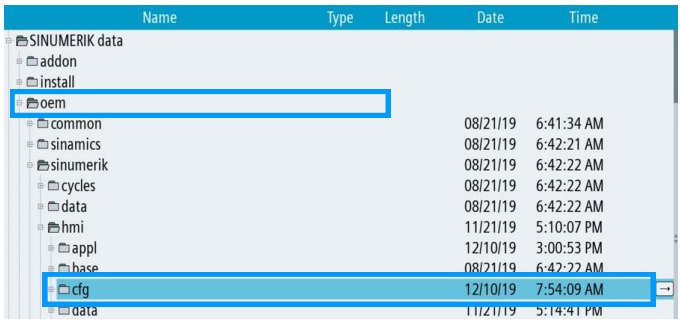
Creating the “slsidescreen.ini” file on the controller

This file sets up the link for the keyboard in the Sidescreen. The file has to be located in one of the following paths/ directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/cfg”

IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/cfg

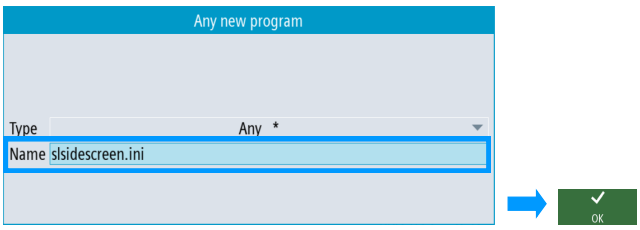
Use the cursor keys to locate and open this directory.



Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing.

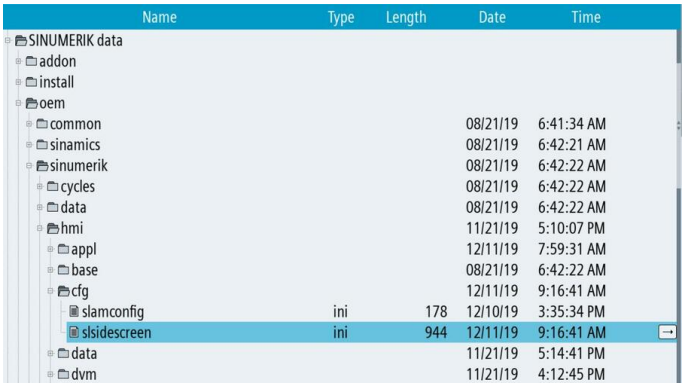
Example: slsidescreen.ini file
The following example will show the content of the “**slsidescreen.ini**” file for the “Keyboard” in the Sidescreen area of the controller.

```
[Sidescreen]

PAGE002= name:=KeyboardPage, implementation:=SlGfwSideScreenPage
```



Below: The **slsidescreen.ini** file located in the **cfg** directory

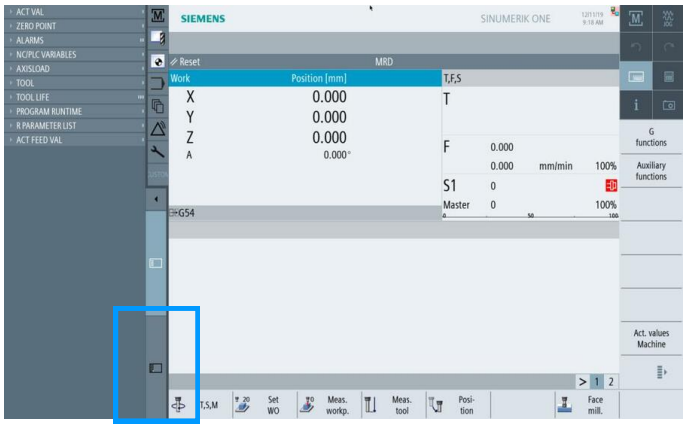


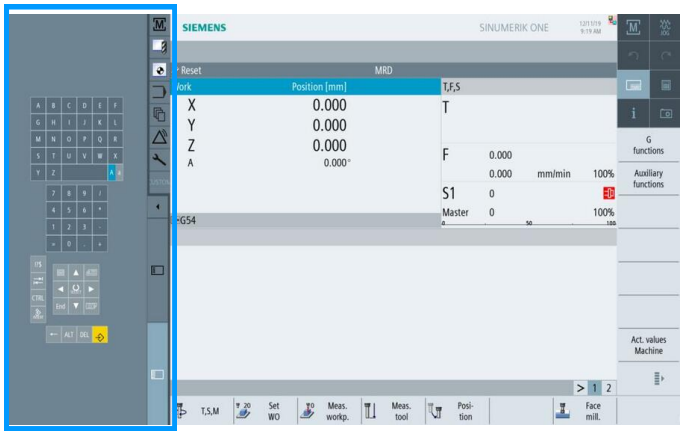
To make active perform.



Notes:

The result is shown below.

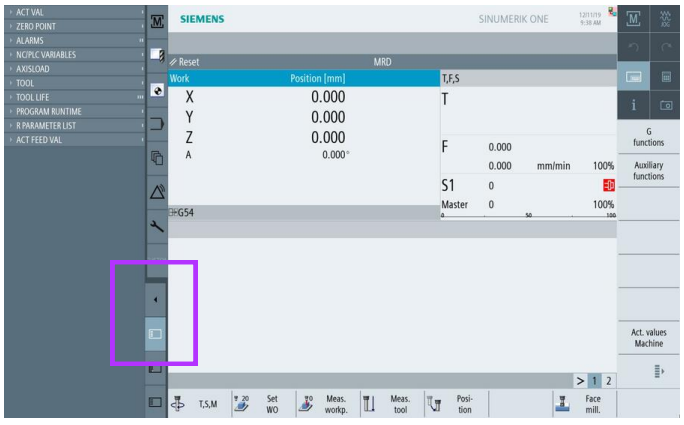


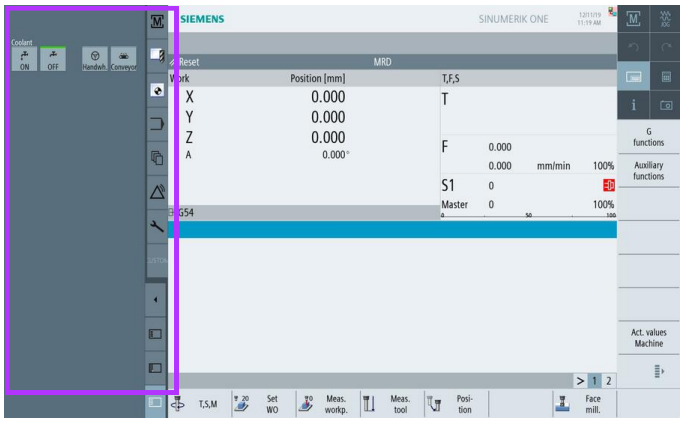


Creating an OEM control panel Sidescreen

This example will create an OEM Control Panel on the side screens.

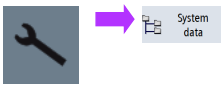
This picture below shows the objective of this section.

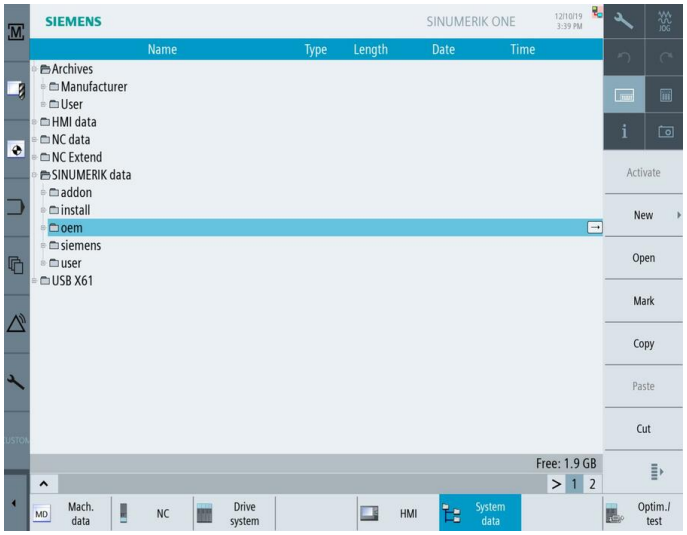




Notes:

The files required for the OEM Control Panel Sidescreen area need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





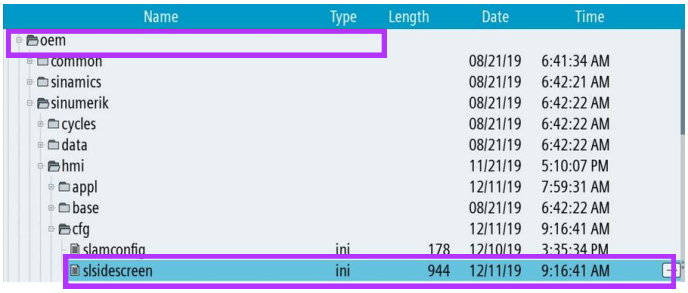
Creating the “slsidescreen.ini” file on the controller

This file sets up the link for the OEM control panel in the Side-screen area. The file has to be located in any of the following paths/directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



In this example the “slsidescreen.ini” file created earlier in the module is going to be modified. The “Open” soft-key allows the file to be opened and then edited by the user

The file will be opened in preparation for editing.

Example: slsidescreen.ini file
The following example will show how the “**slsidescreen.ini**” file is used to specify the “Keyboard” and the “OEM Control Panel” the Sidescreen area of the controller.

```
[Sidescreen]

PAGE002= name:=KeyboardPage, implementation:=SlGfwSideScreenPage

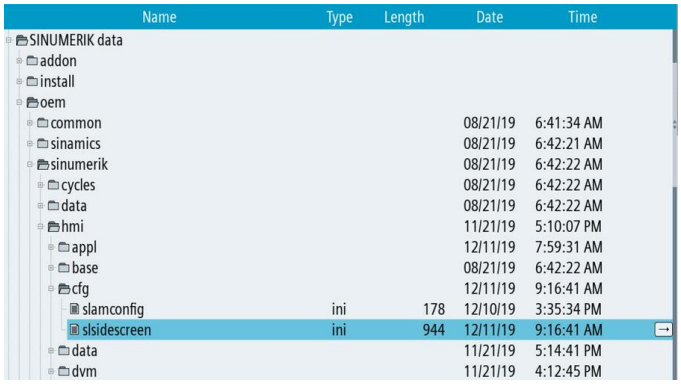
PAGE003= name:=McpPage, implementation:=slsidescreenmcppage.SlSideScreenMcpPage
```





Notes:

Below: The **slsidescreen.ini** file located in the **cfg** directory



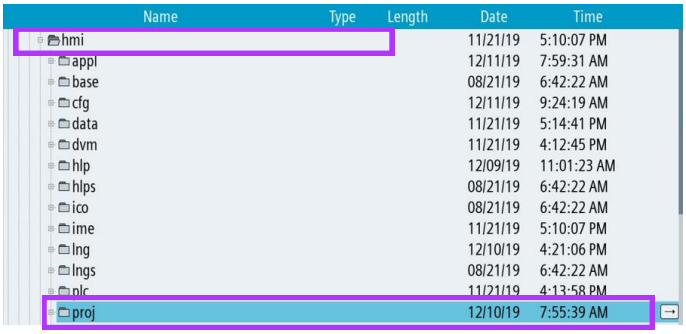
Creating the "slsidescreenmcppageconfig.xml" file on the controller for the "OEM Control Panel"

This file sets up the link for the OEM Control Panel in the Sidescreen. The file has to be located in one of the following paths/directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/proj

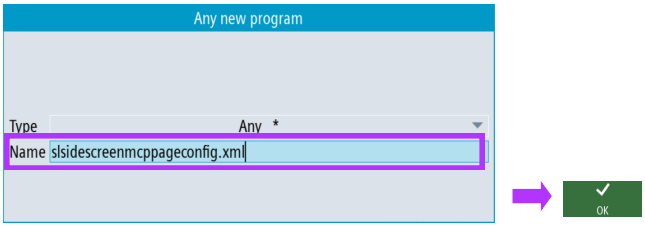
IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/proj

Use the cursor keys to locate and open this directory.



To create the file select the "New" soft-key and ensure that the "Any" soft-key is selected.





Notes:

The newly created file will be automatically opened in preparation for editing.

Example: slsidescreenmcppageconfig.xml file
The following example will show the content "**slsidescreenmcppageconfig.xml**" file in the "OEM Control Panel" in the side-screen area of the controller.

```
<?xml version="1.0" encoding="utf-8" ?>

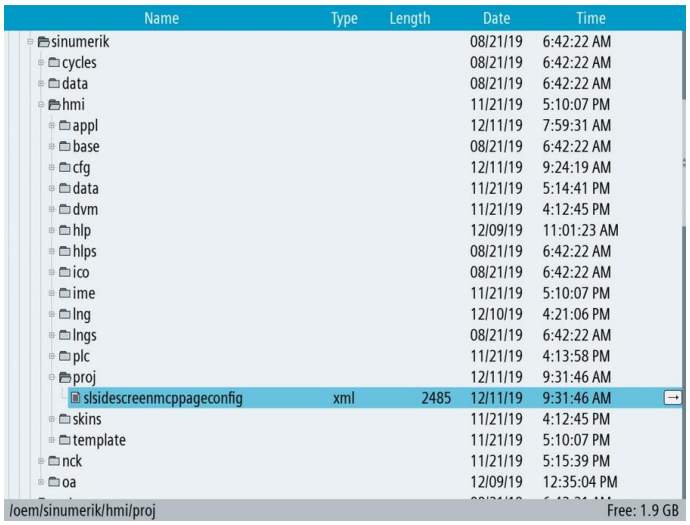
<sideScreenControlPanel>
<mainBlock id="sideScreenMcpBlock">
<!-- block 1 -->
<block position="1">
<row position="1">
<buttonGroup spacing="false" title="COOLANT">
<button position="1" text="ON" icon="slsidescreenmcp_cooling.png">
<onPressed address="HmiUserKey1"/>
<active address="HmiUserLed1"/>
</button>
<button position="2" text="OFF" icon="slsidescreenmcp_cooling_off.png">
<onPressed address="HmiUserKey2"/>
<active address="HmiUserLed2"/>
</button>
</buttonGroup>
</row>
</block>
```

```
<!-- block 2 -->
<block position="2">
<row position="1">
<button position="1" text="Handwh." icon="slsidescreenmcp_handwheel.png">
<onPressed address="HmiUserKey3"/>
<active address="HmiUserLed3"/>
</button>
<button position="2" text="Conveyer" icon="slsidescreenmcp_chip_conveyor.png">
<onPressed address="HmiUserKey4"/>
<active address="HmiUserLed4"/>
</button>
</row>
</block>
</mainBlock>
</sideScreenControlPanel>
```



Notes:

Below: The **slsidescreenmcppageconfig.xml** file located in the **proj** directory



Adaptation options of the file "slsidescreenmcppageconfig.xml"

You can make the settings for your own keys in the <sideScreenControlPanel> tag:

Tag	Description	
<mainBlock id="sideScreenMcpBlock">	Is the main section in the Sidescreen display	
<block position="1" title="SPINDLE">	A main section has any number of blocks.	
	position	Order of the blocks
	title	Block header
<row position="1">	Each block has any number of lines.	
	position	Positioning of the lines
<buttonGroup spacing="false" title="COOLANT">	A key group combines two keys under one header.	
	spacing	The property enables the display type of the keys to be determined: together "false" or separate "true". The default is separate.
	title	All the keys in a key group have the same header.

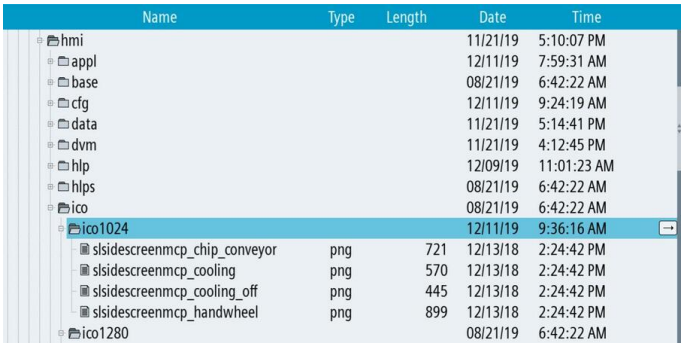
Notes:

Tag	Description	
<button position="1" text="ON" icon="SLicons_kuehlmittel.png" style="redStyle" title="Spindle">	Defines the key properties:	
	position	Position of the key
	text	Labeling shown on the key
	icon	Icon of the key
	style	Changes the background color assigned to the key Possible values are: "redStyle" for a red key "greenStyle" for a green key. The default is "defaultStyle".
	title	A key that does not belong to any group can have its own header.
<onPressed address="DBxxx.DBXyy.y"/>	If the key is pressed, then the PLC bit is set to 1. The value remains at 1 as long as the key remains actuated. With a brief press, the value remains at 1 for at least 300 ms.	
	address	Addressed PLC bit of the key
<active address="DBxxx.DBXyy.y"/>	The key is shown as active (LED animation at the top edge of the key).	
	address	PLC bit from which the active identifier is derived

Storing own icons

You can create your own icon files under one of the following paths:
/oem/sinumerik/hmi/ico or
/user/sinumerik/hmi/ico
Depending on the screen resolution, under the following sub-directories:
/ico1024 for HD Ready
/ico1920 for Full HD

In the example, four icons are used:
slsidescreenmcp_chip_conveyor.png
slsidescreenmcp_cooling.png
slsidescreenmcp_cooling_off.png
slsidescreenmcp_handwheel.png











To make active perform.



Notes:

Buttons
Buttons are 50 pixels x50 pixels

Button colour
By default buttons are grey in colour, but red or green can be specified using the “style” tag

Button Icons
Icons are of a PNG file type, and are 50 pixels wide x 29 pixels high, with a transparent background, using the “icon” tag.

Button TEXT
Text will fit into the lower part of the button 50 pixels x 21 pixels, using the “text” tag.

Example
<button position="1" text="AUTO" icon="slsidescreenmcp_cooling_off.png" style="greenStyle">

Button Interface signals
If the key is pressed, then the PLC bit associated with the alias name in "address" becomes set to 1. A valid alias name for onPressed" has the following form: HmiUserKeyXX - where XX is a whole number is between 1 and 64. On a SINUMERIK HmiUserKey1 to HmiUserKey64 refer to DB19.DBB404.0 - DB19.DBB411.7 As long as the button is pressed, the value remains at 1. With a short press, the value stays at 1 for at least 300 milliseconds.

Example
<onPressed address="HmiUserKey1"/>

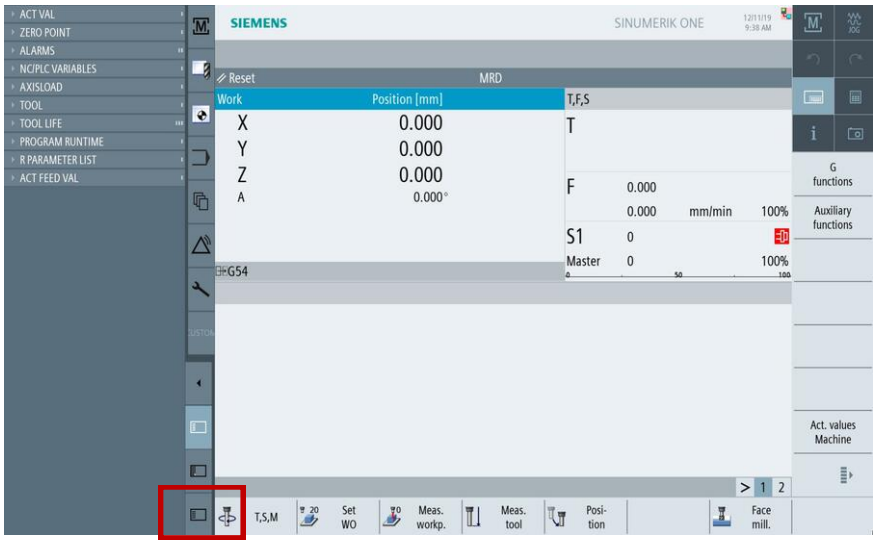
LED Interface signals

With "active" a key can be displayed actively (LED at the upper edge lights up). A valid alias name for "active" has the following form: HmiUserLedXX - where XX is a whole number is between 1 and 64. On a SINUERIK ONE, HmiUserLed1 to HmiUserLed64 refer to DB19.DBB412.0 - DB19.DBB419.7.

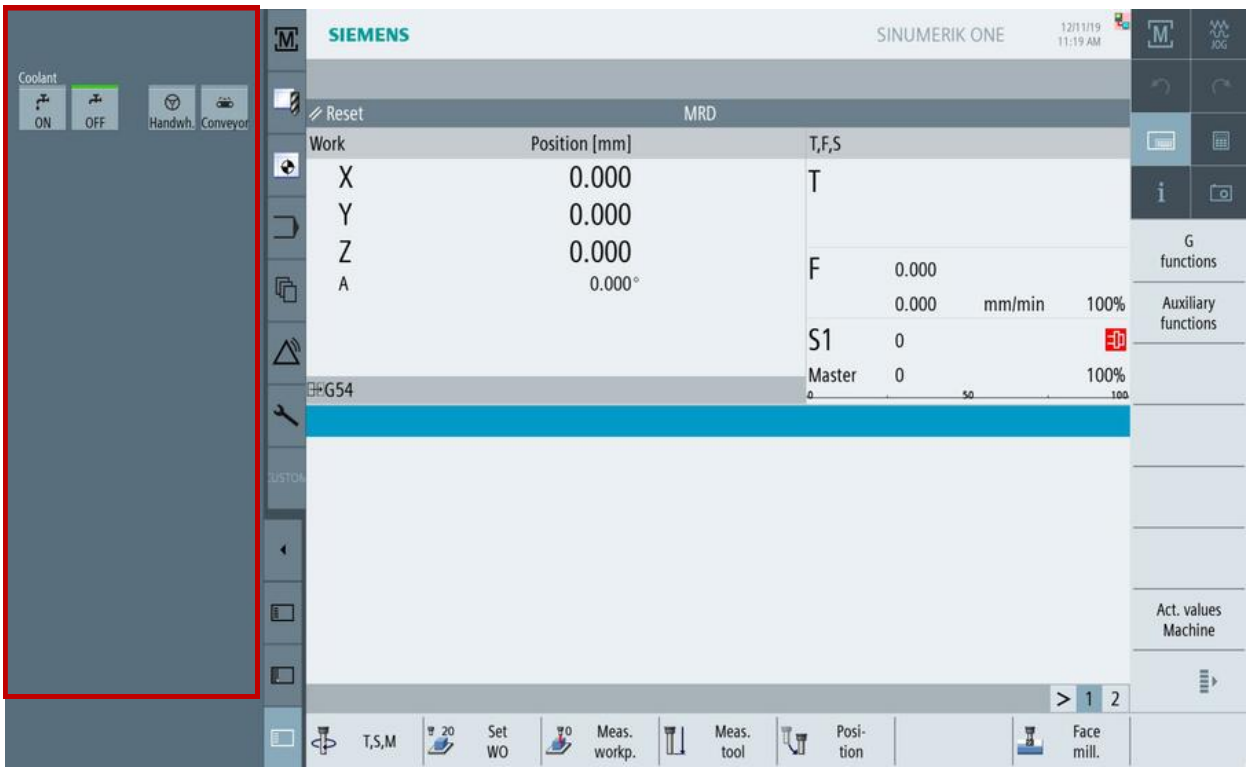
Example
<active address="HmiUserLed1"/>

To make active a complete power off/on of the control is required.

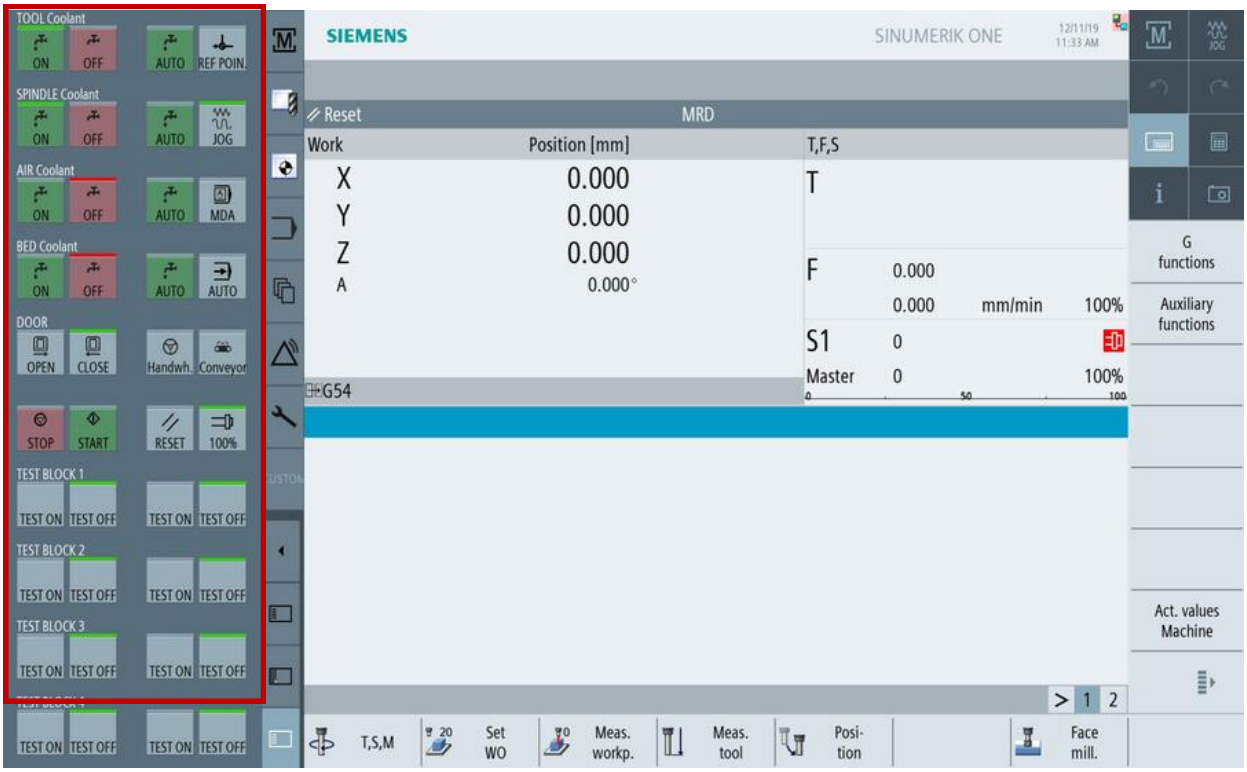
The result is shown below.



Notes:



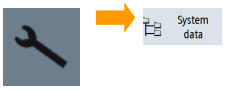
OEM Control Panel Sidescreen example expanded is show below.

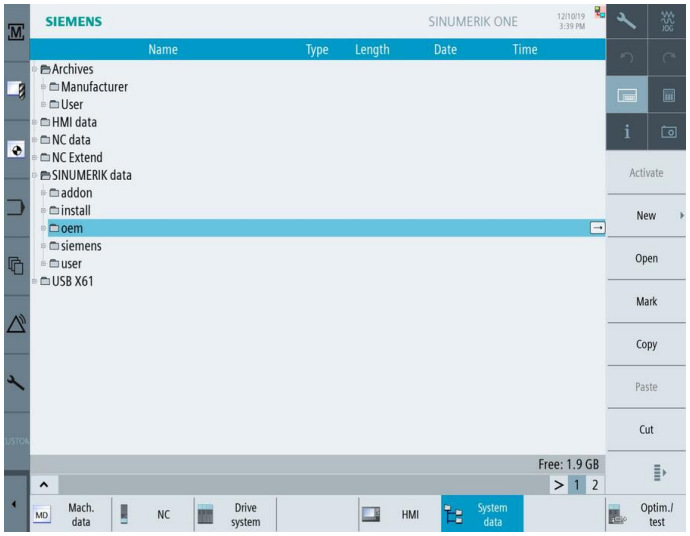


Notes:

Creating a language dependant file for the OEM control Panel

The files required for the Custom Control Panel language in the Sidescreen function need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





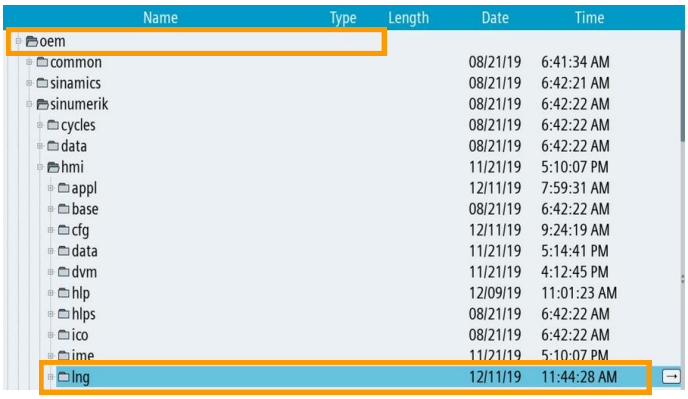
Creating the “slsidescreenmcppage_deu.ts” file on the controller

This file sets up the link for the keyboard in the sidescreen. The file has to be located in one of the following paths/ directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/lng

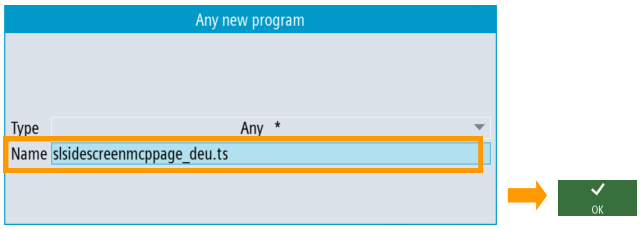
IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/lng

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing.

Example: slsidescreenmcppage.ts file
The following example will show the content of the “**slsidescreenmcppage.ts**” file for the “Keyboard” in the sidescreen area of the controller.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS>
<TS>
<context>
<name>sideScreenMcpBlock</name>
<!-- Title above the buttons -->
<message>
<source>COOLANT</source>
<translation>KUHLMITTEL</translation>
```

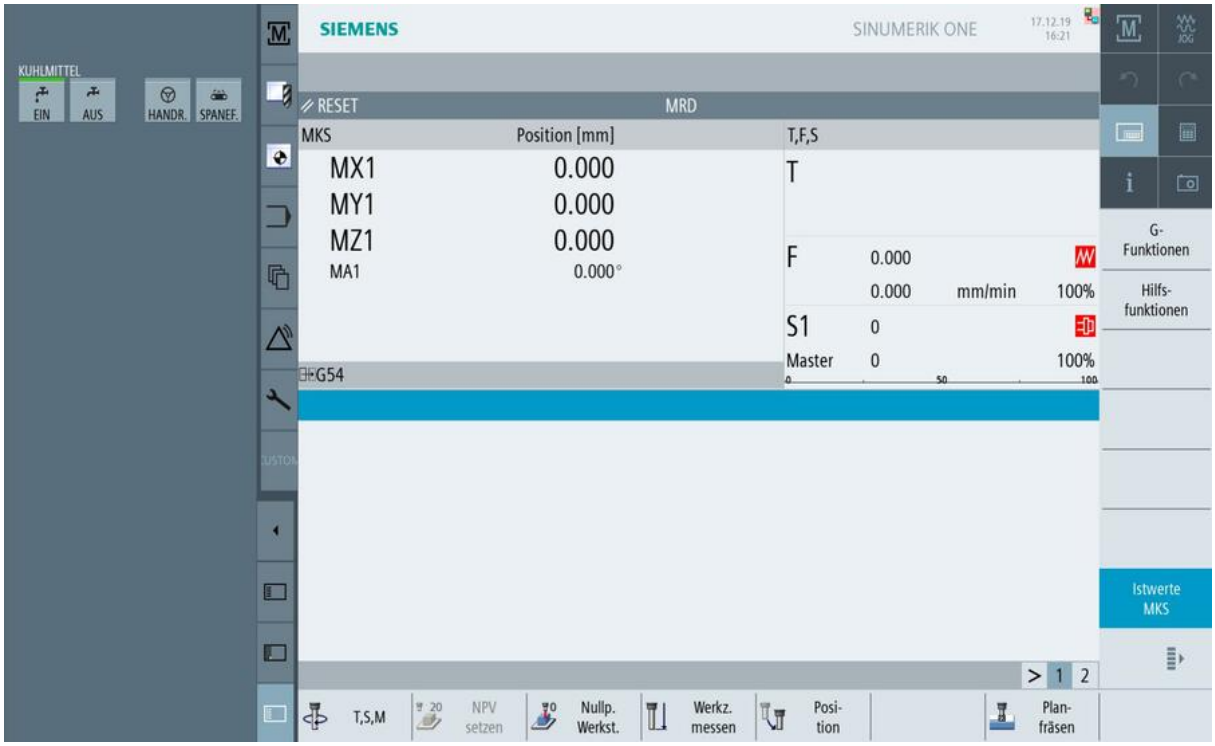
Notes:

```
<chars>16</chars>
</message>
<!-- Button text -->
<message>
<source>ON</source>
<translation>EIN</translation>
<chars>8</chars>
</message>
<!-- Button text -->
<message>
<source>OFF</source>
<translation>AUS</translation>
<chars>8</chars>
</message>
<!-- Button text -->
<message>
<source>Handwh.</source>
<translation>HANDR.</translation>
<chars>8</chars>
</message>
<!-- Button text -->
<message>
<source>Conveyor</source>
<translation>SPANEF.</translation>
<chars>8</chars>
</message>
</context>
</TS>
```



To make active a complete **power off/on** of the control is required.

The result is shown below.

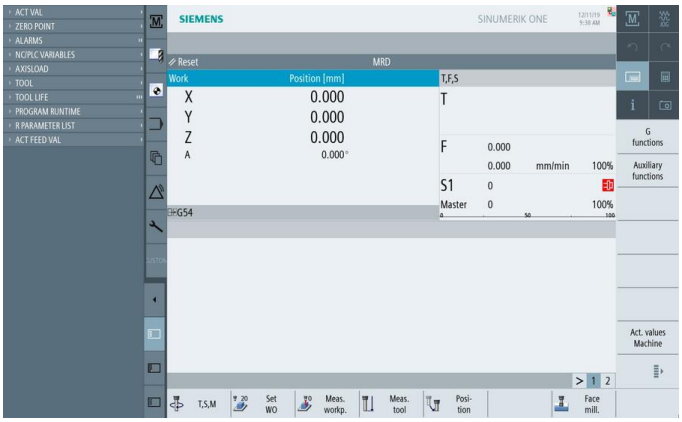


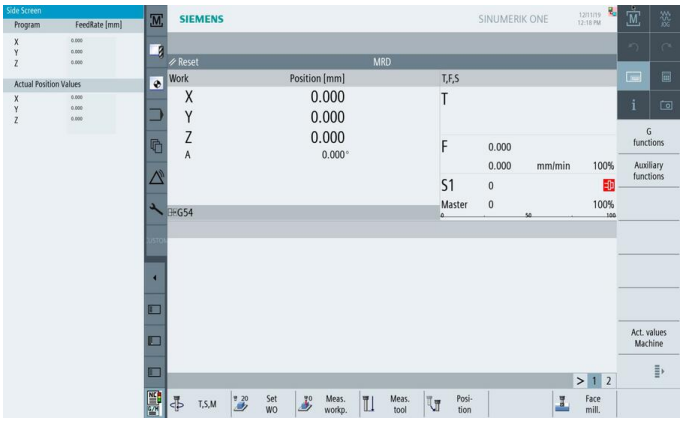
Notes:

Creating an OEM SideScreen

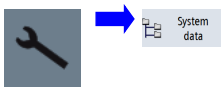
This example will create a keyboard on the side screens.

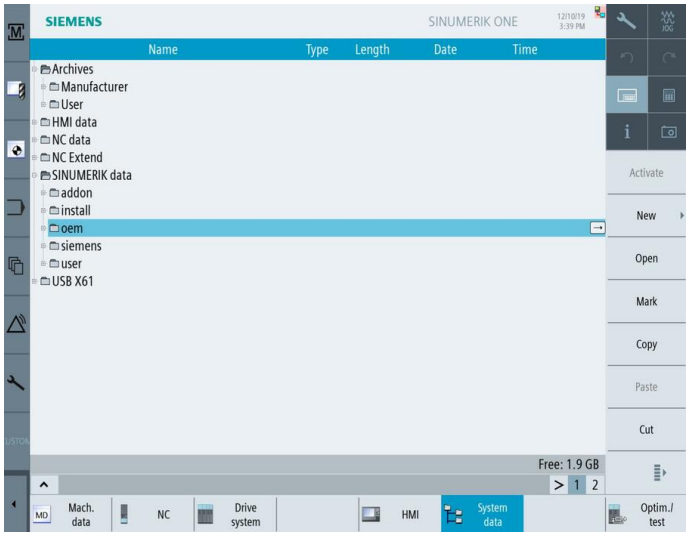
This picture below shows the objective of this section.





The files required for the Alpha numerical keyboard in the Sidescreen area need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





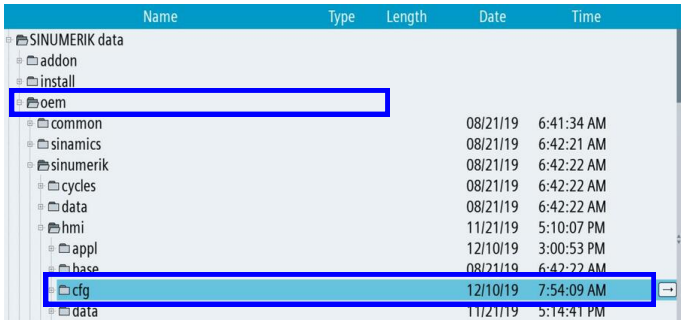
Creating the “slsidescreen.ini” file on the controller

This file sets up the link for the keyboard in the Sidescreen. The file has to be located in one of the following paths/ directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/cfg”

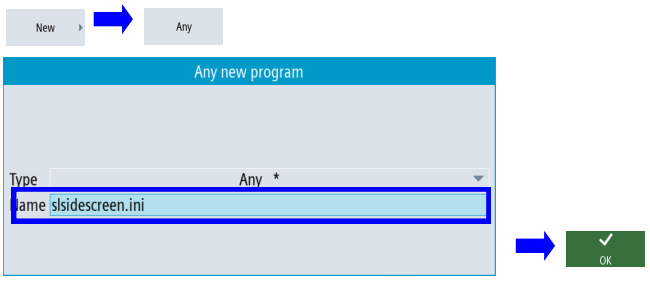
IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.

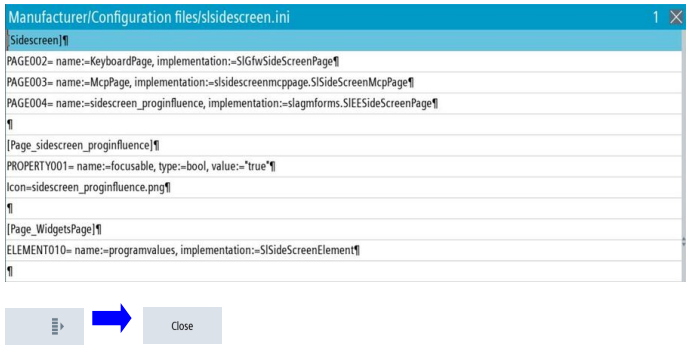


The newly created file will be automatically opened in preparation for editing.

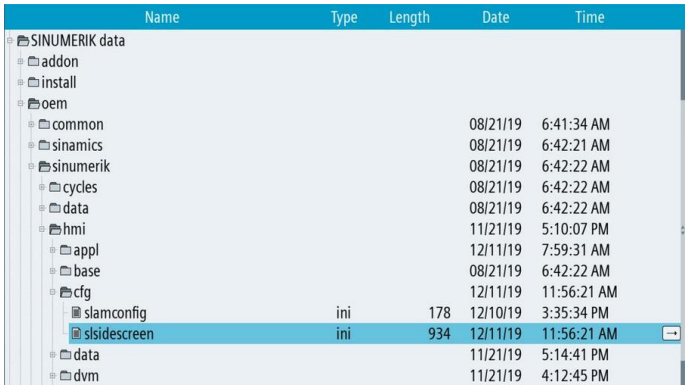
Example: slsidescreen.ini file
The following example will show the content of the “**slsidescreen.ini**” file for the OEM SideScreen in the Sidescreen area of the controller.

```
[Sidescreen]
PAGE004= name:=sidescreen_proginfluence, implementation:=slagmforms.SlEESideScreenPage

PAGE003= name:=McpPage, implementation:=slsidescreenmcppage.SlSideScreenMcpPage
```



Below: The **slsidescreen.ini** file located in the **cfg** directory



Notes:

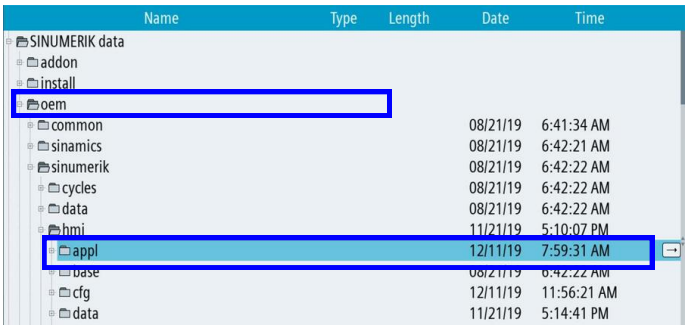
Creating the "sidescreen_proginfluence.xml" file on the controller for the "OEM SideScreen"

This file sets up the link for the OEM Side Screen in the Side-screen area. The file has to be located in one of the following paths/directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/appl

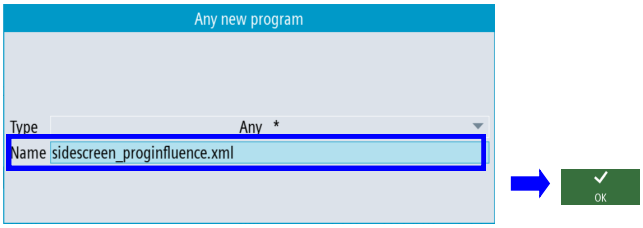
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/appl

Use the cursor keys to locate and open this directory.



To create the file select the "New" soft-key and ensure that the "Any" soft-key is selected.





The newly created file will be automatically opened in preparation for editing.

Example: sidescreen_proginfluence.xml file
The following example will show the content of the "**sidescreen_proginfluence.xml**" file of the "OEM Control Panel" in the side-screen area of the controller.

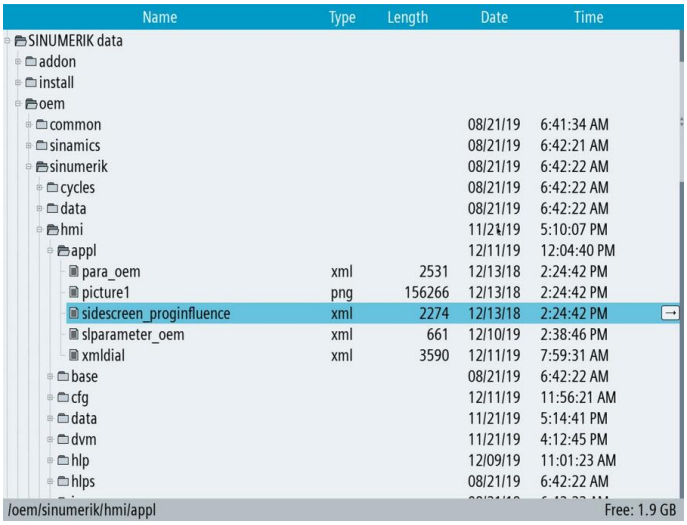
```
<DialogGui >
<menu name = "main">
<open_form name = "prog_infuence_form" />
</menu>
<form name = "prog_infuence_form">
<init>
<caption>Program                          FeedRate [mm]</caption>
<data_access type="true" />
<control name = "b1" xpos = "125" ypos = "60" refvar="nck/Channel/GeometricAxis/actFeedRate[1]"
hotlink="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" col-
or_bk="#e5ecf0"/>
<control name = "b2" xpos = "125" ypos = "80" refvar="nck/Channel/GeometricAxis/actFeedRate[2]"
hotlink="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" col-
or_bk="#e5ecf0"/>
<control name = "b3" xpos = "125" ypos = "100" refvar="nck/Channel/GeometricAxis/actFeedRate
[3]" hotlink="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast"
color_bk="#e5ecf0"/>
<control name = "b4" xpos = "125" ypos = "165" refvar="nck/Channel/GeometricAxis/actProgPos[1]"
hotlink="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" col-
or_bk="#e5ecf0"/>
<control name = "b5" xpos = "125" ypos = "185" refvar="nck/Channel/GeometricAxis/actProgPos[2]"
hotlink="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" col-
or_bk="#e5ecf0"/>
>
```

Notes:

```
<control name = "b6" xpos = "125" ypos = "205" refvar="nck/Channel/GeometricAxis/actProgPos[3]"
hotlink="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" col-
or_bk="#e5ecf0"/>
</init>
<paint>
<text xpos = "23" ypos = "60">X</text>
<text xpos = "23" ypos = "80">Y</text>
<text xpos = "23" ypos = "100">Z</text>
<text xpos = "23" ypos = "165">X</text>
<text xpos = "23" ypos = "185">Y</text>
<text xpos = "23" ypos = "205">Z</text>
<box xpos="0" ypos="0" width="276" height="26" color="#0099C8" />
<text xpos="8" ypos="2" color="#ffffff">Side Screen</text>
<box xpos="0" ypos="27" width="276" height="26" color="#c5d2da" />
<text xpos="23" ypos="28" color="#000000">Program                    FeedRate [mm]</text>
<box xpos="0" ypos="137" width="276" height="26" color="#c5d2da" />
<text xpos="23" ypos="138" color="#000000">Actual Position Values</text>
</paint>
</form>
</DialogGui>
```



Below: The **sidescreen_proginfluence.xml** file located in the **appl** directory



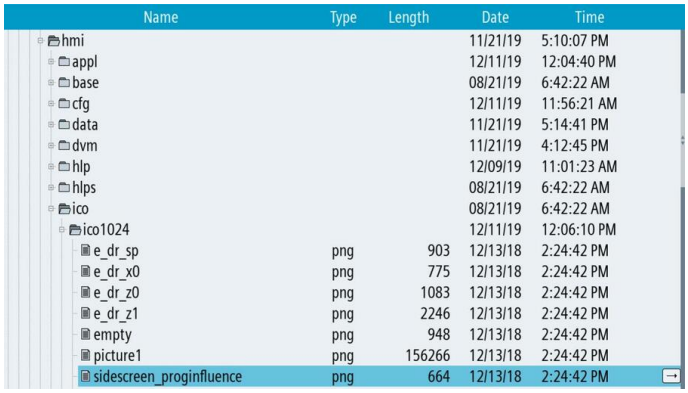
Storing of icons

You can create your own icon files under one of the following paths:
/oem/sinumerik/hmi/ico or
/user/sinumerik/hmi/ico

Depending on the screen resolution, under the following sub-directories:
/ico1024 for HD Ready
/ico1920 for Full HD

In the example, one icons is used:
sidescreen_proginfluence.png



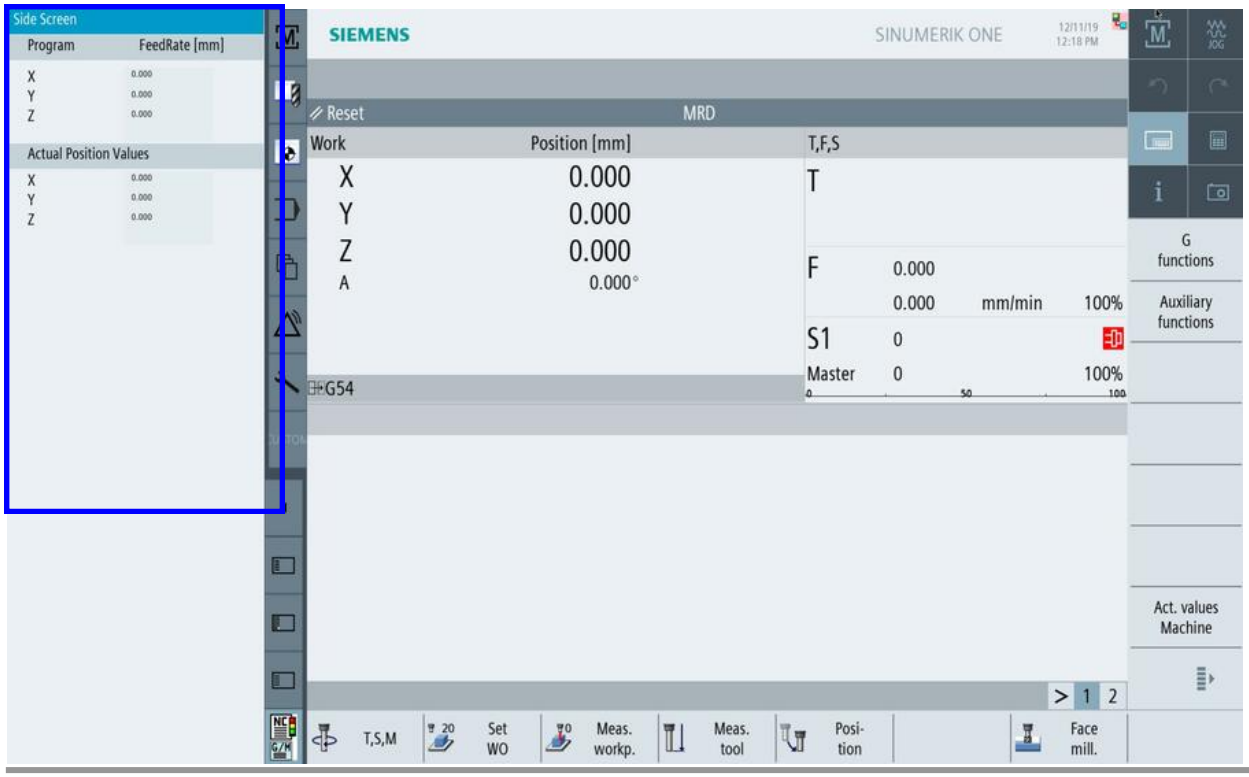


To activate the files, perform:



Notes:

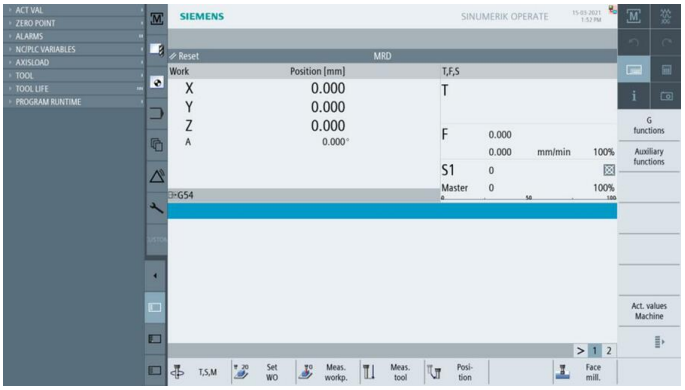
The result is shown below.

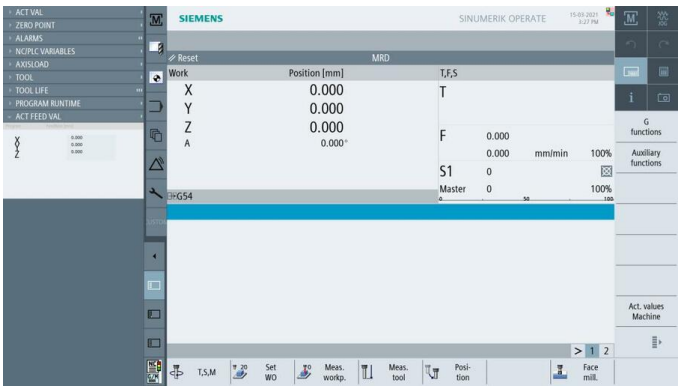


Creating an OEM SideScreen Widget

This example will create an OEM Sidescreen widget on the side screens.

This picture below shows the objective of this section.

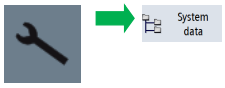


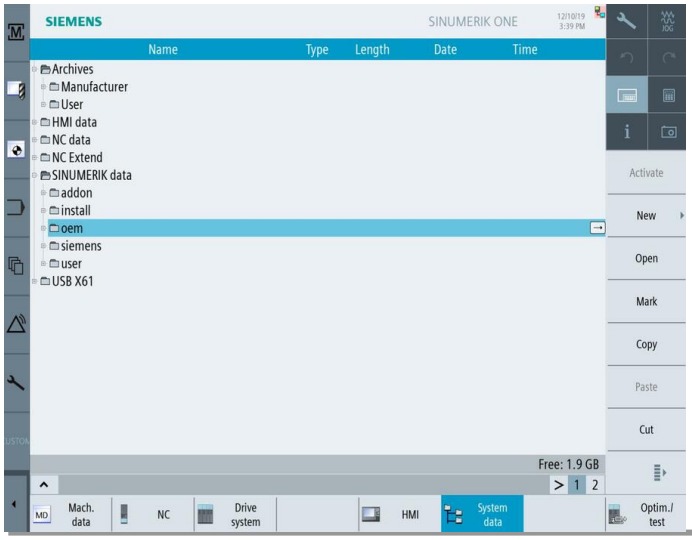


Notes:

The Sidescreen widget frame gives a quick and permanent view of important information and enable you to operate virtual keys for quick selection

The files required for the Widget in the Sidescreen area need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





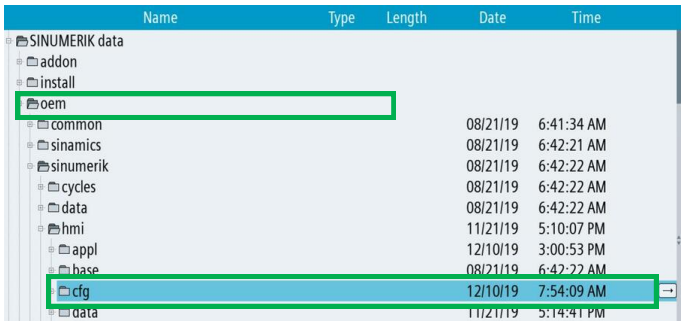
Creating the “slsidescreen.ini” file on the controller

This file sets up the link for the keyboard in the Sidescreen. The file has to be located in one of the following paths/ directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/cfg

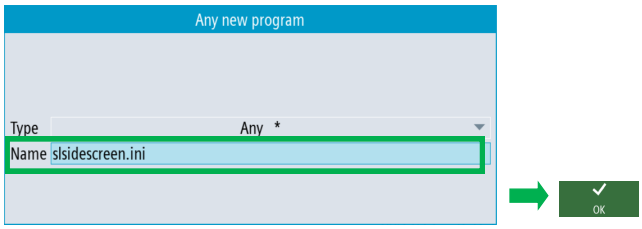
IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





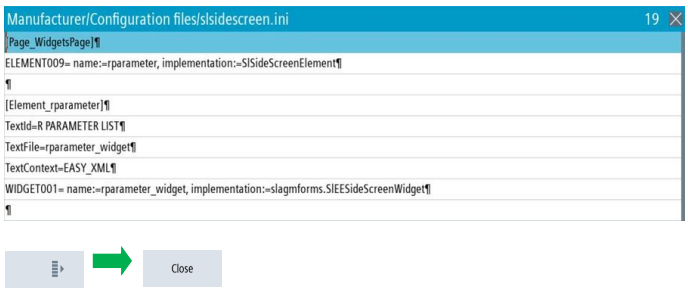
The newly created file will be automatically opened in preparation for editing.

Example: slsidescreen.ini file
The following example will show the content of the “**slsidescreen.ini**” file for the OEM Sidescreen Widget in the Sidescreen area of the controller.

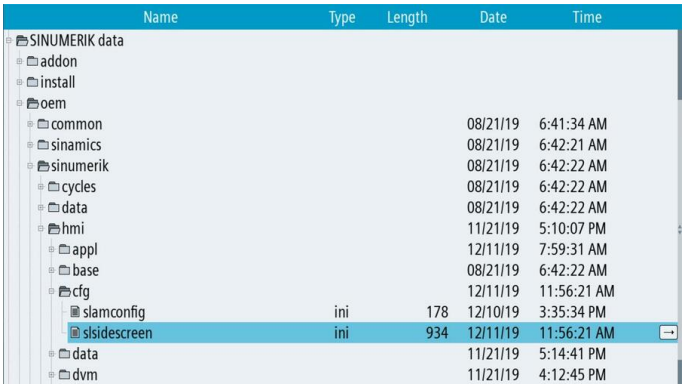
```
[Page_WidgetsPage]
ELEMENT009= name:=Programvalues, implementation:=SlSideScreenElement

[Element_Programvalues]
TextId=ACT FEED VAL
TextFile=programvalues_widget
TextContext=EASY_XML
WIDGET001= name:=programvalues_widget, implementation:=slagmforms.SlEESideScreenWidget
```

Notes:



Below: The **slsidescreen.ini** file located in the **cfg** directory



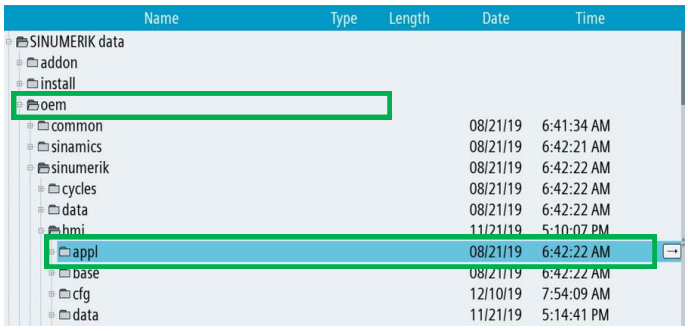
Creating the "programvalues_widget.xml" file on the controller for the "OEM SideScreen widget"

This file sets up the link for the OEM Sidescreen Widget in the Sidescreen. The file has to be located in one of the following paths/directories:-

SINUMERIK Data - CF Card/oem/sinumerik/hmi/appl

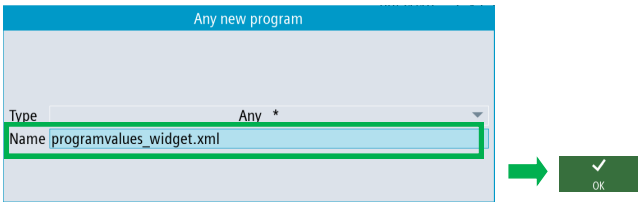
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/appl

Use the cursor keys to locate and open this directory.



To create the file select the "New" soft-key and ensure that the "Any" soft-key is selected.





Notes:

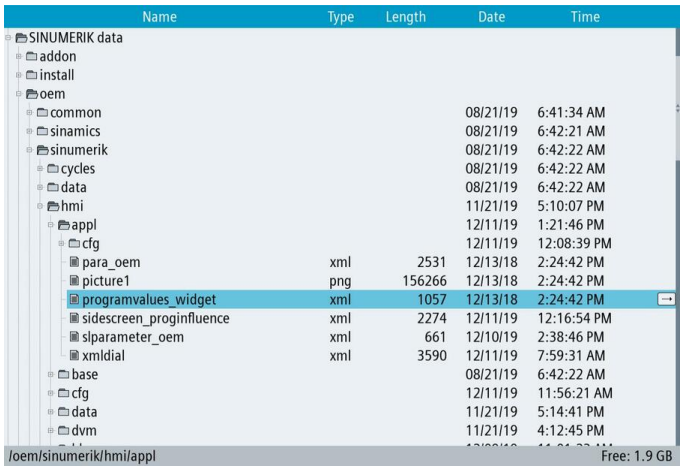
The newly created file will be automatically opened in preparation for editing.

Example: programvalues_widget.xml file
The following example will show the content **"programvalues_widget.xml"** file for the "OEM Sidescreen Widget" in the side-screen area of the controller.

```
<DialogGui >
<menu name = "main">
<open_form name = "PROGRAMVALUES_LIST" />
</menu>
<form name = "PROGRAMVALUES_LIST">
<init>
<caption>Program                          FeedRate [mm]</caption>
<data_access type="true" />
<control name = "b1" xpos = "125" ypos = "34" refvar="nck/Channel/GeometricAxis/actFeedRate[1]"
hotlink="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" col-
or_bk="#e5ecf0"/>
<control name = "b2" xpos = "125" ypos = "54" refvar="nck/Channel/GeometricAxis/actFeedRate[2]"
hotlink="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" col-
or_bk="#e5ecf0"/>
<control name = "b3" xpos = "125" ypos = "74" refvar="nck/Channel/GeometricAxis/actFeedRate[3]"
hotlink="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" col-
or_bk="#e5ecf0"/>
</init>
<paint>
<text xpos = "23" ypos = "34">X</text>
<text xpos = "23" ypos = "54">Y</text>
<text xpos = "23" ypos = "74">Z</text>
</paint>
</form>
```



Below: The **"programvalues_widget.xml"** file located in the **appl** directory

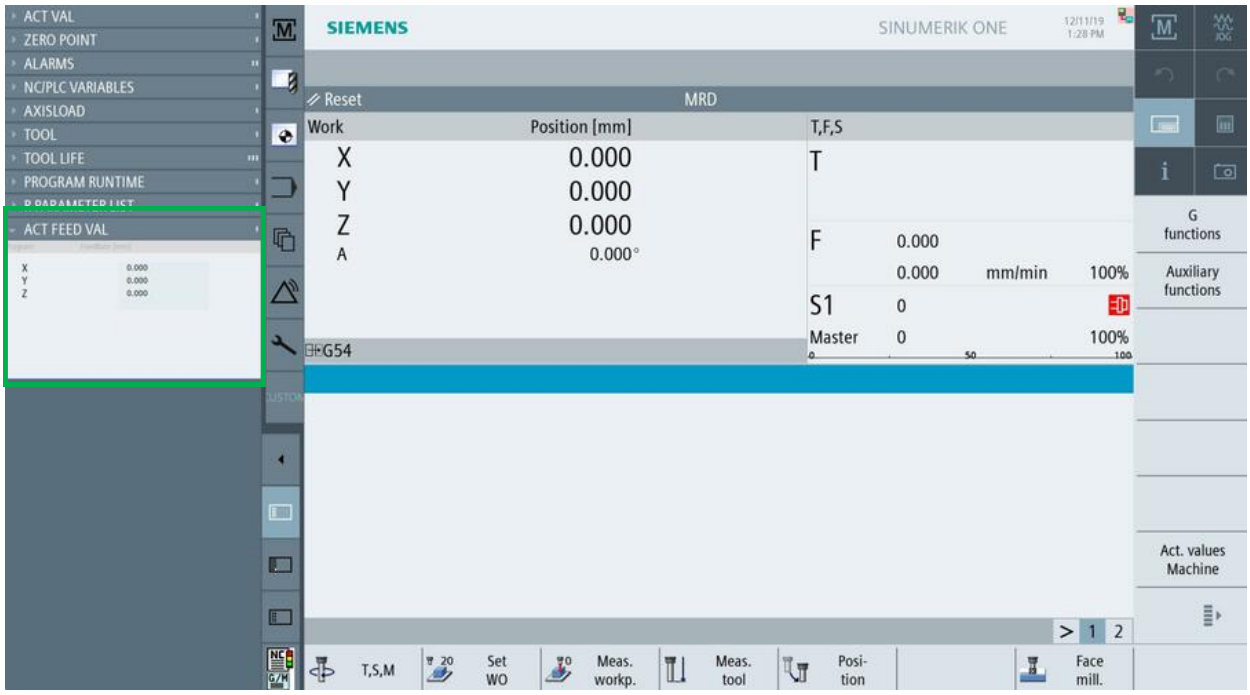


To activate the files, perform:



Notes:

The result is shown below.



129: END

Notes:

131

SINUMERIK

Run MyScreens

Module Description:

Run MyScreens is a function of the SINUMERIK which allows the manufacturer to write custom screens. The OEM screens are intended as an interface to the specific functions offered by the OEM.

Module objective:

Upon completion of this module you will be able to perform the following tasks:

- Create Run MyScreens display screens.

Content:

General Information

Creating CUSTOM screen OEM Files

Basic Configuration Information

Creating Run MyScreens OEM Files

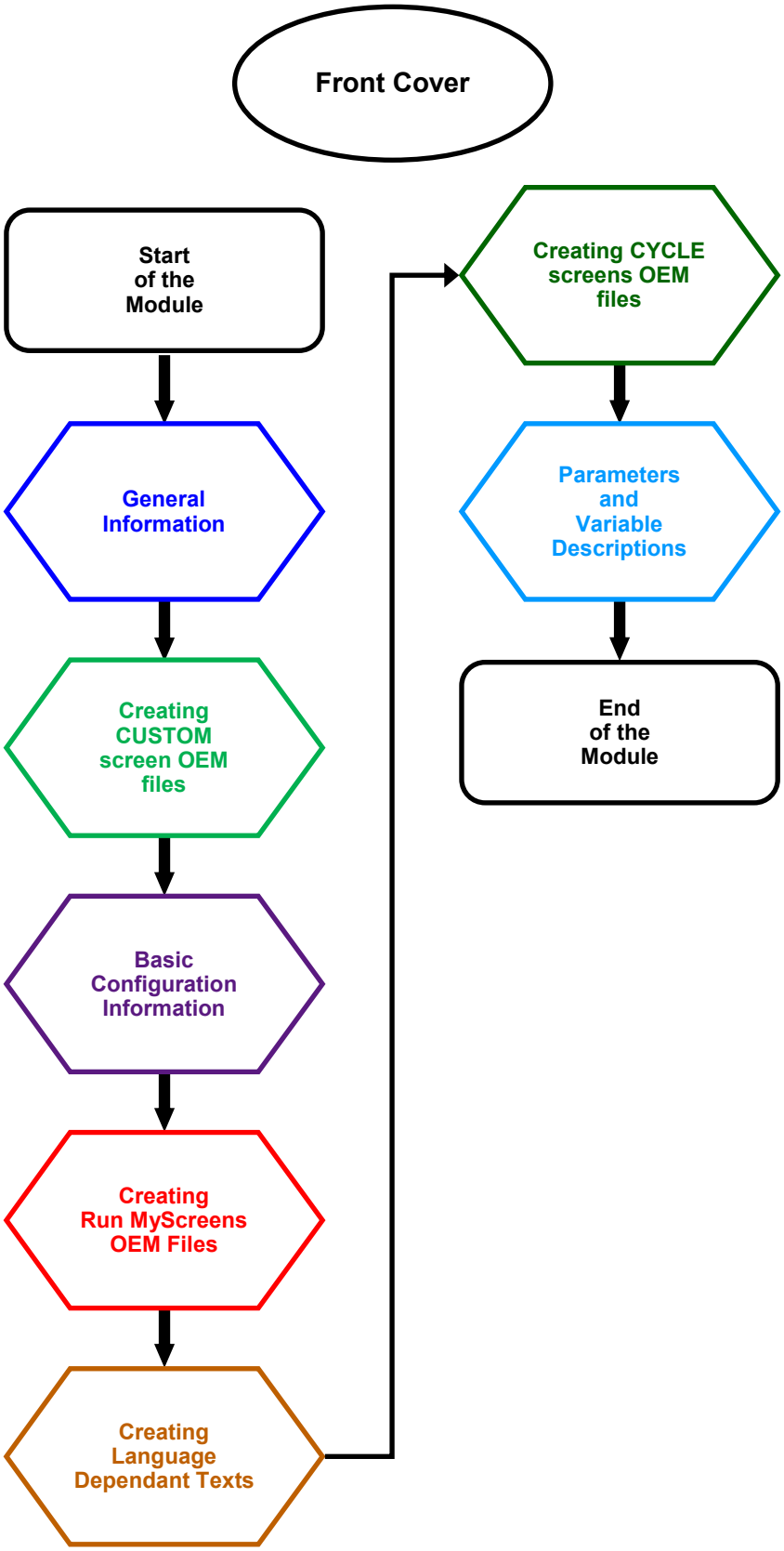
Creating Language Dependent Texts

Creating CYCLE screens OEM Files

Parameters and Variable Description

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
General Information
Creating CUSTOM screen OEM files
Basic Configuration Information
Creating Run MyScreens OEM Files
Creating Language Dependant Texts
Creating CYCLE screens OEM files
Parameters and Variable Descriptions
End of the Module

General Information

The Run MyScreens function is configured using ASCII files. These files contain the descriptions of the user interface.

In order to use images in the screens, a graphic programme is required, e.g. Paint Shop Pro® and Paint® are programmes which are compatible with the SINUMERIK controller.

Run MyScreens can be used for the following functions:

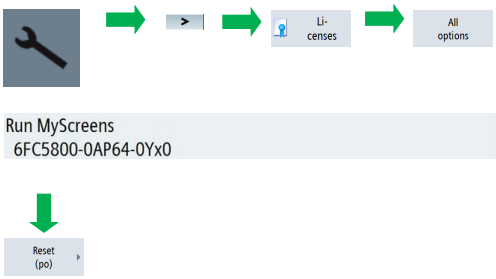
1. Display of dialogs and provision of:
   - Soft-keys
   - Variable tables (this function will be available with a later level of implementation)
   - Text and help text
   - Graphic displays and help displays
2. Calling up dialogs by means of:
   - The (entry) soft-keys
   - PLC selection
3. Dynamic re-structuring of dialogs:
   - Changing or deleting soft-keys
   - Defining and designing variable fields

   Showing, replacing and deleting (language-dependent/ in-dependent) display texts
4. Initiating actions when:
   - Showing dialogs
   - Entering values (variables)
   - Pressing soft-keys
   - Closing dialogues
5.. Exchange of data between dialogs.
6. Variables
   - Reading (NC, PLC, user variables)
   - Writing (NC, PLC, user variables)
   - Linking with mathematical, comparative or logic operators
7. Executing functions:
   - Subroutines
   - File functions
   - PI Services
8. Taking into account levels of protection defined for user groups.

For SINUMERIK ONE, no licence is required.

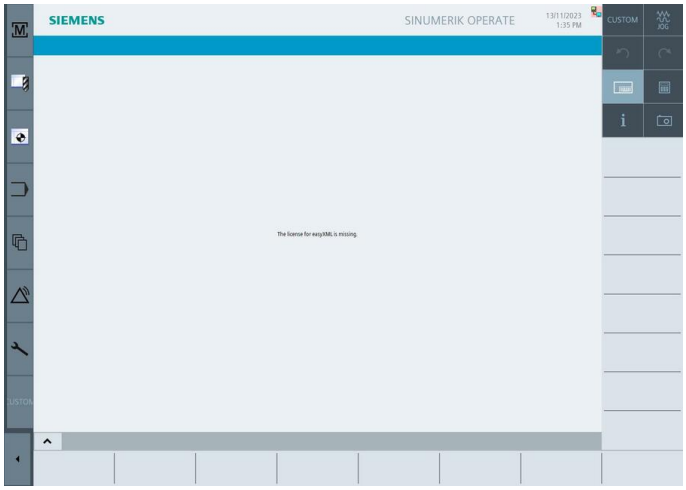
A licence is only required for SINUMERIK 828D when greater than 5 screens are created.

The option - **6FC5800-0AP64-0Yx0 -** needs to be set.



Activate the option with an NCK reset:
After the reset it will be possible to display Easy XML scrip on the HMI.

Below shows the HMI of the CUSTOM are of the MACHINE page on hardware of the SINUMERIK without the license set.

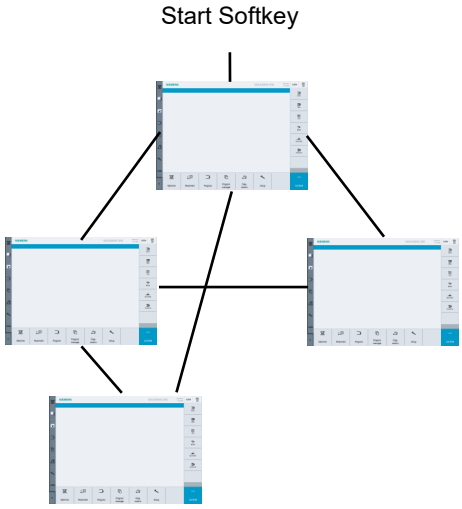


Notes:

Structure of the menu tree

Menu tree principle:

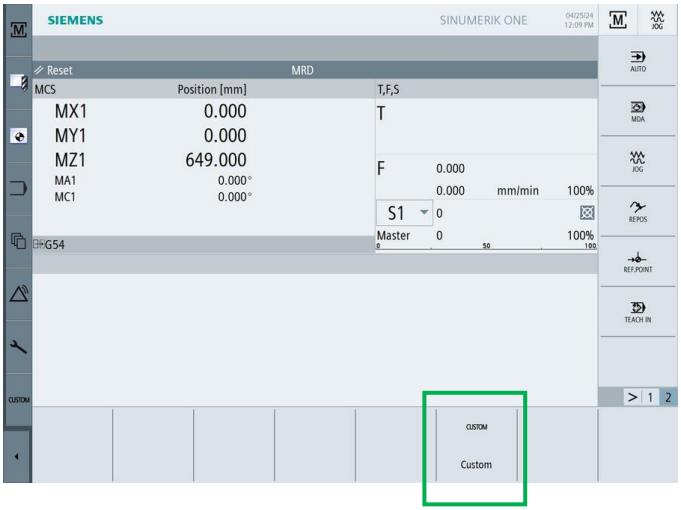
Several interlinked dialogs create a menu tree. A link exists if you can switch from one dialog
to another. You can use the newly defined horizontal/vertical softkeys in this dialog to call the
preceding or any other dialog.
A menu tree can be created behind each start softkey:



CUSTOM Screen OEM files

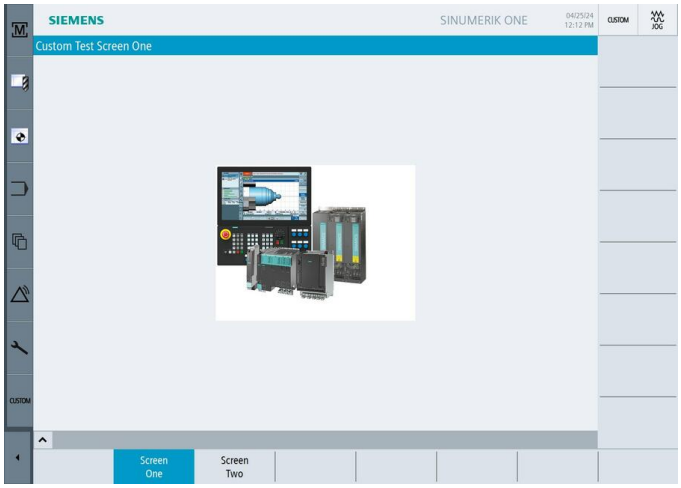
The “CUSTOM softkey is an extra “AREA” softkey so that the OEM can create there own dialogs or additional softkey bars and can be loaded and used for executing additional actions.

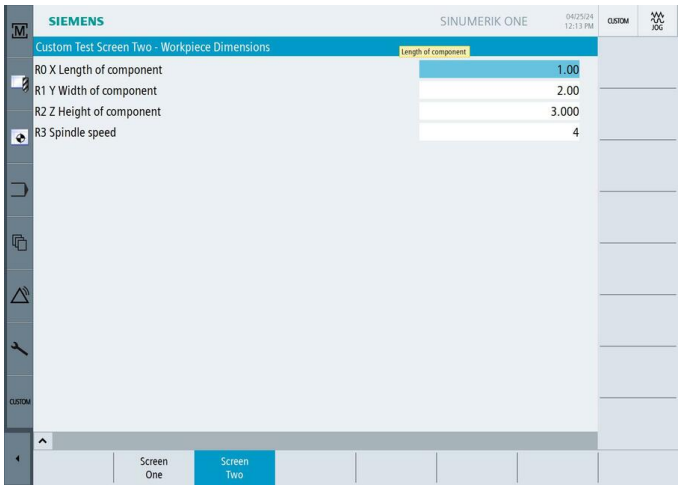
This picture below shows the objective of this section.



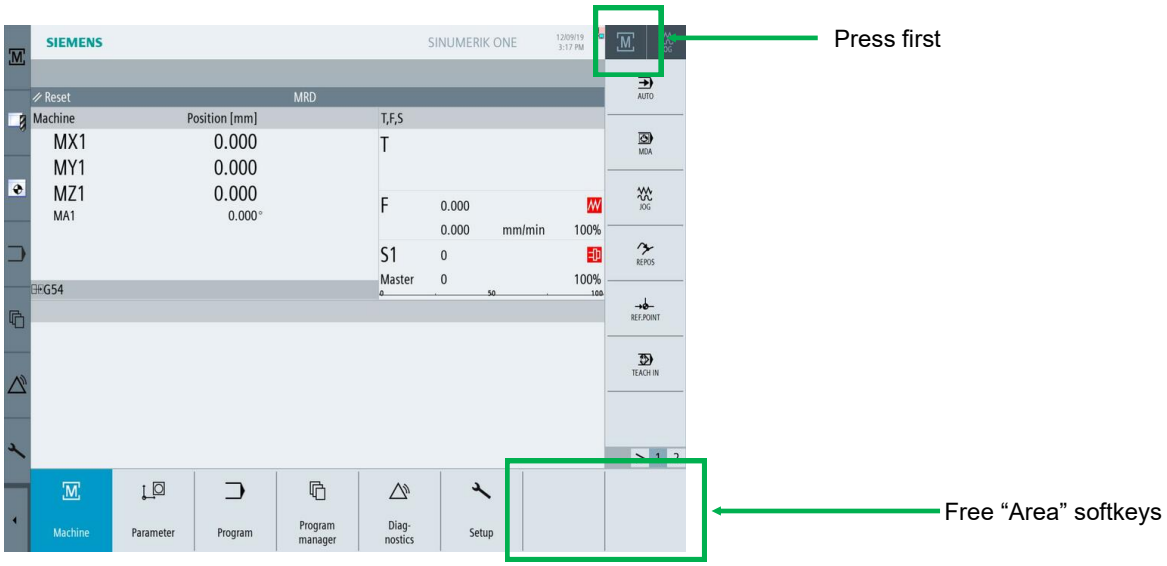


Notes:





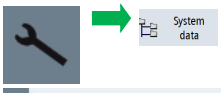
The start point for the “CUSTOM” screens is the “menu select “ (main) area as shown below.

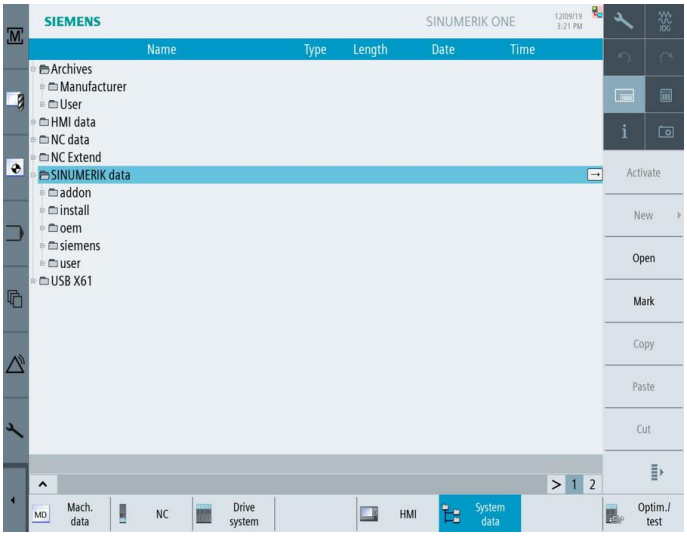


The "custom.ini" file is pre-named by the system and is automatically called when the softkey is activated, this is used to initiate the OEM operating sequences.
The initial menu is defined by the form name "startup" in the "custom.ini" file. You can use the "startup" menu to link additional dialogs or additional softkey bars so that they can be loaded and used for executing additional actions.

Notes:

A configuration file required for the Run MyScreen function, needs to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the folder “System data” area.





The labelling and position of the softkey for the "Custom" operating area are configured in the slamconfig.ini file.

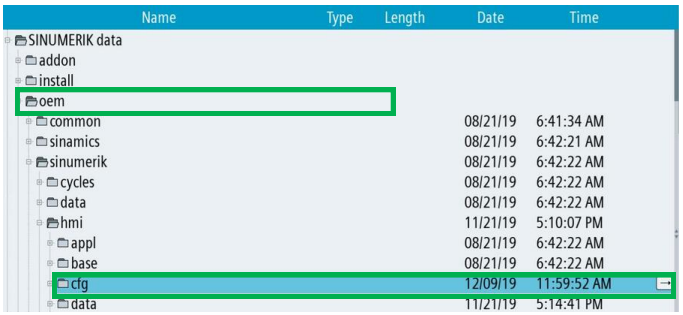
Creating the “slamconfig.ini” file on the controller

This file activates the softkey in the “Area” and activates the **custom.ini** file which is to be used to create the user screens, it also determine which operating areas these files are available in. The file has to be located in one of the following paths/ directories:-

SINUMERIK data - CF Card/oem/sinumerik/hmi/cfg

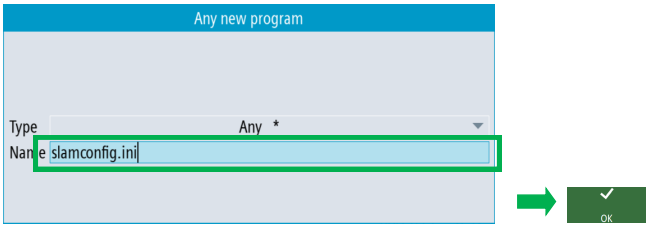
IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing, enter the text below.

Creating the User Files on an external PC

As mentioned previously it is possible to create the Easy XML files directly on the controller or offline using an ASCII editor (e.g. Notepad®).
These files have to be transferred to the controller and placed in the correct folders using the USB Interfaces or with Access MyMachine software.

Example: **slamconfig.ini** file.

It is necessary to specify a soft-key to be used as the entry point to the “MENU” area.

The following example will show content of the “**slamconfig.ini** ” file for the “OEM Control Panel” in the side-screen area of the controller.

```
[Custom]
TextId=Run%nMyScreen
Picture=hd_Custom.png
SoftkeyPosition = 14
Visible=true
```

The example defines the 12th horizontal soft-key as the entry point, the soft-key will have the text “CUSTOM”.

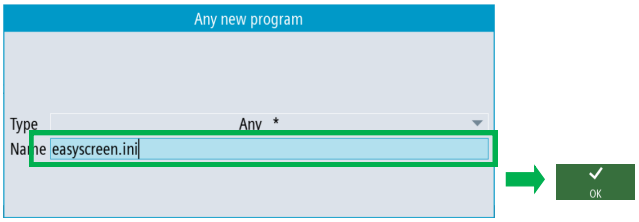


You need the easyscreen.ini and custom.ini files to configure the operating area.
The file easyscreen.ini already contains a definition line for the "Custom" operating area:

Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing.

Example: easyscreen.ini file
The following example will show how the “easyscreen.ini” file is used to specify the “Custom” area of the controller as the area from where the “Run MyScreens” are to be accessed.
(area := Area Custom)

The “com” file, the content of which will determine what appears on the Run MyScreens screens, is also declared here.
(startfile := custom.com)

The first step is to enter the header (**[STARTFILES]**)
The code for specifying the controller area and “com” file is then entered on a new line as shown below.

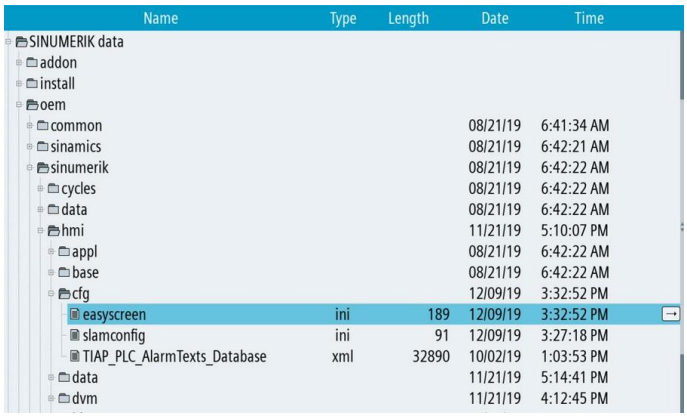
[STARTFILES]

StartFile05 = area := Custom, dialog := SIEsCustomDialog, startfile := custom.com

Below: The **easyscreen.ini** file located in the **cfg** directory

To activate the files, perform:







Creating the “custom.ini” file

The **custom.ini** file has been declared in the “easyscreen.ini” file and now needs to be created.

Code contained within the **custom.ini** file is used to create soft keys and display texts and graphics etc.

The file needs to be created and placed in one of the following folders:

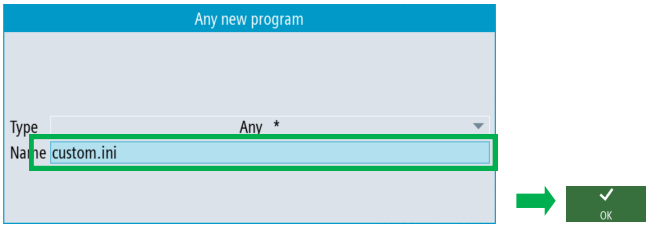
SINUMERIK data - CF Card/oem/sinumerik/hmi/cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing, enter the text below.

Example: **custom.ini** file.

The following example will show content of the “**custom.ini** ” file for the “OEM custom Area” of the controller.

```
; Text on the caption bar of the form in the start screen
[Header]
Text=Custom Test Page

; Picture shown on the form in the start screen
[Picture]
Picture=custom1.png
```

The example defines the name of the “Mask” to be loaded and the name of the file with the screen content for the “CUSTOM” area.

```
Text=Custom Test Page
```

This is the text that will be shown in the “Caption bar” at the top left hand side of the screen.

```
Picture=custom1.png
```

Is the name of the picture that will be placed in the cntre of the screen.



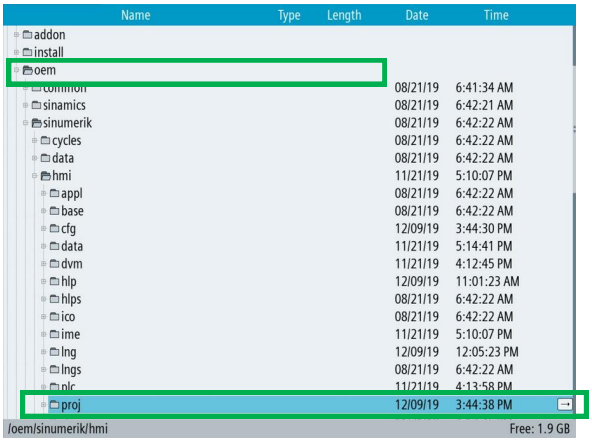
Creating the “custom.com” file on the controller

This file activates the content of the custom screens as stated in the **custom.ini** file. The file has to be located in one of the following paths/directories:-

SINUMERIK data - CF Card/oem/sinumerik/hmi/proj

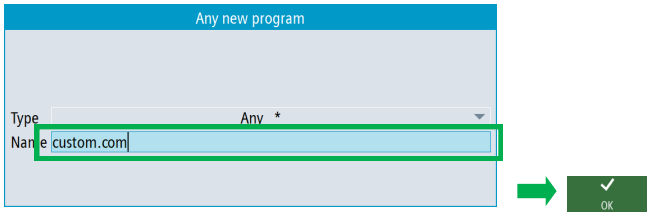
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/proj

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing, enter the text below.

Example: **custom.com** file.

The following example will show content of the “**custom.com** ” file for the “OEM custom Area” of the controller.

Notes:

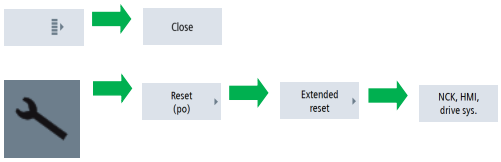
```
//s(start)

hs2=("Screen%nOne")
press(hs2)
lm("mask1","custommasks1.com")
end_press

hs3=("Screen%nTwo")
press(hs3)
lm("mask2","custommasks2.com")
end_press

//end
```

To activate the files, perform:



Storage path for the image files:

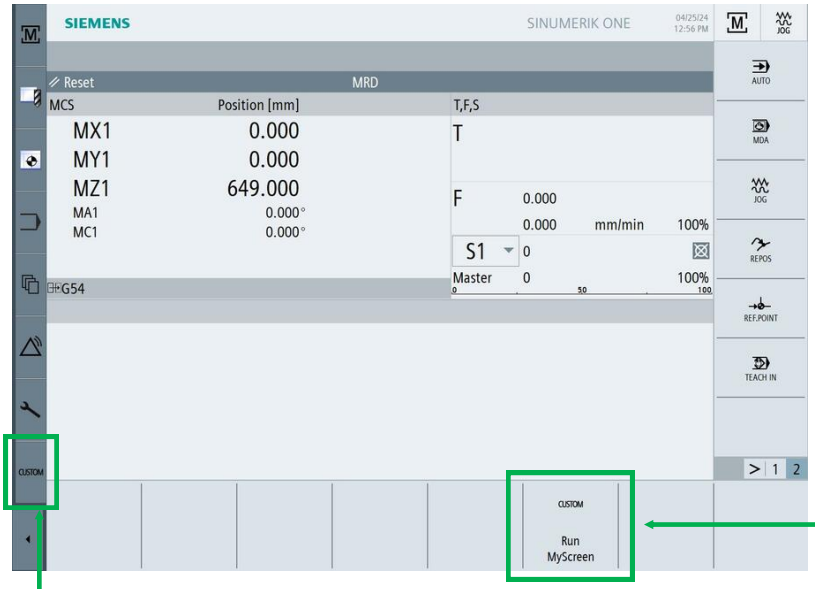
SINUMERIK data - CF Card/oem/sinumerik/hmi/ico/ico1024

IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/ico/ico1024

To see the result of the example the “CUSTOM” Area softkey has to be located on the control.



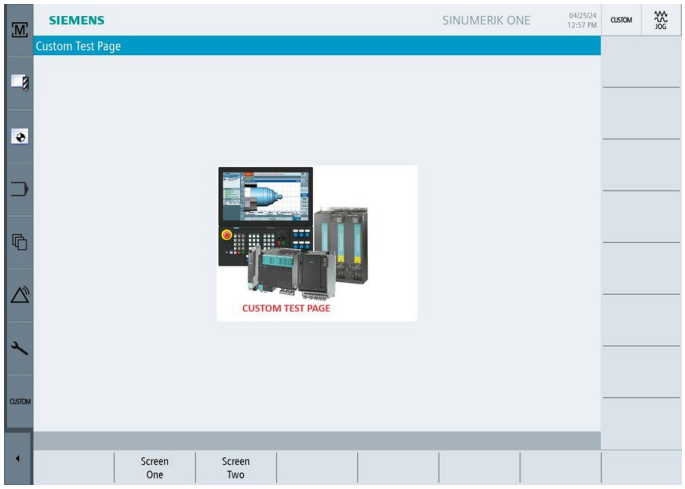
The 14th horizontal soft-key contains the text specified in the “**slamconfig.ini** ” file: **“SoftkeyPosition = 14".**



The new “CUSTOM” softkey MAIN area

The new “CUSTOM” softkey placed on the sidescreen.

Notes:



Two screens are now going to be created for the softkeys:

Screen One

Screen Two

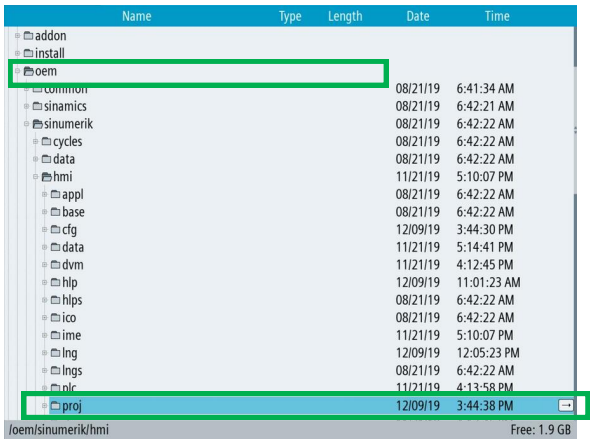
Creating the "custom.com" file on the controller

This file activates the content of the custom screens as stated in the **custom.ini** file. The file has to be located in one of the following paths/directories:-

SINUMERIK data - CF Card/oem/sinumerik/hmi/proj

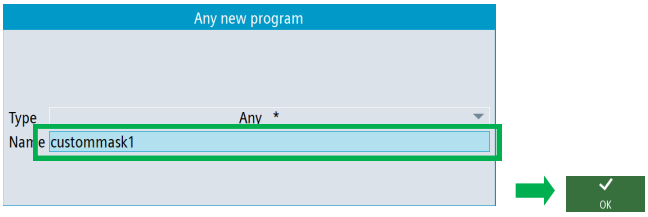
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/proj

Use the cursor keys to locate and open this directory.



To create the file select the "New" soft-key and ensure that the "Any" soft-key is selected.





The newly created file will be automatically opened in preparation for editing, enter the text below.

Example: **custommask1.com** file.

The following example will show content of the "**custommask1.com** " file for the "OEM custom Area" of the controller.

```
;s(menu1)

;m(mask1/"Custom Test Screen
One"/"picture1.png"///181,110)


hs2=("Screen%nOne", se3)
hs3=("Screen%nTwo")


press(hs3)
lm("mask2","custommasks2.com")
end_press

;end

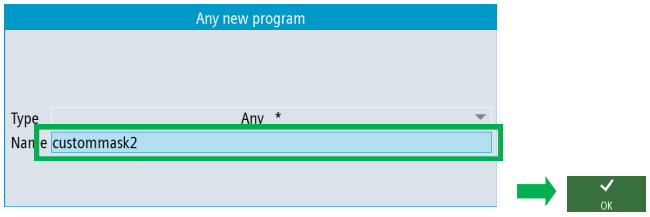
exit
```



Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing, enter the text below.

Example: **custommask1.com** file.

The following example will show content of the “**custommask1.com** ” file for the “OEM custom Area” of the controller.

```
//s(menu2)

//m(mask2/"Custom Test Screen Two - Workpiece Dimensions"/////////)
def var1=(R2///"","R0 X Length of component",,,"Length of component"///"$R[0]")
def var2=(R2///"","R1 Y Width of component",,,"Width of component"///"$R[1]")
def var3=(R3///"","R2 Z Height of component",,,"Height of component"///"$R[2]")
def var4=(I///"","R3 Spindle speed",,,"Main spindle speed"///"$R[3]")

hs2=("Screen%nOne")
hs3=("Screen%nTwo" , se3)


press(hs2)
lm("mask1","custommasks1.com")
end_press

press(vs6)
exit
end_press

//end
```



Notes:

To activate the files, perform:

Storage path for the image files:

SINUMERIK data - CF Card/oem/sinumerik/hmi/ico/ico1024

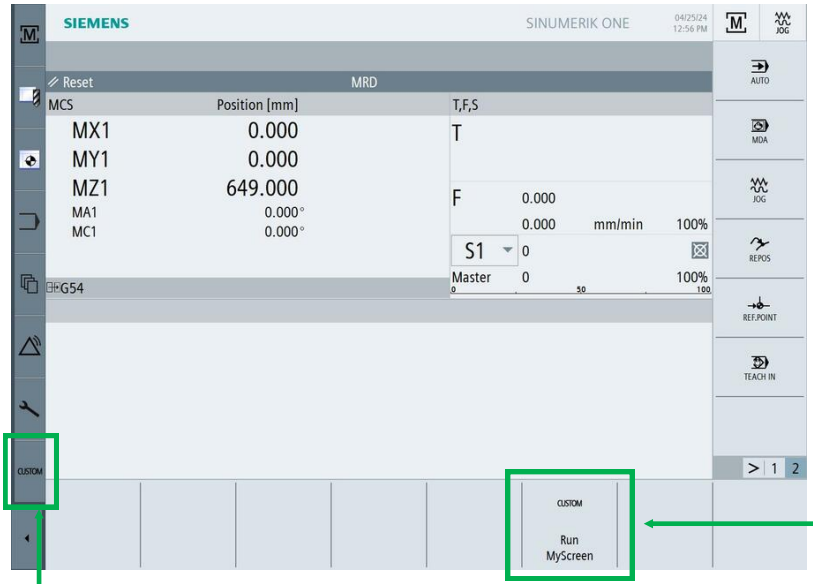
IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/ico/ico1024



To see the result of the example the “CUSTOM” Area softkey has to be located on the control.



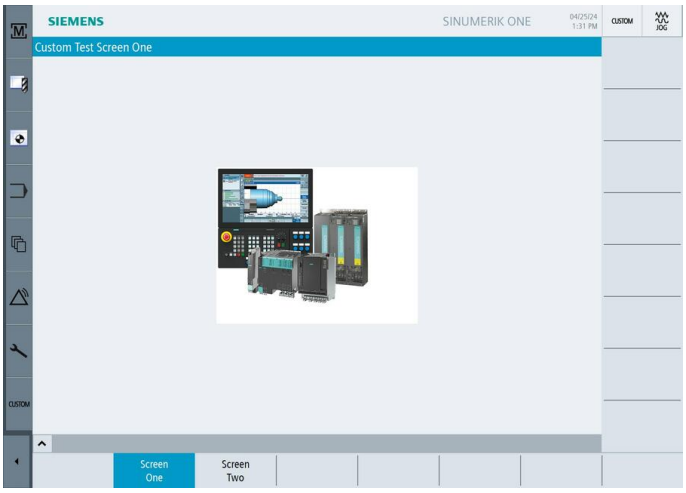
The 14th horizontal soft-key contains the text specified in the “**slamconfig.ini** ” file: “`SoftkeyPosition = 14`”.

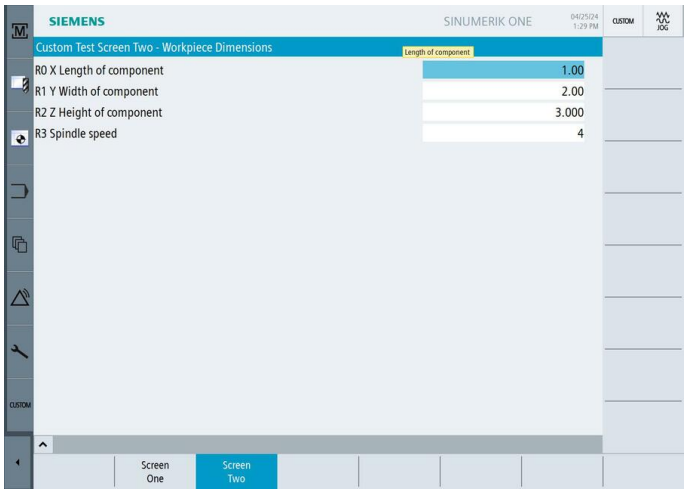


The new “CUSTOM” softkey MAIN area

The new “CUSTOM” softkey placed on the sidescreen.

Notes:





When the “CUSTOM” softkey is pressed the “Run MyScreen” code in the file **easyscreen.ini** and **custom.ini** are used.

Code contained within the **custom.com**, **custom-masks1.com and custommasks2.com** files are used to create soft keys and display texts and graphics etc for “Screen One” and “Screen Two”.

To navigate back to first custom screen press the “return” softkey.



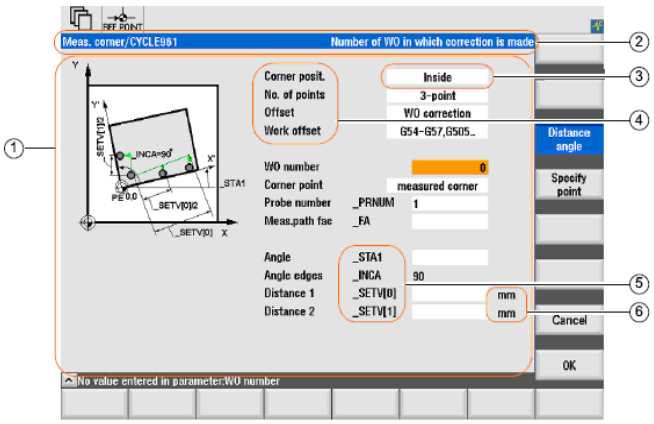
This “return” softkey is available all three screens the have been created.

Notes:

Basic Configuration Information

Configuration files are used to create the screens.
These configuration files are automatically interpreted by the controller.
The files are not available as standard and have to be created by the user using an ASCII editor (e.g. Notepad® or directly on the controller).

The “easyscreen.ini“ file is a master file which is required to specify which other files are to be used to create a user screen.
These files contain the information which determine what is displayed on a particular screen.
The screen is split into different areas:



1 Main dialog area
2 Title bar containing screen name (Left) and “Long text” description of the selected variable (Right)
3 Variable input/output field
4 Variable name “Short text”
5 Graphics related text
6 Unit text

Operating areas overview

To enable custom screens in other areas of the controller, the “easyscreen.ini” file has to be modified.
There are eight areas in total:-

Machine Area-Sub divided into:-
- Jog
- MDA
- Auto

Parameter Area

Program Edit Area

Program Manager Area

Diagnosis Area

Start-up Area

Each area should have a “com” file specified.

Notes:

Operating area	Permissible position for start soft-keys	File Name
Machine JOG	HSK6	ma_jog.com
Machine MDA	HSK6	ma_mda.com
Machine AUTO	HSK6	ma_auto.com
Parameter	HSK7	param.com
Program	HSK6	aeditor.com
Program-Manager	HSK2-8, 12-16, if not used for networked drives	progman.com
Diagnostics	HSK7	diag.com
Commissioning	HSK7	startup.com

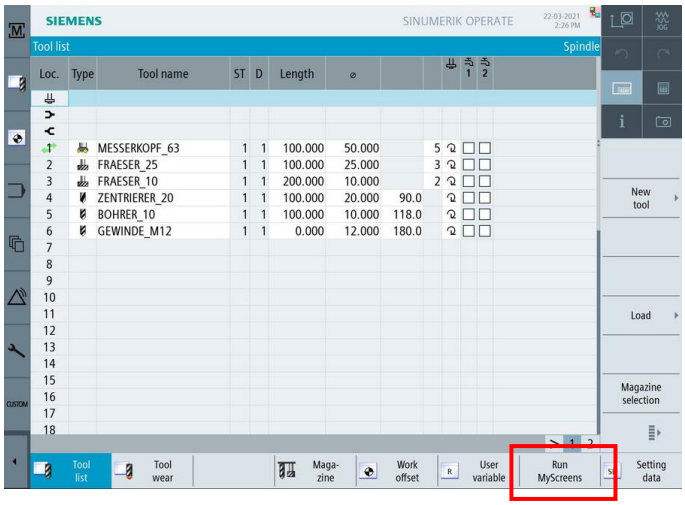
[STARTFILES]

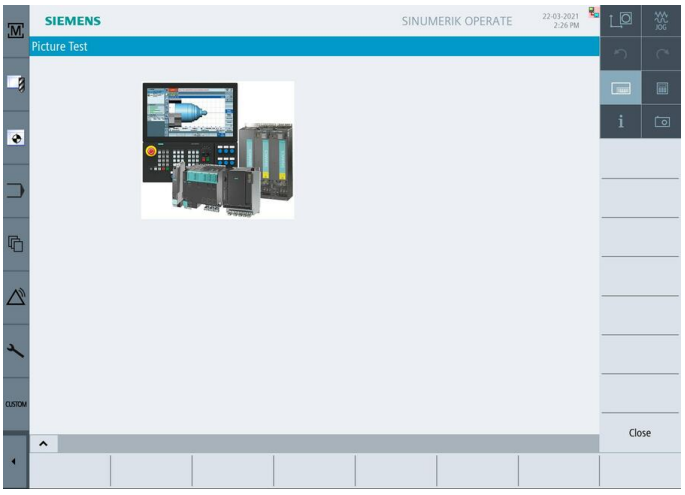
StartFile01 = area := AreaMachine, dialog := SIMachine, menu := SIMaJogMillMenuHU, startfile := ma_jog.com
StartFile02 = area := AreaMachine, dialog := SIMachine, menu := SIMaAutoMenuHU, startfile := ma_auto.com
StartFile03 = area := AreaMachine, dialog := SIMachine, menu := SIMaMdaMenuHU, startfile := ma_mda.com
StartFile04 = area := AreaParameter, dialog := SIParameter, startfile := param.com
StartFile05 = area := AreaProgramEdit, dialog := SIProgramEdit, startfile := aeditor.com
StartFile06 = area := AreaProgramManager, dialog := SIPmDialog, startfile := progman.com
StartFile07 = area := AreaDiagnosis, dialog :=SIDgDialog, startfile := diag.com
StartFile08 = area := AreaStartup, dialog :=SISuDialog, startfile := startup.com

Creating Run MyScreens OEM Files

The files required for the Run MyScreen function need to be created, this in turn well create new softkeys and screens where required in different areas as stated above.
This example will create a NEW softkey in the PARAMETER AREA and when pressed a new screen will appear.

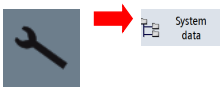
This picture below shows the objective of this section.

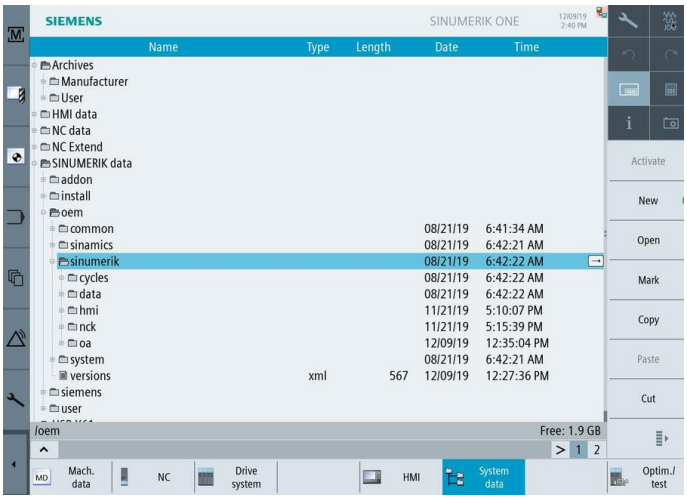




Notes:

The files required for the Run MyScreens function need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





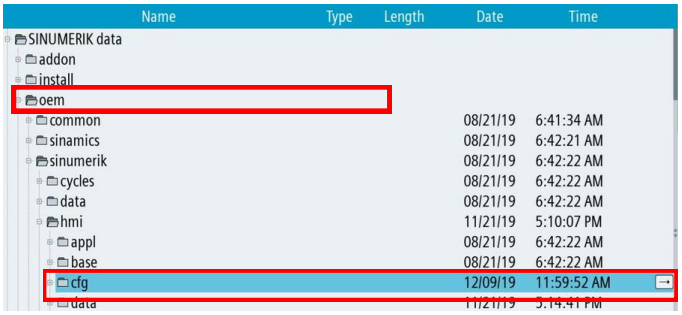
Creating the “easyscreen.ini” file on the controller

This file sets up the links to the files which are to be used to create the user screens, it also determine which operating areas these files are available in. The file has to be located in one of the following paths/directories:-

SINUMERIK data - CF Card/oem/sinumerik/hmi/cfg

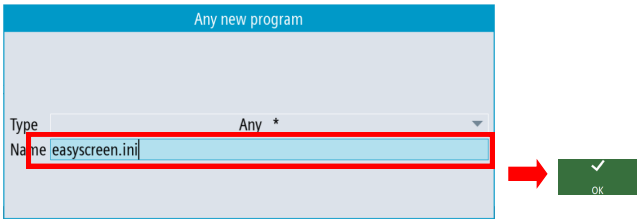
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.
The “Any” soft-key allows the file type to be determined by the user. In the dialogue box type in the name of the file and confirm with the “OK” soft-key.
The file name must be in lowercase due to the LINUX operating system.





The newly created file will be automatically opened in preparation for editing.

Example: easyscreen.ini file
The following example will show how the “easyscreen.ini” file is used to specify the “Parameter” area of the controller as the area from where the “Run MyScreens” are to be accessed.
(area := Area Parameter)

The “com” file, the content of which will determine what appears on the Run MyScreens screens, is also declared here.
(startfile := param.com)

The first step is to enter the header (**[STARTFILES]**)
The code for specifying the controller area and “com” file is then entered on a new line as shown below.

```
[STARTFILES]
StartFile04 = area := AreaParameter, dialog := SlParameter, startfile := param.com
```

Creating the User Files on an external PC

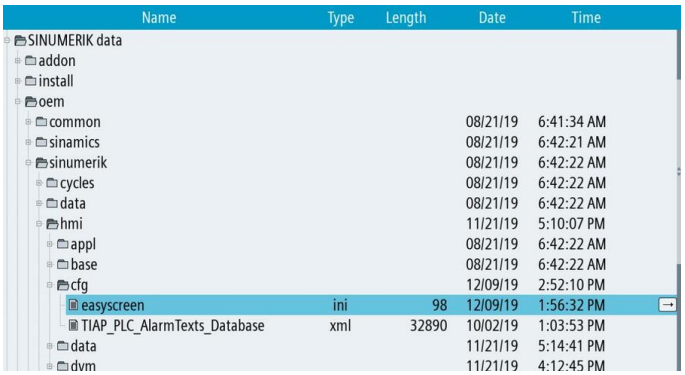
As mentioned previously it is possible to create the Run MyScreens files directly on the controller or offline using an ASCII editor (e.g. Notepad®).
These files have to be transferred to the controller and placed in the correct folders using the USB Interfaces or with Access MyMachine software.

Below: The **easyscreen.ini** file located in the **cfg** directory

Notes:

To activate the files, perform:







Creating the “param.com” file

The **param.com** file has been declared in the “easyscreen.ini” file and now needs to be created.

Code contained within the **param.com** file is used to create soft keys and display texts and graphics etc.

The file needs to be created and placed in one of the following folders:

SINUMERIK data - CF Card/oem/sinumerik/hmi/proj

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/proj

Example: param.com file

It is necessary to specify a soft-key to be used as the entry point to the Run MyScreens.
Further soft-keys can be assigned within the Run MyScreens if required.

```
//S(START)
HS7=("Run%nMyScreens")
PRESS(HS7)
   LM("mask1")
END_PRESS
//END

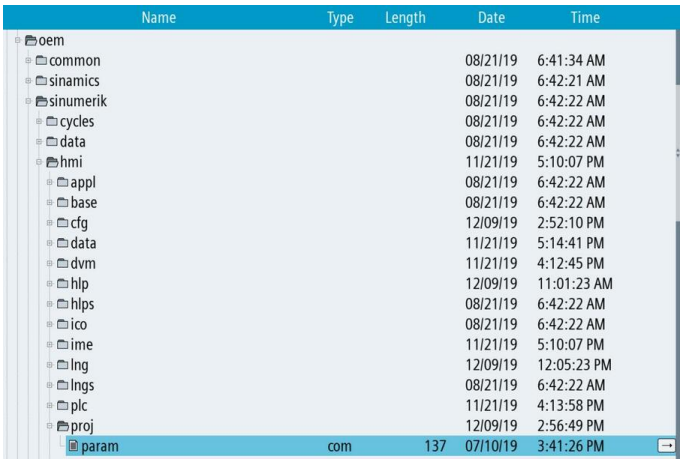
//M(mask1/"Picture
Test"/"picture1.png"///110,25)

//END
```

To activate the files, perform:



Below: The **param.com** file located in the **proj** directory.





Notes:

The example defines the 7th horizontal soft-key as the entry point, the soft-key will have the text “Run MyScreens” (on 2 lines).

```
//S(START)
HS7=("Run%nMyScreens")
```

When the soft-key is pressed a Mask (mask1) will be loaded:

```
PRESS(HS7)
   LM("mask1")
END_PRESS
//END
```

The Mask contains the code which determines the content of the Run MyScreens.
The first line of the mask (Mask header) contains the mask name, text and graphic information.

The following example shows **mask1** with the title of **Picture Test**. An image named picture1.png, will be displayed when **HS7** is pressed. The image will be positioned **110** pixels from the left and **25** pixels from the top of the screen.

```
//M(mask1/ "Picture Test" /
"picture1.png" / / / 110, 25)

//END
```

Storage path for the image files:

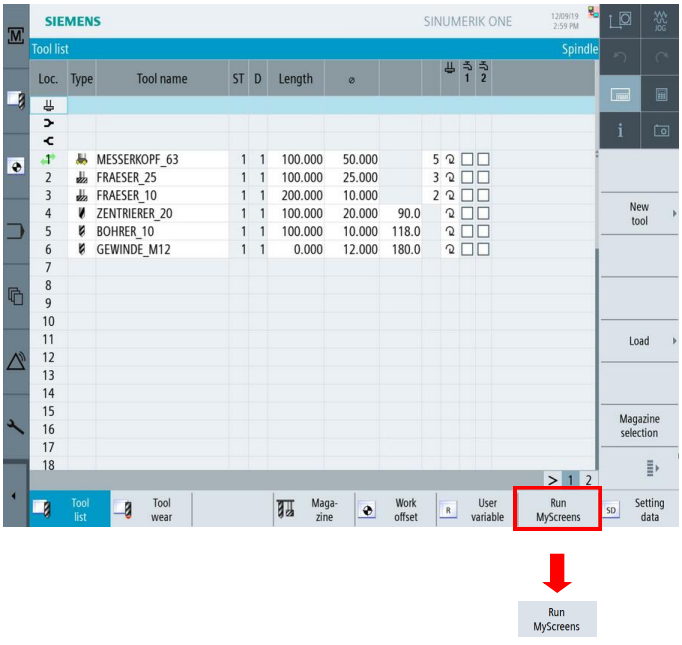
SINUMERIK data - CF Card/oem/sinumerik/hmi/ico/ico1024

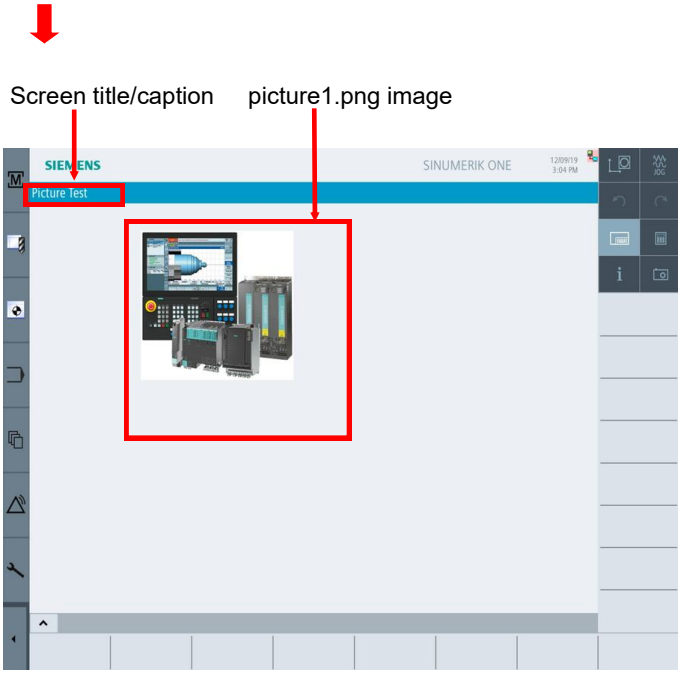
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/ico/ico1024

To see the results of the example we have to locate the “parameter” area of the control.



The 7th horizontal soft-key contains the text specified in the “param.com” file: `HS7=("Run%nMyScreens").`





Notes:

Mask header information:

The various parameters are separated by the “Forward slash” character (/).

//M(Identifier/header/Graphic/Dimension/Variable/Graphic position/Attributes)

Parameter descriptions:

Parameter	Description
Identifier	Label of dialogue/mask.
Header	Name of the screen.
Graphics	Name of image to be displayed.
Dimension	Position and size of the dialogue. (Distance from the left hand side, right hand side, width, height) Values in pixels and separated by commas
Variable	The cursor can be assigned to a variable. The first variable has an index of 1, the remaining variables are indexed in the order that they were configured.

Parameter	Description
Graphic Position	The position of the graphic in pixels. Distances Position are from the left and top of the dialogue screen and are separated by a comma.
Attributes	**CM** (Column Mode) Column alignment. **CM0:** Default, the column distribution is carried out separately for each line. **CM1:** The column distribution is of the ling with the most columns applies to all lines. **CB:** (Change Block) Response when the dialogue is opened. The CB attribute can also be used on individual variables, in this case the individual setting has priority over the CB setting in the Mask Header. **CB0:** All change blocks associated with the dialogue are processed when it is opened. **CB1:** Change blocks are only processed if the relevant value changes.

Adding Dialogue Elements:

Dialogue Elements refer to long and short texts, graphics texts, unit texts and I/O fields. Additional soft-keys can also be added.

The dialogue elements are added below the mask header.

In the following example (Shown in red text) a soft-key with the label “Variable Test” is added. When the soft-key is pressed the screen will display a list of four variables. The four variables are defined under a new mask (mask2-Workpiece Dimensions).

The parameters of the variables defined in mask2 are separated by the forward slash character (/).

A detailed description of the Dialogue Element parameters can be found in chapter ”Parameters and Variable Descriptions”.

```
//S(START)

HS7=("Run%nMyScreens")
PRESS(HS7)
  LM("mask1")
END_PRESS
//END

//M(mask1/"Picture Test"/"picture1.png"///110,25)
HS1=("Variable%nTest")
VS8=(softkey_nav_back

PRESS(HS1)
  LM("mask2")
END_PRESS

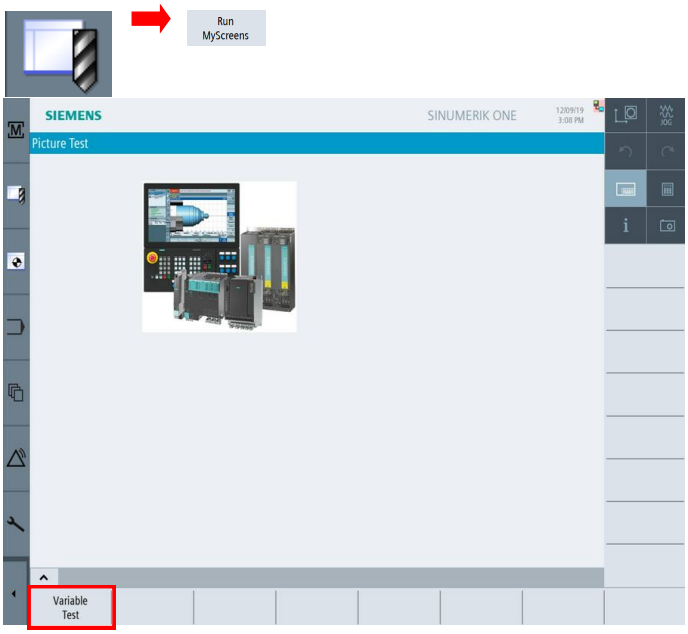
PRESS(VS8)
  EXIT
END_PRESS
//END

//M(mask2/"Workpiece Dimensions"/)
Def Var1=(R2///"Length of component","R0 X
Length",,,"Length of component"///"$R[0]")
Def Var2=(R2///"Width of component","R1 Y
Width",,,"Width of component"///"$R[1]")
Def Var3=(R3///"Height of component","R2 Z
Height",,,"Height of component"///"$R[2]")
Def Var4=(I///"Main spindle speed","R3 Spindle
speed",,,"Main spindle speed"///"$R[3]")

VS8=(softkey_nav_back)
PRESS(VS8)
  LM("mask1")
END_PRESS
//END
```

Notes:

The additional soft-key can now be seen.

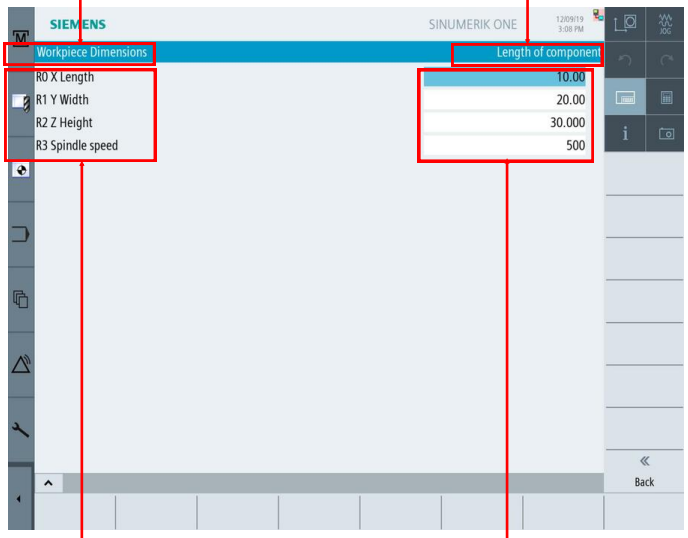


When the soft-key is pressed the defined variable will be displayed.



Screen title/name

Long text of the selected variable.



Short text

Input field

Values entered in the input field will be transferred to relevant variable.

Detailed Analysis:

```
//M(Mask2/"Workpiece Dimensions")
```

Name of Mask1:




```
HS1=("Variable %nTest")
```

Determines the text for horizontal soft-key 1 as “Variable Test”:

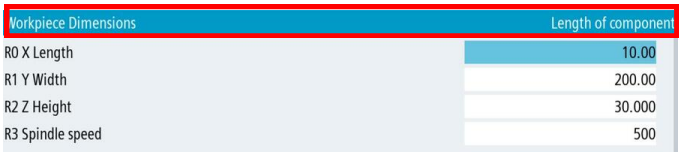


When the soft-key is pressed mask2 will be loaded.

```
PRESS(HS1)
  LM("mask2")
END_PRESS
```

```
Def Var1=(R2///"Length of component","R0 X Length"///"$R[0]")
```

R parameter 0 as a real number to 2 decimal places, sets the component length:




```
Def Var2=(R2///"Width of component","R1  Y Width"///"$R[1]")
```

R parameter 1 as a real number to 2 decimal places, sets the component width:



Notes:

```
Def Var3=(R3///"Height of component","R2  Z Height"///"$R[2]")
```

R parameter 2 as a real number to 3 decimal places, sets the component height:

Workpiece Dimensions	Height of component
R0 X Length	10.00
R1 Y Width	200.00
R2 Z Height	30.000
R3 Spindle speed	500

```
Def Var4=(I///"Main spindle speed","R3  Spindle speed"///"$R[3]")
```

R parameter 3 as an integer, sets the spindle speed:

Workpiece Dimensions	Main spindle speed
R0 X Length	10.00
R1 Y Width	200.00
R2 Z Height	30.000
R3 Spindle speed	500

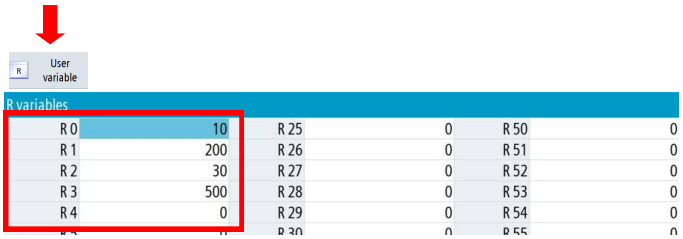
```
VS8=(sofkey_nav_back)
```

Determines a predetermined softkey of the sysem.



Checking the entered values

Values can be entered directly on the custom screen. These values are transferred directly to the variables.
They can be checked by viewing the variables on the standard screen by pressing the “User variable” soft-key.



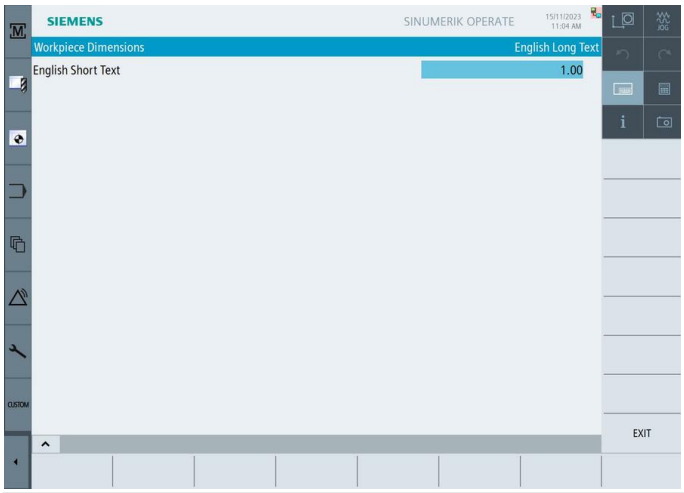
Creating a language dependant file

The files required for the Run MyScreen function need to be created, this in turn well create language depend screens, depending on the system language selected.

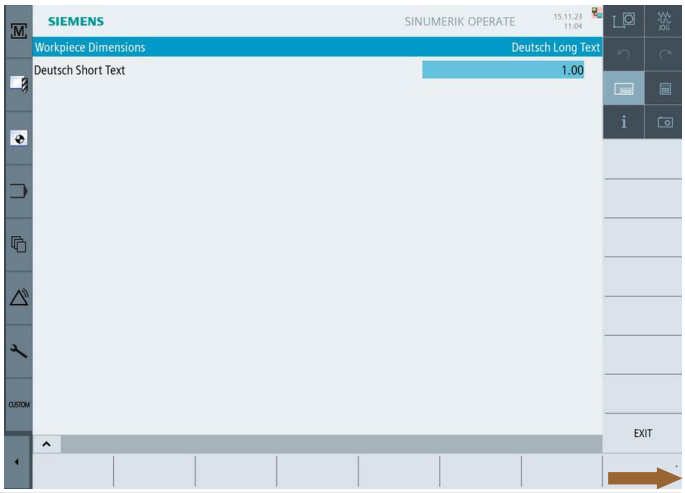
This example will create language dependant screens, for the newly created softkey, in the PARAMETER AREA and when pressed.

This picture below shows the objective of this section.

English:



German:



Notes:

It is possible to configure the Run MyScreens configuration files so that the Run MyScreens language matches that of the system.
In order to do this it is necessary to create one text file for each required language.
The text that is to be displayed is contained within the text files.
The Run MyScreens configuration files use identifiers to ensure that the correct text is displayed.

The files containing the text can be placed in one of the following locations:

SINUMERIK data - CF Card/oem/sinumerik/hmi/lng

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/lng

Example Text File:

```
85100        0    0      "English%nMyScreens"
85101        0    0      "English Mask1 Header"
85102        0    0      "English Mask2 Header"
85103        0    0      "English Long Text"
85104        0    0      "English Short Text"
```



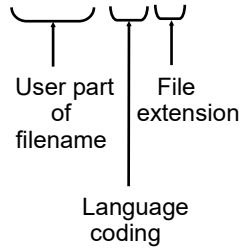
Allowable range of identifiers 85000 to 89899

Part of the file name can be assigned by the user but must also contain the correct 3 letter language coding. This is so that the system can display the contents of the correct file.

Filename Examples:

```
easyscr_eng.txt
easyscr_deu.txt
```

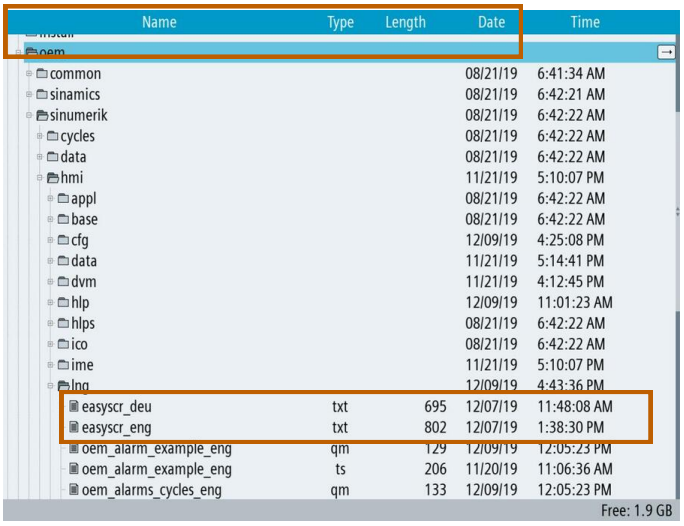
English filename example
German filename example



Example of a second text file for the German language

```
85100        0    0      "Deutsch%nMyScreens"
85101        0    0      "Deutsch Mask1 Header"
85102        0    0      "Deutsch Mask2 Header"
85103        0    0      "Deutsch Long Text"
85104        0    0      "Deutsch Short Text"
```

The files for English and German can be seen located in the OEM area.



The identifier is used in place of the text in the param.com file.
The identifier has to be prefixed by the **$** symbol.
The text associated with the identifier will be displayed.

Notes:

Example **param.com** file for language dependent text.

```
//S(START)
HS7=($85100)

PRESS(HS7)
  LM("mask1")
END_PRESS
//END

//M(mask1/$85101/"picture1.png"///110,25)
HS1=("Variable%nTest")
PRESS(HS1)
  LM("mask2")
END_PRESS
//END

//M(mask2/$85102/)

Def Var1=(R2///$85103, $85104///"$R[0]")

VS8=("EXIT")
PRESS(VS8)
  EXIT
END_PRESS

//END
```

The language file needs to be declared in the "**easyscreen.ini**" file. No language coding is required. Declarations of other text files may also be present.

```
[STARTFILES]

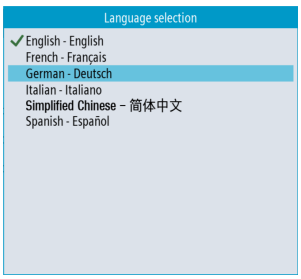
StartFile04 = area := AreaParameter, dia-
log := SlParameter, startfile := param.com

[LANGUAGEFILES]
LngFile01 = almc.txt
LngFile02 = alsc.txt
LngFile03 = aluc.txt
LngFile04 = easyscr.txt
```

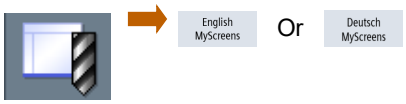
Setting the controller language:

The controller language is selected via the "Start-up" menu.





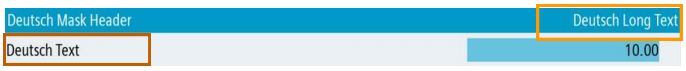
When using the examples shown earlier it can be seen that the text displays change with the selected language.



English long and short texts:



German long and short texts:

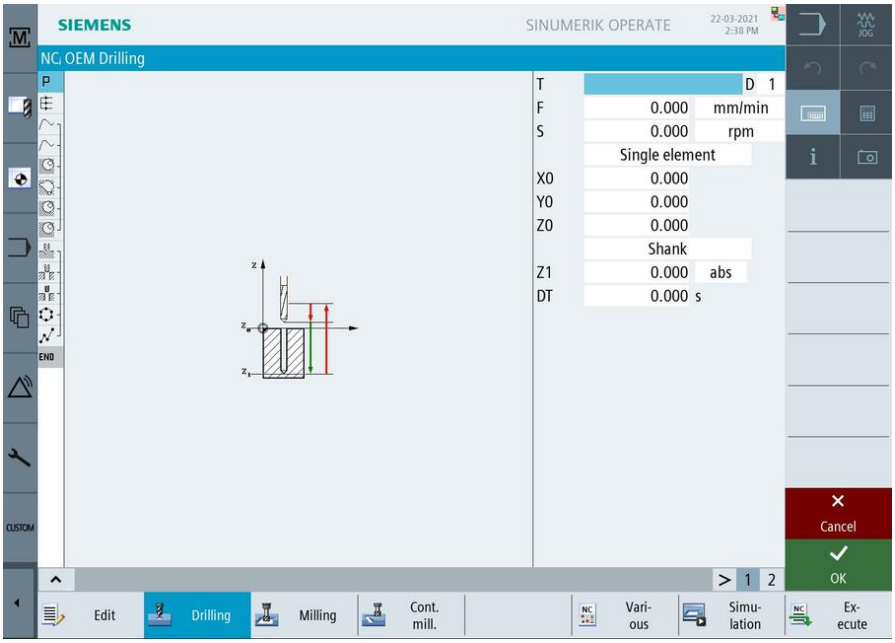


Notes:

Creating CYCLE screens OEM files

This example will create a CYCLE OEM screen.

This picture below shows the objective of this section.

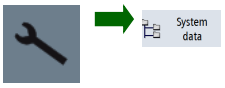


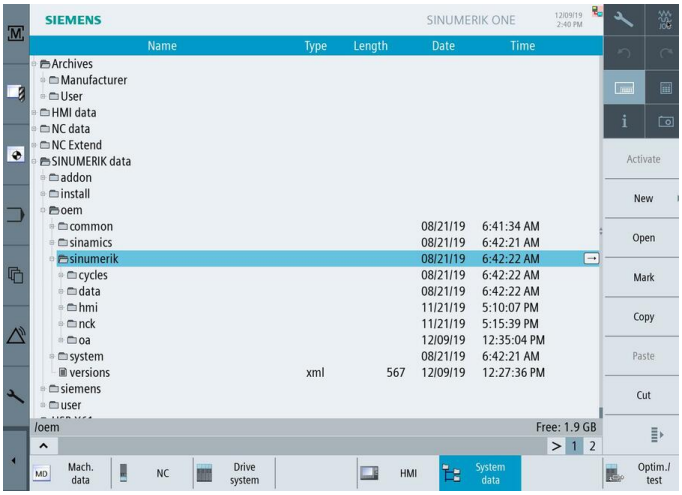
NOTE:
This example is based on files that are located in the following paths/directories:-

"HMI data\Templates\Examples\Easyscreen\Standard"

This includes the SPF (e_dr_o1.spf) required for this example

The files required for the Run MyScreens function need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the "System data" area.





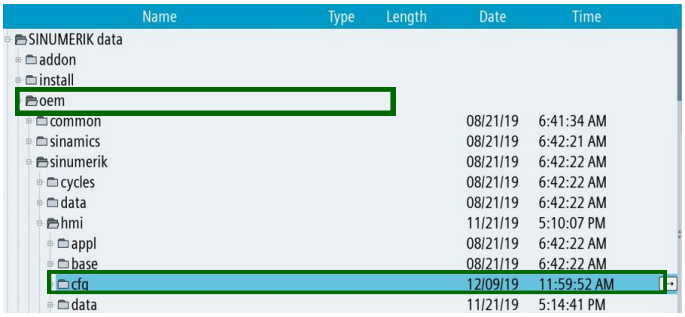
Creating the "easyscreen.ini" file on the controller

This file sets up the links to the files which are to be used to create the user screens, it also determine which operating areas these files are available in. The file has to be located in one of the following paths/directories:-

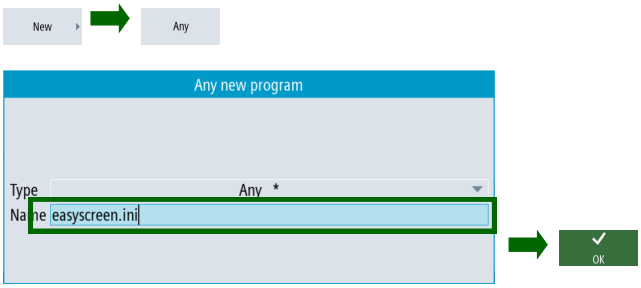
SINUMERIK data - CF Card/oem/sinumerik/hmi/cfg
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.

Notes:



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.
The “Any” soft-key allows the file type to be determined by the user. In the dialogue box type in the name of the file and confirm with the “OK” soft-key.
The file name must be in lowercase due to the LINUX operating system.



The newly created file will be automatically opened in preparation for editing.

Example: easyscreen.ini file
The following example will show how the “easyscreen.ini” file is used to specify the “Parameter” area of the controller as the area from where the “Run MyScreens” are to be accessed.
(area := Area AreaProgramEdit)

The “com” file, the content of which will determine what appears on the Run MyScreens screens, is also declared here.
(startfile := e_dr_o1.com)

The first step is to enter the header (**[STARTFILES]**)
The code for specifying the controller area and “com” file is then entered on a new line as shown next:

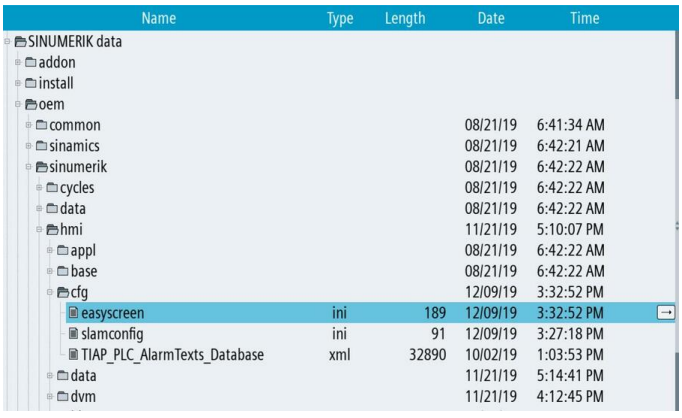
```
[STARTFILES]

StartFile04 = area := AreaParameter, dialog := SlParameter, startfile := param.com

StartFile05 = area := Custom, dialog := SlEsCustomDialog, startfile := mycustommasks.com

StartFile06 = area := AreaProgramEdit, dialog  := SlProgramEdit, screen:= SlStepShopMillDrill-
Screen, startfile := e_dr_o1.com
```

Below: The **easyscreen.ini** file located in the **cfg** directory



Notes:

Creating the “e_dr_o1.com” file

The **e_dr_o1.com** file has been declared in the “easyscreen.ini” file and now needs to be created.

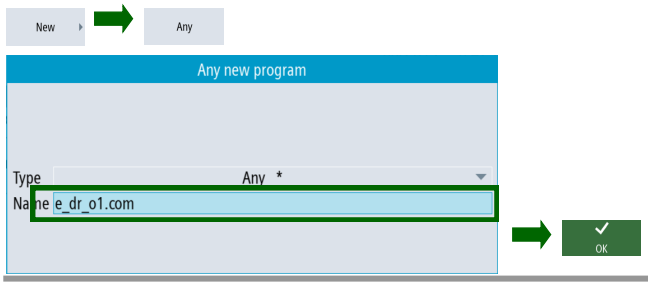
Code contained within the **param.com** file is used to create soft keys and display texts and graphics etc.

The file needs to be created and placed in one of the following folders:

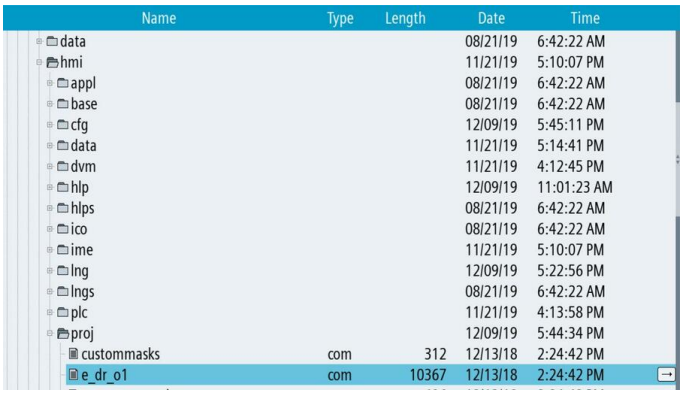
SINUMERIK data - CF Card/oem/sinumerik/hmi/proj

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/proj

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.
The “Any” soft-key allows the file type to be determined by the user. In the dialogue box type in the name of the file and con-



Below: The **e_dr_o1.com** file located in the **proj** directory.



Example: e_dr_o1.com file

It is necessary to specify a soft-key to be used as the entry point to the Run MyScreens.
Further soft-keys can be assigned within the Run MyScreens if required.

The example defines the 6th vertical in the “Program Editor for ShopMill, Drilling section” soft-key as the entry point, the soft-key will have the text “OEM Drilling” (on 2 lines).

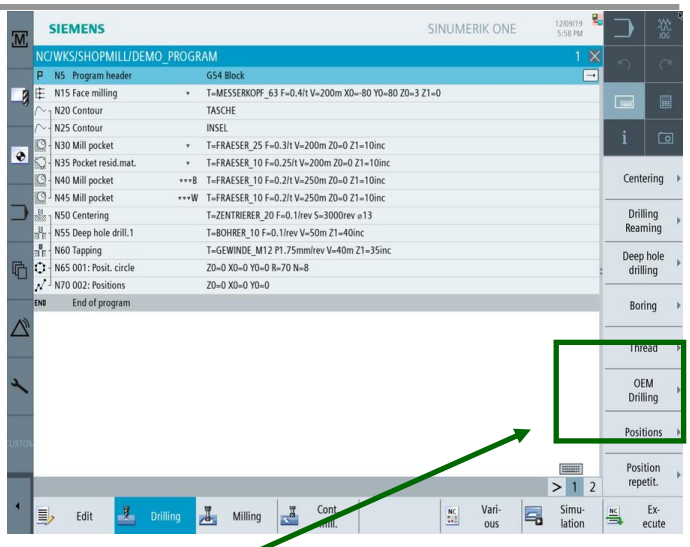
```
//S(START)
  VS6=(["OEM%nDrilling"],TP0,SE1,PA1)
```

When the soft-key is pressed a Mask (MASK_E_DR_O1_MAIN) will be loaded:

```
  PRESS(VS6)
    LM("MASK_E_DR_O1_MAIN")
  END_PRESS
//END
```

The Mask contains the code which determines the content of the Run MyScreens.
The first line of the mask (Mask header) contains the mask name, text and graphic information.

The following example shows **MASK_E_DR_O1_MAIN** with the title of **OEM Drilling**. Images are used to help with PARAMETER inputs that are required for the OEM DRILLING cycle, are displayed when **VS6** softkey is pressed.





The 6th vertical soft-key contains the text specified in the “**e_dr_o1.com**” file:

```
VS6=(["OEM%nDrilling"],TP0,SE1,PA1)
```

Notes:

```
//S(START)
  VS6=(["OEM%nDrilling"],TP0,SE1,PA1)
  PRESS(VS6)
    LM("MASK_E_DR_01_MAIN")
  END_PRESS
//END

//M(MASK_E_DR_01_MAIN/"OEM Drilling"/"empty.png"///50,60/XG1,FA2,TA7//)

 DEF VarTF=(S20////wr0///0,0,24/44,0,24)
 DEF VarPOS=(IDD////wr0///0,0,24/44,0,24)
 ; name for the program step
 DEF VarOemName=(S17//"OEM Drilling"//wr0///0,0,8/0,0,208/10,5)
 ;  --  Technology block tool, feed, spindle --
 DEF VarTool=(S20///,"T",,,"Tool name"//"e_dr_z1.png"//2,2,24/41,2,129),
     VarD=(I/* 1,2//,"D",,,"Cut edge number"//"e_dr_z1.png"//171,2,10/181,2,25)
 ; feed + unit toggle
 DEF VarF=(R//0.0/,"F",,,"Feed"//"e_dr_z1.png"//2,,24/41,,85///"F_Unit"),
     F_Unit=(I/*3="mm/rev", 1="mm/min"/1///"e_dr_z1.png"//,,0/131,,75)
 ; speed + unit toggle
 DEF VarS=(R//0.0/,"S",,,"Spindle speed"//"e_dr_z1.png"//2,,24/41,,85///"S_Unit"),
     S_Unit=(I/*1="rpm", 2="m/min"/1///"e_dr_z1.png"//,,0/131,,75)
 ;  -- Actual step data
 ; individual element/group element
 DEF VarModeDispl=(I/* 4="Single element", 1="Group element"//,,,,"Machining posi-
tion"//"e_dr_x0.png"//,,0/41,,136)
 DEF VarX0=(R3//0.000/,"X0",,,"Reference point X0"//"e_dr_x0.png"//2,,24/41,,85)
 DEF VarY0=(R3//0.000/,"Y0",,,"Reference point Y0"//"e_dr_x0.png"//2,,24/41,,85)
 DEF VarZ0=(R3//0.000/,"Z0",,,"Reference point Z0"//"e_dr_z0.png"//2,,24/41,,85)
```

```
 ; shaft/tip (inc/abs)
 DEF VarDrillModeDispl=(I/* 0="Shank",1="Tip"//,,,,"Drilling depth%nrelated to
tip"//"e_dr_sp.png"//,,0/41,,136)
 ; depth + unit toggle
 DEF VarZ1=(R3//0.000/,"Z1",,,"Drilling depth"//"e_dr_z1.png"//2,,24/41,,85///"Z_Unit"),
     Z_Unit=(I/*0="abs", 1="inc"/0///"e_dr_z1.png"//,,0/131,,42)
 DEF VarDT=(R3//0.000/,"DT",,"s","Dwell time"//"e_dr_z1.png"//2,,24/41,,85,,5,60)
 ;  -- Auxiliary variable
 DEF ZZ1A=(I//0//wr0///2,,0/24,,44)  ; shaft/tip Z1 abs/inc
 ;  --  The following variables are only required for recompilation
 ;  --  They are only assigned when executing the program.
 ;  --  As a consequence there are no input fields.
 DEF VarBO
 DEF VarIDD
 ;  -- Vertical softkey bar
 VS1=("")
 VS2=("")
 VS3=("")
 VS4=("")
 VS5=("")
 VS6=("")
 VS7=(["Cancel","\\sk_cancel.png"])        ; "Cancel"
 VS8=(["OK","\\sk_ok.png"])            ; "accept"
 ;  --  LOAD - Block
 LOAD
   ; cutting speed
   IF S_Unit==2
     VarS.tt="Constant%ncutting speed"
     VarS.st="V"
   ENDIF
```



Notes:

```
  ; initializing the fields
  CALL("StateChange")
      ;  --  Z1 abs/ink  --
  IF ZZ1A == 1 OR ZZ1A == 91            ;shaft/tip depth ink?
    Z_Unit = 1                          ;ink
    VarZ1.tt="Drilling depth%nrelated to Z0"
  ELSE
    Z_Unit = 0                          ;abs
    VarZ1.tt="Drilling depth"
  ENDIF

  IF ZZ1A == 0 OR ZZ1A == 1             ;shaft
    VarDrillModeDispl.tt="Drilling depth%nrelated to shaft"         ;depth referred to shaft
    VarDrillModeDispl=0
  ELSE
    IF ZZ1A == 90 OR ZZ1A == 91         ;tip
      VarDrillModeDispl.tt = "Drilling depth%nrelated to tip"     ;depth referred to tip
      VarDrillModeDispl = 1
    ENDIF
  ENDIF

END_LOAD
```

```
;  ------ Switchover speed/cuting speed --------
CHANGE(S_Unit)
  IF S_Unit==2
    VarS.tt = "Constant%ncutting speed"
    VarS.st="V"
  ELSE
    IF S_Unit==1
      VarS.tt="Spindle speed"
      VarS.st="S"
    ENDIF
  ENDIF
END_CHANGE
;  ------ Switchover individual/group element --------
CHANGE(VarModeDispl)
    ;group element/individual element
    CALL("StateChange")
END_CHANGE
;  ------ Switchover shaft/tip -----
CHANGE(VarDrillModeDispl)
  IF VarDrillModeDispl == 0         ; shaft
    VarDrillModeDispl.tt = "Drilling depth%nrelated to shaft"
    IF Z_Unit == 1                  ; Z1 inc
      ZZ1A=1
    ELSE
      IF Z_Unit == 0                ; Z1 abs
        ZZ1A=0
      ENDIF
    ENDIF
  ELSE
```

Notes:

```
    IF VarDrillModeDispl == 1        ; tip
      VarDrillModeDispl.tt = "Drilling depth%nrelated to tip"
      IF Z_Unit == 1                 ; Z1 inc
        ZZ1A=91
      ELSE
        IF Z_Unit == 0               ; Z1 abs
          ZZ1A=90
        ENDIF
      ENDIF
    ENDIF
  ENDIF
END_CHANGE
;  ------  Switchover Z1 abs/inc     -----
CHANGE(Z_Unit)
  IF Z_Unit == 0                     ; abs
    VarZ1.tt = "Drilling depth"                ; drilling depth
    IF VarDrillModeDispl == 0        ; shaft
      ZZ1A=0
    ELSE
      IF VarDrillModeDispl == 1      ; tip
        ZZ1A=90
      ENDIF
    ENDIF
  ELSE
    IF Z_Unit == 1                   ; inc
      VarZ1.tt = "Drilling depth%nrelated to Z0"             ; depth refered to Z0
      IF VarDrillModeDispl == 0      ; shaft
        ZZ1A=1
      ELSE IF
```

```
        IF VarDrillModeDispl == 1    ; tip
          ZZ1A=91
        ENDIF
      ENDIF
    ENDIF
  ENDIF
END_CHANGE
;  -- Initialization or state change
SUB(StateChange)
   ;  -- Group element/individual element --
  IF VarModeDispl == 1
    VarX0.wr = 4
    VarY0.wr = 4
    VarZ0.wr = 4
  ELSE
    ; individual element-> X0,Y0,Z0 - initialize fields
    VarX0.wr = 2
    VarY0.wr = 2
    VarZ0.wr = 2
    VarX0.st = "X0"                          ;X0
    VarY0.st = "Y0"                          ;Y0
    VarZ0.st = "Z0"                          ;Z0
  ENDIF

END_SUB
;  -- Checking the dwell time
CHANGE(VarDT)
  IF 0 > VarDT
    VS8.se=2
    DLGL("Dwell time too small")
```

Notes:

```
    VarDT.bc = "#ffbea5"          ; background color red
    VarDT.vld = FALSE
  ELSE
    VS8.se=1
    VarDT.bc = 10         ; background color white
  ENDIF
END_CHANGE
;  -- Exit the screen form with cursor left and generate code
PRESS(SL)

  ; For these fields "Cursor left" not execute any accept.
  IF (FOC <> "VarD") AND (FOC <> "F_Unit") AND (FOC <> "S_Unit") AND (FOC <> "Z_Unit")
    ; Check for validity
    IF VarDT.vld == TRUE
      IF VarF.vld == TRUE
        IF VarS.vld == TRUE
          IF VarD.vld == TRUE
            ;  -- generate code and exit --
            GC("OUT_E_DR_O1")
            EXIT
          ELSE
            DLGL("Tool nose value incorrect.")
          ENDIF
        ELSE
          DLGL("Spindle value incorrect.")
        ENDIF
      ELSE
        DLGL("Feed incorrect.")
      ENDIF
```

```
    ELSE
      DLGL("Dwell time too small")
    ENDIF
  ENDIF

END_PRESS  ;  <--  cursor left
;  --  PRESS(VS7) : exit screen form without generating code
PRESS(VS7)
  EXIT
END_PRESS  ;  <--  VS7
;  --  PRESS(VS8) : generate code and exit screen form
PRESS(VS8)
  ;  -- interrogate several parameters --
  IF VarDT.vld == TRUE
    IF VarF.vld == TRUE
      IF VarS.vld == TRUE
        IF VarD.vld == TRUE
          ;  -- generate code and exit --
          GC("OUT_E_DR_O1")
          EXIT
        ELSE
          DLGL("Tool nose value incorrect.")
        ENDIF
      ELSE
        DLGL("Spindle value incorrect.")
      ENDIF
    ELSE
      DLGL("Feed incorrect.")
    ENDIF
```



Notes:

```
ELSE
    DLGL("Dwell time too small")
   ENDIF

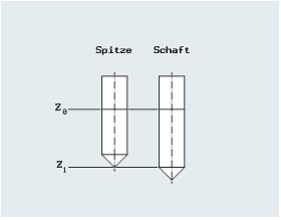
 END_PRESS  ;  <--  VS8
 ;  --  OUTPUT - Block
 OUTPUT(OUT_E_DR_O1)
    "E_DR_O1(" VarModeDispl "," VarBO "," VarIDD ","""" VarTool """,""" VarTF """," VarD ","
VarF "," F_Unit "," VarS "," S_Unit "," VarPOS "," VarZ1 "," ZZ1A "," VarDT "," VarX0 "," VarY0
"," VarZ0 ")"
 END_OUTPUT

//END  ;  <--  M (MASK_E_DR_O1_MAIN)
```

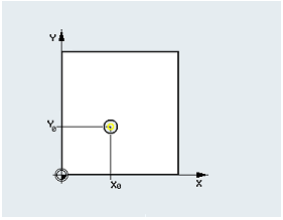
Storing of icons

Icon files are stored under one of the following paths:
/oem/sinumerik/hmi/ico or
/user/sinumerik/hmi/ico
Depending on the screen resolution, under the following sub-directories:
/ico1024 for HD Ready
/ico1280
/ico1600
/ico640
/ico800

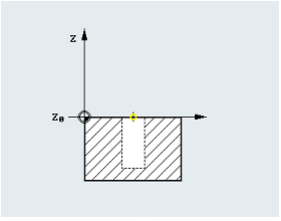
In the example, four icons are used:
e_dr_sp.png
e_dr_x0.png
e_dr_z0.png
e_dr_z1.png



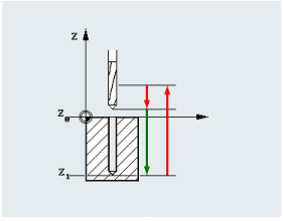
e_dr_sp.png



e_dr_x0.png



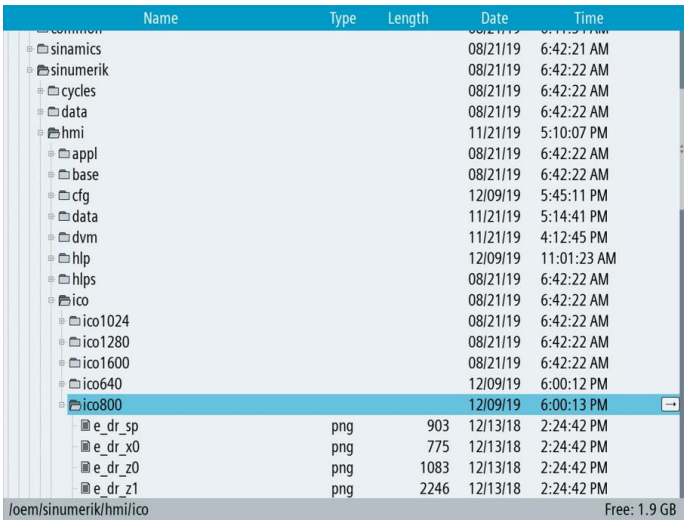
e_dr_z0.png



e_dr_z1.png

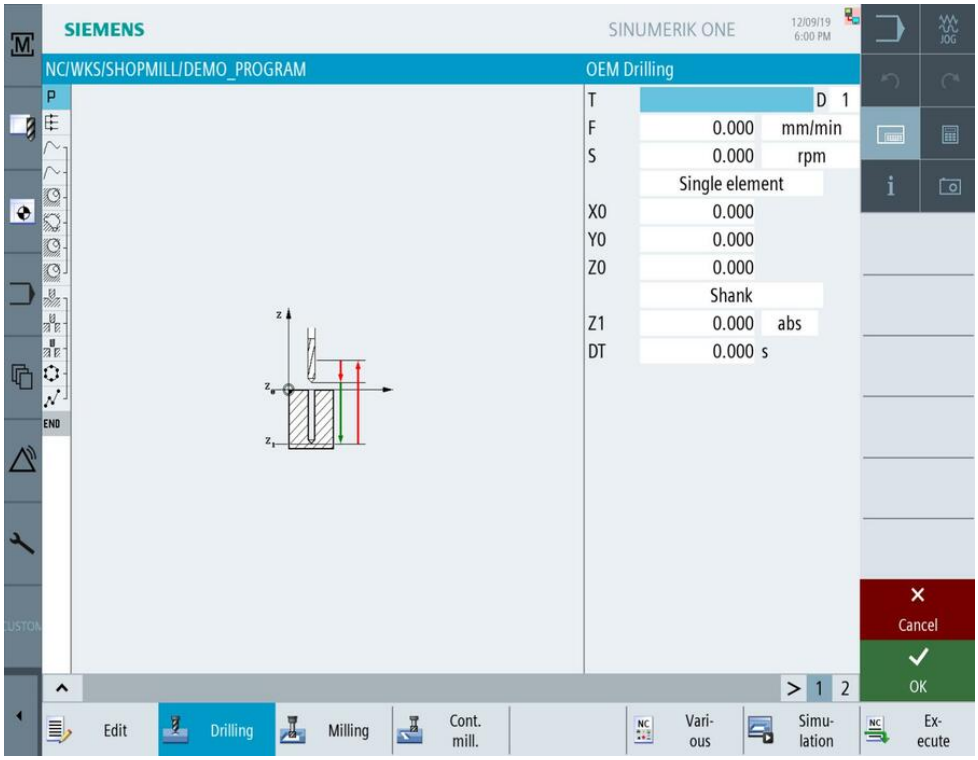
Notes:

The four icons placed in the ico640 directory



To activate the files, perform a power off / on.

The completed cycle mask



Notes:

Parameters and Variables Descriptions

When a variable is assigned certain properties can be set.
These properties are contained rounded brackets, each property is separated by a forward slash (/)
If a property is not used, the slash still needs to be present.

Basic description:

DEF Var1=(**1**/**2**/**3**/**4**/**5**/**6**/**7**/**8**/**9**/**10**)

1. **Variable type**
2. **Limit values, toggle field or table identifier**
3. **Default**
4. **Texts**
5. **Attributes**
6. **Help display**
7. **System or user variable**
8. **Position of short text**
9. **Position of I/O field (Left, top, width, height)**
10. **Colours**

Detailed description:

1. Variable type

R[x]	Real number, "x" indicates the number of decimal places displayed
I	Integer.
S[x]	String,"x" indicates the string length.
C	Character (Individual character)
B	Boolean number
V	Variant type

2a. Limit values-
A minimum and a maximum value can be specified. The values are separated by a comma, the minimum value is specified first.
Values can be in I, C and R decimal formats or as characters in the format "A" , "F"
The **limit value** also works in conjunction with the **limit** attribute, see the **attribute** property for more information.

2b. Toggle field-
Creates a list of predefined entries in the I/O field. The list is initiated by the "*****" character.
The entries can be assigned a value.
The **Limit value** is interpreted as a list for the **Toggle field.**
If only a "*****" is used, a variable Toggle field is generated.

2c. Table Identifiers-
Identifiers can be assigned to tables containing NCK/PLC values of the same type. The "**%**" character is used to differentiate an **identifier** from a **toggle field** or **limit value.**
A comma is required after the identifier, the file name containing the table information is then typed.

3. Default setting-
The value in the assigned variable can take on various states. This depends on whether a default value, system or user variable (Or both) has been assigned to the variable field (I/O or toggle field)
Toggling is not possible until a valid value is assigned to the variable.

4. Texts-
Text possibilities are-

- Long text = text displayed when each element is highlighted.
- Short text = name of the element
- Graphics text = text relating to the graphics, if used.
- Unit text = unit of the variable e.g. mm, rpm etc

The texts types are separated by commas and need enclosing in speech marks:- **/"Long", "Short", "Graphics", "Unit"/**

5. Attributes-
The following properties can be influenced by the attributes

- Input mode
- Access level
- Alignment of short text
- Font size
- Handling limit values
- Response when screen is opened (Change block)

The attributes are separated by commas, and are **not** valid for **toggle fields.**

Input mode:-

Attribute	Effect
wr0	I/O field invisible, short text visible.
wr1	Read only, no inputting of data, short text visible. (Default setting)
wr2	Read & write possible, long and short text visible.
wr3	As with wr1 but with I/O field visible, long and short text visible.
wr4	All elements invisible.
wr5	As wr2 except entries are immediately active. with wr2 the entry becomes active exiting the field or pressing enter/input/return.

Notes:

Access level:-

Attribute	Level	Password/Key position
ac0	0	Not released
ac1	1	SUNRISE
ac2	2	EVENING
ac3	3	CUSTOMER
ac4	4	Key switch pos 3
ac5	5	Key switch pos 2
ac6	6	Key switch pos 1
ac7	7	Key switch pos 0

Alignment of short text:-

Attribute	Effect
al0	Left justified (Default)
al1	Right justified
al2	Centred

Font size:

Attribute	Effect
fs1	8pt. font size (Default)
fs2	16pt font size

The clearance between lines is defined. The default font size allows 16 lines. The graphics and unit text can only be configured in the default font size.

Limits:
Checking whether the value of the variable is within the limits specified by the **limit values.**

Attribute	Effect
li0	No check.
li1	Check with respect to minimum
li2	Check with respect to maximum.
li3	Check with respect to minimum and maximum.

Behaviour when opening (Change block):

Attribute	Effect
cb0	The change block defined for this variable is processed when the screen opens (Default).
cb1	The change block defined for this variable is processed if there is a change in the value of the variables.

The “cb” attributes, specified in the variables definition, take precedence over the globally specified attributes in the “screen form” definition.

6. Help Display-

The name of the HTML file containing the help text needs to be typed in speech marks. The display appears automatically if the cursor is positioned on the variable.

7. System or user variables-

The variable to be viewed is entered here, these have to be contained within speech marks. The variable must be addressed in the correct way.

Addressing Variables:

Machine Data	Syntax
Global machine data	$MN_...
Axis specific machine data	$MA_...
Channel specific machine data	$MC_...

Setting Data	Syntax
Global setting data	$SN_...
Axis specific setting data	$SA_...
Channel specific setting data	$SC_...

PLC Data	Syntax
Byte y of bit z of data block x	DBx.DBXy.z
Byte y of data block x	DBx.DBBy
Word y of data block x	DBx.DBWy
Double word y v. of data block x	DBx.DBDy
String y with length z from data block x	DBx.DBSy.z
Real y of data block x	DBx.DBRy
Flag byte x bit y	Mx.y
Flag byte x	MBx
Flag word x	MWx
Flag double word x	MDx



Notes:

PLC Data	Syntax
Input byte x bit y	Ix.y
Input byte x	IBx
Input word x	IWx
Input double word x	IDx
Output byte x bit y	Qx.y
Output byte x	QBx
Output word x	QWx
Output double word	QDx

Soft-key Status	Syntax
Visible (Default)	Se1
Disabled (Grey text)	Se2
Displayed(Last soft-key used)	Se3

8. Position of short text

The Short Text defined in attribute No.4 is positioned using parameter No.8.
The measurements are in pixels and relate to the top left corner of the main dialogue body.

Three values can be entered to define the position from the left, the position from the top and the width of the short text.
Each entry has to be separated by a comma.

The following example would position the text “Short text” 200 pixels from the left and 100 pixels from the top with a width of 250 pixels:

Def Var1=(R2///" ","Short text"///"$R[0]"/200, 100, 250//)

9. Position of I/O field

The I/O field can be positioned and sized using parameter No.9.
The measurements are in pixels and relate to the top left corner of the main dialogue body.

Four values can be entered to define the position from the left, the position from the top and the width and height of the I/O field.
Each entry has to be separated by a comma.

The following example would position the I/O field for the defined variable, 600 pixels from the left and 100 pixels from the top with a width of 100 pixels and a height of 150 pixels:

Def Var1=(R2///" "," "///"$R[0]"//600, 400, 100, 150/)

1. Colours

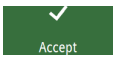
The foreground and background colours of the I/O field can be chosen by using parameter No.10.
The default colours are foreground: Black, background: White.
These are determined by the “Write mode” (See parameter 5)

The following table shows index numbers to be used to achieve the required colour.
The foreground colour is defined first, followed by a comma then the background colour:

Index	Colour
1	Black
2	Red/brown
3	Dark green
4	Light grey
5	Dark grey
6	Blue
7	Red
8	Brown
9	Yellow
10	White

Notes:

A list of predefined softkeys :-

Name	Softkey
SOFTKEY_OK	
SOFTKEY_CANCE	
SOFTKEY_APPLY	
SOFTKEY_MORE	
SOFTKEY_BACK	
SOFTKEY_ASSISTANT_NEXT	
SOFTKEY_ASSISTANT_PREVIOUS	
SOFTKEY_NAV_BACK	

131: END

Notes:

Notes:

133

SINUMERIK

Display Manager

Module Description:

Display Manager is a function of the SINUMERIK , is only available when using 22” or 24” Multitouch panel.
This will allow the manufacturer to utilise the area of the screen when adding custom screens.
The OEM screens are intended as an interface to the specific functions offered by the OEM, also the manufacturer can activate an alpha/numeric keyboard or an OEM control panel.

Module objective:

Upon completion of this module the course participant will be able to

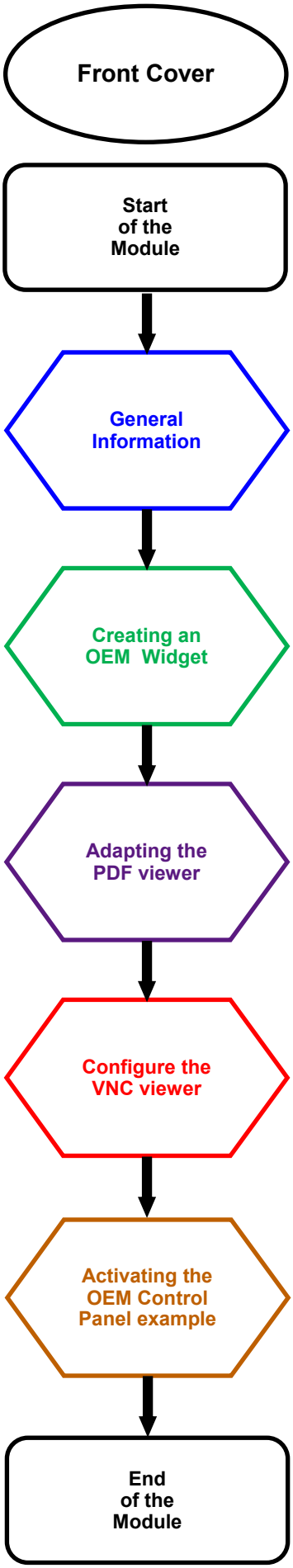
- Utilize the functions of the Display Manager on the Multitouch 22” and 24” panel.

Content:

Sinumerik

General Information
Creating an OEM Widget
Adapting the PDF viewer
Activating the VNC viewer
Activating the OEM Control Panel example

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
General Information
Creating an OEM Widget
Adapting the PDF viewer
Configure the VNC viewer
Activating the OEM Control Panel example
End of the Module

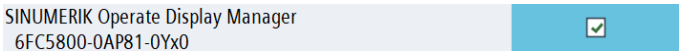
General Information

Using the Display Manager and display configurations, it is possible to reduce SINUMERIK Operate, in other words, run in a non-full-screen mode, and to allocate the vacant space to other applications. These applications are either standard Windows
applications (.EXE) or applications that belong to the Display Manager. Both types of applications can also be developed by third parties and integrated into a display configuration. To display the applications, the screen is divided into "frames". Each frame may contain an application or a menu for controlling an application displayed in another frame. Several screen segments or "Displays" are possible and you can navigate between them during the program runtime. These displays are combined into a display configuration. There may be multiple display configurations, for example, for different screen resolutions.
A display configuration is defined for a specific screen resolution. The available display configurations must be made known to the system in the [displayConfigurations] section of the “systemconfguration.ini” file. The display configuration itself is described in a separate configuration file in the [displays] section of the sldm_1920x1080.ini file.

The option stated below is required.

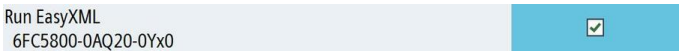
SINUMERIK Operate Display Manager

(Option number: **6FC5800-0AP81-0Yx0**)



Run EasyXML

(Option number - **6FC5800-0AQ20-0Yx0)**



To activate “Display Manager” machine data must be set, which is located in the Setup area of the controller:



Set “Display MD” 9114 $MM_SIDESCREEN = 2 (SINUMERIK Operate Display Manager activated)

Display machine data

9006	$MM_DISPLAY_SWITCH_OFF_INTERVAL	60	min	po
9009	$MM_KEYBOARD_STATE	2		po
9032	$MM_HMI_MONITOR	""		po
9056	$MM_ALARM_ROTATION_CYCLE	3000	ms	po
9057	$MM_ENABLE_CHANNEL_MSG_FILTER	1		po
9100	$MM_CHANGE_LANGUAGE_MODE	1		im
9102	$MM_SHOW_TOOLTIP	1		im
9103	$MM_TOOLTIP_TIME_DELAY	1	s	im
9104	$MM_ANIMATION_TIME_DELAY	10	s	im
9105	$MM_HMI_WIDE_SCREEN	0		po
9106	$MM_SERVE_EXTCALL_PROGRAMS	1		po
9107	$MM_DRV_DIAG_DO_AND_COMP_NAMES	3H		im
9108	$MM_SINUMERIK_INTEGRATE	0		im
9110	$MM_ACCESS_HMI_EXIT	1		po
9111	$MM_HMI_MEM_LIMIT_USER	0		po
9112	$MM_HMI_SKIN	2		po
9113	$MM_EASY_XML_DIAGNOSE	0H		im
9114	$MM_SIDESCREEN	2		po
9115	$MM_SAVE_CREDENTIALS	0		im
9116	$MM_PANEL_CLEAN_MODE_TIME	30	s	im
9900	$MM_MD_TEXT_SWITCH	0		im
9990	$MM_SW_OPTIONS	0H		im

SINUMERIK Operate Sidescreen

Note: The control needs to be switched off/on to activate the new machine data.

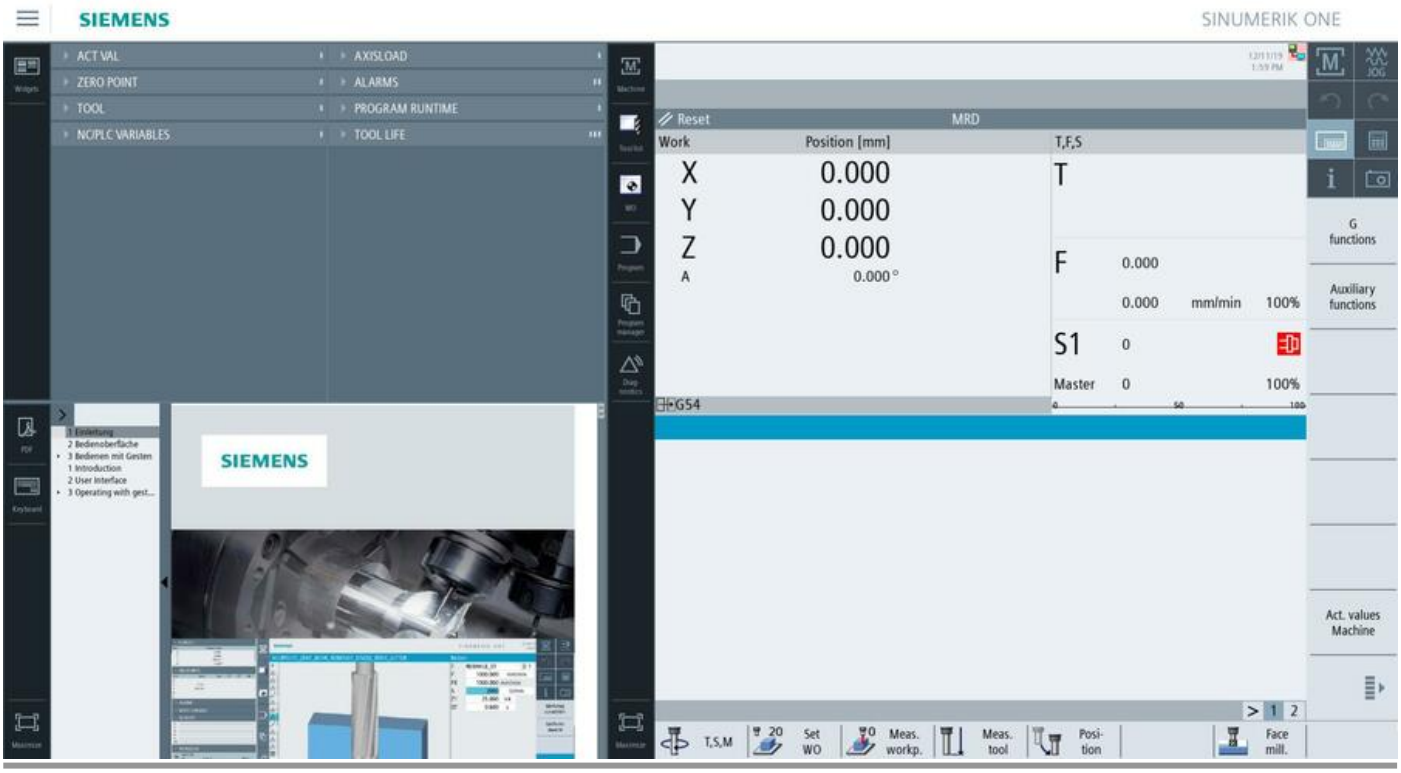
The following applications are included in the standard version:

- PDF Viewer
- VNC Viewer
- Keyboard
- Control Panel
- Application for displaying widgets
- SINUMERIK

This module will help you understand and adapt these standard functions of “Display Manager”.

Notes:

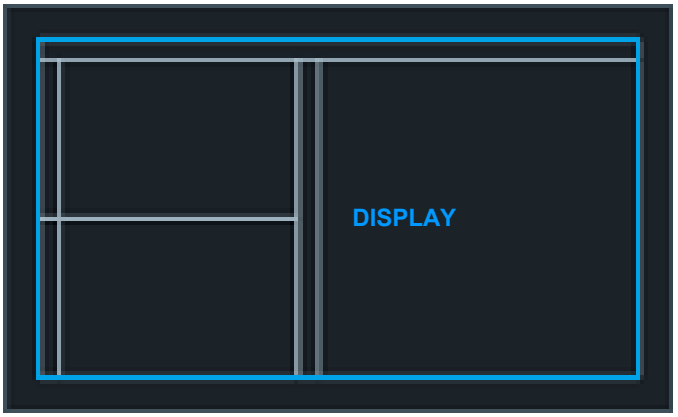
Once “Display Manager” has been activated, this will be shown on the touch screen.

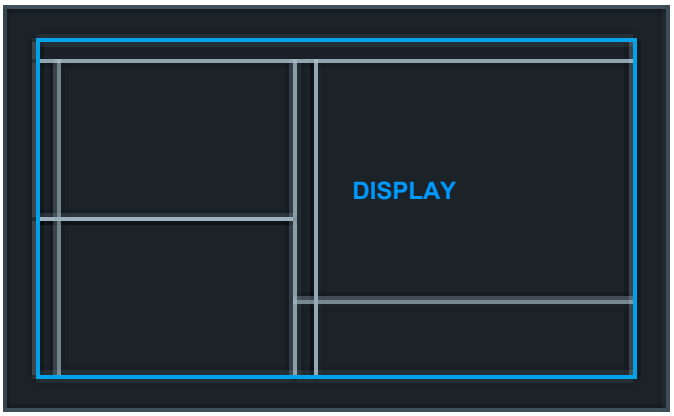


Displays

One or more displays can be defined in the respective configuration file for each display configuration of the touch screen. A name and a list of frames are specified for each of these displays.
The frames describe the layout of the display into rectangular areas in which applications or menus can be displayed.

As shown to the right:





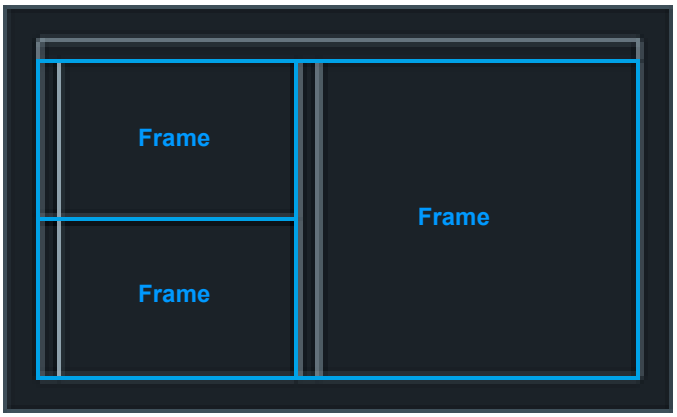
Notes:

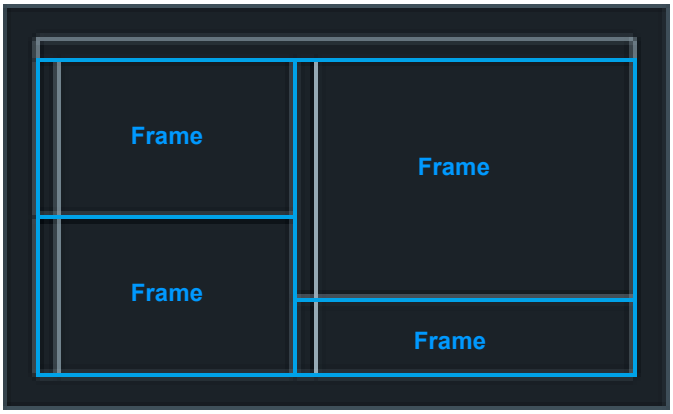
Frames

A frame is an area for applications or menus and primarily describes the desired position and size of the application and menus in pixels. In addition, this is also where the application or menu that should be displayed in the frame after the Display Manager starts is specified. The applications that can be assigned to the frames are specified in the systemconfiguration.ini file.

If SINUMERIK is to be displayed in a frame, the appli-cation name is specified as "OPERATE". It should be noted that SINUMERIK can only be used in certain sizes.
The currently supported sizes (in pixels) are:
- 640x480
- 760x505
- 800x480
- 800x600
- 1024x768
- 1024x1014
- 1080x810
- 1280x768
- 1280x800
- 1280x1024
- 1300x768
- 1366x768
- 1440x1080
- 1454x1080
- 1854x1014

As shown to the right:





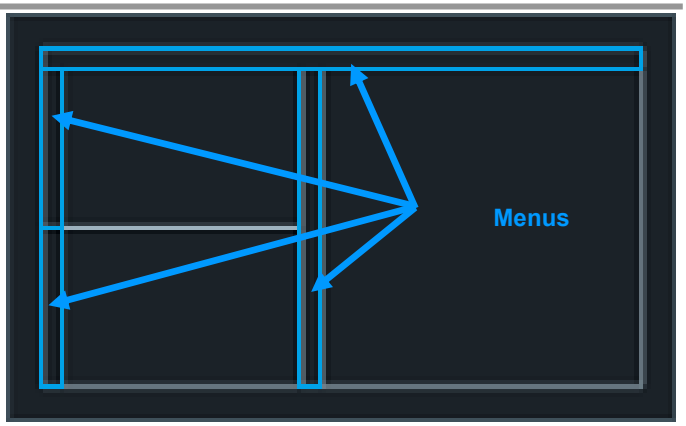
Menus

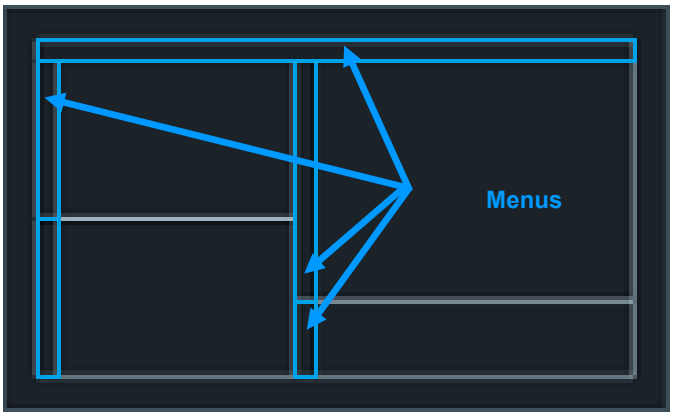
In addition to frames, menus can also be displayed in the frames of a display. These menus are typically used to display certain applications in another frame or to switch displays.

A menu configuration consists of four parts:

- Description of the individual menu items (section [menuitems])
- Description of the look and feel of the menu items (section [menuitemstyle])
- Description of the menu structure (section [menus])
- Description of the look and feel of the menu (section [menustyle])

As shown to the right:



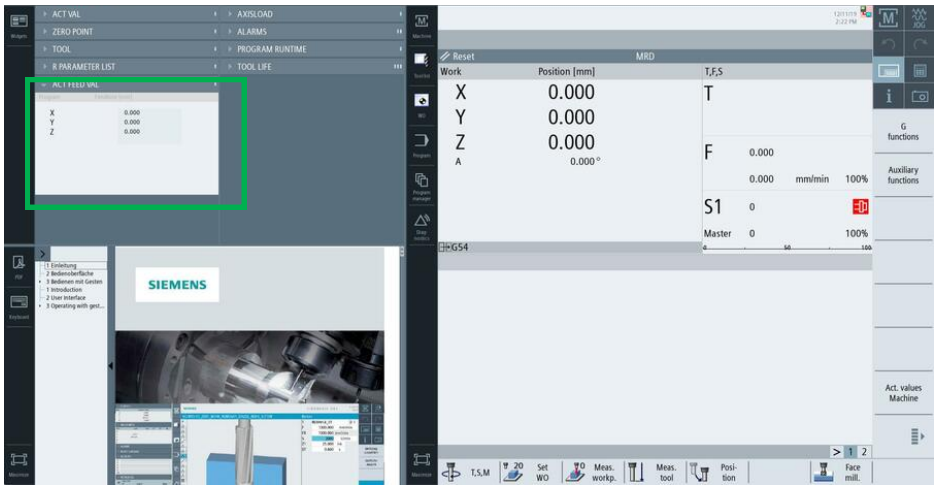


Notes:

Creating an OEM Widget

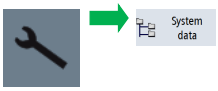
The widget frame (top left-hand frame) allows a quick and permanent view of important information (system or OEM) and enables the selection of this information for quick viewing.

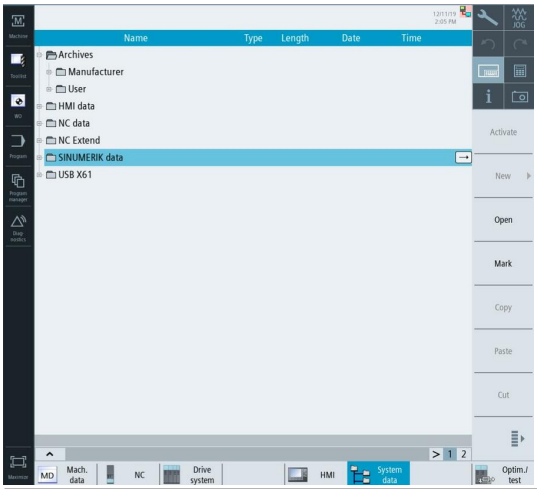
This picture below shows the objective of this section.



The widget frame (top left-hand frame) allows a quick and permanent view of important information (system or OEM) and enables the selection of this information for quick viewing.

The files required for the Widget in the screen area need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





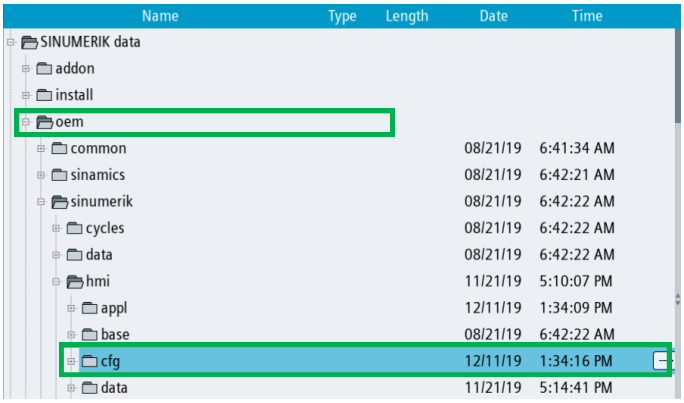
Creating the “sldmwidgets1.ini” file on the controller

This file sets up the link for the widgets in the screen. The file has to be located in one of the following paths/directories:-

SINUMERIK data\oem\sinumerik\hmi\cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing.

Example: sldmwidgets1.ini file
The following example shows the content of the “**sldmwidgets1.ini**” file for the OEM Widget in the left-hand column of the widget frame of the display (to display in the right-hand column of the widget frame, create **“sldmwidgets2.ini”.**

```
[Page_WidgetsPage]
ELEMENT005= name:=Programvalues, implementation:=SISideScreenElement

[Element_Programvalues]
TextId=ACT FEED VAL
TextFile=programvalues_widget
TextContext=EASY_XML
WIDGET001= name:=programvalues_widget, implementation:=slagmforms.SIEESideScreenWidget
```

This being the name of the xml file for the application content.



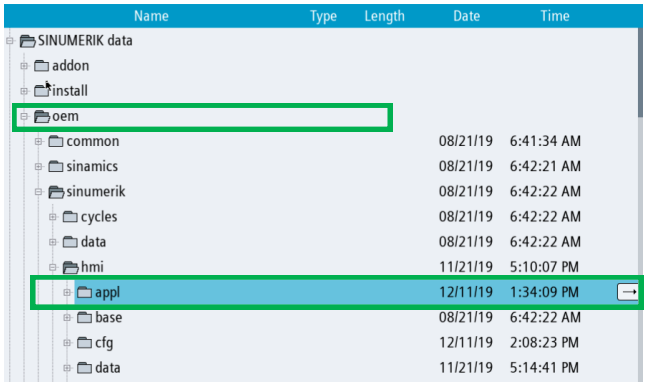
Creating the **“programvalues_widget.xml”** file on the controller for the “OEM widget”.

This file sets up the application content of the widget in the screen. The file has to be located in one of the following paths/directories:-

SINUMERIK data\oem\sinumerik\hmi\appl

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/appl

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing.

Notes:

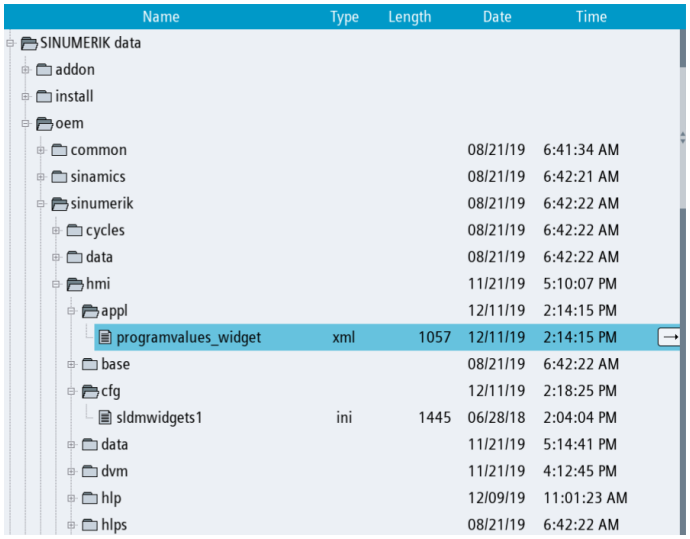
Example: programvalues_widget.xml file

The following example will show the content **"programvalues_widget.xml"** file for the "OEM Widget" in the screen area of the controller.

```
<DialogGui >
  <menu name = "main">
    <open_form name = "PROGRAMVALUES_LIST" />
  </menu>
<form name = "PROGRAMVALUES_LIST">
        <init>
           <caption></caption>
           <data_access type="true" />
        <control name = "b1" xpos = "125" ypos = "34" refvar="nck/Channel/GeometricAxis/actFeedRate[1]" hot-
link="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" color_bk="#e5ecf0"/>
        <control name = "b2" xpos = "125" ypos = "54" refvar="nck/Channel/GeometricAxis/actFeedRate[2]" hot-
link="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" color_bk="#e5ecf0"/>
        <control name = "b3" xpos = "125" ypos = "74" refvar="nck/Channel/GeometricAxis/actFeedRate[3]" hot-
link="true" height="34" fieldtype="readonly" format="%9.3f" font = "1" time="superfast" color_bk="#e5ecf0"/>
        </init>
        <paint>
                <text xpos = "23" ypos = "34">X</text>
                <text xpos = "23" ypos = "54">Y</text>
                <text xpos = "23" ypos = "74">Z</text>
                <box xpos="0" ypos="0" width="276" height="26" color="#c5d2da" />
                <text xpos="3" ypos="3" color="#000000">Program                FeedRate [mm]</text>
        </paint>
</form>
```



Below: The **"programvalues_widget.xml"** file located in the **appl** directory

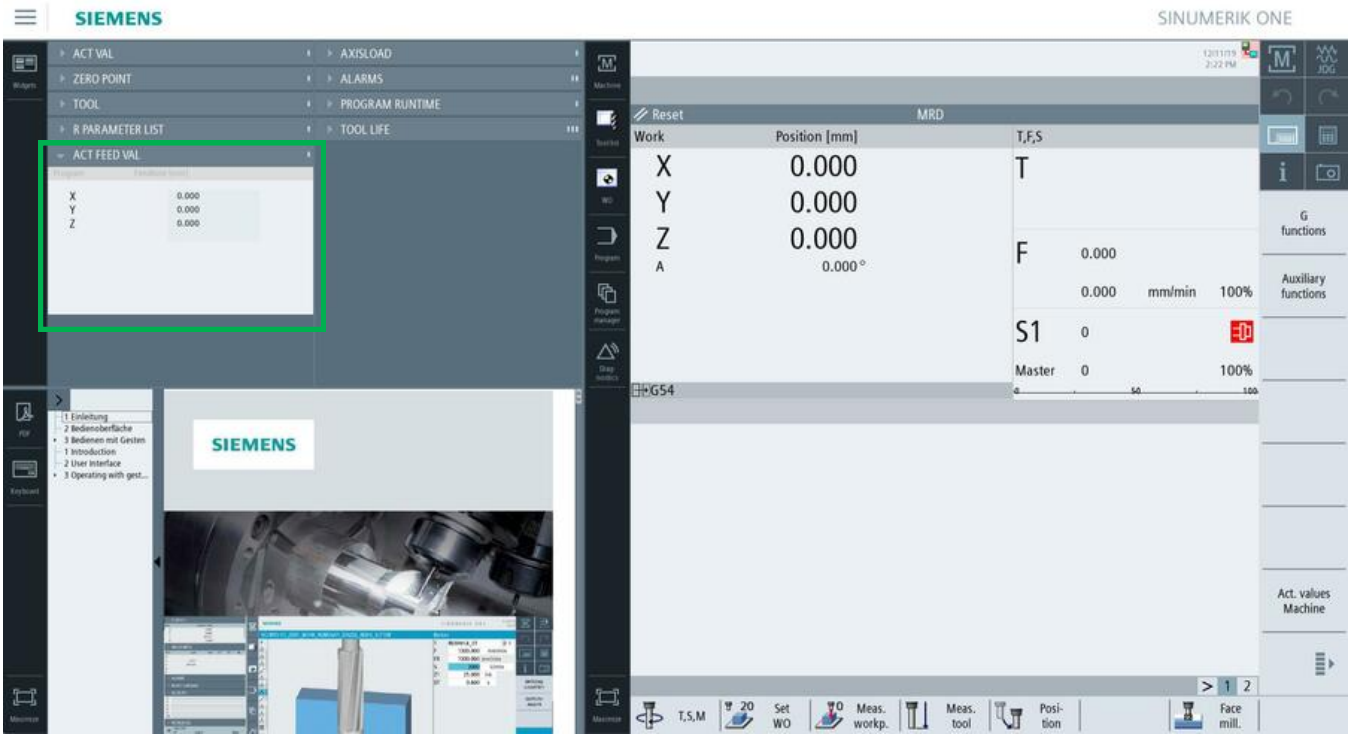


To activate the files, perform:



Notes:

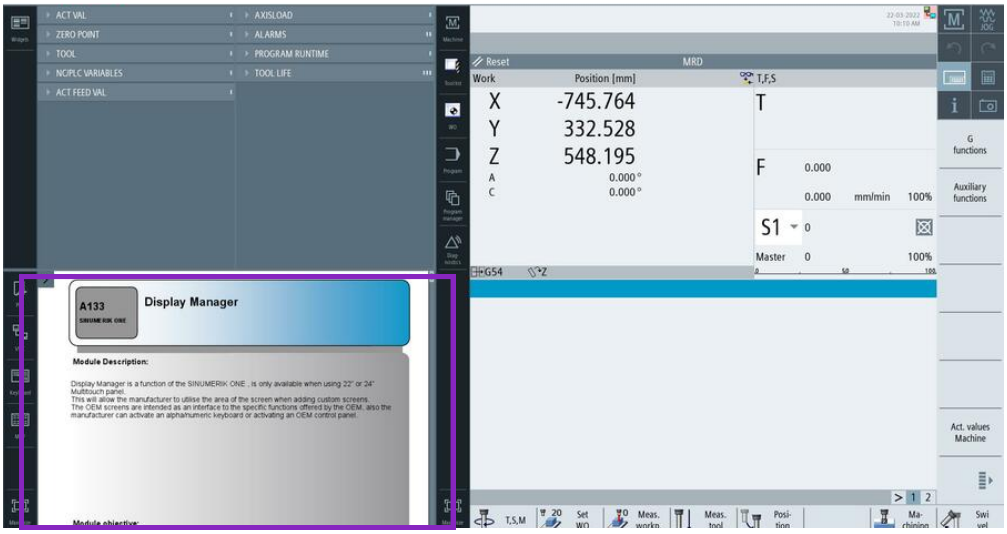
The result is shown below.



Adapting the PDF viewer

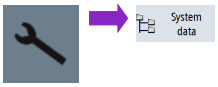
The PDF viewer frame gives a quick and permanent view of important information that is required for the operation of the machine (drawings, tool information, machine operation, etc)

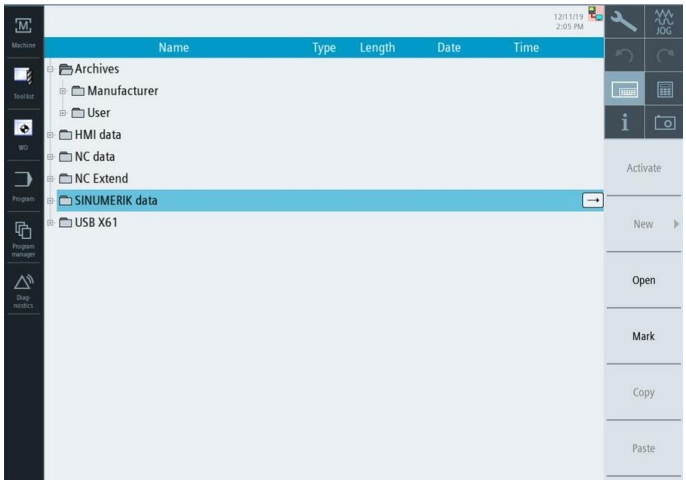
This picture below shows the objective of this section.



Notes:

The file required for the PDF viewer area will need to be created, so that a different file can be shown. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





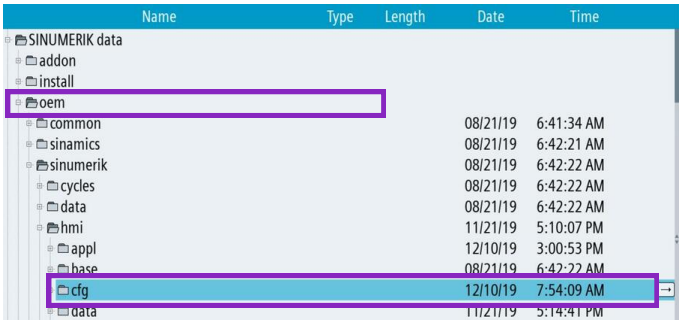
Creating the “systemconfiguration.ini” file on the controller

This file sets up the link for the PDF viewer. The file has to be located in one of the following paths/directories:-

SINUMERIK data\oem\sinumerik\hmi\cfg

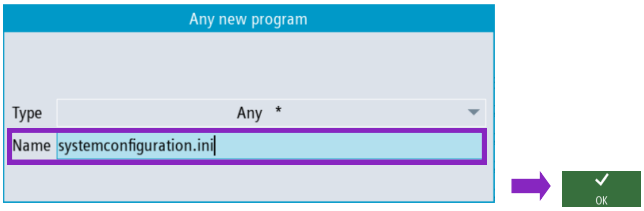
IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing.

Example: systemconfiguration.ini file
The following example will show the content of the “**systemconfiguration.ini**” file for the PDF Viewer**.**

[dialogs]
DLG107= name:=SIPdfApp, implementation:=sldmpdfviewerapp.SIDmPdfViewerApp, process:=SIHmiHost1, pre-load:=false cmdline:="-pdfFile 133_EN Display Manager 0615.pdf -matteColor #3b4c58"

This being the name of the PDF file that the viewer will display.
This file will have to be placed in the “appl” directory.



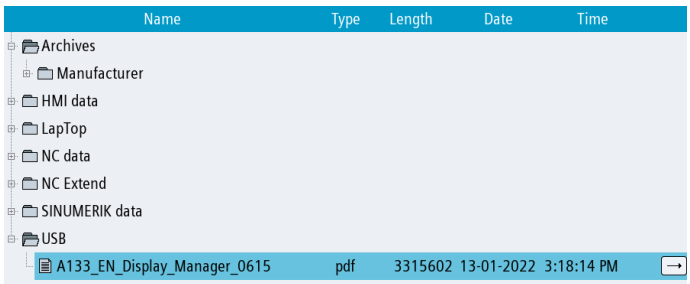
Notes:

In this example the **133_EN Display Manager 0615.pdf** file can be copied from a USB memory stick into the relevant folder on the control.

This file is the content that the PDF file will be displayed in the frame. The file has to be located in one of the following paths/ directories:-
SINUMERIK data\oem\sinumerik\hmi\appl

IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/appl

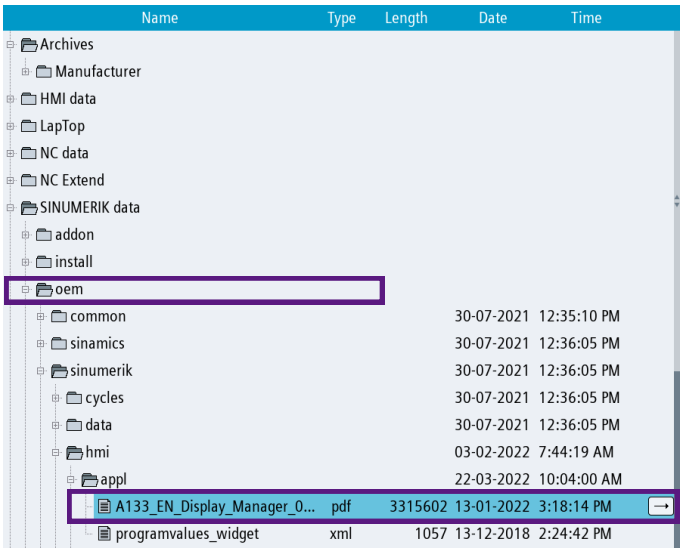
Use the cursor keys to locate and open the USB directory.



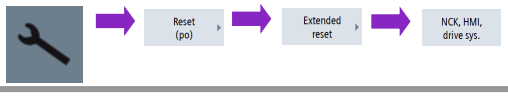


Locate the directory “appl” and paste the PDF file

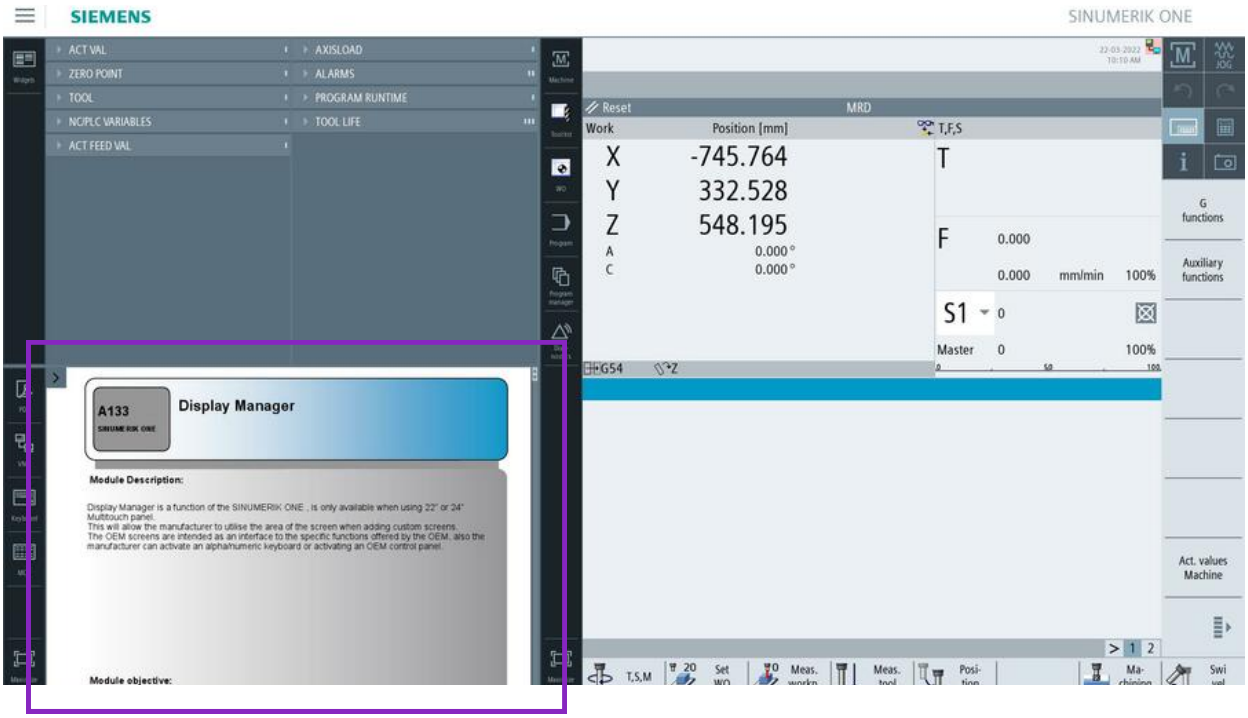




To activate the files, perform:



The result is shown below.

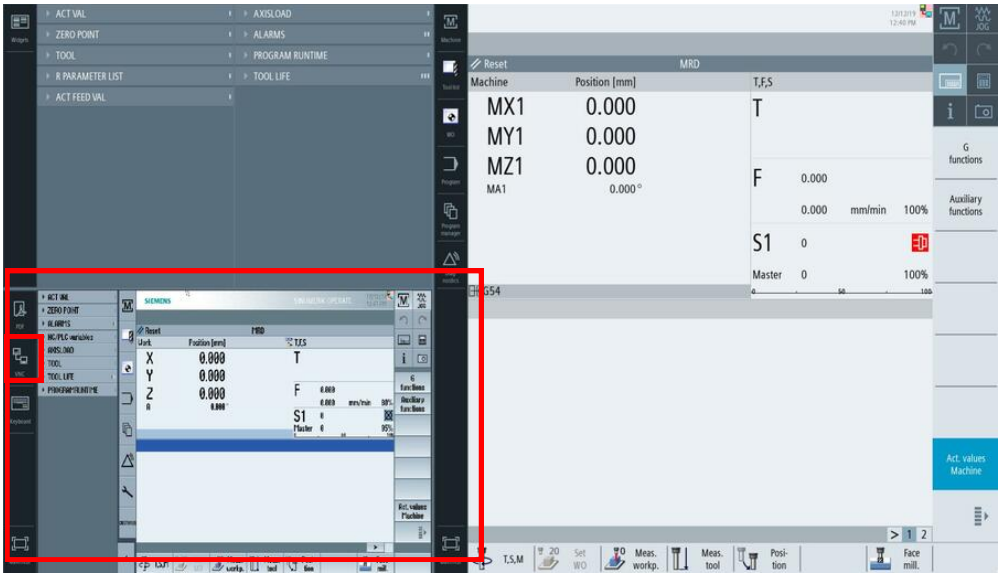


Notes:

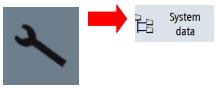
VNC viewer

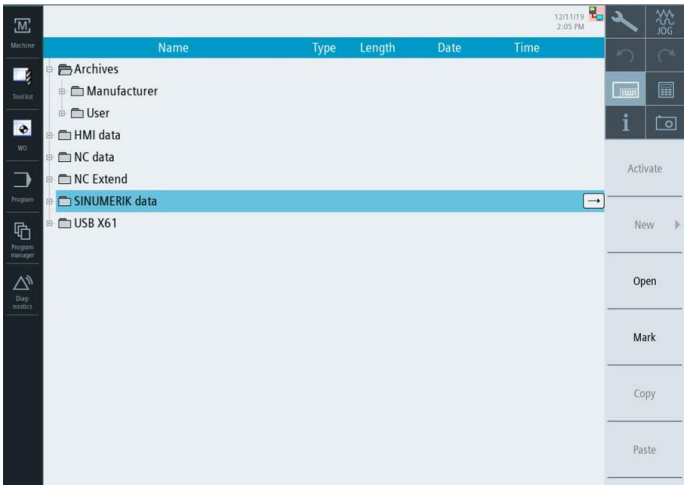
The VNC viewer frame gives a quick and permanent view of another SINUMERIK machine or a PC that has VNC server installed on it.

This picture below shows the objective of this section.



The file required for the VNC viewer frame will need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





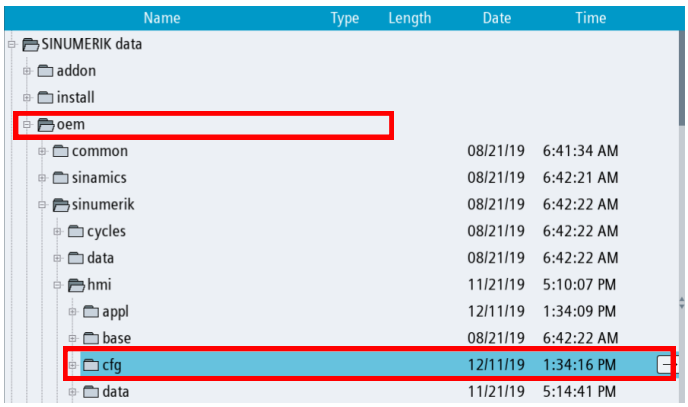
Creating the “systemconfiguration.ini” file on the controller

This file sets up the link for the keyboard on the screen. The file has to be located in one of the following paths/directories:-

SINUMERIK data\oem\sinumerik\hmi\cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

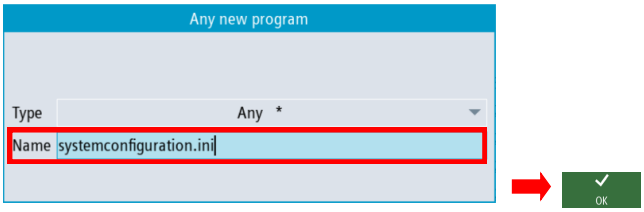
Use the cursor keys to locate and open this directory.



Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing.

Example: systemconfiguration.ini file
The following example will show the content of the “**systemconfiguration.ini**” file for the PDF Viewer**.**

[dialogs]
DLG108= name:=SIVncApp, implementation:=sldmvncviewerapp.SIDmVncViewerApp, process:=SIHmiHost1, pre-load:=false, cmdline:="-configuration MyPC"

This being the name in the configuration file for the connection details of the viewer.



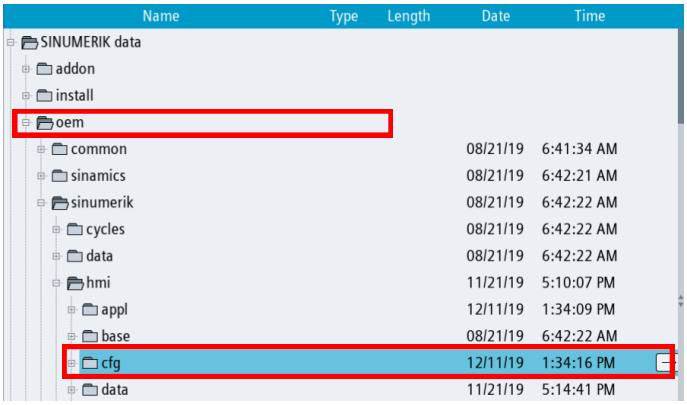
In this example a connection will be made with another SINUMERIK.

Creating the “slvncconfig.ini” file on the controller
This file sets up the link for the VNC viewer. The file has to be located in one of the following paths/directories:-

SINUMERIK data\oem\sinumerik\hmi\cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing.



Notes:

Example: slvncconfig.ini file.

The following example will show the content of **"slvncconfig.ini"** file for the configuration details of the IP address and TCP support for the connection to another machine

[MyPC]
host=192.168.100.2
port=5900
;password=



The controller with "Display Manager" running, set X130 (factory network) with:
IP address 192.168.100.1
Subnetwork musk 255.255.255.0

The controller to be viewed, set X130 (factory network) with:
IP address 192.168.100.2
Subnetwork musk 255.255.255.0

Also set the firewall exceptions for the VCN viewer:



Make sure to connect a network cable between X130 on both controllers.

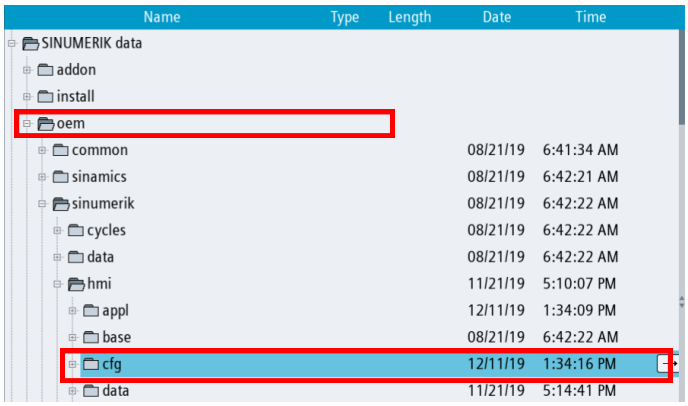
In this example a menu VNC Viewer icon will be created in the menu bar below the PDF icon, this will start the application

Creating the "sldm_1920x1080.ini" file on the controller
This file sets up the link for the VNC viewer. The file has to be located in one of the following paths/directories:-

SINUMERIK data\oem\sinumerik\hmi\cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



To create the file select the "New" soft-key and ensure that the "Any" soft-key is selected.



The newly created file will be automatically opened in preparation for editing.

Notes:

Example: sldm_1920x1080.ini file.

The following example will show the content of **"sldm_1920x1080.ini"** file which will create a menu icon on the menu bar below the PDF icon.

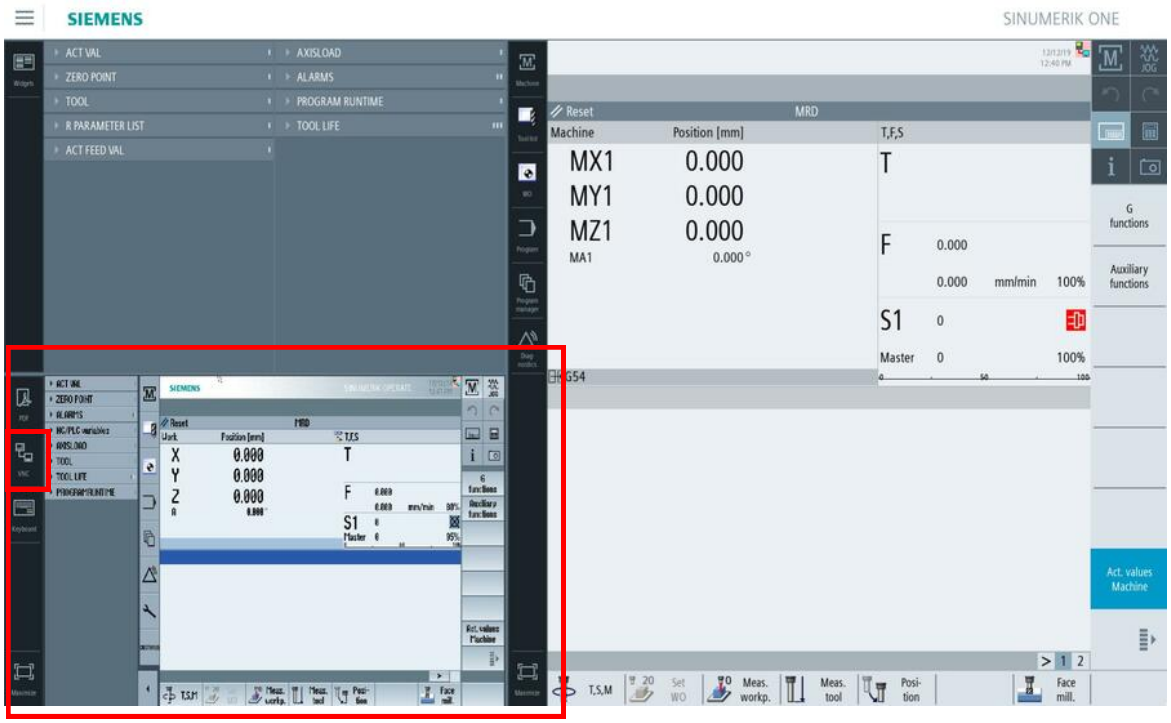
```
[menuitems]
; app buttons
MENUITEM127= name:=miVnc, menuItemStyle:=misMenu, onClicked:="hidePopup(); showApp(defaultFrame,
SlVncApp)", image:=dm_vnc.png, textID:=SL_DM_VNC
```



To activate a complete power off/on of the control is required.

The result is shown below.



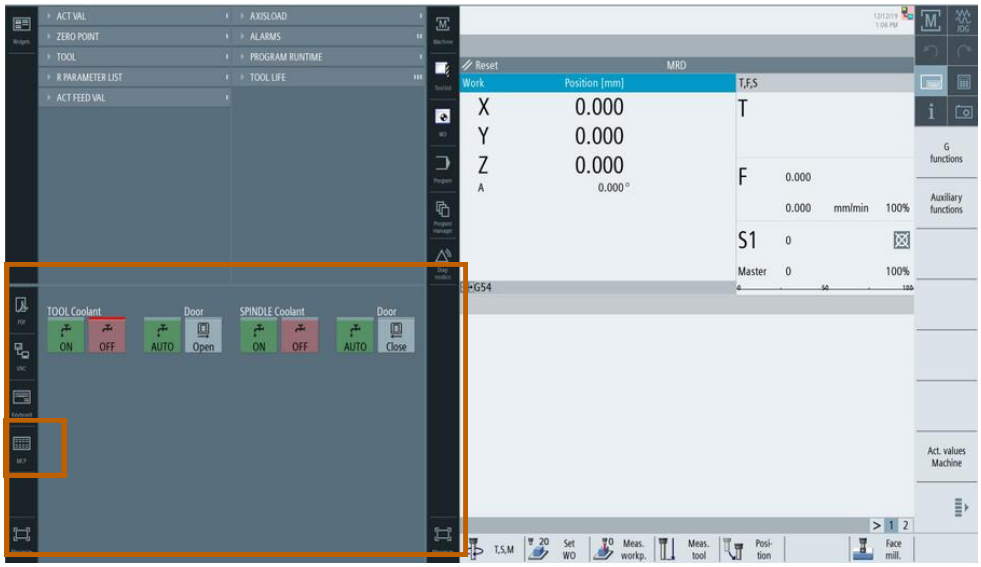


Notes:

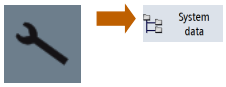
Activating the OEM Control Panel example

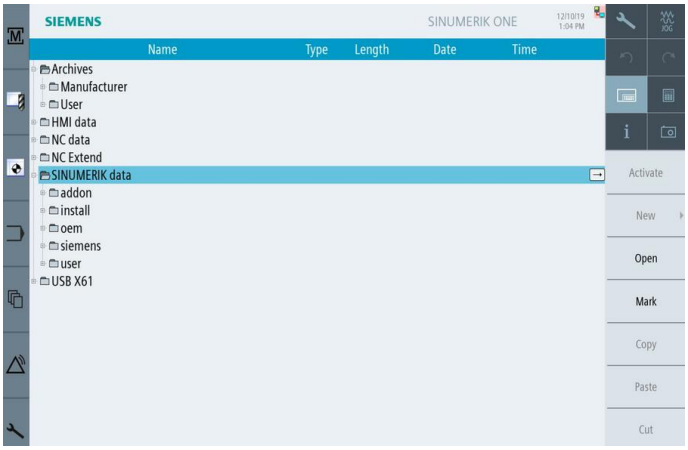
The Control Panel frame gives a quick and permanent view of any OEM control functions.

This picture below shows the objective of this section.



The file required for the Control Panel area will need to be created. This can be done directly on the controller by navigating to the relevant folders. Use the following key sequence to locate the “System data” area.





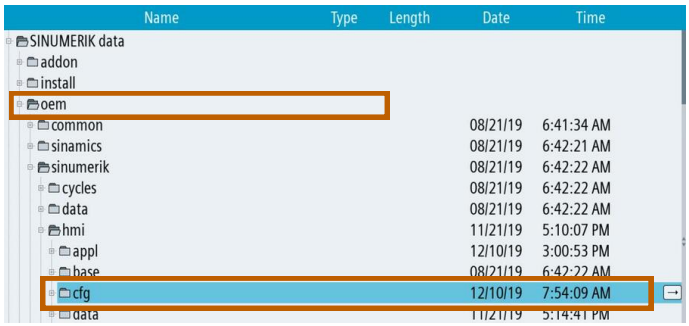
Creating the “systemconfiguration.ini” file on the controller

This file sets up the link for the Control Panel. The file has to be located in one of the following paths/directories:-

SINUMERIK data\oem\sinumerik\hmi\cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



Notes:

To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.





The newly created file will be automatically opened in preparation for editing.

Example: systemconfiguration.ini file
The following example will show the content of the “**systemconfiguration.ini**” file for the Control Panel**.**

DLG110= name:=SIMcpApp, implementation:=sldmsidescreenapp.SISideScreenDialog, process:=SIHmiHost1, pre-load:=false, cmdline:="-sidescreen1 sldmmcppage.ini"

This being the name in the configuration file for the Control Panel.



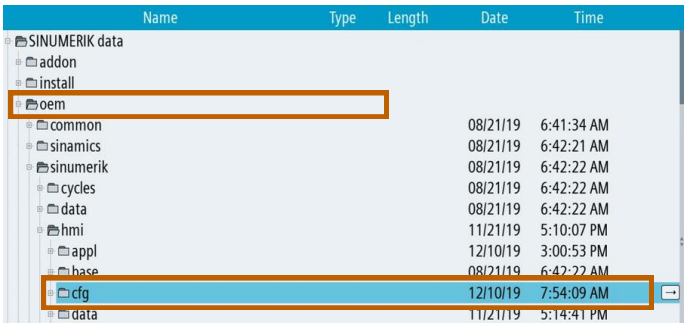
Creating the “sldmmcppage.ini” file on the controller

This file sets up the connection for the VNC viewer. The file has to be located in one of the following paths/directories:-

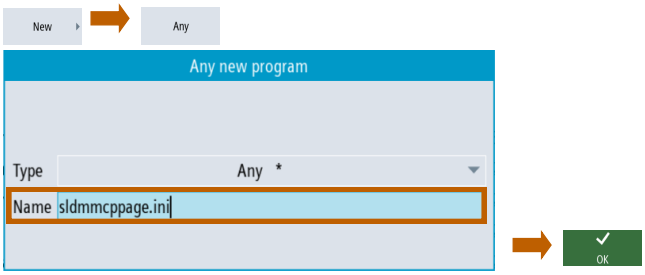
SINUMERIK data\oem\sinumerik\hmi\cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.



The newly created file will be automatically opened in preparation for editing.

Notes:

The newly created file will be automatically opened in preparation for editing.

Example: sldmmcppage.ini file
The following example will show the content of the "**sldmmcppage.ini**" file for the Control Panel**.**

[Sidescreen]
PAGE001= name:=SIMcpApp, implementation:=slsidescreenmcppage.SISideScreenMcpPage

This being the partial name of the xml file for the application content.



In this example the placement of the icons in the frame are created.

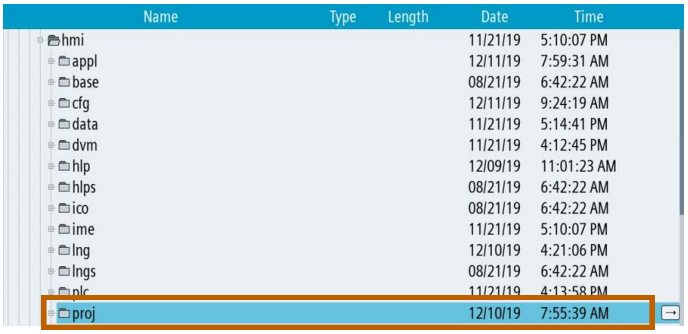
Creating the "sldsidescreenmcppageconfig.xml" file on the controller

This file sets up the link for the Control Panel. The file has to be located in one of the following paths/directories:-

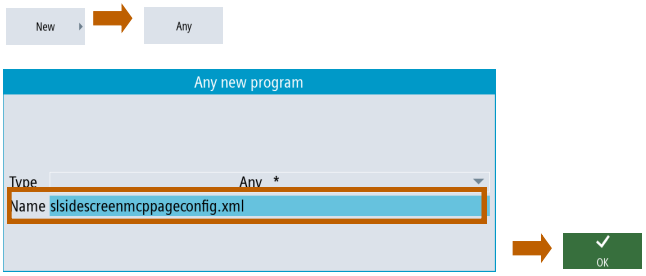
SINUMERIK data\oem\sinumerik\hmi\proj

IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/proj

Use the cursor keys to locate and open this directory.



To create the file select the "New" soft-key and ensure that the "Any" soft-key is selected.



The newly created file will be automatically opened in preparation for editing.

Notes:

The newly created file will be automatically opened in preparation for editing.

Example: sldsidescreenmcppageconfig.xml file
The following example will show the content of the "**sldsidescreenmcppageconfig.xml**" file for the PDF Viewer**.**

```
<?xml version="1.0" encoding="utf-8" ?>
<sideScreenControlPanel>
 <mainBlock id="sideScreenMcpBlock">
  <block position="10">
   <row position="1">
    <buttonGroup spacing="true" title="TOOL Coolant">
        <button position="1"text="ON" icon="slsidescreenmcp_cooling.png" style="greenStyle">
      <onPressed address="HmiUserKey1"/>
      <active address="HmiUserLed1"/>
     </button>
     <button position="2" text="OFF" icon="slsidescreenmcp_cooling_off.png" style="redStyle">
      <onPressed address="HmiUserKey2"/>
      <active address="HmiUserLed2"/>
     </button>
    </buttonGroup>
   </row>
  </block>
  <block position="11">
   <row position="1">
    <buttonGroup spacing="true" title="              Door">
     <button position="1" text="AUTO" icon="slsidescreenmcp_cooling.png" style="greenStyle">
      <onPressed address="HmiUserKey3"/>
      <active address="HmiUserLed3"/>
     </button>
     <button position="2" text="Open" icon="slsidescreenmcp_door_open.png">
      <onPressed address="HmiUserKey4"/>
      <active address="HmiUserLed4"/>
     </button>
    </buttonGroup>
   </row>
```

```
  </block>
  <block position="12">
   <row position="1">
    <buttonGroup spacing="true" title="SPINDLE Coolant">
     <button position="1" text="ON" icon="slsidescreenmcp_cooling.png" style="greenStyle">
      <onPressed address="HmiUserKey5"/>
      <active address="HmiUserLed5"/>
     </button>
     <button position="2" text="OFF" icon="slsidescreenmcp_cooling_off.png" style="redStyle">
      <onPressed address="HmiUserKey6"/>
      <active address="HmiUserLed6"/>
     </button>
    </buttonGroup>
   </row>
  </block>
  <block position="13">
   <row position="1">
    <buttonGroup spacing="true"  title="              Door">
     <button position="1" text="AUTO" icon="slsidescreenmcp_cooling.png" style="greenStyle">
      <onPressed address="HmiUserKey7"/>
      <active address="HmiUserLed7"/>
     </button>
     <button position="2" text="Close" icon="slsidescreenmcp_door_close.png">
      <onPressed address="HmiUserKey8"/>
      <active address="HmiUserLed8"/>
     </button>
    </buttonGroup>
   </row>
  </block>
 </mainBlock>
</sideScreenControlPanel>
```



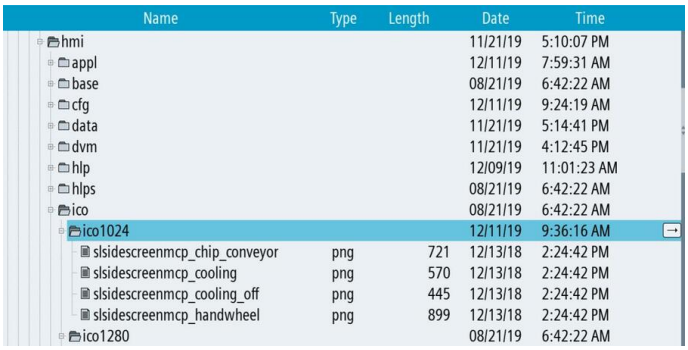
Notes:

Storing own icons

You can create your own icon files under one of the following paths:
/oem/sinumerik/hmi/ico or
/user/sinumerik/hmi/ico
Depending on the screen resolution, under the following sub-directories:
/ico1024 for HD Ready
/ico1920 for Full HD

In the example, four icons are used:
slsidescreenmcp_chip_conveyor.png
slsidescreenmcp_cooling.png
slsidescreenmcp_cooling_off.png
slsidescreenmcp_handwheel.png





In this example a menu icon will be created in the menu bar below the Keyboard icon.

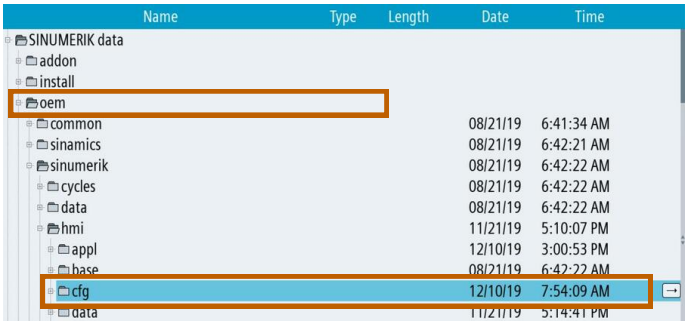
Creating the “sldm_1920x1080.ini” file on the controller

This file sets up the link for the Control Panel The file has to be located in one of the following paths/directories:-

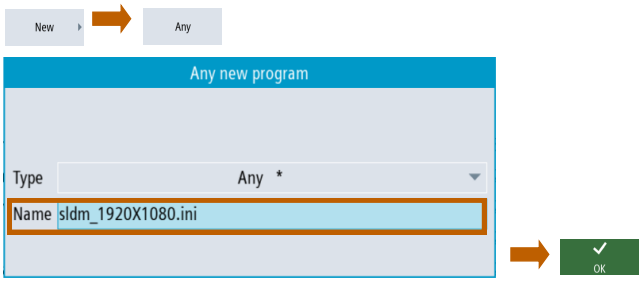
SINUMERIK data\oem\sinumerik\hmi\cfg

IPC: C:/ProgramData/Siemens/MotionControl/oem/ sinumerik/hmi/cfg

Use the cursor keys to locate and open this directory.



To create the file select the “New” soft-key and ensure that the “Any” soft-key is selected.



The newly created file will be automatically opened in preparation for editing.

Notes:

Example: sldm_1920x1080.ini file.

The following example will show the content of **"sldm_1920x1080.ini"** file which will create a menu icon on the menu bar below the PDF icon.

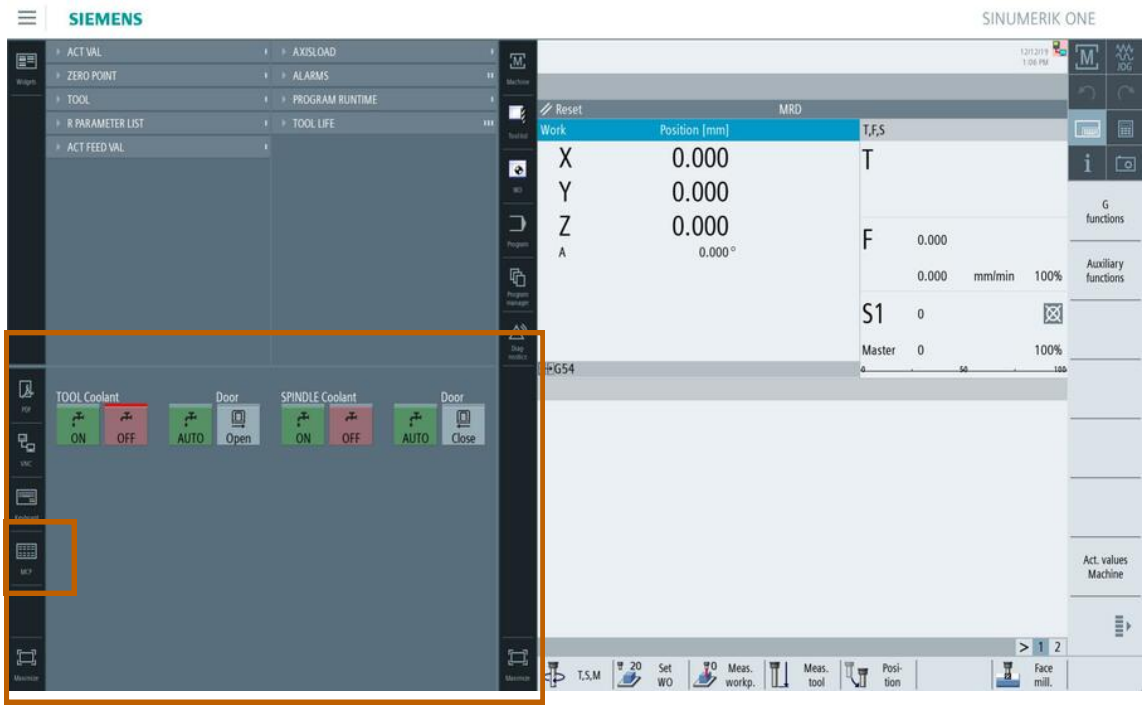
```
[menuitems]
; app buttons
MENUITEM124= name:=miMcp, menuItemStyle:=misMenu, onClicked:="hidePopup(); showApp(defaultFrame, SIM-
cpApp)", image:=dm_mcp.png, textID:=SL_DM_MCP
MENUITEM125= name:=miMcpBelow, menuItemStyle:=misMenu, onClicked:="hidePopup(); showApp(defaultFrame,
SIMcpApp)", image:=dm_mcp_bottom.png, textID:=SL_DM_MCP
```



To activate a complete power off/on of the control is required.

The result is shown below.





133: END

Notes:

A136

SINUMERIK ONE

Virtual Machine Data Management

Module Description:

It is necessary to have procedures in place for saving and restoring the data stored in the digital twin (CMVM).

This module describes a method for saving NC data into an archive.

Module objective:

Upon completion of this module you will be able to perform the following tasks:

- Create and restore archives within CMVM

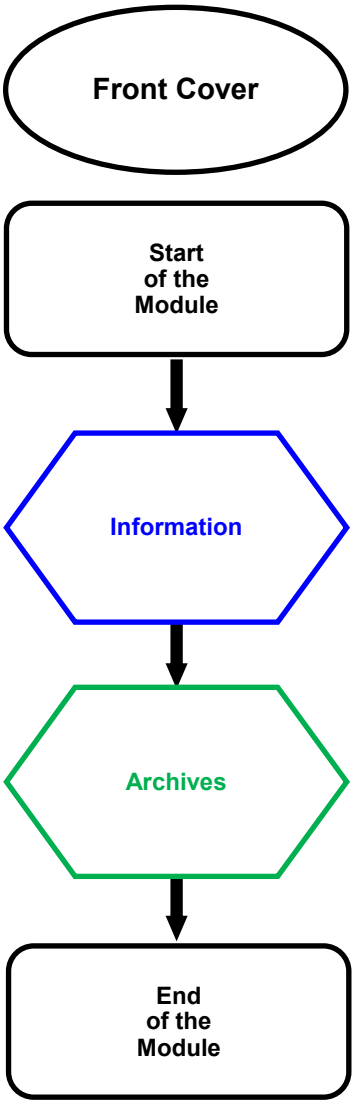
Content:

Information

Archives

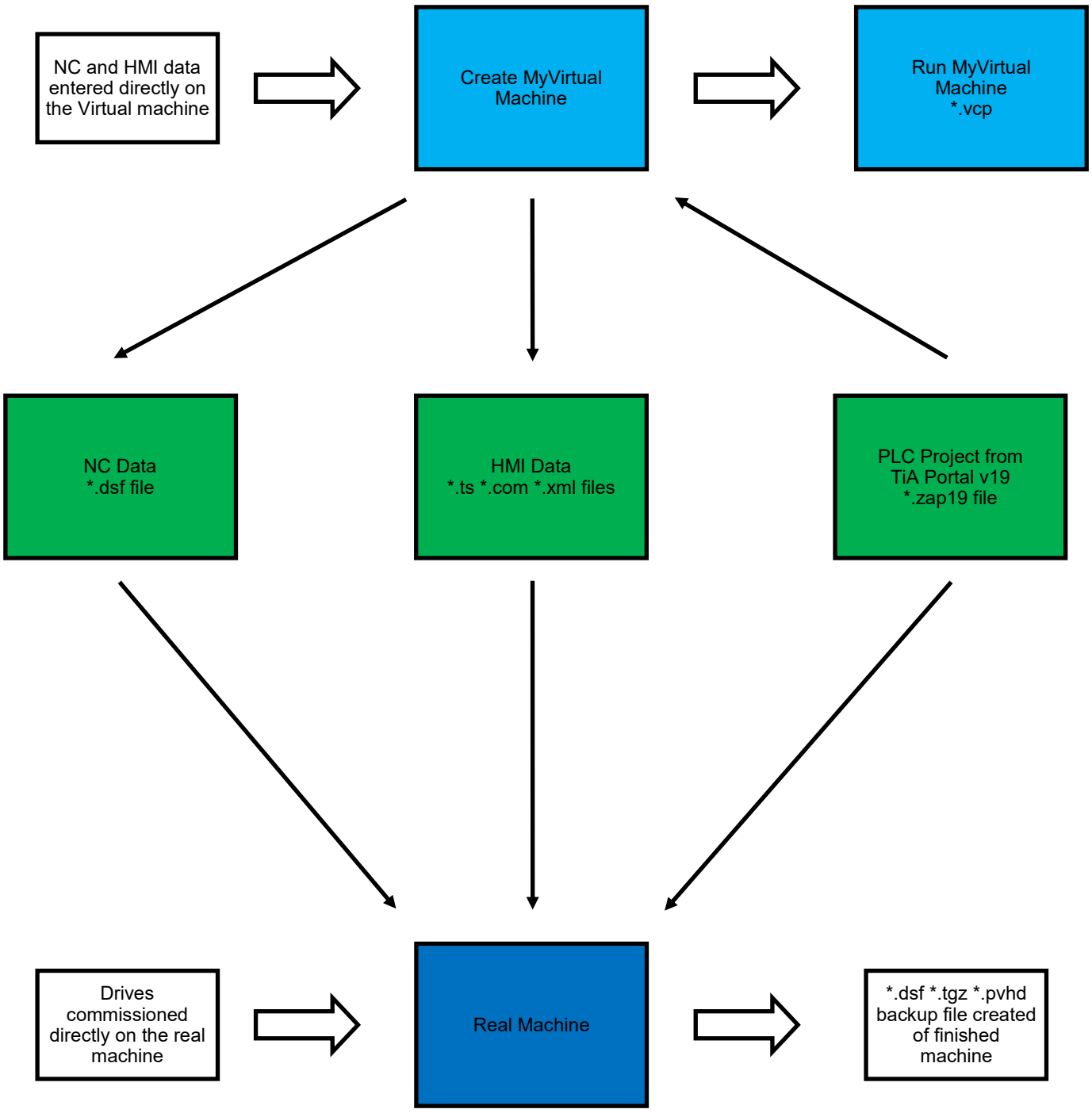
Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Information
Archives
End of the Module

Information

Concept of data movement



Notes:

Archives

Archives can be used to save NC data.
There is no drive data on the digital twin, likewise the PLC must be loaded in the digital twin/real machine via TiA Portal.

To create an Archive locate the relevant area by using the following key sequence.



The screen displayed has five choices:

All data of this machine (backup) - Archives of the five control components (NC, PLC, Drives, HMI, system settings) can be created with this option

Data for a series setup - Archives of the four control components (NC, PLC, Drives, HMI) can be created with this option

Factory settings of this machine (original) - Used to create an archive that will return the machine to its delivery state.

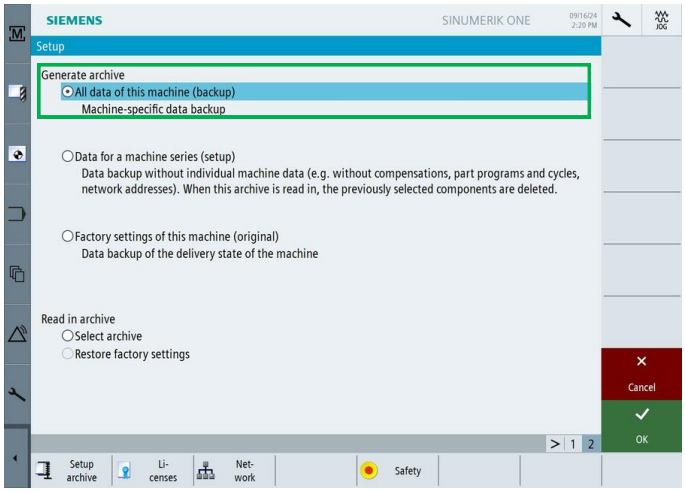
Select archive - Archives are read-in with this option.

Restore factory settings - For reading in the original status archive

Creating an Archive:

The method for creating archives is the same for the virtual machine as it is for a real machine.

Ensure that the option “All data of this machine (backup)” is selected and select “OK”.





Control Components:

The controller data is split into six components

NC data:- The NC data consists of:
All machine and Setting data (except for drive machine data)
Machine data
Setting data
Option data
Global user data (GUD)
Tool and magazine data
Protection zone data
R paramters
Zero offsets
Workpieces, part programs and subprograms
Manufacturer cycles
User cycles
Definitions and macros
Collision avoidance data
Compensation data (QEC, CEC, EEC)
Axis optimization data (AST)
Compile cycles

PLC data: This will create an archive suited to restoring the PLC user program.

Drive data: This will create an archive containing data for the Sinamics drives.

Notes:

HMI data: This saves data that is stored on the HMI.
The list comprises of:-
Cycle storage
Texts
Templates
Applications
Configurations
Version data
Logs
User views
Dictionaries

System settings: This saves data that is stored on the HMI.
The list comprises of:-
Drive configurations
TCU settings
Network settings (for embedded only)
mmc.ini

Local drive: This saves data that is stored on the local drive (NCextend).

The six control components are available for selection. They can be selected in any combination to create archives containing different data.

Creating an NC archive

The following example will create an NC archive.

Use the cursor and select keys to select the necessary check boxes.

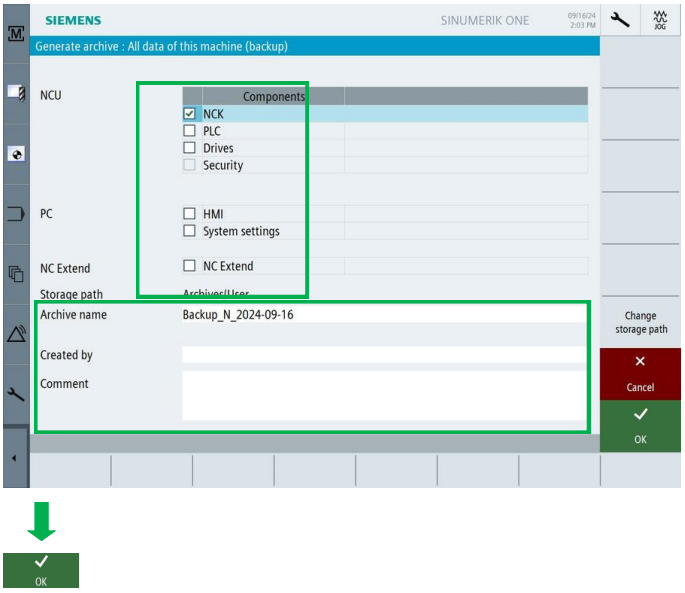
The comment area at the bottom of the screen offers the chance to add information about the archive.

Note:

A HMI archive will not load into an “Embedded” or “IPC” system due to the HMI path.

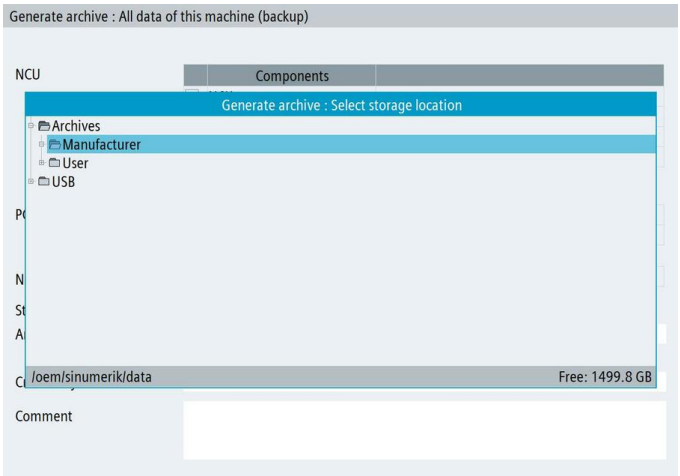
IPC:
C:\ProgramData\Siemens\MotionControl\oem\sinumerik\hmi

Embedded:
SINUMERIK data\oem\sinumerik\hmi



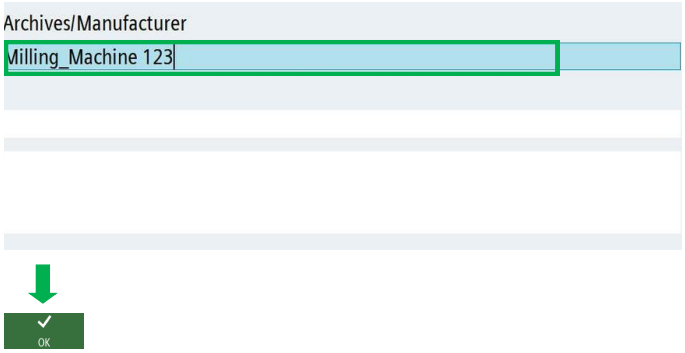
The destination of the archive can be changed, this can be locally on the system of the virtual software or externally to a USB device or networked PC.
It is suggested that an archive be kept locally on the controller. Copies can be made externally for use in the event of a data restore.



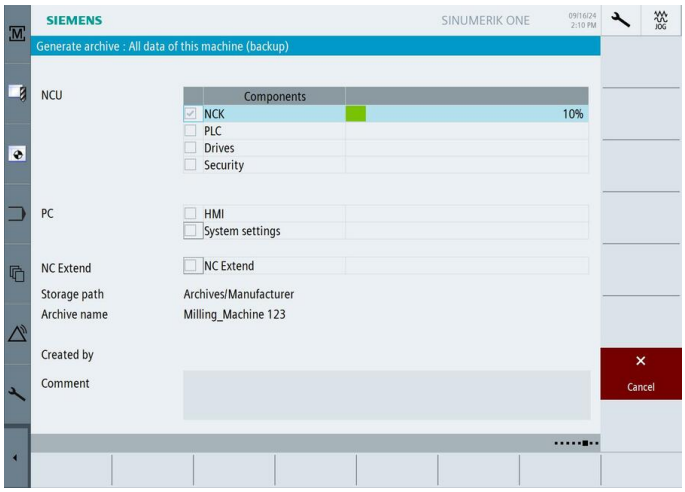


Notes:

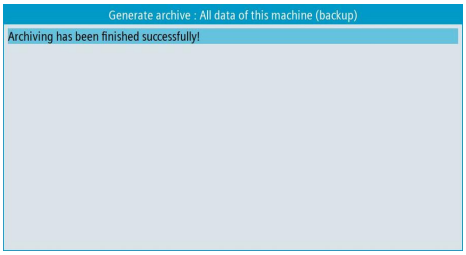
A name is automatically entered for the archive, but this can be changed.

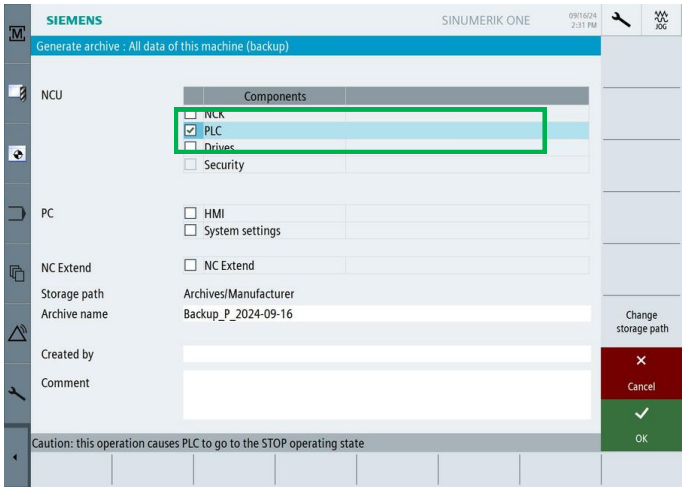


The archive will now be created and stored in the destination specified earlier.



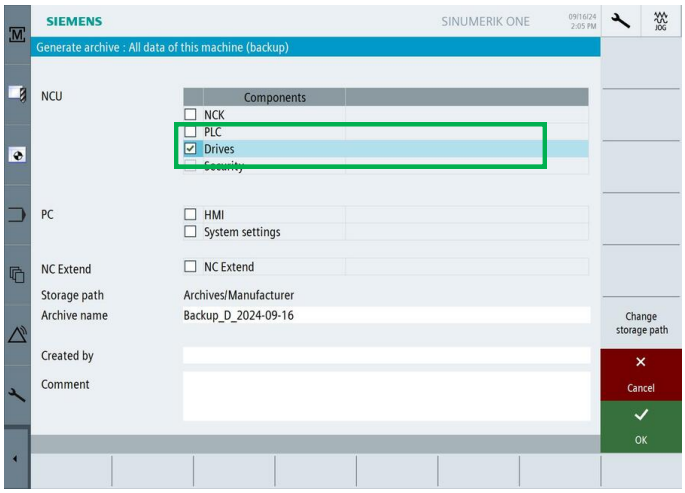
An information box indicates that the archive was successfully created.



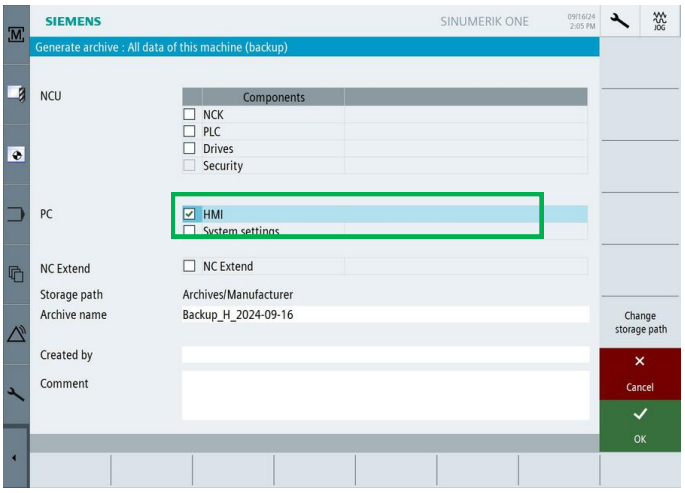


A **PLC** archive may be required when there is a need to test the PLC in another Virtual machine.

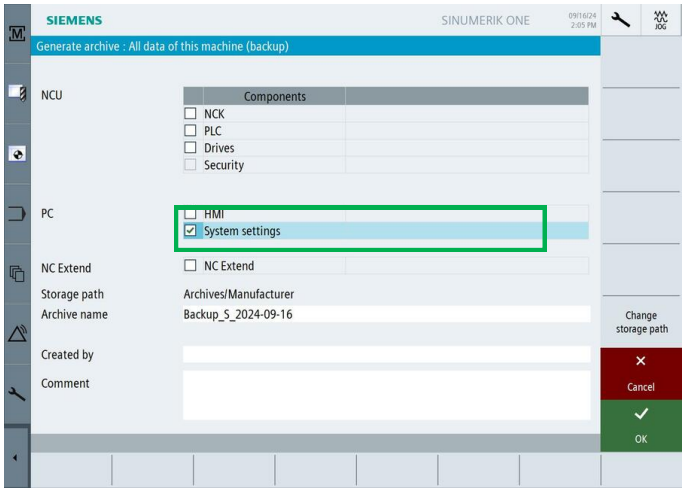
Notes:



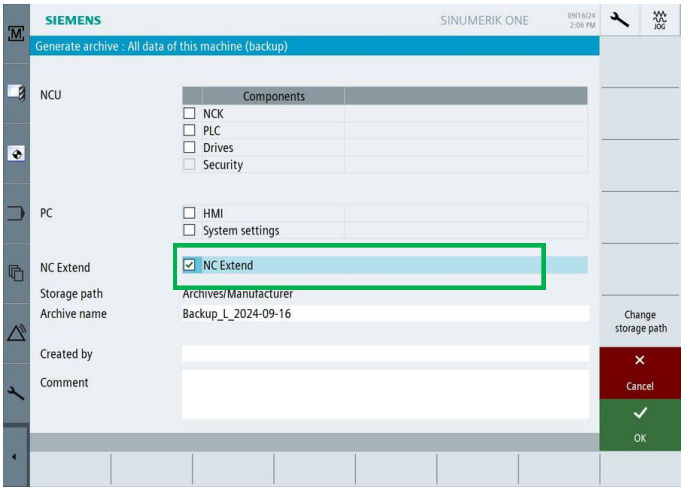
A **Drive** archive may be required when there is a need to test the Drive data in another Virtual machine.



An **HMI** archive may be required when there is a need to test the HMI data in another Virtual machine.



A **System settings archive** would never be required, as it contains Drive configurations, TCU settings, Network settings (for embedded only) mmc.ini.

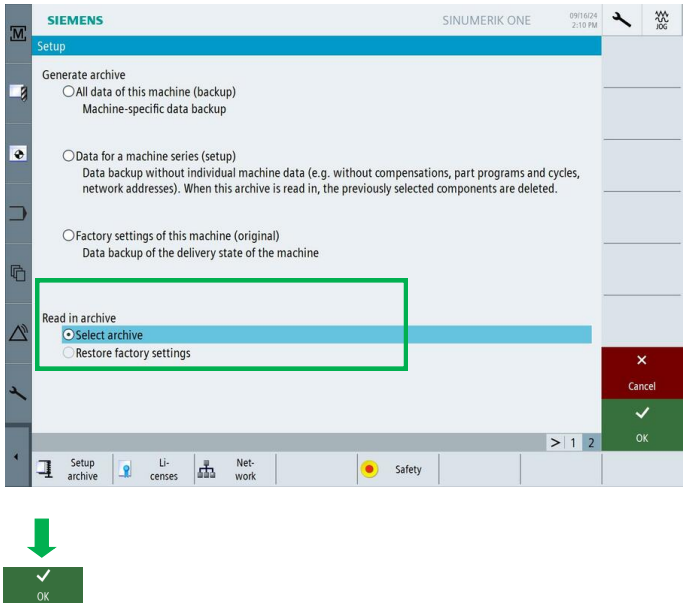


An **NC Extend** archive may be required when there is a need to test the NC Extend data in another Virtual machine.

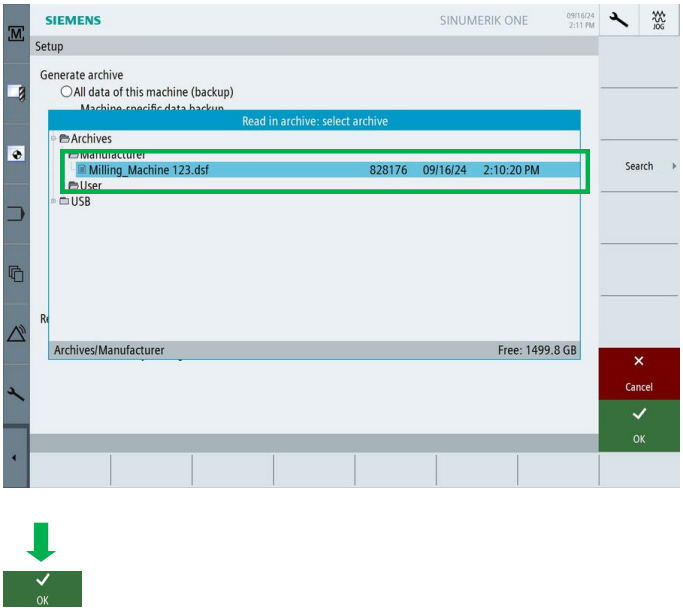
Notes:

Restoring a Series Setup Archive:

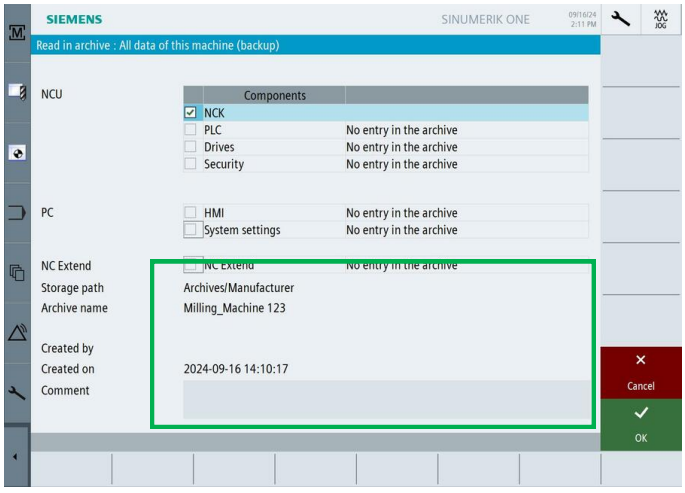
To restore an archive into a virtual machine the option ”Read in setup archive” needs to be selected



Select the source from which the archive is to read in from.

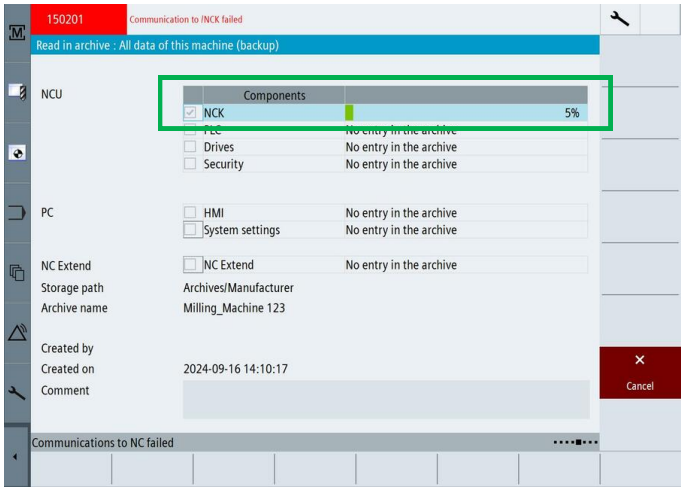


Information about the selected archive is displayed. This includes any comments entered by the creator.





The archive will be read in and a series of status boxes are displayed.

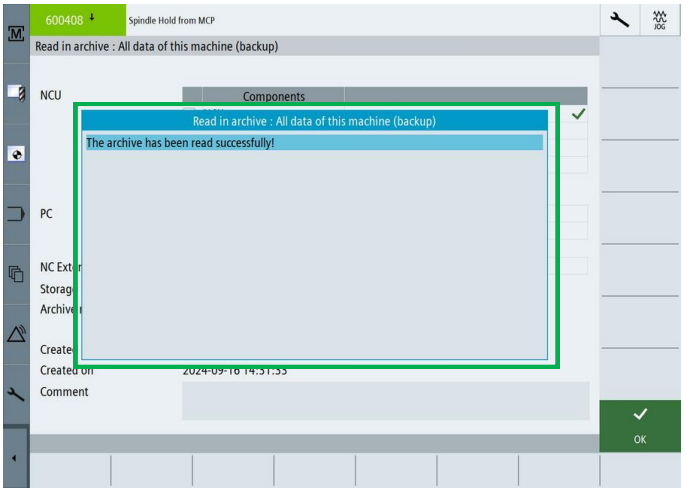




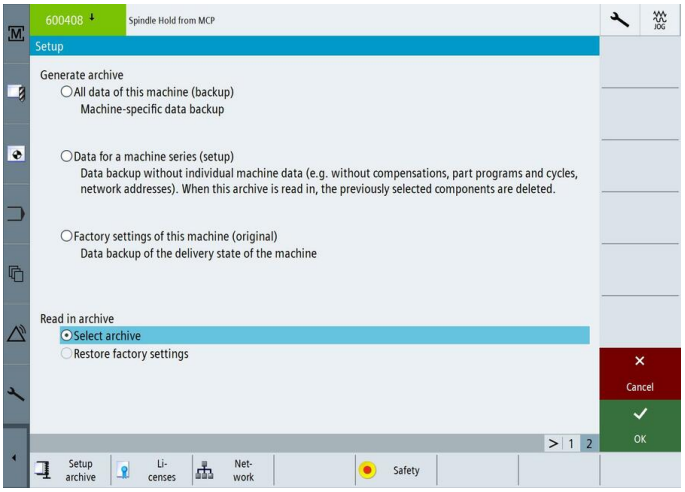


Notes:

A final status box will be presented to show if the archive read -in was successfully or failed.









A136: END

Notes:

A135

SINUMERIK ONE

SINUMERIK ONE Safety Integrated F_PLC + dbSI Configuration

Module Description:

This Module describes the step by step procedures to start-up and commission Sinumerik Safety F_PLC for the Sinumerik ONE.

The procedures described within this module predominantly use TIA Portal & Sinumerik Operate HMI.

The commissioning will start with the Virtual Sinumerik and then progress to the Real Sinumerik.

This Module is valid for >= S/W 06.23.xx.xx

It is expected that the student has a good understanding of the Sinumerik ONE control, including PLC programming, NC & Drive Commissioning prior to using this module.

The commissioning of the Drive Based Safety Integrated Basic/Extended Functions:
STO, SS1, SS2, SOS, SAM, SSM, SLS, SBC, SBT and F-PLC programming via TIA Portal V19 are implemented and, using the respective diagnostics options of HMI-Operate, tested and diagnosed.

Module Description:

After completing the module, you are capable of the following:

- Configuration of the SINUMERIK with F-PLC and the ET200SP with F-I/O
- Commissioning of the Drive Based Safety Integrated Function
- F-PLC program creation using TIA Portal V19
- Diagnostics

Content:

- Introduction
- System Requirements and Information
- Overview: Safety Integrated Function
- OEM Function Table
- Basic Overview: Safety Integrated
- PLC: Enabling Safety Integrated
- PLC: Adding the safety Integrated Hardware
- Safety Integrated Drive Configuration
- F-PLC Logic
- Safe Torque Off - Extended
- Safe Brake Control
- Safe Stop 1 - Extended
- Safe Stop 2 - Safe Operating Stop
- Safe Acceleration Monitoring/Safe Braking Ramp
- Safely Limited Speeds
- Safe Speed Monitoring
- Safe Brake Test
- Test Stop Routine

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Introduction
System requirements and Information
Overview: Safety Integrated Functions
OEM Function Table
Overview: Sinumerik ONE Safety Integrated
PLC: Enabling Safety Integrated
PLC: Adding the Safety Integrated Hardware
Safety Integrated Drive Configuration
F-PLC Logic
Safe Torque OFF - Extended
Safe Brake Management
Safe Stop 1 Extended
Safe Stop 2/ Safe Operating Stop
Safe Acceleration Monitoring/ Safe Braking Ramp
Safety Limited Speeds
Safe Speed Monitoring
Safe Brake Test
Test Stop Routine
Appendix Safety Integrated Hardware Configuration
End of the Module

Introduction:

SINUMERIK Safety Integrated is a comprehensive safety package that helps to protect both personnel as well as machines. This is extremely efficient and cost effective thanks to the safety functions that are completely integrated into the control and drive technology.

The Safety Integrated monitoring limits are set to operate outside/above the NC monitoring limits - Safety Integrated alarms and messages will generally only be seen if the NC limits are exceeded.

The machine can be safely and practically operated under all of the specified operating conditions. For instance, in the setting-up and test modes with the protective door open.

The safety functions comply with the requirements laid down in DIN EN 61508 and DIN EN ISO 13849.
As a consequence, the essential requirements relating to functional safety can be simply and cost-effectively implemented.

The functional scope includes, e.g.:

- Functions to safely monitor speed and standstill.
- Functions to safely differentiate between working and protective zones and to identify the relevant areas (e.g. Safe Cams).
- All safety relevant signals are directly connected and internally logically combined.

Product features

- Functions to protect persons and machines.
- Short response times via cyclic interrupt programming.
- Can be used in all operating modes.
- Replace external electromechanical devices.

Customer benefits

- High degree of safety: Seamless implementation of safety functions in SIL (Safe Integrity Level - EN62061) / PL (Performance Level - EN13849).
- High degree of flexibility: Practical safety and operator concepts can be realized.
- High degree of cost effectiveness: Hardware and installation costs are reduced.
- Electromechanical switching elements that are prone to faults are eliminated due to .

Notes:

System Requirements & Information:



SINUMERIK ONE:

SINUMERIK Safety Integrated is offered for **NCU/PPU 1740/1750/1760** as of **SINUMERIK ONE NC SW V6.23**. The performance of the PLC must be taken into account depending upon on the application.

- NCU 1740 with PLC 1500F 6FC5317-4AA00-0AA0
- NCU 1750 with PLC 1500F 6FC5317-5AA00-0AA0
- NCU 1760 with PLC 1500F 6FC5317-6AA00-0AA0
- PPU 1740 with PLC 1500F 6FC5317-4AA00-1xA0 (x =C or D)

Typical PLC cycle times 1740/1750/1760 15...20ms/10...15ms/5...7ms

(Please refer to the NC63 catalogue for more information)

SINUMERIK software: (general release)

- NCU firmware as of V6.23
- HMI software SINUMERIK OPERATE as of V6.23

Engineering system software:

SINUMERIK Safety Integrated F_PLC + dbSI can only be configured with:

- TIA Portal as of V19 Update 1 (SIMATIC STEP 7 Professional)
- SINUMERIK STEP 7 Toolbox as of V19

Option Description	Option Number	Course Requirement
Safety Integrated /SI Axis/Spindle • Safety Integrated Axis / Spindle Safe motion monitoring function 1 axis / spindle. Prerequisite: Safety Integrated F -PLC	**6FC5800-0BK00-0YB0**	**5**
Safety Integrated /SI Logic • Safety Integrated F-PLC Enables the F-PLC integrated in the SINUMERIK ONE for failsafe programming, contains no axis	**6FC5800-0BS60-0YB0**	✓

Notes:

Overview: Safety Integrated Functions

Safe Functions		Abbreviation	Brief description
Basic Functions	Safe Torque Off	STO	Safe torque off
	Safe Stop 1	SS1	Safe stopping process in accordance with stop category 1
	Safe Brake Control	SBC	Safe brake control
Extended Functions (includes basic functions)	Safe Stop 2	SS2	Safe stopping in accordance with stop category 2 (Not on machine tool equipment)
	Safe Operating Stop	SOS	Safe monitoring of the standstill position
	Safely Limited Speed	SLS	Safe monitoring of the maximum speed
	Safe Speed Monitor	SSM	Safe monitoring of the minimum speed
	Safe Acceleration Monitor	SAM	Safe monitoring of drive acceleration
	Safe Direction	SDI	Safe monitoring of the direction of motion
	Safe Limit Positions	SLP	Safe software limit switches.
	Safe Brake Test	SBT	Safe brake test
	Safe Cams	SCA	Safe software cams

The various safety functions are triggered by the system in response to a safety limit being exceeded or in response to a request from the user safety logic.

Safe Torque Off (STO)

STO is used to achieve a torque free situation for a drive.

Applications/cause

- Always active after the SS1 function (Emergency Stop) has completed.
- Example: manually rotating a spindle with the protective door open

Effects:

- The drive pulses are suppressed
- The power feed is electronically disconnected
- The drive is safely in a no-torque condition

Safe Stop 1 (SS1; Stop B)

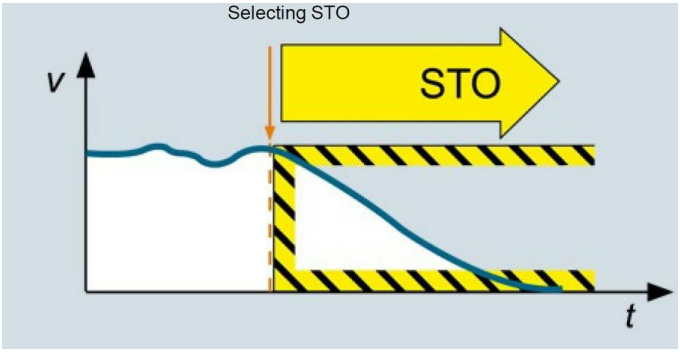
SS1 is a monitored stop of a drive.

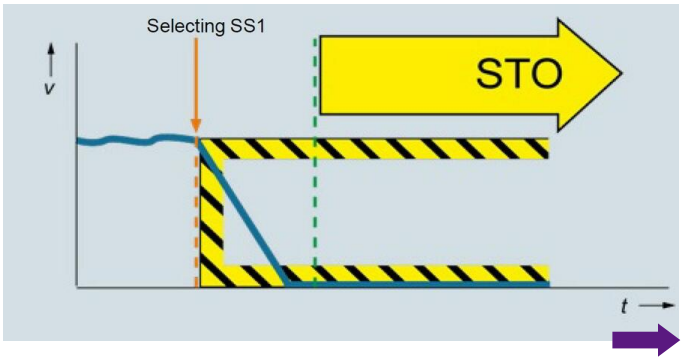
Applications/cause

- Always triggered in response to an Emergency Stop request.
- Can be used in reaction to various Safety Integrated monitoring functions. Example SOS limit exceeded.

Effects:

- The drive is brought to a standstill in accordance with stop category 1 (EN 60204-1)
- Braking along an adjustable ramp (OFF3)
- Transition into STO after an adjustable time





Notes:

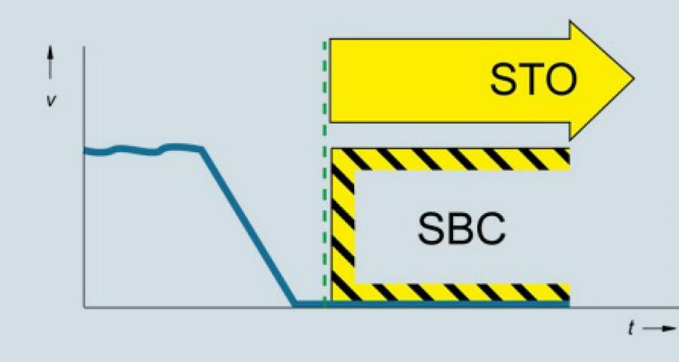
Safe Brake Control (SBC)

Applications/cause:

- Controlling a brake for axes that can fall due to gravity.

Effects:

- Two channel output at the Sinamics Motor Module
- Is initiated in conjunction with STO.
- Brake winding short-circuit and wire breakage are detected.



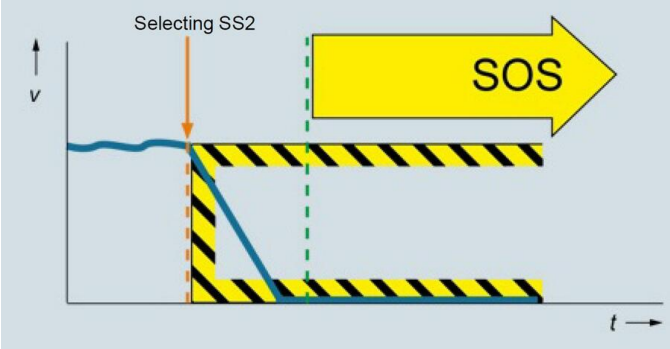
Safe Stop 2 (SS2; Stop C)

Applications/cause:

- Protection of persons

Effects

- The drive is brought to a standstill in accordance with stop category 2 (EN 60204-1)
- Braking along an adjustable ramp (OFF3)
- Transition into SOS after an adjustable time or when a configurable speed is fallen below



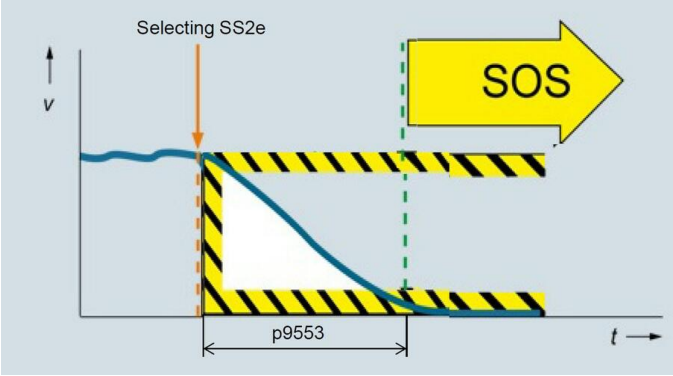
Safe Stop 2 Extended (SS2e; Stop D)

Applications/cause:

- Machine protection.

Effects:

- Braking a drive along a contour.
- During the delay time the speed follows the setpoint provided by the NC control system.
- During the delay time it is not permissible that persons are in the hazardous area.
- Transition into SOS after the delay time has elapsed.



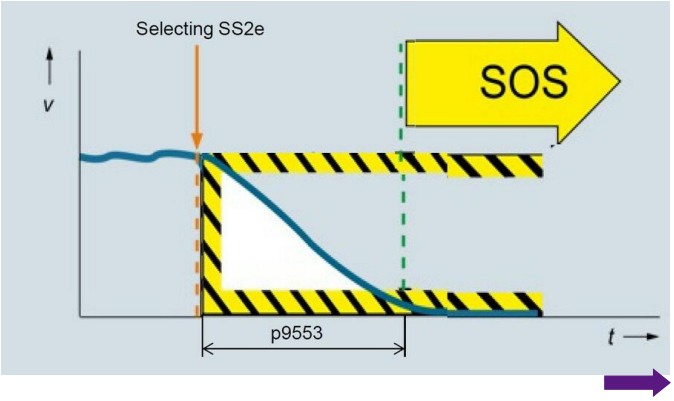
Safe Stop E

Applications/cause:

- Machine/Tool protection.

Effects:

- Retraction motion (ESR).
- Braking a drive path-related.
- During the delay time the speed follows the setpoint provided by the NC control system.
- During the delay time it is not permissible that persons are in the hazardous area.
- Transition into SOS after the delay time has elapsed.



Notes:

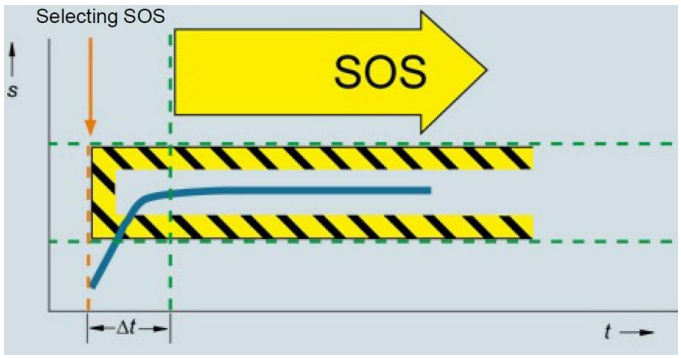
Safe Operating Stop (SOS)

Applications/cause:

- Working in a hazard area with protective doors open
- Axes with rotating tools are held in position
- Axes that can fall due to gravity are maintained at a certain position.

Effects:

- The drive remains in position control
- It is monitored that the drive remains at standstill
- The drive can develop full torque



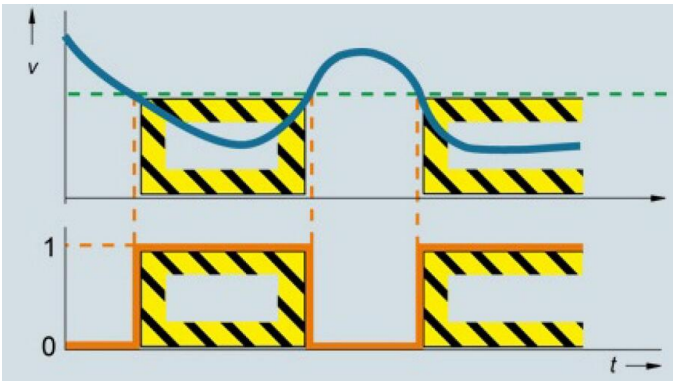
Safe Speed Monitor (SSM; n<nx)

Applications/cause:

- The protective door is only released after all of the drives have come to a standstill (zero speed)

Effects:

- SSM is a signaling function
- It outputs a safety-related output signal once the speed limit value has been reached
- Processing the safe output signal in the F-PLC safe logic



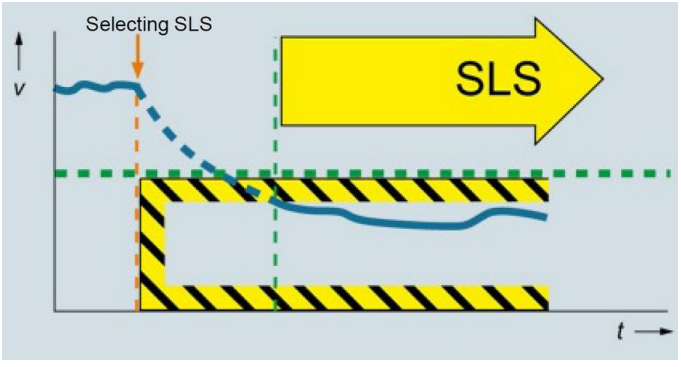
Safe Limited Speed (SLS)

Applications/cause:

- Traversing axes or spindles with the protective door open > Protection for persons
- Traversing axes or spindles with the protective door closed > Machine protection

Effects:

- Safety-related monitoring of the load-side speed
- 4 different speed values per drive can be parameterised



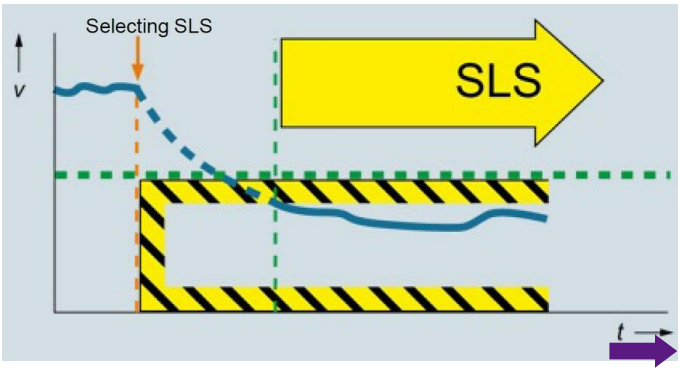
Safe Limited Speed (SLS) sensorless

Applications/cause:

- Monitoring of high speed spindles > Protection for persons and machine

Sensorless means: The actual speed values are calculated from the measured electrical actual values..This is the case if no encoder or no safety capable encoder is used.

- For induction motors
- Safety-related monitoring of the load-side speed
- 4 different speed values per drive can be parameterized. SLS override possible



Notes:

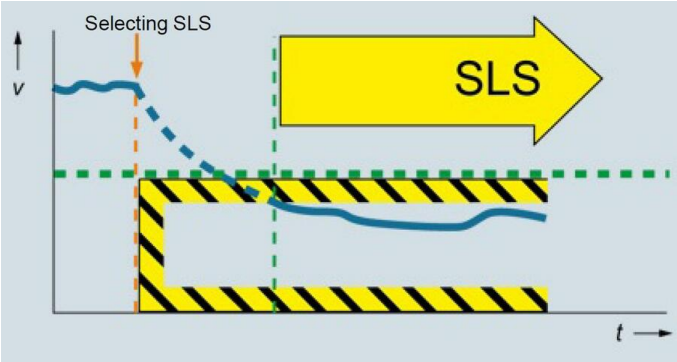
Safe Limited Speed (SLS) via PROFIsafe

Applications/cause:

- As per SLS with variation of SLS1 possible
- Varying diameters, e.g. for grinding wheels or lathe chucks, resulting in varying peripheral surface speeds.

Effects:

- The first SLS limit value (SLS1) can be influenced via PROFIsafe
- Value range 1…...32767 (32767 = 100%)
- Also in this case, speed levels 2, 3 and 4 can be parameterised and selected as normal.



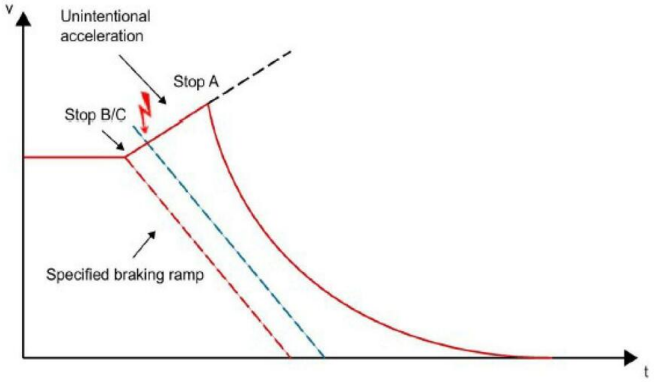
Safe Acceleration Monitor (SAM)

Applications/cause:

- Monitoring the braking operation

Effects:

- Active for SS1 and SS2
- The function monitors whether the speed increases while braking.
- STO is immediately triggered once the monitoring responds



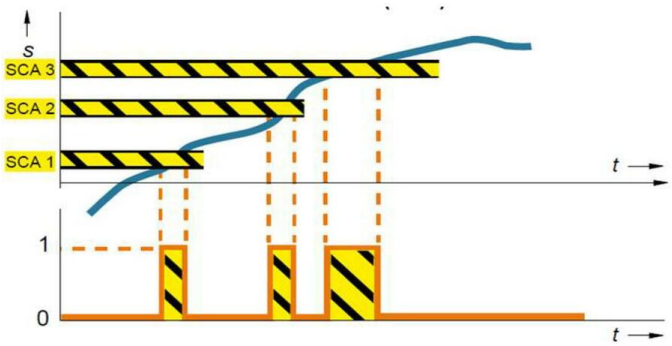
Safe Cam (SCA)

Applications/causes

- Definition of working and protection areas.
- Safety-related switchover of safety functions.

Effects:

- Outputs a safety related-signal
- The safety-related signal is processed in the fail safe program
- 30 safe cam pairs (plus/minus) are available for each axis



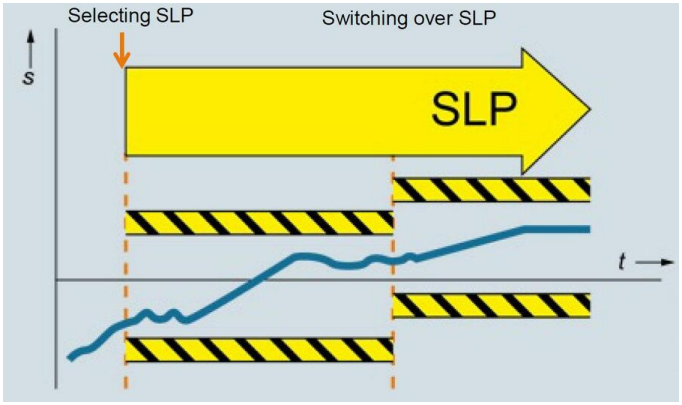
Safe Limited Position (SLP)

Applications/causes:

- Traversing range limits for axes.

Effects:

- Two traversing range limits can be parameterised
- It is possible to switch between the two limits
- Adjustable stop response
- Substitute/replacement for hardware limit switches



Notes:

Safe Position (SP)

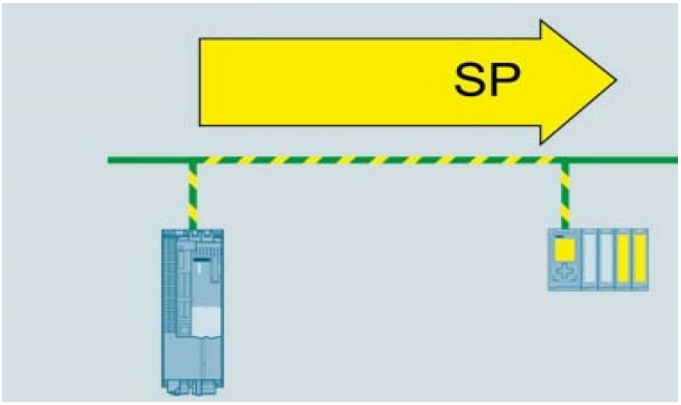
Applications

- Implementing more than 30 safety-related cams
- Calculating the actual speed

Effects

- Transferring safe positions to the fail-safe control system
- Processing the position in the fail-safe program

Note: The position actual value is not transferred in PROFIsafe telegram 903, which is preset for SINUMERIK ONE. (requirement is telegram 901 or 902)



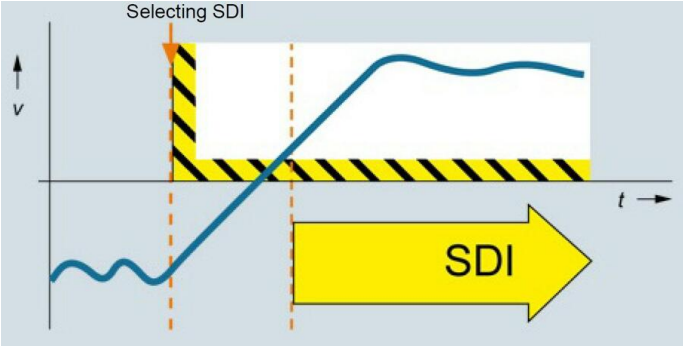
Safe Direction (SDI)

Applications

- Expensive tools, which may only be used for machining in one direction of rotation
- Traversing from an end position in the appropriate direction

Effects

- Using fail-safe signals, the positive or negative direction can be inhibited



Notes:

STOP types:

If a safety function is triggered, a STOP is issued. The STOP type depends on the safety function that was triggered.
It is possible, in some cases, to configure the STOP type.

Stop type	Action	Effect	Initiated in response to	Active Safety Function	Alarm clearance
STOP A	Pulses are immediately cancelled	Drive coasts down	SAM/SLS	STO	Power off/on
STOP B	Setpoint zero is immediately set and timer t_B started. When $t_B = 0$ or $n_{act} < n_{shutdown}$, STOP A is activated.	The drive brakes along the OFF3 ramp - transition into STOP A	SS1/SOS/ SLS	STO	Power off/on
STOP C	Setpoint zero is immediately set and timer t_C is started. When $t_C = 0$, SOS is activated.	The drive brakes along the OFF3 ramp, SOS active.	SLS/SLP	SOS	Reset
STOP D	Motor brakes along the acceleration path and timer t_D is started. When $t_D = 0$, SOS is activated.	Drive brakes as part of a group along the accelera-tion path. SOS active	SLS/SLP	SOS	Reset
STOP D Dynamic	* NCK changes the dynamics * so that the axes can be braked interpo-lating within the transitional time STOP D -> SOS	Drive brakes as part of a group along the accelera-tion path. SOS active	SS2E/SLS/ SLP	SOS	Reset
STOP E	Results in stopping and retraction, plus timer t_E is started. When $t_E = 0$, SOS is activated.	Drive brakes along the programmed retraction and stopping paths. (ESR) SOS is active	SLS/SLP	SOS	Reset

Safety Integrated tasks are illustrated in the following table:

What should be monitored	Which Safety Integrated input signal activates this monitoring?	How can Safety Integrated react when a violation of the tolerance occurs
The axis/spindle must not move	The Safe Operational Stop (SOS) state is selected via two channels in the F-PLC.	If the axes/spindle moves during SOS. Stop A/B is triggered in both channels.
The Axes/spindle may move, but not faster than the specified value	Safely Limited Speed (SLS) is selected via two channels in the F-PLC.	If the SLS limit is exceeded, the response can be configured. For Sinumerik ONE, Stop A or Stop B should be chosen as the response. If Stop C is selected, it is possible that after an alarm acknowl-edgement, the Axes/spindle jerks because a following error still has to be corrected

Identifying machine/equipment movements:

If Safety Integrated is to be used on a machine in order to estimate the danger caused by moving parts, all possible dangerous movements should be listed.

Not only do NC axes/spindles move, but also components such as the tool changer or chip conveyor.

The list might appear as follows:

- Axes X, Y, Z and spindles
- Tool changer or turret
- Tool magazine and tool holder
- Chip conveyor

Why is the list important?
The extended functions provide ***SOS*** and ***SLS*** for the moni-toring of the numerically controlled axes/spindles.
The other movements that are not CNC controlled can be safely shut down, either with safety relays or with contactors that are controlled from the F-PLC.
These outputs can be configured to react to different safety states.

Notes:

OEM Function Table

The function table specifies which safety function should be active for certain machine states.
A function table should be created before starting the configuration.

The risk assessment of a particular machine/machine series is used determine the function table and associated limits.

The function table is included later in the acceptance report, which is when the functionality of the last two columns are tested.

Example function table:

Operating mode	Drive	Which status should be active?
Doors closed	All	**SLS4**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are energised.
Doors open and mode switch at **Auto**. (***Production*** on test simulator)	X, Y, Z, A axes	**SOS**
	Spindle	**STO**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are not energised.
Doors open, mode switch at **Auto** (***Production*** on test simulator) and ***Hold to run*** switch pressed.	X, Y, Z, A axes	**SOS**
	Spindle	**STO**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are not energised.
Doors open and mode switch at ***Set-up***	X, Y, Z, A Axes	**SOS**
	Spindle	**STO**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are not energised.
Doors open, mode switch at ***Set-up*** and ***Hold to run*** switch pressed.	X, Y, Z, A axes	**SLS1**
	Spindle	**STO**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are not energised.

The following table shows example SLS speeds. The SLS speeds are defined by the OEM and will vary between machine types.

	Set-up/door open and hold to run		Auto (***Production*** on test simulator) / door closed	
	Maximum Feed-rate	**SLS1**	Maximum Feed-rate	**SLS4**
X, Y, Z (mm/min)	2000	**2500**	10000	**12500**
A (rpm)	10	**12.5**	100	**125**
Spindle (rpm)			1000	**1250**

Feed-rate/SLS Relationship

Setpoint Limitation is used to limit the feed-rate of an axis to a percentage of the active Safe Limited Speed. Recommended value is 80%, therefore maximum federate is 80% of SLS speed

Selection of required extended functions:

It is clear, from the function table, which safety extended functions are used.
These are **SOS, SLS1** and **SLS4** (for axes X, Y , Z and A) , and **STO, SLS4** (for the spindle)
In addition, **SS1** is used for emergency stop on all axes/spindle.

Notes:

Overview: SINUMERIK ONE Safety Integrated:

As of SINUMERIK ONE NC software version V6.13, **SINUMERIK Safety Integrated F_PLC + dbSI** can be used. This means that, amongst other things

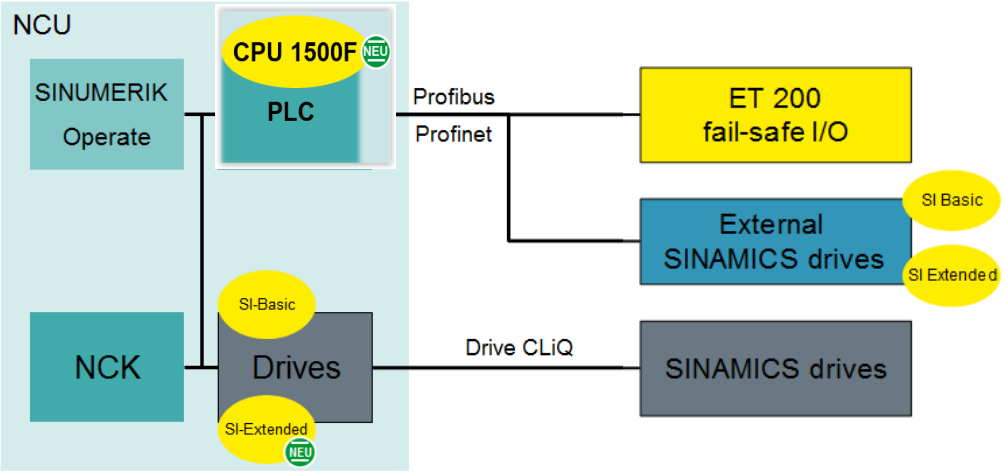
- The motion monitoring is performed exclusively through **drive-based safety** with motion control functionality in the SINAMICS.
- This concept corresponds to the motion monitoring for the SINUMERIK 828.
- The same safety concept is used for axes that are **assigned directly to the NCU** and for **external axes**.

- SI-Basic = Basic Safety Functions of the SINAMICS (STO, SBC, SS1)
- SI-Extended = Extended Safety Functions of the SINAMICS

Licensing

- **SINAMICS drives**
- **External SINAMICS drives**

see Licensing of axes/spindles and F-PLC



Safety functions of SINUMERIK ONE:

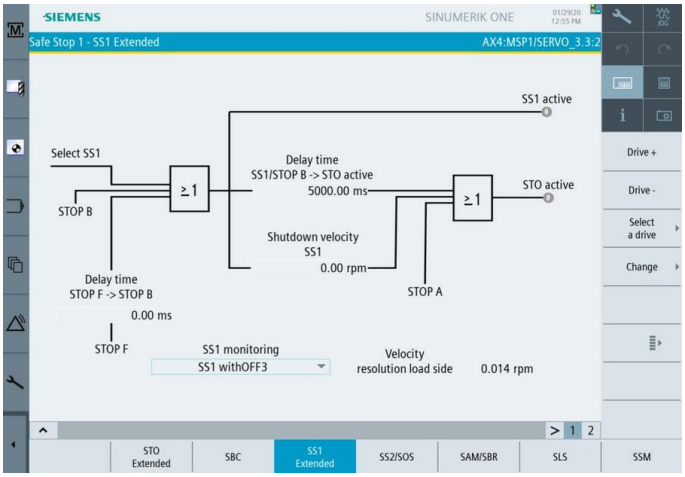
There are various safety features that are not available on the SINUMERIK ONE virtual twin. These include:

- 2nd channel monitoring (only operation with one encoder systems is possible).
- Checksums.
- Safety Integrated Acceptance Test (SIAT).
- Safety Telegrams are only simulated.
- The Safety Integrated drive alarms are not generated.

Notes:

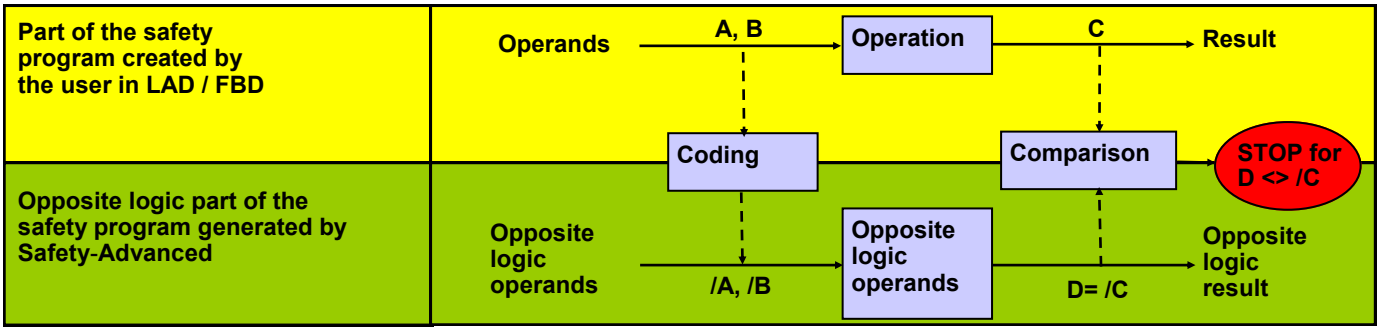
Characteristics of SINUMERIK Safety Integrated with integrated F-PLC:

- In the integrated F-PLC, both the ***non-safety-relevant cyclical Step 7 program,*** and also the ***failsafe F-PLC program*** are processed.
- The safety functions in SINAMICS are activated by PROFIsafe telegrams. This means, with regard to the motion control, the same safety concept for axes assigned directly to the NCK, as well as for the axes coupled externally via PROFIBUS/PROFINET, is applicable.
- The ***Safety Integrated Acceptance Test (SIAT),*** to be performed prior to the delivery is available in SINUMERIK Operate.
- Certified SIMATIC F-blocks are available, for example, for emergency stop.
- Depending on the PLC type, SINUMERIK Safety Integrated allows as many as 32 secure connections to be implemented.
- SINUMERIK Safety Integrated provides new functions. Safe rotational direction, e.g. for tools that can machine in just one direction, encoderless safe speed monitoring, e.g. for accelerating spindles, and safe position, e.g. for a large number of safe cams.
- For SINUMERIK Safety Integrated, an easily understandable screen form for the graphically supported commissioning in which the commissioning engineer can enter the individual parameters is available for each safety function. These screen forms largely correspond to the Safety Integrated screen forms of the STARTER software for commissioning external SINAMICS drives.

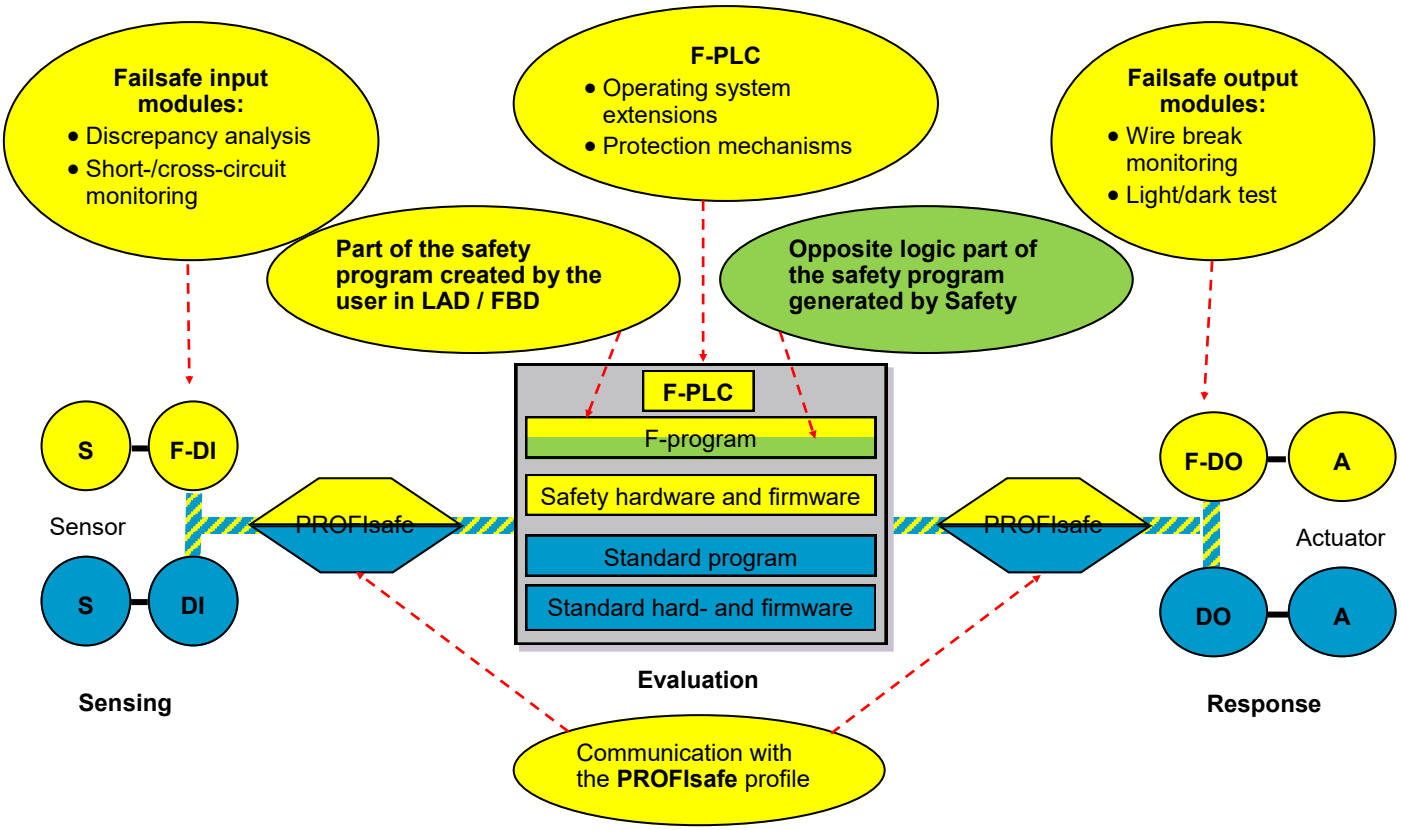


Safety concept: (see following representation)

- ***Hardware and firmware extensions:***
  The safety-oriented F-PLC combines the functionality of a standard PLC with that of a safety PLC. The operating system extended with protective mechanisms allows standard and safety-oriented user programs to be executed on a PLC.
- ***F-program (Called in Cyclic Interrupt OB)***:
  The safety program for controlling safety oriented functions consists partly of the user created logic in FBD or LAD and partly of Safety Advanced generated logic, with opposite logic operands/operations.
  The standard and safety-oriented user programs are created in the same programming environment.
  TÜV-certified function blocks for all common safety functions further simplify the programming.



Notes:



Safety-relevant processing with Sinumerik ONE components:

Safety-oriented internal and external signals in the safety-oriented user program (safety program) of the F-PLC are processed in SINUMERIK Safety Integrated (F-PLC) mode:

- The F-I/O are connected to the F-PLC and controlled via PROFIsafe:
  The safety functions are contained principally in the safety program (F-PLC) and in the fail-safe inputs and outputs (F-I/O).

- Safety functions integrated in the drive are also controlled by PROFIsafe:
  For this purpose, the PROFIsafe telegrams (SIEMENS telegram 903) are transferred on the DP Integrated interface from the PLC to the SINAMICS Integrated (DRIVE).

Status and control information is transferred with SIC/SCC (Safety Info Channel / Safety Control Channel - Telegram 701) between integrated subcomponents of the Sinumerik ONE (PLC, NCK and SINAMICS Integrated).

PROFIsafe communication via Sinumerik ONE subcomponents.
PROFIsafe communication is only possible between F-components. With the SINUMERIK ONE, PROFIsafe communication is performed between the following components:

- F-PLC as PROFIsafe host
- SINAMICS Integrated (DRIVE) as PROFIsafe slave/device
- F-I/O as PROFIsafe slave/device

Notes:

PLC: Enabling Safety Integrated

Equipment Information

A standard milling machine with 3 linear axes (X, Y, Z), 1 rotary axis (A) and 1 spindle has been pre-configured.

No Safety Integrated equipment is present.

A separate addon simulator is available which contains the required Safety Integrated equipment.

This will be added and configured.

The Safety Integrated equipment to be added to the PLC hardware configuration:

- IM155-6PN Interface Module x 1
- F-DI 8 x 24V DC Failsafe input module x 3
- F-DQ 4 x 24V DC Failsafe output module x 2
- DI 8 x 24V DC Standard input module x 1
- F-PM-E 24V DC Failsafe power module x 1
- DQ 8X24V DC Standard output module x 1
- Server Module x 1

Various buttons and switches, including a locking door switch, are connected to the PLC hardware to allow the simulation of various machine states.

Enable the Safety Integrated Options

The following Safety Integrated options must be set.



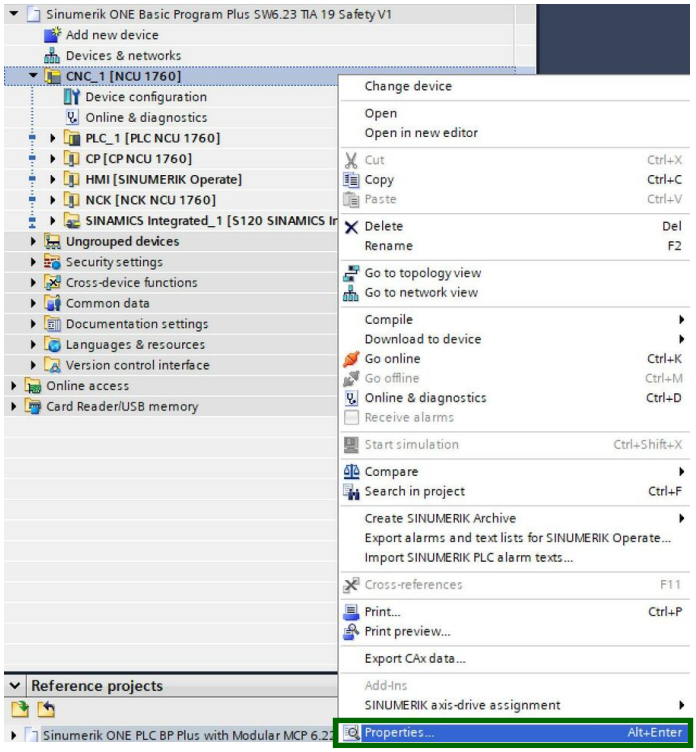


Setting the Safety Integrated Mode

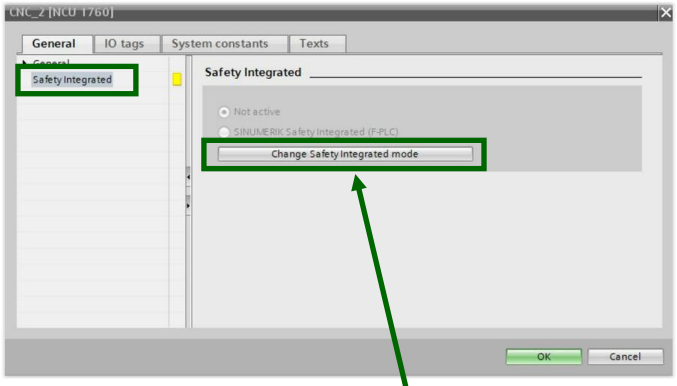
The first modification is to change the Safety Integrated mode

The safety integrated mode is then enabled in the NCU.

Right click on the NCU and select ***properties:***

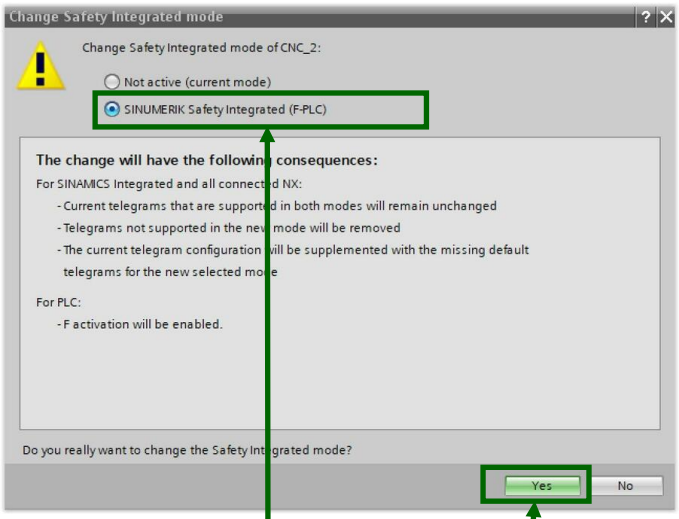


In the ***General/Safety Integrated*** area select ***Change Safety Integrated mode:***

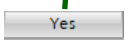


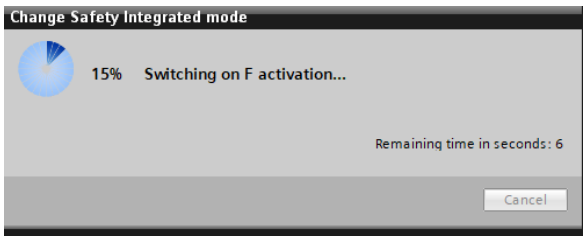
Select ***Change Safety Integrated mode.***

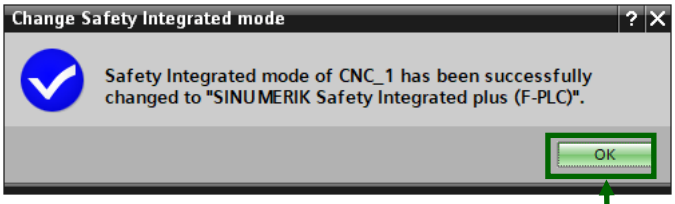
Notes:

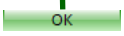


Select ***Sinumerik Safety Integrated (F-PLC)***

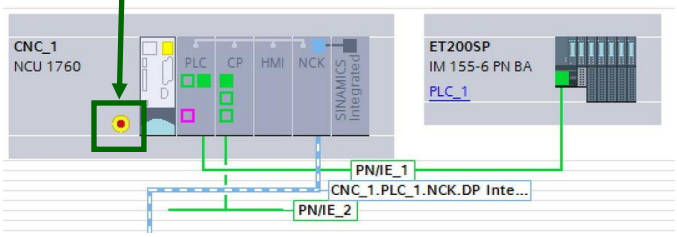


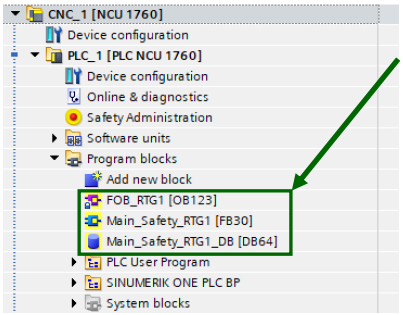






The Emergency Stop symbol indicates that Safety Integrated is enabled.



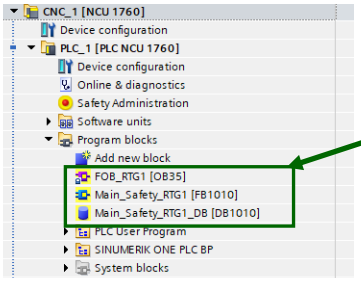


The blocks of runtime group 1 are created automatically. In the Safety Administration it is also possible to ***Create a new F-runtime group*** if for example a second group was required that runs at a faster/slower priority class. ***The maximum number of F-runtime groups is 2.***

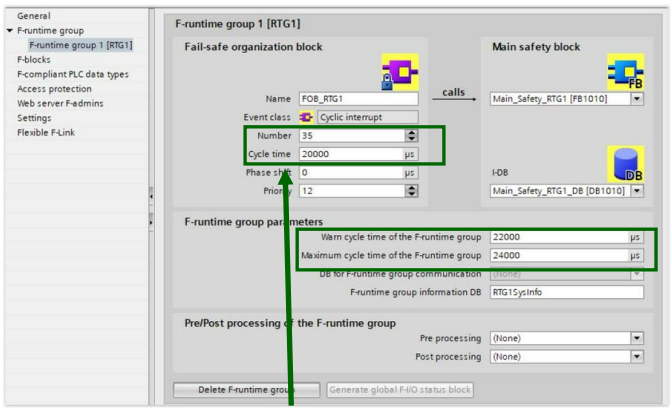
- Calling block: FOB_RTG1 [OB123]
- Main safety block: Main_Safety_RTG1 e.g.[FB30]
- I-DB for the main safety block: Main_Safety_RTG1_DB e.g.[DB64]

The block numbers will depend upon which block numbers have already been used in the project. It is therefore better to renumber them as FOB_RTG1 [OB35], Main_Safety_RTG1 [FB1010] and Main_Safety_RTG1_DB [DB1010].
This can be done in **Safety Administration** for the OB and in block properties (right click on block) for the FB and DB. (Reserved area F-FB/FC/DB 1010...1019).

Also in the properties of FB1010 the language should be changed from FBD to LAD for the F-PLC logic used later.



The FB/DB blocks of runtime group 1 are renumbered, by right clicking and then selecting ***Properties.***

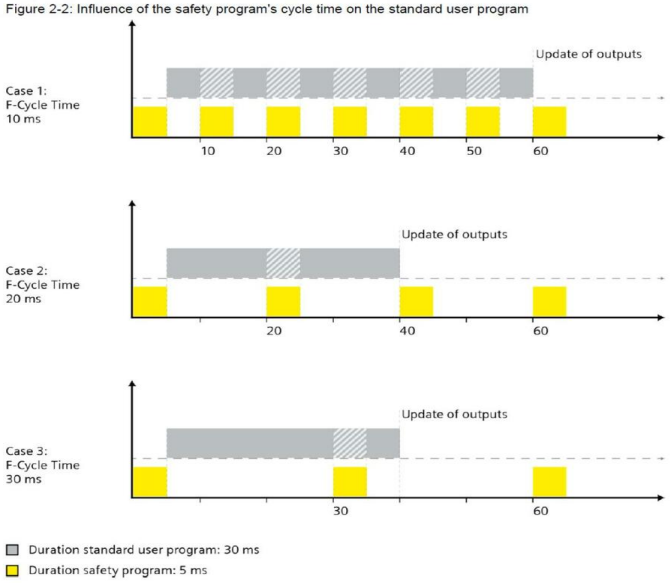


The block number for the **Failsafe OB** can be changed in the drop down box and the cycle time and warning cycle times should also be reduced. The default for OB35 is 100ms, this should be reduced depending upon the application to around 20ms and warnings set to 22ms and 24ms.(+10%/+20%)

Notes:

The following figure shows the influence of the safety program's cycle time on the time that is available for processing the standard user program.

Figure 2-2: Influence of the safety program's cycle time on the standard user program



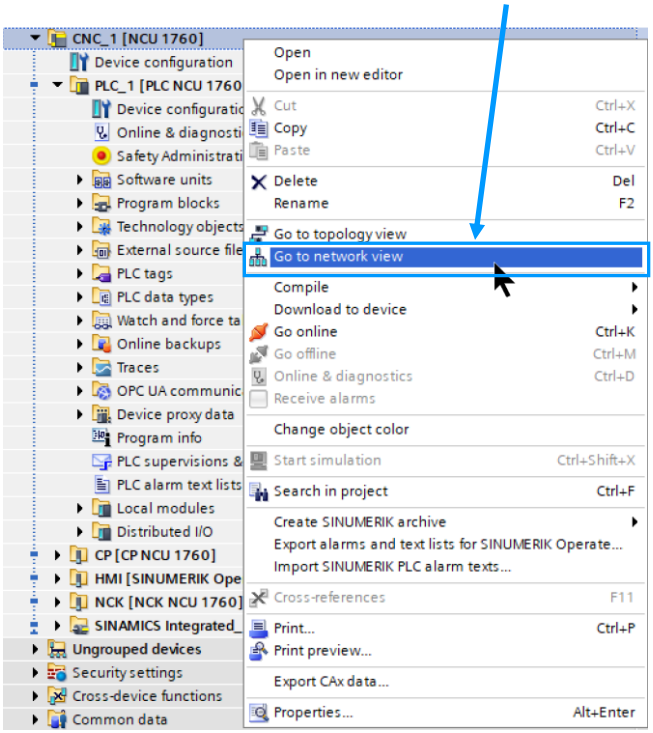
Note Please note that higher-priority organization blocks (e.g., cyclic interrupt OBs or motion control OBs) can interrupt the safety program in the same way as shown in Figure 2-2.

To make sure that the safety program cannot be interrupted, you can customize the priorities in the properties of the appropriate OBs.

NOTICE The cycle time must be longer than the execution duration of the safety program.

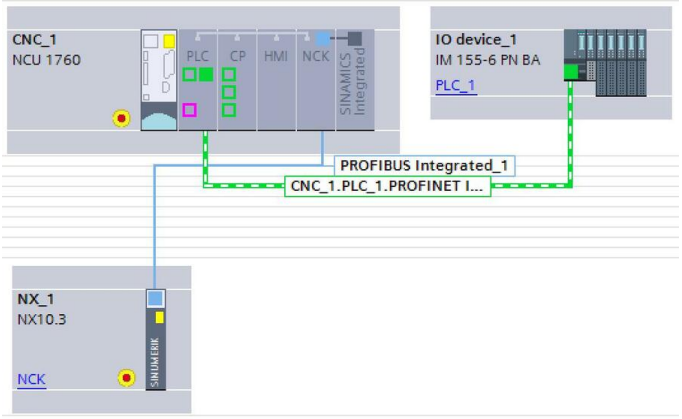
Adding the Safety Integrated Hardware

Open the project and select ***Project view.***
Right click on the NCU 1760 and select ***Go to network view***.

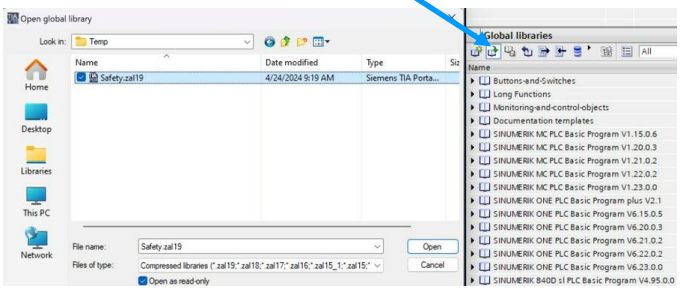


Network view

The current hardware configuration contains the NCU, an ET200SP and an NX module.

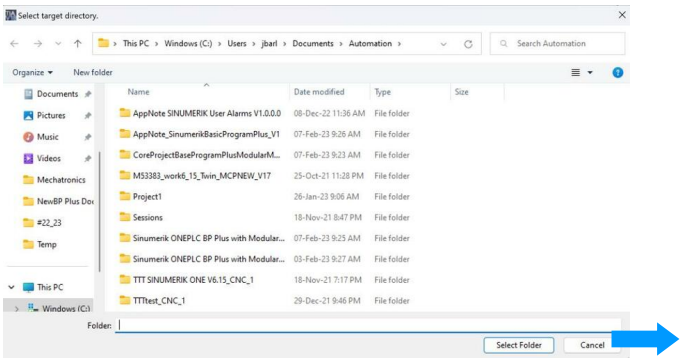


Now select “Open global library”



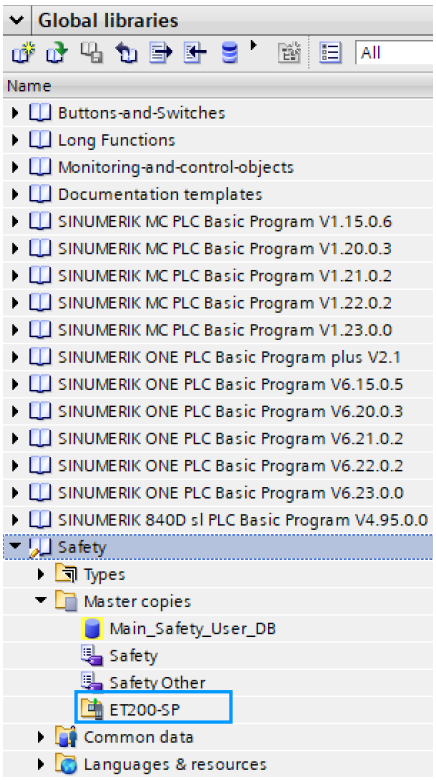
Navigate to the compressed library .zal19.

Select “Open” and then “Select Folder”.

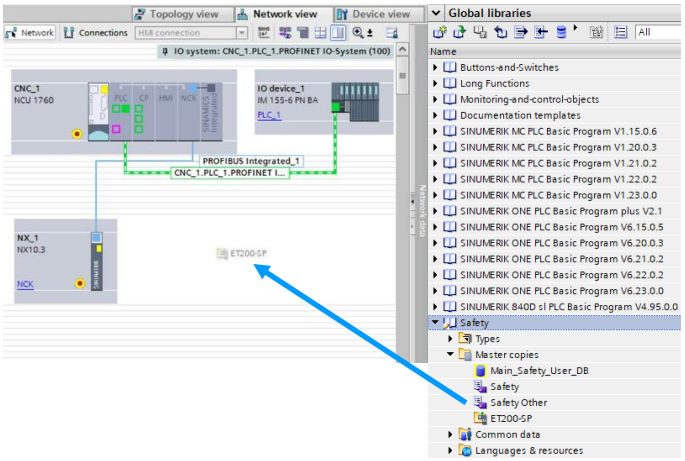


Notes:

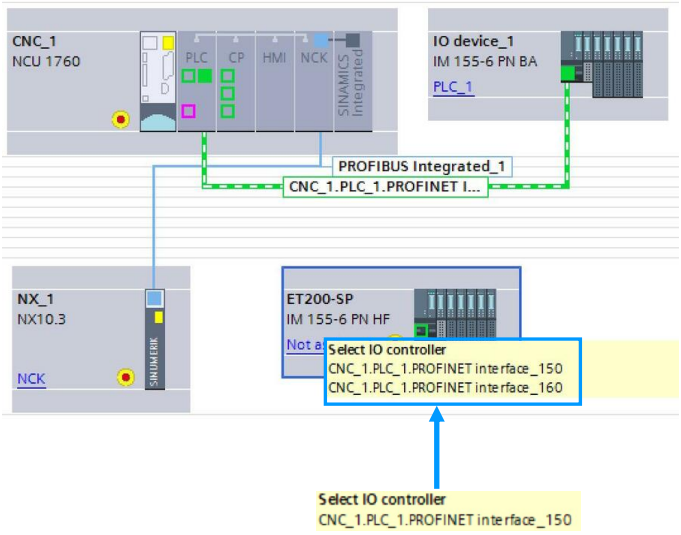
The Safety Global library is now shown with the ET200-SP safety hardware.

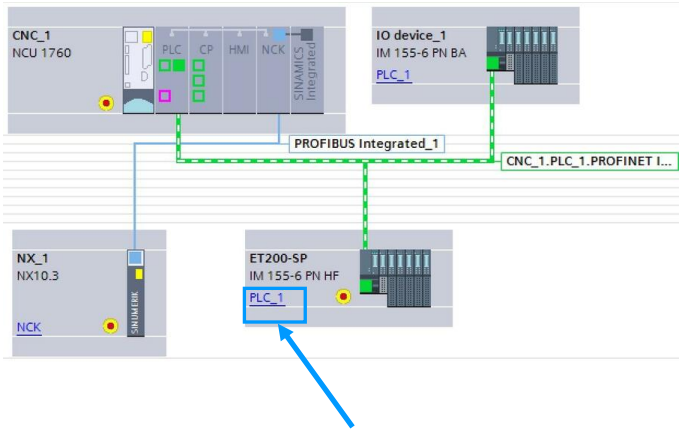


The safety hardware can now be copied into the project using “Drag and Drop”.



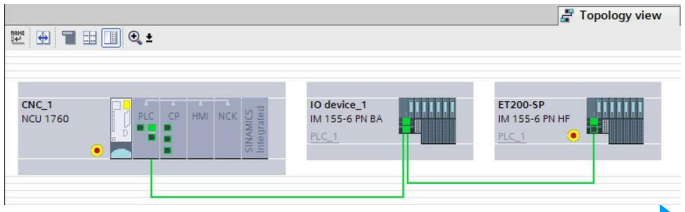
The device needs to be assigned to the CNC Profinet interface:





The ET200-SP hardware is now assigned to the PLC

The modules can also be connected in the Topology view taking into account the correct connections e.g Port_1[X1.P1] of ET200-SP to Port_2[X1.P2] of IO device_1.

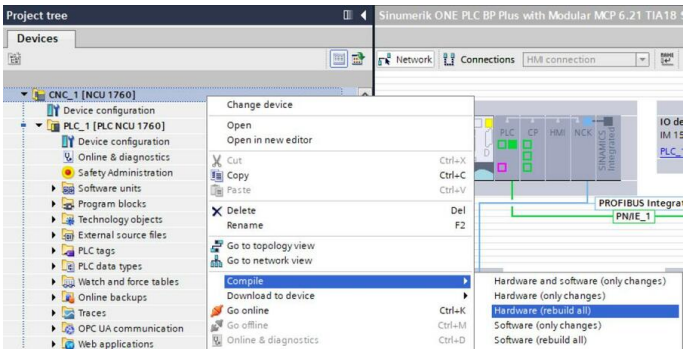


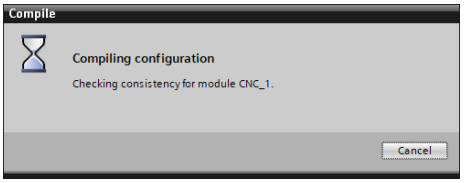
Notes:

Compiling the project:

The project needs to be compiled and downloaded.

The hardware should be compiled first ***(Rebuild all)***



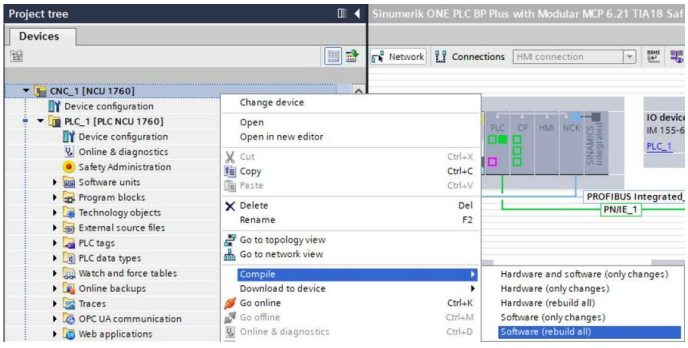


There should be no errors and two warnings.

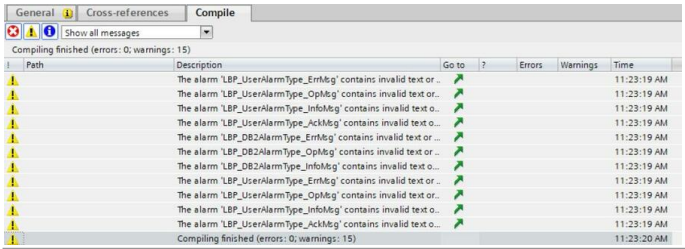
The warnings shown are expected will be dealt with later.



Then the Software should then be compiled ***(Rebuild all)***



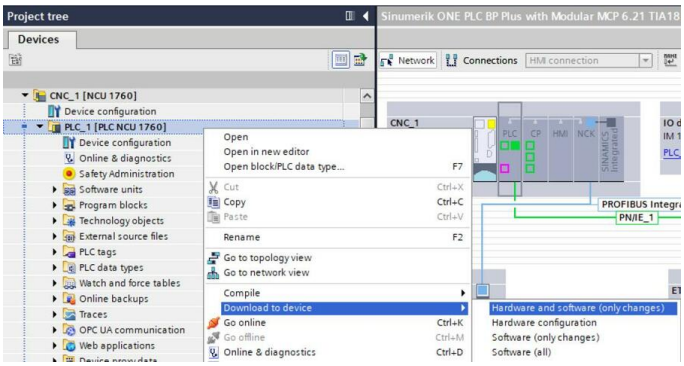
There will be some warnings but no errors



Downloading the project:

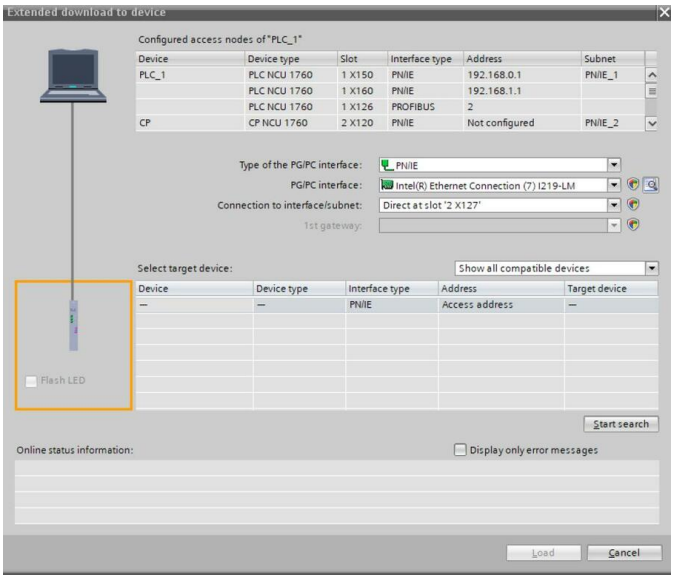
The project can now be downloaded to the virtual machine.

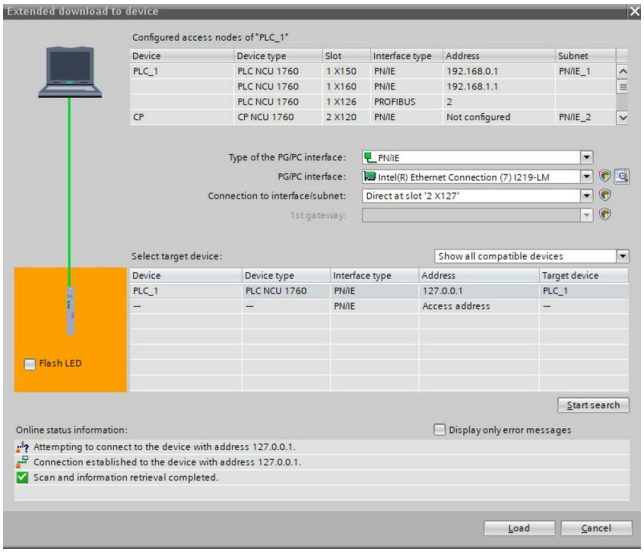
Note: prior to downloading to the Virtual machine the PLC needs to be put into *Stop* mode. This is to prevent a *PLC-SimAdv* error.



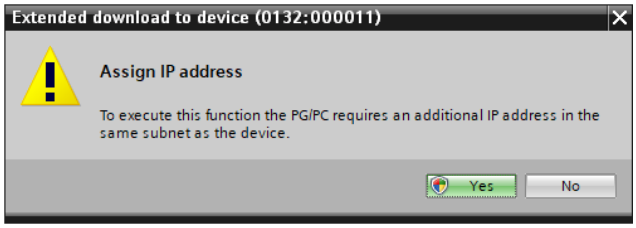
Notes:

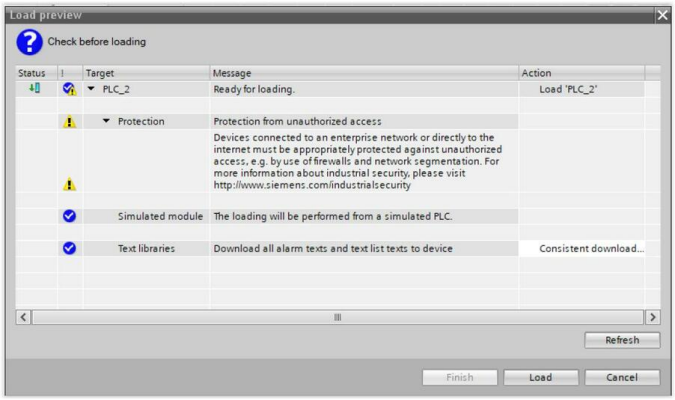
The PG/PC interface setting must be checked for the correct adaptor etc.

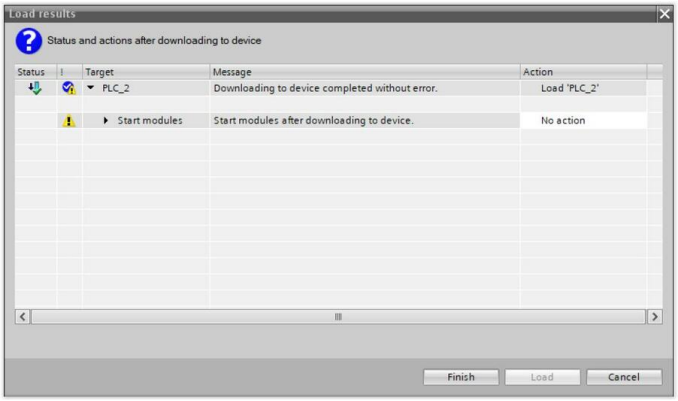




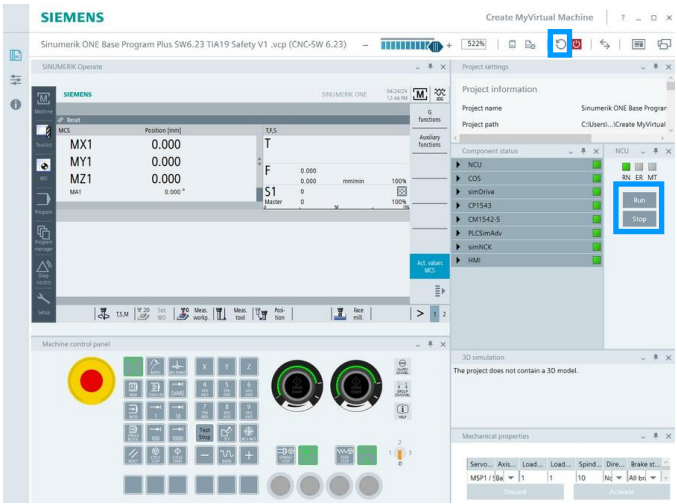
NOTE: if downloading to a REAL machine (X127) 192.168.215.1







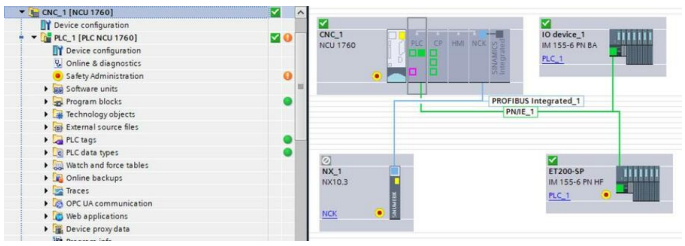
On the virtual machine reset the simulation and then set the PLC to Run and the virtual machine will now have a running PLC with safety hardware but no F-PLC logic



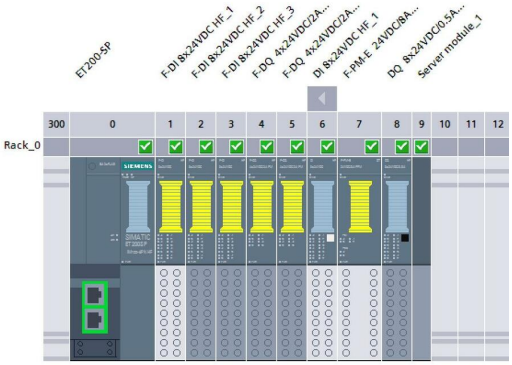
Notes:

Going online in TIA now shows the new hardware loaded and running. The warning in Safety Administration is because no password has been specified for modifying safety.

The safety runtime group is loaded but at the moment there is no F-PLC logic.



If you double click on the ET200-SP device you can see that all of the safety modules are running correctly.



Expected alarms: (Not on Virtual ONE)

Expected alarms are seen below

201654 ↓ Axis MA1 SERVO_3.3:6 (6): SI P1 (CU): Deviating PROFIsafe configuration. 1.

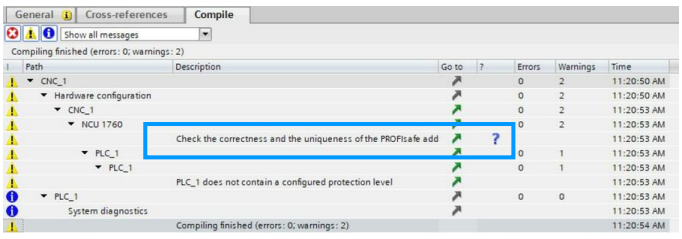
Alarms

Date	Delete	Number	Text
01/01/12 2:05:49.394 AM		201654	Axis MA1 SERVO_3.3:6 (6): SI P1 (CU): Deviating PROFIsafe configuration. 1.
01/01/12 2:05:49.378 AM		201654	Axis MZ1 SERVO_3.3:5 (5): SI P1 (CU): Deviating PROFIsafe configuration. 1.
01/01/12 2:05:49.360 AM		201654	Axis MY1 SERVO_3.3:4 (4): SI P1 (CU): Deviating PROFIsafe configuration. 1.
01/01/12 2:05:49.344 AM		201654	Axis MX1 SERVO_3.3:3 (3): SI P1 (CU): Deviating PROFIsafe configuration. 1.
01/01/12 2:05:49.332 AM		201654	Axis MSP1 SERVO_3.3:2 (2): SI P1 (CU): Deviating PROFIsafe configuration. 1.
01/01/12 2:05:46.416 AM		2130	Note: The standard password is still active for at lea
---		25022	Axis MA1 encoder 1 warning 0

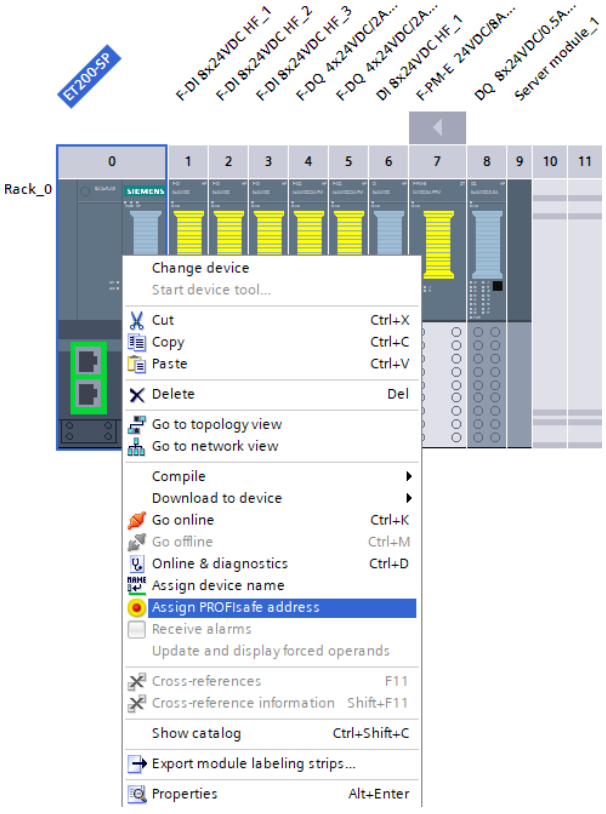
Assigning Profisafe Addresses (not on Virtual ONE)
because the hardware does not exist.

The Profisafe addresses can now be assigned.

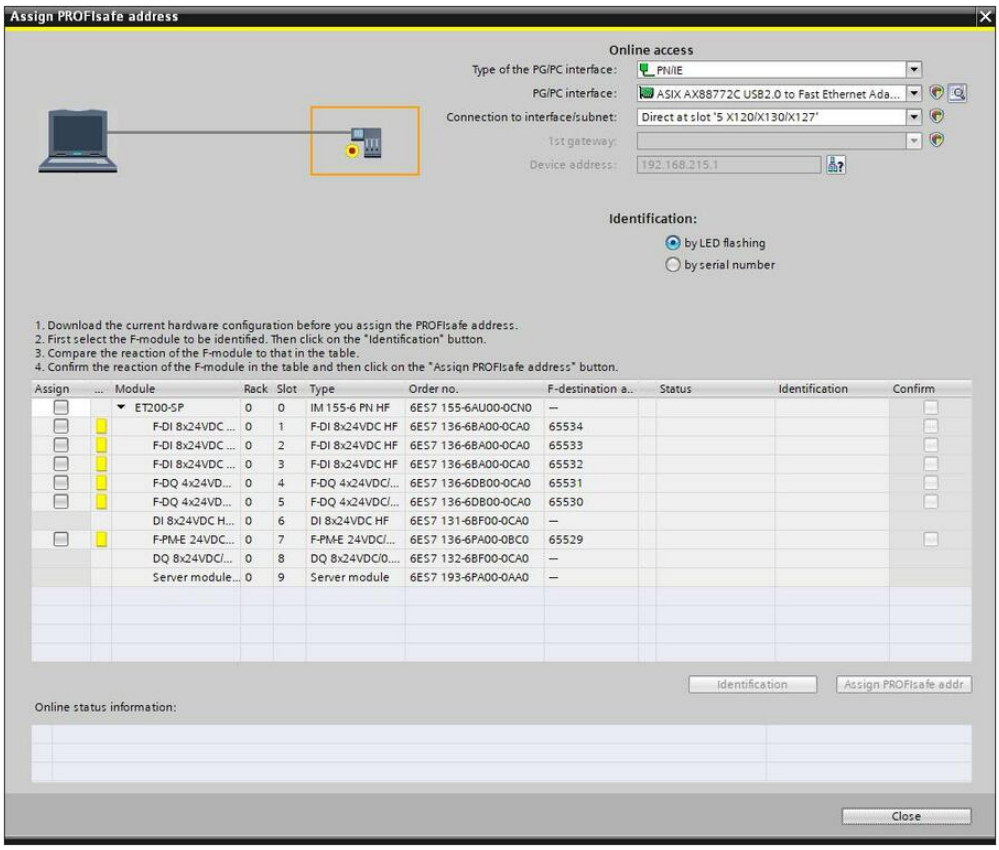
Note: The network cable needs to be connected to X127.



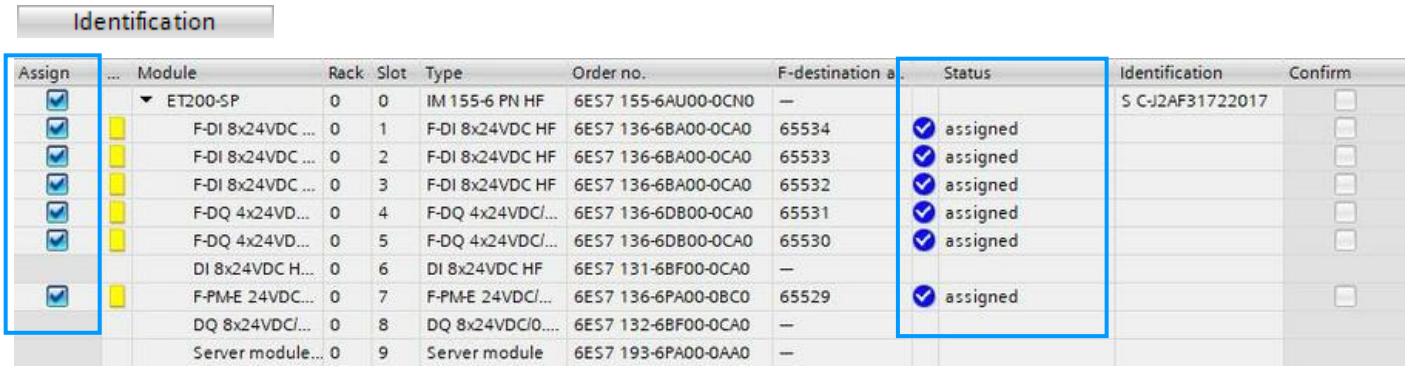
In Device View highlight the ET200-SP IM Module and then right click with the mouse



Notes:



Select the ***Identification*** button and tick the ***Assign*** boxes for the modules





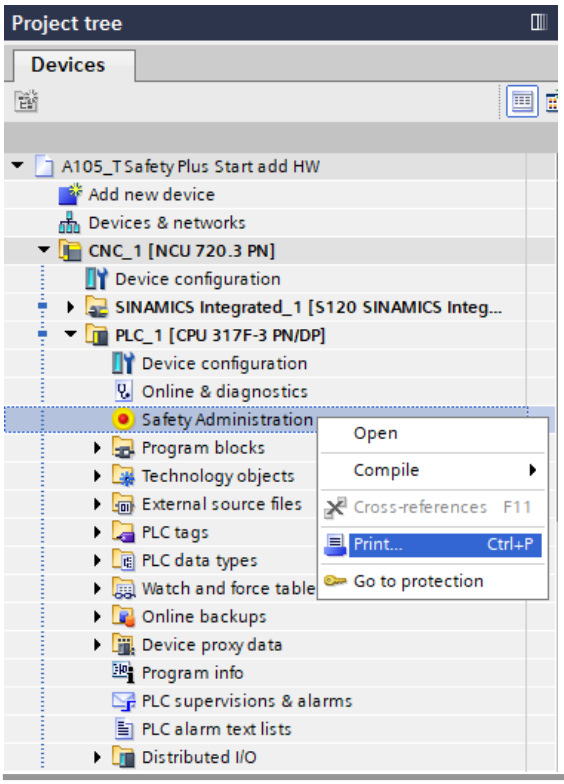
Activate the ***Reset (po)*** procedure

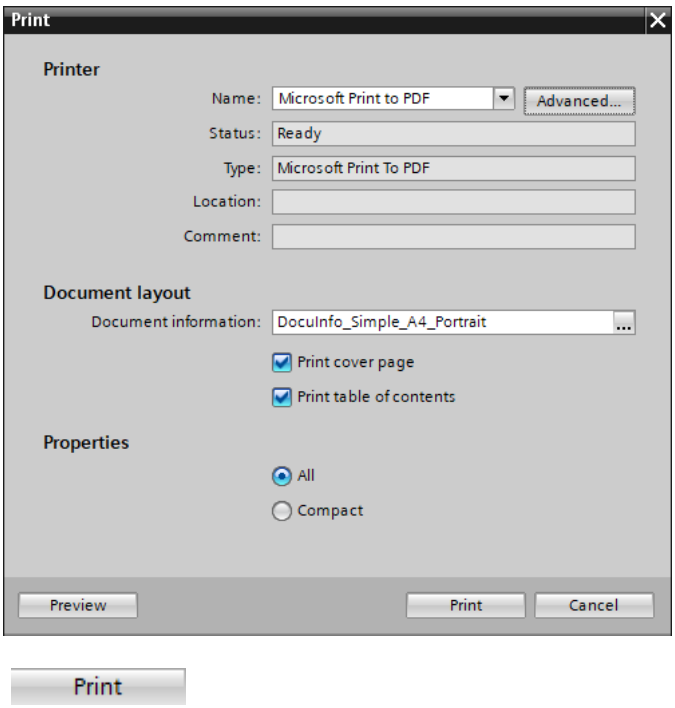


NOTE: ON REAL SINUMERIK ONE
Under "Identification" select how you wish to identify the F-modules:
– "By flashing the LEDs"
This is the default setting. During identification, the STATUS LEDs of the F-modules to be identified flash.
– "Using the serial number"
If you cannot see the F-modules clearly, you can identify them by the serial number of the interface module.
In column "Assign" select all F-modules where you wish to assign an F-destination address.
Click the "Identification" button.
– If you have selected LED flashing for the identification type, then check whether the status LEDs of that F-module flash green whose F-destination address you wish to assign.
– If you have selected the serial number for the identification type, compare the displayed serial number with the serial number of the module.
– Confirm successfully identified F-modules in the "Confirm" column of the table.
– Using button "Assign F-destination address" assign the configured F-destination addresses to the F-modules, and confirm dialog "Acknowledge assignment" (within 60 seconds).

Notes:

Detailed Safety Integrated relevant information of the PLC can be saved:





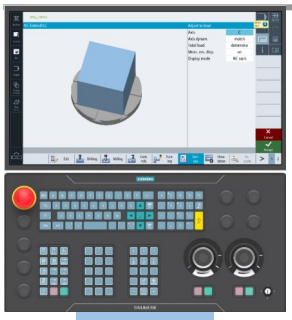
Safety information can be seen in the file:

SINAMICS Integrated_1					
Rail - Slot	Module	Start address	F-destination address	F-monitoring time	Parameter signature (w/o addresses)
0	DriveAxis01FIODB	6700	1	150 ms	0xEE5 (3813)
0	DriveAxis02FIODB	6716	2	150 ms	0xEE5 (3813)
0	DriveAxis03FIODB	6732	3	150 ms	0xEE5 (3813)
0	DriveAxis04FIODB	6748	4	150 ms	0xEE5 (3813)
0	DriveAxis05FIODB	6764	5	150 ms	0xEE5 (3813)
0	DriveAxis06FIODB	6780	6	150 ms	0xEE5 (3813)

ET200-SP					
Rail - Slot	Module	Start address	F-destination address	F-monitoring time	Parameter signature (w/o addresses)
1	6ES7 136-6BA00-0CA0 F-DI 8x24VDC HF_1	512	65534	150 ms	0x2B3C (11068)
2	6ES7 136-6BA00-0CA0 F-DI 8x24VDC HF_2	518	65533	150 ms	0x8AFE (35582)
3	6ES7 136-6BA00-0CA0 F-DI 8x24VDC HF_3	524	65532	150 ms	0xD61 (3425)
4	6ES7 136-6DB00-0CA0 F-DQ 4x24VDC/2A PM HF_1	530	65531	150 ms	0xDE3B (56891)
5	6ES7 136-6DB00-0CA0 F-DQ 4x24VDC/2A PM HF_2	535	65530	150 ms	0xDE3B (56891)
7	6ES7 136-6PA00-0BC0 F-PM-E 24VDC/8A PPM ST_1	540	65529	150 ms	0xE1D4 (57812)

The information should be part of the machine technical documentation and is required for the CE certification:

Notes:



X Y Z SP A

Setpoints/actual Values
Addr. 4xxx (136)

NC-SI
Addr.6xxx (903)
Addr.5xxx (701)

TIA Portal
PLC config

SP X/Y Z/A



Axis-drive assignment in the TIA Portal with safety telegrams

Adapting the physical sequence of the telegram configuration (telegram 136 and safety telegrams 903/701)

As a result of the technical situation in the control cabinet, the load with the highest power consumption (in the example, spindle MSP1) should be located directly next to the infeed, and is therefore assigned to the first Motor Module and is configured as Servo_3.3:3 (Drive Object 3). The I/O addresses of spindle MSP1 (drive axis 04) must therefore be located at the uppermost position in the telegram configuration.

For the telegram configuration the add-in Sinumerik-AxisDrivesAssignment.addin can be used. The add-in is part of the SINUMERIK STEP 7 Toolbox.

After changing, the addresses of the telegrams should correspond to the addresses in the following diagram. The **NC configuration** is left at the default setting. The addresses are adapted in the TIA Portal for the **PLC configuration**.

NC configuration

Machine data remain at the default setting

Axis	MD Index x	Machine Axis = x + 1	MD30110[x] CTRLOUT_ MODULE_NR	MD11350[x] DRIVE_LOGIC_ ADDRESS
MX1	0	1	1	4100
MY1	1	2	2	4140
MZ1	2	3	3	4180
MSP1	3	4	4	4220
MA1	4	5	5	4260

PLC configuration (TIA Portal)
I/O addresses - telegram configuration

Before → After

PROFI drive I/O .	SIC/ SCC	PROFI safe I/O
4100	5800	6700
4140	5816	6716
4180	5832	6732
4220	5848	6748
4260	5864	6764

PROFI drive I/O .	SIC/ SCC	PROFI safe I/O	SINAMICS Integrated
4220	5848	6748	Drive Axis 4
4100	5800	6700	Drive Axis 1
4140	5816	6716	Drive Axis 2
4180	5832	6732	Drive Axis 3
4260	5864	6764	Drive Axis 5

Drive configuration — Drive objects

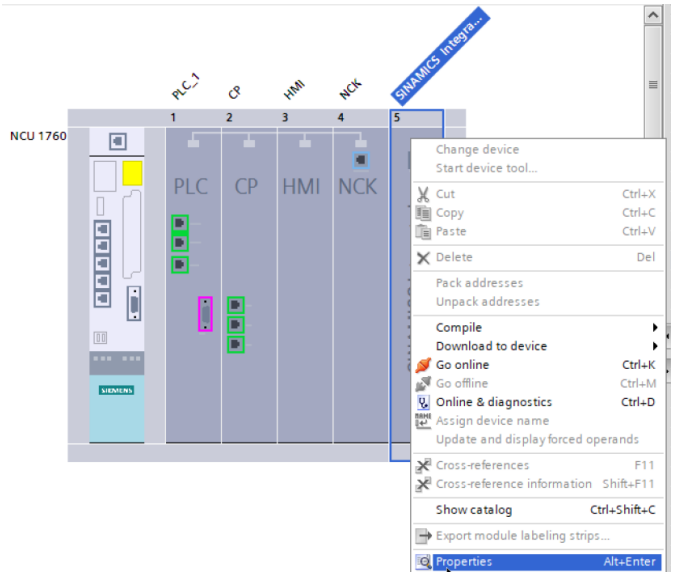
SINAMICS Integrated	Index x	p978[x] List of drive objects (DO)	(No. drive object) and name
Drive Axis 4	0	3	(3) Drive 1 MSP1
Drive Axis 1	1	4	(4) Drive 2 MX1
Drive Axis 2	2	5	(5) Drive 3 MY1
Drive Axis 3	3	6	(6) Drive 4 MZ1
Drive Axis 5	4	7	(7) Drive 5 MA1

Notes:

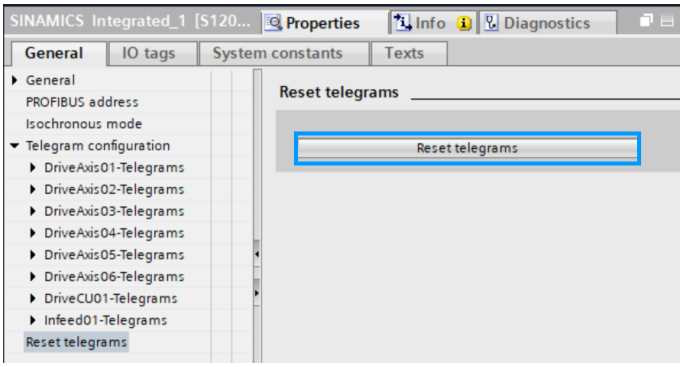
Telegrams: The drive telegrams have been assigned in the Hardware configuration. The telegram addresses now need to be adapted to suit the order of the addressing requirement shown previously.

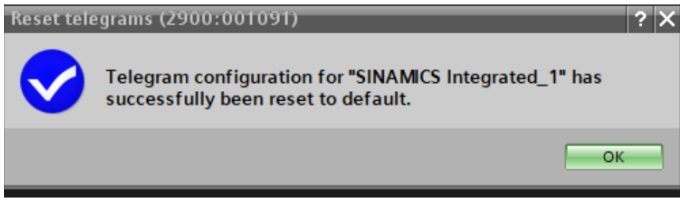
Device view

Right click on Sinamics Integrated



The telegram addresses can be reset as shown below. This is to establish a basic condition before carrying out the axis assignment.

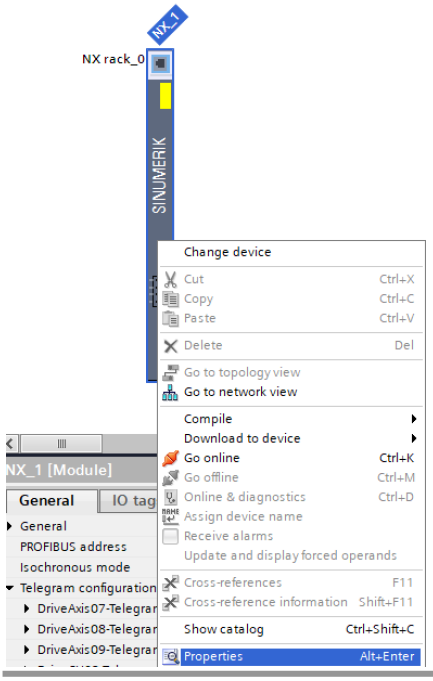




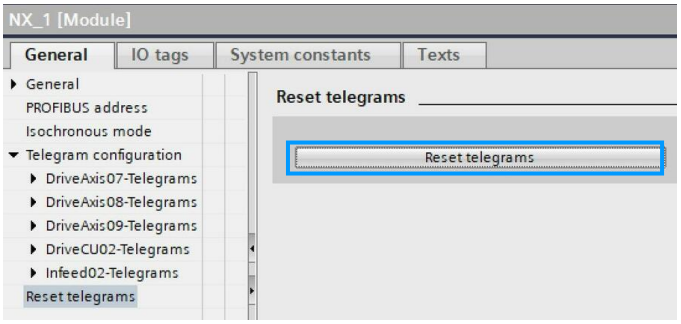
If an NX module is available the same procedure should be used.

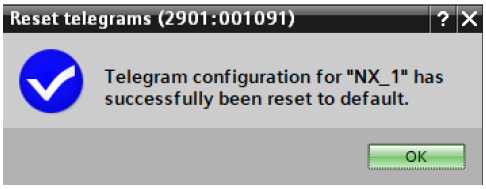
Device view

Right click on NX_1



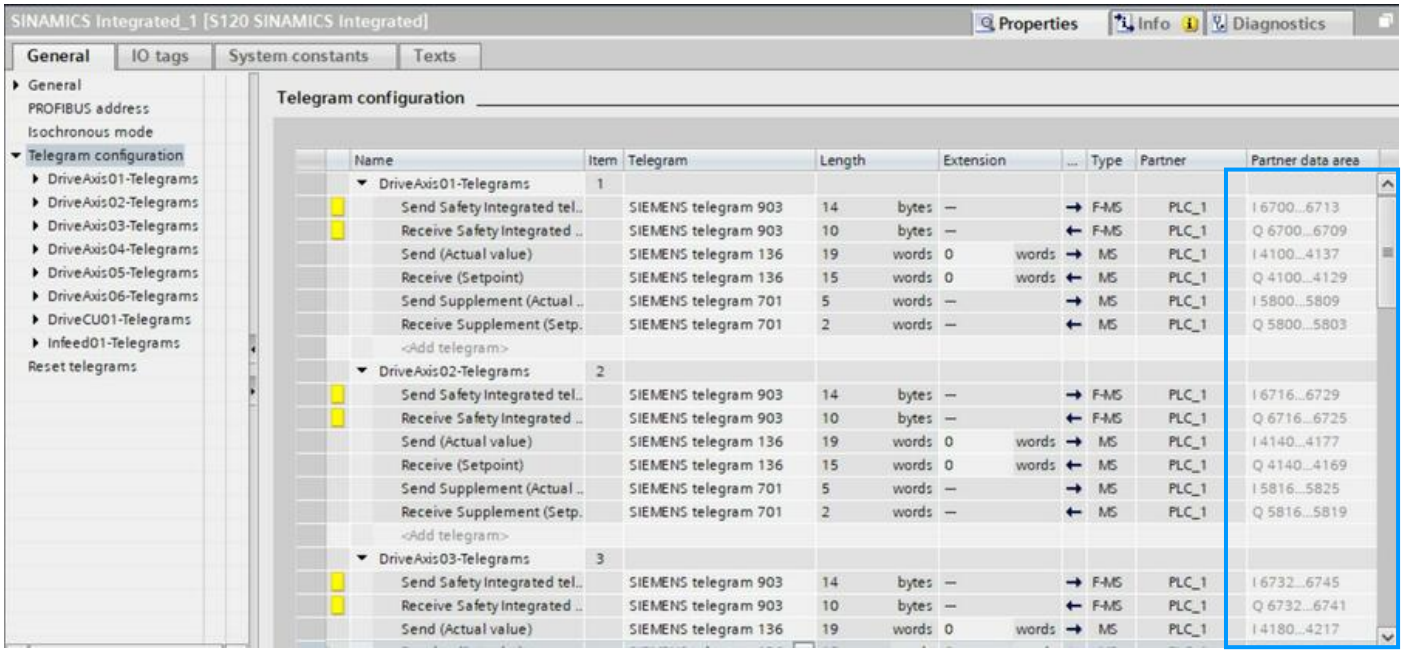
The telegram addresses on the NX if available can be reset as shown below. This is to establish a basic condition before carrying out any axis assignment on the NX.





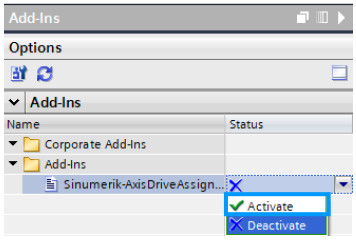
Notes:

Default setting of the safety telegram addresses after resetting the telegrams.

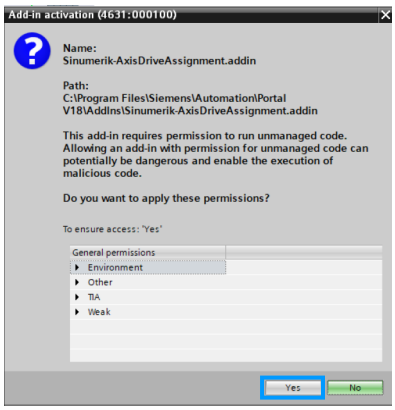


Manage SINAMICS telegrams using the “SINUMERIK-Axis drive Assignment add In

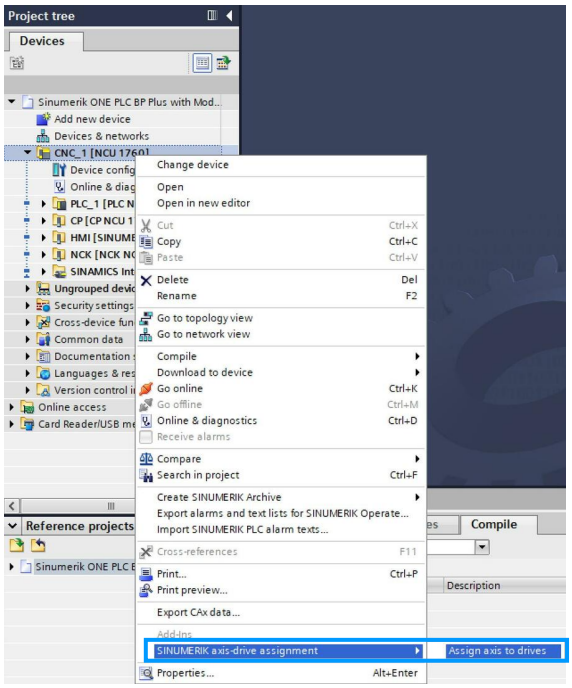
The Add-In “SINUMERIK axis-drive assignment” should be activated.



Continue with “Yes”

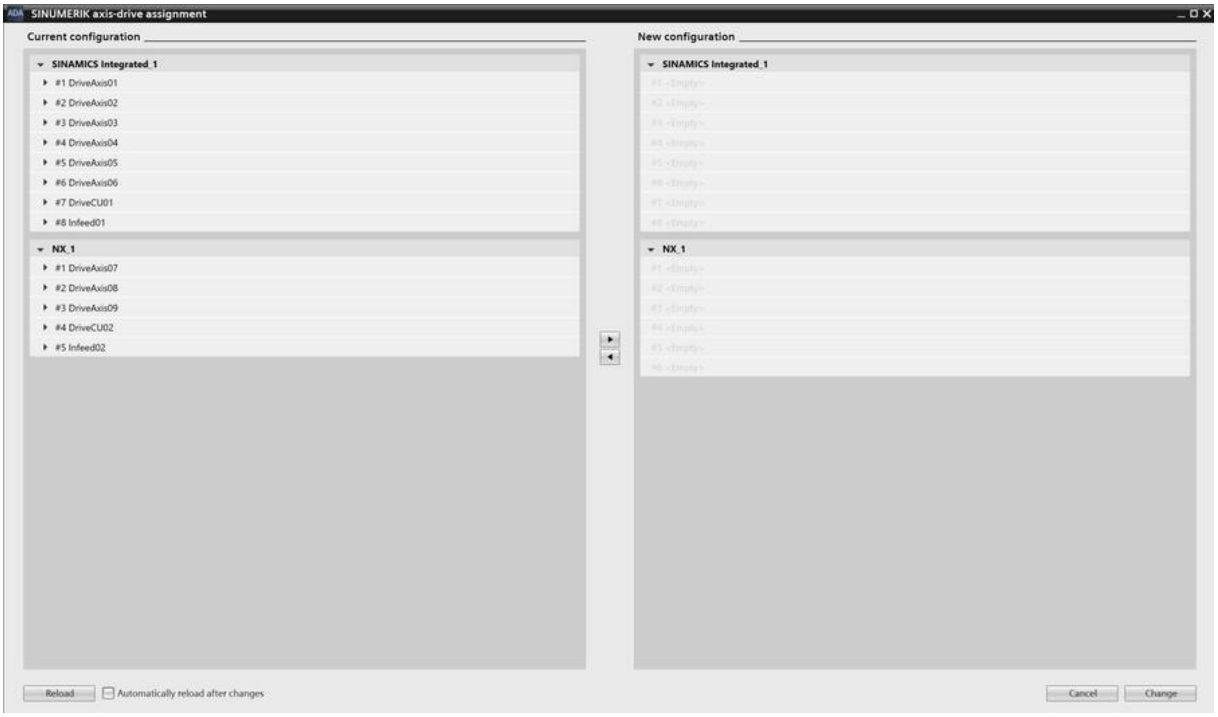


Right click on the NCU and select “SIMUMERIK axis-drive assignment > Assign axes to drives”

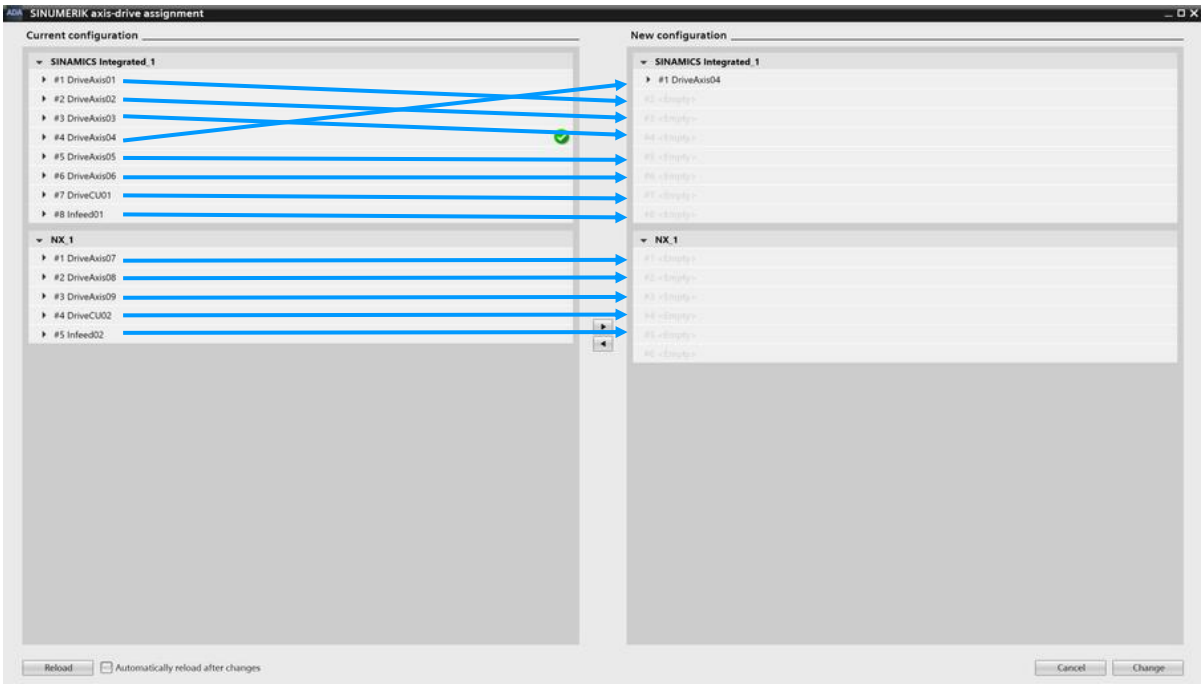


Notes:

The "Current configuration" area provides an overview of all configured telegrams sorted by devices.

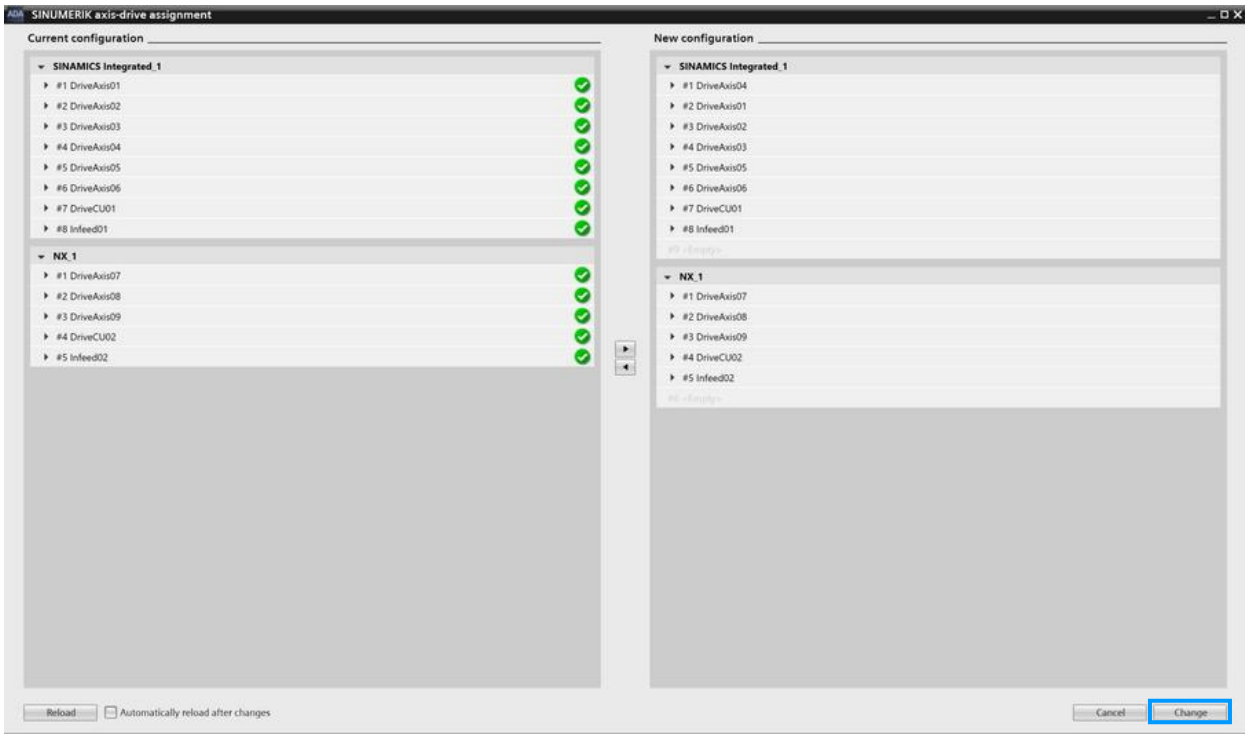


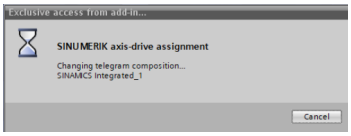
Changes to the configuration can be made using “Drag and Drop”. The devices **must be fully populated**, even if axes/infeeds are not used.

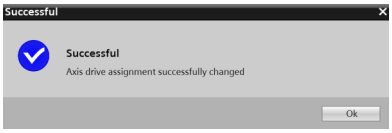


Notes:

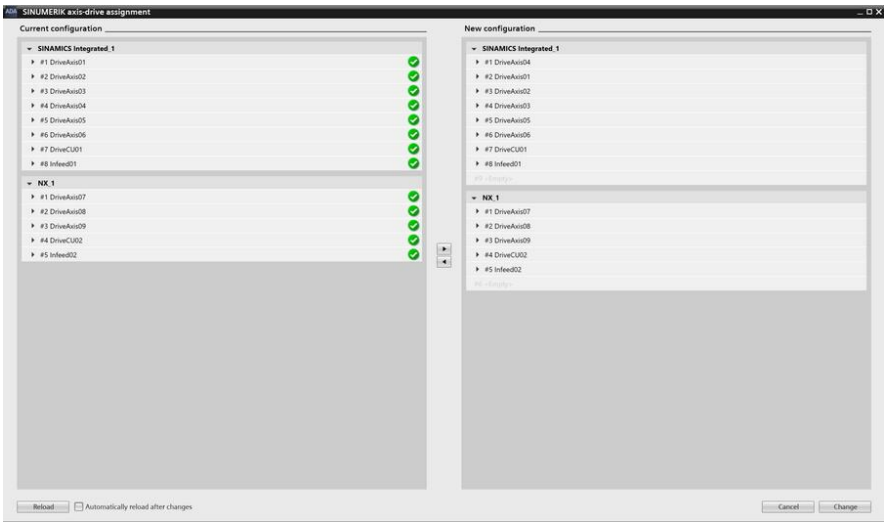
Once all Drive Axes, CUs and Infeeds have been transferred to the “New configuration” then confirm the changes with “Change”







The modified telegram configuration is transferred to the configuration in TIA Portal



After finally acknowledging that the change has been completed, the dialog "SINUMERIK axis-drive assignment" is automatically displayed again.

The content displayed depends on whether the checkmark "Automatically reload after change" was previously set or not (default setting).

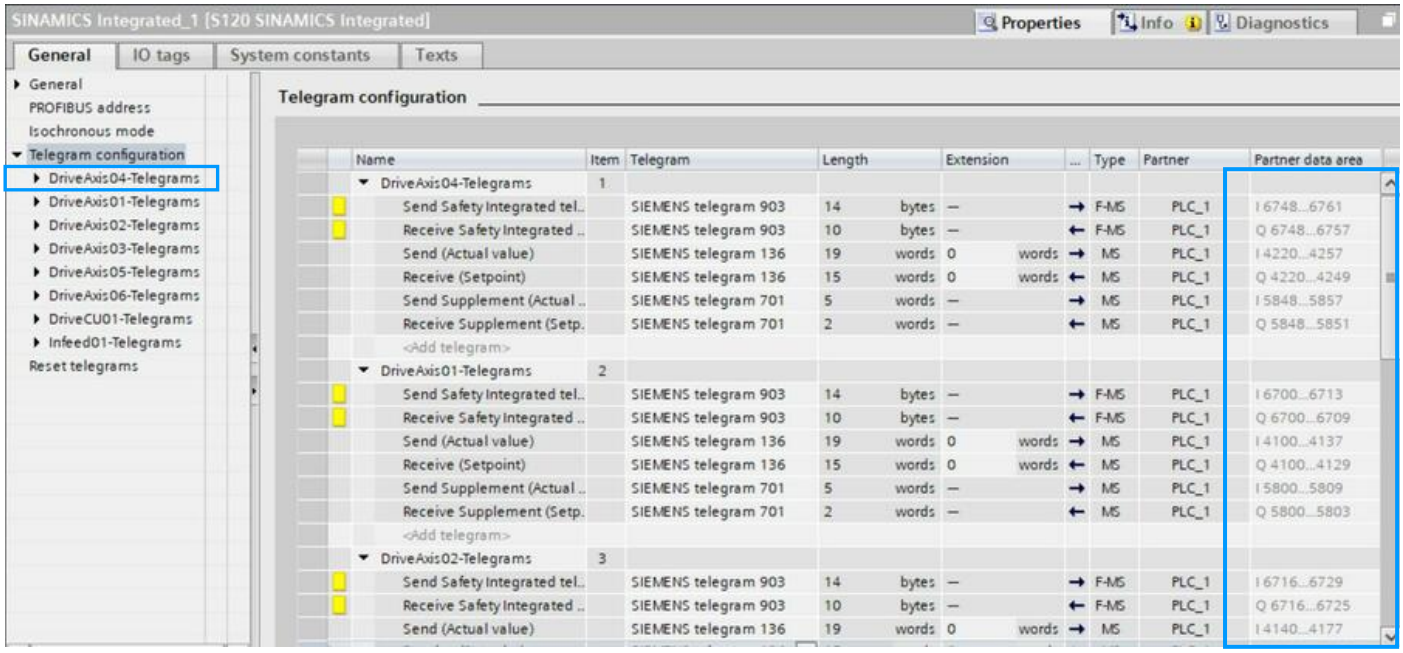
If the checkmark is not set, the status of the change is shown in area "Current configuration", and the modified configuration in the "New configuration" area.

In this way you can again view the original configuration, and when required, restore it to avoid inadvertent changes.

If the checkmark was set, then the content under "Current configuration" corresponds to the updated configuration, and area "New configuration" is empty

Notes:

New setting of the safety telegram addresses with DriveAxis04 (Spindle) at the top of the telegram list



Notes:

The hardware configuration has resulted in the following Input/ Output assignments.

These will be used to achieve the functionality of the OEM function table when writing the F-PLC logic (described later in the document)

Training Simulator Output Functionality	
Function	Address
DO 0	Q100.0
DO 1	Q100.1
DO 2	Q100.2
DO 3	Q100.3
DO 4	Q100.4
DO 5	Q100.5
DO 6	Q100.6
DO 7 Door Unlock	Q100.7
Q0 Relay K1 & LED	Q530.0
Q1 Relay K2 & LED	Q530.1
Q2 Safety Internal Event LED	Q530.2
Q3 Emergency Stop Ack Request LED	Q530.3
F-DO 0	Q535.0
F-DO 1	Q535.1
F-DO 2	Q535.2
F-DO 3	Q535.3
Q Safety Power Module	Q540.0

Training Simulator Input Functionality	
Function	Address
DI 0 Door Unlock Request	I100.0
DI 1	I100.1
DI 2	I100.2
DI 3	I100.3
DI 4	I100.4
DI 5	I100.5
DI 6 K1 & K2 Relay Feedback	I100.6
DI 7	I100.7
Emergency Stop	I512.0
F-DI 2 & F-DI6	I512.1
Door Locked	I512.2
Door Closed	I512.3
F-DI 0 & F-DI4	I518.0
Emergency Stop Reset	I518.1
Set-up Mode	I518.2
F-DI 3 & F-DI7	I518.3
Hold-To-Run PB	I524.0
F-DI 1 & F-DI5	I524.1
F-DI 2 & F-DI6	I524.2
F-DI 3 & F-DI7	I524.3

Safety

	Name	Data type	Address
1	SafeSpeedMonitorA	Bool	%I6765.7
2	AckIntEventA	Bool	%Q6764.7
3	AckIntEventSP	Bool	%Q6748.7
4	AckIntEventX	Bool	%Q6700.7
5	AckIntEventY	Bool	%Q6716.7
6	AckIntEventZ	Bool	%Q6732.7
7	DeselectSOSA	Bool	%Q6764.3
8	DeselectSS1A	Bool	%Q6764.1
9	DeselectSS2A	Bool	%Q6764.2
10	DeselectSTOA	Bool	%Q6764.0
11	DeselectSOSSP	Bool	%Q6748.3
12	DeselectSS1SP	Bool	%Q6748.1
13	DeselectSS2SP	Bool	%Q6748.2
14	DeselectSTOSP	Bool	%Q6748.0
15	DeselectSOSX	Bool	%Q6700.3
16	DeselectSS1X	Bool	%Q6700.1
17	DeselectSS2X	Bool	%Q6700.2
18	DeselectSTOX	Bool	%Q6700.0
19	DeselectSOSY	Bool	%Q6716.3
20	DeselectSS1Y	Bool	%Q6716.1
21	DeselectSS2Y	Bool	%Q6716.2
22	DeselectSTOY	Bool	%Q6716.0
23	DeselectSOSZ	Bool	%Q6732.3
24	DeselectSS1Z	Bool	%Q6732.1
25	DeselectSS2Z	Bool	%Q6732.2
26	DeselectSTOZ	Bool	%Q6732.0
27	HoldToRunButton	Bool	%I524.0
28	InternalEventActiveA	Bool	%I6764.7
29	InternalEventActiveSP	Bool	%I6748.7
30	InternalEventActiveX	Bool	%I6700.7
31	InternalEventActiveY	Bool	%I6716.7
32	InternalEventActiveZ	Bool	%I6732.7
33	SLSBit0A	Bool	%Q6765.1
34	SLSBit0SP	Bool	%Q6749.1
35	SLSBit0X	Bool	%Q6701.1
36	SLSBit0Y	Bool	%Q6717.1
37	SLSBit0Z	Bool	%Q6733.1
38	SLSBit1A	Bool	%Q6765.2

Safety

	Name	Data type	Address
39	SLSBit1SP	Bool	%Q6749.2
40	SLSBit1X	Bool	%Q6701.2
41	SLSBit1Y	Bool	%Q6717.2
42	SLSBit1Z	Bool	%Q6733.2
43	SafeSpeedMonitorSP	Bool	%I6749.7
44	SafeSpeedMonitorX	Bool	%I6701.7
45	SafeSpeedMonitorY	Bool	%I6717.7
46	SafeSpeedMonitorZ	Bool	%I6733.7
47	SOSActiveX	Bool	%I6700.3
48	SOSActiveA	Bool	%I6764.3
49	SOSActiveSP	Bool	%I6748.3
50	SOSActiveZ	Bool	%I6732.3
51	SOSActiveY	Bool	%I6716.3
52	SetupSwitch	Bool	%I518.2
53	EmergencyStopResetPB	Bool	%I518.1
54	Q PowerModule	Bool	%Q540.0
55	Mode3Switch	Bool	%I518.3
56	DoorLocked	Bool	%I512.2
57	DoorClosed	Bool	%I512.3
58	End Limit Override	Bool	%I518.0
59	Safe Cam1 DO 0 LED	Bool	%Q535.0
60	Safe Cam2 DO 0 LED	Bool	%Q535.1
61	Q1 Relay K2 Door Closed	Bool	%Q530.1
62	Q0 Relay K1 Door Closed and L...	Bool	%Q530.0
63	Internal Event Lamp	Bool	%Q530.2
64	EmergencyStopPB	Bool	%I512.0
65	SPDeselectSLS	Bool	%Q6748.4
66	SelectSLPRange2X	Bool	%Q6702.3
67	SLPUpperLimitX	Bool	%I6703.6
68	SLPLowerLimitX	Bool	%I6703.7
69	SDIPositiveX	Bool	%Q6701.4
70	SafelyReferencedX	Bool	%I6702.7
71	SDINegativeX	Bool	%Q6701.5
72	SafeCamsActivatedX	Bool	%Q6702.7
73	SafeCamValid	Bool	%I6707.6
74	SafeCam1X	Bool	%I6704.0
75	SafeCam2X	Bool	%I6704.1
76	ESTOPResetRequired	Bool	%Q530.3

The Tag tables for the F-PLC can be prepared as shown here, or the tag names can be entered individually as the PLC logic is created.

Safety Other

	Name	Data type	Address
1	DoorUnlockButton	Bool	%I100.0
2	K1/K2feedback	Bool	%I100.6
3	DoorUnlockRequest	Bool	%Q100.7

Notes:

Safety Integrated Drive Configuration

The required Safety Integrated functionality for each drive are configured in the Sinumerik Operate Set-up area.

This section will achieve the following:

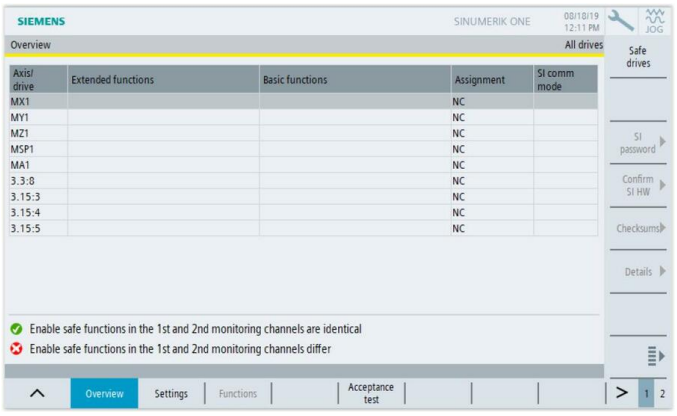
- Enable Safety Integrated extended functions for each drive (see the adjacent table)
- Setpoint Limitation - used to limit the feed-rate of an axis to a percentage of the active Safe Limited Speed
- Setting of the Test Stop interval time
- Safe encoder configuration
- Actual value synchronisation for 2nd encoders
- Drive telegram configuration

Basic and Extended Safety Functions:

Safe Functions		Abbreviation
Basic Functions	Safe Torque Off	STO
	Safe Stop 1	SS1
	Safe Brake Control	SBC
Extended Functions (includes basic functions)	Safe Stop 2	SS2
	Safe Operating Stop	SOS
	Safely Limited Speed	SLS
	Safe Speed Monitor	SSM
	Safe Acceleration Monitor	SAM
	Safe Direction	SDI
	Safe Limit Positions	SLP
	Safe Brake Test	SBT
	Safe Cams	SCA

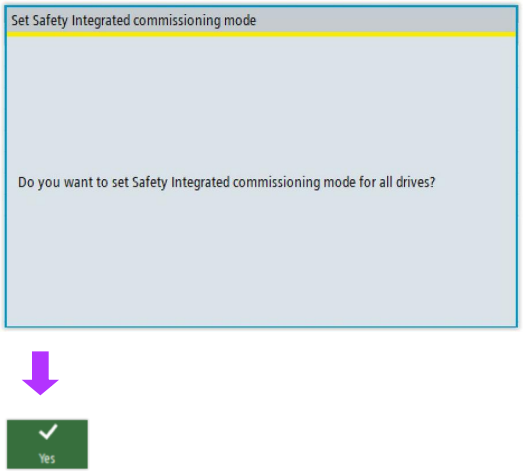
Activating Safety Integrated Commissioning Mode:





The commissioning mode needs to be set for the axes:





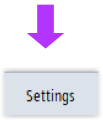
The active commissioning mode status is displayed in the right hand column:

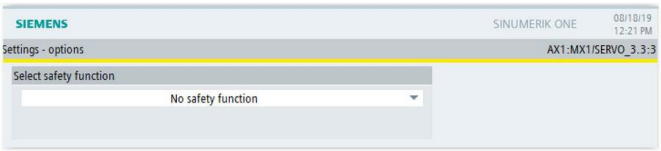


Notes:

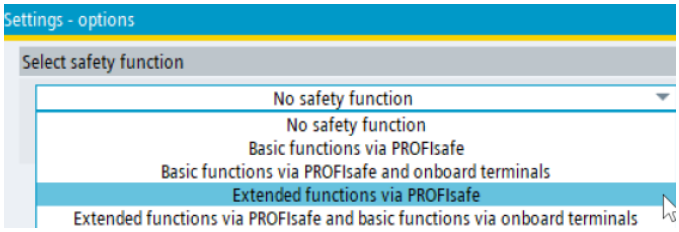
Settings.

Extended Functions

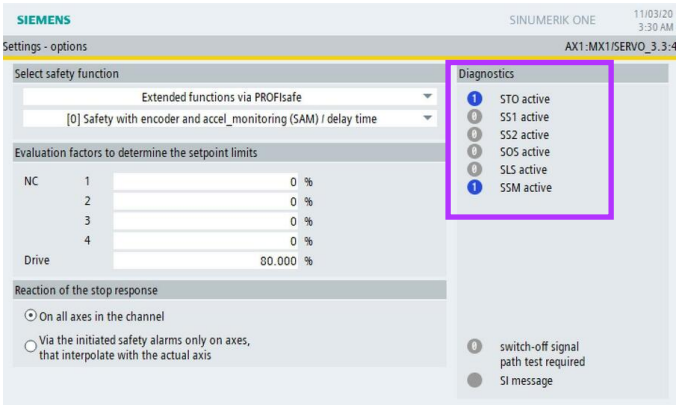




Select the ***Extended functions via PROFsafe*** option. This offers the following Safe functions:



Shown below are the Safety Functions available when the Extended functions via PROFIsafe option is selected

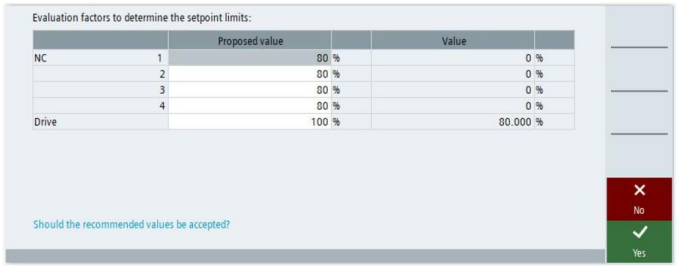


Set-point Limitation

Set-point limitation is used to limit the feed-rate of an axis.to a percentage of the active safe limited speed (**SLS**).

Selecting Recommended values proposes values of 80% and these values should be accepted.







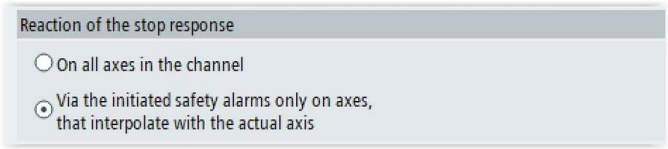
The 80% set-point limitation will limit the normal operating feed rates to 80% of the active SLS limitation.
The SLS limitations are set later in the document.

Reaction to a stop response

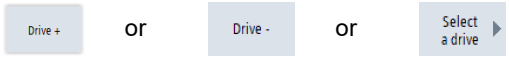
Each axis can be set to react to a stop response in one of two ways:

1. On all axes in the channel
2. Via the initiated safety alarms on axes that interpolate with the actual axis (recommended)

The selection made will depend on the individual machine configuration. For example, a fault on an auxiliary axis, such as a tool changer, can be ignored and the machining axes allowed to continue.



Now select and repeat the procedure on the next drive. Re-peat this one drive at a time. **(SEE NOTE)**

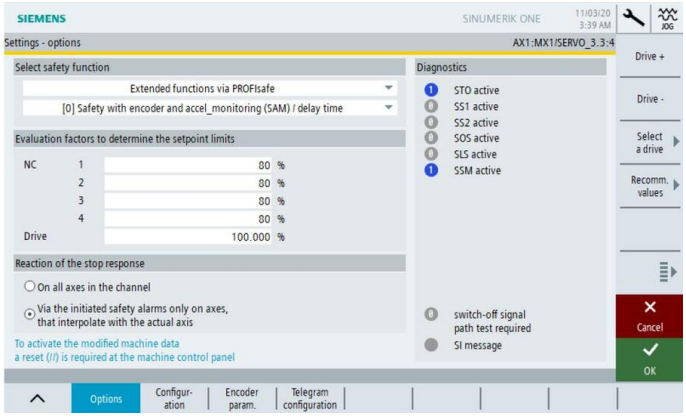


Note: on the virtual machine it maybe necessary to set the values and then copy/confirm/save each drive indi-vidually. On the real machine all drives can be copied/ confirmed/saved

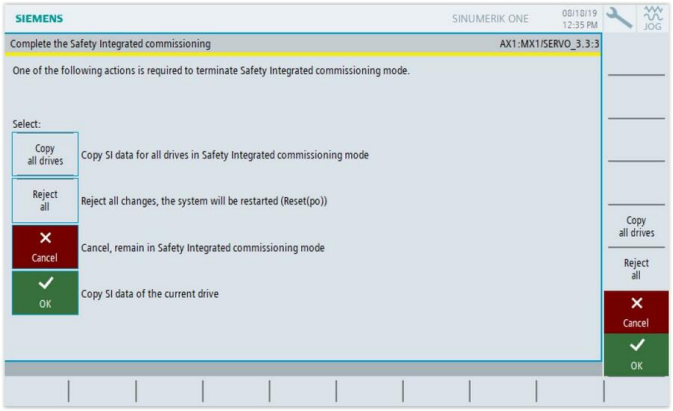
Notes:

Saving the changes:

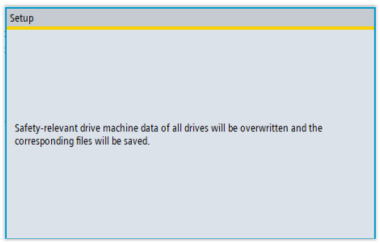
The changes need to be copied and confirmed. The values entered into the screens go into the Control Unit specific data. With the copy function the data is copied into the Motor Module data. With confirm the *Control Unit and Motor Module data* checksums will then match each other. The checksums are only valid for the ***real*** Sinumerik ONE and not the Virtual machine.
The checksums are monitored in two channels (The Control Unit and the Motor Module). Any discrepancies will result in Safety Integrated alarms.

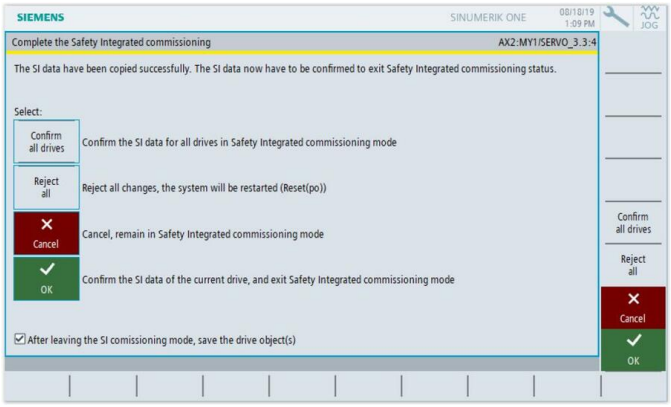


OK — Start the procedure for copy/confirm

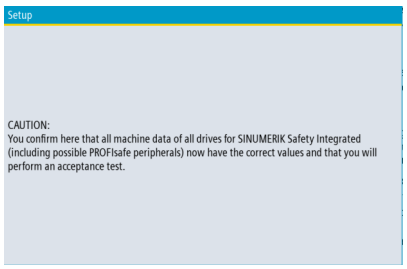


Copy all drives — Copy the SI data of all drives

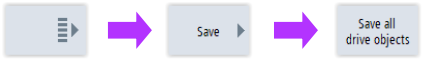


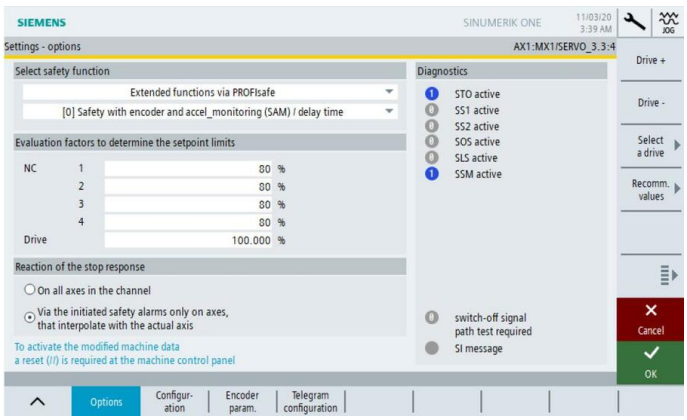


Confirm all drives — Confirm the SI data of all drives

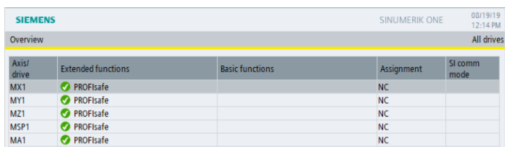


Save the SI data of all drives





Extended functions via Profisafe are now set for all relevant drives



Notes:

Configuration of the Safety Integrated features.



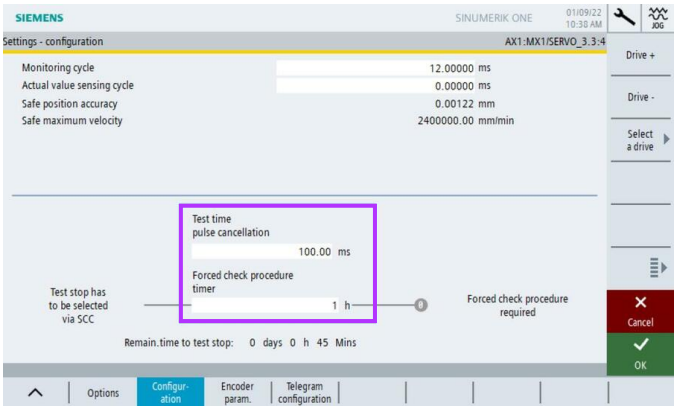
Activate the ***SI commissioning*** mode:



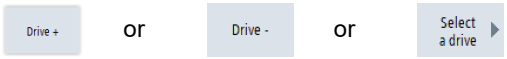
Test Stop Interval Time: Forced checking procedure

The forced checking procedure is used to detect faults/errors in the software and hardware of the two monitoring channels. In order to do this, the safety - related parts in both channels must be processed at least once during a defined period in all safety-related branches. Any faults/errors in the monitoring channel would cause deviations and will be detected by the crosswise data comparison.

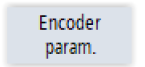
For the purpose of the training this can be set to a low value, e.g. 1 hour. This will allow the relevant Test Stop interface signals/messages to be monitored without the need to wait. All other settings can be left as standard.



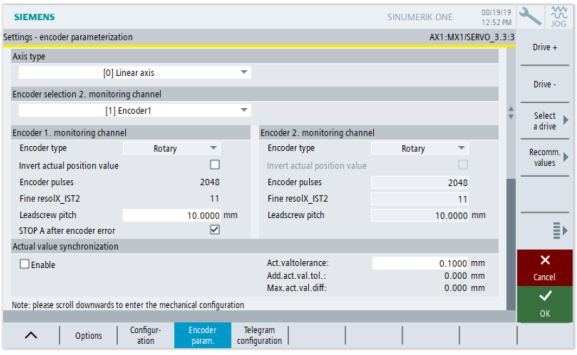
Now repeat for all of the drives



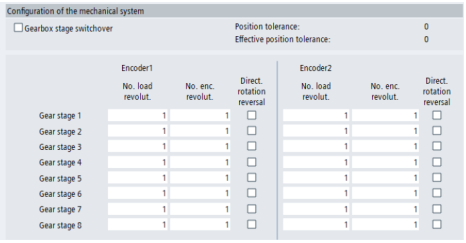
Encoder Parameters: (Only Encoder 1 on Virtual ONE)



Note: The Virtual machine can only be used with a single encoder for each axis.

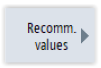


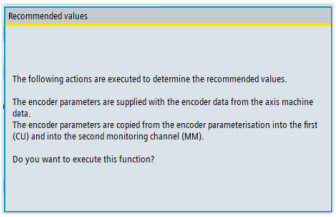
The axis type and encoder information etc. has to be entered for each drive.Gearbox information is also available by using the scroll bar to go to the lower half of the screen



Example configuration of the training equipment can be seen in the table below

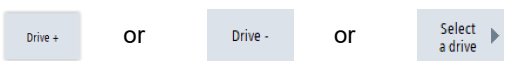
Axis	Type	Encoders	Leadscrew	Gears
MX1	Linear	1	10.00 mm	n/a
MY1	Linear	1	10.00 mm	n/a
MZ1	Linear	1	10.00 mm	n/a
MSP1	Rotary	2	n/a	n/a
MA1	Rotary	1	n/a	n/a





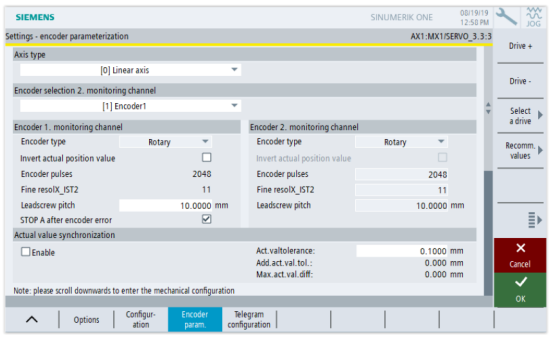


Now repeat for all of the drives

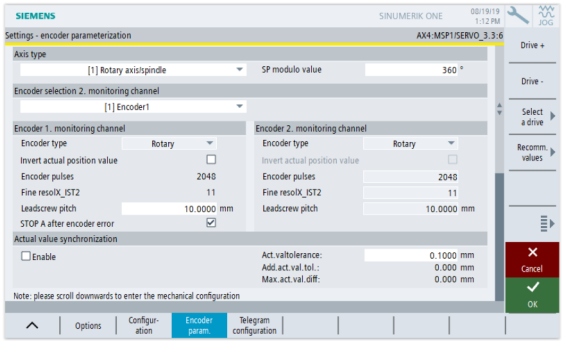


Notes:

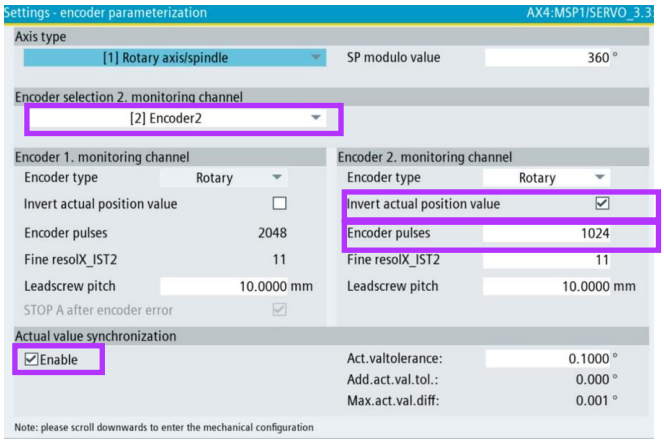
Below, X, Y and Z after recommended values



For the Main Spindle and the A Axis, Axis type [1] Rotary axis/spindle and SP modulo value 360 should be used.



NOTE: On a Real Sinumerik ONE machine with second encoder the [2] Encoder2 should be selected.
Also Invert position actual value.
Encoder pulses 1024.
Enable Actual value synchronization.

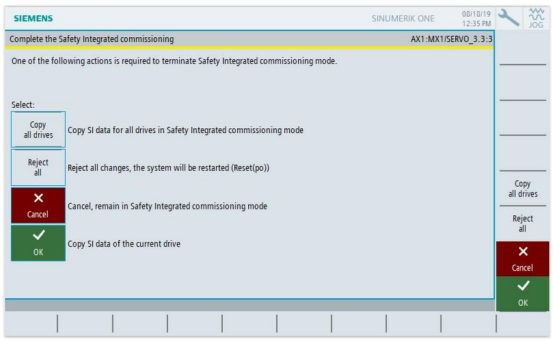


Position actual value is inverted due to the mounting of the spindle 2nd encoder (on the front of the spindle) therefore the feedback pulses are in the opposite direction.
The 2nd encoder has 1024 pulses/rev.

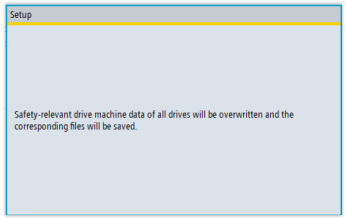
The actual value synchronization allows a tolerance to be entered on systems with two encoders. This prevents nuisance alarms which can occur due to the mechanical properties of gear boxes or belt driven systems etc.

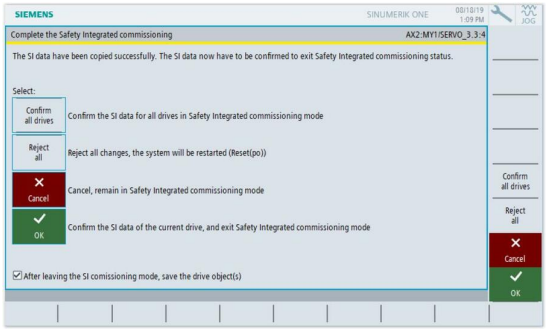
OK Start the procedure for copy/confirm

NOTE: On the Real SINUMERIK ONE it will be necessary to copy/confirm twice because of Spindle and A Axis units change.

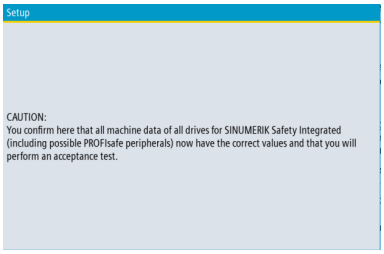


Copy all drives Copy the SI data of all drives

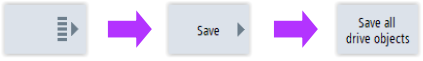




Confirm all drives Confirm the SI data of all drives



Save the SI data of all drives



Notes:

Telegram Configuration:

The **Telegram configuration** area is used to enable the ***S**afety **I**nformation **C**hannel (**SIC**), the **S**afety **C**ommand **C**hannel (**SCC**)* and ***PROFIsafe*** for each drive.The required Telegrams are also selected here, this allows interfacing between the drive and user PLC for monitoring and controlling of the safety integrated functions.

The required SIC/SCC telegram is telegram 701. This allows the Safe Brake Test to be performed and the starting of the Test Stop procedure via the axis interface DB.

The standard PROFIsafe telegram, is telegram 903. This can be changed to achieve different functionality (refer to the Safety Integrated theory document)

The SCC/SIC & PROFIsafe drive module numbers are also set here. These values depend on the drive topology/drive number and the Axis Index. The training equipment configuration is as follows:

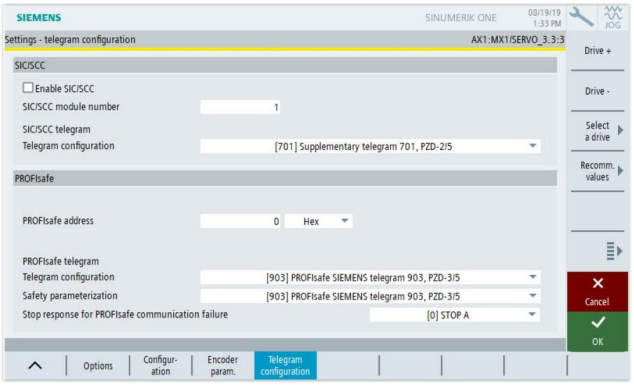
Axis	MD37954 (SIC/SCC NR) SAFE_INFO_ MODULE_NR		MD13374 SAFE_INFO_ DRIVE_LOGIC _ADDRESS		From Telegram to Axis DB	Addressing For Telegram 701
MX1	1		[0] = 5800		DB31	
MY1	2		[1] = 5816		DB32	
MZ1	3		[2] = 5832		DB33	
MSP1	4		[3] = 5848		DB34	
MA1	5		[4] = 5864		DB35	

Drive	Telegram Configuration
DO3	DriveAxis04
DO4	DriveAxis01
DO5	DriveAxis02
DO6	DriveAxis03
DO7	DriveAxis05

Item	F– Destination Address		Start Address		Addressing For Telegram 903
1	4		6748		
2	1		6700		
3	2		6716		
4	3		6732		
5	5		6764		

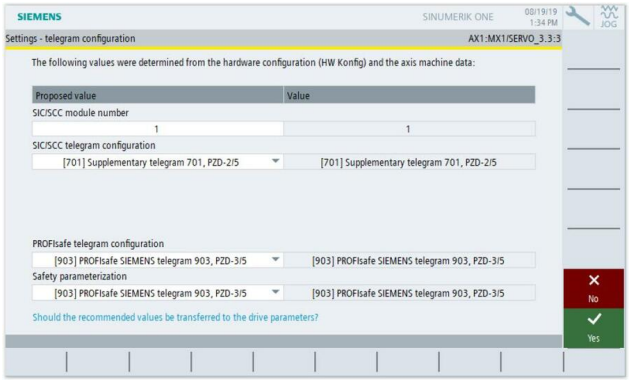
Drive	Telegram Configuration
DO3	DriveAxis04
DO4	DriveAxis01
DO5	DriveAxis02
DO6	DriveAxis03
DO7	DriveAxis05

Telegram configuration



Recomm. values

Check that the values match the values in the tables above. If they do not then edit them manually.



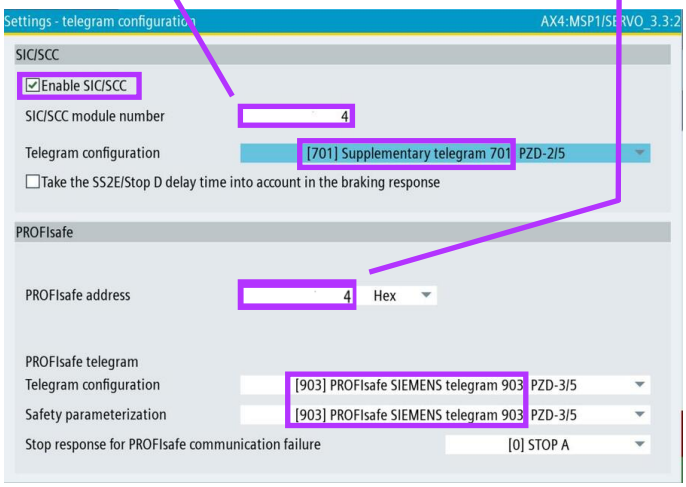
The SIC/SCC 701 telegram is also set here and also the PROFIsafe 903 telegram.

OK

Notes:

Axis	MD37954 (SIC/SCC NR) SAFE_INFO_MODULE_NR
MX1	1
MY1	2
MZ1	3
MSP1	4
MA1	5

F– Destination Address
4
1
2
3
5



- Set the ***Enable SIC/SCC*** option
- Check that ***Telegram 701*** has been selected for the telegram configuration for the SIC/SCC.
- Check that ***Telegram 903*** has been selected for the Telegram configuration and Safety parameterization of PROFIsafe

Check the following settings in accordance with the table on the previous page.

- SIC/SCC module number
- PROFISafe address - ***set manually***

Now repeat for all of the drives

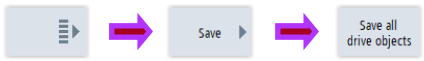
Drive + or Drive - or Select a drive

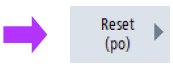
OK Start the procedure for copy/confirm

Copy all drives Copy the SI data of all drives

Confirm all drives Confirm the SI data of all drives

Save the SI data of all drives





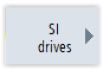
Confirm all drives Confirm all drives a second time if required because of Spindle/A Axis units

Notes:

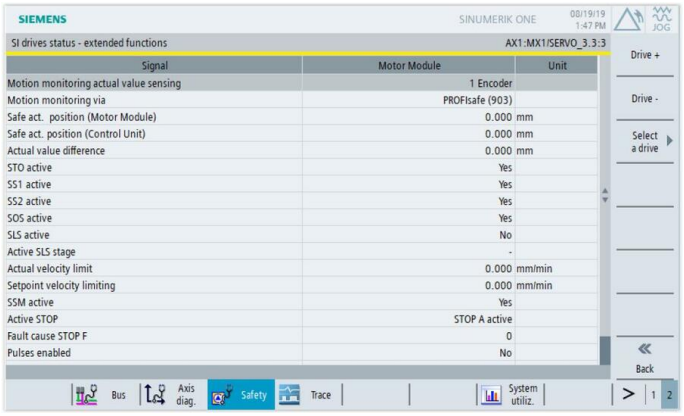
Safety Integrated Drive/Telegram Status.

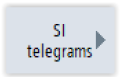
The telegram and drive statuses can be seen in the Safety Integrated diagnostics area





The status of the standard and extended safety functions can be seen on the ***SI drives*** screen.

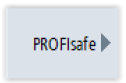




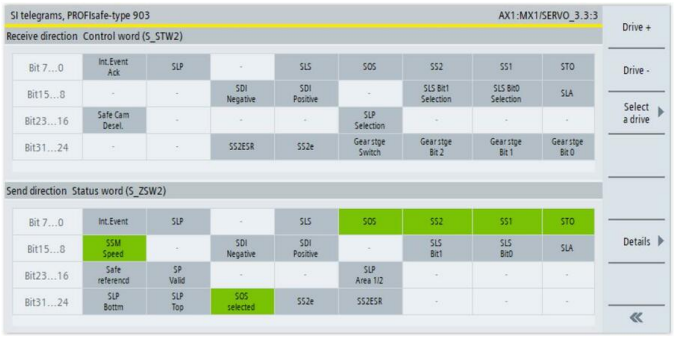
The assigned telegrams can be seen on ***SI telegrams*** screen.

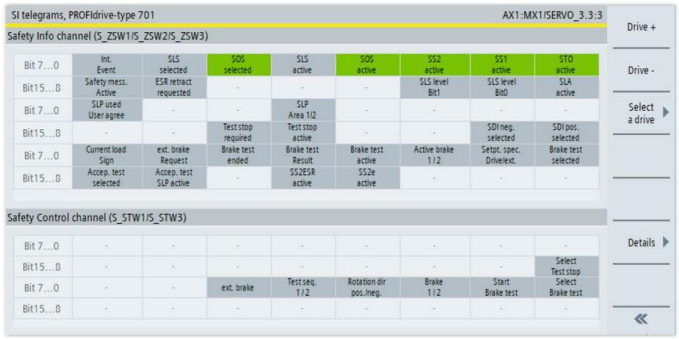
Overview of SI telegrams

Axis/drive	PROFIsafe telegram type	SIC/SCC telegram type
AX1:MX1/SERVO_3.3:3	903	701
AX2:MY1/SERVO_3.3:4	903	701
AX3:MZ1/SERVO_3.3:5	903	701
AX4:MSP1/SERVO_3.3:2	903	701
AX5:MA1/SERVO_3.3:6	903	701

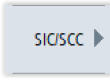


The PROFIsafe telegram status of each drive can be viewed.
The Status isn't displaying anything due to the missing F-PLC logic.





These signals are used to initiate/monitor the Test Stop and Safe Brake Test routines.



The Safety Information Channel (SCC) and Safety Command Channel (SCC) status of each drive can be viewed.

Notes:

F-PLC Logic

PLC I/O used to achieve Safety functionality

The following tables show the training equipment Input/Output assignments which were determined during the hardware configuration.
These signals will be used to achieve the functionality of the OEM function table.

Training Simulator Output Functionality	
Function	Address
DO 0	Q100.0
DO 1	Q100.1
DO 2	Q100.2
DO 3	Q100.3
DO 4	Q100.4
DO 5	Q100.5
DO 6	Q100.6
DO 7 Door Unlock	Q100.7
Q0 Relay K1 & LED	Q530.0
Q1 Relay K2 & LED	Q530.1
Q2 Safety Internal Event LED	Q530.2
Q3 Emergency Stop Ack Request LED	Q530.3
F-DO 0	Q535.0
F-DO 1	Q535.1
F-DO 2	Q535.2
F-DO 3	Q535.3
Q Safety Power Module	Q540.0

Training Simulator Input Functionality	
Function	Address
D I0 Door Unlock Request	I100.0
DI 1	I100.1
DI 2	I100.2
DI 3	I100.3
DI 4	I100.4
DI 5	I100.5
DI 6 K1 & K2 Relay Feedback	I100.6
DI 7	I100.7
Emergency Stop	I512.0
F-DI 2 & F-DI6	I512.1
Door Locked	I512.2
Door Closed	I512.3
F-DI 0 & F-DI4	I518.0
Emergency Stop Reset	I518.1
Set-up Mode	I518.2
F-DI 3 & F-DI7	I518.3
Hold-To-Run PB	I524.0
F-DI 1 & F-DI5	I524.1
F-DI 2 & F-DI6	I524.2
F-DI 3 & F-DI7	I524.3

Safety Functionality between the Drive and the PLC.

The required Safety Integrated functionality is achieved by setting the interface signals (determined by the PROFISafe telegram structure refer to the previous page) as shown below.

PLC to Drive (Telegram S_STW2 Control words 1 & 2)					
Function	Sp QB6748	X QB6700	Y QB6716	Z QB6732	A QB6764
Deselect STO	.0	.0	.0	.0	.0
Deselect SS1	.1	.1	.1	.1	.1
Deselect SS2	.2	.2	.2	.2	.2
Deselect SOS	.3	.3	.3	.3	.3
Deselect SLS	.4	.4	.4	.4	.4
Reserved Out 1	.5	.5	.5	.5	.5
Deselect SLP	.6	.6	.6	.6	.6
Ack. Int. Event	.7	.7	.7	.7	.7
	QB6749	QB6701	QB6717	QB6733	QB6765
Reserved Out 2	.0	.0	.0	.0	.0
Select SLS bit 0	.1	.1	.1	.1	.1
Select SLS bit 1	.2	.2	.2	.2	.2
Reserved Out 3	.3	.3	.3	.3	.3
Deselect SDI Pos	.4	.4	.4	.4	.4
Deselect SDI Neg	.5	.5	.5	.5	.5
Reserved Out 4	.6	.6	.6	.6	.6
Reserved Out 5	.7	.7	.7	.7	.7

PLC to Drive (Telegram S_STW2 Control words 2 & 3)					
Function	Sp QB6750	X QB6702	Y QB6718	Z QB6734	A QB6766
Reserved Out 6	.0	.0	.0	.0	.0
Reserved Out 7	.1	.1	.1	.1	.1
Reserved Out 8	.2	.2	.2	.2	.2
Select SLP range	.3	.3	.3	.3	.3
Reserved Out 9	.4	.4	.4	.4	.4
Reserved Out 10	.5	.5	.5	.5	.5
Reserved Out 11	.6	.6	.6	.6	.6
Des. Safe Cam	.7	.7	.7	.7	.7
	QB6751	QB6703	QB6719	QB6735	QB6767
Select Gear bit 0	.0	.0	.0	.0	.0
Select Gear bit 1	.1	.1	.1	.1	.1
Select Gear bit 2	.2	.2	.2	.2	.2
Deselect Gear Sw	.3	.3	.3	.3	.3
Deselect SS2E	.4	.4	.4	.4	.4
Deselect SS2ESR	.5	.5	.5	.5	.5
Reserved Out 14	.6	.6	.6	.6	.6
Reserved Out 15	.7	.7	.7	.7	.7
SLS1 Limit	QW6752	QW6704	QW6720	QW6736	QW6768

Notes:

Safety Integrated Status

The Safety Integrated status signals can be monitored and used in the F-PLC logic- as shown in the following tables.

Drive to PLC (Telegram S_ZSW2 Status words 1 & 2)					
Function	Sp IB6748	X IB6700	Y IB6716	Z IB6732	A IB6764
STO Active	.0	.0	.0	.0	.0
SS1 Active	.1	.1	.1	.1	.1
SS2 Active	.2	.2	.2	.2	.2
SOS Active	.3	.3	.3	.3	.3
SLS Active	.4	.4	.4	.4	.4
Reserved Input 1	.5	.5	.5	.5	.5
SLP Active	.6	.6	.6	.6	.6
Int. Event Active	.7	.7	.7	.7	.7
	IB6749	IB6701	IB6717	IB6733	IB6765
Reserved In 2	.0	.0	.0	.0	.0
SLS bit 0 Active	.1	.1	.1	.1	.1
SLS bit 1 Active	.2	.2	.2	.2	.2
Reserved Input 3	.3	.3	.3	.3	.3
SDI Pos. Active	.4	.4	.4	.4	.4
SDI Neg. Active	.5	.5	.5	.5	.5
Reserved Input 4	.6	.6	.6	.6	.6
SSM n>nx	.7	.7	.7	.7	.7

Drive to PLC (Telegram S_ZSW2 Status words 3 & 4)					
Function	Sp IB6750	X IB6702	Y IB6718	Z IB6734	A IB6766
Reserved Input 5	.0	.0	.0	.0	.0
Reserved Input 6	.1	.1	.1	.1	.1
Reserved Input 7	.2	.2	.2	.2	.2
SLP Range Active	.3	.3	.3	.3	.3
Reserved Input 8	.4	.4	.4	.4	.4
Reserved Input 5	.5	.5	.5	.5	.5
Safe Pos. Valid	.6	.6	.6	.6	.6
Safely Referenced	.7	.7	.7	.7	.7
	IB6751	IB6703	IB6719	IB6735	IB6767
F-DI 0	.0	.0	.0	.0	.0
F-DI 1	.1	.1	.1	.1	.1
F-DI 2	.2	.2	.2	.2	.2
SS2E ESR	.3	.3	.3	.3	.3
SS2E Active	.4	.4	.4	.4	.4
SOS Selected	.5	.5	.5	.5	.5
SLP Upper Limit	.6	.6	.6	.6	.6
SLP Lower Limit	.7	.7	.7	.7	.7

The signals from PROFIsafe telegram 701 are transferred into the relevant Axis interface PLC DBs 31/32/33/34/35 etc

The PLC interface can then be used to initiate and start the Safe Brake Test and also the Test Stop (Forced checking procedure. Each axis has a data block reserved for safety integrated signals to/from the PLC and drive.

Signals from the PLC to the drive (Safety Command Channel):

DB31 ...DB61	Signals from PLC to Drive (Safety Command Channel—SCC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB140								Start Test Stop
DBB143			Brake Closed	Torque 1 or 2	Direction of Rotation	Brake 1 or 2	BT Start	Selection of SBT

Signals from the drive to the PLC (Safety Information Channel):

DB31 ...DB61	Signals from Drive to PLC (Safety Information Channel—SIC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB166			Test Stop Required	Test Stop Active				
DBB173	Sign Load Torque	Brake Req.	B test finished	B test OK	B test Active	Brake 1 or 2	B test Start	Selection Safe BT

Notes:

Safety Integrated PLC Logic

The F-PLC logic should be written to achieve the functionality of the OEM function table. The program is written in the program editor in fail-safe FBs, FCs and DBs **in the programming language LAD or FDB.**

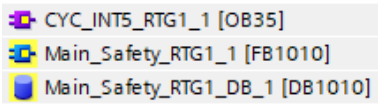
Note: the logic examples shown is this document are for training purposes. Logic should be written for each specific application. It is the responsibility of the OEM to ensure that the logic meets the required safety standards.

Runtime Group 1 (RTG1)
Safety specific PLC blocks are used to for the F-PLC logic. These were created when the option for Safety was activated on the PLC CPU. These are part of Runtime Group 1 (RTG1).

- A maximum of 2 Runtime Groups can be specified

The Runtime Group consists of:

- CYC_INT5_RTG1 [OB35] - Cyclic OB for Safety.
- Main_Safety_RTG1 [FB1010] - Function Block for the F-PLC logic.
- Main_Safety_RTG1 [DB1010} - Instance Data Block for FB1010.



Number range reserved for RTG1 Sinumerik:

- F-FB 1010...1019
- F-FC 1010...1019
- F-DB 1010...1019

Data exchange between standard user program and safety program

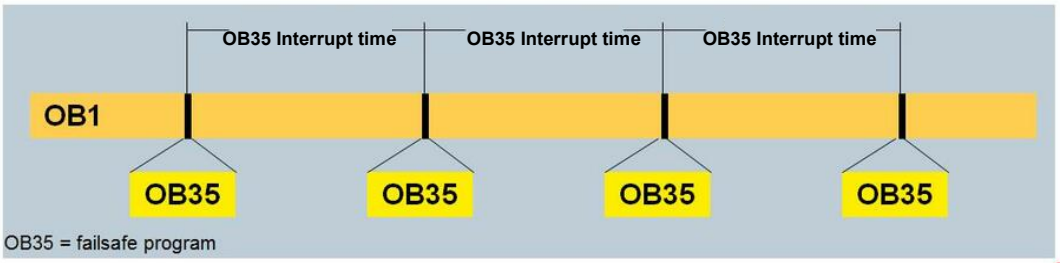
	From the standard user program		From the safety program	
	Read access	Write access	Read access	Write access
Variable from DB	permissible	permissible	Either read access *OR* write access to a variable from the DB	
Variable from F-DB	permissible	**not permissible**	permissible	permissible
Bit memory	permissible	permissible	Either read access *OR* write access to a bit memory	

		From the standard user program		From the safety program	
		Read access	Write access	Read access	Write access
Process image Standard I/O	PAE	permissible	permissible	permissible	**not permissible**
	PAA	permissible	permissible	**not permissible**	permissible
Process image F-I/O	PAE	permissible	**not permissible**	permissible	**not permissible**
	PAA	permissible	**not permissible**	**not permissible**	permissible

WARNING

Since variables from the standard user program are not generated safely, it must be ensured with the implementation of additional process specific plausibility checks in the safety program that there is no risk of hazardous conditions developing.

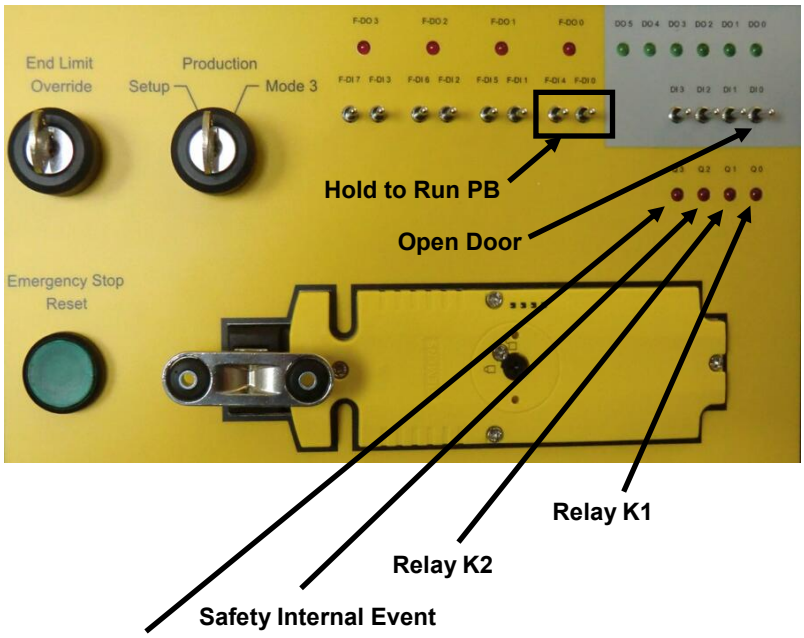
OB35 Cycle Time;
The response time/performance is dependant on the number of axes/safety functions in use :



The default OB35 interrupt time is 100ms. This can be optimised to a lower value and will be dependant on the number of Safety Integrated axes/inputs/outputs/functions in use.

Notes:

Safety Simulator
Real Sinumerik ONE



Safety Simulator
Virtual Sinumerik ONE

PLC I/O table

Symbolic Name	Address	I/O	Initial Value	Comment
fastStopOff3	%I80.0	0 1	0 1	
fastStopOff3toNCU	%Q80.0			
contrlEnable	%I80.1	0 1	0 1	
pulseEnable	%I80.2	0 1	0 1	
posMeas1	%I80.3	0 1	0 1	
posMeas2	%I80.4	0 1	0 1	
delayRef	%I80.6	0 1	0 1	
mainAirPressureLow	%I80.7	0 1	0 1	
				safetyIntegrated
EmergencyStopPB	%I512.0	0 1	0 1	
HoldToRunButton	%I524.0	0 1	0 1	
DoorLocked	%I512.2	0 1	0 1	
DoorClosed	%I512.3	0 1	0 1	
EmergencyStopResetPB	%I518.1	0 1	0 1	
SetupSwitch	%I518.2	0 1	0 1	
ESTOPResetRequired	%Q530.3			
Internal Event Lamp	%Q530.2			

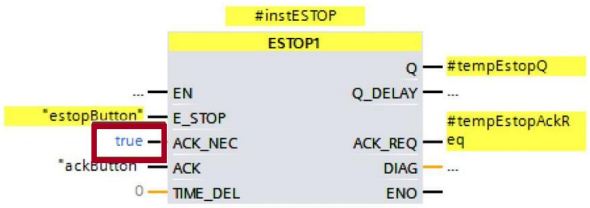
Test Stop

User key Test Stop is used to start the test stop routine

F-PLC Programming logic operations

Recommendations

- Use mainly AND and OR logic elements
- Keep the use of SR blocks to a minimum
- Avoid jumps in binary logic
- Keep the number of timer blocks to a minimum



True & False

Regarding the use of "TRUE" and "FALSE" signals in safety programs, there are two different use cases:

- Actual parameters on blocks
- Assignments on operations

Actual parameters on blocks

For S7-1200/1500 controllers, you can use the Boolean constants "FALSE" for "0" and "TRUE" for "1" as actual parameters for interconnecting formal parameters during block calls in the safety program. Only the keyword "FALSE" or "TRUE" is written to the formal parameter.

Assignments on operations

To generate "TRUE" / "FALSE" signals for operations, proceed as follows:
1. Create two static tags, "statTrue" and "statFalse", of the Bool data type.
2. Assign the default "true" to the "statTrue" tag.
3. Assign the default "false" to the "statFalse" tag.

	Name	Data type	Start value	Retain
1	Static			
2	statTrue	Bool	true	
3	statFalse	Bool	false	

Notes:

Data exchange between standard user program and safety program

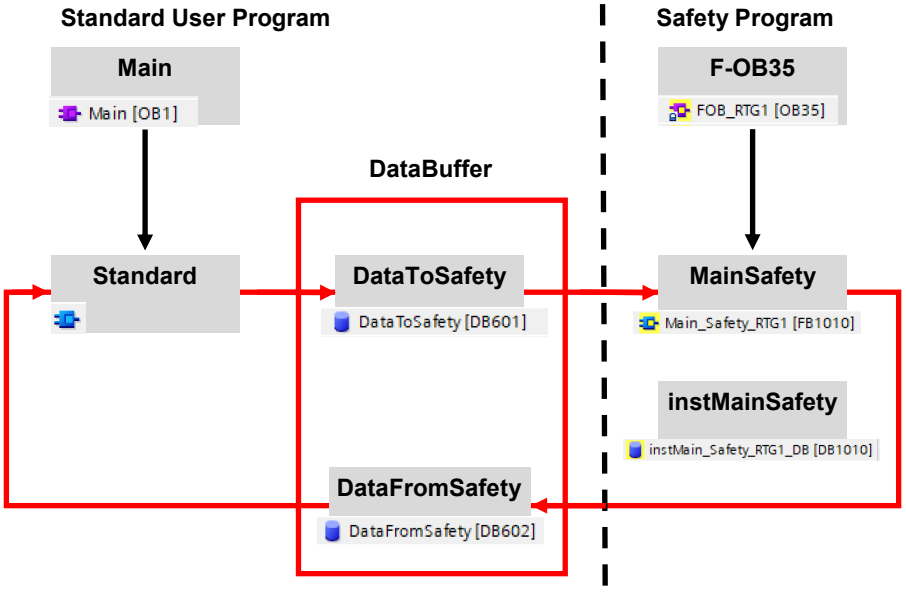
The safety program's task is to execute all the functions that represent a risk-reducing action. All other operational functions and functions for operation and maintenance are part of the standard user program.
In practice, information for the diagnostic and signaling concept is also generated in the safety program and operational information is also relevant to the safety program. As a result, it is not possible to completely separate both program parts.
In order to move non-safety-related functions to the standard user program, you have to define an interface. Global data blocks are best suited for this purpose.

Recommendation

Use global standard data blocks to exchange data between the standard user program and the safety program.
To ensure a good overview of which program part reads and which one writes, it is recommended to create two data blocks for the two directions.
You should not store any other information (e.g., diagnostic data from the standard user program) in the data blocks as each modification of the data block involves a modification of the safety program.

Using non-safe inputs in the safety program

Standard inputs that are required directly in the safety program must be read directly in the safety program. A "detour" via the standard user program should be avoided.
The background to this is that non-safety-related signals are also included in the application's systematic integrity. Typical examples are acknowledgment / reset buttons or mode selectors. Which button / switch is allowed to reset which safety function is a direct result of the risk assessment. A change of the command devices must therefore influence the signature and must be made only accompanied by a reassessment and an acceptance test for changes.



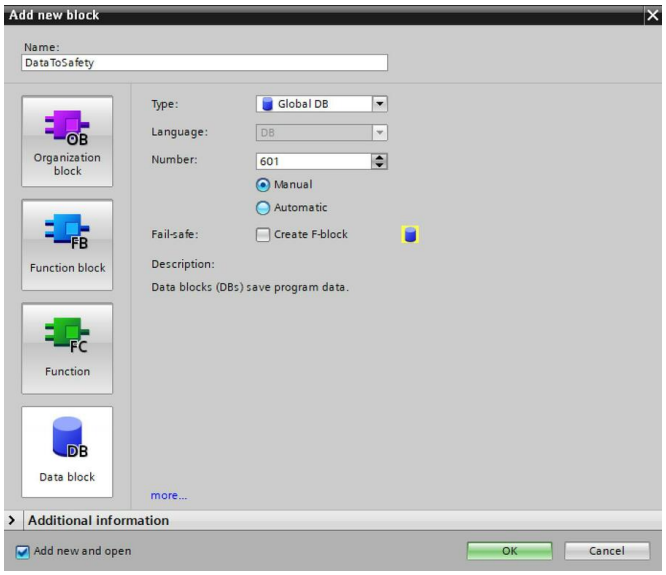
Data exchange between standard user program and safety program

Notes:

User Data Blocks for DataBuffer e.g. DB601 & DB602

Two user data blocks are required to exchange data between the standard user program and the safety program.
In this example DBs 601 and 602 are being created as Global Data Blocks.

- Create DB601 DataToSafety as a Global Data Block
- Create the static variables as shown on the next page.



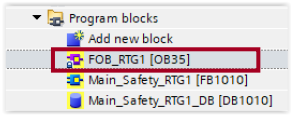
DataToSafety

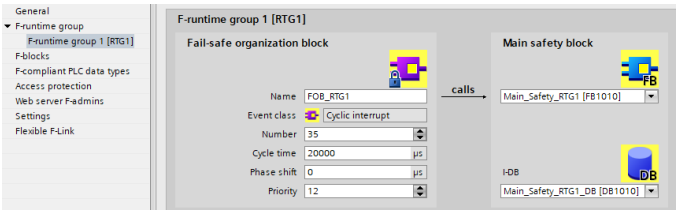
		Name	Data type
1		Static	
2		K1RelayTest	Bool
3		K2RelayTest	Bool
4		ChannelReset	Bool
5		FlashLight	Bool

Global Data Block DB602 DataFromSafety can also be created although in this example we are not using it.

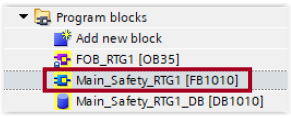
Notes:

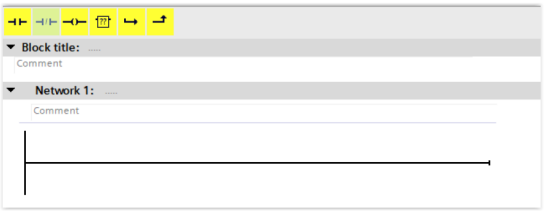
Safety Organisation Block - OB35 shown in the Runtime Group with a call to FB1010/DB1010.





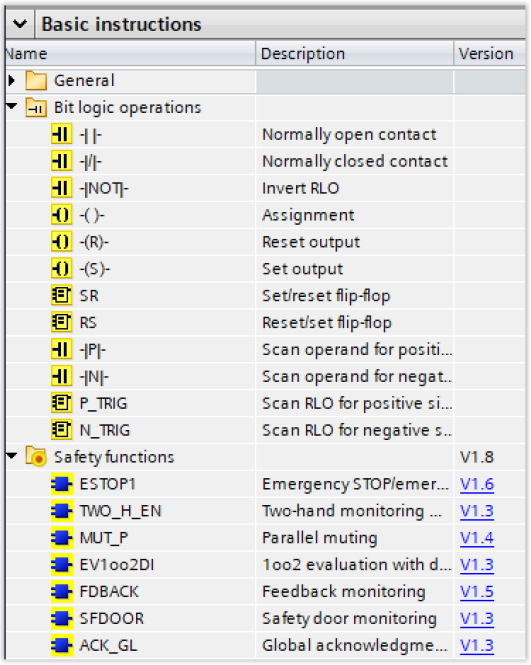
Safety Function Block - FB1010 which will contain the F-PLC logic. This can only be written in **ladder** or **function block diagram**.





Basic PLC Instructions

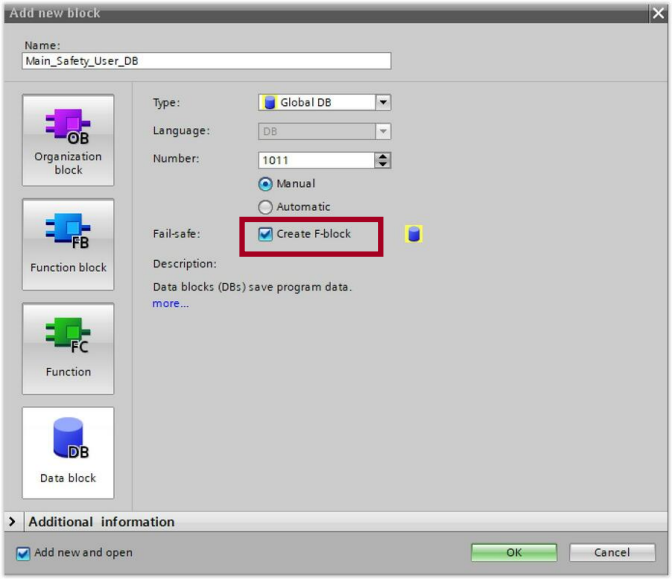
The logic for the various safety functions can now be added. The instructions are found in the basic Instructions tab. These instructions have a yellow background when they are safety relevant



User Data Block e.g. DB1011

A user data block is required for handling the safety tags required in the F-PLC logic.
In this example DB1011 is being created as an F-Block.

- Create DB1011 as an F-Block
- Edit the Tag content as shown on the next page.



Main_Safety_User_DB

	Name	Data type	Start value
1	Static		
2	statTrue	Bool	true
3	statFalse	Bool	false
4	AxesEstopOK	Bool	false
5	AxesEstopDelayOK	Bool	false
6	AxesEstopAckReq	Bool	false
7	SpinEstopOK	Bool	false
8	SpinEstopDelayOK	Bool	false
9	SpinEstopAckReq	Bool	false
10	AllAxesSOS	Bool	false
11	SpindleSOS	Bool	false
12	EstopToNC	Bool	false
13	AxesSSM	Bool	false
14	SpindleSSM	Bool	false
15	AxesNoInternalEvent	Bool	false
16	SpindleNoInternalEvent	Bool	false
17	EstopReleased	Bool	false

Compile the project to activate the new DB tags

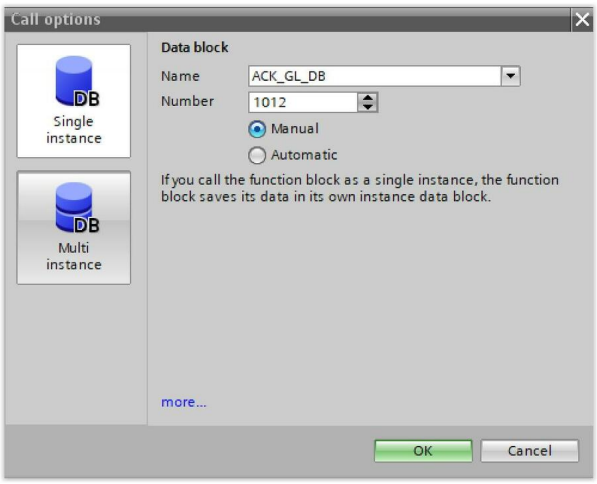
Notes:

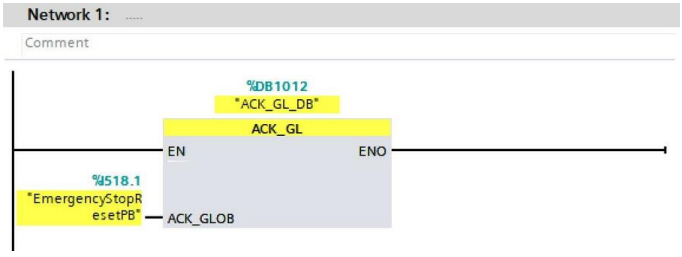
Safety Global Acknowledgement Block - ACK_GL

An Acknowledgement of any safety alarms is achieved by first calling the ACK_GL system function block in Network 1 of FB1010.

The function can be called by using drag and drop from the Instructions task card to FB1010.

An instance DB will also need to be created and in this example we will use DB1012

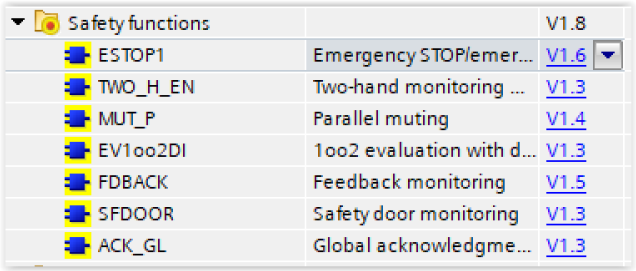




Use the Emergency Reset Pushbutton (I518.1) as the acknowledgement signal.

Emergency Stop

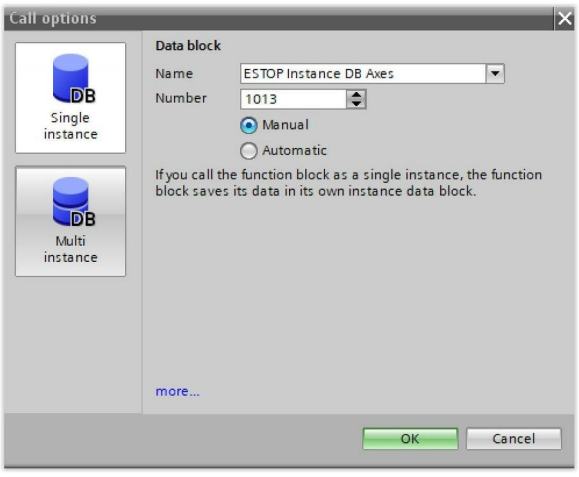
The ***ESTOP1*** safety function is available for handling a category 1 emergency stop request



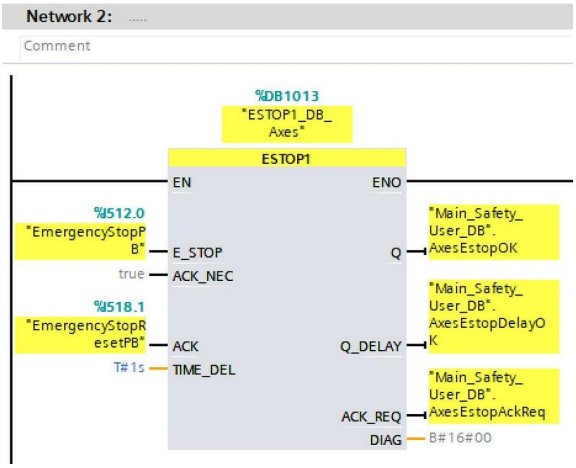
The ESTOP1 system function block needs to be called twice in FB1010, once for the axes (network 2) and once for the spindle (network 3). This is achieved by using drag and drop from the instructions task card.

Each call of the ESTOP1 function will require an instance DB. In this example we will use DB1013 and DB1014.

The Instance DBs should be given useful names e.g. ESTOP Instance DB Axes and ESTOP Instance DB Spindle



The ESTOP1 system function block is parameterized for use with the axes in network 2 of FB1010.



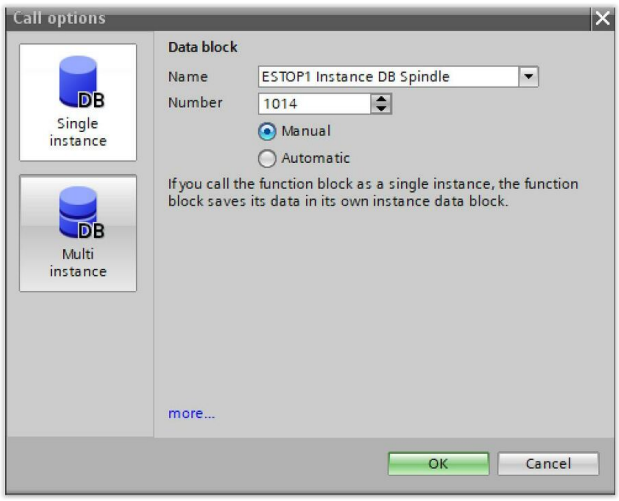
The Tags on the input side should be given useful names and a time delay entered for maximum axes braking ramp.

The Outputs “AxesEstopOK” and “AxesEstopDelayOK” should be True for a non ESTOP condition

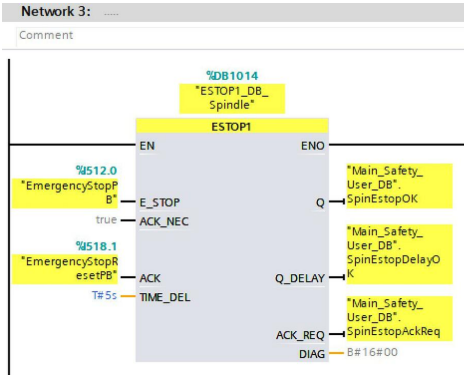
“AxesEstopAckReq” output will be True if an acknowledgement is required after an ESTOP condition.

Notes:

Creation of the Instance DB for the 2nd call of the ESTOP1 System function block.



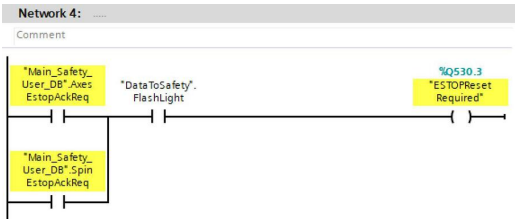
The ESTOP1 system function block is parameterized for use with the spindle in network 3 of FB1010.



The delay time is usually longer in this case to allow the Spindle to brake in the case of an ESTOP condition

Emergency Stop Acknowledgment Required:

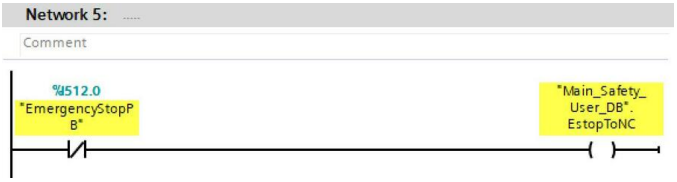
If an axis/spindle ESTOP acknowledgment is required then output Q530.3 will flash.
Pressing the “Emergency Stop Reset PB” will reset the Emergency Stop condition.



Emergency Stop signal to the NC:

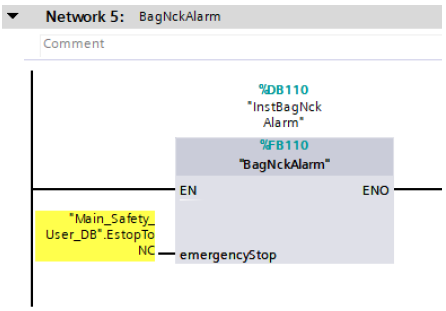
To initiate an NC Emergency Stop in the NC it is necessary to transfer the signal from the F-PLC logic to the relevant part of the non safety PLC program.

Example: The Emergency Stop button controlling a bit in the ”Main Safety User DB” 1011.



This Main Safety User DB bit “EstopToNC” is then used/ edited in OB1 in the non safety part of the user PLC.

OB1 in this example



“Main Safety User DB” bit then being used in the non safety part of the user PLC program to replace the signal generating the NC ESTOP interface signal

Emergency Stop Status:

Checking the status of the axes and spindle Emergency Stop and creating an additional Tag for all conditions OK.:

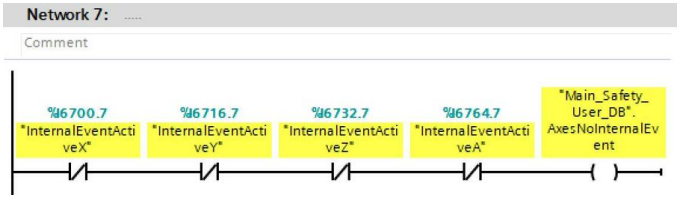


Notes:

Internal Event - Axes/Spindle:

An internal event is triggered by Safety if there is a fault condition.

Axes internal event:

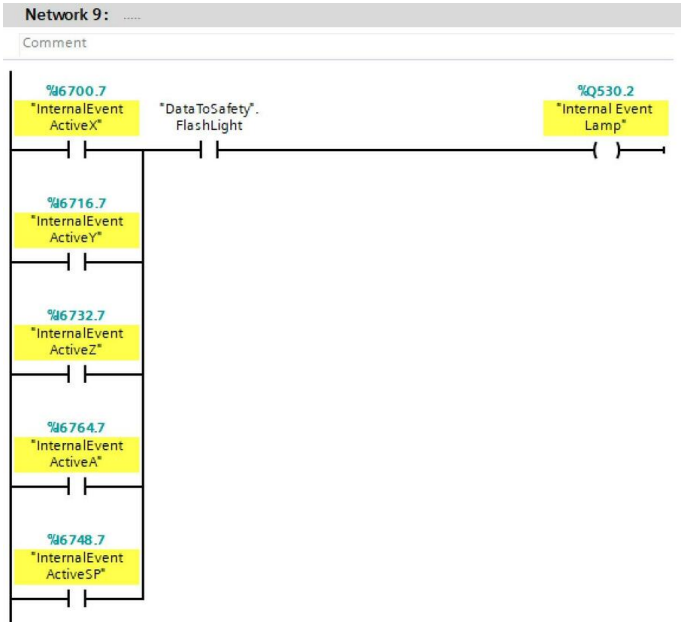


Spindle Internal event



Internal Event Active Lamp :

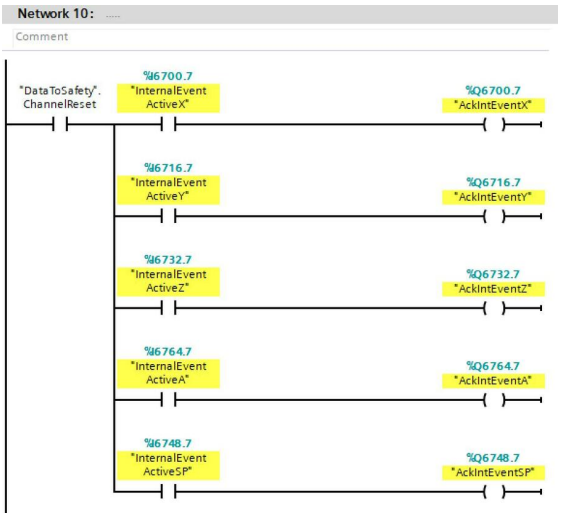
The Internal Event lamp is illuminated flashing if there are any safety errors.



Acknowledging internal events:

If a STOP response is acknowledged in a safety-related fashion by the machine operator, but without PROFIdrive RESET acknowledgment, then the part program is not canceled and is continued with the acknowledgment. When the machine restarts unexpectedly, this can put people in dangerous situations or cause material damage.

- In the safety program logically combine safe acknowledgment (PROFIsafe Internal Event ACK) with a RESET at the machine control panel (PROFIdrive RESET).



Deselect Safe Stop 1 (SS1)

In the case of the Emergency Stop button being pressed the relevant axes/spindle signals will be deselected as per networks 2/3

Deselect Axes SS1



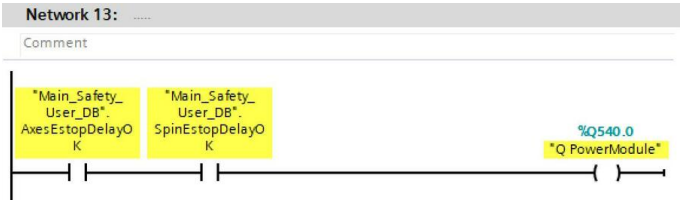
Deselect Spindle SS1



Notes:

Safety Power Module

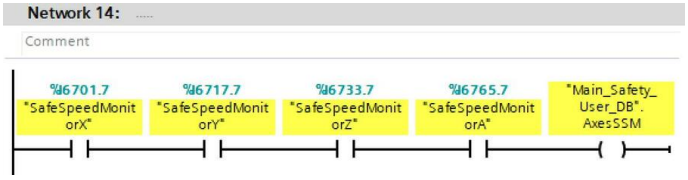
To transfer power to the non-safe outputs in slot 8 the Power Module (PME) output has to be set (Q540.0). Then in the case of an Emergency Stop condition power is removed to the non safe outputs of the module in slot 8.



Safe Speed Monitoring (SSM)

The axes and spindle SSM signals are checked to ensure that they are below the safe speed i.e. stationary

Axes SSM:



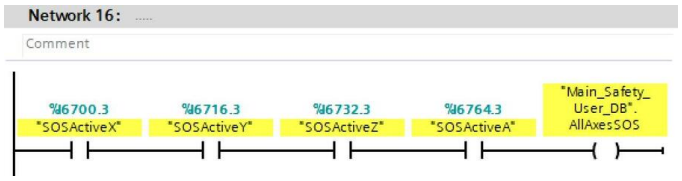
Spindle SSM:



Monitor Axes Safe Operational Stop (SOS)

The axes and spindle SOS signals are being checked individually. With SOS active the Axes/Spindle are monitored for any uncontrolled movement.

Axes SOS:

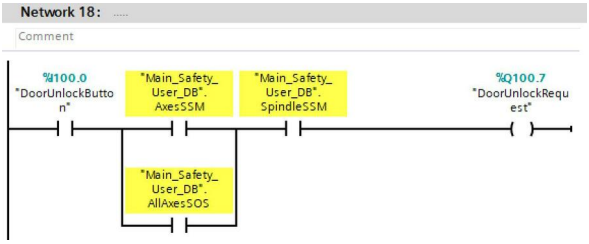


Monitor Spindle SOS:



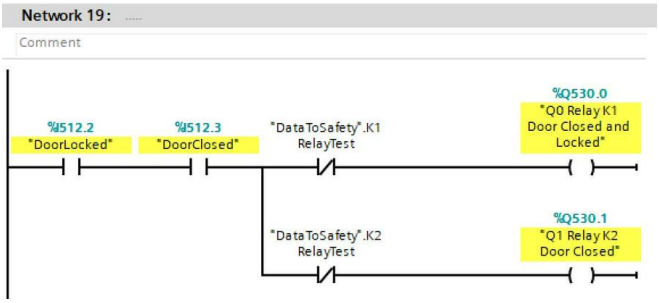
Door Unlock

The door can be unlocked when all axes/spindle have SOS or SSM active. i.e. all axes and spindle are stationary



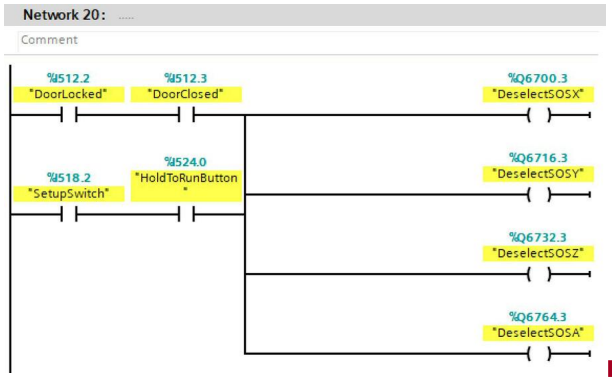
K1& K2 Auxiliary relays

The K1 and K2 relays are connected to two safe outputs Q530.0 & Q530.1.
These could be used for example to inhibit the movement of equipment such as a turret or tailstock.
In this example they are controlled by the door closed/locked switches.
“K1/K2RelayTest” signals from DB601 DataTo Safety are created later in the non safety user PLC.



Deselect Safe Operational Stop (SOS)

Axes: Safe Operational Stop is deselected for the axes when the door is closed and locked or Set-up Mode and Hold to Run Pushbutton is pressed.



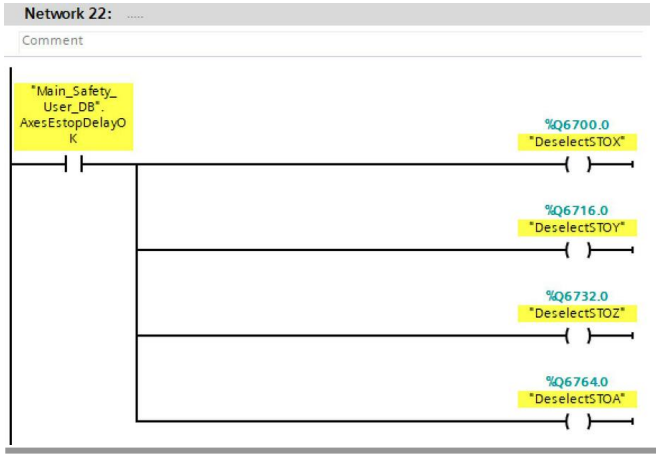
Notes:

Spindle: Safe Operational Stop is permanently deselected for the spindle (refer to the function table).

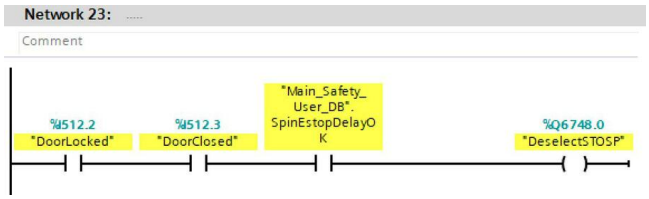


Deselect Safe Torque Off (STO)

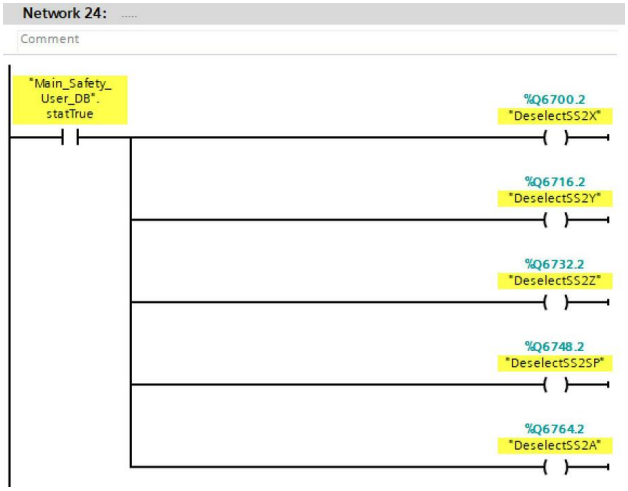
Axes: Safe Torque Off is deselected for the axes when the Emergency Stop pushbutton is pressed and the delay time "AxesEstopDelayOK" from network 2 has expired.



Spindle: Safe Torque Off is deselected for the spindle when the Emergency Stop is pushbutton is pressed and the door is locked/closed. (after delay time in network 3)



Deselect Safe Stop 2 (SS2) This feature is not used.

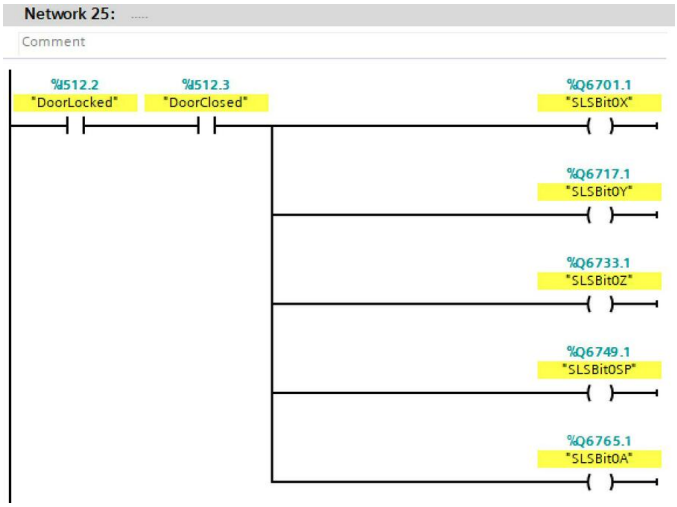


Safe Limited Speed

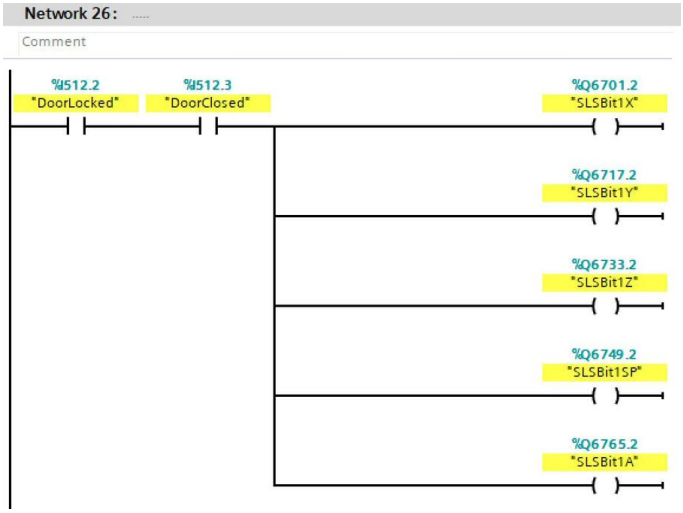
The door status is used to select SLS1 (door open) & SLS4 (door closed)

SLS Level	SLS bit 0 =	SLS bit 1 =
1	0	0
2	1	0
3	0	1
4	1	1

SLS bit 0

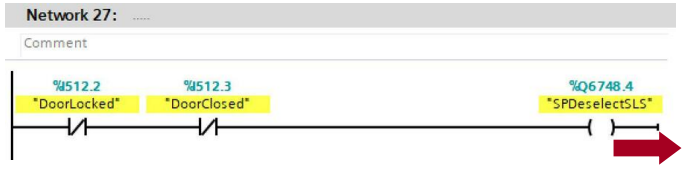


SLS bit 1



Spindle Deselect SLS

Spindle SLS is deselected if the door is open.



Notes:

The PLC project should now be compiled and downloaded

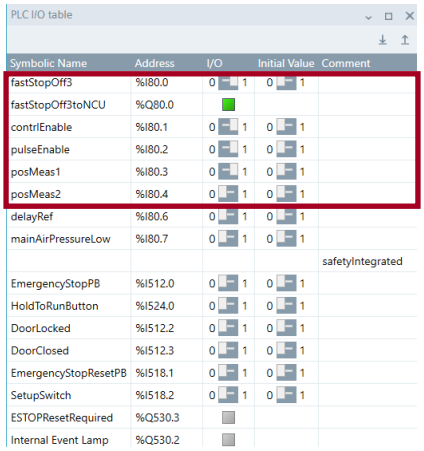
Checking the logic:
The logic for the various switches can be checked using the OEM function table and the diagnostic screens.
This will be further tested as part of the Safety Integrated Acceptance Test (SIAT).

Operating mode	*Door*	*Axes / spindle*	*SLS level*
Auto	Closed	Axes	SLS 4
		Spindle	SLS 4
	Open	Axes	SOS
		Spindle	STO
Setup	Closed	Axes	SLS 4
		Spindle	SLS 4
	Open	Axes	SOS or SLS1 With Hold-To-Run PB
		Spindle	STO

If on a Virtual Sinumerik ONE the PLC I/O Table is too small, then Float should be used to create/manage a bigger table



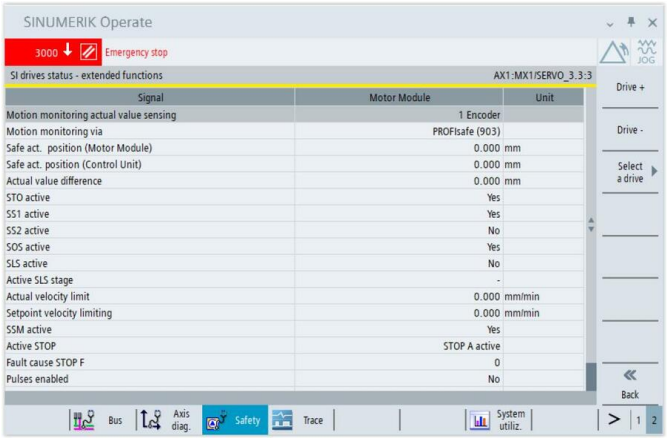
The I/O can now be configured in the table, and then used to create the different conditions as shown in the logic table.



The normal axis enabling switches should be switched on first

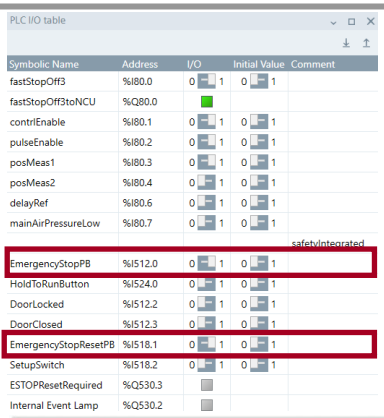


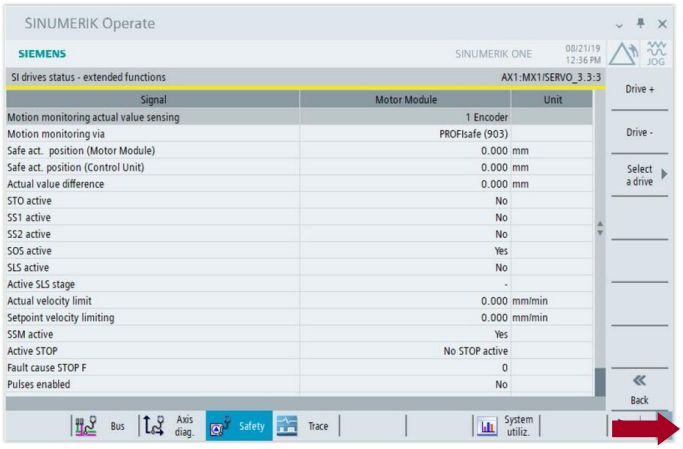
In the diagnostic screens it can be seen that after download/ simulation reset, all axes/spindle are shown with SS1, STO.



Use the sequence %I512.0 ON, Reset PB on MCP, %I518 ON/OFF to reset the Emergency Stop condition.

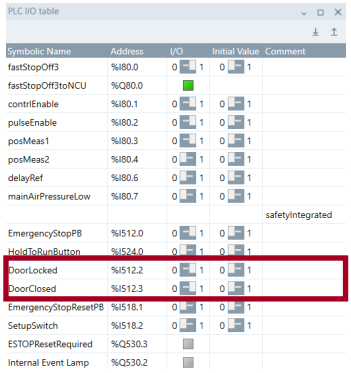
The Emergency Stop Message should disappear and then the Axes should show only SOS and the Spindle should show STO.



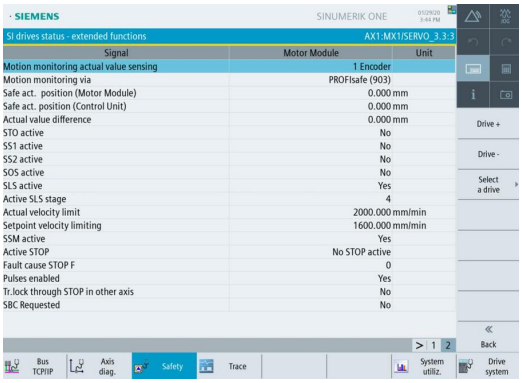


Notes:

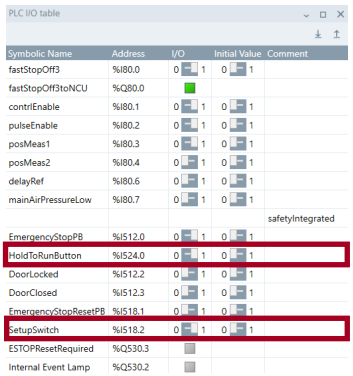
Switching ON %I512.2 and %I512.3 means that the Operator Door is closed and locked. This means that all axes and spindle should switch to SLS4.

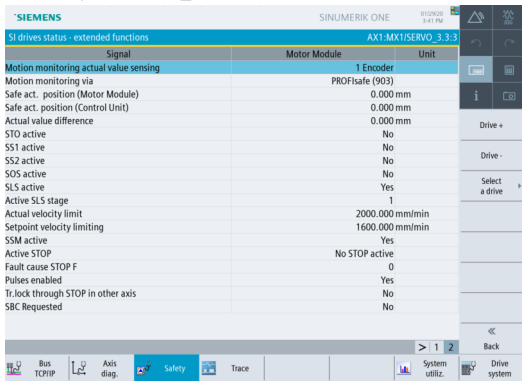


At this stage the SLS setpoint velocity limitation is shown as 1600 mm/min because these limits have not been set up yet. The default SLS speeds are 2000 mm/min for all 4 SLS values therefore with 80% setpoint limitation a value of 1600 mm/min is correct.



With the Operator Door Open, then %I518.2 ON and %I524.0 ON all Axes are showing SLS1. This simulates the Keyswitch being in Setup Mode and the Hold to Run PB being pressed.

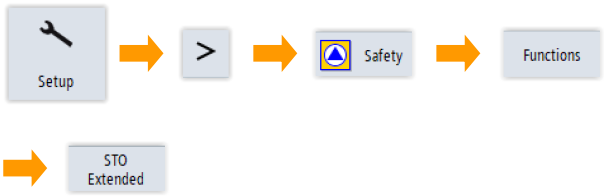


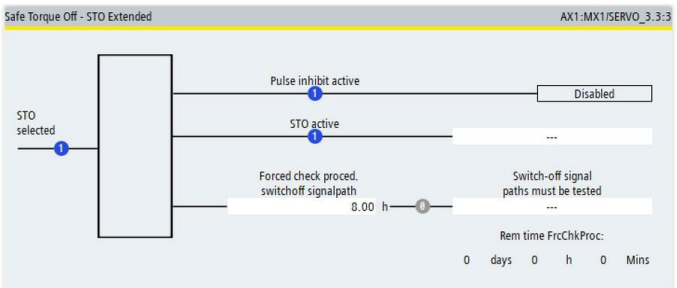


The commissioning now continues on the HMI in the Setup/ Safety area.

The PLC will be modified later as more Safety functionality is commissioned.

Safe Torque Off-Extended (STO Extended)





The ***Safe Torque Off - Extended*** function allows for the STO status of each safe drive to **be monitored, via a BiCo**, in the CU or any of the drives.

Example: the signal could be used to set a digital output on the CU.

It is also possible to set a time period for the ***Forced check proced. Switchoff signalpath.*** (Test stop interval time) and, when the time value has expired, set a CU/Drive BiCo via the ***Switch-off paths must be tested*** area.

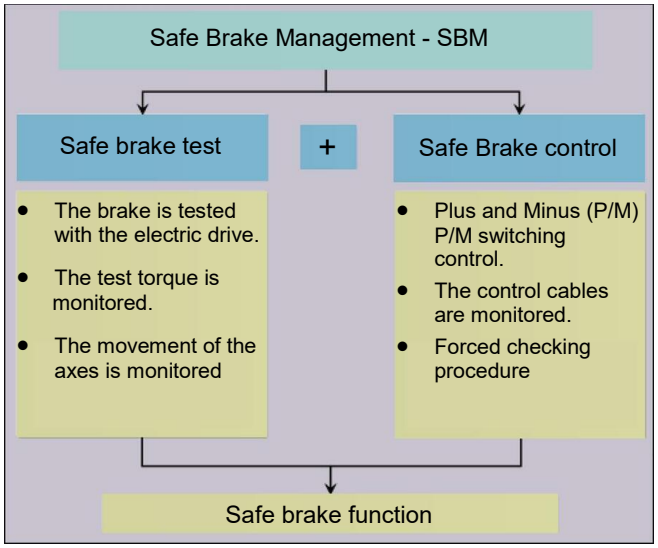
Alarm, ***201699 - Axis ??, component ??: SI P1(CU) Switch -off signal paths must be tested.***, will also be displayed. This will clear once the test stop has been completed. The ***Rem time FrcChkProc*** will be also reset.

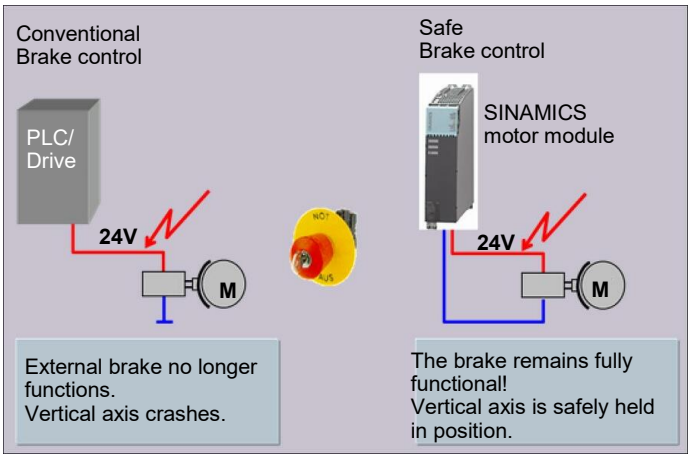
Note: The default settings can be used in this example.

Notes:

Safe Brake Management (SBC not on virtual ONE)

Safe Brake Control (SBC) and Safe Brake Test (SBT), together form Safe Brake Management and is available for axes fitted with mechanical brakes (integral motor brakes or external brakes).

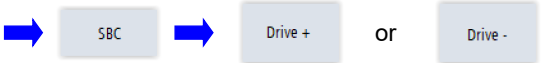


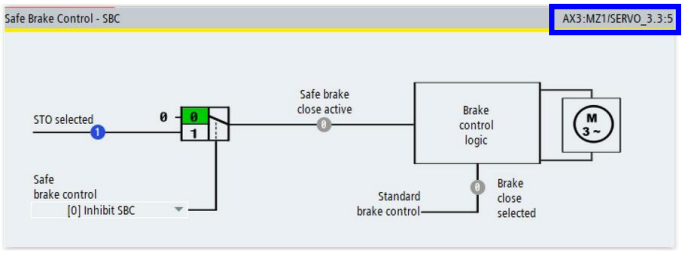


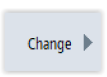
Safe Brake Control - SBC

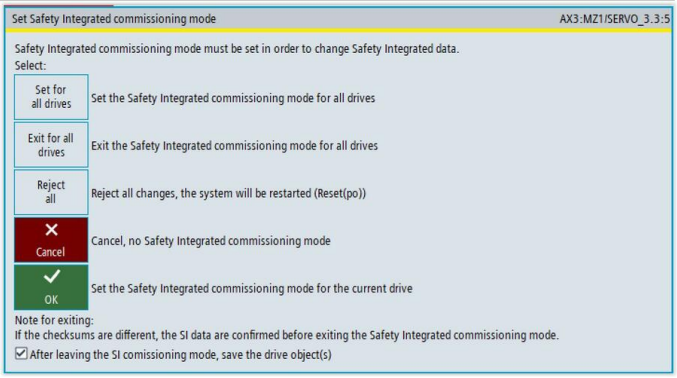
The training equipment ***Z axis* is fitted with a brake**, therefore SBC can be enabled.

- Disable the drives.
- Select the ***Z axis servo***







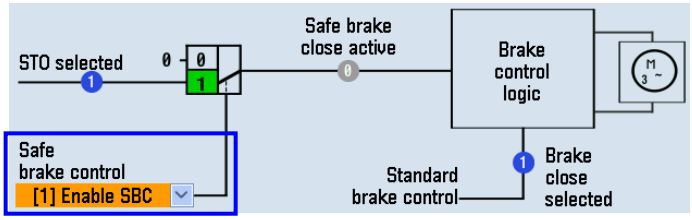


Only one drive is being changed, therefore select OK.



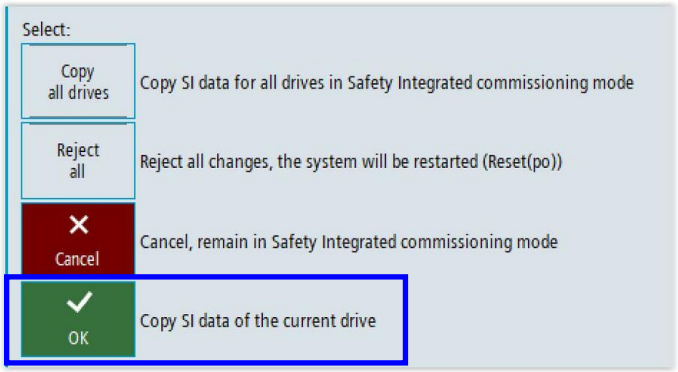
Notes:

Enable the SBC:



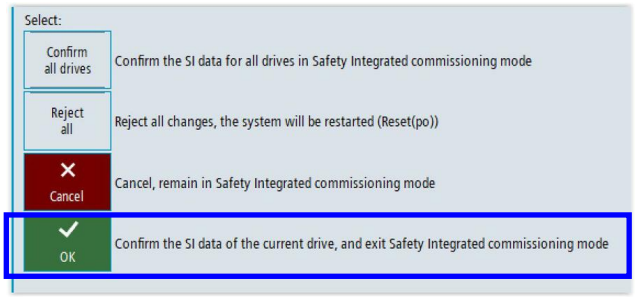
OK

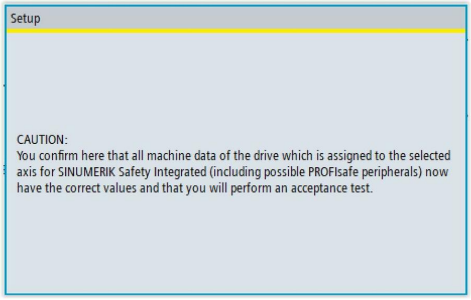
The changes need to be copied:



OK

The changes need to be confirmed





OK

Safe Brake Control is now enabled.

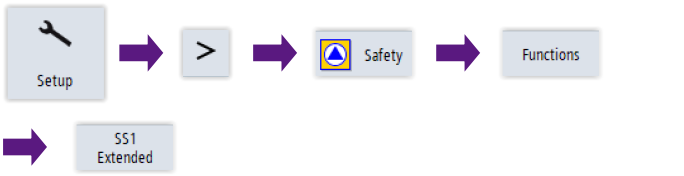
The Safe Brake Test is described in the ***Test Stop*** section

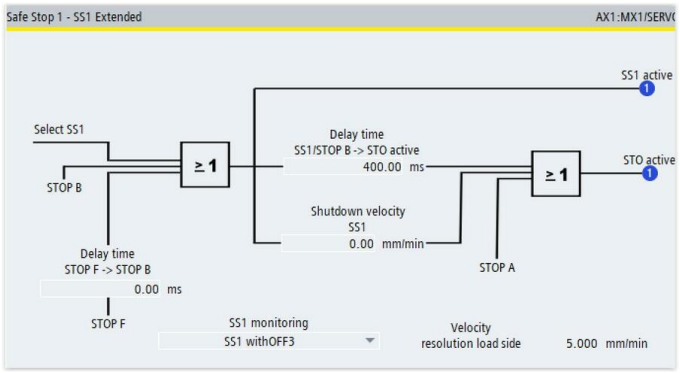
Notes:

SS1 Extended (Safe Stop 1 - Extended)

The **SS1 Extended** parameters need to be adjusted to ensure the correct transition between an active STOP/SS1 and STO.

Example: When SS1 is activated It is important to allow the drives to decelerate under current limit, to standstill, before applying Safe Torque Off. Axes with a large mass, e.g. a large chuck on a turning machine, may need several seconds to reach standstill from maximum speed.





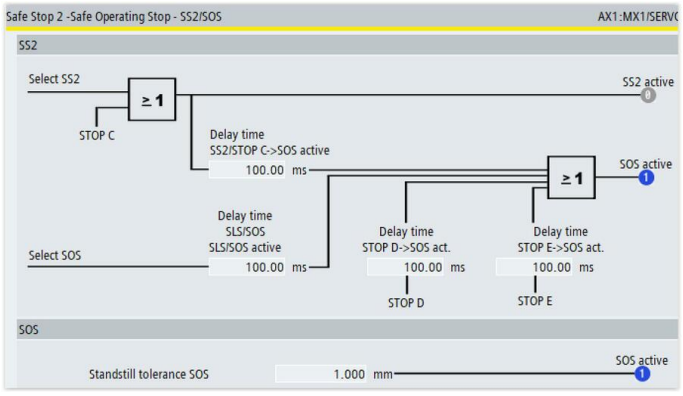
Parameter	Description
Delay time: STOP F to STOP B	Transition time from STOP F to STOP B (ms)
Delay time: SS1/STOP B to STO **E.g. for axes 500ms** **For spindles 5000ms**	Transition time from SS1/STOP B to STO (ms). Axes with long deceleration times need to come to a halt using current limit before STO is applied.
Shutdown velocity	A velocity can be set, beneath which the axis is assumed to be at standstill and STO is then applied. Note: Only if **SS1 With OFF3** is selected
SS1 monitoring	**SS1 With OFF3:** Safe Stop 1 in accordance with EN60204, stop category 1. **SS1E external stop:** Safe Stop 1 with external stop. This requires an externally initiated stop in order to be in compliance with stop category 1.

SS2/SOS (Safe Stop 2/Safe Operating Stop)

The **SS2** function results in an SOS condition. This is not suitable for an Emergency Stop situation.

The transition time, from an **SOS** request, or from STOPs C, D or E, can be adjusted, depending on the application.





Parameter	Description
Delay time: STOP C to SOS	Transition time from STOP to SOS (ms)
Delay time: SOS request to SOS active	Transition time from SOS request to SOS active. Training equipment: when the door is opened, SOS is requested on X, Y, Z & A axes.
Delay time: STOP D to SOS active **If errors occur regarding STOPD then this time should be increased**	Transition time from STOP D request to SOS active.
Delay time: STOP E to SOS active	Transition time from STOP E request to SOS active.
Standstill tolerance SOS	Determines how far an axis can move, when SOS is active, before the system triggers an alarm.

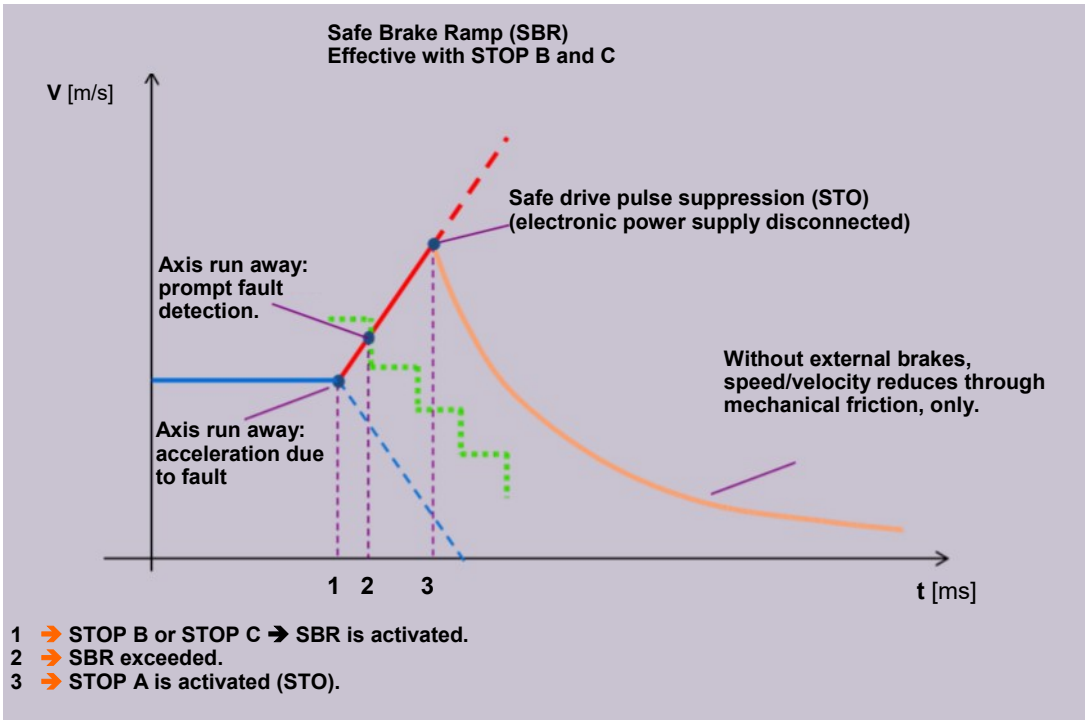
Notes:

Safe Acceleration Monitoring/Safe Braking Ramp

The **Safe Acceleration Monitoring (SAM)** function is responsible for monitoring the braking along **OFF3** ramp.

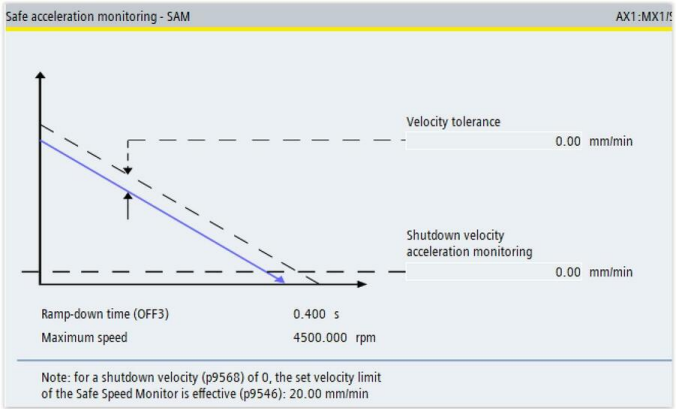
This was the default choice during the enabling of Safety Integrated on the drives.

The function is active for SS1, SS2, STOP B & STOP C.
If the motor does not decelerate, as expected, the system selects STO which results in the motor coasting to a standstill.



Setting the SAM:

SAM/SBR



Parameter	Description
Velocity tolerance (p9548)	The deviation from the contour for an axis undergoing an ***OFF3*** ramp down.
Shutdown velocity acceleration monitoring. (p9568)	During ramp down, the level at which the SAM ceases to be active. Note: If set to 0, the value in the Safe Speed Monitoring will be used.

Calculating the Safe Acceleration Monitoring:

SAM = Acceleration * Safe Clock Cycle

Linear Axes:
SBR[mm/min] = a[m/s^2] * t_{safe} * 1000[mm/min] * 60[s/min]

Rotary Axes:
SBR[rev/min] = a[rev/s^2] * t_{safe} * 60[s/min]

E.g. Calculation of the X axis SBR:

Linear Axes Formula:
SBR[mm/min] = a[m/s^2] * t_{safe} * 1000[mm/min] * 60[s/min]

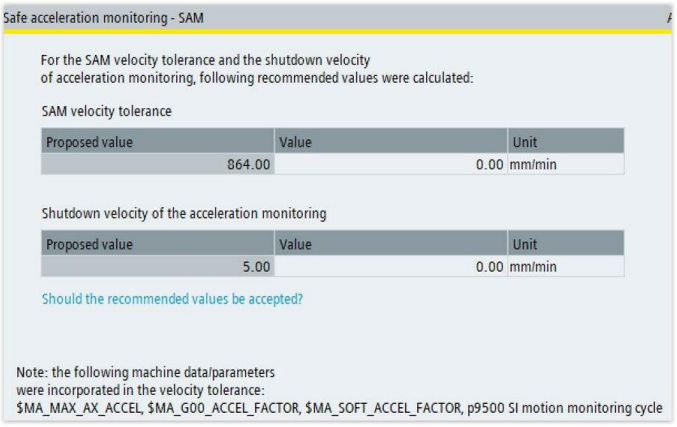
MD p9500 SI Motion monitoring clock cycle = 0.012s
MD 32300 $MA_MAX_AX_ACCEL = m/s^2 = 1

SBR Tol = (1 * 0.012 * (60*1000)) + 20% = **864 mm/min**

Notes:

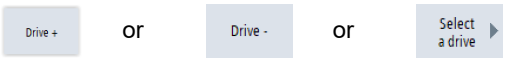
The SAM can be automatically calculated for linear and rotary axes, but ***not*** for the **spindle**:



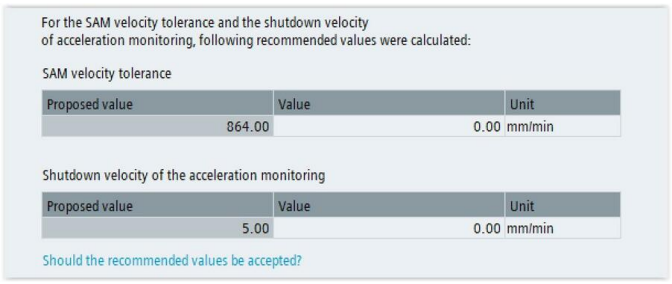


The proposed values can be accepted or adjusted if necessary.

Now repeat for all of the drives



Example SAM calculations for the X, Y & Z axes:



Spindle SAM calculation.

Description:
Following a SI stop request (STOP B/STOP C) the motor speed must be reduced.
If this fails to happen and the **SBR** setting is exceeded, the system issues a **STOP A** (STO - drive pulse suppression).
For axes without external brakes the stopping of the motors will be dependent on mechanical friction.

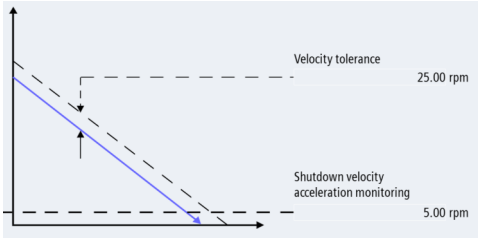
The system can calculate the SBR tolerance for axes, via a soft -key. The spindle, however, needs to be manually calculated using the following formula:

SBR[rev/min] = a[rev/s^2] * t_{safe} * 60[s/min] * 1.2

a[rev/s^2] = MD 32300 $MA_MAX_AX_ACCEL = 30m/s^2

t_{safe} = MD p9500 SI Motion monitoring clock cycle = 0.012s.

Therefore: 30 * 0.012 * 60 * 1.2 = 25.92
This needs to be manually entered as the SAM value for the spindle:

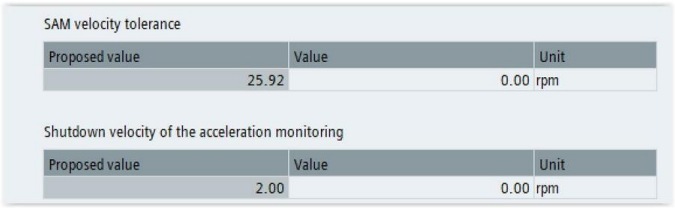


Example manually calculated SAM values entered for the Spindle:

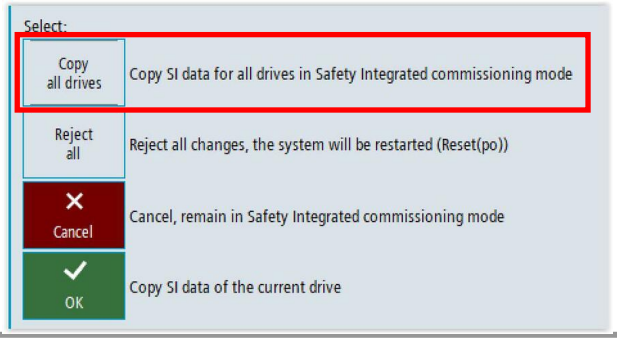
Also, depending upon the machine configuration, it may be necessary to consider other machine data:

MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL
MD 35210: GEAR_STEP_POSCTRL_ACCEL
MD 35410: SPIND_OSCILL_ACCEL

A Axis: example SAM calculations for the A axis:



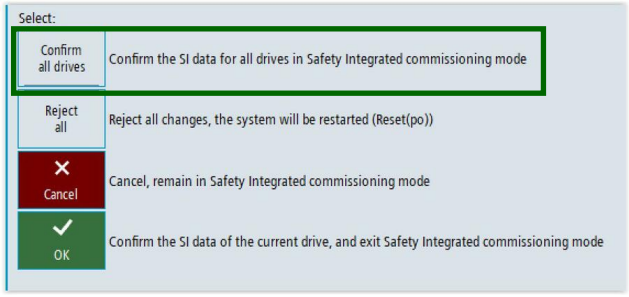
The changes need to be copied:

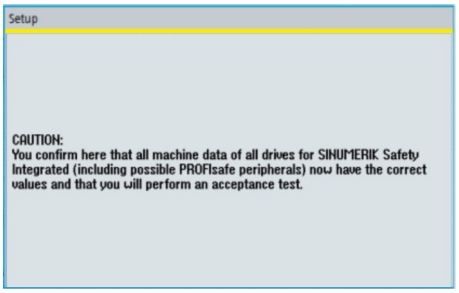


Notes:



The changes need to be confirmed





If alarm 22001 is displayed, the online help shows the possible causes of this. The alarm means that with the current MD settings the axis can not follow the SAM settings

22001 [Channel %1:] Block %2 Axis %3: Brake ramp longer than Stop D time. Reason: %4.

Parameters:
%1 = Channel number
%2 = Block number, label
%3 = Axis name
%4 = Identification of cause

Explanation: The current axis dynamic response is not sufficent to come to a standstill on time when a Stop D is triggered. The reasons specified in parameter 4 are:
1: MD32300 $MA_MAX_AX_ACCEL too small.
2. MD32431 $MA_MAX_AX_JERK too small.
3. Excessive acceleration reduction activated with ACC or subsequent to FXST.
4. Excessive jerk reduction programmed with JERKLIMA.

Reaction: Alarm display.

Remedy: Increase MD36953 $MA_SAFE_STOP_SWITCH_TIME_D. Increase MD32300 $MA_MAX_AX_ACCEL and MD32431 $MA_MAX_AX_JERK. Increase programmed acceleration (ACC) or jerk (JERKLIMA). Correct torque and acceleration reduction subsequent to FXST.
Alarm can be suppressed with MD11415 $MN_SUPPRESS_ALARM_MASK_2 bit 13.

Programm continuation: Clear alarm with the Delete key or NC START.

If the axis/spindle has been fully commissioned then if the alarms occur p 9553 should be increased under
SS2/SOS (Safe Stop 2/Safe Operating Stop)
See P84

Safely Limited Speeds

The Safely Limited Speed (**SLS)** function is used to protect the drives against unintentional high speeds in both directions.

There are four available SLS levels for each axis.

The system monitors the active SLS level and causes a STOP condition should it be exceeded.

It is possible to choose the STOP type when an SLS level is exceeded.

By default, all the axes have each SLS level set to 2000mm/min, for linear axes, and 2000 rpm, for rotary axes.

The active SLS level is controlled by the door and mode switches in accordance to the OEM function table

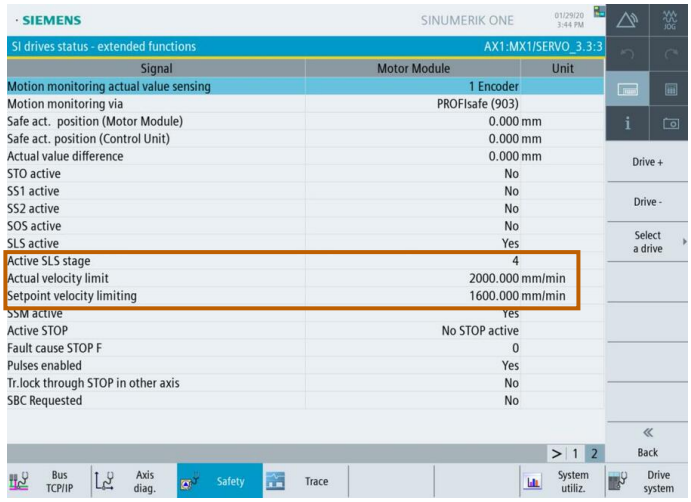
Function table example showing the SLS/switch relationship for the training equipment:

Operating mode	*Door*	*Axes / spindle*	*SLS level*
Auto	Closed	Axes	SLS 4
		Spindle	SLS 4
	Open	Axes	SOS
		Spindle	STO
Setup	Closed	Axes	SLS 4
		Spindle	SLS 4
	Open	Axes	SOS or SLS1 With Hold-To-Run PB
		Spindle	STO

Notes:

The current **Safe actual velocity limit** and the **Safe velocity limitation** can be seen on the diagnostics screen.
SLS is active with SLS stage 4 selected and the SLS speed is the default value

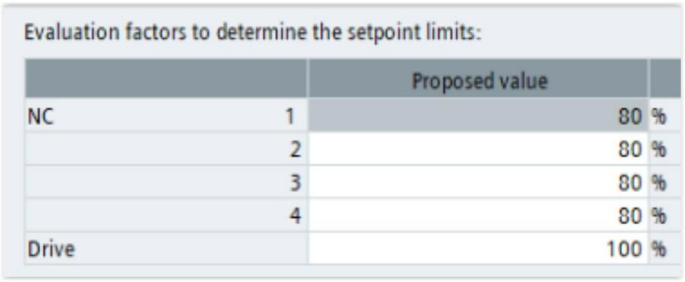




The **Safe actual velocity limit** is as the default SLS setting.

The **Set velocity limitation** (1600.000mm/min) is due to the value that was set using the recommended settings and the default SLS setting (2000.000mm/min)
i.e. **80%** of **Safe actual velocity limit** =
80% of 2000.000 = 1600.000

Below, the previously calculated set-point limitation:



Calculating the required SLS levels:

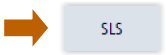
Example: The OEM has stated the following feed requirements:

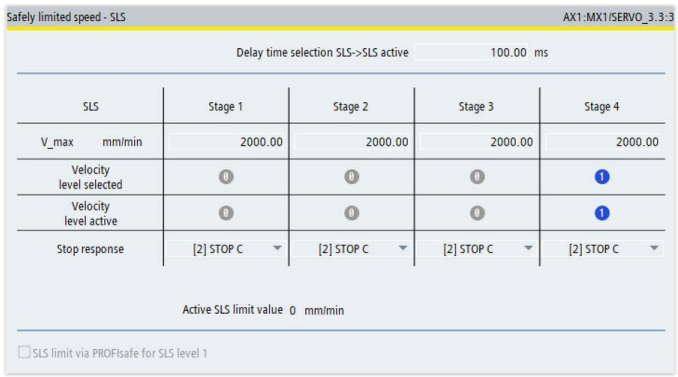
	Set-up/door open/ hold to run		Auto/ door closed	
	Maximum Feed-rate	**SLS1**	Maximum Feed-rate	**SLS4**
X, Y, Z (mm/min)	2000	**2500**	10000	**12500**
A (rpm)	10	**12.5**	100	**125**
Spindle (rpm)			1000	**1250**

The SLS levels have been calculated using the 80% limitation factor (feed-rate / 0.8):

The calculated values need to be entered via the SLS set-up screen:



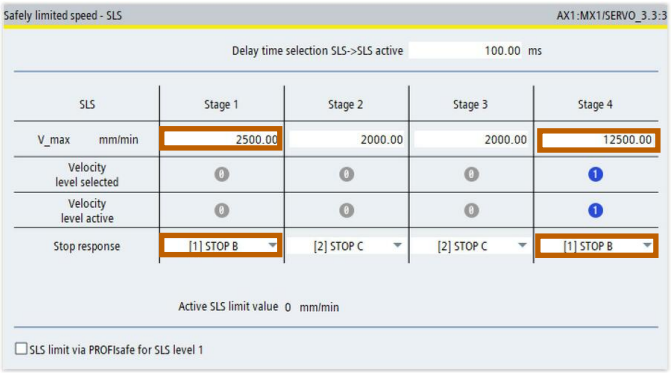






Notes:

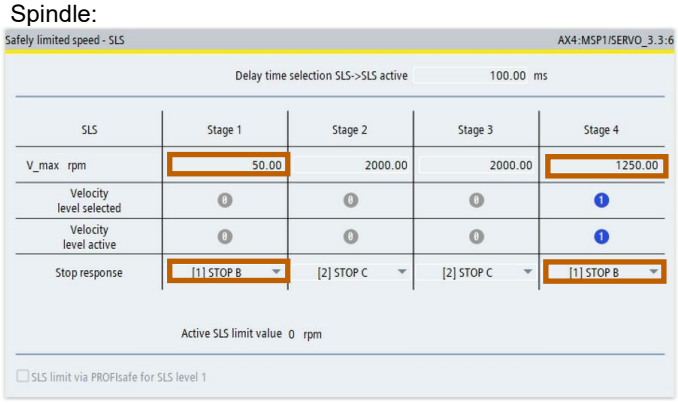
X, Y & Z axes:



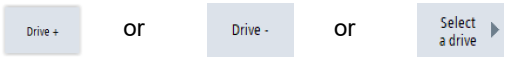
A axis



Spindle:



Now repeat for all of the drives





Start the procedure for copy/confirm



Copy the SI data of all drives



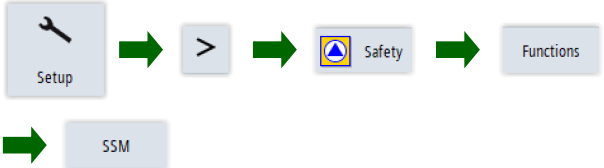
Confirm the SI data of all drives

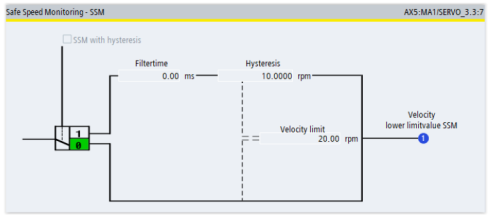
Safe Speed Monitoring (SSM)

Safe Speed Monitoring is used to set a velocity limit, below which the axis is considered at standstill.
This can then be used to set an output, for example, **to release a door lock.**

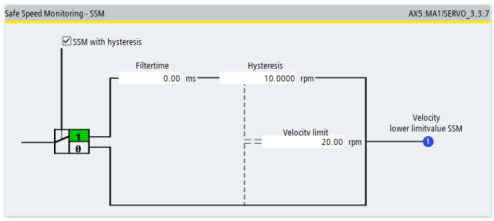
Parameter Description:

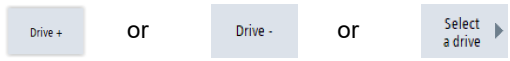
Parameter	Description
SSM with hysteresis (p9501.16)	Enables the SSM function.
Filter time (ms) (p9545)	Sets a filter time for SSM to detect standstill.
Hysteresis (mm/min or rpm) (p9545)	Sets the speed hysteresis SSM to detect standstill.
Speed limit (mm/min or rpm) (p9546)	Sets the speed limit for SSM to detect standstill.













Start the procedure for copy/confirm



Copy the SI data of all drives



Confirm the SI data of all drives

Notes:

SBT - Safe Brake Test (can also be configured on the virtual machine)

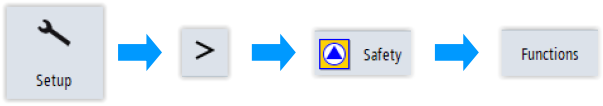
The Safe Brake Test is designed to test the functionality of the brake.

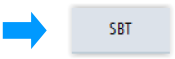
It is possible to test up to two brakes per axis.
There are two brake test parameter sets available for each brake.

During the test the brake is closed and the motor is given a command to move and the axis is monitored for actual movement.
Too much movement indicates a worn/faulty brake and the system will issue an alarm.

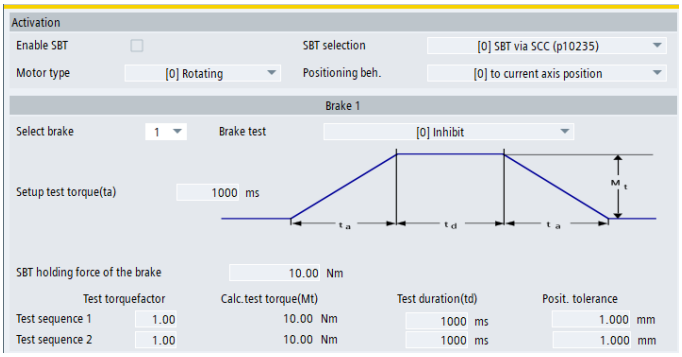
The test is initiated and monitored in the user PLC via the SIC/SCC interfaces.

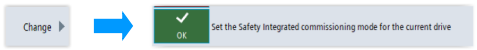
Note: The safe brake test should be initiated immediately before the running of the test stop routine.
This is because the test stop routine removes the drive pulses as part of the checking procedure and could result in the axis falling, were the brake to be faulty.
In the event of a faulty brake being detected, the test stop should be prevented from running until the faulty brake has been dealt with.

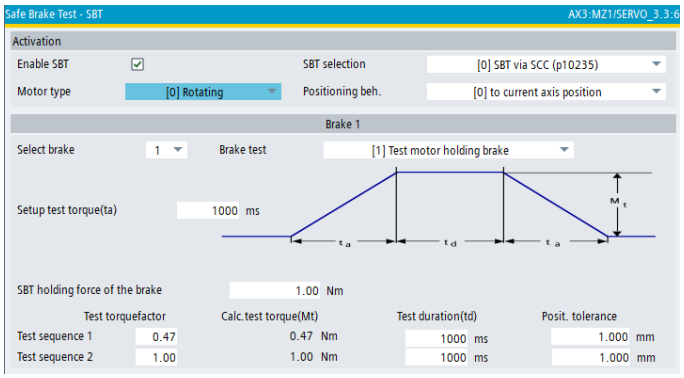




Select the Z axis drive because the servo motor has a brake







Example - basic settings for the training equipment:

- Set ***Enable SBT*** tick-box
- Select the ***Motor type*** to ***[0] Rotating.***
- Select the ***SBT selection*** to ***[0]SBT via SCC (p10235)***
- Select the ***Positioning beh.*** to ***[0] to current axis position***

Description of the Brake Test Parameters:

Parameter	Description
Select brake (p10202)	Selection of the brake to be tested. • 1: Brake 1 (p10202[0]) • 2: Brake 2 (p10202[1])
Brake test (p10202)	Defines whether and which brake should be tested • 0: Inhibit • 1: Test motor holding brake • 2: Test external brake
Setup test torque(ta) (p10208)	Torque ramp up/down time during the brake test
SBT Holding force of the brake (p10209)	A value should be entered based upon the holding capabilities of the brake. This value is used in the calculation of the actual torque used in the brake test.
Test torquefactor (p10210)	The actual torque used in the brake test. is the result of: **Holding force x Test torquefactor.** The **Test torquefactor** is calculated from the holding torque of the axis.

Notes:

Parameter	Description
Calc.test.torque(mt)	Value calculated from “SBT holding torque of the brake” and “Test torque factor”
Test duration(td) (p10211)	The duration of the SBT, excluding torque ramp up/down.
Posit tolerance (p10212)	Maximum distance the axis can move, during the test, without an alarm being triggered.

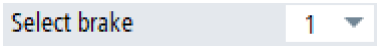
Setting the Brake Test Parameters:

The brake test parameters must be set so that the test can be completed successfully.

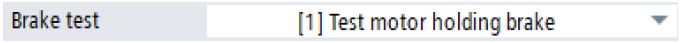
Selecting the brake to be tested:

The training equipment has a motor brake on the Z axis, therefore, only one set of brake parameters needs to be set. The ***Brake 1*** parameter set will therefore be used for this example:

- Select the ***Select brake*** option to ***1***:



- Select the ***Brake test [1] Test motor holding brake***

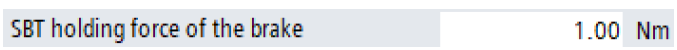


Setting the holding torque:

This should be entered based upon the holding capabilities of the brake.
This value is then used in a calculation to determine the actual torque to be used in the brake test:
The following table shows the motor holding brake capabilities for he 1FK7 motor range.

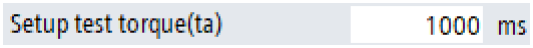
Motor Type	Brake Holding Torque (Nm)
1FK701_	0.4
1FK702_	1
1FK703_	1.9
1FK704_	4
1FK706_	13
1FK708_	22
1FK7100	23
1FK7101	43
1FK7103	
1FK7105	

The training equipment Z axis is fitted with a **1FK702** type motor, therefore set the Holding torque to 1Nm:



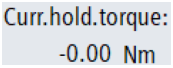
Setting the Setup test torque:

The default ramp up/ramp down time (1000ms), of the brake test torque, can be used for the training:



Test torque factor.

The ***Test torque factor*** is calculated using the ***current holding torque (p1509)*** of the axis.
This can be seen on the Safe Brake Test screen:



Note: on an actual machine there would be value present.
The direction of the ***current holding torque*** is dependant on the machine configuration.

Notes:

Note: the training equipment has no vertical axis, therefore, there the ***Curr. Holding torque*** will be **0.**
Therefore, for the purpose of the training, an assumed value will be used to calculate a realistic ***Test torque factor***

Test torque factor calculation:

The following calculations can then be used to determine the ***Test torquefactor and SBT testtorque***

1. ***Test torque factor*** **=**
   (Curr. Holding torque * 1.3/Brake holding torque)
   = (r1509 * 1.3 / p10209)

2. ***SBT testtorque*** **=**
   Brake holding torque * Test torque factor
   = p10209 * p10210

Note: the calculation results an **additional 30% torque** being applied to the brake during the brake test.
This is designed to test the brake beyond the normal holding torque of the brake.

Training equipment example:

Note: an assumed value of ***0.36Nm*** is being used as the ***Curr. Holding torque (r1509).***

The ***Test torquefactor*** can then be calculated as:

= (0.36 * 1.3 / 1) = 0.47

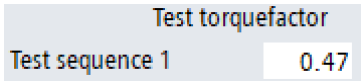
Calculation of the ***SBT testtorque*****:**

The torque to be used during the SBT (0,47Nm) is set using:

Brake holding torque * Test torque factor
1 * 0.47 = 0.47 Nm

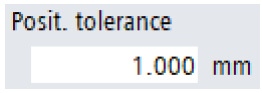
The ***SBT testtorque*** can be seen in the display

Therefore, the ***Test torquefactor*** can be set to 0.47



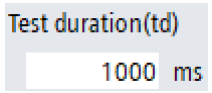
Position tolerance:

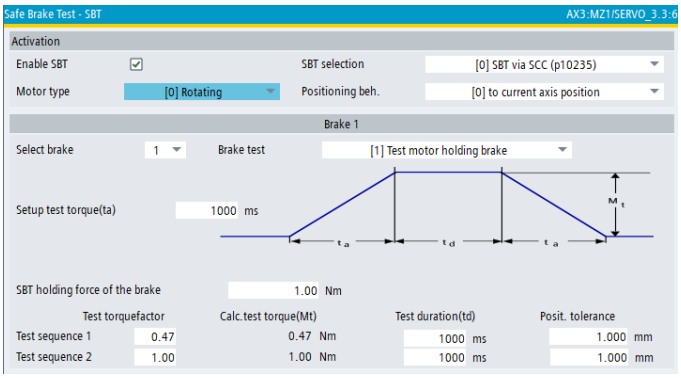
The default position tolerance(1 mm) can be used for the training:



Test duration:

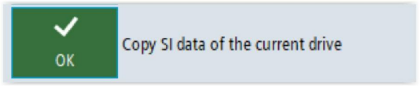
The default test duration(1000 ms) can be used for the training:

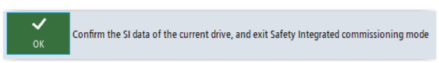






Carry out the copy and confirm procedure for the relevant drive





Notes:

Initiating the Brake Test:

The user PLC interface is used to initiate and start the Safe Brake Test.
Each axis has bytes in the axis interface data block (DB31...DB61) reserved for safety integrated signals to/from the PLC/Drive.

Signals from the PLC to the drive (Safety Command Channel):

DB31 ...DB61	Signals from PLC to Drive (Safety Command Channel—SCC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB143			Brake Closed	Torque 1 or 2	Direction of Rotation	Brake 1 or 2	BT Start	Selection of SBT

Signals from the drive to the PLC (Safety Information Channel):

DB31 ...DB61	Signals from Drive to PLC (Safety Information Channel—SIC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB173	Sign Load Torque	Brake Req.	B test finished	B test OK	B test Active	Brake 1 or 2	B test Start	Selection Safe BT

Signals from the PLC to the drive (Safety Command Channel):

Bit 0 - Selection of Safe Brake Test: Activate the drive in readiness for the Safe Brake Test.

Bit 1 - Safe Brake Test start - Start the Safe Brake Test

Bit 2 - Selection of brake 1 or 2 - set to 0 for brake 1, set to 1 for brake 2.

Bit 3 - Direction of rotation - set to 0 for positive, set to 1 for negative.

Bit 4 - Torque 1 or 2 - set to 0 for Test torquefactor 1, set to 1 for Test torquefactor 2.

Bit 5 - Brake closed - Set to 1 to confirm that the brake is closed (External brakes only)

Signals from the drive to the PLC (Safety Integrated Channel):

Bit 0 - Selection of Safe Brake Test: Safe Brake Test is active.

Bit 1 - Safe Brake Test start - Safe Brake Test has started.

Bit 2 - Brake 1 or 2 active - 0 = brake 1, 1 = brake 2.

Bit 3 - Brake test is active - 1 = brake test active.

Bit 4 - Brake test ok - 1 = brake test was successful.

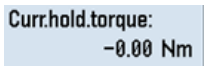
Bit 5 - Brake test finished - brake test has finished.

Bit 6 - Brake test required - brake test is required.

Bit 7 - Load torque sign - direction of the load torque, 0 = positive, 1 = negative. Can be used to determine the SBT direction.

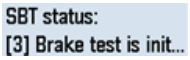
Notes:

Due to the training equipment not having a holding torque, the ***Curr.hold torque*** display is 0.00Nm.



Note: on an actual machine there would be value present. The direction of the ***current holding torque*** is important because the brake test direction should be in the same direction as the force due to the weight of the vertical axis i.e. opposite to the ***current holding torque.***
The direction of the SBT on the training equipment is not important.

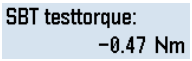
- Enable the drives and set the feedrate to 100%
- Set the ***Select SBT*** interface bit for Z axis (DB33.DBX143.0). The ***SBT status*** will change to 3.



Interface signals, DB33.DBX173.0 & DB33.DBX173.1, will change to 1.

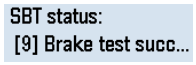
- Set the ***Start SBT*** interface bit for Z axis (DB33.DBX143.1)
The test will start; the axis will be driven against the closed brake using the specified SBT testtorque.

The ***SBT testtorque*** will be displayed on the screen:



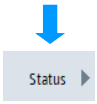
The During the test, interface signal, DB33.DBX173.3 will be set to 1.
On successful completion of the test, DB33.DBX173.4 & DB33.DBX173.5, will be set to 1.

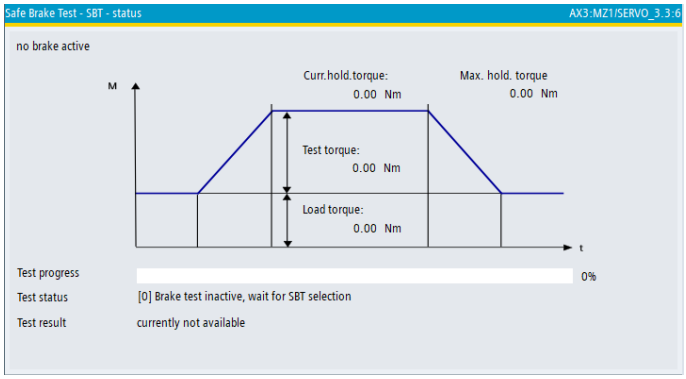
A successful test is indicated in the ***SBT status*** area:



Trace Feature:

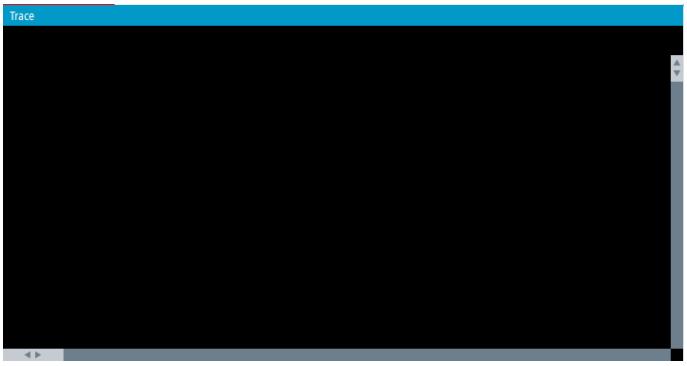
The trace feature can be used to view detailed information.



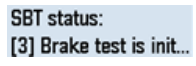




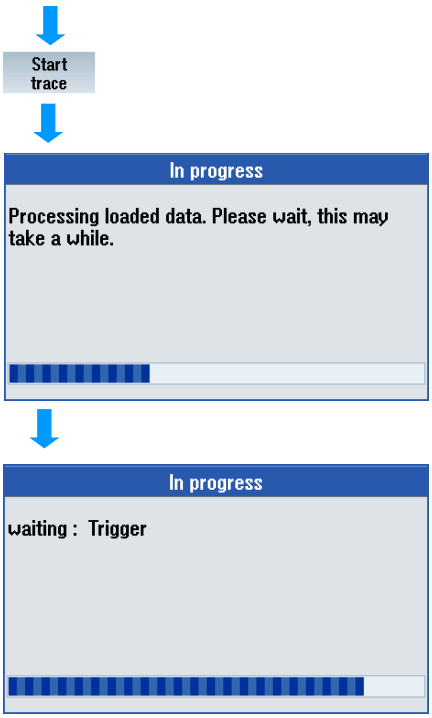
Wait until following screen is displayed:



- Initialise the Safe Brake Test:

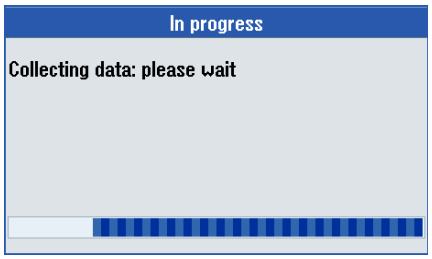


Notes:

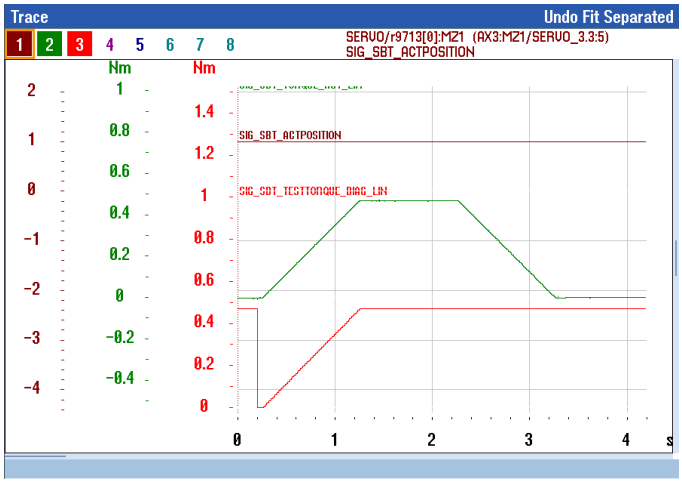


- Start the Safe Brake Test

The trace will be started:



The trace will be displayed automatically:



Green Trace : SBT actual torque.
Red Trace: SBT test torque.
Brown Trace: Actual position.

The traces show that the SBT was in the positive direction and that torque values are the same as was calculated (0.47Nm).

Notes:

1: The green trace had an initial value of **0**.
On an actual machine this would not be the case due to the weight of the vertical axis.

2: The red trace shows that **SBT test torque** had a value of **–0.47 Nm**, prior to the test beginning.
This is a residual value due to the brake having already been tested.
The **SBT test torque** is set to 0 at the beginning of each test.
On power up and before a brake test has been carried out, the **SBT test torque** will have a value of 0.

Note: The safe brake test should be initiated immediately before the running of the test stop routine.
This is because the test stop routine removes the drive pulses as part of the checking procedure and could result in the axis falling, were the brake to be faulty.
In the event of a faulty brake being detected, the test stop routine should be prevented from running until the faulty brake has been dealt with.

Notes:

Test Stop Routine:

NOTE: Any PLC code shown in this document is to be treated as an example and for guidance purposes only!

Test Stop (forced checking procedure) Description:

Functions and switch-off signal paths must be regularly tested using the forced checking procedure (test stop). The forced checking procedure should be performed each time that the control system runs up and must be performed at least once a year. This involves components from the SINUMERIK ONE / SINAMICS S120 system.

The forced checking procedure is not performed in the safety program but in the standard PLC program. This means that you insert the affected program code in your PLC user program in order to trigger the forced check procedure during run-up of the controller.

Safe Brake Test Priority:

Because the Test Stop routine results in the drive pulses being disabled, it is necessary to carry out the Safe Brake Test immediately before the TEST Stop is started.

PLC interface signals are available to report the result of the Safe Brake Test. These can be used to initiate/inhibit the Test Stop, depending on the result of the Safe Brake Test.

The Test Stop routine is started using the Test Stop user Key on the machine control panel.

The forced checking procedure must be initiated by the user or integrated as an automatic procedure.
For example:

- When the axes are stationary after the system has been powered--up
- When the protective door is opened
- In defined cycles (e.g. every 8 hours) The maximum permissible is 1 year).
- In the automatic mode, dependent on the time and event.

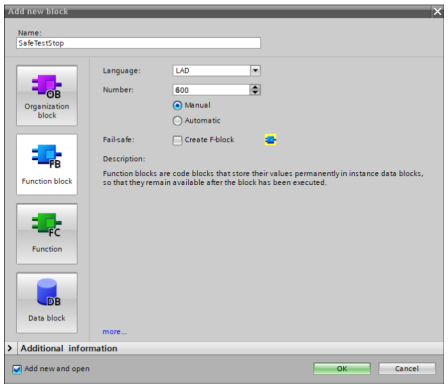
DB31 ...DB61	Signals from PLC to Drive (Safety Command Channel—SCC)
Byte	Bit 0
DBB140	Start Test Stop

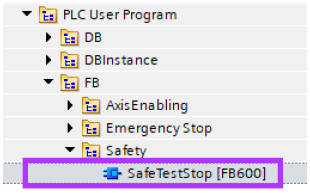
DB31 ...DB61	Signals from Drive to PLC (Safety Information Channel—SIC)	
Byte	Bit 5	Bit 4
DBB166	Test Stop Required	Test Stop Active

Additional Interface signals required for Test Stop

Automating the Safe Brake Test and the Test Stop.

The Safe Brake Test and Test Stop logic is added to an FB in the non safety user PLC. In this example a new FB called SafeTestStop and numbered FB600 is added.





The following Input variable is required

SafeTestStop

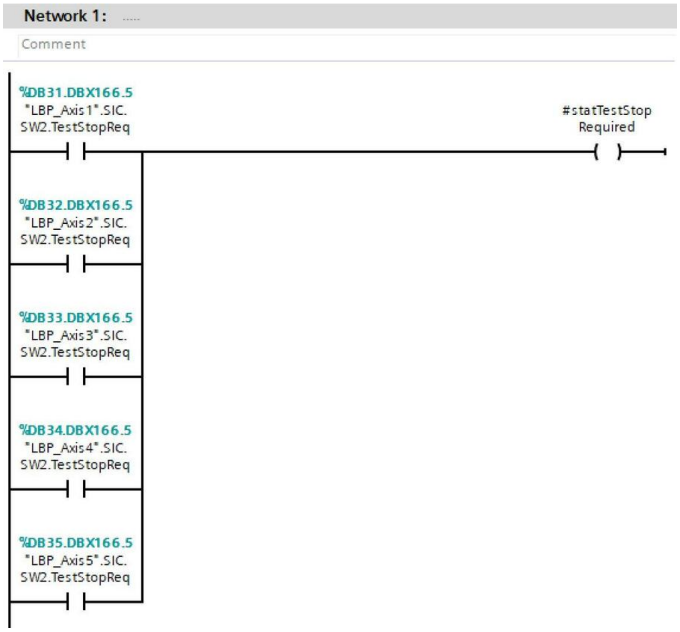
	Name	Data type
1	Input	
2	K1/K2Feedback	Bool

The following static variables are required

8	Static	
9	statTestStopRequired	Bool
10	statAllAxesandSpindleSOS&DoorClosed	Bool
11	statReadyForTestStop	Bool
12	statSafeBrakeTestPrepare	Bool
13	statSafeBrakeTestStart	Bool
14	statSafeBrakeTestOK	Bool
15	statStartTestStop	Bool
16	statStartTestStopPulse	Bool
17	statK1RelayTest	Bool
18	statK1FeedbackFault	Bool
19	statK2RelayTest	Bool
20	statK2FeedbackFault	Bool
21	statTestStopBusy	Bool

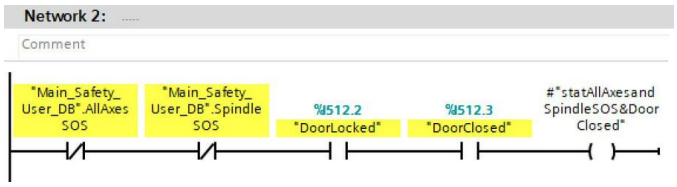
Notes:

When a Test Stop is required, by a drive, an interface signal from the axis interface is set to 1.

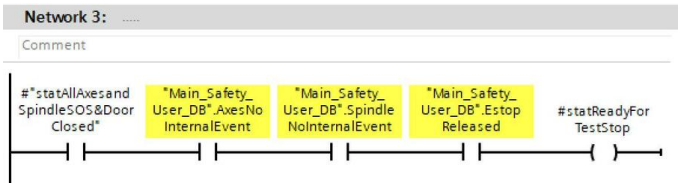


The Test Stop Request signal can be used to first start the brake test, and then, assuming that the brake test is successful, initiate the Test Stop.

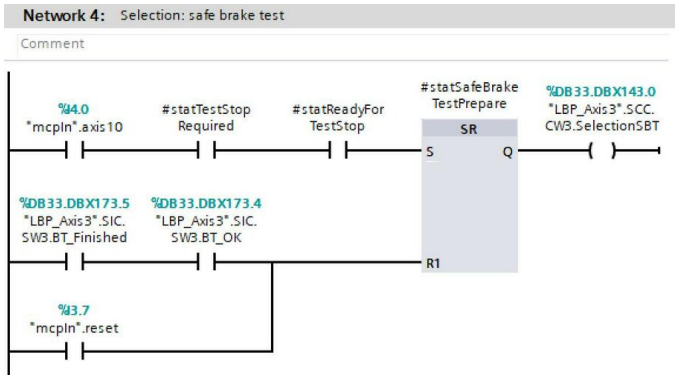
Example: monitoring for no SOS active and that the door is closed and locked.



Example: monitoring for no SOS active and that the door is closed and locked., no internal event and no emergency stop.

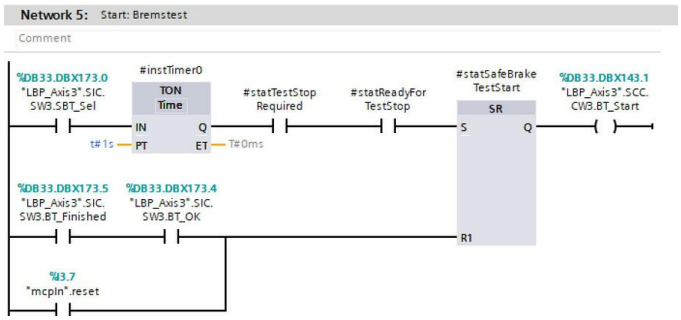


For the purpose of the training the Test stop User key is used to start the Brake Test providing that the conditions are met.



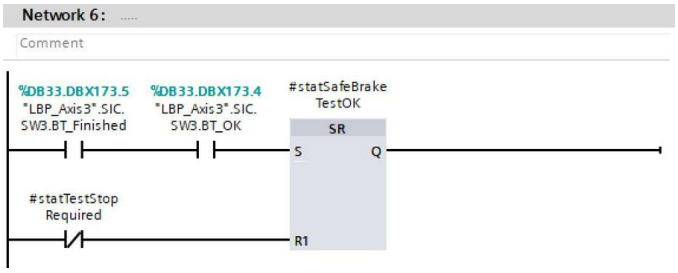
The Brake Test selected signal is used to start the Brake Test sequence.

Note: when selecting IEC timers an Instance Data Block is required. If several IEC timers are to be used it is best to use a Multi Instance DB so that just one DB is used rather than an instance DB for each timer.

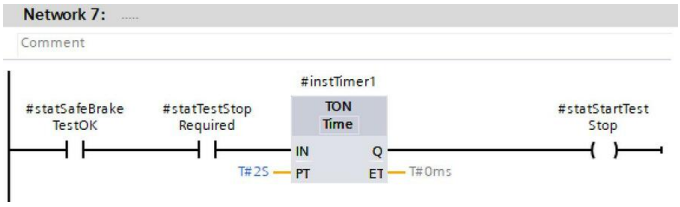


Notes:

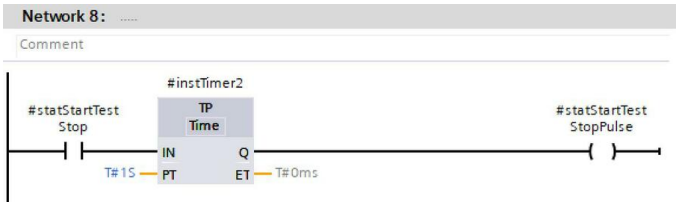
If the Brake Test is successful the start Test Stop sequence is initiated.



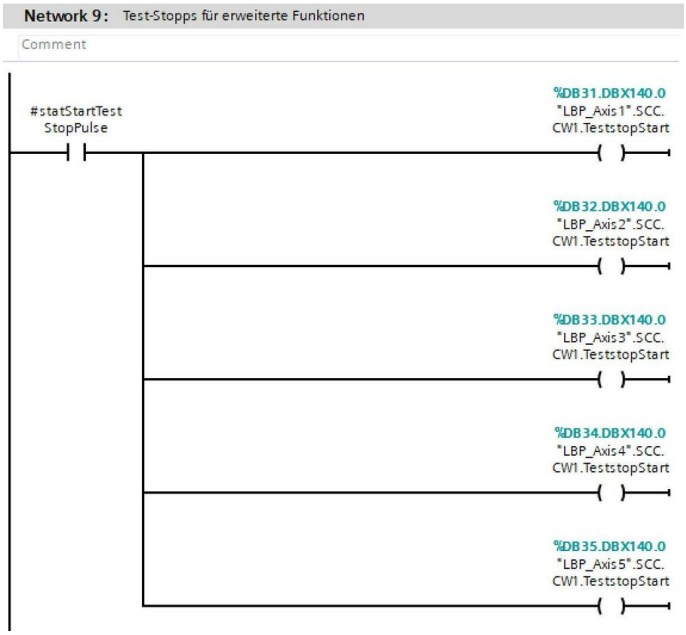
A 2 second delay allows the Safe Brake Test signals to normalise before starting the Test Stop.



The Test Stop signal will remain high for 1 second to allow the pulses to be disabled/enabled.



The Test Stop start signal is set for each axis via the axis DBs.



The disabling/enabling of the pulses can be seen in the axes diagnostics area:

Service overview

Select: All NC axes and drives

	AX1	AX2	AX3	AX4	AX5
On/Off1	●	●	●	●	●
Off2	✓	✓	✓	✓	✓
Off3	✓	✓	✓	✓	✓
From drive: Operation enabled	●	●	●	●	●
From power supply:Operation enabled	✓	✓	✓	✓	✓
Pulse enable	●	●	●	●	●
Speed controller enable NC	●	●	●	●	●
Pulses enabled	●	●	●	●	●
Drive ready	●	●	●	●	●
Heatsink temperature	✓	✓	✓	✓	✓
Power section in i^2t limitation	✓	✓	✓	✓	✓
Motor temperature	✓	✓	✓	✓	✓
Measuring system 1 active	●	●	●	●	●
Measuring system 2 active	●	●	●	●	●

- The Test Stop interval time will now be reset.

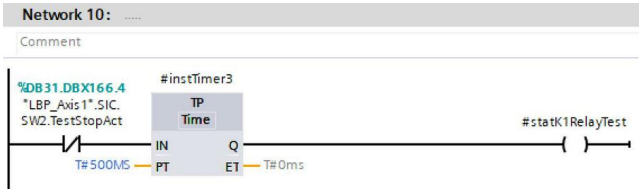
Notes:

Testing the K1 & K2 relays

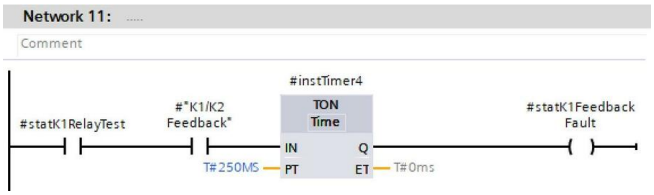
The correct operation of the K1 and K2 relays can also be tested during the Test Stop procedure.
The relays each have a normally closed contact connected in series to input I100.6
By momentarily switching the relays off/on the input can be monitored for the correct operation of the K1 and K2 relays.

Enter the example logic for testing the relays.
Start the K1 Relay Test after the Test Stop has finished:

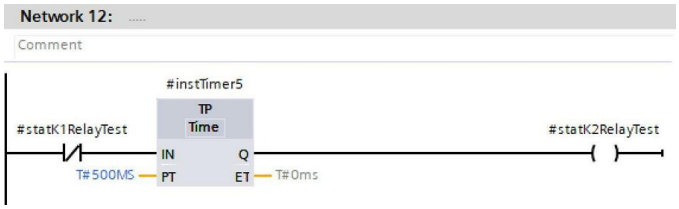
The “K1/K2RelayTest” signals are used in the Safety PLC.



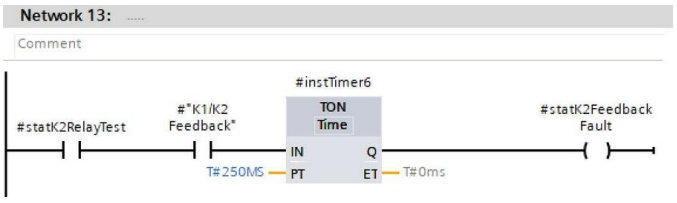
Monitor the K1 Relay feedback input switches within 250ms. Trigger a message if it fails.



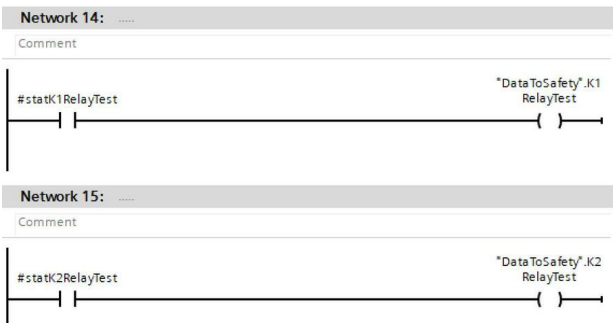
Start the K2 Relay Test once the K1 Relay Test has completed has finished:



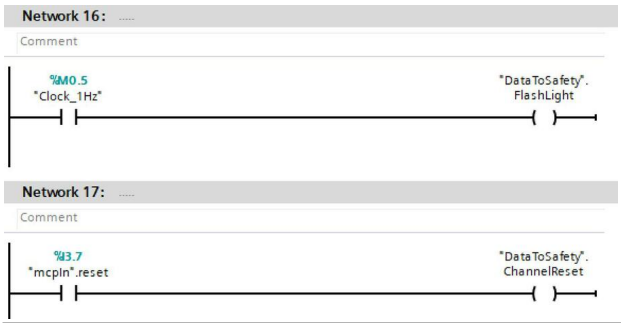
Monitor the K2 Relay feedback input switches within 250ms. Trigger a message if it fails:



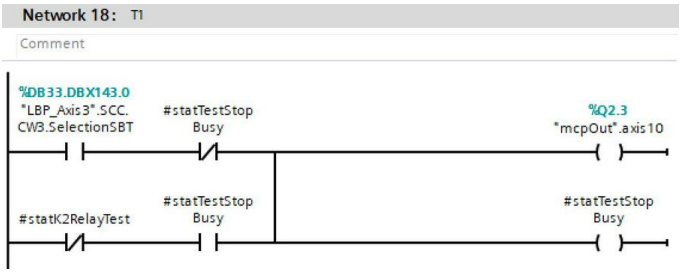
The K1/K2 Relay test signals are assigned to variables in the Global DB601 for use in the main safety FB1010. By momentarily switching the relays off/on the input can be monitored for the correct operation of the K1 and K2 relays.



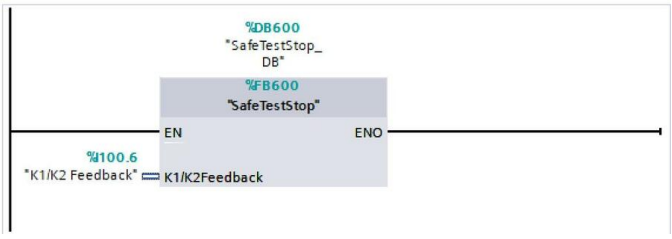
The 2Hz clock signal and MCP reset signal are also transferred to the Global DB601 for use in the main safety FB1010.



The “Test Stop” lamp on the MCP is on while the Brake Test and Test Stop routines are running.



FB600 should be called in OB1, in this example DB600 is used as the multi instance DB



Notes:

Notes:

Appendix 1 Safety Simulator Drawings

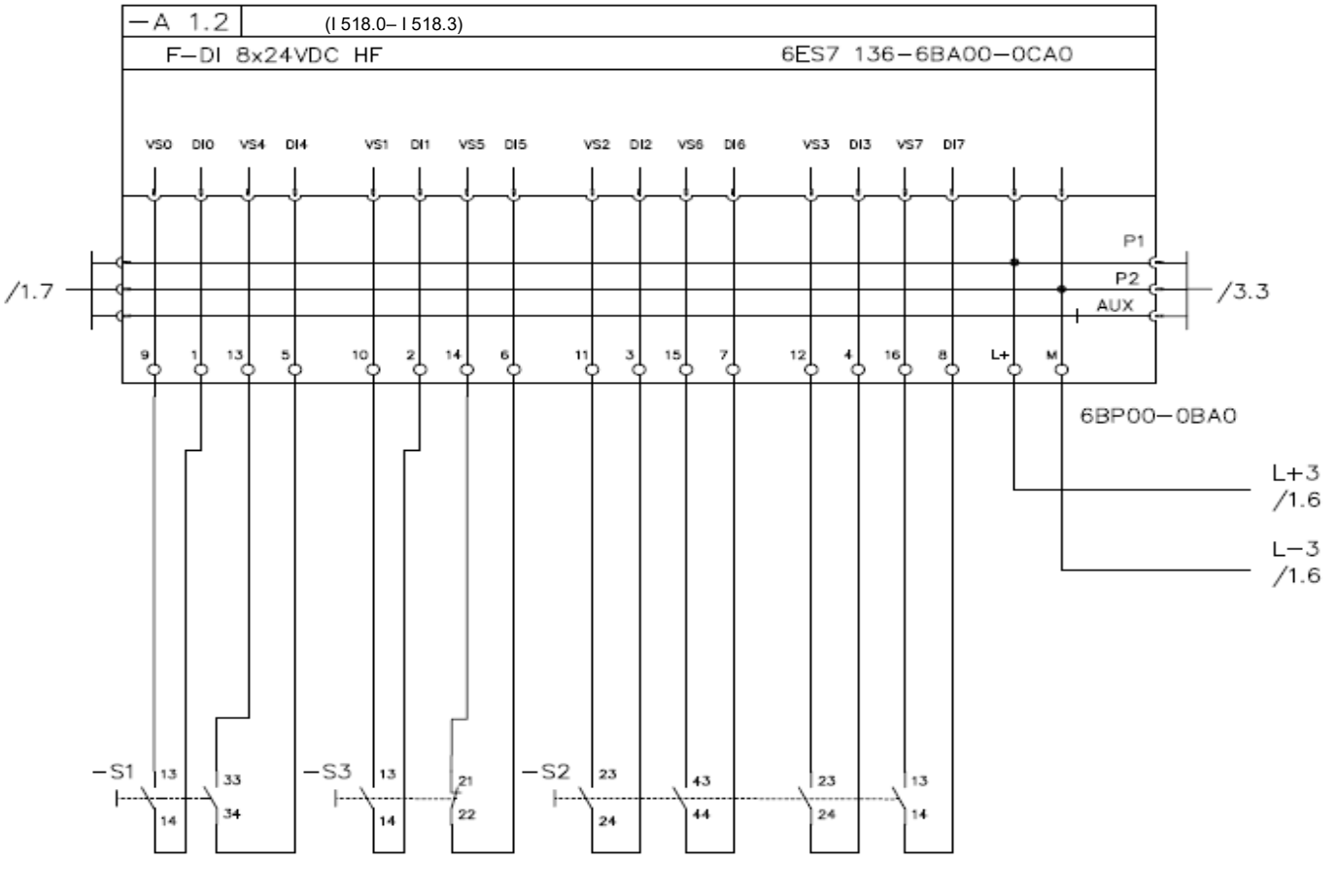
(I 512.0– I 512.3)

Locked Closed

Emergency Stop

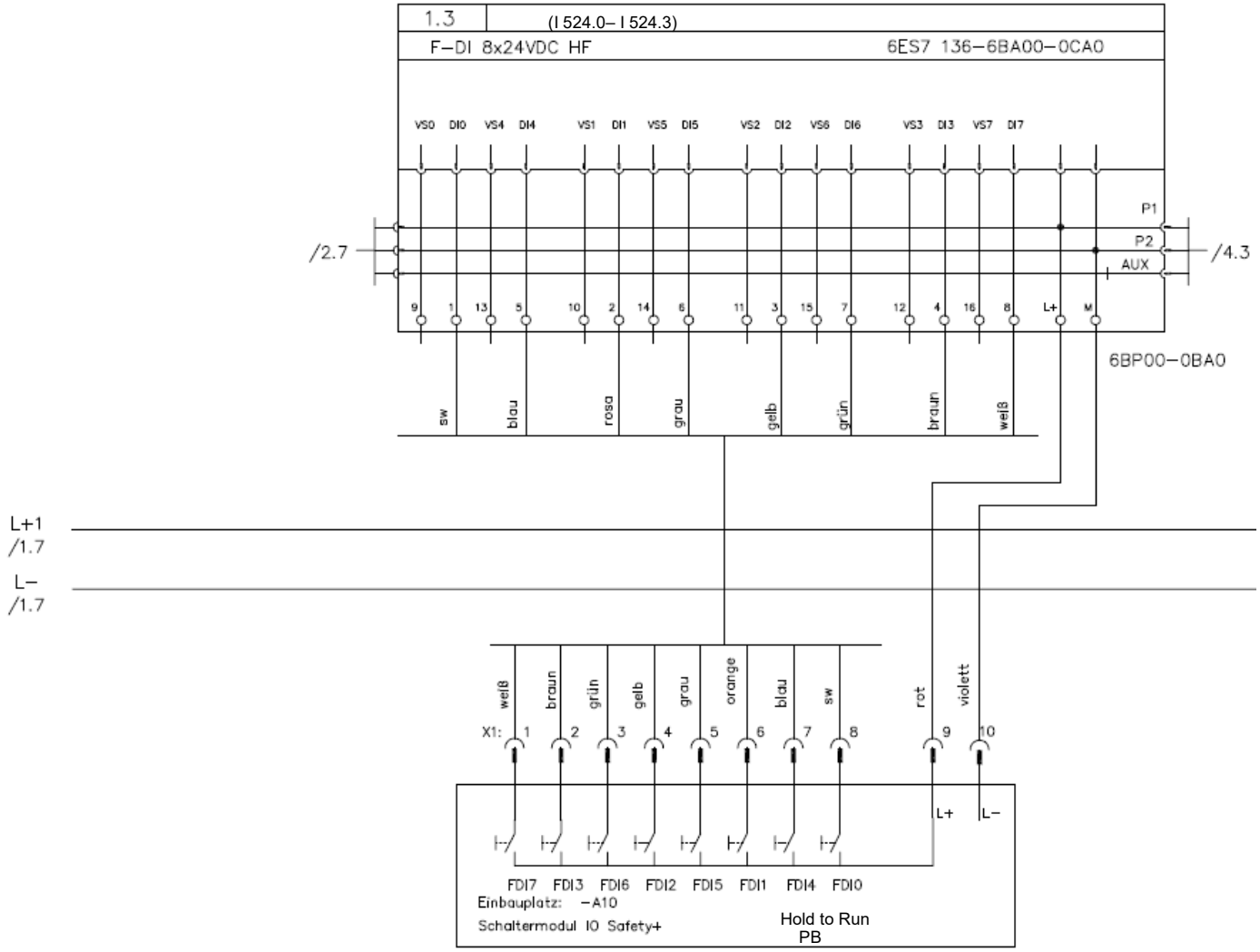
Safety Door

Notes:



Emergency Stop reset Set-up Mode

Notes:


1.3
(I 524.0– I 524.3)
F–DI 8x24VDC HF
6ES7 136–6BA00–0CA0
VS0 DI0 VS4 DI4 VS1 DI1 VS5 DI5 VS2 DI2 VS6 DI6 VS3 DI3 VS7 DI7
P1
P2
AUX
/2.7
/4.3
9 1 13 5 10 2 14 6 11 3 15 7 12 4 16 8 L+ M
6BP00–0BA0
sw blau rosa grau gelb grün braun weiß
L+1
/1.7
L–
/1.7
weiß braun grün gelb grau orange blau sw rot violett
X1: 1 2 3 4 5 6 7 8 9 10
L+ L–
FDI7 FDI3 FDI6 FDI2 FDI5 FDI1 FDI4 FDI0
Einbauplatz: –A10
Schaltermodul IO Safety+
Hold to Run
PB

(Q 530.0– Q 530.3)



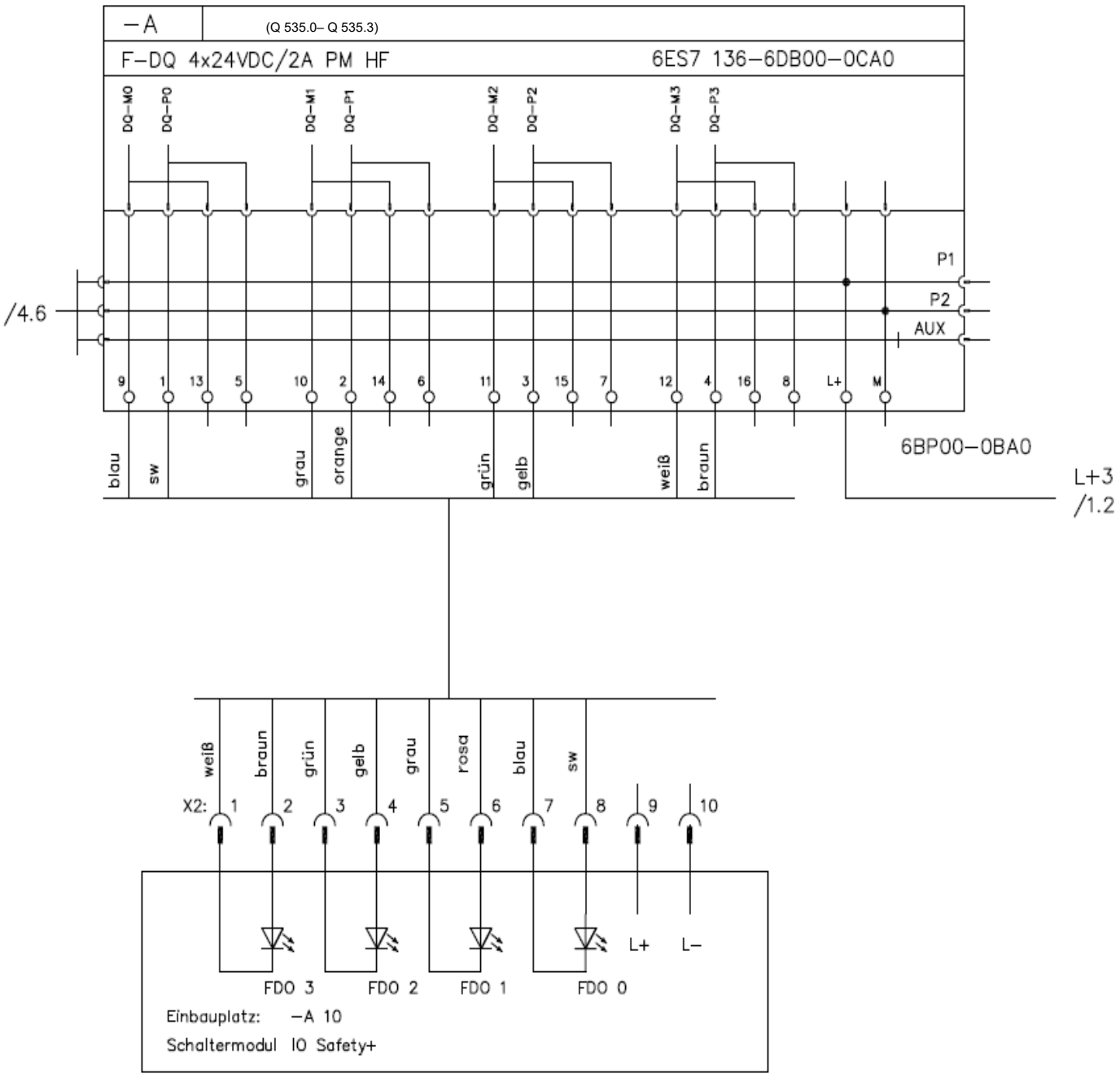
ESTOP ACK LED

INT EVENT LED

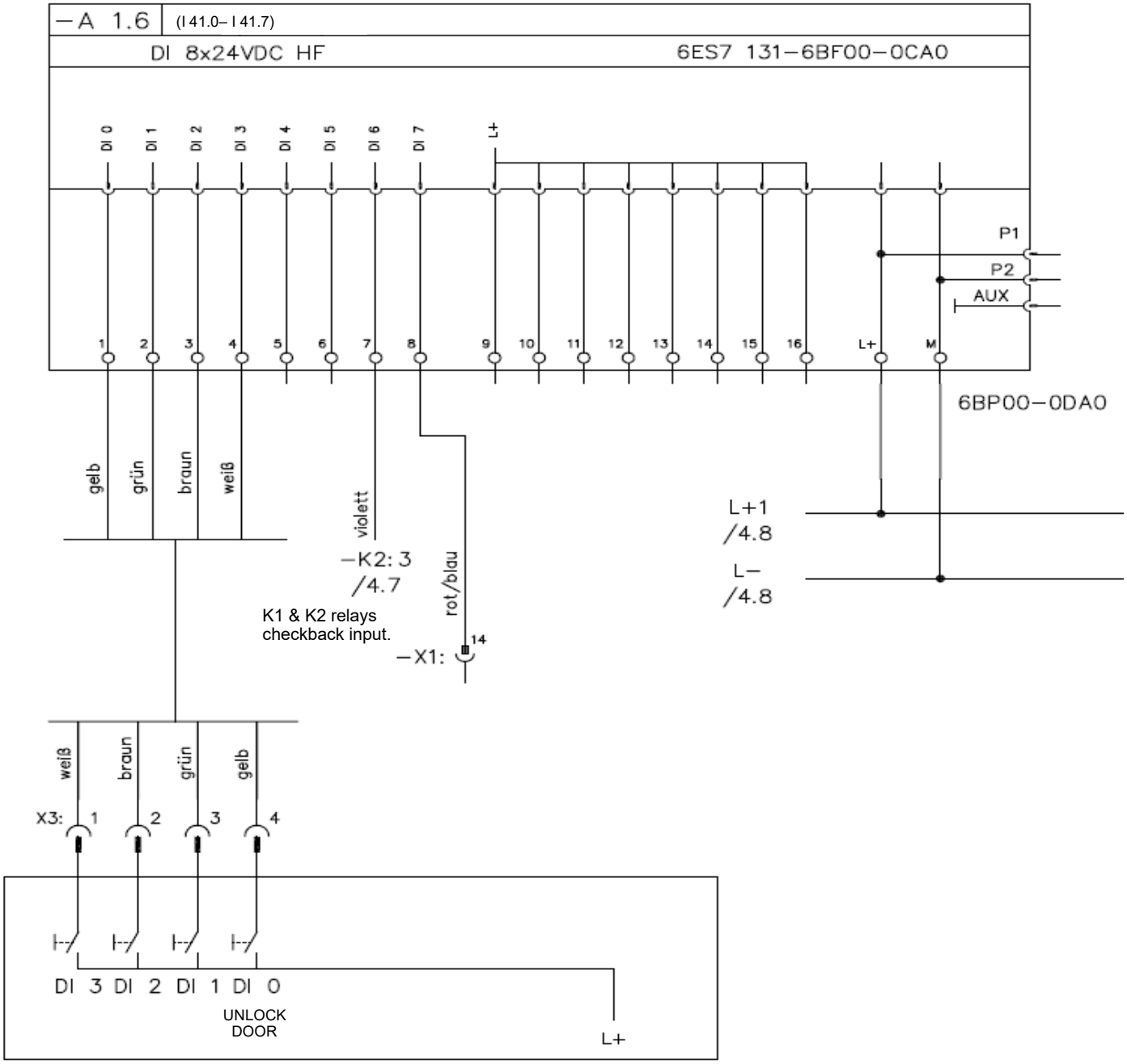
K2 LED

K1 LED

Notes:


−A
(Q 535.0– Q 535.3)
F−DQ 4x24VDC/2A PM HF
6ES7 136−6DB00−0CA0
DQ−M0
DQ−P0
DQ−M1
DQ−P1
DQ−M2
DQ−P2
DQ−M3
DQ−P3
P1
P2
AUX
/4.6
9 1 13 5 10 2 14 6 11 3 15 7 12 4 16 8 L+ M
blau
sw
grau
orange
grün
gelb
weiß
braun
6BP00−0BA0
L+3
/1.2
weiß
braun
grün
gelb
grau
rosa
blau
sw
X2: 1 2 3 4 5 6 7 8 9 10
L+
L−
FDO 3
FDO 2
FDO 1
FDO 0
Einbauplatz: −A 10
Schaltermodul IO Safety+

Notes:


–A 1.6
(I 41.0– I 41.7)
DI 8x24VDC HF
6ES7 131–6BF00–0CA0
DI 0
DI 1
DI 2
DI 3
DI 4
DI 5
DI 6
DI 7
L+
P1
P2
AUX
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
L+
M
6BP00–0DA0
gelb
grün
braun
weiß
violett
rot/blau
–K2: 3
/4.7
K1 & K2 relays
checkback input.
–X1:
14
L+1
/4.8
L–
/4.8
X3:
1
2
3
4
DI 3
DI 2
DI 1
DI 0
UNLOCK
DOOR
L+

Notes:

Q540.0

QB41

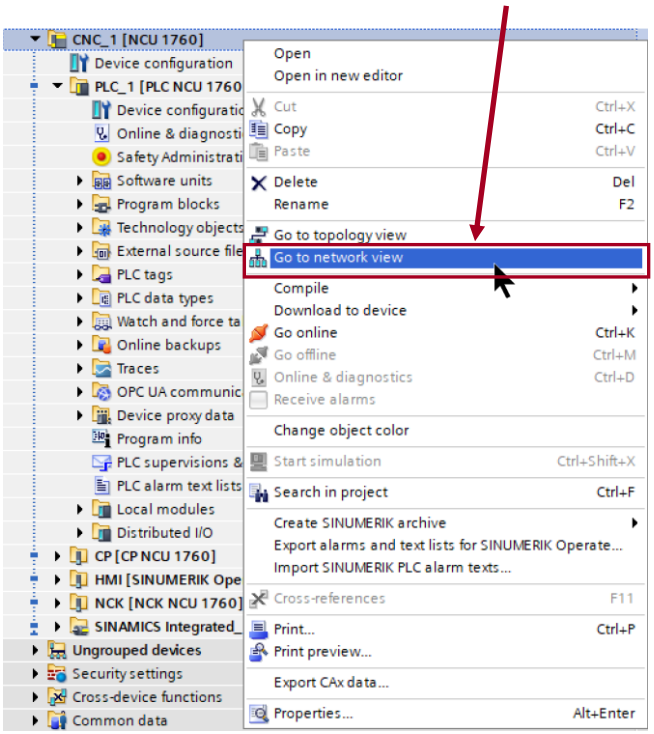
Safety door lock/unlock

Q100.7 Door Unlock

Notes:

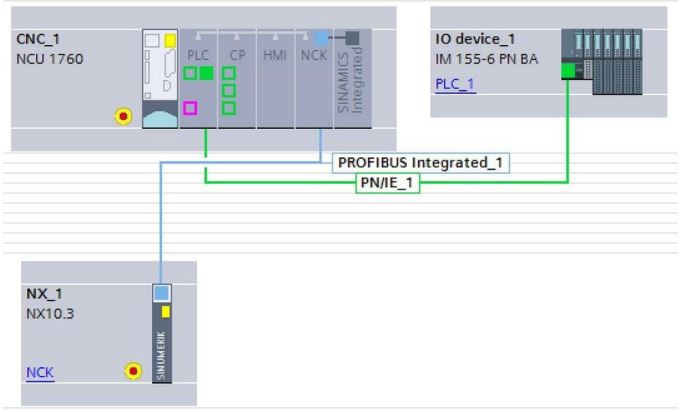
Adding the Safety Integrated Hardware

Open the project and select ***Project view.***
Right click on the NCU 1760 and select ***Go to network view***.

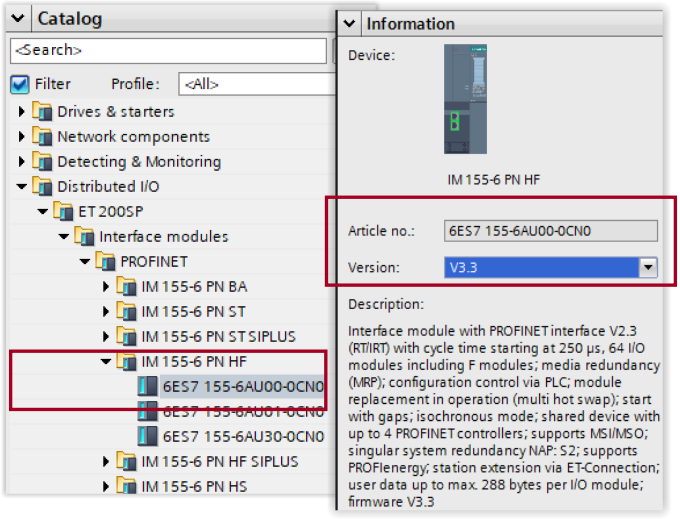


Network view

The current hardware configuration contains the NCU, an IO device_1 and an NX module.

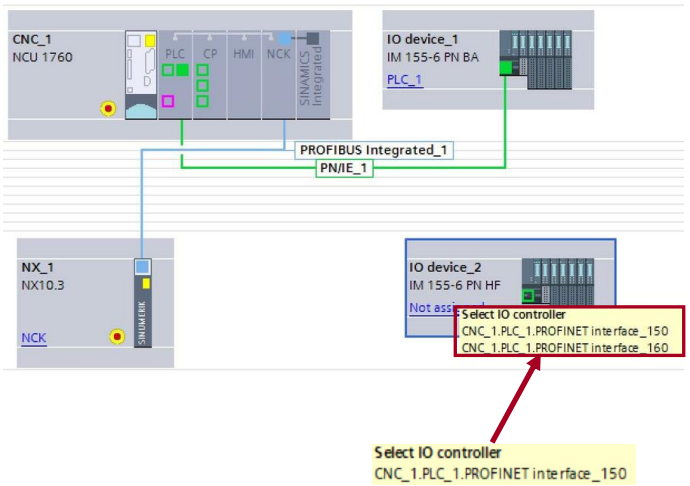


The module **IM 155-6 PN HF 6ES7155-6AU00-0CN0** from the **hardware catalog** is inserted here into the **network view** using the **drag and drop** function.

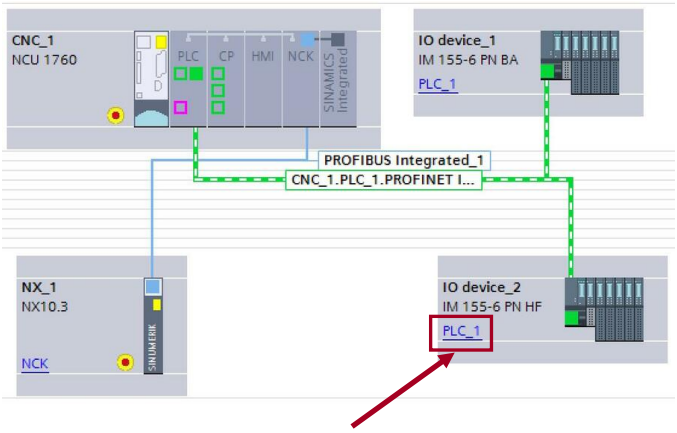


Select the corresponding **version** of the module being used (in this case V3.3) under **Information**.

The device needs to be assigned to the CNC Profinet interface:

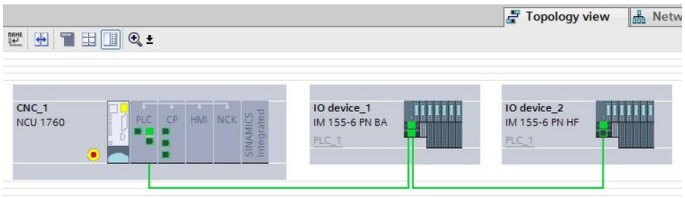


Notes:



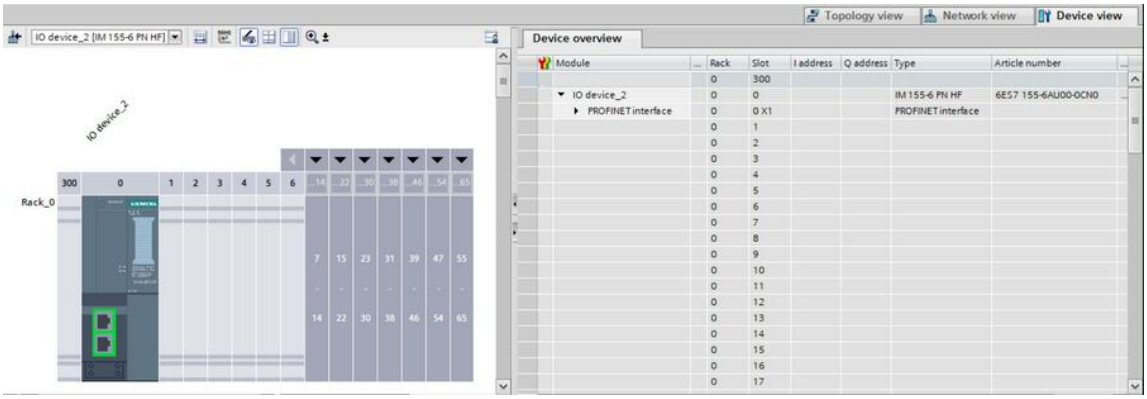
The IM155 is now assigned to PLC

The modules can also be connected in the Topology view taking into account the correct connections e.g Port_1[X1.P1] of IO device_2 to Port_2[X1.P2] of IO device_1.



Adding the Input/Output Devices:

Device view



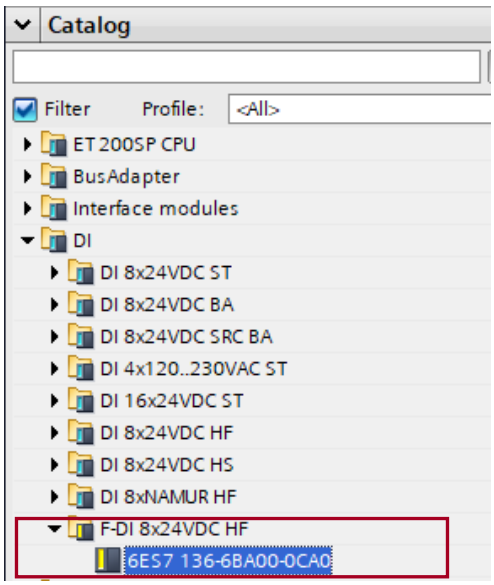
Right: the PLC hardware equipment and slot locations.

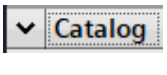
Drag and drop the components into the correct slots of the IM155 rack.

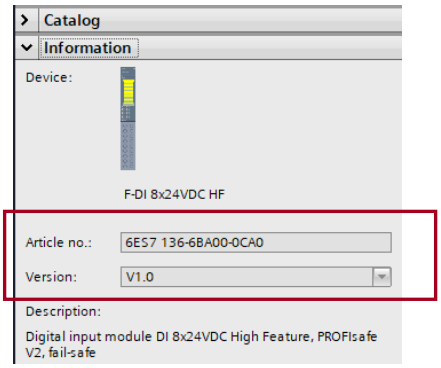
Slot:	Item:
1	6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC)
2	6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC)
3	6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC)
4	6ES7136-6DB00-0CA0 (F-DQ 4 x 24V DC)
5	6ES7136-6DB00-0CA0 (F-DQ 4 x 24V DC)
6	6ES7131-6BF00-0CA0 (DI 8 x 24V DC)
7	6ES7136-6PA00-0BC0 (F-PM-E 24V DC)
8	6ES7132-6BF00-0CA0 (DQ 8X24V DC 24V DC)
9	6ES7193-6PA00-0AA0 (Server module)

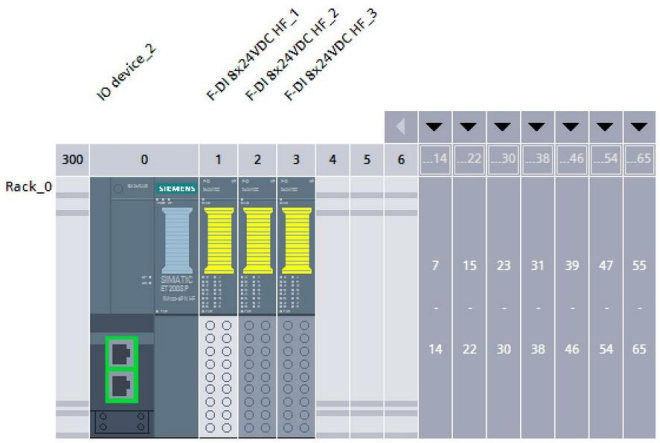
Notes:

Slots 1 to 3: 6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC)

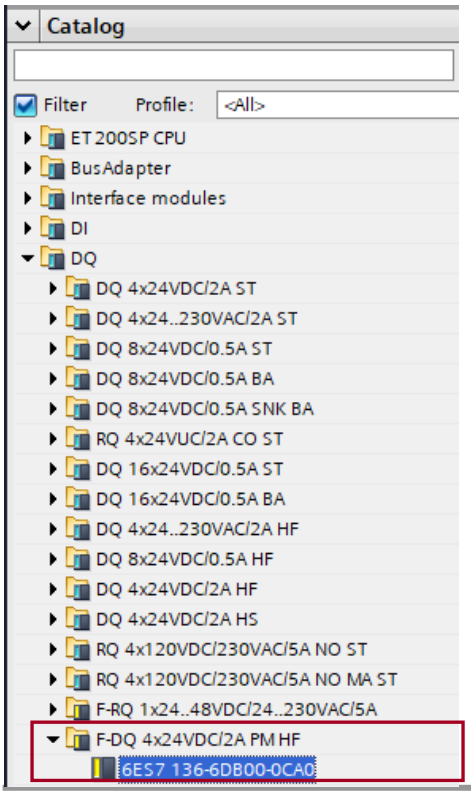


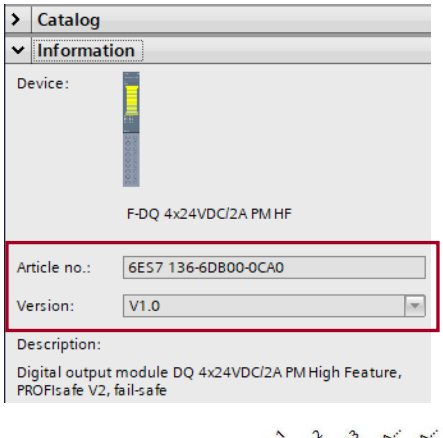


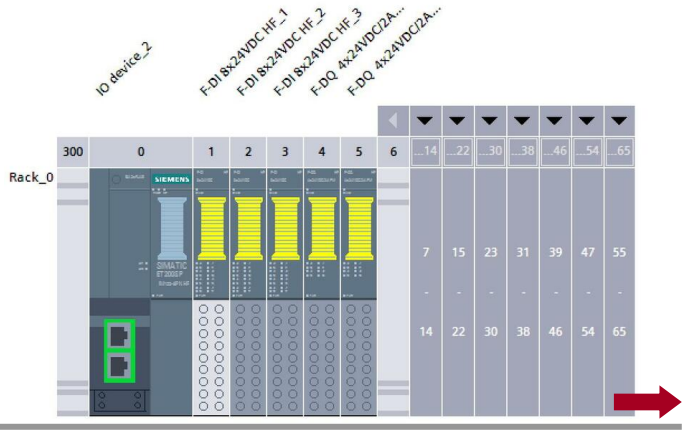




Slots 4 & 5: 6ES7136-6DB00-0CA0 (F-DQ 4 x 24V DC)

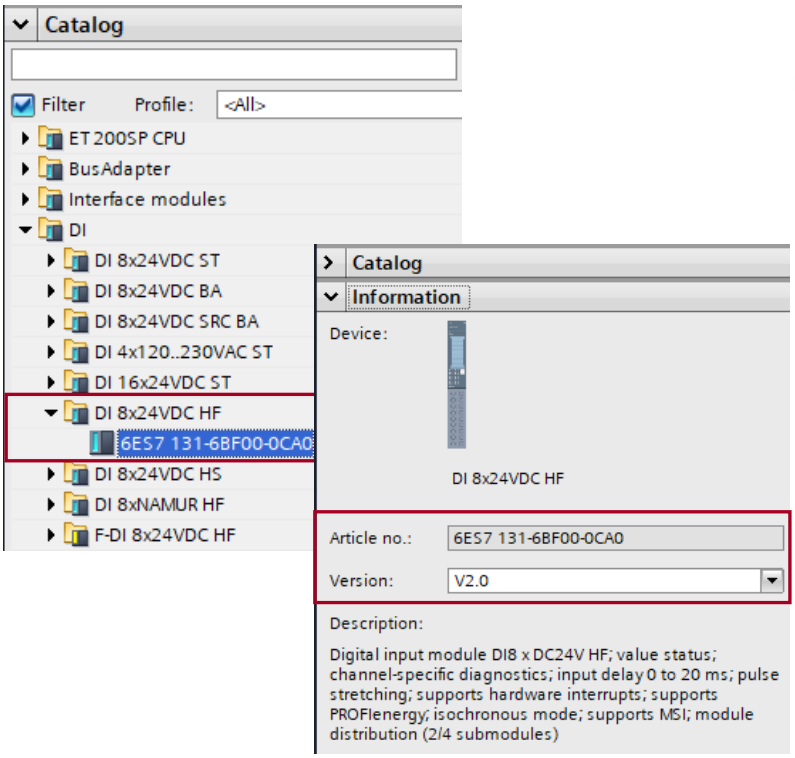


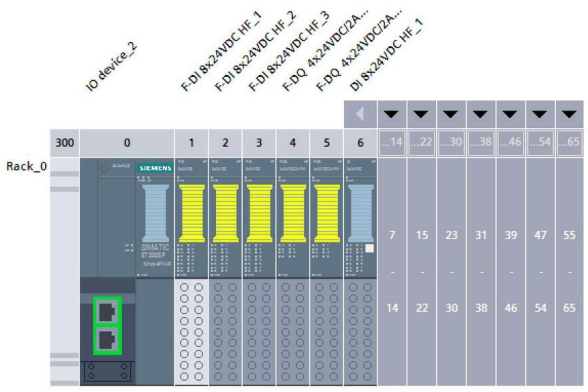




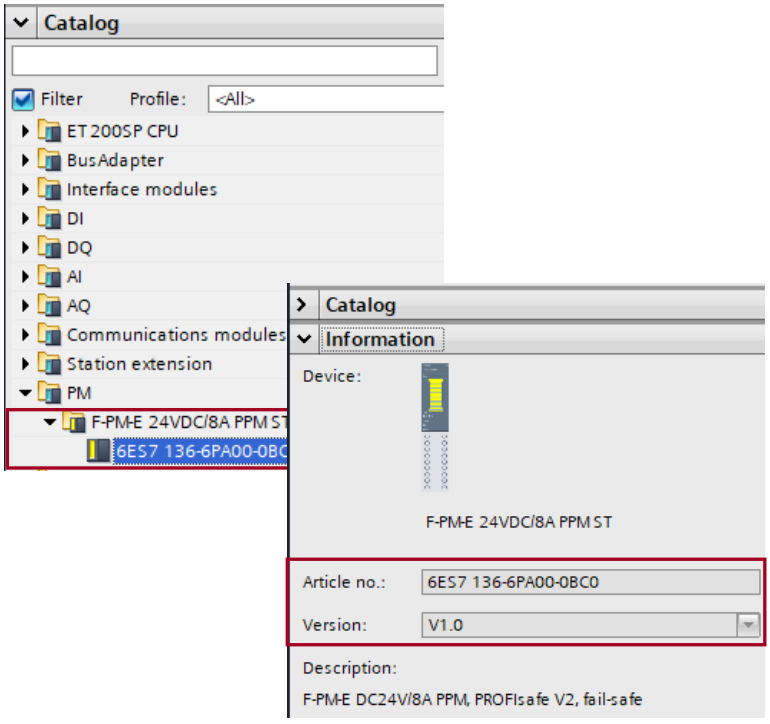
Notes:

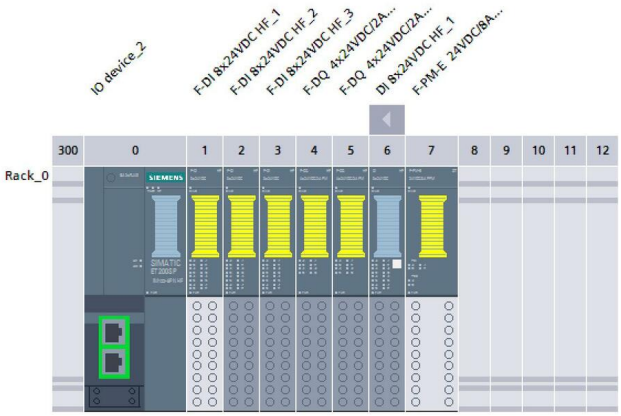
Slot 6: 6ES7131-6BF00-0CA0 (DI 8 x 24V DC)





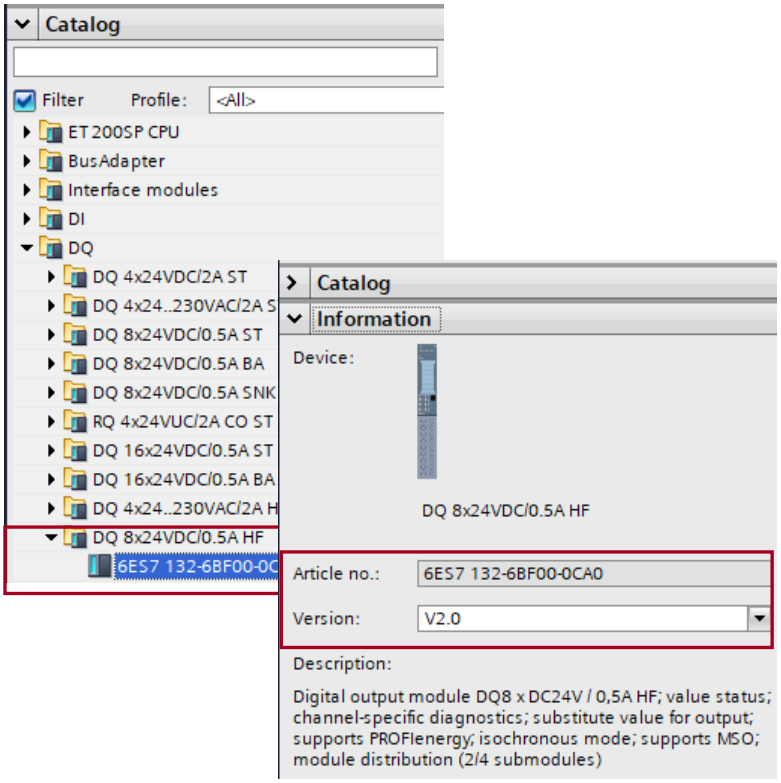
Slot 7: 6ES7136-6PA00-0BC0 (F-PM-E 24V DC)

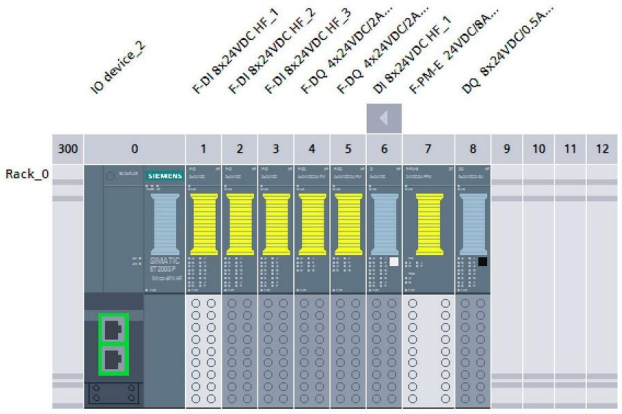




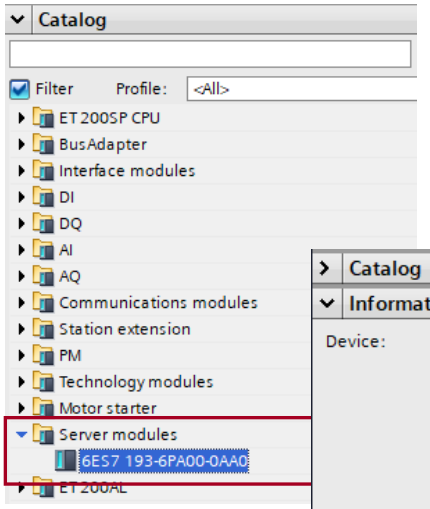
Notes:

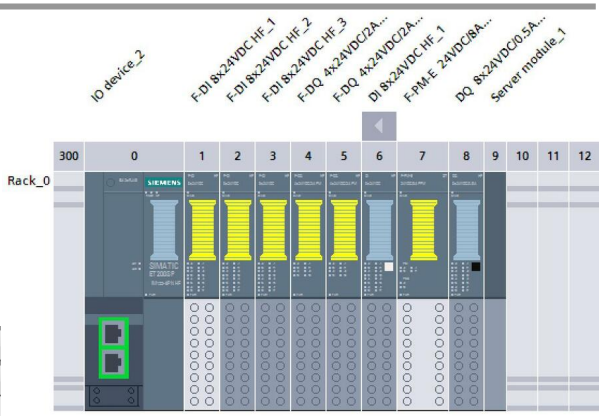
Slot 8: 6ES7132-6BF00-0CA0 (DQ 8X24V DC 24V DC)





Slot 9: 6ES7193-6PA00-0AA0 (Server module)





Notes:

Assigning the Input/Output addresses.

Default addresses of the new hardware:

Device overview

Module	...	Rack	Slot	I address	Q address	Type	Article number	Firmware	Comment	Access
		0	300							
IO device_2		0	0			IM 155-6 PN HF	6ES7 155-6AU00-0CN0	V3.3		PLC_1
PROFINET interface		0	0 X1			PROFINET interface				PLC_1
F-DI 8x24VDC HF_1		0	1	0...5	0...3	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0	V1.0		PLC_1
F-DI 8x24VDC HF_2		0	2	6...11	6...9	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0	V1.0		PLC_1
F-DI 8x24VDC HF_3		0	3	12...17	12...15	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0	V1.0		PLC_1
F-DQ 4x24VDC/2A PM HF_1		0	4	18...22	18...22	F-DQ 4x24VDC/2A ...	6ES7 136-6DB00-0CA0	V1.0		PLC_1
F-DQ 4x24VDC/2A PM HF_2		0	5	23...27	23...27	F-DQ 4x24VDC/2A ...	6ES7 136-6DB00-0CA0	V1.0		PLC_1
DI 8x24VDC HF_1		0	6	28		DI 8x24VDC HF	6ES7 131-6BF00-0CA0	V2.0		PLC_1
F-PM-E 24VDC/8A PPM ST_1		0	7	29...35	29...33	F-PM-E 24VDC/8A P...	6ES7 136-6PA00-0BC0	V1.0		PLC_1
DQ 8x24VDC/0.5A HF_1		0	8		4	DQ 8x24VDC/0.5A ...	6ES7 132-6BF00-0CA0	V2.0		PLC_1
Server module_1		0	9			Server module	6ES7 193-6PA00-0AA0	V1.1		PLC_1

The addresses are set in accordance to the manufacturers requirements::

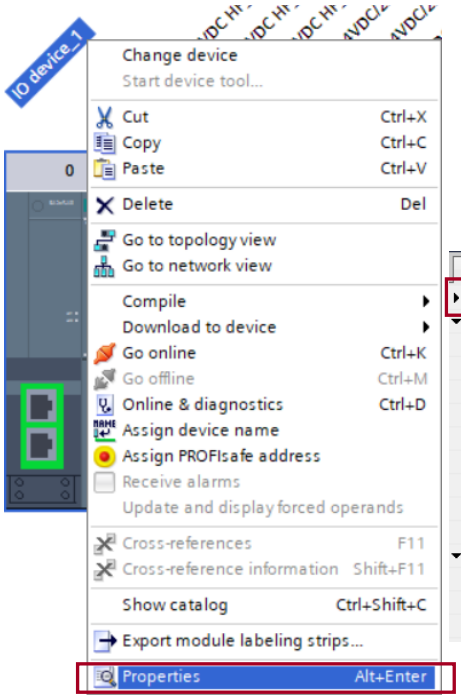
Device overview

Module	...	Rack	Slot	I address	Q address	Type	Article number
		0	300				
IO device_2		0	0			IM 155-6 PN HF	6ES7 155-6AU00-0CN0
PROFINET interface		0	0 X1			PROFINET interface	
F-DI 8x24VDC HF_1		0	1	512...517	512...515	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0
F-DI 8x24VDC HF_2		0	2	518...523	518...521	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0
F-DI 8x24VDC HF_3		0	3	524...529	524...527	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0
F-DQ 4x24VDC/2A PM HF_1		0	4	530...534	530...534	F-DQ 4x24VDC/2A ...	6ES7 136-6DB00-0CA0
F-DQ 4x24VDC/2A PM HF_2		0	5	535...539	535...539	F-DQ 4x24VDC/2A ...	6ES7 136-6DB00-0CA0
DI 8x24VDC HF_1		0	6	100		DI 8x24VDC HF	6ES7 131-6BF00-0CA0
F-PM-E 24VDC/8A PPM ST_1		0	7	540...546	540...544	F-PM-E 24VDC/8A P...	6ES7 136-6PA00-0BC0
DQ 8x24VDC/0.5A HF_1		0	8		100	DQ 8x24VDC/0.5A ...	6ES7 132-6BF00-0CA0
Server module_1		0	9			Server module	6ES7 193-6PA00-0AA0

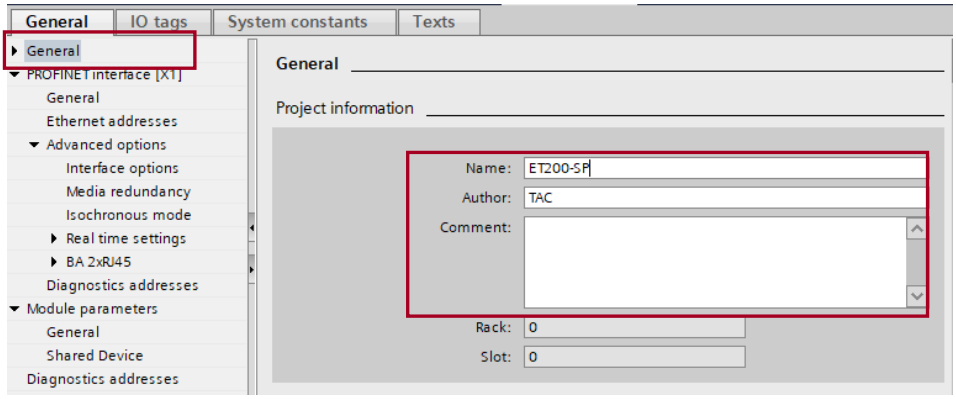
Example: the I/O address ranges that are used with the examples within this document.

Notes:

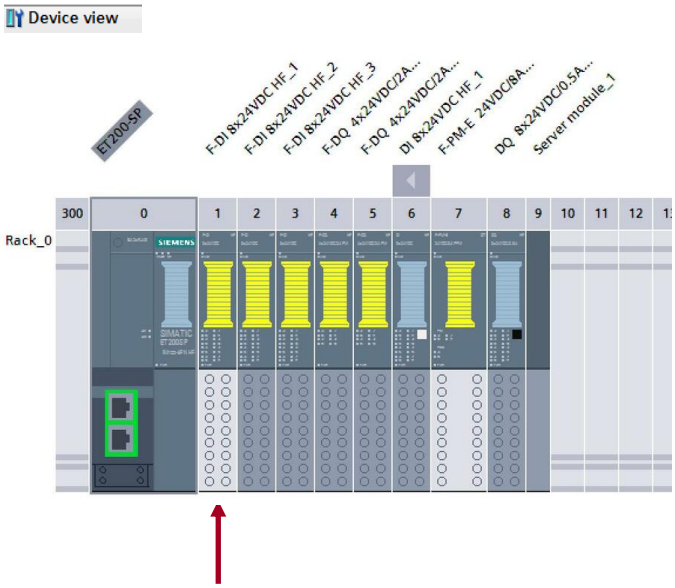
Assigning a name etc. to the IM155-6PN:



- Right click on the ET200-SP and select ***Properties.***
- Select the ***General*** tab.
- Type in the name e.g. ET200-SP, and other information if required.



Setting the Properties of the Input/Output Modules:



Potential groups:

The potential groups are created in the ET200SP distributed I/O system by a systematic arrangement of the Base Units.

For the formation of potential groups, ET200SP distinguishes between 2 Base Units:

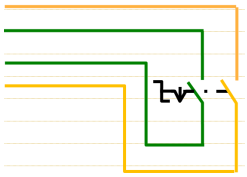
- Base Units BU...D (light coloured terminal block and DIN rail mounting): Opening of a new potential group (power and AUX bar is interrupted to the left)
  - Infeed of the supply voltage L+ up to an infeed current of 10 A.
- Base Units BU...B (dark terminal block and DIN rail mounting): Continuing the potential group (power and AUX bar continued w/o interruption) Tapping the supply voltage L+ for external components or Looping through with a maximum total current of 10 A.

Notes:

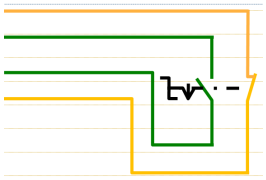
Switch Contact Arrangements.

Two channel switching devices can be of the ***equivalent*** type - where both channels are monitored for simultaneously being open or closed, or the ***non-equivalent*** type - where one of the channels is monitored for being open, while the second for being closed, depending on the relevant switch/button/relay setting.
Non-equivalent types are useful in the case where one supply for both channels is used.
If a wire break in the supply occurs the system will react with a discrepancy error.

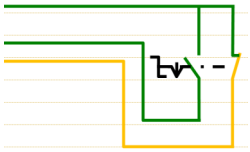
Example of a ***2 channel equivalent*** switch:



Example of a ***2 channel non-equivalent*** switch with a separate supply for each channels:

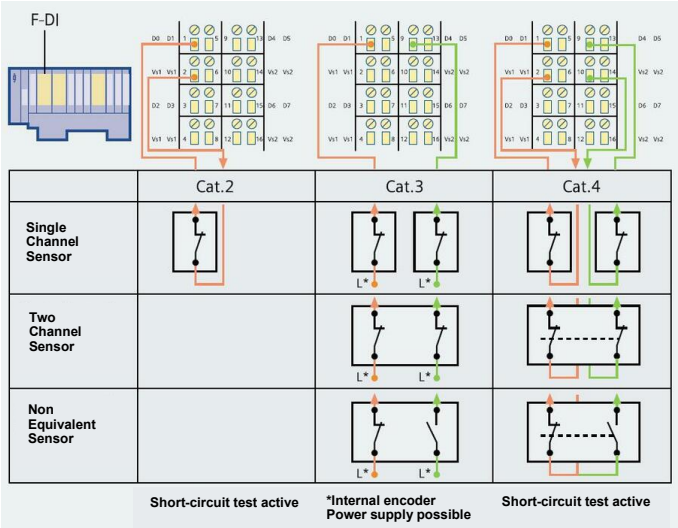


Example of a ***2 channel non-equivalent*** switch with a single supply to both channels:



The different safety categories achievable depending upon switch contact arrangements and use of internal/external supplies.

The following diagram shows the Safe Integrity Level (SIL category 2 to 4) which can be achieved using different switch/sensor types and supplies:



Training Equipment F-DI Module 1

Slot 1: 6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC)

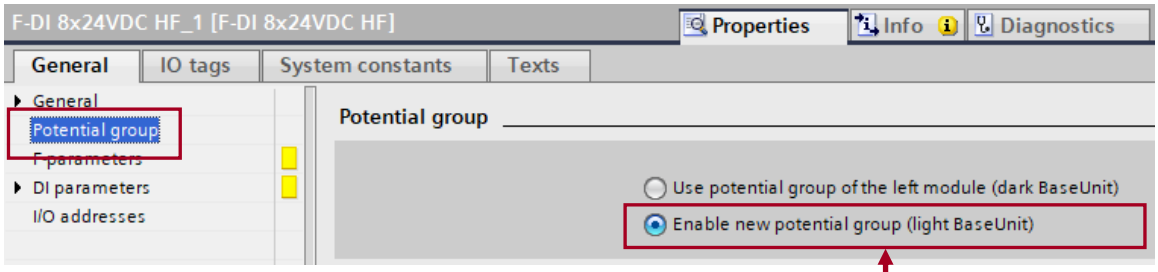
Slot 1		6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC) New potential group.				
F-PLC inputs	Channels	Function	Short circuit test ?	Switch/sensor type	Discrepancy time	Sensor supply
I512.0	0 & 4	Emergency Stop	×	1002 equivalent	500ms	External
I512.1	1 & 5	F-DI 1 & 5	×	1002 equivalent	500ms	External
I512.2	2 & 6	Door Locked Switch	✓	1002 equivalent	500ms	2 & 6
I512.3	3 & 7	Door Closed Switch	✓	1002 non-equivalent	500ms	3 & 7

Training Equipment Features of F-DI Module 1:

- An ***external supply*** exists for the inputs for **emergency stop** and so the short-circuit test must therefore be deactivated.
- The 2-channel input for **Safety door closed** is connected to a non-equivalent switch type.
  For non-equivalent devices, the **NC contact** must be wired **to the lower channel address** of the input module to ensure that the 0 signal can be evaluated in the F-program when the button is actuated.

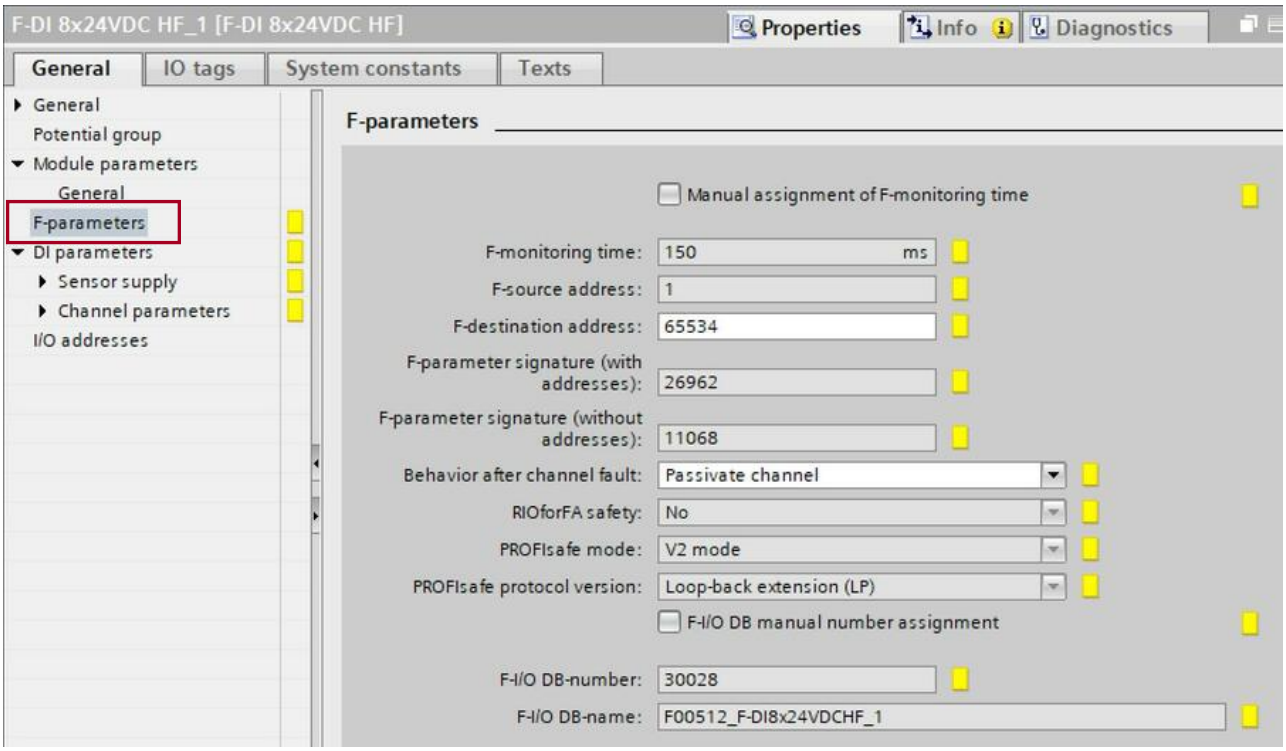
Notes:

- The **discrepancy times are increased** here to prevent sporadic errors occurring upon switching the emergency stop, the acknowledgment button and the safety door. (The discrepancy time must be adapted in accordance with the respective equipment being used, beginning with the standard value)
- Max. PLd (Cat.3) / SIL2 is attained for external supplies (emergency stop = Channel 0 & 4, acknowledgment button = Channel 1 & 5).
- Max. PLe (Cat.4) / SIL3 is attained for internal supplies (safety door locked = Channel 2 & 6, safety door closed = Channel 3 & 7).



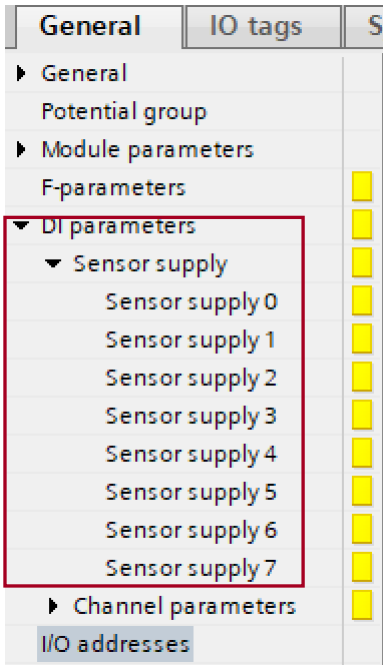
The potential group is set to match the hardware base unit. In this case it is a light coloured base unit, therefore a new potential group is enabled.

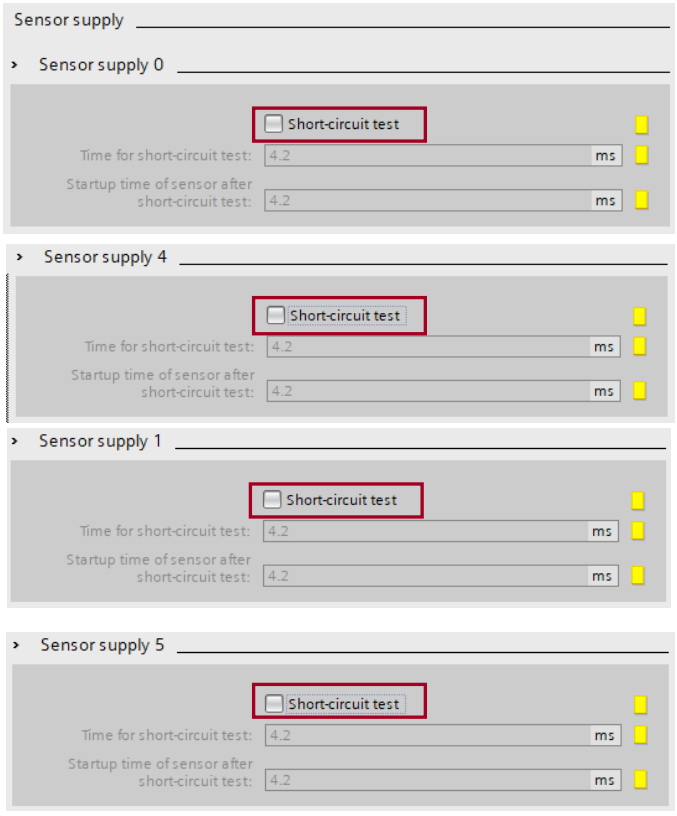
F-Parameter Settings: various setting can be made depending on the application. No changes are made here for the training.



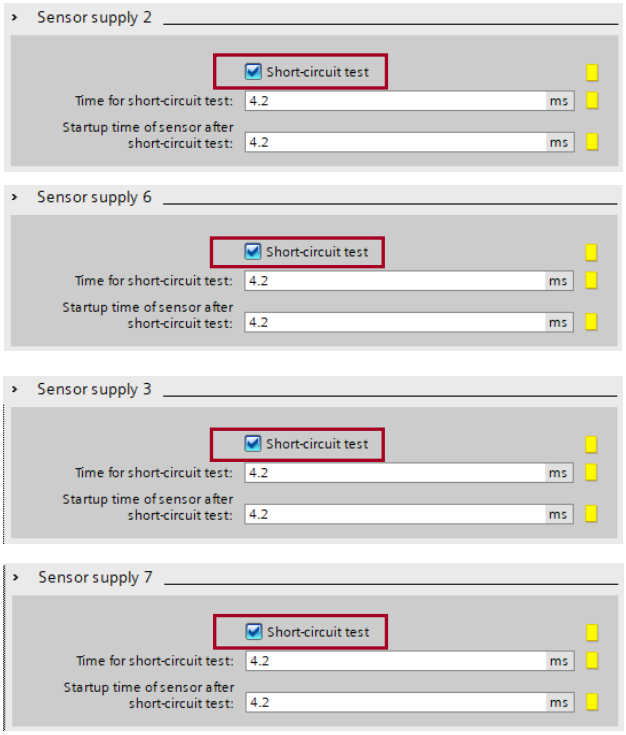
Notes:

Sensor Supply Settings:



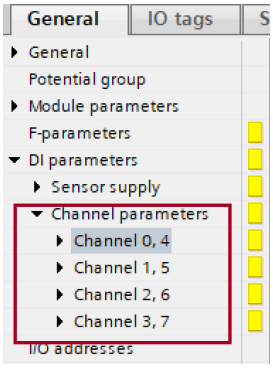


Enable the ***Short circuit test*** option for the sensor supplies 2 & 6 and 3 & 7:



Channel Parameters:

Each safe input consists of two channels. The sensor type etc. are set here.
The following settings match the sensor types found on the training equipment.

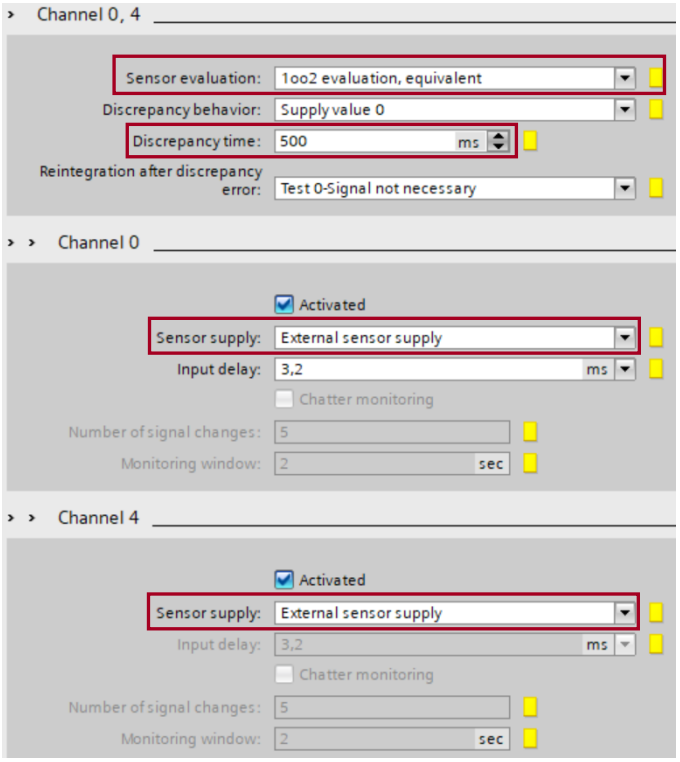


Discrepancy Time:

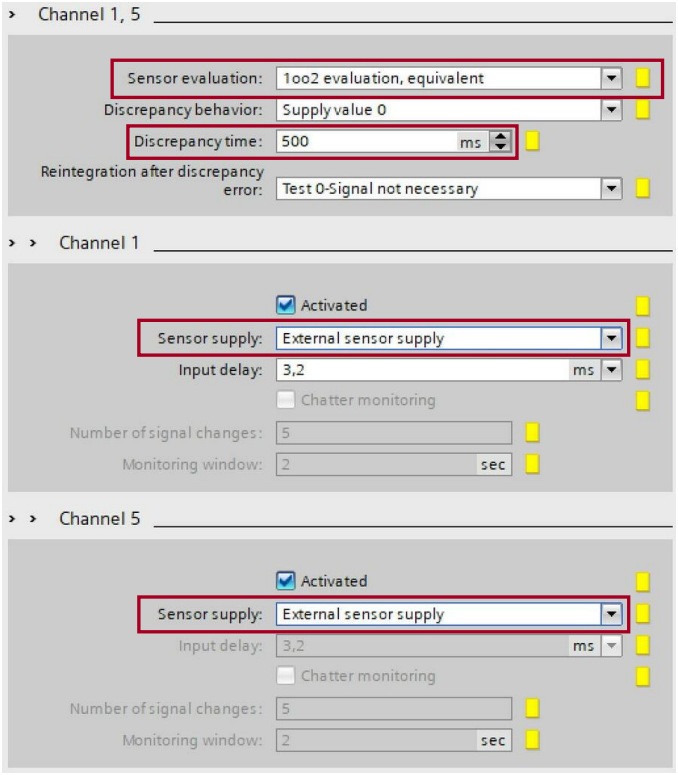
The discrepancy time is used to eliminate channel errors caused by the mechanical characteristics of switches/relays etc.
Setting a higher value e.g. 500ms will help eliminate nuisance errors during the commissioning stages. This can be matched later to the connected devices.

Notes:

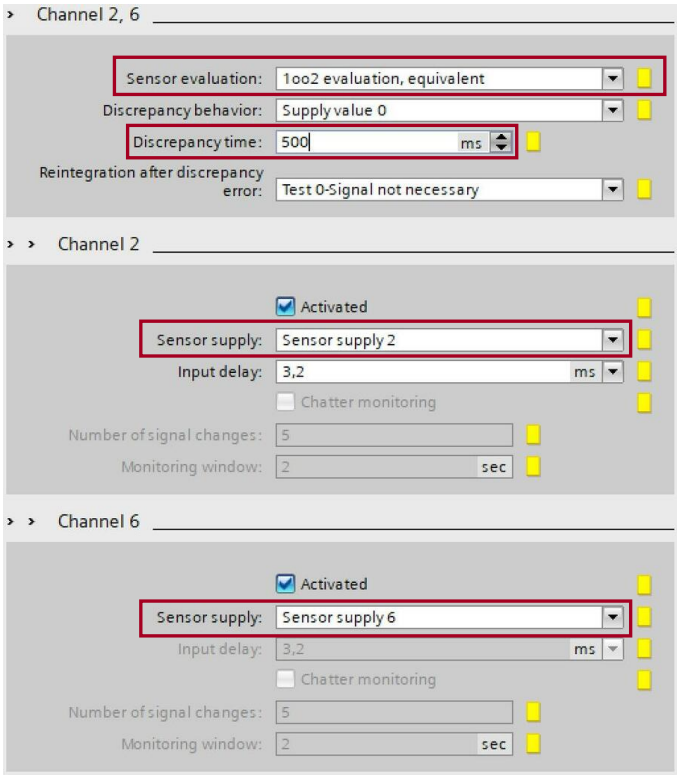
Emergency Stop switch - channels 0 & 4 - settings:



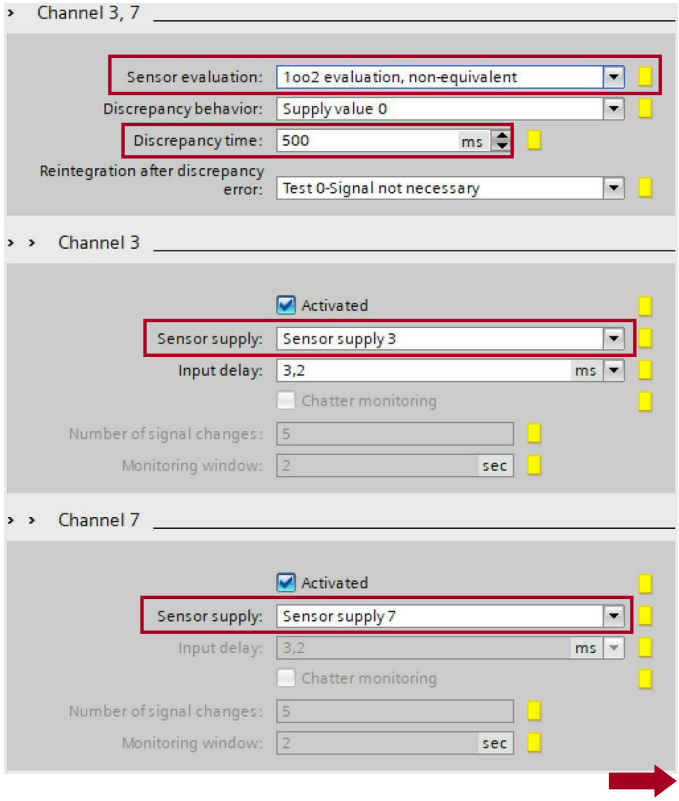
Hold to run button - channels 1 & 5 - settings:



Door locked - channels 2 & 6 - settings:

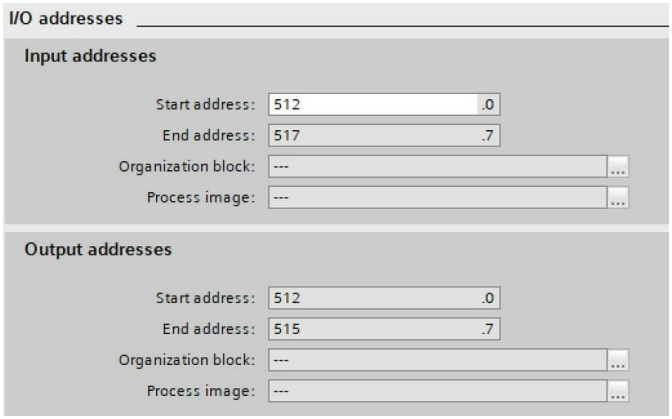


Door closed - channels 3 & 7 - settings:



Notes:

The previously set I/O addresses can also be viewed and changed here:



Training Equipment F-DI Module 2

Slot 2: 6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC)

Slot 2		6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC)				
F-PLC inputs	Channels	Function	Short circuit test ?	Switch/sensor type	Discrepancy time	Sensor supply
I518.0	0 & 4	F-DI 0 & 4	✓	1002 equivalent	500ms	0 & 4
I518.1	1 & 5	ESTOP Reset PB	✓	1002 non-equivalent	500ms	1 & 5
I518.2	2 & 6	Set-up Switch	✓	1002 equivalent	500ms	2 & 6
I516.3	3 & 7	F-DI 3 & 7	✓	1002 equivalent	500ms	3 & 7

Training Equipment Features of F-DI Module 2:

- An ***internal supply*** is used for all of the inputs. It is therefore possible to activate the short circuit test.
- The 2-channel input for **Reset** button is connected to a non-equivalent switch type.
  For non-equivalent devices, the **NC contact** must be wired **to the lower channel address** of the input module to ensure that the 0 signal can be evaluated in the F-program when the button is actuated.
- The **discrepancy times are increased** here to prevent sporadic errors occurring upon switching the devices.
  discrepancy time must be adapted in accordance with the respective equipment being used)

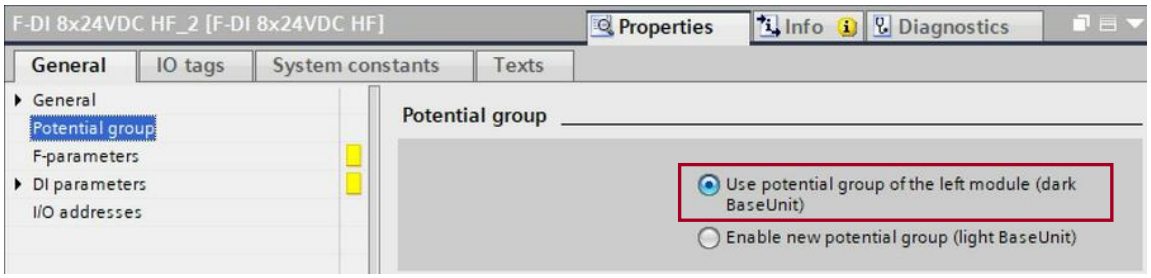
Notes:

- Max. PLe (Cat.4) / SIL3 is attained for internal supplies - all the devices connected to F-DI Module 2

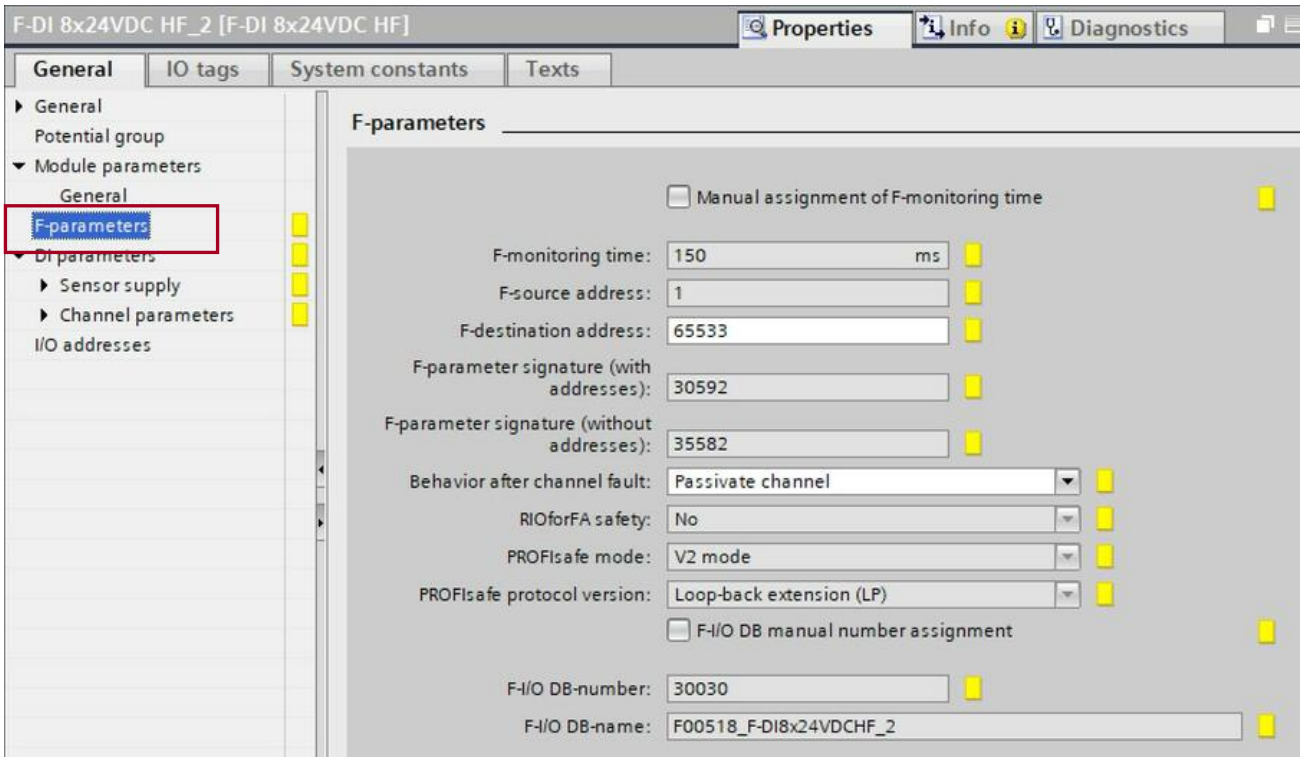
Right: internal supply connection requirements. When the internal supply is used it is necessary to connect the inputs to the correct supply

Channel	CH0	CH4	CH1	CH5	CH2	CH6	CH3	CH7
Supply	VS0	VS4	VS1	VS5	VS2	VS6	VS3	VS7
Input	DI0	DI4	DI1	DI5	DI2	DI6	DI3	DI7

Potential Group: The unit has a dark base and uses the potential group from the left module.

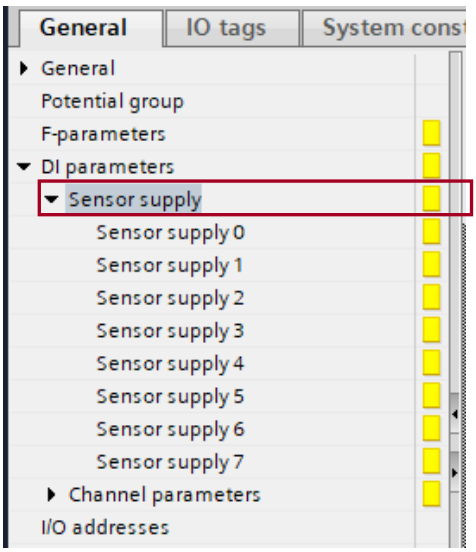


General Settings - the default settings are used.



Notes:

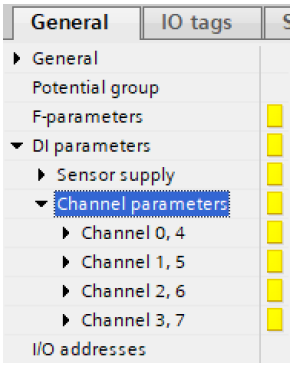
Sensor Supply - all the devices connected to the module are using the internal sensor supply. It is therefore possible to use the ***Short circuit test***.





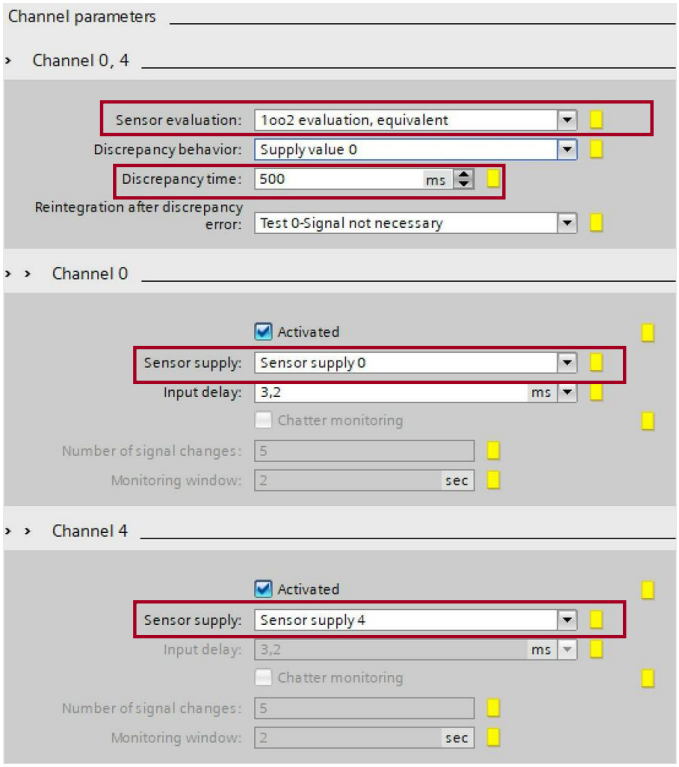


Channel Parameters - the settings on the following page are required for the training equipment.

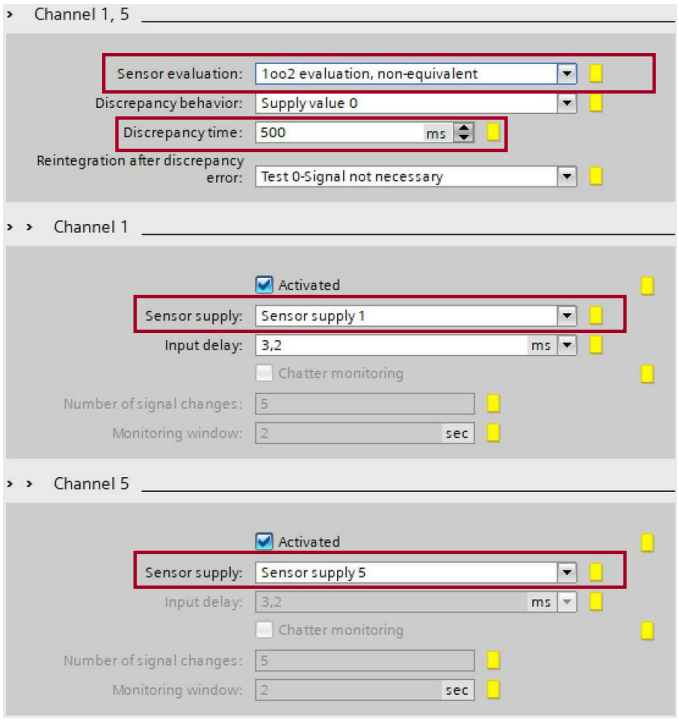


Notes:

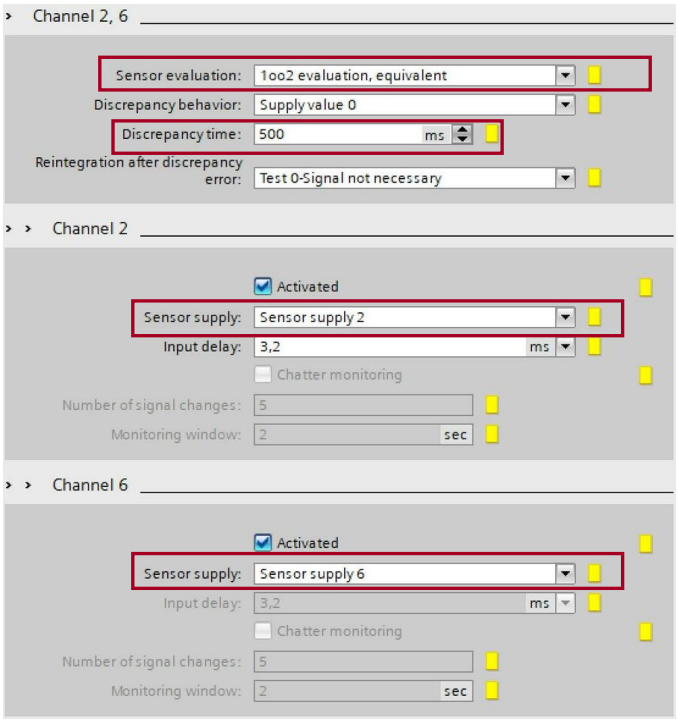
End Limit Override switch - channels 0 & 4 - settings:



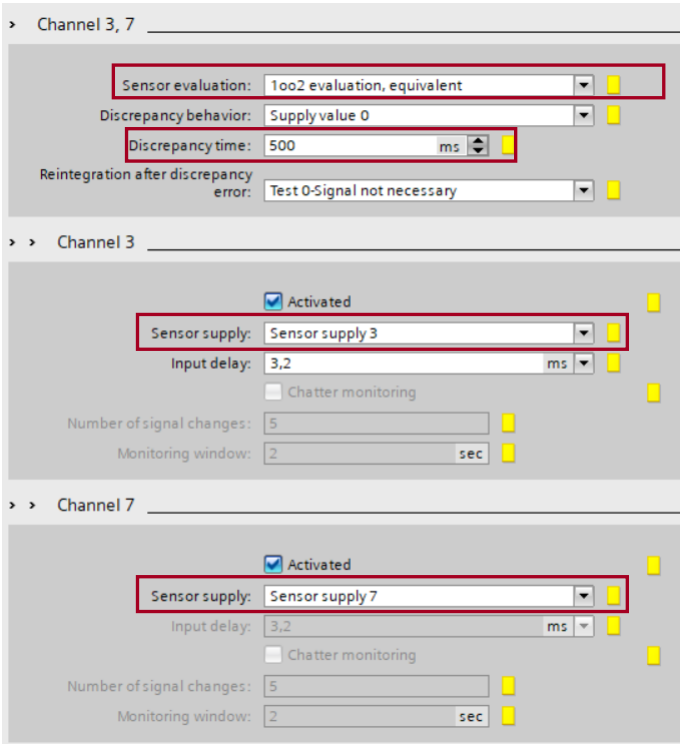
Emergency Stop Reset button - channels 1 & 5 - settings:



Set-up Mode switch - channels 2 & 6 - settings:



Mode 3 switch - channels 3 & 7 - settings:



Notes:

Training Equipment F-DI Module 3

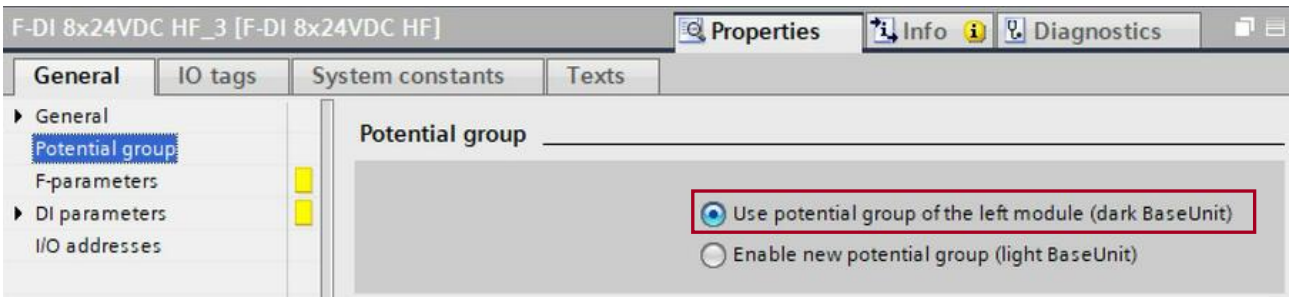
Slot 3: 6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC)

Slot 3		6ES7136-6BA00-0CA0 (F-DI 8 x 24V DC)				
F-PLC inputs	Channels	Function	Short circuit test ?	Switch/sensor type	Discrepancy time	Sensor supply
I524.0	0 & 4	Hold-to-Run PB	×	1002 equivalent	500ms	External
I524.1	1 & 5	F-DI 1 & 5	×	1002 equivalent	500ms	External
I524.2	2 & 6	F-DI 2 & 6	×	1002 equivalent	500ms	External
I524.3	3 & 7	F-DI 3 & 7	×	1002 equivalent	500ms	External

Training Equipment Features of F-DI Module 3:

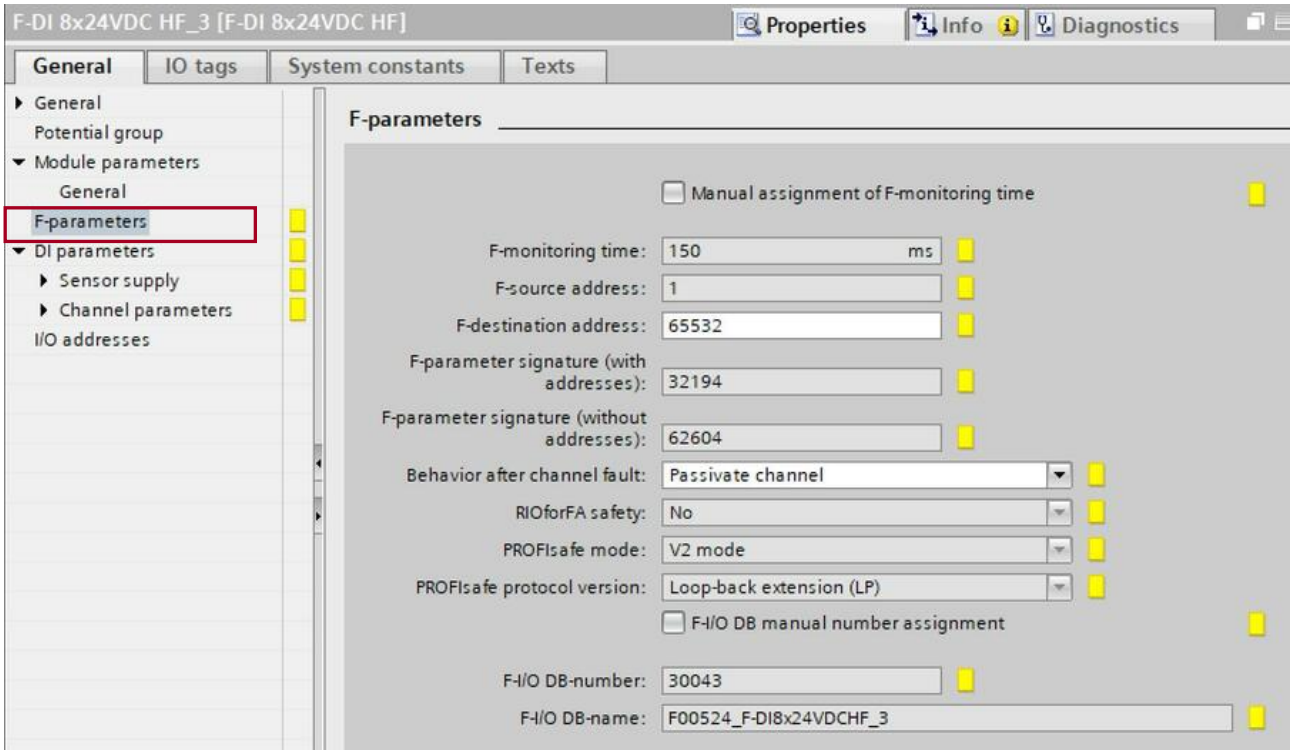
- An ***external supply*** is used for all of the inputs. It is therefore not possible to activate the short circuit test.
- The 2-channel input for **Hold-to-Run PB** is connected to an equivalent switch type. The 6 other individual inputs have no specific function for the training and are free to be assigned for experimental/testing purposes. The following example shows the switches configured as 4 channels - 2 equivalent channels per safe input.
- The **discrepancy times are increased** here to prevent sporadic errors occurring upon switching the devices. (discrepancy time must be adapted in accordance with the respective equipment being used)
- Max. PLe (Cat.3) / SIL2 is attained for external supplies - all the switches connected to F-DI Module 3

Potential Group: The unit has a dark base and uses the potential group from the left module.

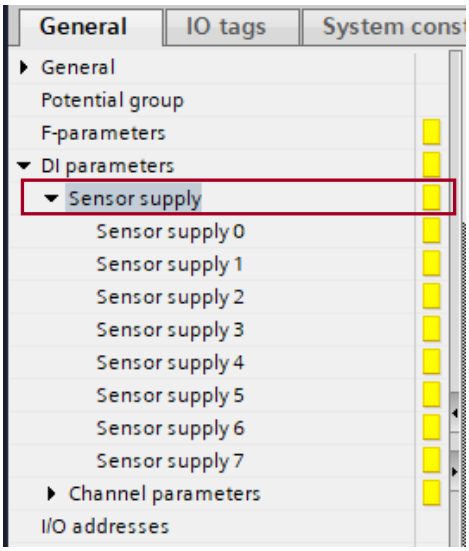


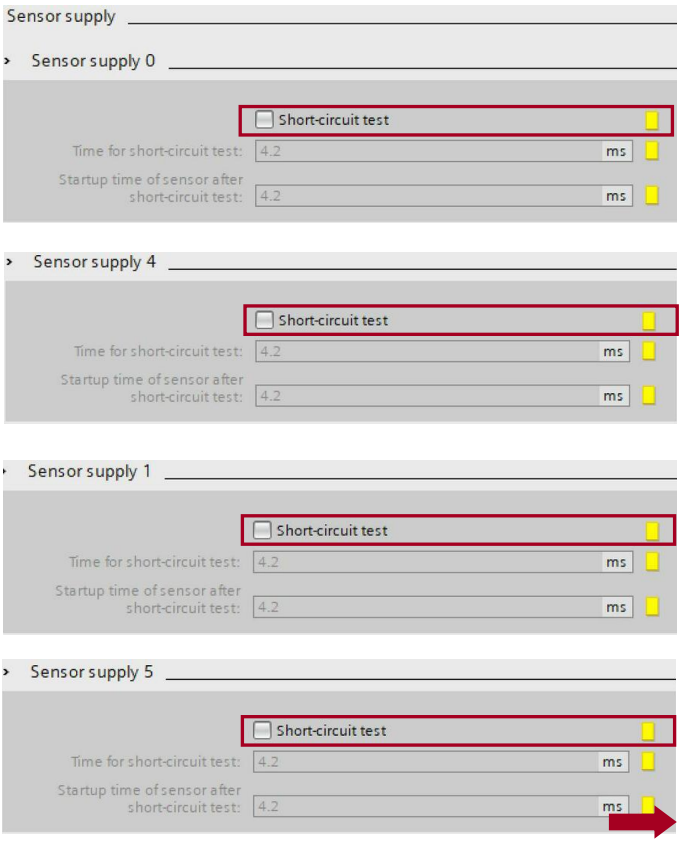
Notes:

General Settings - the default settings are used.

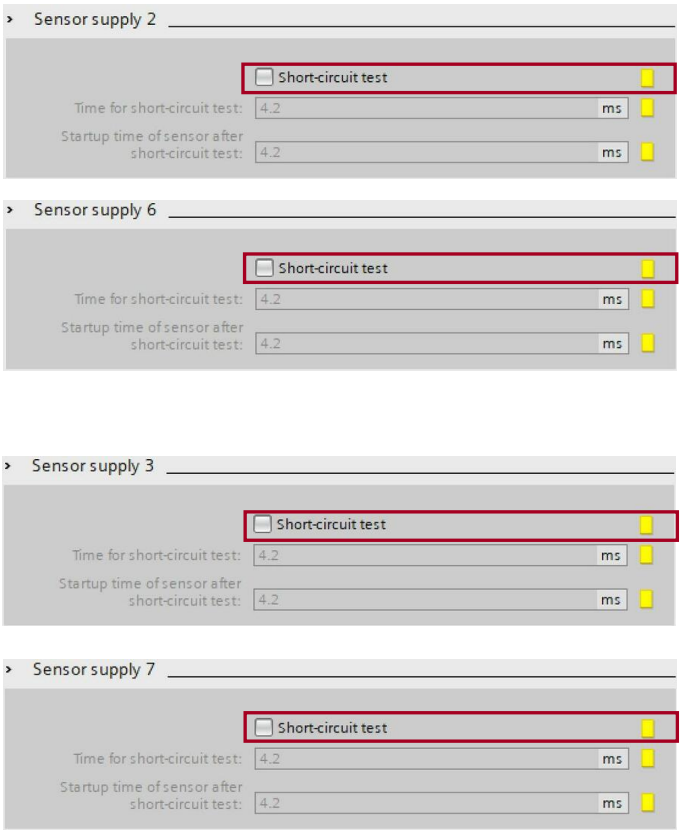


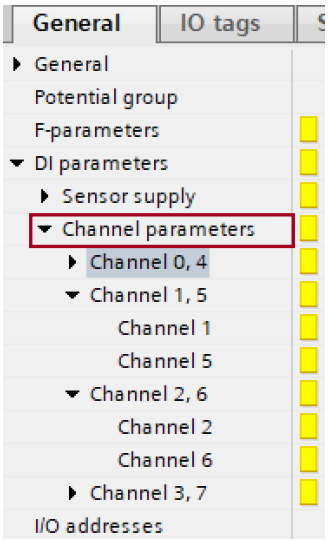
Sensor Supply - all the switches connected to the module are using an external sensor supply. It is therefore not possible to use the ***Short circuit test***.



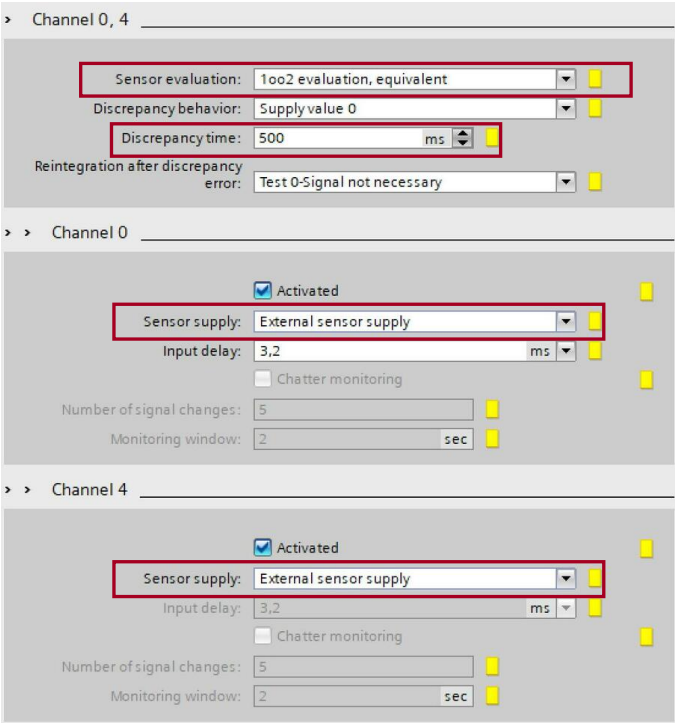


Notes:

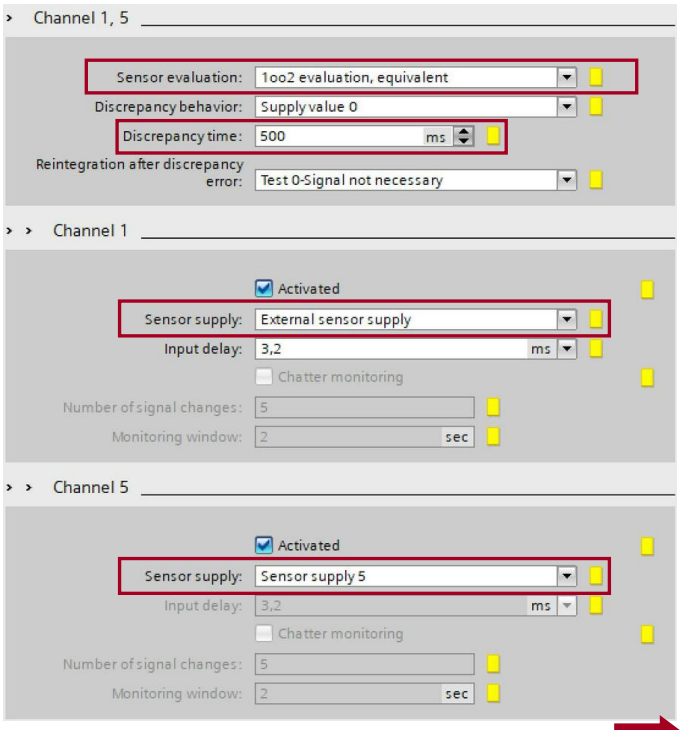




F-DI 0 & 4 - channels 0 & 4 - settings:

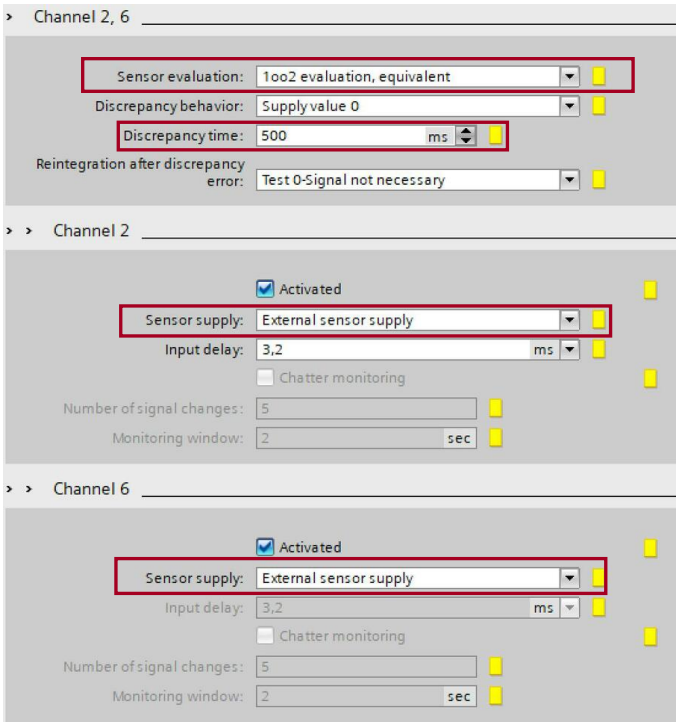


F-DI 1 & 5 - channels 1 & 5 - settings:

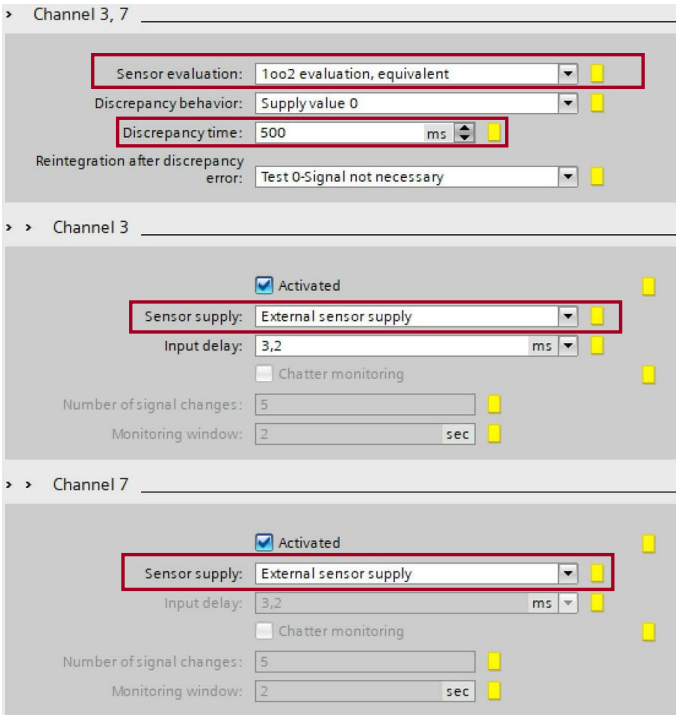


Notes:

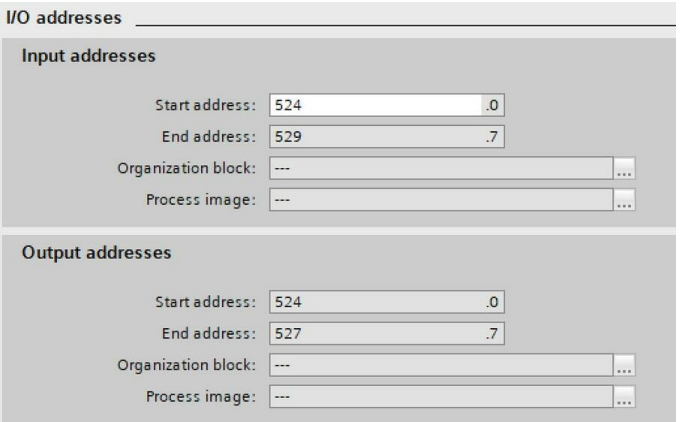
F-DI 2 & 6 - channels 2 & 6 - settings:



F-DI 3 & 7 - channels 3 & 7 - settings:



The previously set I/O addresses can also be viewed and changed here:



Notes:

Training Equipment F-DQ Module 4

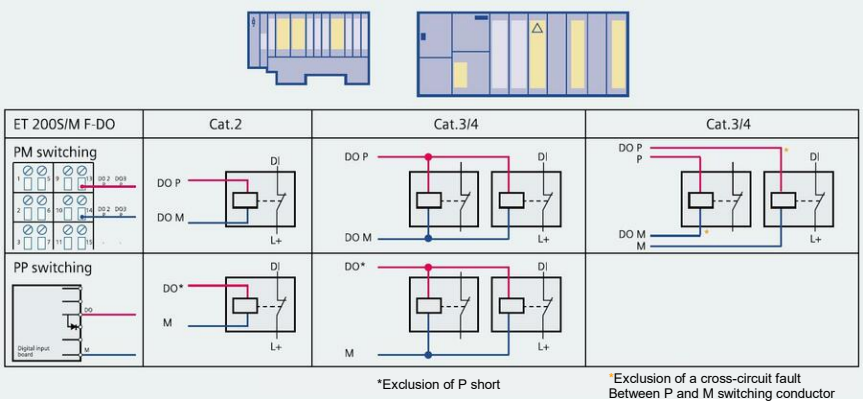
Slot 4: 6ES7136-6DB00-0CA0 (F-DQ 4 x 24V DC)

Slot 4		6ES7136-6DB00-0CA0 (F-DQ 4 x 24V DC)		
F-PLC outputs	Channel	Function	Light test?	Wire break?
Q530.0	0	K1 relay/Q0 LED	✓	✓
Q530.1	1	K2 relay/Q1 LED	✓	✓
Q530.2	2	Internal Event LED	✓	✓
Q530.3	3	ESTOP Ack Req LED	✓	✓

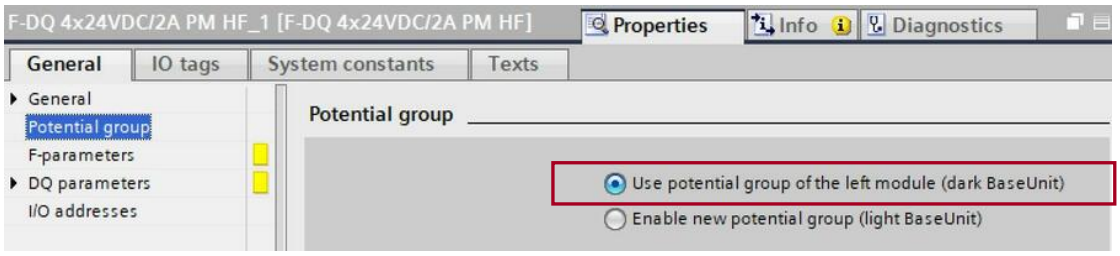
Training Equipment Features of F-DQ Module 4:

- Two relays, K1 / K2, are controlled by PM-switching of F-DQ 0 and F-DQ 1.
- If the relays are qualified actuators, SIL3 / Cat.4 / PLe will be attained.
- Series connection of the relay contacts corresponds with SIL3 / Cat.4 / PLe and can be used for the control of auxiliary motors (chip conveyors), for example.
- The relays can be used as part of the Test Stop routine by monitoring the series contacts, via an input, when the relays are energized/de-energized during the test.
- Two outputs are also used for Internal Event and Emergency Stop Acknowledgment required LEDs

The different safety categories achievable depending upon connection of safety-related actuators and plus-minus /plus-plus switching

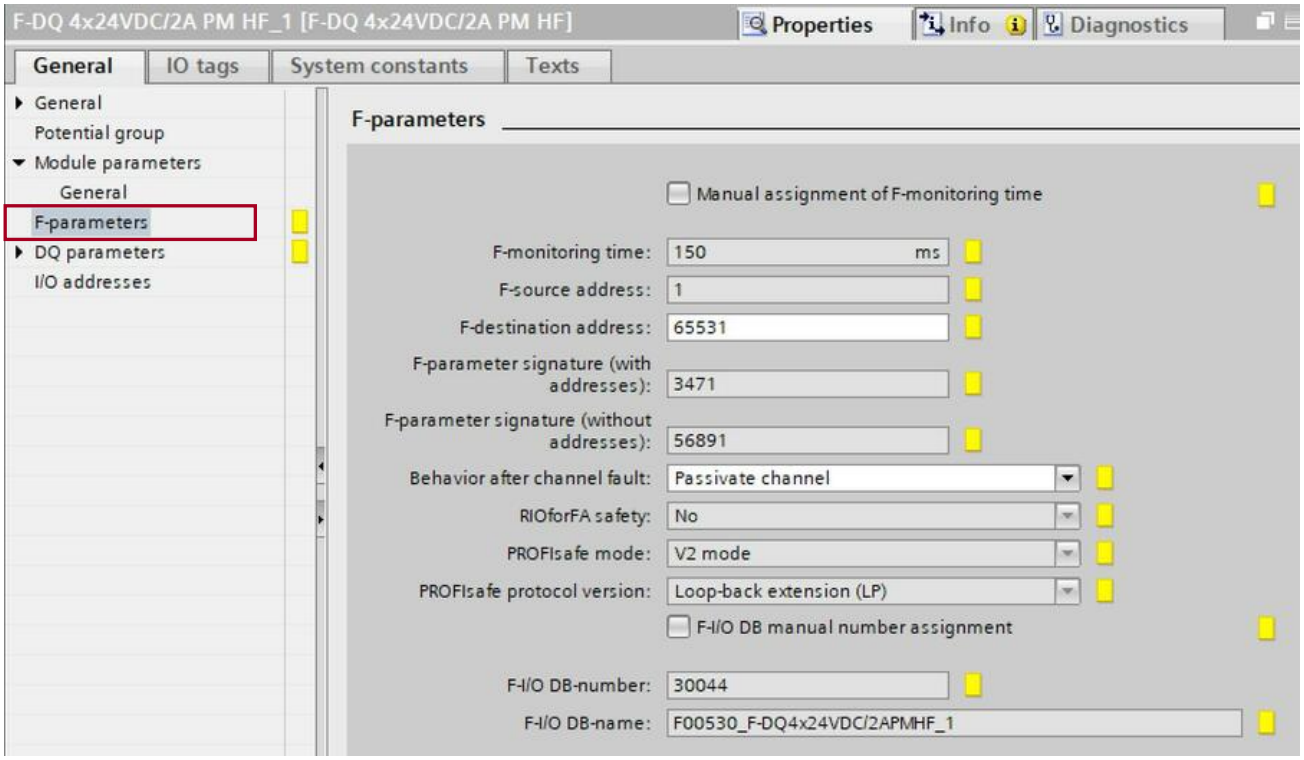


Potential Group: The unit has a dark base and uses the potential group of the left module.



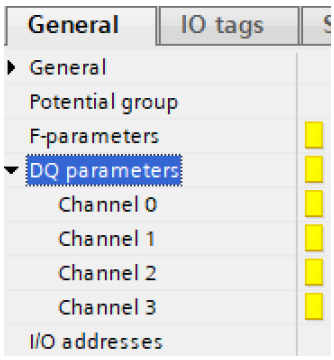
Notes:

General Settings - the default settings are used.



DQ Channel Parameters:

The digital output channel parameters can be individually set.



Maximum test time:
This parameter allows the time within which the light test, dark test and switch-on test are performed module-wide to be defined.
The tests are repeated when this time expires. In the event of an error, the test time is shortened to 60 seconds.

The ***Maximum test time*** parameter is a module parameter, thus the test cycle for the entire fail-safe output module is performed within the parameterized maximum test time.
If the bit pattern test is not executed in the parameterised time (or in the test time shortened to 60 s in the event of an error), the module is switched to error status.

Max. read-back time dark test:
Dark tests are switch-off tests for the bit pattern test.
During the dark test, a test signal is switched to the output channel while the output channel has a 1-signal.
The output channel is then switched off briefly (= dark time) and read back.
The speed of the test means that actuator does not react and remains switched on.
The time for the read-back is set with this parameter. If the anticipated signals could not be correctly read back following expiry of the dark test read-back time, the output channel is passivated.
While a bit pattern is active (switch test is being performed), no new process values are switched to the output channels.
A higher maximum read-back time thus increases the reaction time of the F-module for the dark test.

Notes:

Max. read-back time switch-on test:
The switch-on test is an integral part of the bit pattern test.
For the switch-on test in PM mode, the P-switch and M-switch of the output channel are closed and read back alternately if the output channel is inactive (output signal 0). In contrast to the light test, no current flows through the connected load for the switch-on test.
The time for the read-back is set with this parameter. If the anticipated signals could not be correctly read back following expiry of this time, the output channel is passivated. The switch-on test identifies the following faults:

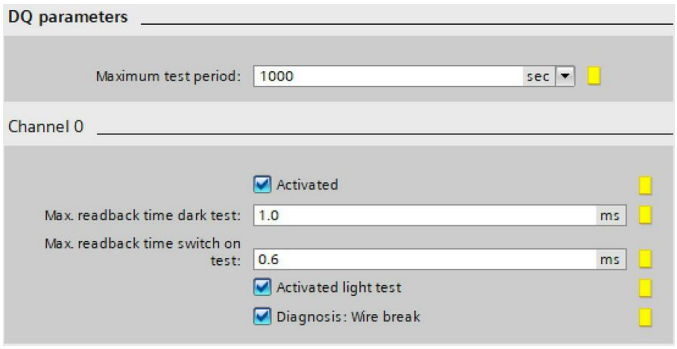
- Short-circuit to L+ with output signal "0"
- Short-circuit to M with output signal "0"

Light test activated:
With a 0 signal at the output, overload and wire break are detected. Light test and dark test are part of the module-internal bit pattern test.
For the light test, the fail-safe output module switches test-contingent 1-signals to the output while the output is inactive (process value = 0). The speed of the test means that actuator does not react and remains switched off.
In contrast to the switch-on test, the P-switch and M-switch switch simultaneously for the light test and current flows through the connected load.

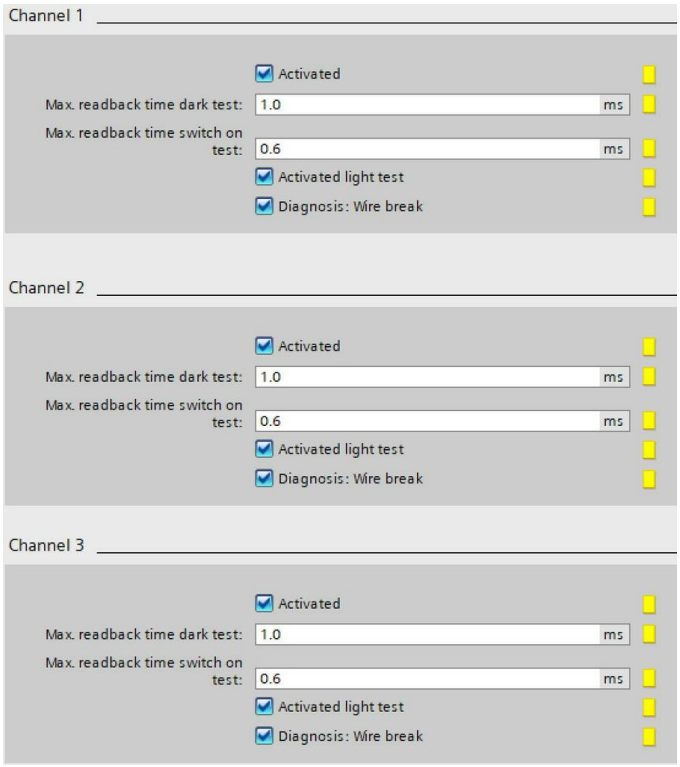
Diagnostics: wire break
A wire break test is applied for monitoring of the connection from the output channel to the actuator.
Upon activation of the checkbox, the wire break monitoring for the corresponding channel is activated.
To detect a wire break with a "0" output signal, you must activate the light test.
Warning:
The wire break diagnostics of the outputs is not a safety integrated feature and therefore cannot be evaluated for safety-orientated activities.

Set the required DQ settings for the training equipment:

The settings for channel 0:



The settings for channels 1 to 3:



Notes:

Training Equipment F-DQ Module 5

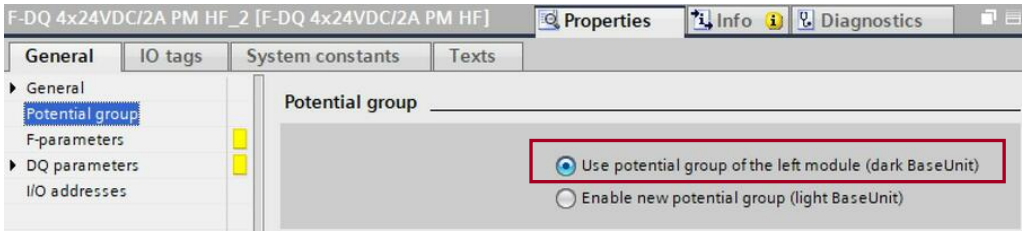
Slot 5: 6ES7136-6DB00-0CA0 (F-DQ 4 x 24V DC)

Slot 5		6ES7136-6DB00-0CA0 (F-DQ 4 x 24V DC)		
F-PLC outputs	Channel	Function	Light test?	Wire break?
Q535.0	0	F-DO 0 LED	×	×
Q535.1	1	F-DO 1 LED	×	×
Q535.2	2	F-DO 2 LED	×	×
Q535.3	3	F-DO 3 LED	×	×

Training Equipment Features of F-DQ Module 5:

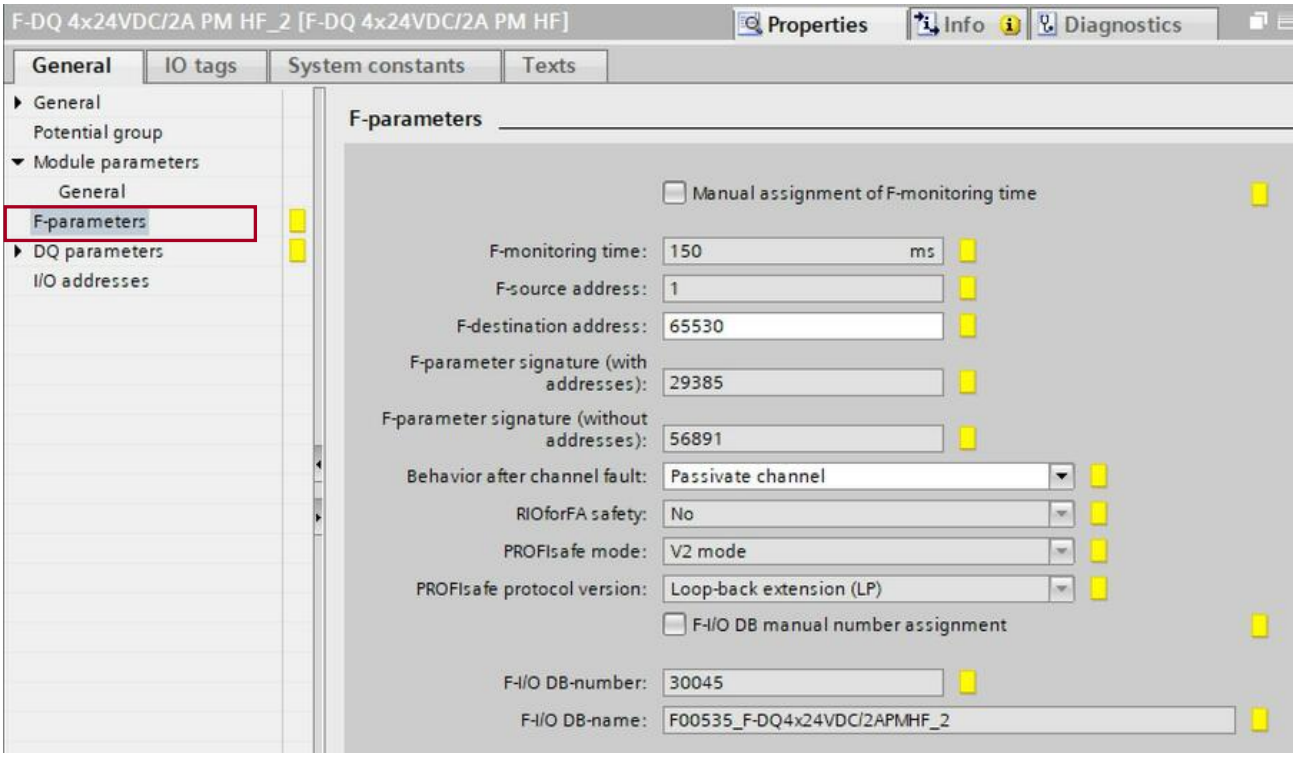
- The output module is used for driving 4 LEDs and has no Safety Integrated functionality on the training equipment..
- The **light test** and the **wire break diagnostics** must be deactivated for the channels, otherwise the following alarm will be issued due to the module wiring: **430003 F-module 535: CRC/ sequence number error detected by F-I/O**
- In order to attain SIL3 / Cat.4 / PLe with this PM-switching module, Safety Integrated certified actuators must be used.

Potential Group: The unit has a dark base and uses the potential group from the left module.

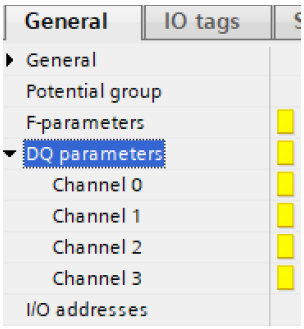


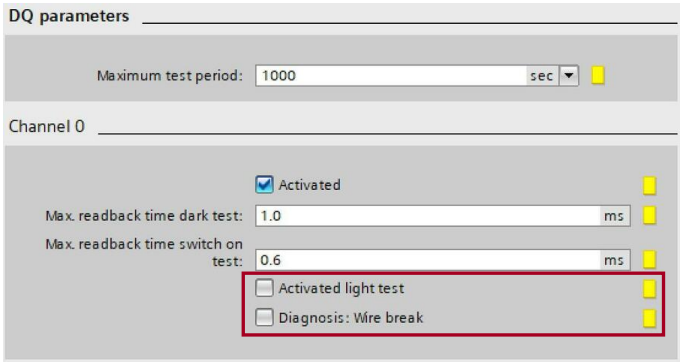
Notes:

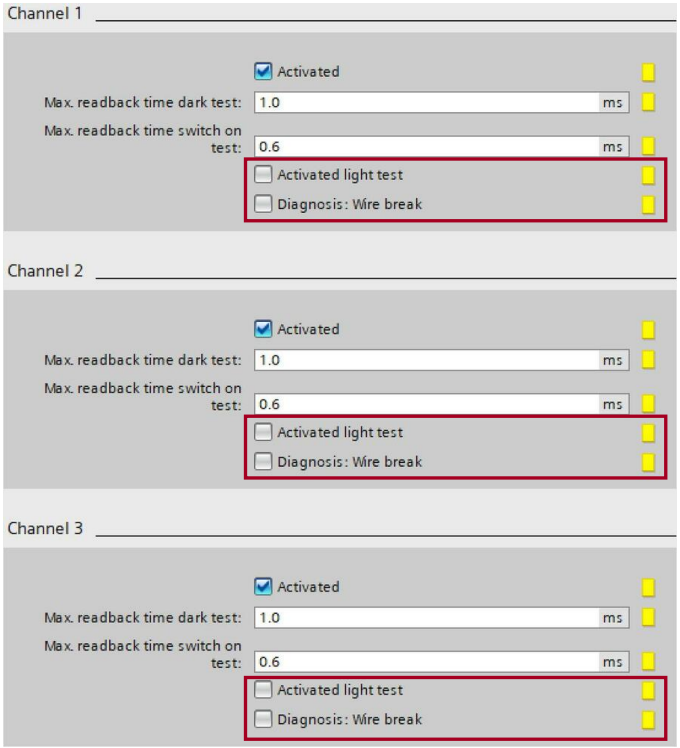
General Settings - the default settings are used.



DQ Parameters - the ***Light test*** and ***Wire break*** tests need to be deactivated.

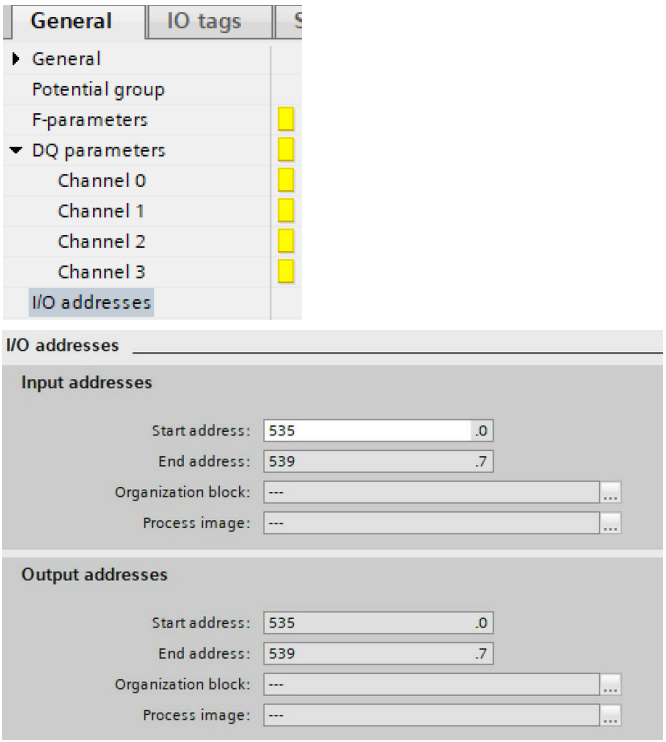






Notes:

The previously set I/O addresses can also be viewed and changed here:



Training Equipment DI Module 6

Slot 6: 6ES7131-6BF00-0CA0 (DI 8 x 24V DC)

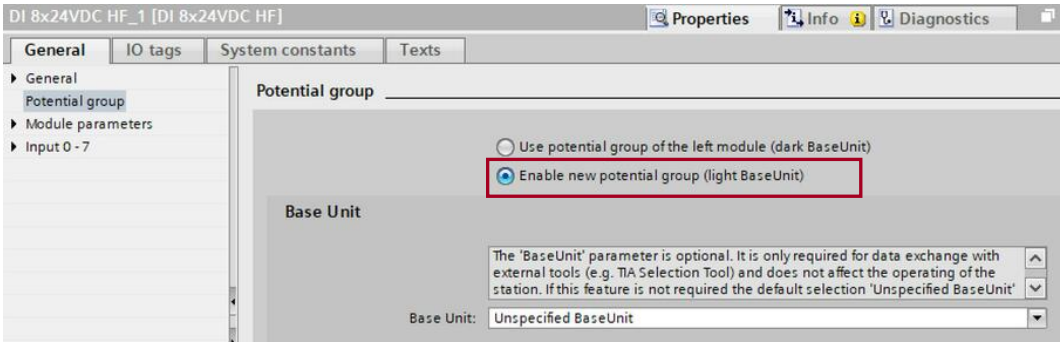
		6ES7131-6BF00-0CA0 (DI 8 x 24V DC) - New potential group.
Slot 6	Input	Function
I100.0	0	Unlock door
I100.1	1	DI1 switch
I100.2	2	DI2 switch
I100.3	3	DI3 switch
I100.4	4	Not connected.
I100.5	5	Not connected.
I100.6	6	K1/K2 feedback
I100.7	7	Not connected

Notes:

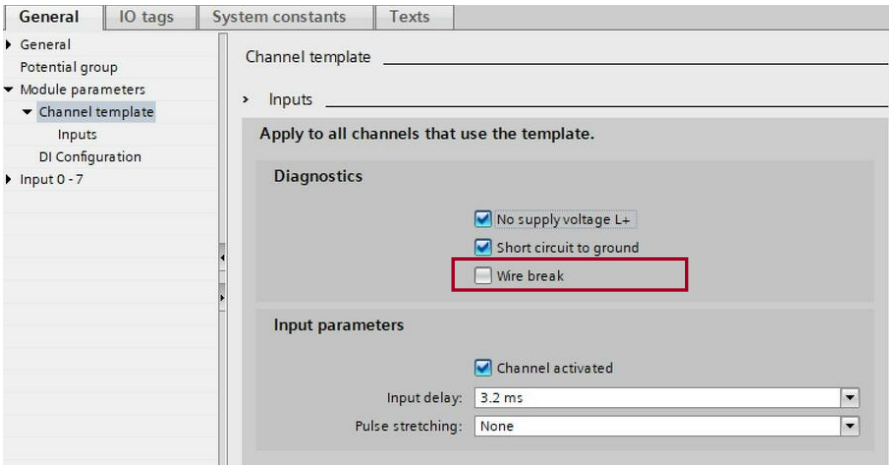
Training Equipment Features of DI Module 6:

- The module is connected to 4 single channel switches (DI 0, DI 1, DI 2, DI 3), K1 & K2 relays checkback input (DI 6)
- A single supply is used for all the inputs (1 - wire connection)

Potential Group: The unit has a light base which means that a new potential group has to be enabled.

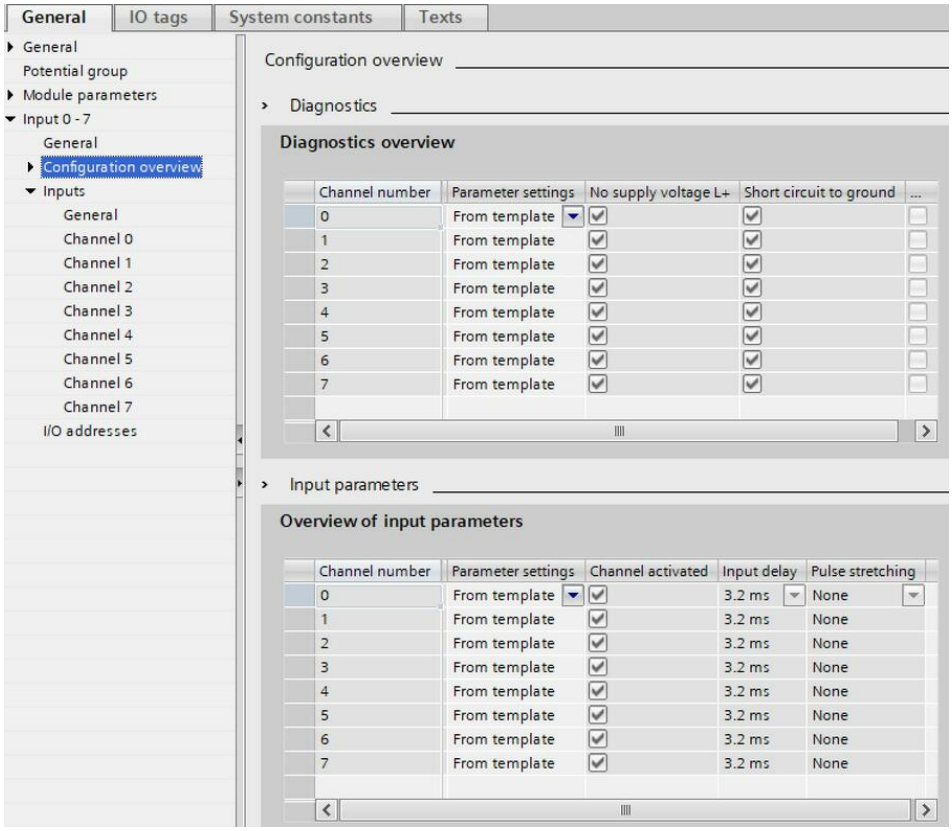


Channel Template: The ***Wire break*** option has to be deselected because of the 1 - wire connection type. Changing the channel template applies the setting to all of the inputs.

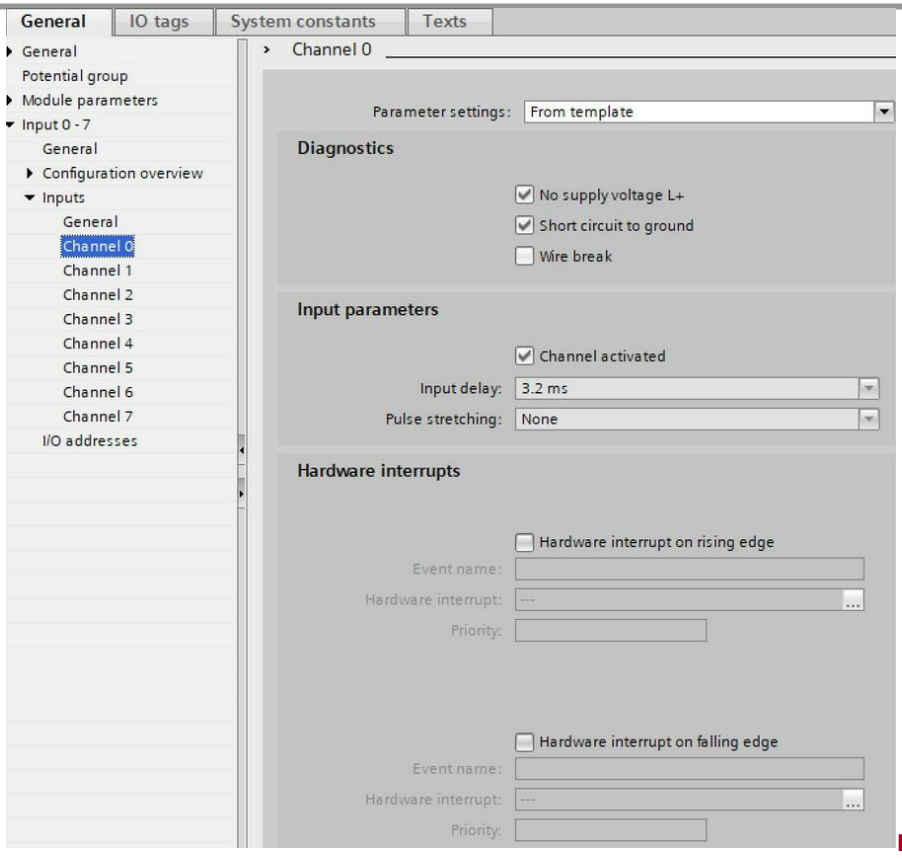


Notes:

Configuration Overview:
An overview of the settings can be seen.

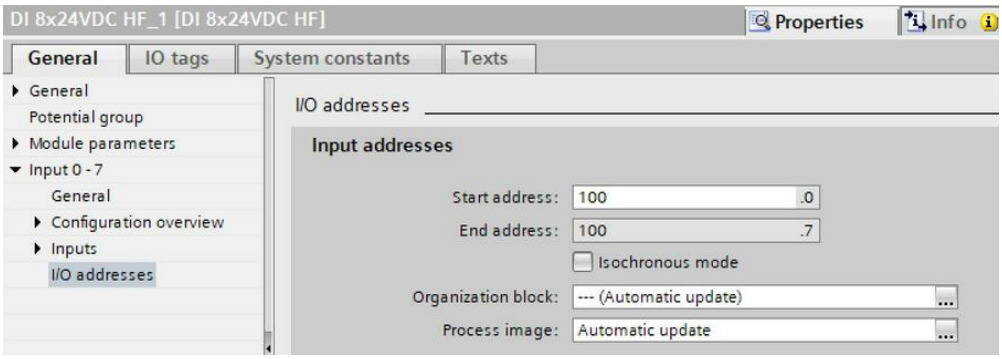


Channel settings:
Each channel can be individually configured if necessary.



Notes:

I/O addresses:
The previously set I/O addresses can be viewed and changed.



Training Equipment F-PM-E Module 7

Slot 7: 6ES7136-6PA00-0BC0 (F-PM-E 24V DC)

Slot 7		6ES7136-6PA00-0BC0 (F-PM-E 24V DC) - New potential group.
F-PLC	Channels – inputs	Function
	0	Not connected
	1	Not connected
	Channel - output	
Q540.0	0	Q PowerModule

Training Equipment Features of F-PM-E Module 7:

Safety-related shutdown of standard DQ modules by the F-PM-E 24VDC/8A PPM ST:
With this cost-effective solution, when a fault is detected in the process or at the F-PM-E 24VDC/8A PPM ST power module, a comprehensive and simultaneous shutdown of all affected outputs of the standard DQ modules takes place.
For the safety-related shutdown of standard DQ modules, you can achieve **SIL2/Cat.3/PLd.**
You can use the F-PM-E 24VDC/8A PPM ST power module with all standard DQ modules **within a potential group**.
Note: When commissioning, make sure that you only use the power module with the Base Unit type C0.

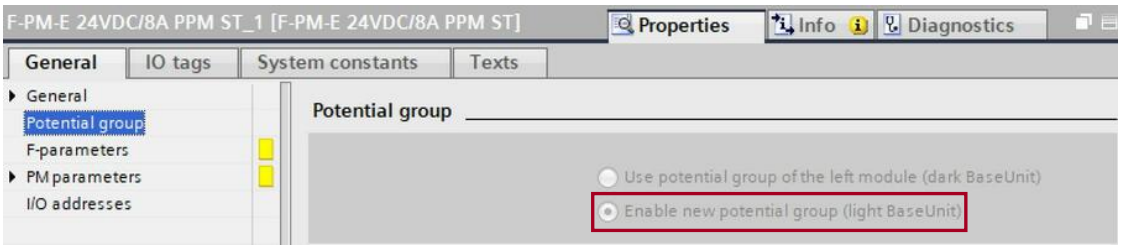
Notes:

The module offers:

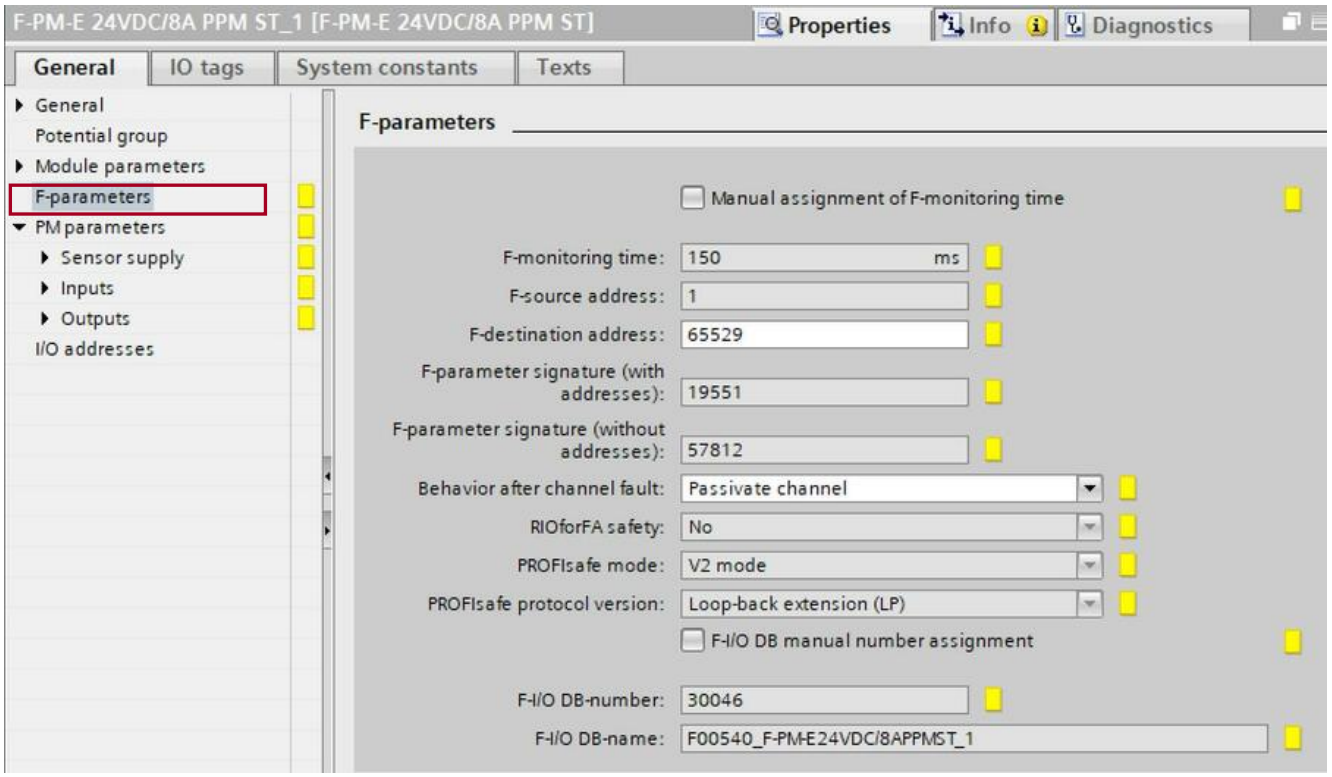
- **2 fail-safe inputs** (SIL3 /PLe) DI0 and DI1, which can be combined to form one two-channel input (2v2 evaluation).
- **1 output** PM or PP switching, output current 8 A (SIL3 /PLe)

Potential group:
The PM module requires a special (20 mm) Base Unit, which opens a new potential group (light Base Unit).

Potential Group: The unit has a light base which means that a new potential group has to be enabled.

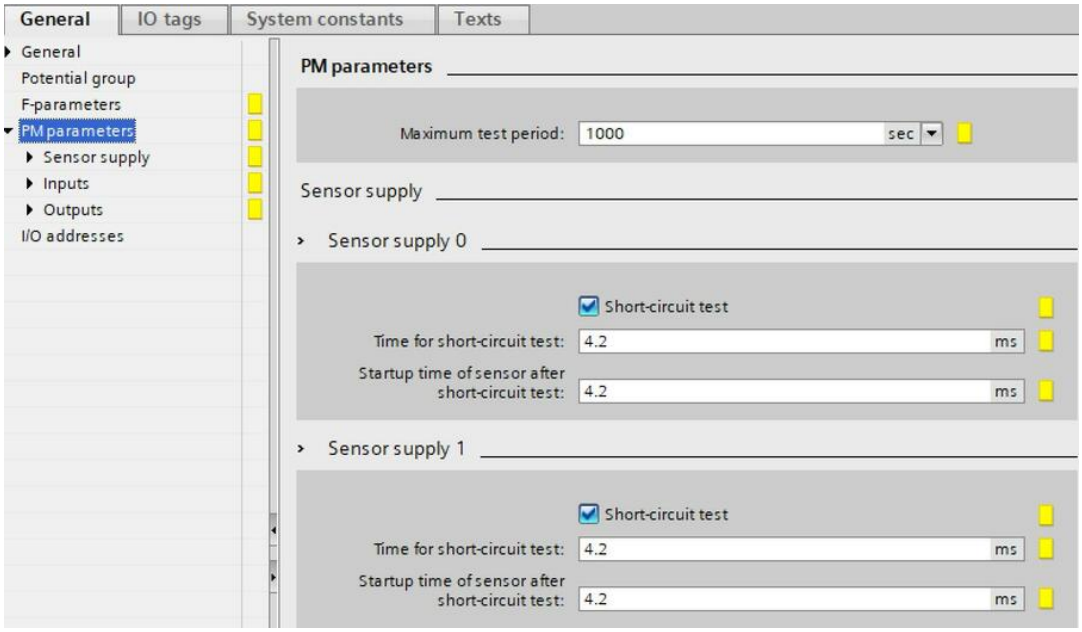


F-parameters - the default settings are used.

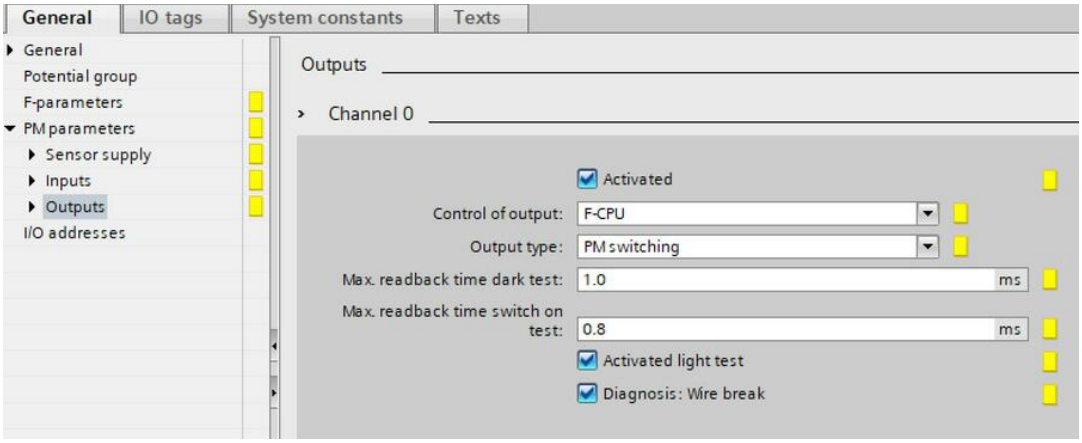


Notes:

PM-parameters - the default settings are used.



PM Outputs.



The fail-safe two-channel digital output DQ-P0 / DQ-M0 switches the voltage L+ and M via two electronic switches.
The switched voltage and ground are channeled to the internal busbars P1 and P2.
Additionally, the switched voltage and ground are also available at the Base Unit at DQ-P0 and DQ-M0.

This results in two connection options, which also can be used simultaneously:

- Load directly on the Base Unit DQ-P0 and DQ-M0
- Standard modules supplied via the internal potential rails P1 and P2, which can be switched off in a safety-related manner. In turn, loads can be connected to the standard modules.

Notes:

Note:
With cross-circuit between L+ and DQ, the activated actuator is no longer disabled. To prevent cross-circuits between L+ and DQ, the lines for connection of the actuators must be arranged in such a way that they are protected against cross-circuit, for example independently sheathed or in separate cable ducts. The ground cable to the Base Unit for the F-PM-E must be laid in a double arrangement as a safety measure. Otherwise busbar P2 will no longer be disabled safety-oriented with interruption of a single ground cable.

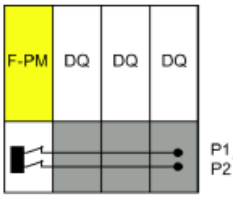
Control of the output:
(load to DQ-P0 — DQ-PM and standard module to P1 — P2)

Control can be performed as follows:

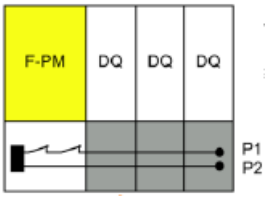
- In the F-program of the ***F-CPU*** (F-PLC) via the process value for DQ0
- In the F-program of the ***F-CPU*** (F-PLC) via the process value for DQ0 **AND** via **onboard F-DI** (&-logic operation: Process value for DQ0 & Evaluation of the module inputs DI0 / DI1) for quick group shutdown

Output type:

Type **PM:**



Type **PP:**



Training Equipment DQ Module 8

Slot 8: 6ES7132-6BF00-0CA0 (DQ 8X24V DC)

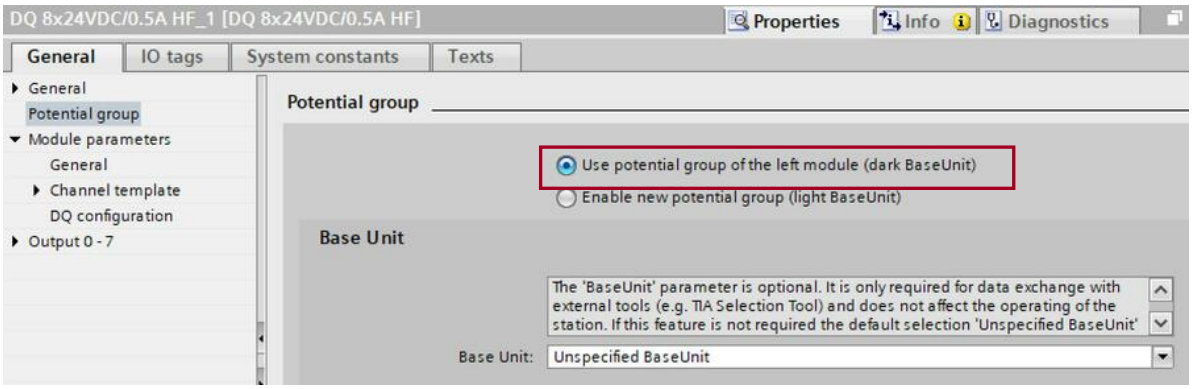
Slot 8		6ES7132-6BF00-0CA0 (DQ 8X24V DC 24V DC)
Output bit	Output	Function
Q100.0	DQ 0	DQ 0 LED
Q100.1	DQ 1	DQ 1 LED
Q100.2	DQ 2	DQ 2 LED
Q100.3	DQ 3	DQ 3 LED
Q100.4	DQ 4	DQ 4 LED
Q100.5	DQ 5	DQ 5 LED
Q100.6	DQ 6	DQ 6 LED
Q100.7	DQ 7	Door Unlock

Notes:

Training Equipment Features of DQ Module 8:

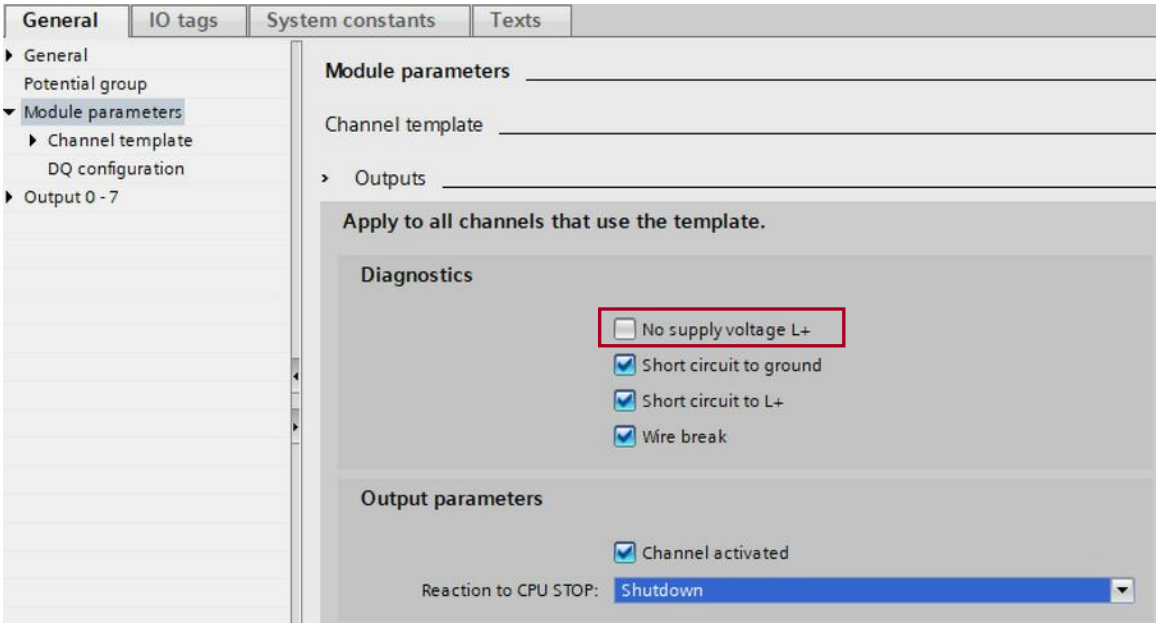
- The module is powered from the **F-PM-E Module** in slot 7 - see previous pages. This allows standard output modules to achieve **SIL2/Cat.3/PLd**
- A single supply is used for all the inputs (1 - wire connection)

Potential Group: The unit has a dark base and uses the potential group from the left module.



Channel template:

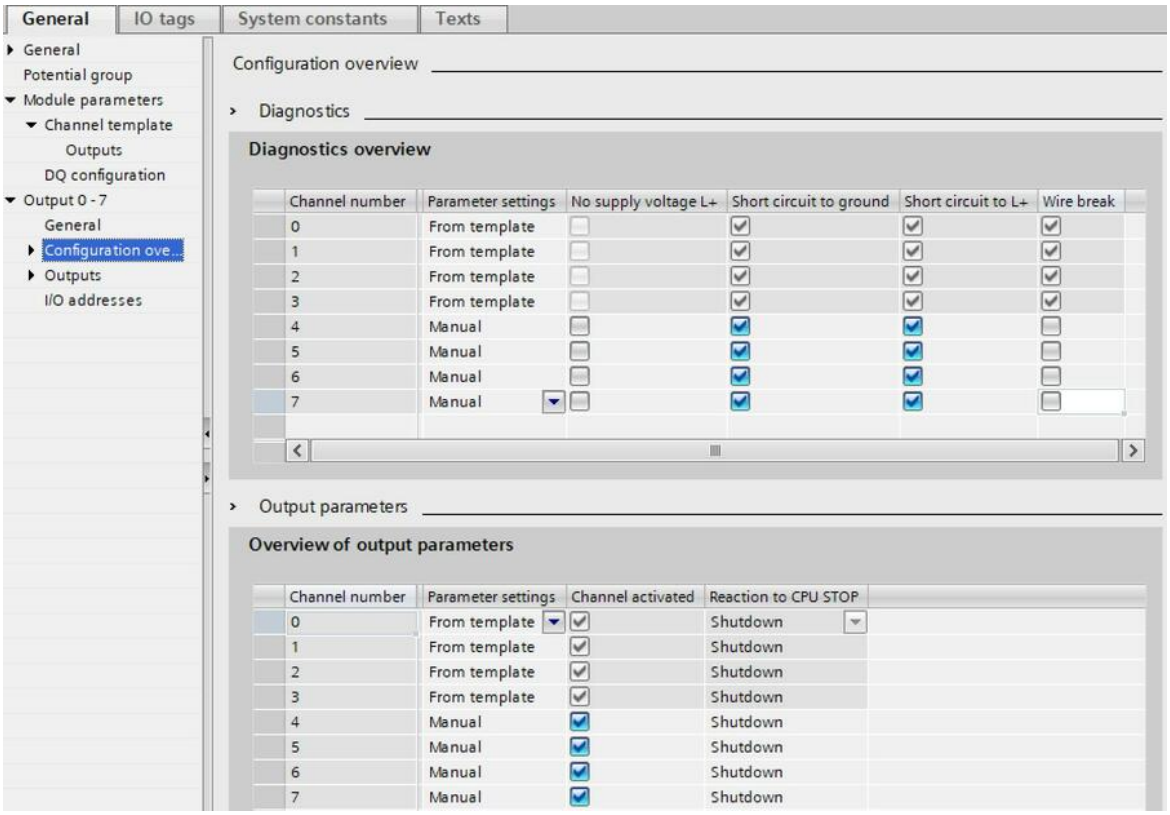
In the channel template, **No supply voltage L +** is **deactivated** due to the safety-related shutdown of the standard DQ module by means of the F-PM-E module, otherwise the module will fail if the F-PM-E output is not activated.



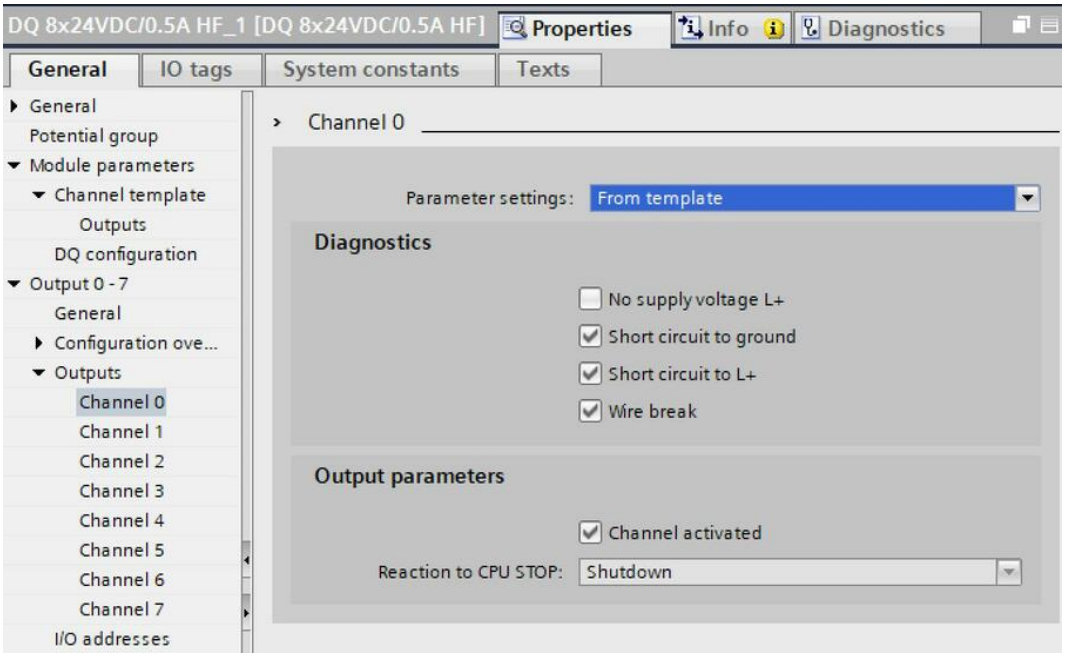
Notes:

Configuration overview and channel outputs:

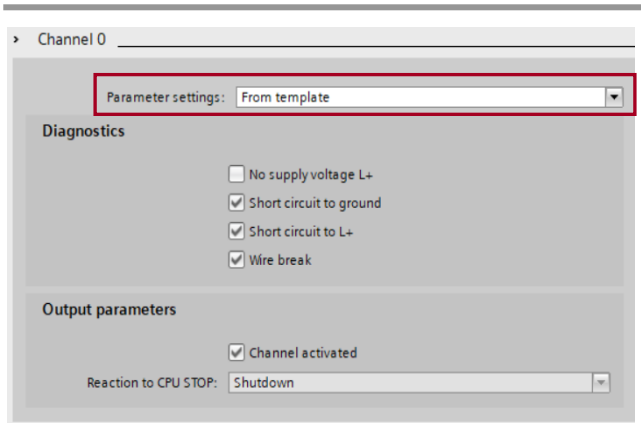
The previously described channel default values can be adopted in the following dialogs, or the respective channel parameters can be adapted manually.

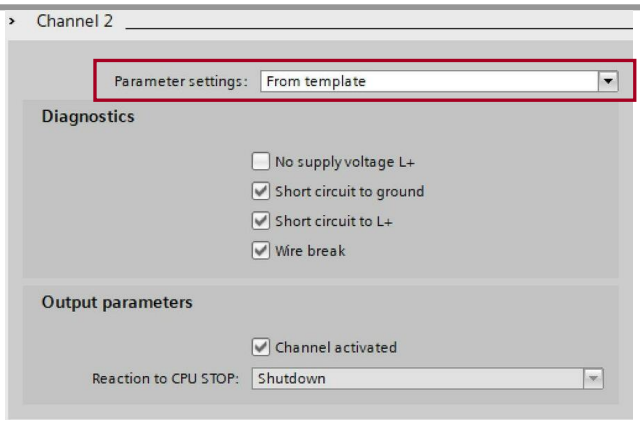


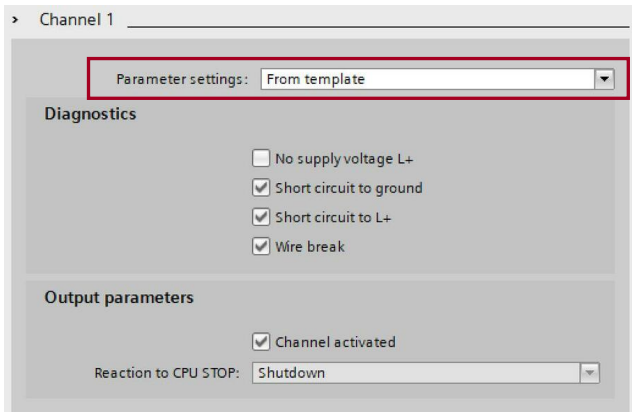
Individual Channel settings:

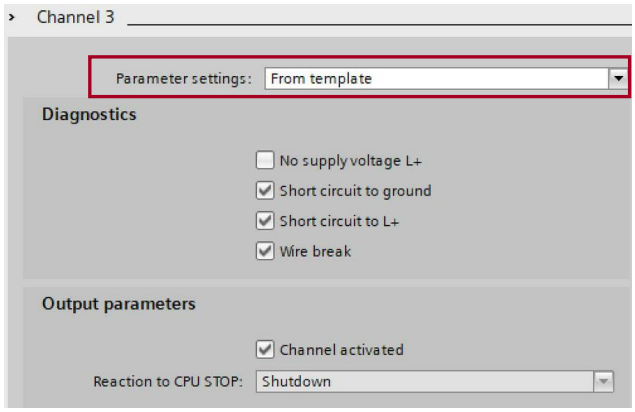


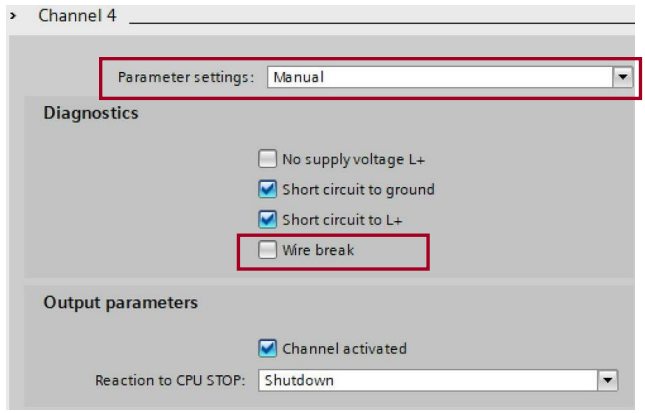
Notes:

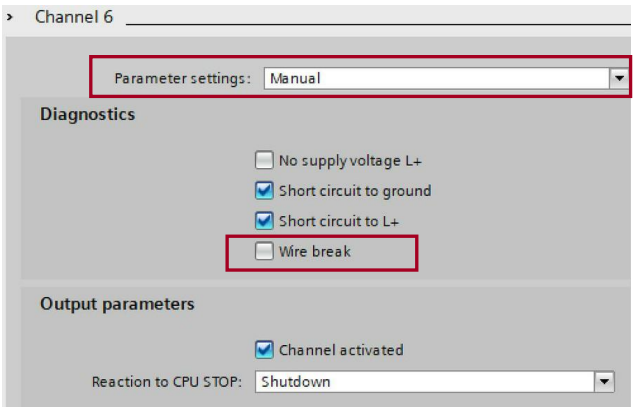

Channel 0
Parameter settings: From template
Diagnostics
No supply voltage L+
Short circuit to ground
Short circuit to L+
Wire break
Output parameters
Channel activated
Reaction to CPU STOP: Shutdown

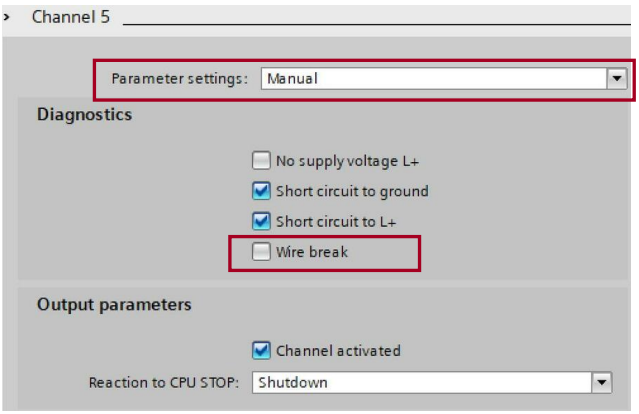

Channel 2
Parameter settings: From template
Diagnostics
No supply voltage L+
Short circuit to ground
Short circuit to L+
Wire break
Output parameters
Channel activated
Reaction to CPU STOP: Shutdown

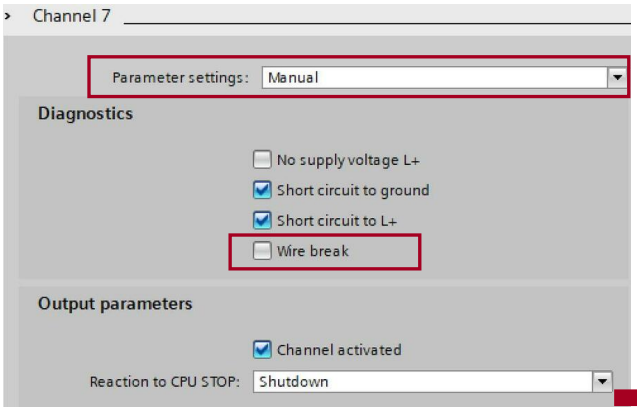

Channel 1
Parameter settings: From template
Diagnostics
No supply voltage L+
Short circuit to ground
Short circuit to L+
Wire break
Output parameters
Channel activated
Reaction to CPU STOP: Shutdown


Channel 3
Parameter settings: From template
Diagnostics
No supply voltage L+
Short circuit to ground
Short circuit to L+
Wire break
Output parameters
Channel activated
Reaction to CPU STOP: Shutdown


Channel 4
Parameter settings: Manual
Diagnostics
No supply voltage L+
Short circuit to ground
Short circuit to L+
Wire break
Output parameters
Channel activated
Reaction to CPU STOP: Shutdown


Channel 6
Parameter settings: Manual
Diagnostics
No supply voltage L+
Short circuit to ground
Short circuit to L+
Wire break
Output parameters
Channel activated
Reaction to CPU STOP: Shutdown


Channel 5
Parameter settings: Manual
Diagnostics
No supply voltage L+
Short circuit to ground
Short circuit to L+
Wire break
Output parameters
Channel activated
Reaction to CPU STOP: Shutdown


Channel 7
Parameter settings: Manual
Diagnostics
No supply voltage L+
Short circuit to ground
Short circuit to L+
Wire break
Output parameters
Channel activated
Reaction to CPU STOP: Shutdown

Notes:

A135: END

Notes:

A207

SINUMERIK ONE

SINUMERIK ONE Safety Integrated F_PLC + dbSI Configuration

Module Description:

This Module describes the step by step procedures to start-up and commission Sinumerik Safety F_PLC for the Sinumerik ONE.

The procedures described within this module predominantly use TIA Portal & Sinumerik Operate HMI.

The commissioning will start with the Virtual Sinumerik and then progress to the Real Sinumerik.

This Module is valid for >= S/W 06.23.xx.xx

It is expected that the student has a good understanding of the Sinumerik ONE control, including PLC programming, NC & Drive Commissioning prior to using this module.

The commissioning of the Drive Based Safety Integrated Basic/Extended Functions:
STO, SS1, SS2, SOS, SAM, SSM, SLS, SBC, SBT and F-PLC programming via TIA Portal V19 are implemented and, using the respective diagnostics options of HMI-Operate, tested and diagnosed.

Module Description:

After completing the module, you are capable of the following:

- Configuration of the SINUMERIK with F-PLC and the ET200SP with F-I/O
- Commissioning of the Drive Based Safety Integrated Function
- F-PLC program creation using TIA Portal V19
- Diagnostics

Content:

- Introduction
- System Requirements and Information
- Overview: Safety Integrated Function
- OEM Function Table
- Commissioning Virtual to Real
- Safety Integrated Drive Configuration
- F-PLC Logic
- Safe Torque Off - Extended
- Safe Brake Control
- Safe Stop 1 - Extended
- Safe Stop 2 - Safe Operating Stop
- Safe Acceleration Monitoring/Safe Braking Ramp
- Safely Limited Speeds
- Safe Speed Monitoring
- Safe Brake Test
- Test Stop Routine

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Introduction
System requirements and Information
Overview: Safety Integrated Functions
OEM Function Table
Commissionin g Virtual to Real
Safety Integrated Drive Configuration
F-PLC Logic
Safe Torque OFF - Extended
Safe Brake Management
Safe Stop 1 Extended
Safe Stop 2/ Safe Operating Stop
Safe Acceleration Monitoring/ Safe Braking Ramp
Safe limited Speeds
Safe Speed Monitoring
Safe Brake Test
Safe Stop Routine
End of the Module

Introduction:

SINUMERIK Safety Integrated is a comprehensive safety package that helps to protect both personnel as well as machines. This is extremely efficient and cost effective thanks to the safety functions that are completely integrated into the control and drive technology.

The Safety Integrated monitoring limits are set to operate outside/above the NC monitoring limits - Safety Integrated alarms and messages will generally only be seen if the NC limits are exceeded.

The machine can be safely and practically operated under all of the specified operating conditions. For instance, in the setting-up and test modes with the protective door open.

The safety functions comply with the requirements laid down in DIN EN 61508 and DIN EN ISO 13849.
As a consequence, the essential requirements relating to functional safety can be simply and cost-effectively implemented.

The functional scope includes, e.g.:

- Functions to safely monitor speed and standstill.
- Functions to safely differentiate between working and protective zones and to identify the relevant areas (e.g. Safe Cams).
- All safety relevant signals are directly connected and internally logically combined.

Product features

- Functions to protect persons and machines.
- Short response times via cyclic interrupt programming.
- Can be used in all operating modes.
- Replace external electromechanical devices.

Customer benefits

- High degree of safety: Seamless implementation of safety functions in SIL (Safe Integrity Level - EN62061) / PL (Performance Level - EN13849).
- High degree of flexibility: Practical safety and operator concepts can be realized.
- High degree of cost effectiveness: Hardware and installation costs are reduced.
- Electromechanical switching elements that are prone to faults are eliminated due to .

Notes:

System Requirements & Information:



SINUMERIK ONE:

SINUMERIK Safety Integrated is offered for **NCU/PPU 1740/1750/1760** as of **SINUMERIK ONE NC SW V6.23**. The performance of the PLC must be taken into account depending upon on the application.

- NCU 1740 with PLC 1500F 6FC5317-4AA00-0AA0
- NCU 1750 with PLC 1500F 6FC5317-5AA00-0AA0
- NCU 1760 with PLC 1500F 6FC5317-6AA00-0AA0
- PPU 1740 with PLC 1500F 6FC5317-4AA00-1xA0 (x =C or D)

Typical PLC cycle times 1740/1750/1760 15...20ms/10...15ms/5...7ms

(Please refer to the NC63 catalogue for more information)

SINUMERIK software: (general release)

- NCU firmware as of V6.23
- HMI software SINUMERIK OPERATE as of V6.23

Engineering system software:

SINUMERIK Safety Integrated F_PLC + dbSI can only be configured with:

- TIA Portal as of V19 Update 1 (SIMATIC STEP 7 Professional)
- SINUMERIK STEP 7 Toolbox as of V19

Option Description	Option Number	Course Requirement
Safety Integrated /SI Axis/Spindle • Safety Integrated Axis / Spindle Safe motion monitoring function 1 axis / spindle. Prerequisite: Safety Integrated F -PLC	**6FC5800-0BK00-0YB0**	**5**
Safety Integrated /SI Logic • Safety Integrated F-PLC Enables the F-PLC integrated in the SINUMERIK ONE for failsafe programming, contains no axis	**6FC5800-0BS60-0YB0**	✓

Notes:

Overview: Safety Integrated Functions

Safe Functions		Abbreviation	Brief description
Basic Functions	Safe Torque Off	STO	Safe torque off
	Safe Stop 1	SS1	Safe stopping process in accordance with stop category 1
	Safe Brake Control	SBC	Safe brake control
Extended Functions (includes basic functions)	Safe Stop 2	SS2	Safe stopping in accordance with stop category 2 (Not on machine tool equipment)
	Safe Operating Stop	SOS	Safe monitoring of the standstill position
	Safely Limited Speed	SLS	Safe monitoring of the maximum speed
	Safe Speed Monitor	SSM	Safe monitoring of the minimum speed
	Safe Acceleration Monitor	SAM	Safe monitoring of drive acceleration
	Safe Direction	SDI	Safe monitoring of the direction of motion
	Safe Limit Positions	SLP	Safe software limit switches.
	Safe Brake Test	SBT	Safe brake test
	Safe Cams	SCA	Safe software cams

The various safety functions are triggered by the system in response to a safety limit being exceeded or in response to a request from the user safety logic.

STOP types:

If a safety function is triggered, a STOP is issued. The STOP type depends on the safety function that was triggered.
It is possible, in some cases, to configure the STOP type.

Stop type	Action	Effect	Initiated in response to	Active Safety Function	Alarm clearance
STOP A	Pulses are immediately cancelled	Drive coasts down	SAM/SLS	STO	Power off/on
STOP B	Setpoint zero is immediately set and timer t_B started. When $t_B = 0$ or $n_{act} < n_{shutdown}$, STOP A is activated.	The drive brakes along the OFF3 ramp - transition into STOP A	SS1/SOS/ SLS	STO	Power off/on
STOP C	Setpoint zero is immediately set and timer t_C is started. When $t_C = 0$, SOS is activated.	The drive brakes along the OFF3 ramp, SOS active.	SLS/SLP	SOS	Reset
STOP D	Motor brakes along the acceleration path and timer t_D is started. When $t_D = 0$, SOS is activated.	Drive brakes as part of a group along the accelera-tion path. SOS active	SLS/SLP	SOS	Reset
STOP D Dynamic	* NCK changes the dynamics * so that the axes can be braked interpo-lating within the transitional time STOP D -> SOS	Drive brakes as part of a group along the accelera-tion path. SOS active	SS2E/SLS/ SLP	SOS	Reset
STOP E	Results in stopping and retraction, plus timer t_E is started. When $t_E = 0$, SOS is activated.	Drive brakes along the programmed retraction and stopping paths. (ESR) SOS is active	SLS/SLP	SOS	Reset

Notes:

OEM Function Table

The function table specifies which safety function should be active for certain machine states.
A function table should be created before starting the configuration.

The risk assessment of a particular machine/machine series is used determine the function table and associated limits.

The function table is included later in the acceptance report, which is when the functionality of the last two columns are tested.

Example function table:

Operating mode	Drive	Which status should be active?
Doors closed	All	**SLS4**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are energised.
Doors open and mode switch at **Auto**. (***Production*** on test simulator)	X, Y, Z, A axes	**SOS**
	Spindle	**STO**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are not energised.
Doors open, mode switch at **Auto** (***Production*** on test simulator) and ***Hold to run*** switch pressed.	X, Y, Z, A axes	**SOS**
	Spindle	**STO**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are not energised.
Doors open and mode switch at ***Set-up***	X, Y, Z, A Axes	**SOS**
	Spindle	**STO**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are not energised.
Doors open, mode switch at ***Set-up*** and ***Hold to run*** switch pressed.	X, Y, Z, A axes	**SLS1**
	Spindle	**STO**
	Axes/equipment not NC controlled.	Contactors: e.g. Turrets, conveyors etc. are not energised.

The following table shows example SLS speeds. The SLS speeds are defined by the OEM and will vary between machine types.

	Set-up/door open and hold to run		Auto (***Production*** on test simulator) / door closed	
	Maximum Feed-rate	**SLS1**	Maximum Feed-rate	**SLS4**
X, Y, Z (mm/min)	2000	**2500**	10000	**12500**
A (rpm)	10	**12.5**	100	**125**
Spindle (rpm)			1000	**1250**

Feedrate/SLS Relationship

Setpoint Limitation is used to limit the feed-rate of an axis to a percentage of the active Safe Limited Speed. Recommended value is 80%, therefore maximum federate is 80% of SLS speed

Selection of required extended functions:

It is clear, from the function table, which safety extended functions are used.
These are **SOS, SLS1** and **SLS4** (for axes X, Y , Z and A) , and **STO, SLS4** (for the spindle)
In addition, **SS1** is used for emergency stop on all axes/spindle.

Notes:

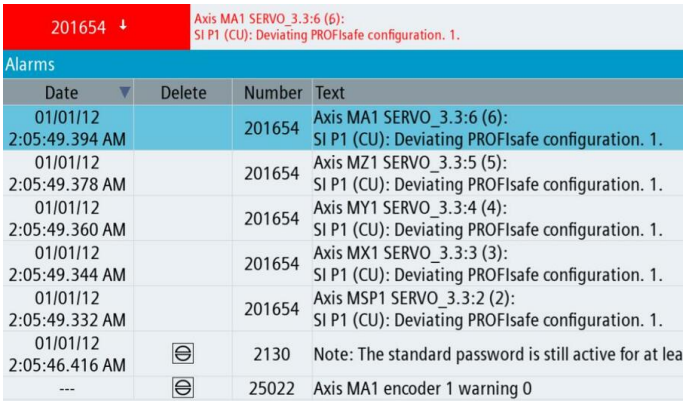
Commissioning Virtual to Real

At this stage the procedure in the module A068 Commissioning Virtual to Real should have been carried out

- Preparing the controller
- Read in Data Virtual to Real
- Check options/licenses
- Drives Firmware Update/Configuration/Factory Settings
- DMS Encoder Configuration

Expected alarms:

Expected alarms are seen below



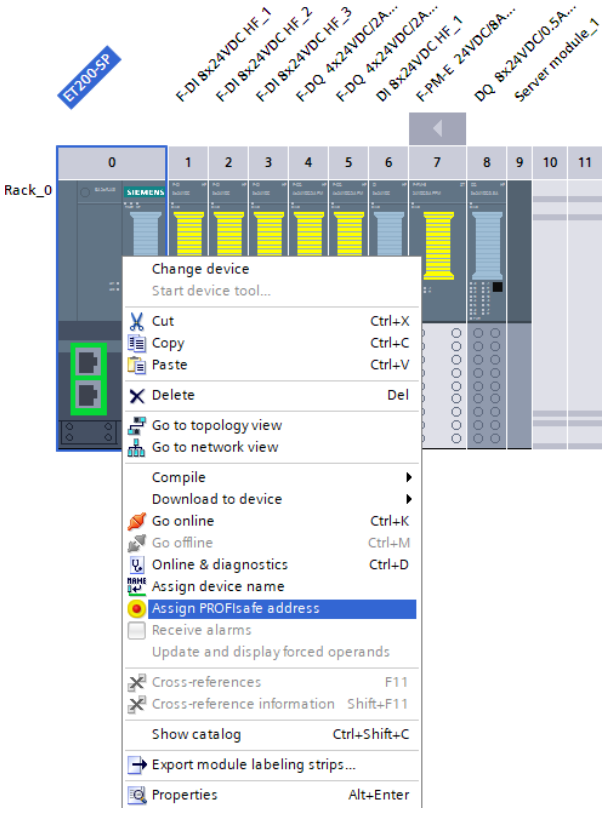
Assigning Profisafe Addresses.

The Profisafe addresses can now be assigned.

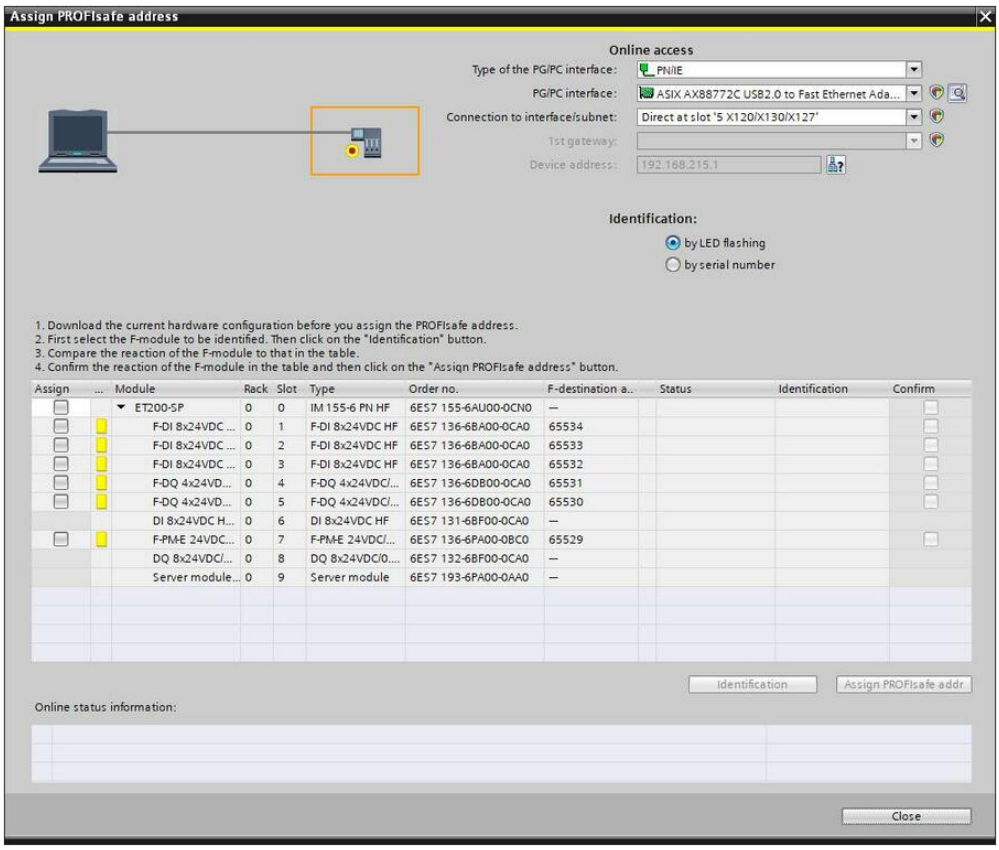
Note: The network cable needs to be connected to X127.



In Device View highlight the ET200-SP IM Module and then right click with the mouse

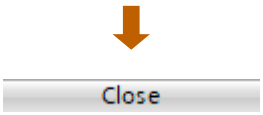


Notes:



Tick the ***Assign*** boxes for the modules and Select the ***Identification*** button





Activate the ***Reset (po)*** procedure

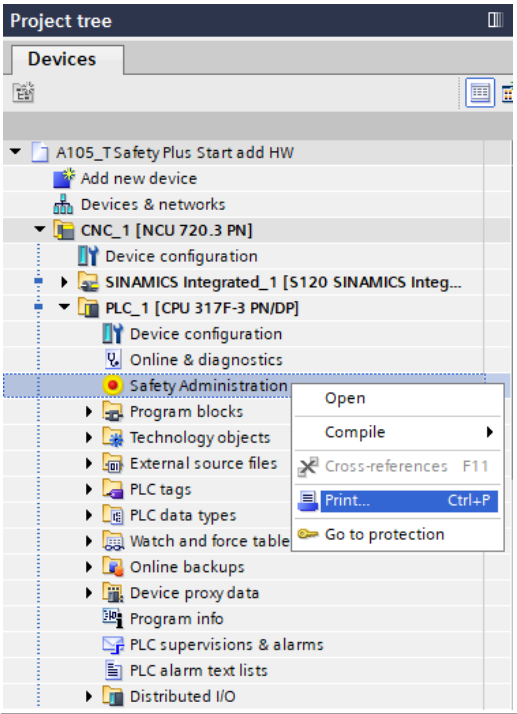


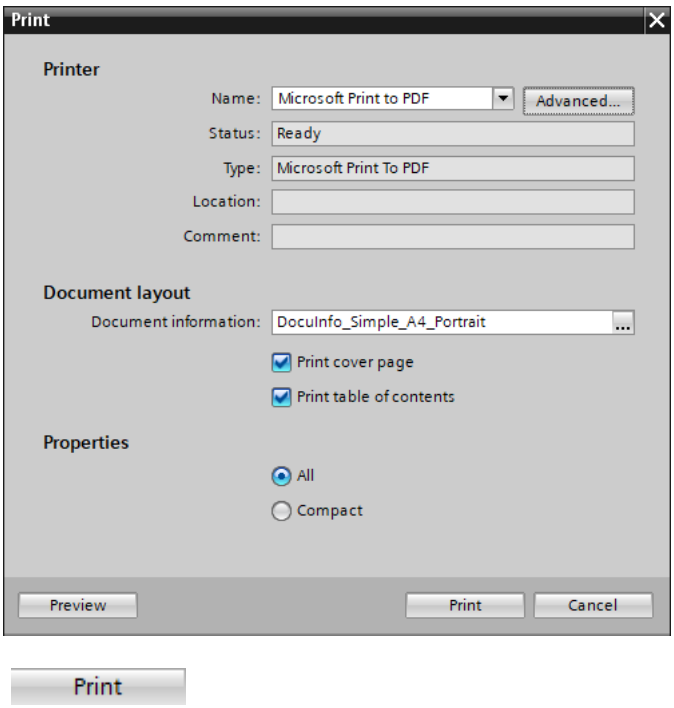
NOTE:
Under "Identification" select how you wish to identify the F-modules:
– "By flashing the LEDs"
This is the default setting. During identification, the STATUS LEDs of the F-modules to be identified flash.
– "Using the serial number"
If you cannot see the F-modules clearly, you can identify them by the serial number of the interface module.
In column "Assign" select all F-modules where you wish to assign an F-destination address.
Click the "Identification" button.
– If you have selected LED flashing for the identification type, then check whether the status LEDs of that F-module flash green whose F-destination address you wish to assign.
– If you have selected the serial number for the identification type, compare the displayed serial number with the serial number of the module.
– Confirm successfully identified F-modules in the "Confirm" column of the table.
– Using button "Assign F-destination address" assign the configured F-destination addresses to the F-modules, and confirm dialog "Acknowledge assignment" (within 60 seconds).

Notes:

Print out Safety Information:

Detailed Safety Integrated relevant information of the PLC can be saved:





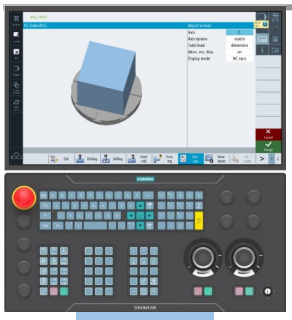
Safety information can be seen in the file:

SINAMICS Integrated_1					
Rail - Slot	Module	Start address	F-destination address	F-monitoring time	Parameter signature (w/o addresses)
0	DriveAxis01FIODB	6700	1	150 ms	0xEE5 (3813)
0	DriveAxis02FIODB	6716	2	150 ms	0xEE5 (3813)
0	DriveAxis03FIODB	6732	3	150 ms	0xEE5 (3813)
0	DriveAxis04FIODB	6748	4	150 ms	0xEE5 (3813)
0	DriveAxis05FIODB	6764	5	150 ms	0xEE5 (3813)
0	DriveAxis06FIODB	6780	6	150 ms	0xEE5 (3813)

ET200-SP					
Rail - Slot	Module	Start address	F-destination address	F-monitoring time	Parameter signature (w/o addresses)
1	6ES7 136-6BA00-0CA0 F-DI 8x24VDC HF_1	512	65534	150 ms	0x2B3C (11068)
2	6ES7 136-6BA00-0CA0 F-DI 8x24VDC HF_2	518	65533	150 ms	0x8AFE (35582)
3	6ES7 136-6BA00-0CA0 F-DI 8x24VDC HF_3	524	65532	150 ms	0xD61 (3425)
4	6ES7 136-6DB00-0CA0 F-DQ 4x24VDC/2A PM HF_1	530	65531	150 ms	0xDE3B (56891)
5	6ES7 136-6DB00-0CA0 F-DQ 4x24VDC/2A PM HF_2	535	65530	150 ms	0xDE3B (56891)
7	6ES7 136-6PA00-0BC0 F-PM-E 24VDC/8A PPM ST_1	540	65529	150 ms	0xE1D4 (57812)

The information should be part of the machine technical documentation and is required for the CE certification:

Notes:



X Y Z SP A

Setpoints/actual Values
Addr. 4xxx (136)

NC-SI
Addr.6xxx (903)
Addr.5xxx (701)

TIA Portal
PLC config

SP X/Y Z/A



Axis-drive assignment in the TIA Portal with safety telegrams

Adapting the physical sequence of the telegram configuration (telegram 136 and safety telegrams 903/701)

As a result of the technical situation in the control cabinet, the load with the highest power consumption (in the example, spindle MSP1) should be located directly next to the infeed, and is therefore assigned to the first Motor Module and is configured as Servo_3.3:3 (Drive Object 3). The I/O addresses of spindle MSP1 (drive axis 04) must therefore be located at the uppermost position in the telegram configuration.

For the telegram configuration the add-in Sinumerik-AxisDrivesAssignment.addin can be used. The add-in is part of the SINUMERIK STEP 7 Toolbox.

After changing, the addresses of the telegrams should correspond to the addresses in the following diagram. The **NC configuration** is left at the default setting. The addresses are adapted in the TIA Portal for the **PLC configuration**.

NC configuration

Machine data remain at the default setting

Axis	MD Index x	Machine Axis = x + 1	MD30110[x] CTRLOUT_ MODULE_NR	MD11350[x] DRIVE_LOGIC_ ADDRESS
MX1	0	1	1	4100
MY1	1	2	2	4140
MZ1	2	3	3	4180
MSP1	3	4	4	4220
MA1	4	5	5	4260

PLC configuration (TIA Portal)
I/O addresses - telegram configuration

Before → After

PROFI drive I/O .	SIC/ SCC	PROFI safe I/O
4100	5800	6700
4140	5816	6716
4180	5832	6732
4220	5848	6748
4260	5864	6764

PROFI drive I/O .	SIC/ SCC	PROFI safe I/O	SINAMICS Integrated
4220	5848	6748	Drive Axis 4
4100	5800	6700	Drive Axis 1
4140	5816	6716	Drive Axis 2
4180	5832	6732	Drive Axis 3
4260	5864	6764	Drive Axis 5

Drive configuration — Drive objects

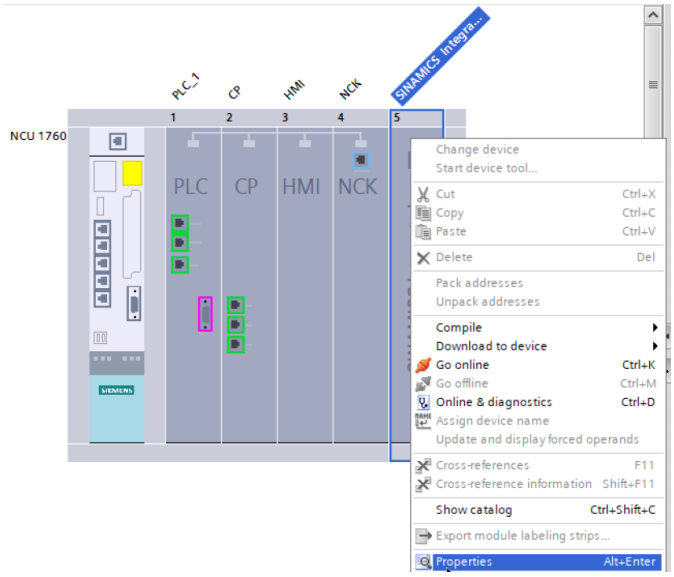
SINAMICS Integrated	Index x	p978[x] List of drive objects (DO)	(No. drive object) and name
Drive Axis 4	0	3	(3) Drive 1 MSP1
Drive Axis 1	1	4	(4) Drive 2 MX1
Drive Axis 2	2	5	(5) Drive 3 MY1
Drive Axis 3	3	6	(6) Drive 4 MZ1
Drive Axis 5	4	7	(7) Drive 5 MA1

Notes:

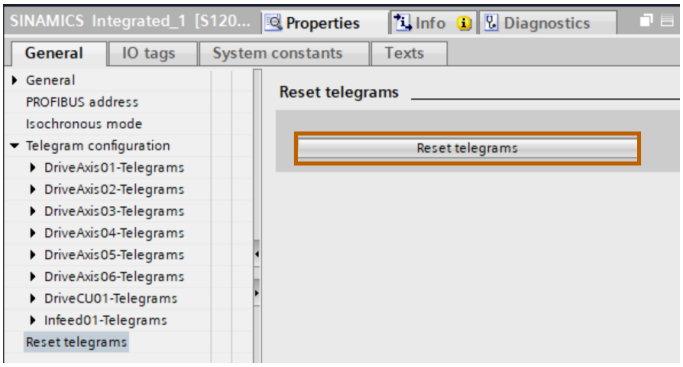
Telegrams: The drive telegrams have been assigned in the Hardware configuration. The telegram addresses now need to be adapted to suit the order of the addressing requirement shown previously.

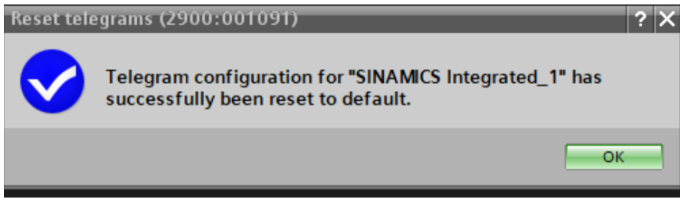
Device view

Right click on Sinamics Integrated



The telegram addresses can be reset as shown below. This is to establish a basic condition before carrying out the axis assignment.

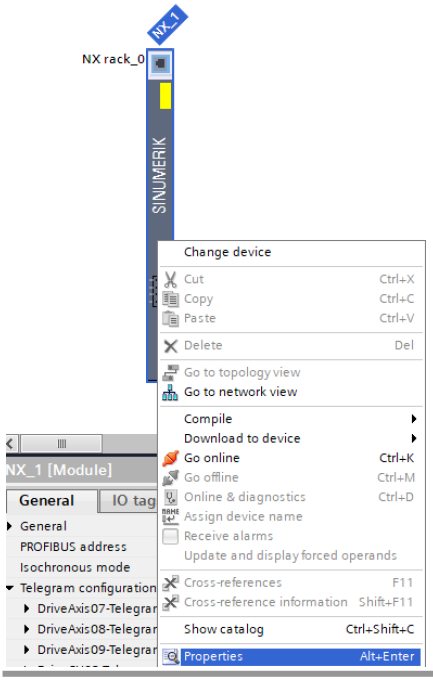




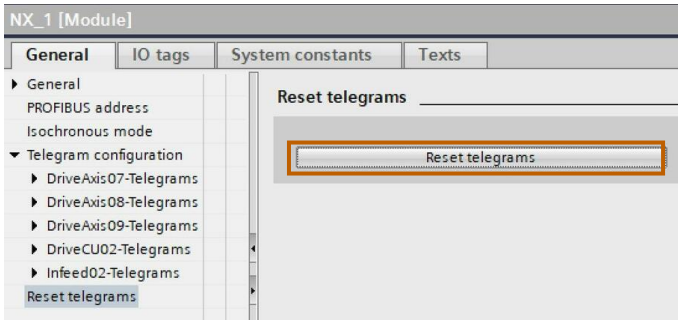
If an NX module is available the same procedure should be used.

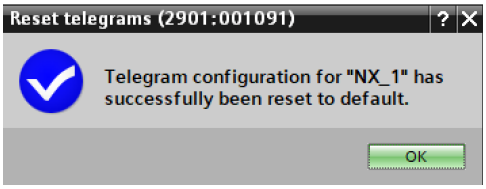
Device view

Right click on NX_1



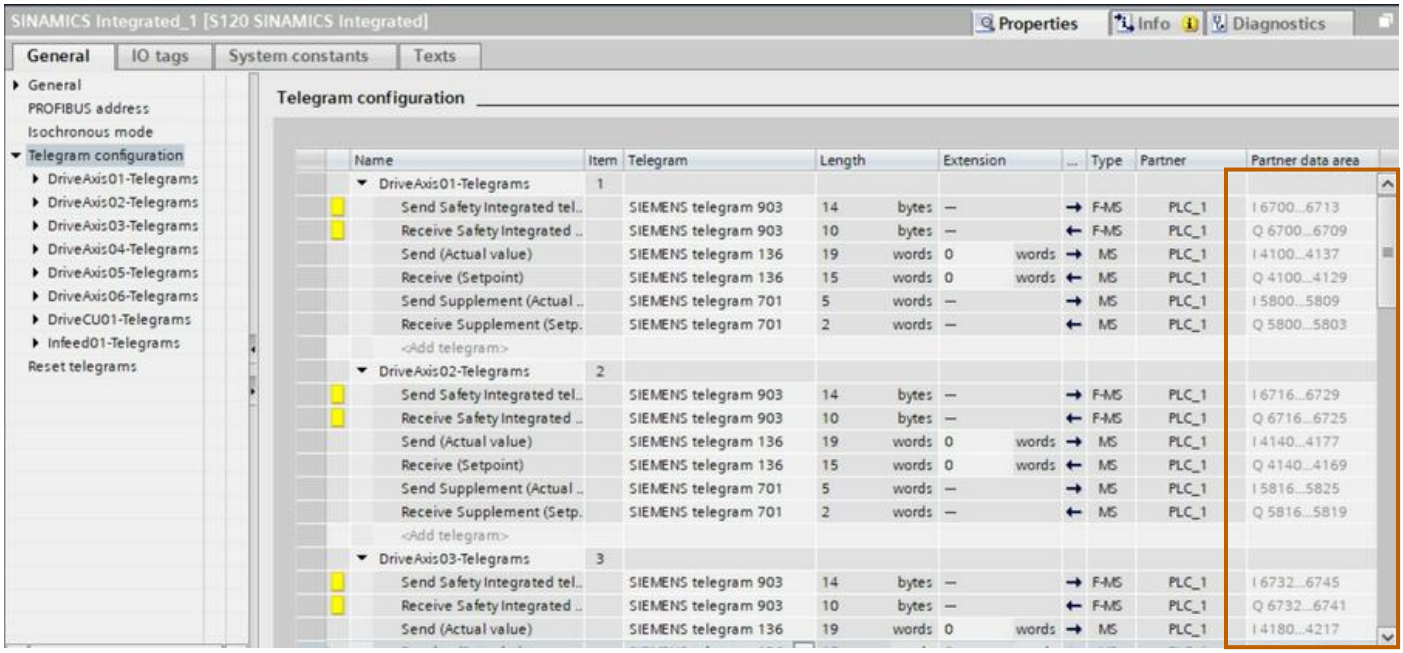
The telegram addresses on the NX if available can be reset as shown below. This is to establish a basic condition before carrying out any axis assignment on the NX.





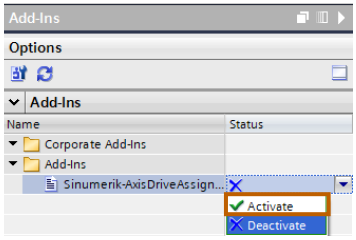
Notes:

Default setting of the safety telegram addresses after resetting the telegrams.

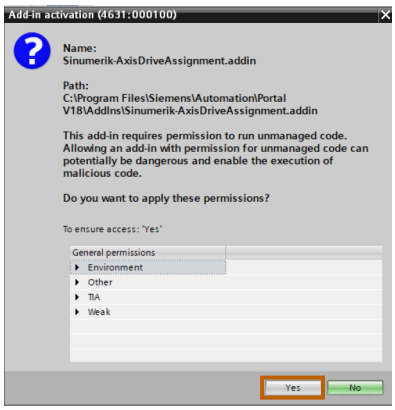


Manage SINAMICS telegrams using the “SINUMERIK-Axis drive Assignment add In

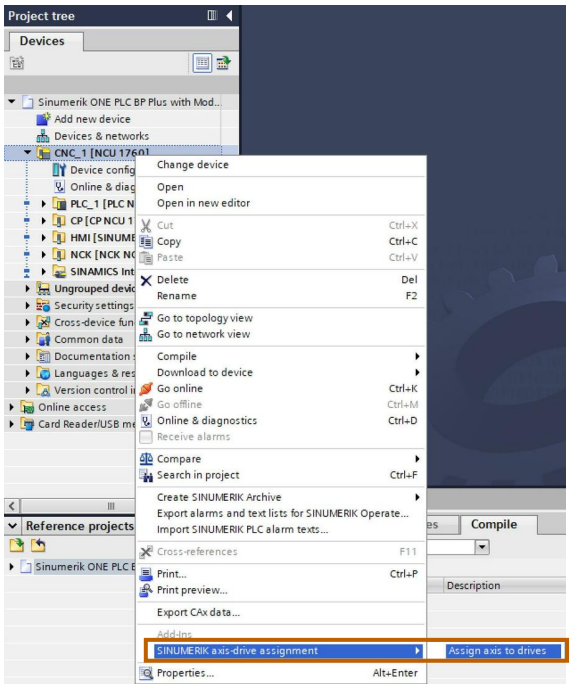
The Add-In “SINUMERIK axis-drive assignment” should be activated.



Continue with “Yes”

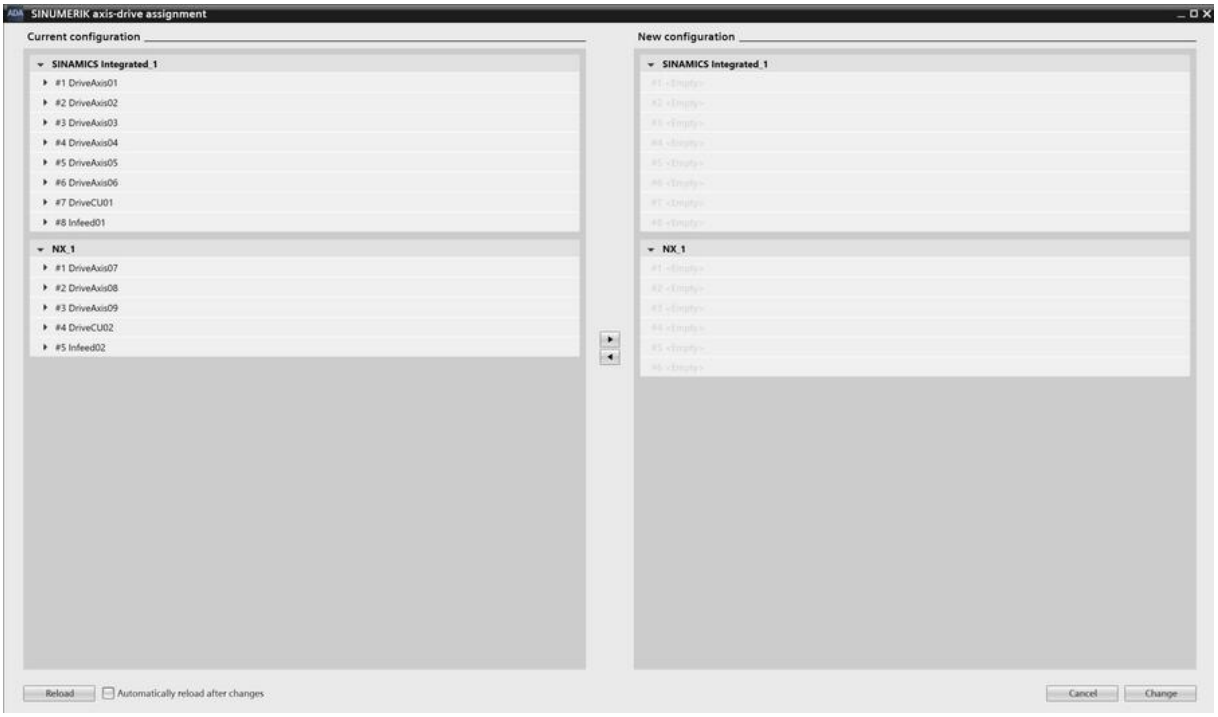


Right click on the NCU and select “SIMUMERIK axis-drive assignment > Assign axes to drives”

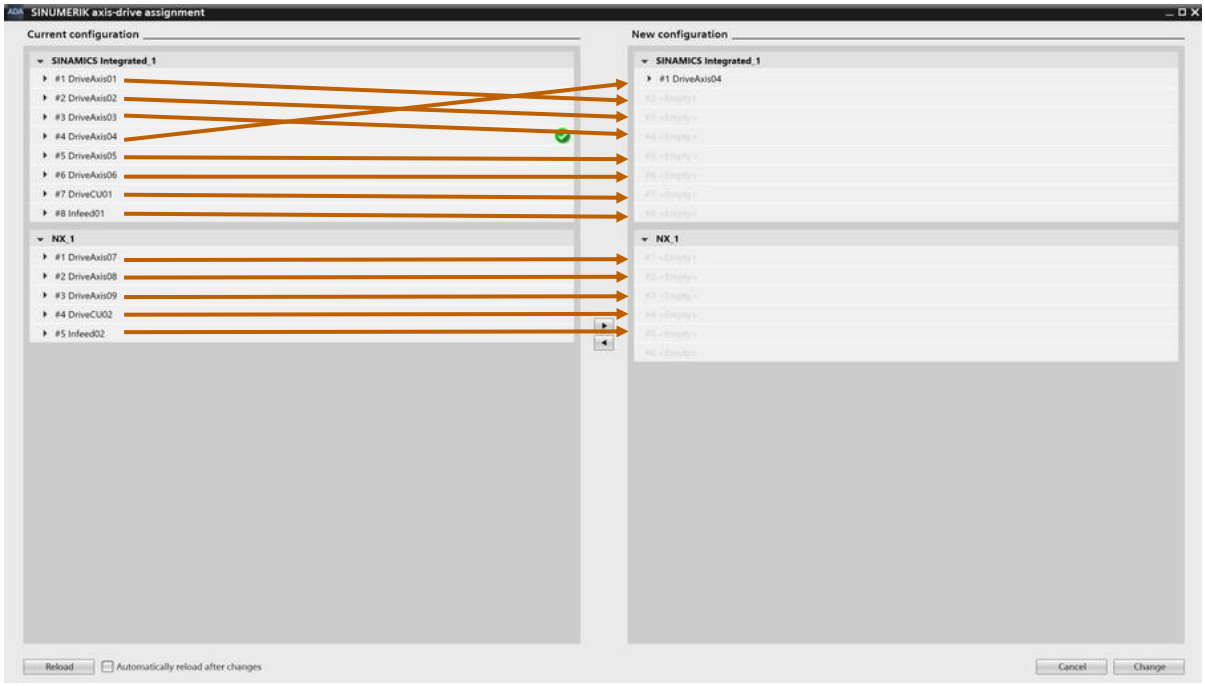


Notes:

The "Current configuration" area provides an overview of all configured telegrams sorted by devices.

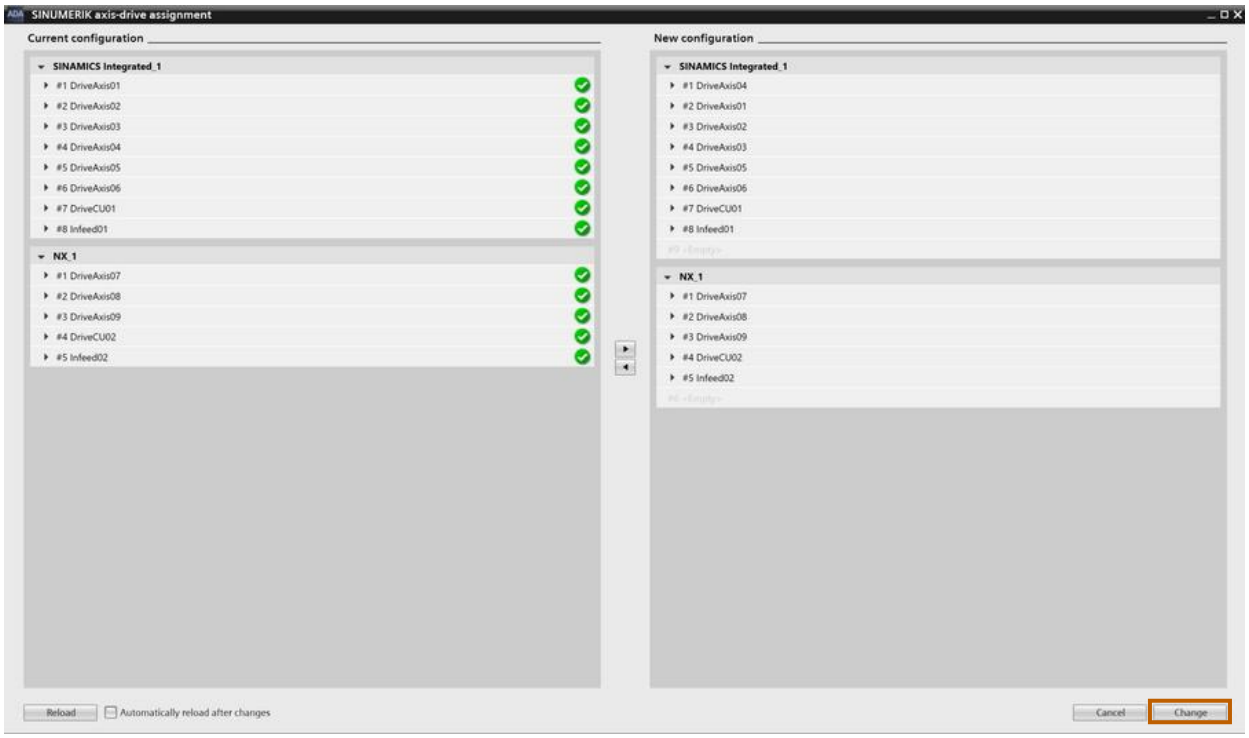


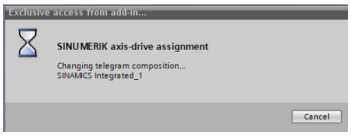
Changes to the configuration can be made using “Drag and Drop”. The devices **must be fully populated**, even if axes/infeeds are not used.

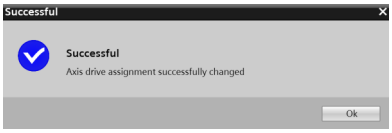


Notes:

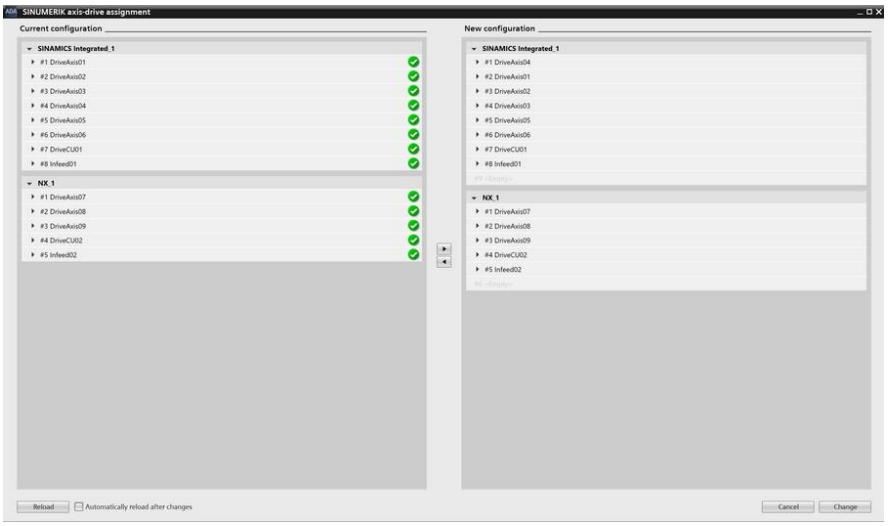
Once all Drive Axes, CUs and Infeeds have been transferred to the “New configuration” then confirm the changes with “Change”







The modified telegram configuration is transferred to the configuration in TIA Portal



After finally acknowledging that the change has been completed, the dialog "SINUMERIK axis-drive assignment" is automatically displayed again.

The content displayed depends on whether the checkmark "Automatically reload after change" was previously set or not (default setting).

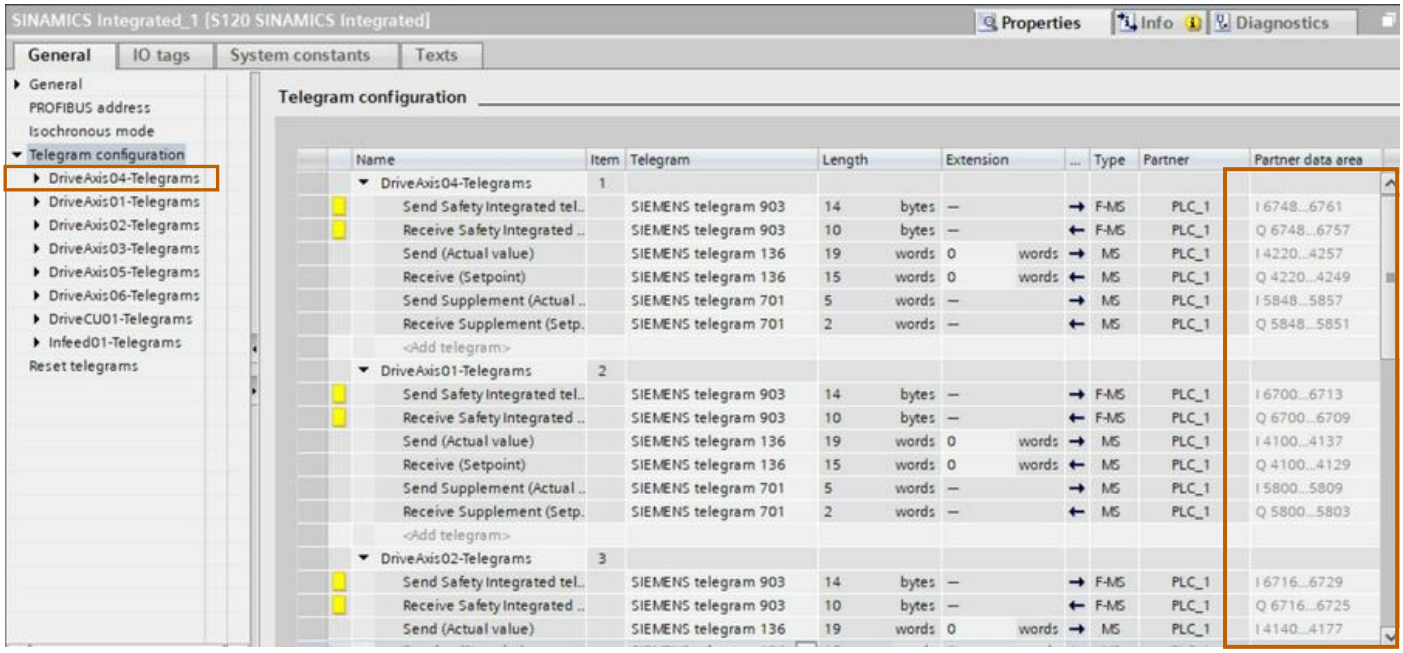
If the checkmark is not set, the status of the change is shown in area "Current configuration", and the modified configuration in the "New configuration" area.

In this way you can again view the original configuration, and when required, restore it to avoid inadvertent changes.

If the checkmark was set, then the content under "Current configuration" corresponds to the updated configuration, and area "New configuration" is empty

Notes:

New setting of the safety telegram addresses with DriveAxis04 (Spindle) at the top of the telegram list



Notes:

The hardware configuration has resulted in the following Input/ Output assignments.

These will be used to achieve the functionality of the OEM function table when writing the F-PLC logic (described in Module A105a)

Training Simulator Output Functionality	
Function	Address
DO 0	Q100.0
DO 1	Q100.1
DO 2	Q100.2
DO 3	Q100.3
DO 4	Q100.4
DO 5	Q100.5
DO 6	Q100.6
DO 7 Door Unlock	Q100.7
Q0 Relay K1 & LED	Q530.0
Q1 Relay K2 & LED	Q530.1
Q2 Safety Internal Event LED	Q530.2
Q3 Emergency Stop Ack Request LED	Q530.3
F-DO 0	Q535.0
F-DO 1	Q535.1
F-DO 2	Q535.2
F-DO 3	Q535.3
Q Safety Power Module	Q540.0

Training Simulator Input Functionality	
Function	Address
DI 0 Door Unlock Request	I100.0
DI 1	I100.1
DI 2	I100.2
DI 3	I100.3
DI 4	I100.4
DI 5	I100.5
DI 6 K1 & K2 Relay Feedback	I100.6
DI 7	I100.7
Emergency Stop	I512.0
F-DI 2 & F-DI6	I512.1
Door Locked	I512.2
Door Closed	I512.3
F-DI 0 & F-DI4	I518.0
Emergency Stop Reset	I518.1
Set-up Mode	I518.2
F-DI 3 & F-DI7	I518.3
Hold-To-Run PB	I524.0
F-DI 1 & F-DI5	I524.1
F-DI 2 & F-DI6	I524.2
F-DI 3 & F-DI7	I524.3

Safety

	Name	Data type	Address
1	SafeSpeedMonitorA	Bool	%I6765.7
2	AckIntEventA	Bool	%Q6764.7
3	AckIntEventSP	Bool	%Q6748.7
4	AckIntEventX	Bool	%Q6700.7
5	AckIntEventY	Bool	%Q6716.7
6	AckIntEventZ	Bool	%Q6732.7
7	DeselectSOSA	Bool	%Q6764.3
8	DeselectSS1A	Bool	%Q6764.1
9	DeselectSS2A	Bool	%Q6764.2
10	DeselectSTOA	Bool	%Q6764.0
11	DeselectSOSSP	Bool	%Q6748.3
12	DeselectSS1SP	Bool	%Q6748.1
13	DeselectSS2SP	Bool	%Q6748.2
14	DeselectSTOSP	Bool	%Q6748.0
15	DeselectSOSX	Bool	%Q6700.3
16	DeselectSS1X	Bool	%Q6700.1
17	DeselectSS2X	Bool	%Q6700.2
18	DeselectSTOX	Bool	%Q6700.0
19	DeselectSOSY	Bool	%Q6716.3
20	DeselectSS1Y	Bool	%Q6716.1
21	DeselectSS2Y	Bool	%Q6716.2
22	DeselectSTOY	Bool	%Q6716.0
23	DeselectSOSZ	Bool	%Q6732.3
24	DeselectSS1Z	Bool	%Q6732.1
25	DeselectSS2Z	Bool	%Q6732.2
26	DeselectSTOZ	Bool	%Q6732.0
27	HoldToRunButton	Bool	%I524.0
28	InternalEventActiveA	Bool	%I6764.7
29	InternalEventActiveSP	Bool	%I6748.7
30	InternalEventActiveX	Bool	%I6700.7
31	InternalEventActiveY	Bool	%I6716.7
32	InternalEventActiveZ	Bool	%I6732.7
33	SLSBit0A	Bool	%Q6765.1
34	SLSBit0SP	Bool	%Q6749.1
35	SLSBit0X	Bool	%Q6701.1
36	SLSBit0Y	Bool	%Q6717.1
37	SLSBit0Z	Bool	%Q6733.1
38	SLSBit1A	Bool	%Q6765.2

Safety

	Name	Data type	Address
39	SLSBit1SP	Bool	%Q6749.2
40	SLSBit1X	Bool	%Q6701.2
41	SLSBit1Y	Bool	%Q6717.2
42	SLSBit1Z	Bool	%Q6733.2
43	SafeSpeedMonitorSP	Bool	%I6749.7
44	SafeSpeedMonitorX	Bool	%I6701.7
45	SafeSpeedMonitorY	Bool	%I6717.7
46	SafeSpeedMonitorZ	Bool	%I6733.7
47	SOSActiveX	Bool	%I6700.3
48	SOSActiveA	Bool	%I6764.3
49	SOSActiveSP	Bool	%I6748.3
50	SOSActiveZ	Bool	%I6732.3
51	SOSActiveY	Bool	%I6716.3
52	SetupSwitch	Bool	%I518.2
53	EmergencyStopResetPB	Bool	%I518.1
54	Q PowerModule	Bool	%Q540.0
55	Mode3Switch	Bool	%I518.3
56	DoorLocked	Bool	%I512.2
57	DoorClosed	Bool	%I512.3
58	End Limit Override	Bool	%I518.0
59	Safe Cam1 DO 0 LED	Bool	%Q535.0
60	Safe Cam2 DO 0 LED	Bool	%Q535.1
61	Q1 Relay K2 Door Closed	Bool	%Q530.1
62	Q0 Relay K1 Door Closed and L...	Bool	%Q530.0
63	Internal Event Lamp	Bool	%Q530.2
64	EmergencyStopPB	Bool	%I512.0
65	SPDeselectSLS	Bool	%Q6748.4
66	SelectSLPRange2X	Bool	%Q6702.3
67	SLPUpperLimitX	Bool	%I6703.6
68	SLPLowerLimitX	Bool	%I6703.7
69	SDIPositiveX	Bool	%Q6701.4
70	SafelyReferencedX	Bool	%I6702.7
71	SDINegativeX	Bool	%Q6701.5
72	SafeCamsActivatedX	Bool	%Q6702.7
73	SafeCamValid	Bool	%I6707.6
74	SafeCam1X	Bool	%I6704.0
75	SafeCam2X	Bool	%I6704.1
76	ESTOPResetRequired	Bool	%Q530.3

The Tag tables for the F-PLC can be prepared as shown here, or the tag names can be entered individually as the PLC logic is created.

Safety Other

	Name	Data type	Address
1	DoorUnlockButton	Bool	%I100.0
2	K1/K2feedback	Bool	%I100.6
3	DoorUnlockRequest	Bool	%Q100.7

Notes:

Safety Integrated Drive Configuration

The required Safety Integrated functionality for each drive are configured in the Sinumerik Operate Set-up area.

This section will achieve the following:

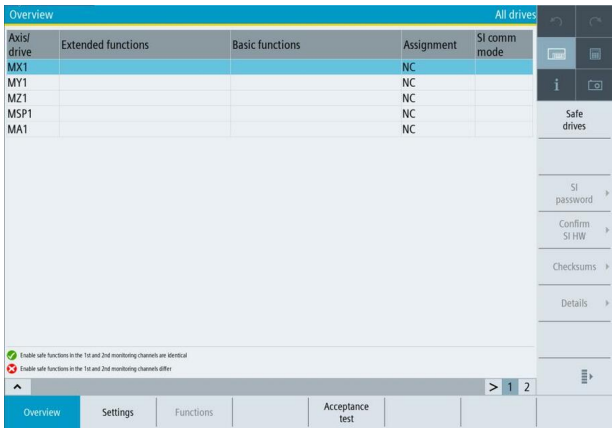
- Enable Safety Integrated extended functions for each drive (see the adjacent table)
- Setpoint Limitation - used to limit the feed-rate of an axis to a percentage of the active Safe Limited Speed
- Setting of the Test Stop interval time
- Safe encoder configuration
- Actual value synchronisation for 2nd encoders
- Drive telegram configuration

Basic and Extended Safety Functions:

Safe Functions		Abbreviation
Basic Functions	Safe Torque Off	STO
	Safe Stop 1	SS1
	Safe Brake Control	SBC
Extended Functions (includes basic functions)	Safe Stop 2	SS2
	Safe Operating Stop	SOS
	Safely Limited Speed	SLS
	Safe Speed Monitor	SSM
	Safe Acceleration Monitor	SAM
	Safe Direction	SDI
	Safe Limit Positions	SLP
	Safe Brake Test	SBT
	Safe Cams	SCA

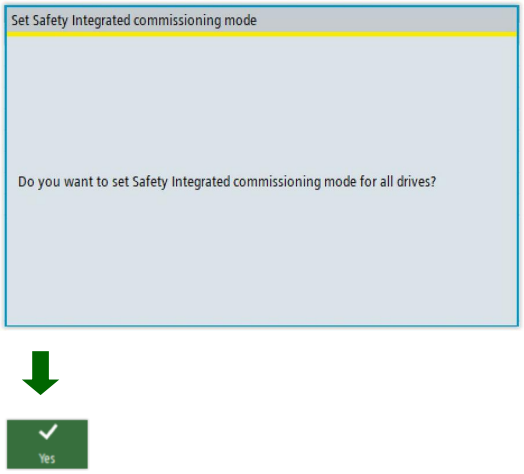
Activating Safety Integrated Commissioning Mode:



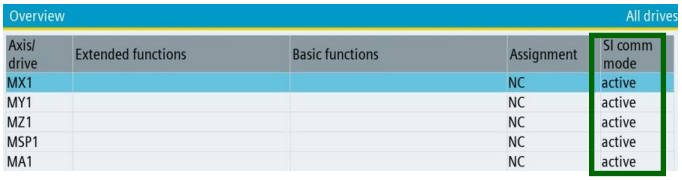


The commissioning mode needs to be set for the axes:





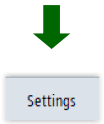
The active commissioning mode status is displayed in the right hand column:

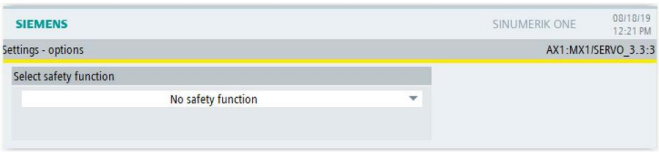


Notes:

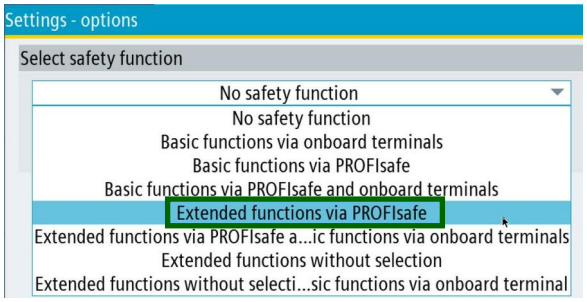
Settings.

Extended Functions

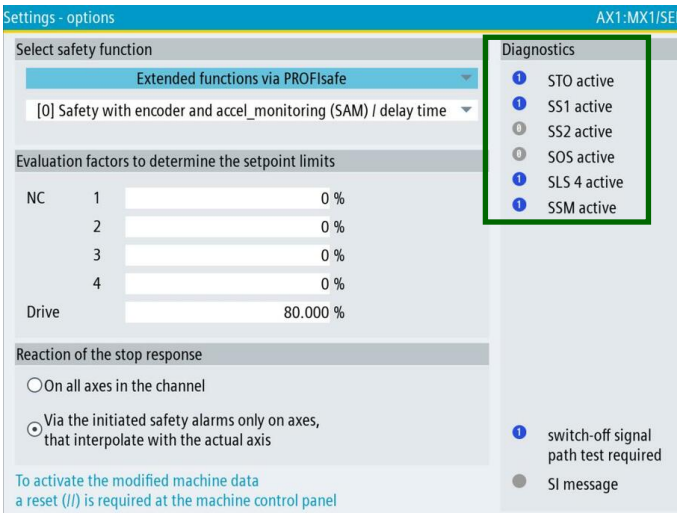




Select the ***Extended functions via PROFsafe*** option. This offers the following Safe functions:



Shown below are the Safety Functions available when the Extended functions via PROFIsafe option is selected

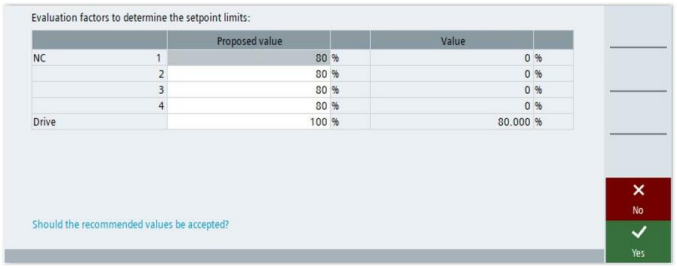


Set-point Limitation

Set-point limitation is used to limit the feed-rate of an axis to a percentage of the active safe limited speed (**SLS**).

Selecting Recommended values proposes values of 80% and these values should be accepted.







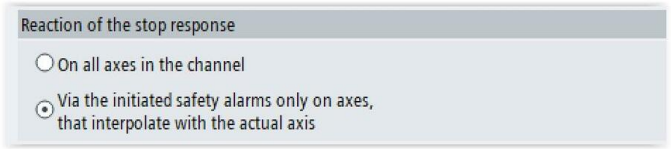
The 80% set-point limitation will limit the normal operating feed rates to 80% of the active SLS limitation.
The SLS limitations are set later in the document.

Reaction to a stop response

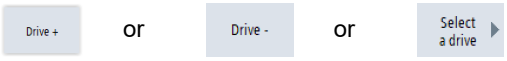
Each axis can be set to react to a stop response in one of two ways:

1. On all axes in the channel
2. Via the initiated safety alarms on axes that interpolate with the actual axis (recommended)

The selection made will depend on the individual machine configuration. For example, a fault on an auxiliary axis, such as a tool changer, can be ignored and the machining axes allowed to continue.



Now repeat for all of the drives

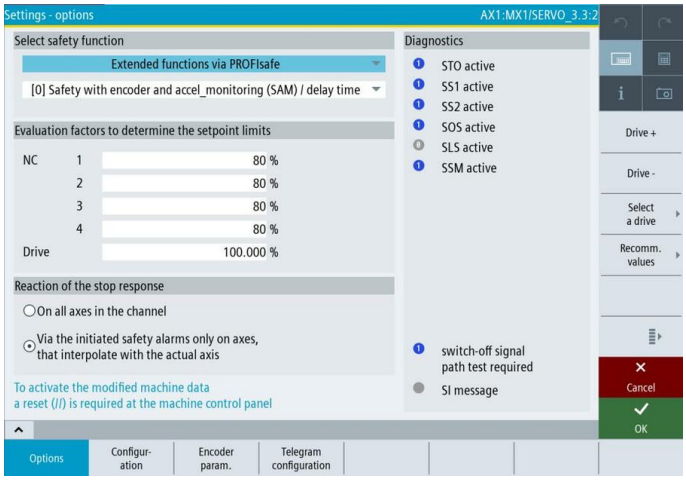


Notes:

Saving the changes:

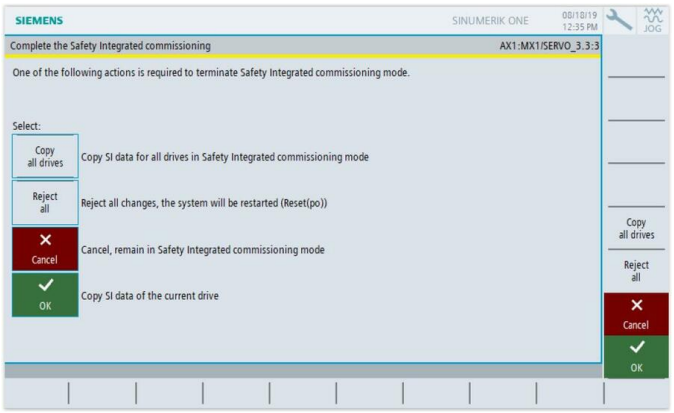
The changes need to be copied and confirmed. The values entered into the screens go into the *Control Unit* specific data. With the copy function the data is copied into the *Motor Module* data. With confirm the *Control Unit and Motor Module data* checksums will then match each other.

The checksums are monitored in two channels *(The Control Unit and the Motor Module).* Any discrepancies will result in Safety Integrated alarms.



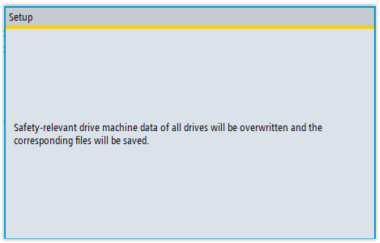


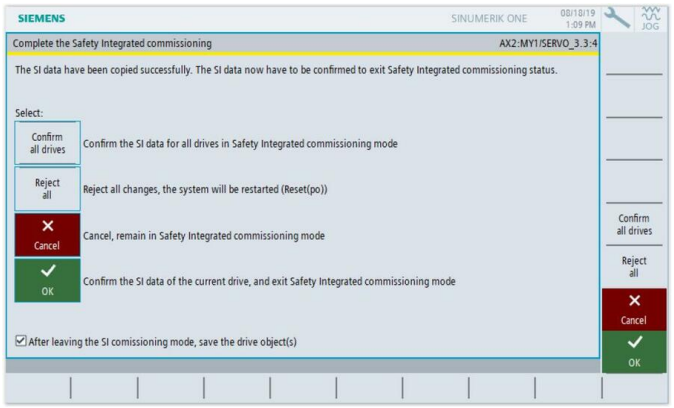
Start the procedure for copy/confirm





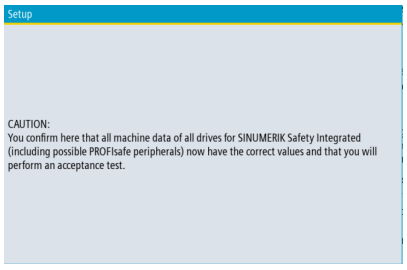
Copy the SI data of all drives



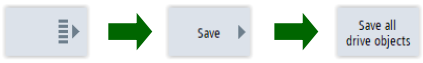


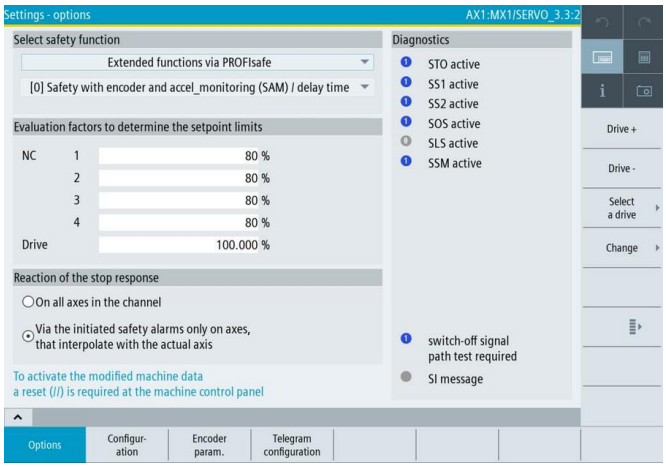


Confirm the SI data of all drives

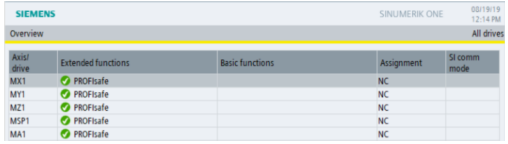


Save the SI data of all drives





Extended functions via Profisafe are now set for all relevant drives



Notes:

Configuration of the Safety Integrated features.



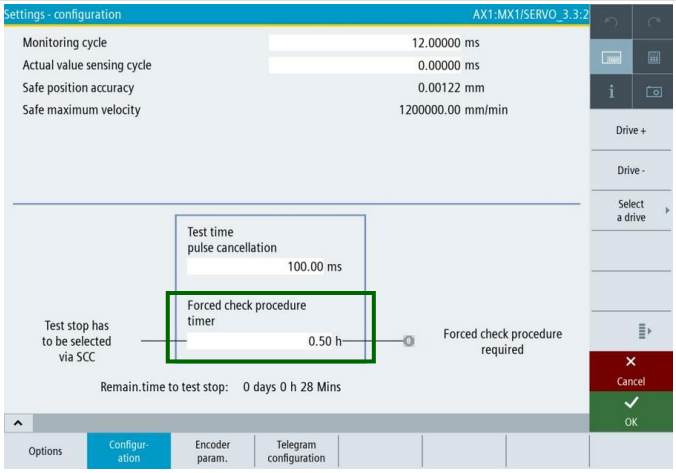
Activate the ***SI commissioning*** mode:



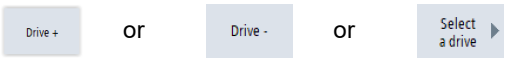
Test Stop Interval Time:

The forced checking procedure is used to detect faults/errors in the software and hardware of the two monitoring channels. In order to do this, the safety - related parts in both channels must be processed at least once during a defined period in all safety-related branches. Any faults/errors in the monitoring channel would cause deviations and will be detected by the crosswise data comparison.

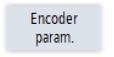
For the purpose of the training this can be set to a low value, e.g. 0.5 hour. This will allow the relevant Test Stop interface signals/messages to be monitored without the need to wait. All other settings can be left as standard.

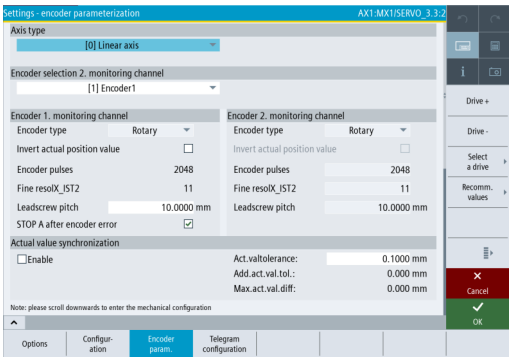


Now repeat for all of the drives

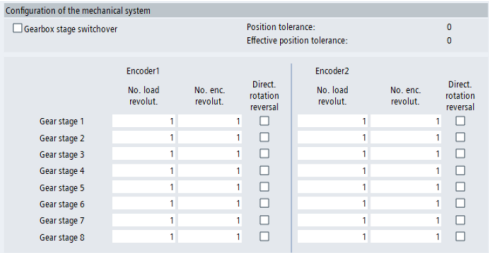


Encoder Parameters:





The axis type and encoder information etc. has to be entered for each drive.Gearbox information is also available by using the scroll bar to go to the lower half of the screen

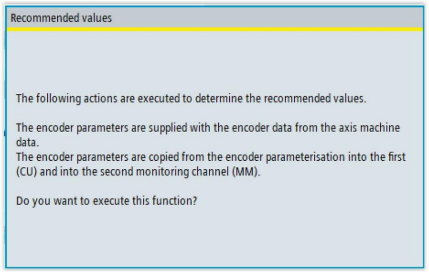


Example configuration of the training equipment can be seen in the table below

Axis	Type	Encoders	Leadscrew	Gears
MX1	Linear	1	10.00 mm	n/a
MY1	Linear	1	10.00 mm	n/a
MZ1	Linear	1	10.00 mm	n/a
MSP1	Rotary	2	n/a	n/a
MA1	Rotary	1	n/a	n/a

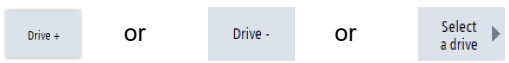


Recommended values can be set



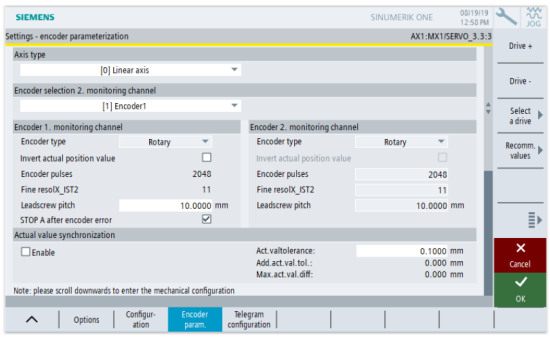


Now repeat for all of the drives

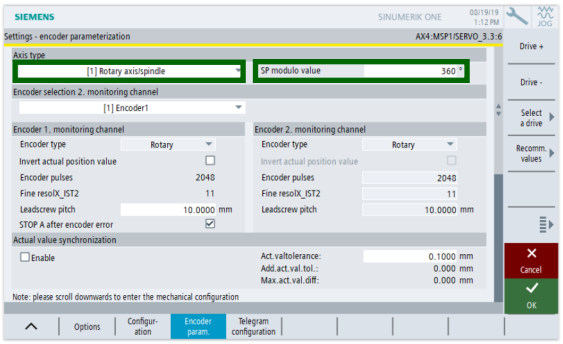


Notes:

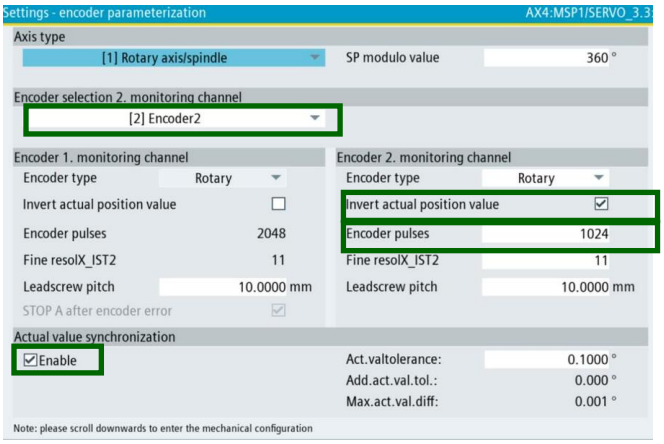
Below, X, Y and Z after recommended values



For the Main Spindle and the A Axis, Axis type [1] Rotary axis/ spindle and SP modulo value 360 should be used.
Encoder pulses for Spindle [1]Encoder1 will be 2048 and for the A Axis [1]Encoder1 will be 512.



For the main spindle with second encoder the [2] Encoder2 should be selected for the monitoring channel.
Also Invert position actual value.
Encoder pulses = 1024.
Enable Actual value synchronization.



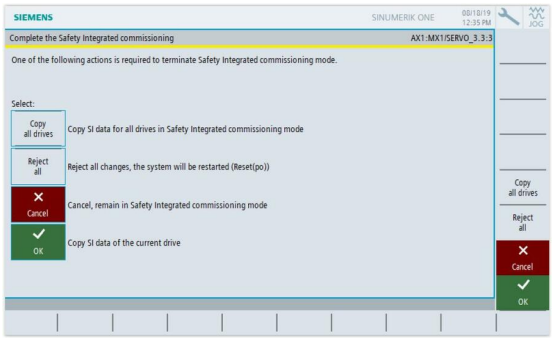
Position actual value is inverted due to the mounting of the spindle 2nd encoder (on the front of the spindle) therefore the feedback pulses are in the opposite direction.
The 2nd encoder has 1024 pulses/rev.

The actual value synchronization allows a tolerance to be entered on systems with two encoders. This prevents nuisance alarms which can occur due to the mechanical properties of gear boxes or belt driven systems etc.



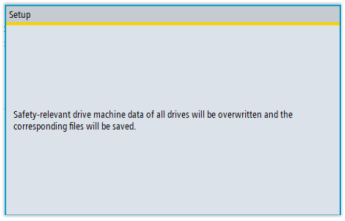
Start the procedure for copy/confirm

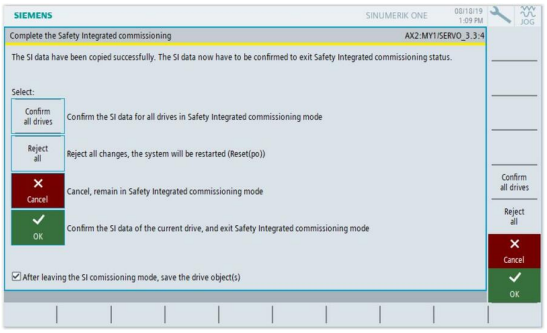
NOTE: It will be necessary to copy/confirm twice because of Spindle and A Axis units change.





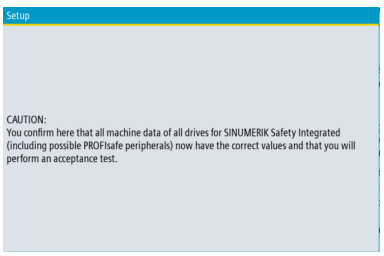
Copy the SI data of all drives



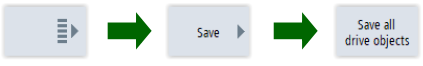




Confirm the SI data of all drives



Save the SI data of all drives



Notes:

Telegram Configuration:

The **Telegram configuration** area is used to enable the ***S**afety **I**nformation **C**hannel (**SIC**), the **S**afety **C**ommand **C**hannel (**SCC**)* and ***PROFIsafe*** for each drive.The required Telegrams are also selected here, this allows interfacing between the drive and user PLC for monitoring and controlling of the safety integrated functions.

The required SIC/SCC telegram is telegram 701. This allows the Safe Brake Test to be performed and the starting of the Test Stop procedure via the axis interface DB.

The standard PROFIsafe telegram, is telegram 903. This can be changed to achieve different functionality (refer to the Safety Integrated theory document)

The SCC/SIC & PROFIsafe drive module numbers are also set here. These values depend on the drive topology/drive number and the Axis Index. The training equipment configuration is as follows:

Axis	MD37954 (SIC/SCC NR) SAFE_INFO_ MODULE_NR		MD13374 SAFE_INFO_ DRIVE_LOGIC _ADDRESS		From Telegram to Axis DB	Addressing For Telegram 701
MX1	1		[0] = 5800		DB31	
MY1	2		[1] = 5816		DB32	
MZ1	3		[2] = 5832		DB33	
MSP1	4		[3] = 5848		DB34	
MA1	5		[4] = 5864		DB35	

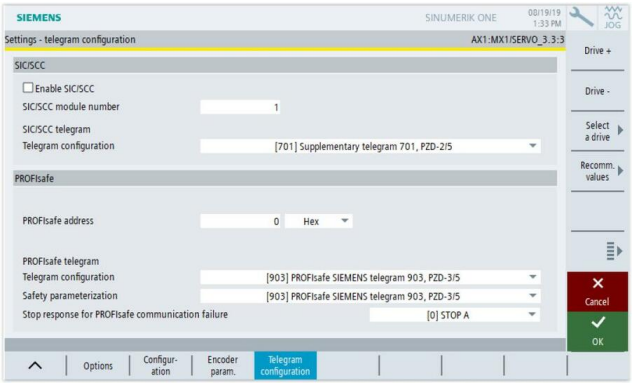
Drive	Telegram Configuration
DO3	DriveAxis04
DO4	DriveAxis01
DO5	DriveAxis02
DO6	DriveAxis03
DO7	DriveAxis05

Item	F– Destination Address		Start Address		Addressing For Telegram 903
1	4		6748		
2	1		6700		
3	2		6716		
4	3		6732		
5	5		6764		

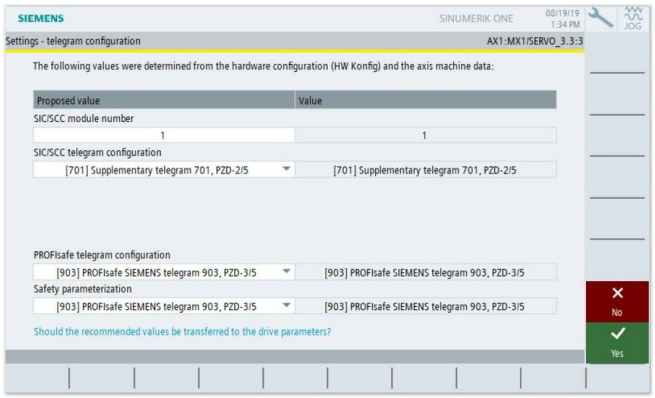
Drive	Telegram Configuration
DO3	DriveAxis04
DO4	DriveAxis01
DO5	DriveAxis02
DO6	DriveAxis03
DO7	DriveAxis05

Check that the values match the values in the previous table. If they do not then edit them manually.

Telegram configuration



Recomm. values



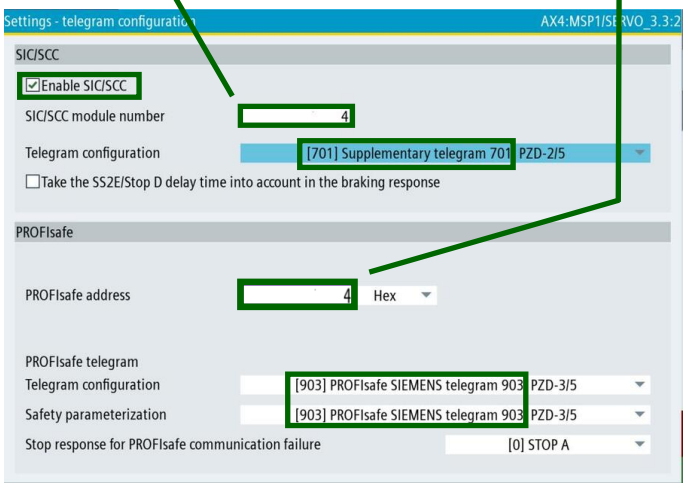
The SIC/SCC 701 telegram is also set here and also the PROFIsafe 903 telegram.

OK

Notes:

Axis	MD37954 (SIC/SCC NR) SAFE_INFO_MODULE_NR
MX1	1
MY1	2
MZ1	3
MSP1	4
MA1	5

F– Destination Address
4
1
2
3
5



- Set the ***Enable SIC/SCC*** option
- Check that ***Telegram 701*** has been selected for the telegram configuration for the SIC/SCC.
- Check that ***Telegram 903*** has been selected for the Telegram configuration and Safety parameterization of PROFIsafe

Check the following settings in accordance with the table on the previous page.

- SIC/SCC module number
- PROFISafe address - ***set manually***

Now repeat for all of the drives

or

or





Start the procedure for copy/confirm

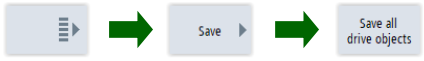


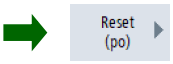
Copy the SI data of all drives



Confirm the SI data of all drives

Save the SI data of all drives







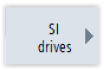
Confirm all drives a second time if required because of Spindle/A Axis units

Notes:

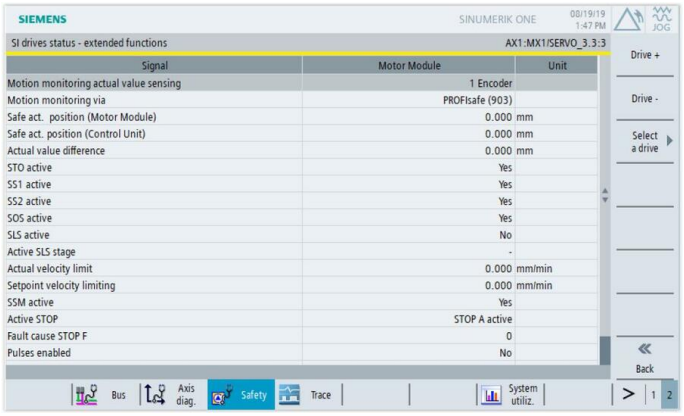
Safety Integrated Drive/Telegram Status.

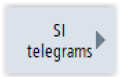
The telegram and drive statuses can be seen in the Safety Integrated diagnostics area





The status of the standard and extended safety functions can be seen on the ***SI drives*** screen.

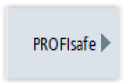




The assigned telegrams can be seen on ***SI telegrams*** screen.

Overview of SI telegrams

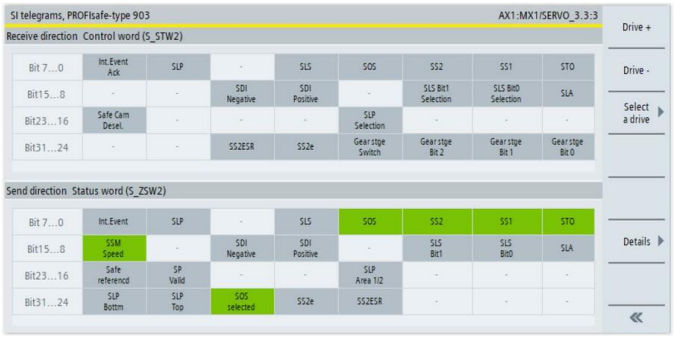
Axis/drive	PROFIsafe telegram type	SIC/SCC telegram type
AX1:MX1/SERVO_3.3:3	903	701
AX2:MY1/SERVO_3.3:4	903	701
AX3:MZ1/SERVO_3.3:5	903	701
AX4:MSP1/SERVO_3.3:2	903	701
AX5:MA1/SERVO_3.3:6	903	701

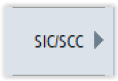


The PROFIsafe telegram status of each drive can be viewed.
The Status isn't displaying anything due to the missing F-PLC logic.



These signals are used to initiate/monitor the Test Stop and Safe Brake Test routines.





The Safety Information Channel (SCC) and Safety Command Channel (SCC) status of each drive can be viewed.

Notes:

F-PLC Logic

PLC I/O used to achieve Safety functionality

The following tables show the training equipment Input/Output assignments which were determined during the hardware configuration.
These signals will be used to achieve the functionality of the OEM function table.

Training Simulator Output Functionality	
Function	Address
DO 0	Q100.0
DO 1	Q100.1
DO 2	Q100.2
DO 3	Q100.3
DO 4	Q100.4
DO 5	Q100.5
DO 6	Q100.6
DO 7 Door Unlock	Q100.7
Q0 Relay K1 & LED	Q530.0
Q1 Relay K2 & LED	Q530.1
Q2 Safety Internal Event LED	Q530.2
Q3 Emergency Stop Ack Request LED	Q530.3
F-DO 0	Q535.0
F-DO 1	Q535.1
F-DO 2	Q535.2
F-DO 3	Q535.3
Q Safety Power Module	Q540.0

Training Simulator Input Functionality	
Function	Address
D I0 Door Unlock Request	I100.0
DI 1	I100.1
DI 2	I100.2
DI 3	I100.3
DI 4	I100.4
DI 5	I100.5
DI 6 K1 & K2 Relay Feedback	I100.6
DI 7	I100.7
Emergency Stop	I512.0
F-DI 2 & F-DI6	I512.1
Door Locked	I512.2
Door Closed	I512.3
F-DI 0 & F-DI4	I518.0
Emergency Stop Reset	I518.1
Set-up Mode	I518.2
F-DI 3 & F-DI7	I518.3
Hold-To-Run PB	I524.0
F-DI 1 & F-DI5	I524.1
F-DI 2 & F-DI6	I524.2
F-DI 3 & F-DI7	I524.3

Safety Functionality between the Drive and the PLC.

The required Safety Integrated functionality is achieved by setting the interface signals (determined by the PROFISafe telegram structure refer to the previous page) as shown below.

PLC to Drive (Telegram S_STW2 Control words 1 & 2)					
Function	Sp QB6748	X QB6700	Y QB6716	Z QB6732	A QB6764
Deselect STO	.0	.0	.0	.0	.0
Deselect SS1	.1	.1	.1	.1	.1
Deselect SS2	.2	.2	.2	.2	.2
Deselect SOS	.3	.3	.3	.3	.3
Deselect SLS	.4	.4	.4	.4	.4
Reserved Out 1	.5	.5	.5	.5	.5
Deselect SLP	.6	.6	.6	.6	.6
Ack. Int. Event	.7	.7	.7	.7	.7
	QB6749	QB6701	QB6717	QB6733	QB6765
Reserved Out 2	.0	.0	.0	.0	.0
Select SLS bit 0	.1	.1	.1	.1	.1
Select SLS bit 1	.2	.2	.2	.2	.2
Reserved Out 3	.3	.3	.3	.3	.3
Deselect SDI Pos	.4	.4	.4	.4	.4
Deselect SDI Neg	.5	.5	.5	.5	.5
Reserved Out 4	.6	.6	.6	.6	.6
Reserved Out 5	.7	.7	.7	.7	.7

PLC to Drive (Telegram S_STW2 Control words 2 & 3)					
Function	Sp QB6750	X QB6702	Y QB6718	Z QB6734	A QB6766
Reserved Out 6	.0	.0	.0	.0	.0
Reserved Out 7	.1	.1	.1	.1	.1
Reserved Out 8	.2	.2	.2	.2	.2
Select SLP range	.3	.3	.3	.3	.3
Reserved Out 9	.4	.4	.4	.4	.4
Reserved Out 10	.5	.5	.5	.5	.5
Reserved Out 11	.6	.6	.6	.6	.6
Des. Safe Cam	.7	.7	.7	.7	.7
	QB6751	QB6703	QB6719	QB6735	QB6767
Select Gear bit 0	.0	.0	.0	.0	.0
Select Gear bit 1	.1	.1	.1	.1	.1
Select Gear bit 2	.2	.2	.2	.2	.2
Deselect Gear Sw	.3	.3	.3	.3	.3
Deselect SS2E	.4	.4	.4	.4	.4
Deselect SS2ESR	.5	.5	.5	.5	.5
Reserved Out 14	.6	.6	.6	.6	.6
Reserved Out 15	.7	.7	.7	.7	.7
SLS1 Limit	QW6752	QW6704	QW6720	QW6736	QW6768

Notes:

Safety Integrated Status

The Safety Integrated status signals can be monitored and used in the F-PLC logic- as shown in the following tables.

Drive to PLC (Telegram S_ZSW2 Status words 1 & 2)					
Function	Sp IB6748	X IB6700	Y IB6716	Z IB6732	A IB6764
STO Active	.0	.0	.0	.0	.0
SS1 Active	.1	.1	.1	.1	.1
SS2 Active	.2	.2	.2	.2	.2
SOS Active	.3	.3	.3	.3	.3
SLS Active	.4	.4	.4	.4	.4
Reserved Input 1	.5	.5	.5	.5	.5
SLP Active	.6	.6	.6	.6	.6
Int. Event Active	.7	.7	.7	.7	.7
	IB6749	IB6701	IB6717	IB6733	IB6765
Reserved In 2	.0	.0	.0	.0	.0
SLS bit 0 Active	.1	.1	.1	.1	.1
SLS bit 1 Active	.2	.2	.2	.2	.2
Reserved Input 3	.3	.3	.3	.3	.3
SDI Pos. Active	.4	.4	.4	.4	.4
SDI Neg. Active	.5	.5	.5	.5	.5
Reserved Input 4	.6	.6	.6	.6	.6
SSM n>nx	.7	.7	.7	.7	.7

Drive to PLC (Telegram S_ZSW2 Status words 3 & 4)					
Function	Sp IB6750	X IB6702	Y IB6718	Z IB6734	A IB6766
Reserved Input 5	.0	.0	.0	.0	.0
Reserved Input 6	.1	.1	.1	.1	.1
Reserved Input 7	.2	.2	.2	.2	.2
SLP Range Active	.3	.3	.3	.3	.3
Reserved Input 8	.4	.4	.4	.4	.4
Reserved Input 5	.5	.5	.5	.5	.5
Safe Pos. Valid	.6	.6	.6	.6	.6
Safely Referenced	.7	.7	.7	.7	.7
	IB6751	IB6703	IB6719	IB6735	IB6767
F-DI 0	.0	.0	.0	.0	.0
F-DI 1	.1	.1	.1	.1	.1
F-DI 2	.2	.2	.2	.2	.2
SS2E ESR	.3	.3	.3	.3	.3
SS2E Active	.4	.4	.4	.4	.4
SOS Selected	.5	.5	.5	.5	.5
SLP Upper Limit	.6	.6	.6	.6	.6
SLP Lower Limit	.7	.7	.7	.7	.7

The signals from PROFIsafe telegram 701 are transferred into the relevant Axis interface PLC DBs 31/32/33/34/35 etc

The PLC interface can then be used to initiate and start the Safe Brake Test and also the Test Stop (Forced checking procedure. Each axis has a data block reserved for safety integrated signals to/from the PLC and drive.

Signals from the PLC to the drive (Safety Command Channel):

DB31 ...DB61	Signals from PLC to Drive (Safety Command Channel—SCC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB140								Start Test Stop
DBB143			Brake Closed	Torque 1 or 2	Direction of Rotation	Brake 1 or 2	BT Start	Selection of SBT

Signals from the drive to the PLC (Safety Information Channel):

DB31 ...DB61	Signals from Drive to PLC (Safety Information Channel—SIC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB166			Test Stop Required	Test Stop Active				
DBB173	Sign Load Torque	Brake Req.	B test finished	B test OK	B test Active	Brake 1 or 2	B test Start	Selection Safe BT

Notes:

Safety Integrated PLC Logic

The F-PLC logic should be written to achieve the functionality of the OEM function table. The program is written in the program editor in fail-safe FBs, FCs and DBs **in the programming language LAD or FDB.**

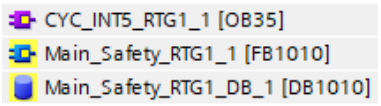
Note: the logic examples shown is this document are for training purposes. Logic should be written for each specific application. It is the responsibility of the OEM to ensure that the logic meets the required safety standards.

Runtime Group 1 (RTG1)
Safety specific PLC blocks are used to for the F-PLC logic. These were created when the option for Safety was activated on the PLC CPU. These are part of Runtime Group 1 (RTG1).

- A maximum of 2 Runtime Groups can be specified

The Runtime Group consists of:

- CYC_INT5_RTG1 [OB35] - Cyclic OB for Safety.
- Main_Safety_RTG1 [FB1010] - Function Block for the F-PLC logic.
- Main_Safety_RTG1 [DB1010} - Instance Data Block for FB1010.



Number range reserved for RTG1 Sinumerik:

- F-FB 1010...1019
- F-FC 1010...1019
- F-DB 1010...1019

Data exchange between standard user program and safety program

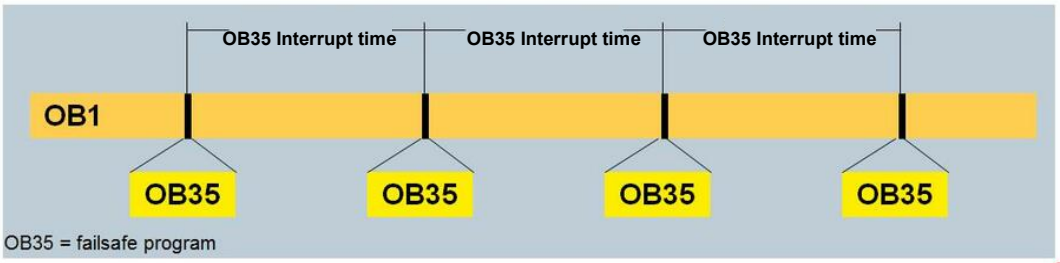
	From the standard user program		From the safety program	
	Read access	Write access	Read access	Write access
Variable from DB	permissible	permissible	Either read access *OR* write access to a variable from the DB	
Variable from F-DB	permissible	**not permissible**	permissible	permissible
Bit memory	permissible	permissible	Either read access *OR* write access to a bit memory	

		From the standard user program		From the safety program	
		Read access	Write access	Read access	Write access
Process image Standard I/O	PAE	permissible	permissible	permissible	**not permissible**
	PAA	permissible	permissible	**not permissible**	permissible
Process image F-I/O	PAE	permissible	**not permissible**	permissible	**not permissible**
	PAA	permissible	**not permissible**	**not permissible**	permissible

WARNING

Since variables from the standard user program are not generated safely, it must be ensured with the implementation of additional process specific plausibility checks in the safety program that there is no risk of hazardous conditions developing.

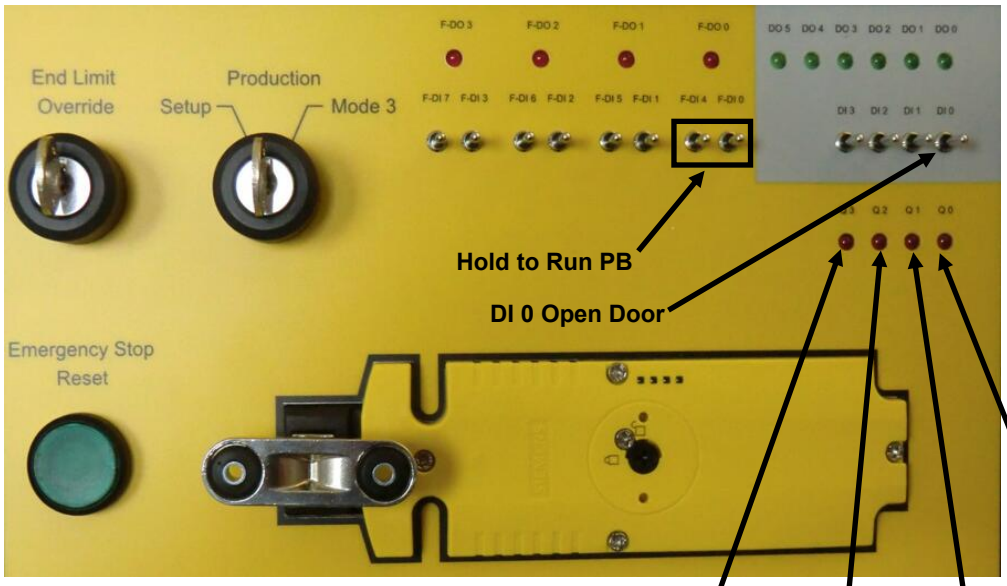
OB35 Cycle Time;
The response time/performance is dependant on the number of axes/safety functions in use :



The default OB35 interrupt time is 100ms. This can be optimised to a lower value and will be dependant on the number of Safety Integrated axes/inputs/outputs/functions in use.

Notes:

Safety Simulator
Sinumerik ONE

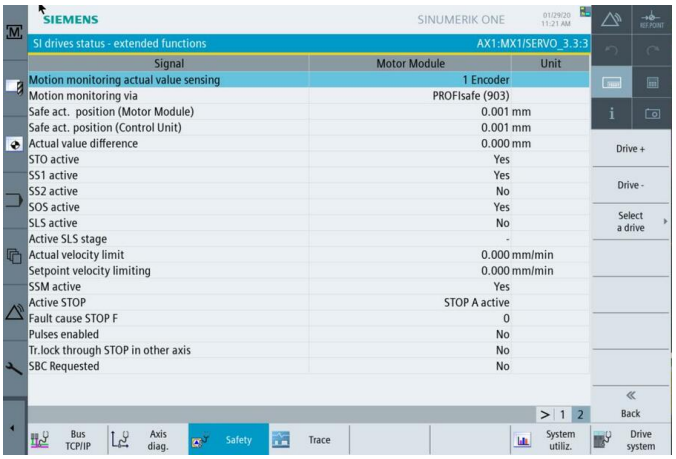


Notes:

Checking the logic:
The logic for the various switches can be checked using the OEM function table and the diagnostic screens.
This will be further tested as part of the Safety Integrated Acceptance Test (SIAT).



In the diagnostic screens it can be seen that after download/ PO reset, all axes/spindle are shown with SS1, STO, SOS.



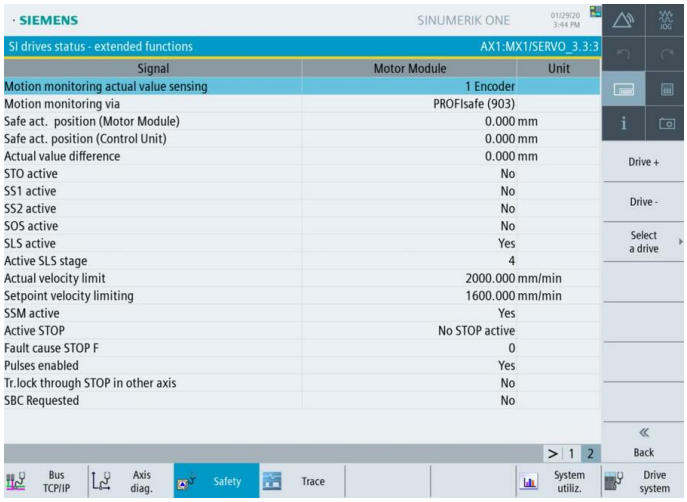
Use the sequence Release ESTOP PB, press Reset PB on MCP, press E-Stop Reset PB to reset the Emergency Stop condition.

The Emergency Stop Message should disappear and then the Axes should show only SOS and the Spindle should show STO.

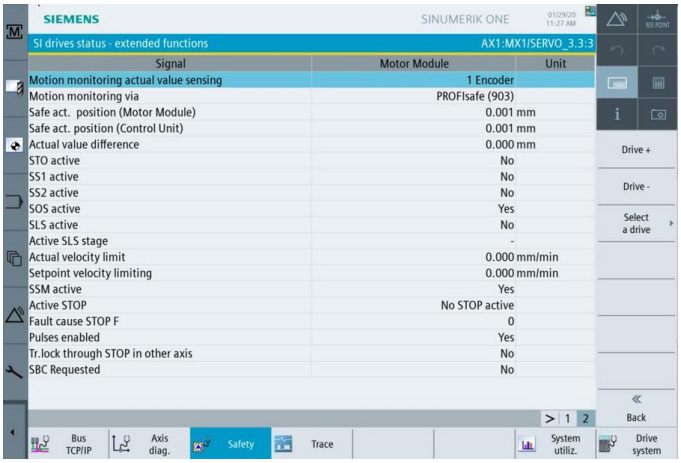
When the Operator Door is closed and locked all axes and spindle should switch to SLS4.

At this stage the SLS setpoint velocity limitation is shown as 1600 mm/min because these limits have not been set up yet. The default SLS speeds are 2000 mm/min for all 4 SLS values therefore with 80% setpoint limitation a value of 1600 mm/min is correct.

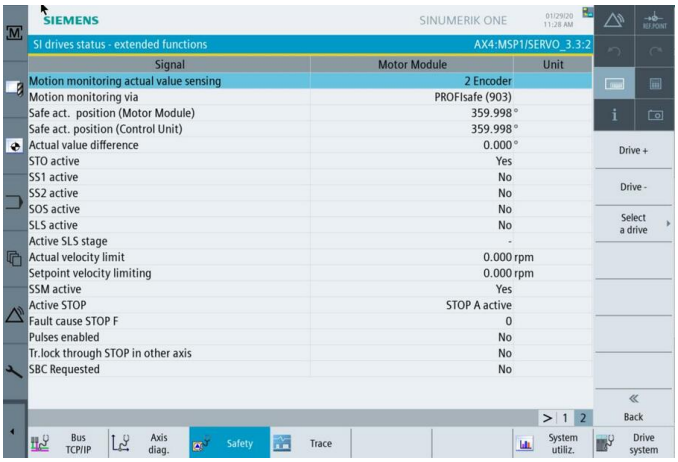
It should now be possible to move the axes in Jog mode.



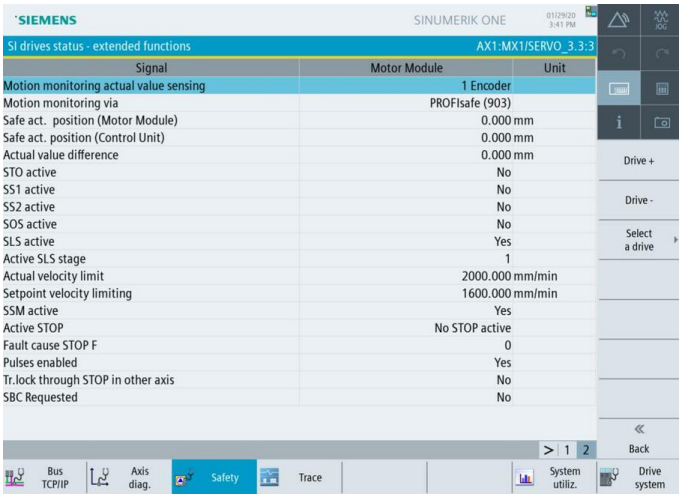
If the Operator Door is opened then the axes should switch to SOS and the spindle should be in an STO condition



Notes:



If the Operator Door is opened then spindle should be in an STO condition

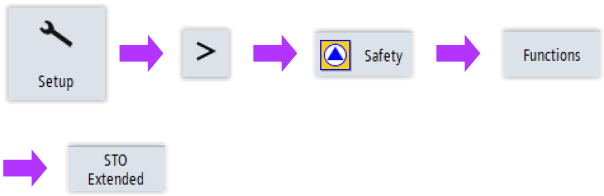


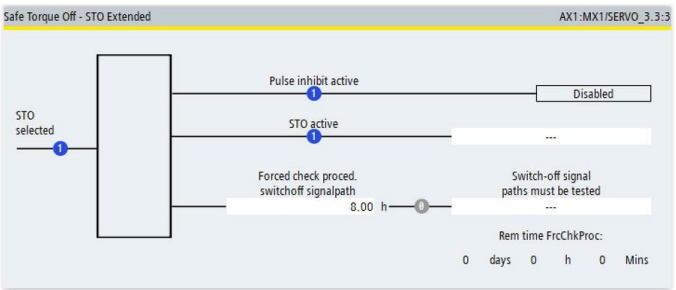
With the Operator Door Open, then the Keyswitch being in Setup Mode and the Hold to Run PB being pressed all Axes are showing SLS1 however the spindle is still in an STO condition.

The commissioning now continues on the HMI in the Setup/ Safety area.

The PLC will be modified later as more Safety functionality is commissioned.

Safe Torque Off-Extended (STO Extended)





The ***Safe Torque Off - Extended*** function allows for the STO status of each safe drive to **be monitored, via a BiCo**, in the CU or any of the drives.

Example: the signal could be used to set a digital output on the CU.

It is also possible to set a time period for the ***Forced check proced. Switchoff signalpath.*** (Test stop interval time) and, when the time value has expired, set a CU/Drive BiCo via the ***Switch-off paths must be tested*** area.

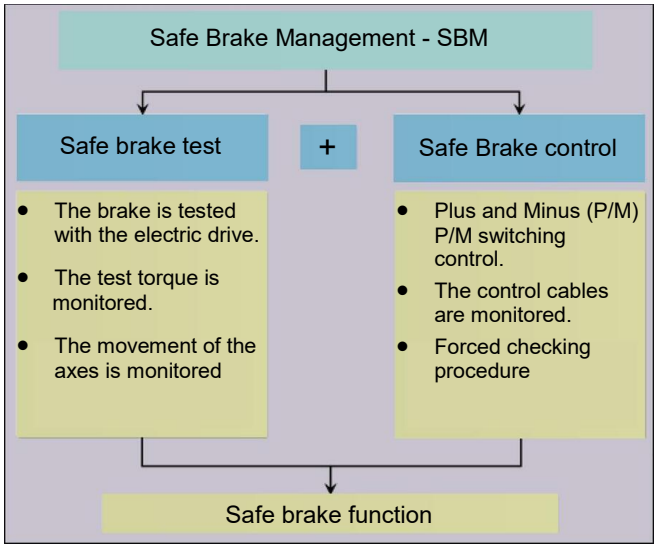
Alarm, ***201699 - Axis ??, component ??: SI P1(CU) Switch -off signal paths must be tested.***, will also be displayed. This will clear once the test stop has been completed. The ***Rem time FrcChkProc*** will be also reset.

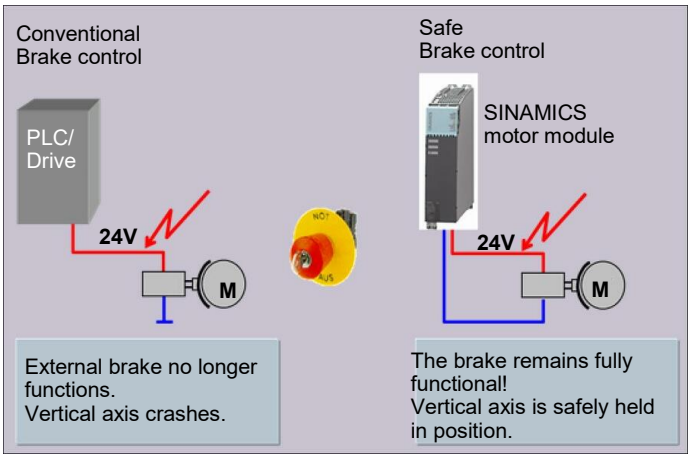
Note: The default settings can be used in this example.

Notes:

Safe Brake Management

Safe Brake Control (SBC) and Safe Brake Test (SBT), together form Safe Brake Management and is available for axes fitted with mechanical brakes (integral motor brakes or external brakes).

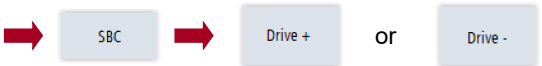


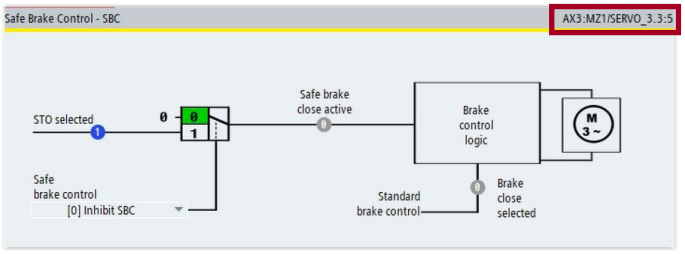


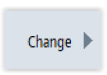
Safe Brake Control - SBC

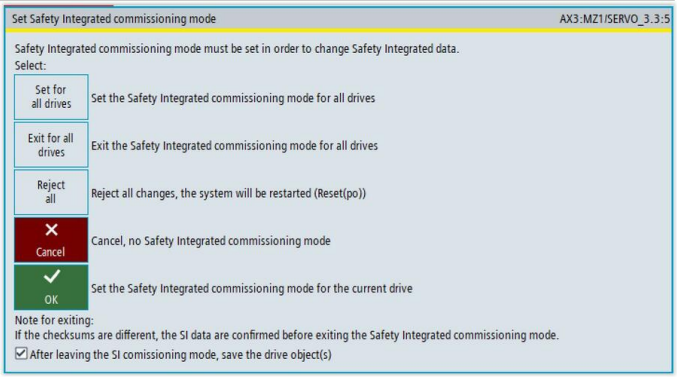
The training equipment ***Z axis* is fitted with a brake**, therefore SBC can be enabled.

- Disable the drives.
- Select the ***Z axis servo***







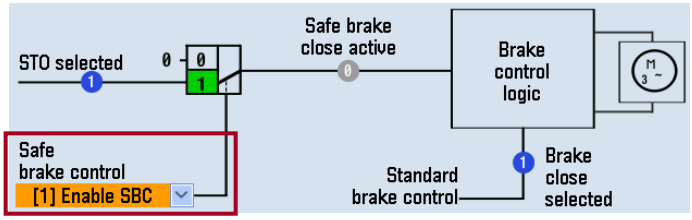


Only one drive is being changed, therefore select OK.



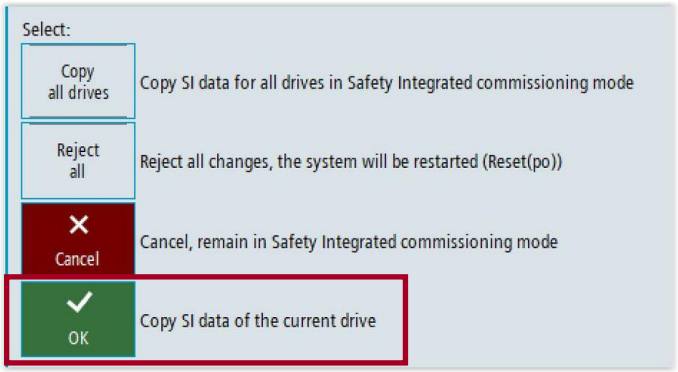
Notes:

Enable the SBC:



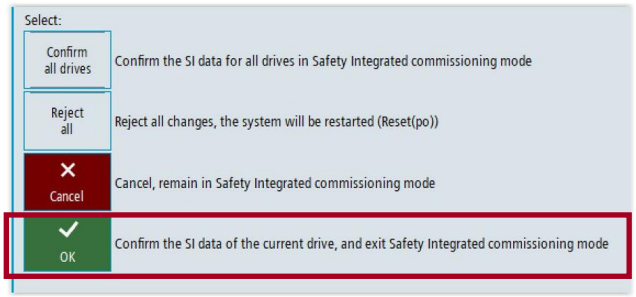
OK

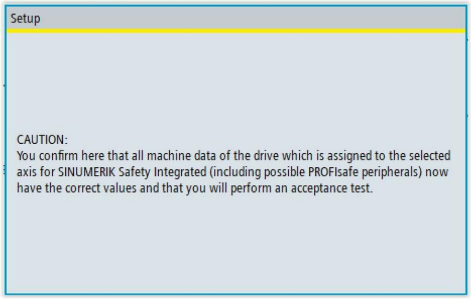
The changes need to be copied:



OK

The changes need to be confirmed





OK

Safe Brake Control is now enabled.

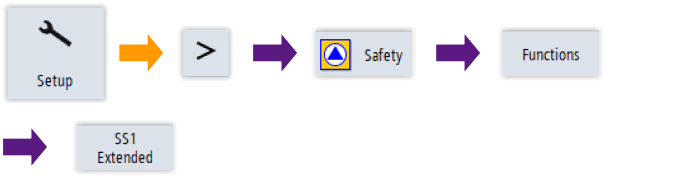
The Safe Brake Test is described in the ***Test Stop*** section

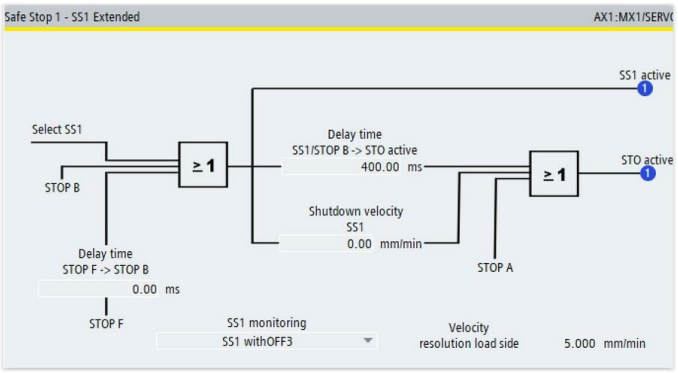
Notes:

SS1 Extended (Safe Stop 1 - Extended)

The **SS1 Extended** parameters need to be adjusted to ensure the correct transition between an active STOP/SS1 and STO.

Example: When SS1 is activated It is important to allow the drives to decelerate under current limit, to standstill, before applying Safe Torque Off. Axes with a large mass, e.g. a large chuck on a turning machine, may need several seconds to reach standstill from maximum speed.





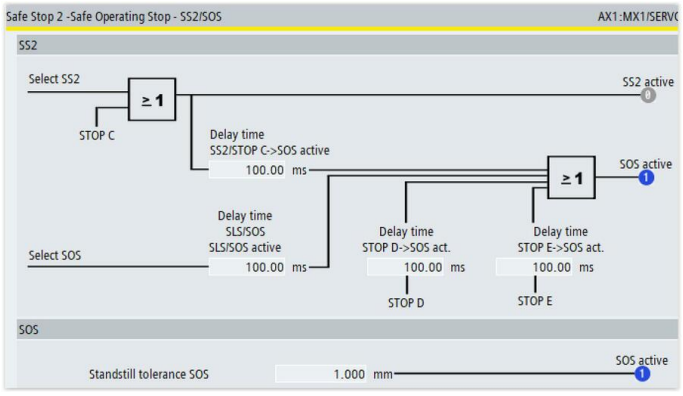
Parameter	Description
Delay time: STOP F to STOP B	Transition time from STOP F to STOP B (ms)
Delay time: SS1/STOP B to STO **E.g. for axes 500ms** **For spindles 5000ms**	Transition time from SS1/STOP B to STO (ms). Axes with long deceleration times need to come to a halt using current limit before STO is applied.
Shutdown velocity	A velocity can be set, beneath which the axis is assumed to be at standstill and STO is then applied. Note: Only if **SS1 With OFF3** is selected
SS1 monitoring	**SS1 With OFF3:** Safe Stop 1 in accordance with EN60204, stop category 1. **SS1E external stop:** Safe Stop 1 with external stop. This requires an externally initiated stop in order to be in compliance with stop category 1.

SS2/SOS (Safe Stop 2/Safe Operating Stop)

The **SS2** function results in an SOS condition. This is not suitable for an Emergency Stop situation.

The transition time, from an **SOS** request, or from STOPs C, D or E, can be adjusted, depending on the application.





Parameter	Description
Delay time: STOP C to SOS	Transition time from STOP to SOS (ms)
Delay time: SOS request to SOS active	Transition time from SOS request to SOS active. Training equipment: when the door is opened, SOS is requested on X, Y, Z & A axes.
Delay time: STOP D to SOS active **If errors occur regarding STOPD then this time should be increased**	Transition time from STOP D request to SOS active.
Delay time: STOP E to SOS active	Transition time from STOP E request to SOS active.
Standstill tolerance SOS	Determines how far an axis can move, when SOS is active, before the system triggers an alarm.

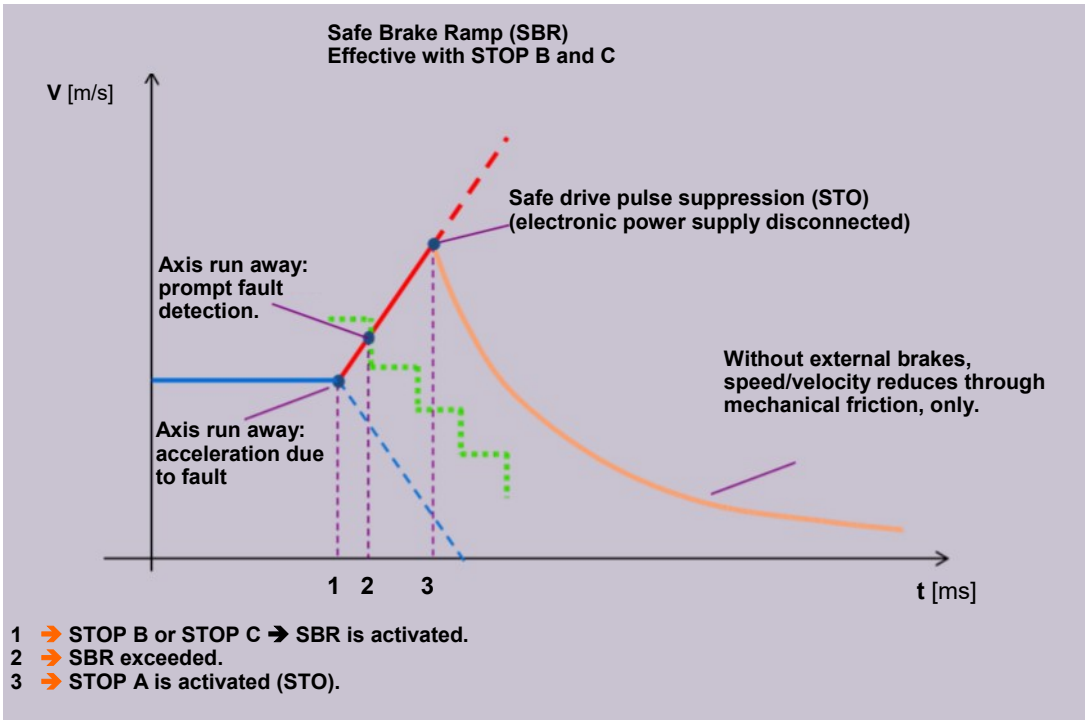
Notes:

Safe Acceleration Monitoring/Safe Braking Ramp

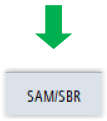
The **Safe Acceleration Monitoring (SAM)** function is responsible for monitoring the braking along **OFF3** ramp.

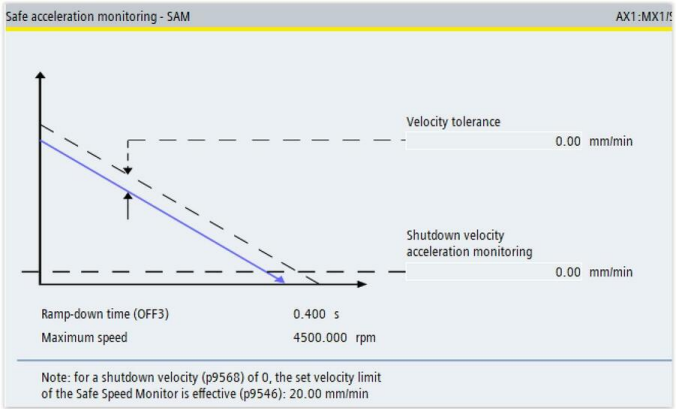
This was the default choice during the enabling of Safety Integrated on the drives.

The function is active for SS1, SS2, STOP B & STOP C.
If the motor does not decelerate, as expected, the system selects STO which results in the motor coasting to a standstill.



Setting the SAM:





Parameter	Description
Velocity tolerance (p9548)	The deviation from the contour for an axis undergoing an ***OFF3*** ramp down.
Shutdown velocity acceleration monitoring. (p9568)	During ramp down, the level at which the SAM ceases to be active. Note: If set to 0, the value in the Safe Speed Monitoring will be used.

Calculating the Safe Acceleration Monitoring:

SAM = Acceleration * Safe Clock Cycle

Linear Axes:
SBR[mm/min] = a[m/s^2] * t_{safe} * 1000[mm/min] * 60[s/min]

Rotary Axes:
SBR[rev/min] = a[rev/s^2] * t_{safe} * 60[s/min]

E.g. Calculation of the X axis SBR:

Linear Axes Formula:
SBR[mm/min] = a[m/s^2] * t_{safe} * 1000[mm/min] * 60[s/min]

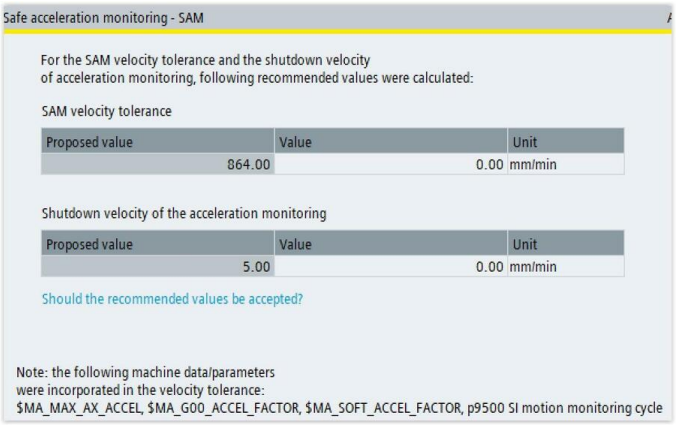
MD p9500 SI Motion monitoring clock cycle = 0.012s
MD 32300 $MA_MAX_AX_ACCEL = m/s^2 = 1

SBR Tol = (1 * 0.012 * (60*1000)) + 20% = **864 mm/min**

Notes:

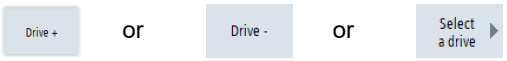
The SAM can be automatically calculated for linear and rotary axes, but ***not*** for the **spindle**:



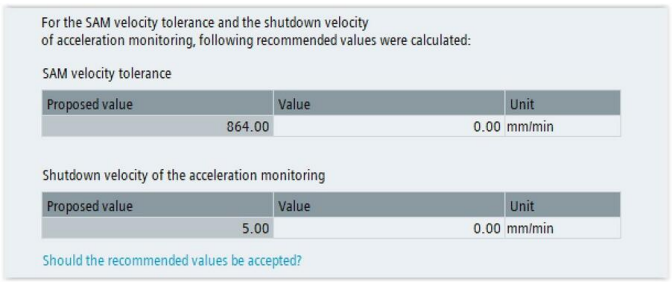


The proposed values can be accepted or adjusted if necessary.

Now repeat for all of the drives



Example SAM calculations for the X, Y & Z axes:



Spindle SAM calculation.

Description:
Following a SI stop request (STOP B/STOP C) the motor speed must be reduced.
If this fails to happen and the **SBR** setting is exceeded, the system issues a **STOP A** (STO - drive pulse suppression).
For axes without external brakes the stopping of the motors will be dependent on mechanical friction.

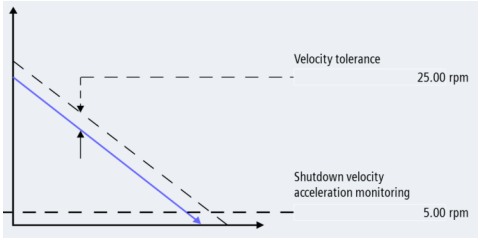
The system can calculate the SBR tolerance for axes, via a soft -key. The spindle, however, needs to be manually calculated using the following formula:

SBR[rev/min] = a[rev/s^2] * t_{safe} * 60[s/min] * 1.2

a[rev/s^2] = MD 32300 $MA_MAX_AX_ACCEL = 30m/s^2

t_{safe} = MD p9500 SI Motion monitoring clock cycle = 0.012s.

Therefore: 30 * 0.012 * 60 * 1.2 = 25.92
This needs to be manually entered as the SAM value for the spindle:

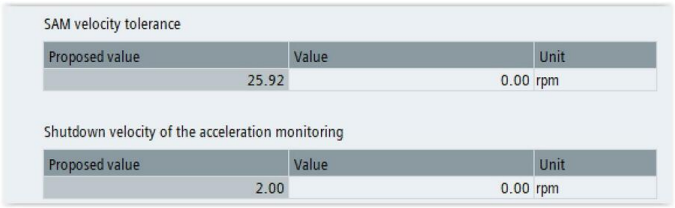


Example manually calculated SAM values entered for the Spindle:

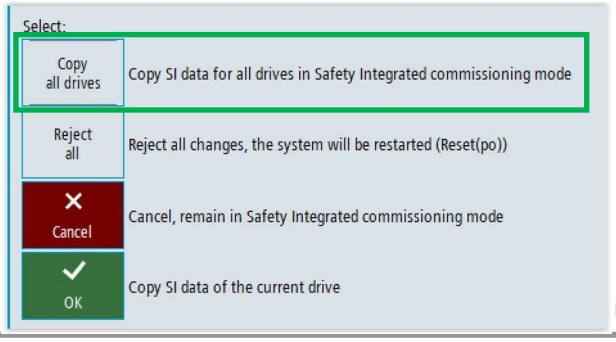
Also, depending upon the machine configuration, it may be necessary to consider other machine data:

MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL
MD 35210: GEAR_STEP_POSCTRL_ACCEL
MD 35410: SPIND_OSCILL_ACCEL

A Axis: example SAM calculations for the A axis:



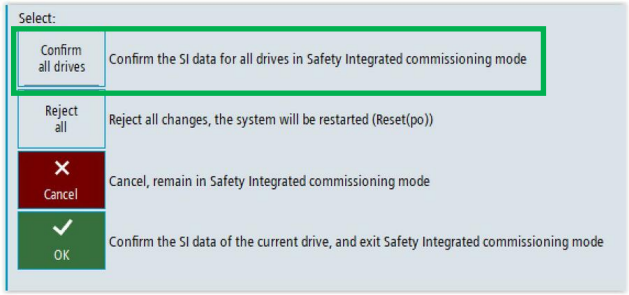
The changes need to be copied:

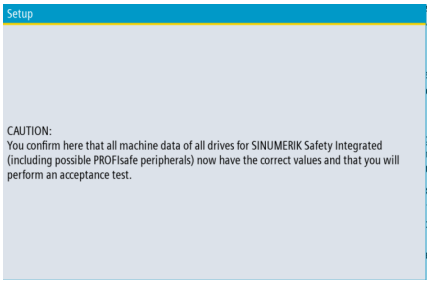


Notes:



The changes need to be confirmed





If alarm 22001 is displayed, the online help shows the possible causes of this. The alarm means that with the current MD settings the axis can not follow the SAM settings

22001 [Channel %1:] Block %2 Axis %3: Brake ramp longer than Stop D time. Reason: %4.

Parameters: %1 = Channel number
%2 = Block number, label
%3 = Axis name
%4 = Identification of cause

Explanation: The current axis dynamic response is not sufficent to come to a standstill on time when a Stop D is triggered. The reasons specified in parameter 4 are:
1: MD32300 $MA_MAX_AX_ACCEL too small.
2. MD32431 $MA_MAX_AX_JERK too small.
3. Excessive acceleration reduction activated with ACC or subsequent to FXST.
4. Excessive jerk reduction programmed with JERKLIMA.

Reaction: Alarm display.

Remedy: Increase MD36953 $MA_SAFE_STOP_SWITCH_TIME_D. Increase MD32300 $MA_MAX_AX_ACCEL and MD32431 $MA_MAX_AX_JERK. Increase programmed acceleration (ACC) or jerk (JERKLIMA). Correct torque and acceleration reduction subsequent to FXST.
Alarm can be suppressed with MD11415 $MN_SUPPRESS_ALARM_MASK_2 bit 13.

Programm continuation: Clear alarm with the Delete key or NC START.

If the axis/spindle has been fully commissioned then if the alarms occur p 9553 should be increased under
SS2/SOS (Safe Stop 2/Safe Operating Stop)
See P84

Safely Limited Speeds

The Safely Limited Speed (**SLS)** function is used to protect the drives against unintentional high speeds in both directions.

There are four available SLS levels for each axis.

The system monitors the active SLS level and causes a STOP condition should it be exceeded.

It is possible to choose the STOP type when an SLS level is exceeded.

By default, all the axes have each SLS level set to 2000mm/min, for linear axes, and 2000 rpm, for rotary axes.

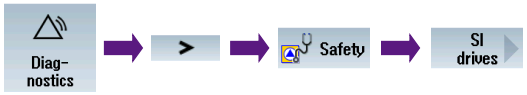
The active SLS level is controlled by the door and mode switches in accordance to the OEM function table

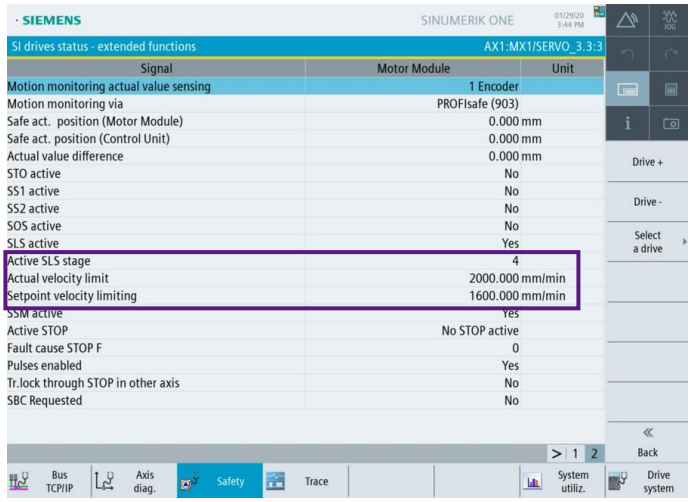
Function table example showing the SLS/switch relationship for the training equipment:

Operating mode	*Door*	*Axes / spindle*	*SLS level*
Auto	Closed	Axes	SLS 4
		Spindle	SLS 4
	Open	Axes	SOS
		Spindle	STO
Setup	Closed	Axes	SLS 4
		Spindle	SLS 4
	Open	Axes	SOS or SLS1 With Hold-To-Run PB
		Spindle	STO

Notes:

The current **Safe actual velocity limit** and the **Safe velocity limitation** can be seen on the diagnostics screen.
SLS is active with SLS stage 4 selected and the SLS speed is the default value

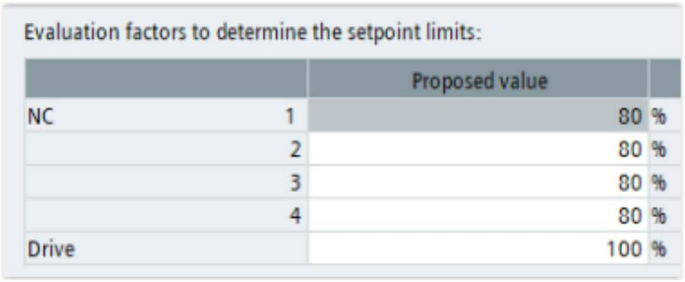




The **Safe actual velocity limit** is as the default SLS setting.

The **Set velocity limitation** (1600.000mm/min) is due to the value that was set using the recommended settings and the default SLS setting (2000.000mm/min)
i.e. **80%** of **Safe actual velocity limit** =
80% of 2000.000 = 1600.000

Below, the previously calculated set-point limitation:



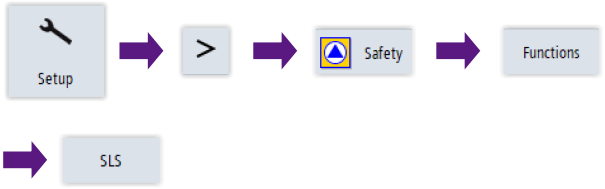
Calculating the required SLS levels:

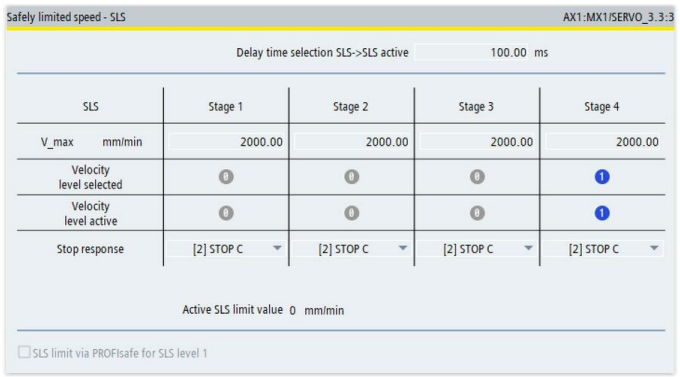
Example: The OEM has stated the following feed requirements:

	Set-up/door open/ hold to run		Auto/ door closed	
	Maximum Feed-rate	**SLS1**	Maximum Feed-rate	**SLS4**
X, Y, Z (mm/min)	2000	**2500**	10000	**12500**
A (rpm)	10	**12.5**	100	**125**
Spindle (rpm)			1000	**1250**

The SLS levels have been calculated using the 80% limitation factor (feed-rate / 0.8):

The calculated values need to be entered via the SLS set-up screen:

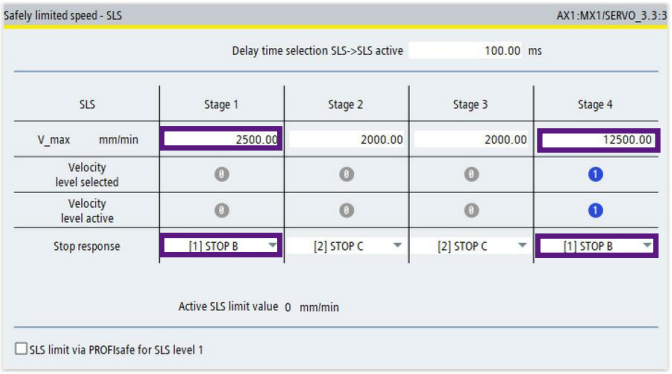




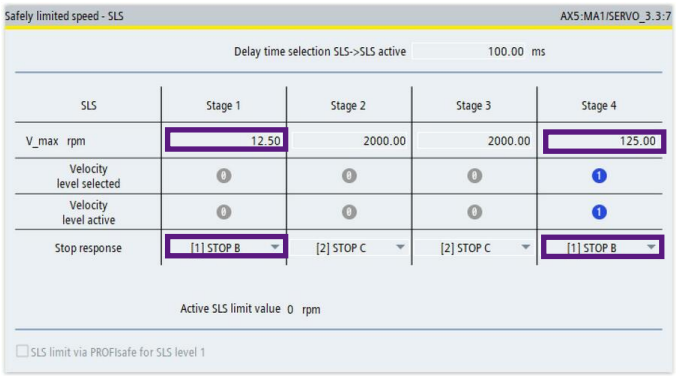


Notes:

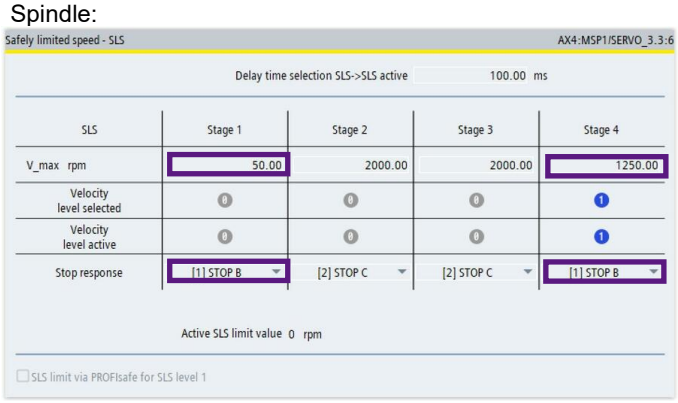
X, Y & Z axes:



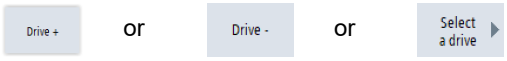
A axis



Spindle:



Now repeat for all of the drives





Start the procedure for copy/confirm



Copy the SI data of all drives



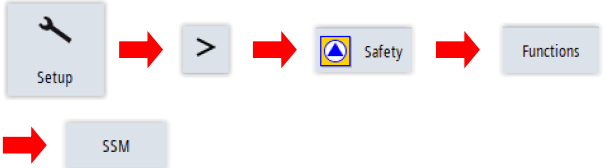
Confirm the SI data of all drives

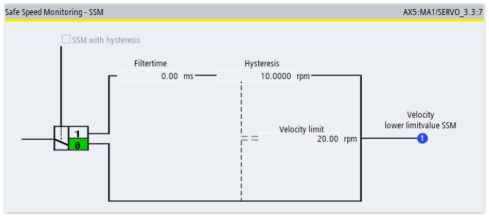
Safe Speed Monitoring (SSM)

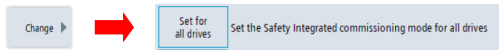
Safe Speed Monitoring is used to set a velocity limit, below which the axis is considered at standstill.
This can then be used to set an output, for example, **to release a door lock.**

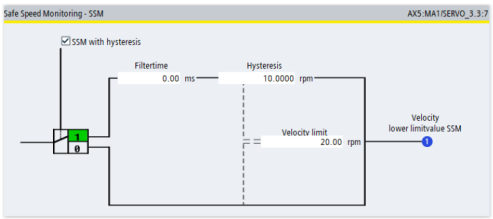
Parameter Description:

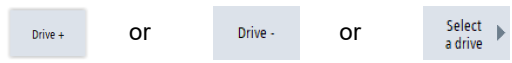
Parameter	Description
SSM with hysteresis (p9501.16)	Enables the SSM function.
Filter time (ms) (p9545)	Sets a filter time for SSM to detect standstill.
Hysteresis (mm/min or rpm) (p9545)	Sets the speed hysteresis SSM to detect standstill.
Speed limit (mm/min or rpm) (p9546)	Sets the speed limit for SSM to detect standstill.













Start the procedure for copy/confirm



Copy the SI data of all drives



Confirm the SI data of all drives

Notes:

SBT - Safe Brake Test

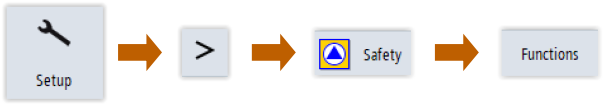
The Safe Brake Test is designed to test the functionality of the brake.

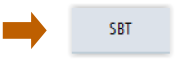
It is possible to test up to two brakes per axis.
There are two brake test parameter sets available for each brake.

During the test the brake is closed and the motor is given a command to move and the axis is monitored for actual movement.
Too much movement indicates a worn/faulty brake and the system will issue an alarm.

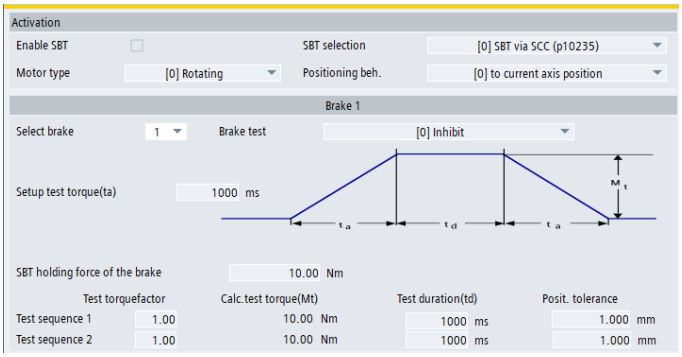
The test is initiated and monitored in the user PLC via the SIC/ SCC interfaces.

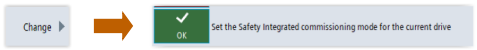
Note: The safe brake test should be initiated immediately before the running of the test stop routine.
This is because the test stop routine removes the drive pulses as part of the checking procedure and could result in the axis falling, were the brake to be faulty.
In the event of a faulty brake being detected, the test stop should be prevented from running until the faulty brake has been dealt with.

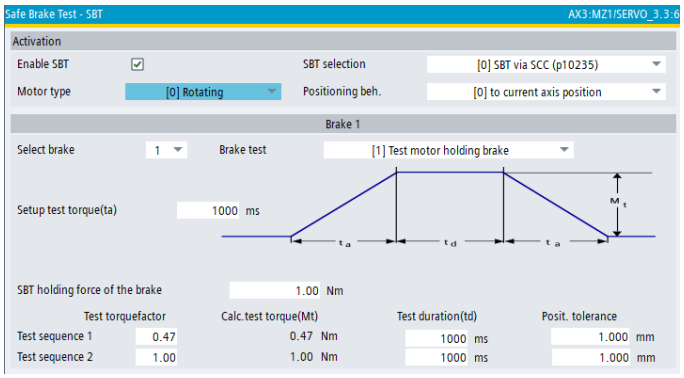




Select the Z axis drive because the servo motor has a brake







Example - basic settings for the training equipment:

- Set ***Enable SBT*** tick-box
- Select the ***Motor type*** to ***[0] Rotating.***
- Select the ***SBT selection*** to ***[0]SBT via SCC (p10235)***
- Select the ***Positioning beh.*** to ***[0] to current axis position***

Description of the Brake Test Parameters:

Parameter	Description
Select brake (p10202)	Selection of the brake to be tested. • 1: Brake 1 (p10202[0]) • 2: Brake 2 (p10202[1])
Brake test (p10202)	Defines whether and which brake should be tested • 0: Inhibit • 1: Test motor holding brake • 2: Test external brake
Setup test torque(ta) (p10208)	Torque ramp up/down time during the brake test
SBT Holding force of the brake (p10209)	A value should be entered based upon the holding capabilities of the brake. This value is used in the calculation of the actual torque used in the brake test.
Test torquefactor (p10210)	The actual torque used in the brake test. is the result of: **Holding force x Test torquefactor.** The **Test torquefactor** is calculated from the holding torque of the axis.

Notes:

Parameter	Description
Calc.test.torque(mt)	Value calculated from “SBT holding torque of the brake” and “Test torque factor”
Test duration(td) (p10211)	The duration of the SBT, excluding torque ramp up/down.
Posit tolerance (p10212)	Maximum distance the axis can move, during the test, without an alarm being triggered.

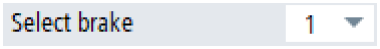
Setting the Brake Test Parameters:

The brake test parameters must be set so that the test can be completed successfully.

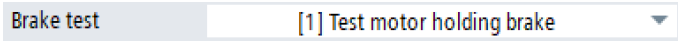
Selecting the brake to be tested:

The training equipment has a motor brake on the Z axis, therefore, only one set of brake parameters needs to be set. The ***Brake 1*** parameter set will therefore be used for this example:

- Select the ***Select brake*** option to ***1***:



- Select the ***Brake test [1] Test motor holding brake***

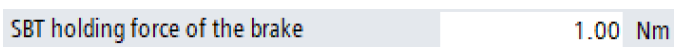


Setting the holding torque:

This should be entered based upon the holding capabilities of the brake.
This value is then used in a calculation to determine the actual torque to be used in the brake test:
The following table shows the motor holding brake capabilities for he 1FK7 motor range.

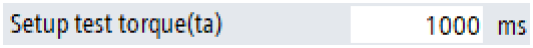
Motor Type	Brake Holding Torque (Nm)
1FK701_	0.4
1FK702_	1
1FK703_	1.9
1FK704_	4
1FK706_	13
1FK708_	22
1FK7100	23
1FK7101	43
1FK7103	
1FK7105	

The training equipment Z axis is fitted with a **1FK702** type motor, therefore set the Holding torque to 1Nm:



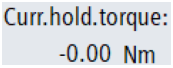
Setting the Setup test torque:

The default ramp up/ramp down time (1000ms), of the brake test torque, can be used for the training:



Test torque factor.

The ***Test torque factor*** is calculated using the ***current holding torque (p1509)*** of the axis.
This can be seen on the Safe Brake Test screen:



Note: on an actual machine there would be value present.
The direction of the ***current holding torque*** is dependant on the machine configuration.

Notes:

Note: the training equipment has no vertical axis, therefore, there the ***Curr. Holding torque*** will be **0.**
Therefore, for the purpose of the training, an assumed value will be used to calculate a realistic ***Test torque factor***

Test torque factor calculation:

The following calculations can then be used to determine the ***Test torquefactor and SBT testtorque***

1. ***Test torque factor*** **=**
   (Curr. Holding torque * 1.3/Brake holding torque)
   = (r1509 * 1.3 / p10209)

2. ***SBT testtorque*** **=**
   Brake holding torque * Test torque factor
   = p10209 * p10210

Note: the calculation results an **additional 30% torque** being applied to the brake during the brake test.
This is designed to test the brake beyond the normal holding torque of the brake.

Training equipment example:

Note: an assumed value of ***0.36Nm*** is being used as the ***Curr. Holding torque (r1509).***

The ***Test torquefactor*** can then be calculated as:

= (0.36 * 1.3 / 1) = 0.47

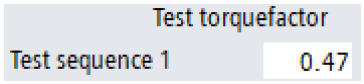
Calculation of the ***SBT testtorque*:**

The torque to be used during the SBT (0,47Nm) is set using:

Brake holding torque * Test torque factor
1 * 0.47 = 0.47 Nm

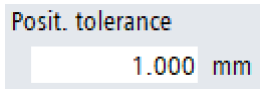
The ***SBT testtorque*** can be seen in the display

Therefore, the ***Test torquefactor*** can be set to 0.47



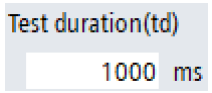
Position tolerance:

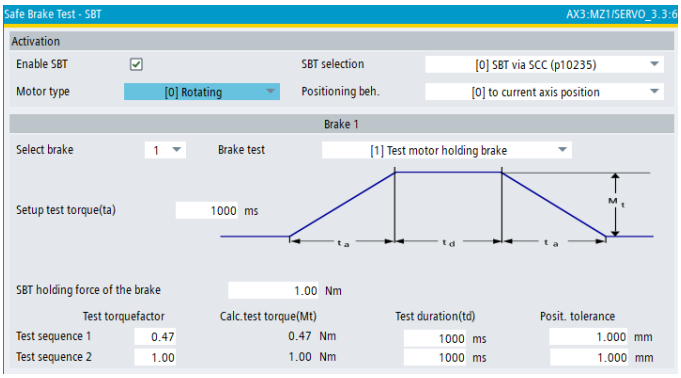
The default position tolerance(1 mm) can be used for the training:

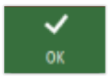


Test duration:

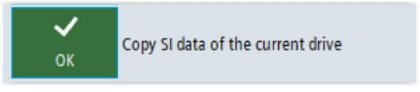
The default test duration(1000 ms) can be used for the training:

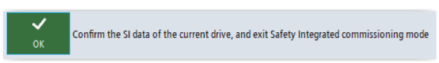






Carry out the copy and confirm procedure for the relevant drive





Notes:

Initiating the Brake Test:

The user PLC interface is used to initiate and start the Safe Brake Test.
Each axis has bytes in the axis interface data block (DB31...DB61) reserved for safety integrated signals to/from the PLC/Drive.

Signals from the PLC to the drive (Safety Command Channel):

DB31 ...DB61	Signals from PLC to Drive (Safety Command Channel—SCC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB143			Brake Closed	Torque 1 or 2	Direction of Rotation	Brake 1 or 2	BT Start	Selection of SBT

Signals from the drive to the PLC (Safety Information Channel):

DB31 ...DB61	Signals from Drive to PLC (Safety Information Channel—SIC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB173	Sign Load Torque	Brake Req.	B test finished	B test OK	B test Active	Brake 1 or 2	B test Start	Selection Safe BT

Signals from the PLC to the drive (Safety Command Channel):

Bit 0 - Selection of Safe Brake Test: Activate the drive in readiness for the Safe Brake Test.

Bit 1 - Safe Brake Test start - Start the Safe Brake Test

Bit 2 - Selection of brake 1 or 2 - set to 0 for brake 1, set to 1 for brake 2.

Bit 3 - Direction of rotation - set to 0 for positive, set to 1 for negative.

Bit 4 - Torque 1 or 2 - set to 0 for Test torquefactor 1, set to 1 for Test torquefactor 2.

Bit 5 - Brake closed - Set to 1 to confirm that the brake is closed (External brakes only)

Signals from the drive to the PLC (Safety Integrated Channel):

Bit 0 - Selection of Safe Brake Test: Safe Brake Test is active.

Bit 1 - Safe Brake Test start - Safe Brake Test has started.

Bit 2 - Brake 1 or 2 active - 0 = brake 1, 1 = brake 2.

Bit 3 - Brake test is active - 1 = brake test active.

Bit 4 - Brake test ok - 1 = brake test was successful.

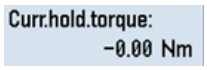
Bit 5 - Brake test finished - brake test has finished.

Bit 6 - Brake test required - brake test is required.

Bit 7 - Load torque sign - direction of the load torque, 0 = positive, 1 = negative. Can be used to determine the SBT direction.

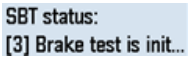
Notes:

Due to the training equipment not having a holding torque, the ***Curr.hold torque*** display is 0.00Nm.



Note: on an actual machine there would be value present. The direction of the ***current holding torque*** is important because the brake test direction should be in the same direction as the force due to the weight of the vertical axis i.e. opposite to the ***current holding torque.***
The direction of the SBT on the training equipment is not important.

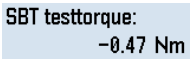
- Enable the drives and set the feedrate to 100%
- Set the ***Select SBT*** interface bit for Z axis (DB33.DBX143.0). The ***SBT status*** will change to 3.



Interface signals, DB33.DBX173.0 & DB33.DBX173.1, will change to 1.

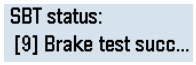
- Set the ***Start SBT*** interface bit for Z axis (DB33.DBX143.1)
The test will start; the axis will be driven against the closed brake using the specified SBT testtorque.

The ***SBT testtorque*** will be displayed on the screen:



The During the test, interface signal, DB33.DBX173.3 will be set to 1.
On successful completion of the test, DB33.DBX173.4 & DB33.DBX173.5, will be set to 1.

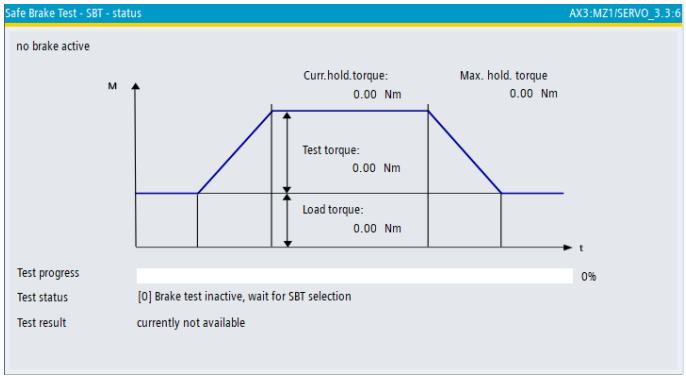
A successful test is indicated in the ***SBT status*** area:



Trace Feature:

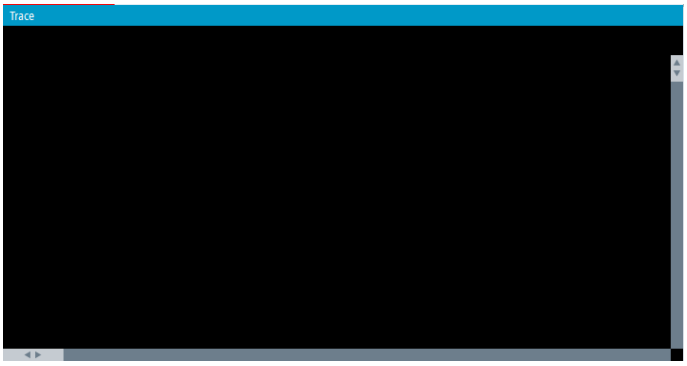
The trace feature can be used to view detailed information.



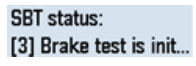




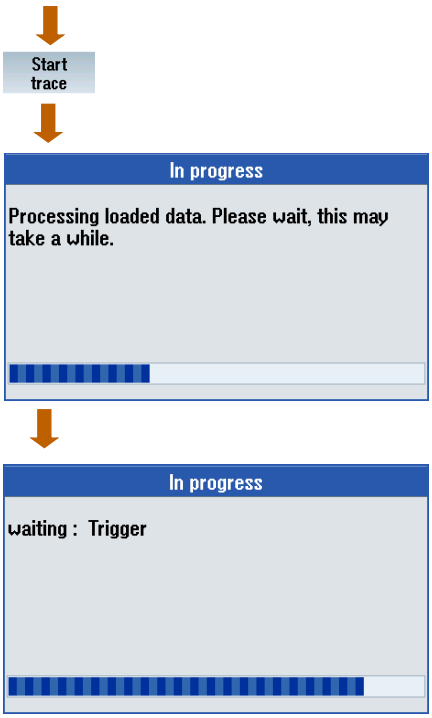
Wait until following screen is displayed:



- Initialise the Safe Brake Test:

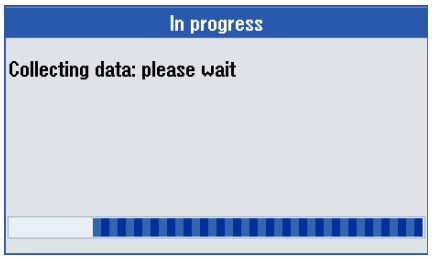


Notes:

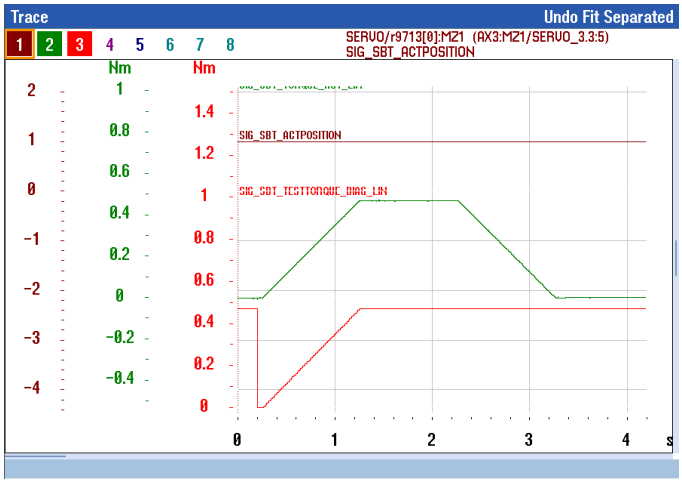


- Start the Safe Brake Test

The trace will be started:



The trace will be displayed automatically:



Green Trace : SBT actual torque.
Red Trace: SBT test torque.
Brown Trace: Actual position.

The traces show that the SBT was in the positive direction and that torque values are the same as was calculated (0.47Nm).

Notes:

1: The green trace had an initial value of **0**.
On an actual machine this would not be the case due to the weight of the vertical axis.

2: The red trace shows that **SBT test torque** had a value of **–0.47 Nm**, prior to the test beginning.
This is a residual value due to the brake having already been tested.
The **SBT test torque** is set to 0 at the beginning of each test.
On power up and before a brake test has been carried out, the **SBT test torque** will have a value of 0.

Note: The safe brake test should be initiated immediately before the running of the test stop routine.
This is because the test stop routine removes the drive pulses as part of the checking procedure and could result in the axis falling, were the brake to be faulty.
In the event of a faulty brake being detected, the test stop routine should be prevented from running until the faulty brake has been dealt with.

Notes:

Test Stop Routine:

NOTE: Any PLC code shown in this document is to be treated as an example and for guidance purposes only!

Test Stop (forced checking procedure) Description:

Functions and switch-off signal paths must be regularly tested using the forced checking procedure (test stop). The forced checking procedure should be performed each time that the control system runs up and must be performed at least once a year. This involves components from the SINUMERIK ONE / SINAMICS S120 system.

The forced checking procedure is not performed in the safety program but in the standard PLC program. This means that you insert the affected program code in your PLC user program in order to trigger the forced check procedure during run-up of the controller.

Safe Brake Test Priority:

Because the Test Stop routine results in the drive pulses being disabled, it is necessary to carry out the Safe Brake Test immediately before the TEST Stop is started.

PLC interface signals are available to report the result of the Safe Brake Test.
These can be used to initiate/inhibit the Test Stop, depending on the result of the Safe Brake Test.

The Test Stop routine is started using the Test Stop user Key on the machine control panel.

The forced checking procedure must be initiated by the user or integrated as an automatic procedure.
For example:

- When the axes are stationary after the system has been powered--up
- When the protective door is opened
- In defined cycles (e.g. every 8 hours) The maximum permissible is 1 year).
- In the automatic mode, dependent on the time and event.

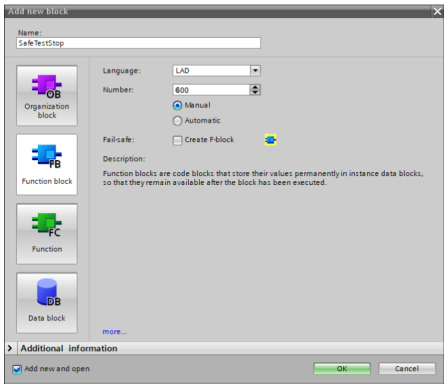
DB31 ...DB61	Signals from PLC to Drive (Safety Command Channel—SCC)
Byte	Bit 0
DBB140	Start Test Stop

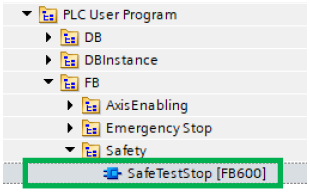
DB31 ...DB61	Signals from Drive to PLC (Safety Information Channel—SIC)	
Byte	Bit 5	Bit 4
DBB166	Test Stop Required	Test Stop Active

Additional Interface signals required for Test Stop

Automating the Safe Brake Test and the Test Stop.

The Safe Brake Test and Test Stop logic is added to an FB in the non safety user PLC. In this example a new FB called SafeTestStop and numbered FB600 is added.





The following Input variable is required

SafeTestStop

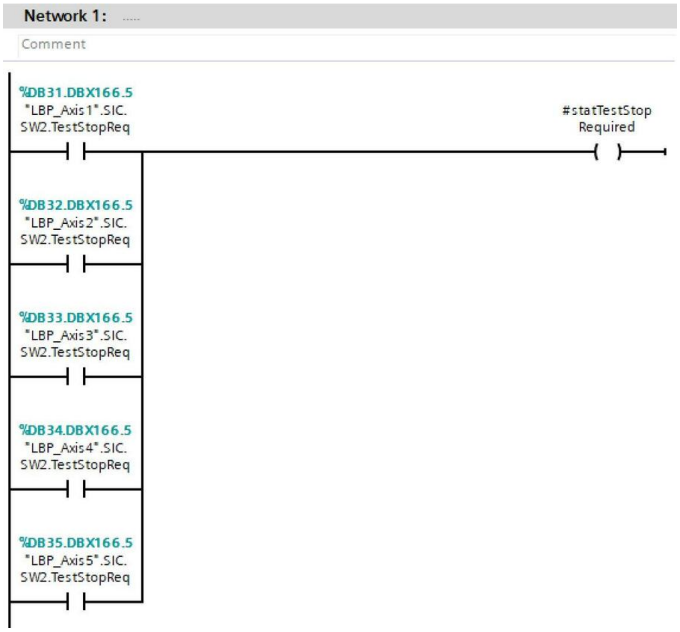
	Name	Data type
1	Input	
2	K1/K2Feedback	Bool

The following static variables are required

8	Static	
9	statTestStopRequired	Bool
10	statAllAxesandSpindleSOS&DoorClosed	Bool
11	statReadyForTestStop	Bool
12	statSafeBrakeTestPrepare	Bool
13	statSafeBrakeTestStart	Bool
14	statSafeBrakeTestOK	Bool
15	statStartTestStop	Bool
16	statStartTestStopPulse	Bool
17	statK1RelayTest	Bool
18	statK1FeedbackFault	Bool
19	statK2RelayTest	Bool
20	statK2FeedbackFault	Bool
21	statTestStopBusy	Bool

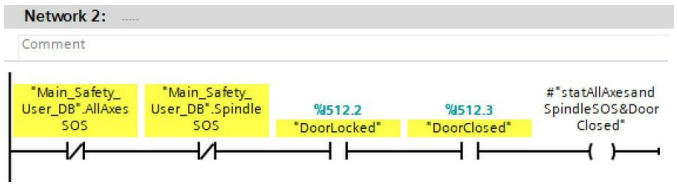
Notes:

When a Test Stop is required, by a drive, an interface signal from the axis interface is set to 1.

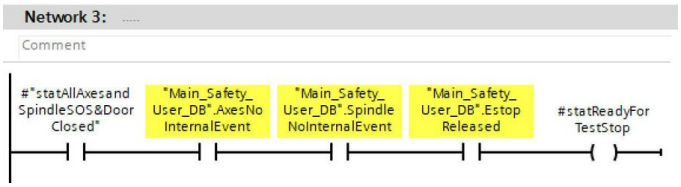


The Test Stop Request signal can be used to first start the brake test, and then, assuming that the brake test is successful, initiate the Test Stop.

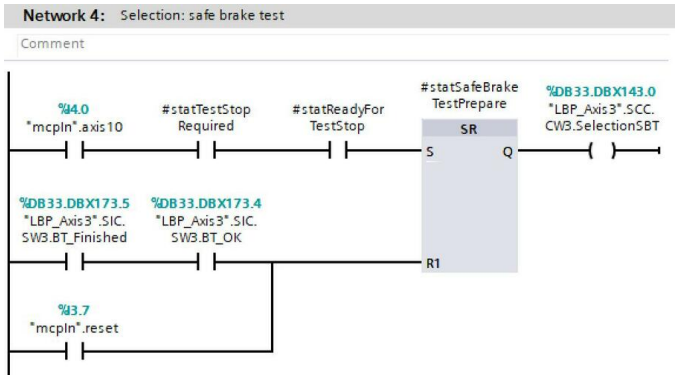
Example: monitoring for no SOS active and that the door is closed and locked.



Example: monitoring for no SOS active and that the door is closed and locked., no internal event and no emergency stop.

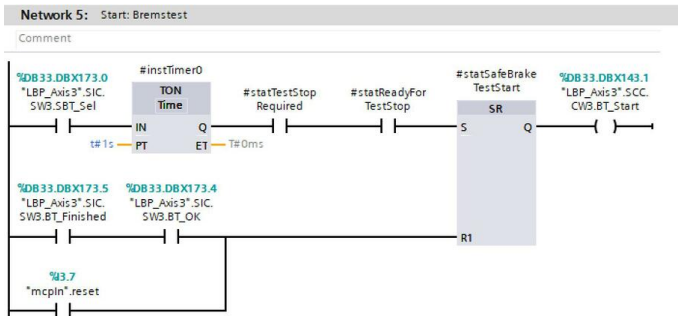


For the purpose of the training the Test stop User key is used to start the Brake Test providing that the conditions are met.



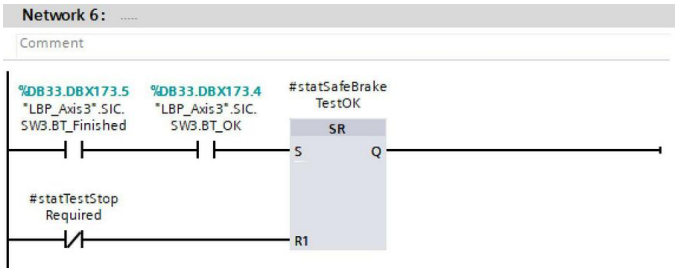
The Brake Test selected signal is used to start the Brake Test sequence.

Note: when selecting IEC timers an Instance Data Block is required. If several IEC timers are to be used it is best to use a Multi Instance DB so that just one DB is used rather than an instance DB for each timer.

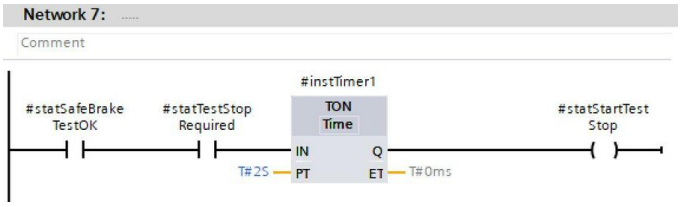


Notes:

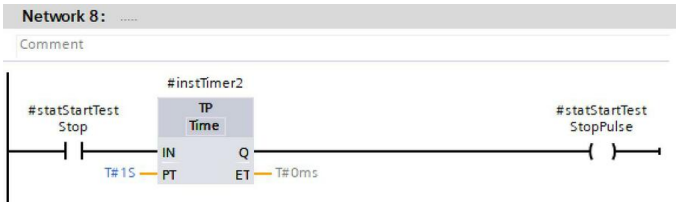
If the Brake Test is successful the start Test Stop sequence is initiated.



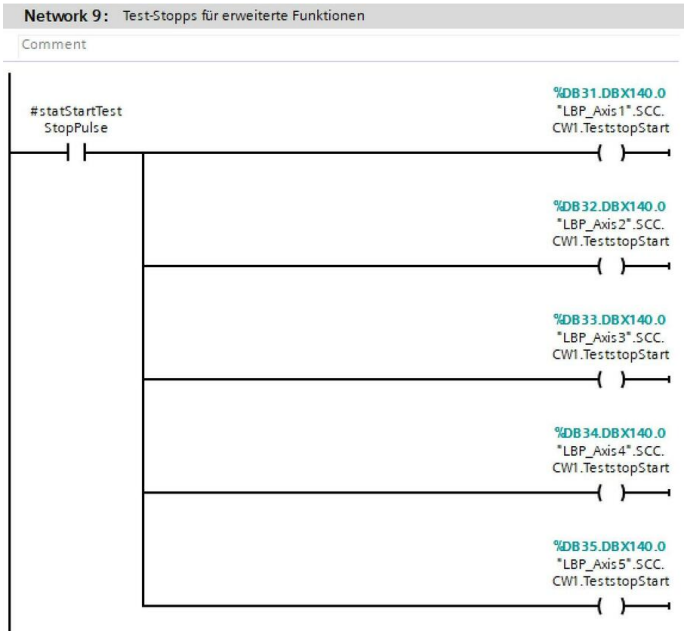
A 2 second delay allows the Safe Brake Test signals to normalise before starting the Test Stop.



The Test Stop signal will remain high for 1 second to allow the pulses to be disabled/enabled.



The Test Stop start signal is set for each axis via the axis DBs.



The disabling/enabling of the pulses can be seen in the axes diagnostics area:

Service overview

Select: All NC axes and drives

	AX1	AX2	AX3	AX4	AX5
On/Off1	●	●	●	●	●
Off2	✓	✓	✓	✓	✓
Off3	✓	✓	✓	✓	✓
From drive: Operation enabled	●	●	●	●	●
From power supply:Operation enabled	✓	✓	✓	✓	✓
Pulse enable	●	●	●	●	●
Speed controller enable NC	●	●	●	●	●
Pulses enabled	●	●	●	●	●
Drive ready	●	●	●	●	●
Heatsink temperature	✓	✓	✓	✓	✓
Power section in i^2t limitation	✓	✓	✓	✓	✓
Motor temperature	✓	✓	✓	✓	✓
Measuring system 1 active	●	●	●	●	●
Measuring system 2 active	●	●	●	●	●

- The Test Stop interval time will now be reset.

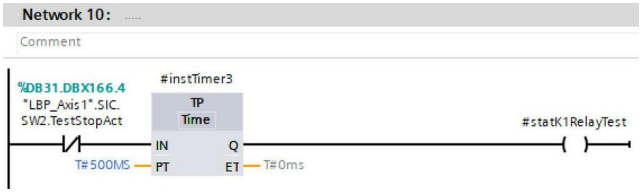
Notes:

Testing the K1 & K2 relays

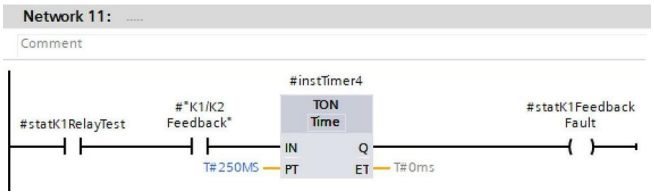
The correct operation of the K1 and K2 relays can also be tested during the Test Stop procedure.
The relays each have a normally closed contact connected in series to input I100.6
By momentarily switching the relays off/on the input can be monitored for the correct operation of the K1 and K2 relays.

Enter the example logic for testing the relays.
Start the K1 Relay Test after the Test Stop has finished:

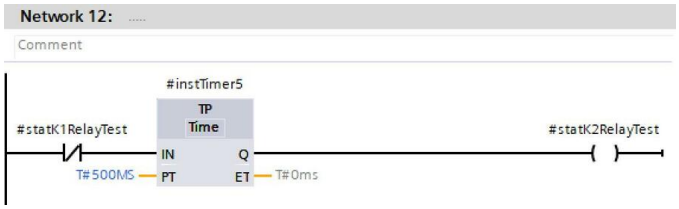
The “K1/K2RelayTest” signals are used in the Safety PLC.



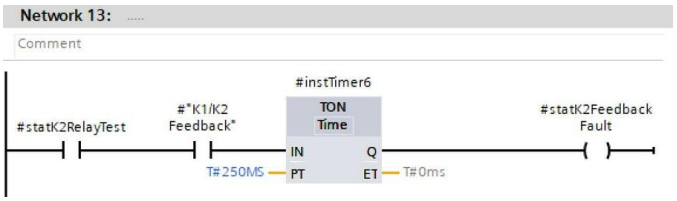
Monitor the K1 Relay feedback input switches within 250ms. Trigger a message if it fails.



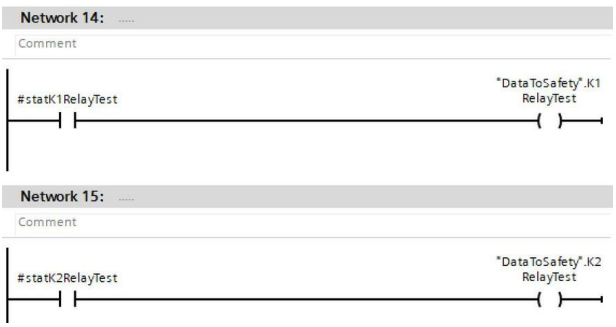
Start the K2 Relay Test once the K1 Relay Test has completed has finished:



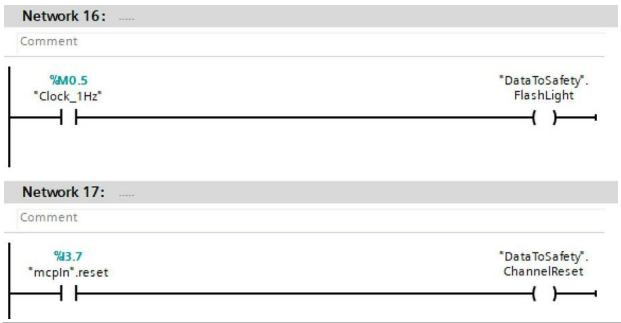
Monitor the K2 Relay feedback input switches within 250ms. Trigger a message if it fails:



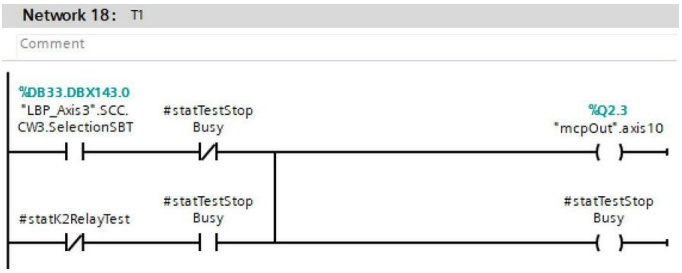
The K1/K2 Relay test signals are assigned to variables in the Global DB601 for use in the main safety FB1010. By momentarily switching the relays off/on the input can be monitored for the correct operation of the K1 and K2 relays.



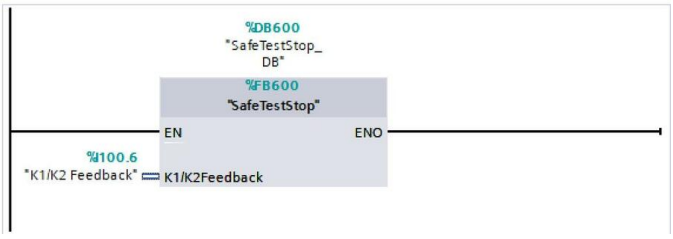
The 2Hz clock signal and MCP reset signal are also transferred to the Global DB601 for use in the main safety FB1010.



The “Test Stop” lamp on the MCP is on while the Brake Test and Test Stop routines are running.



FB600 should be called in OB1, in this example DB600 is used as the multi instance DB



Notes:

Notes:

Appendix 1 Safety Simulator Drawings

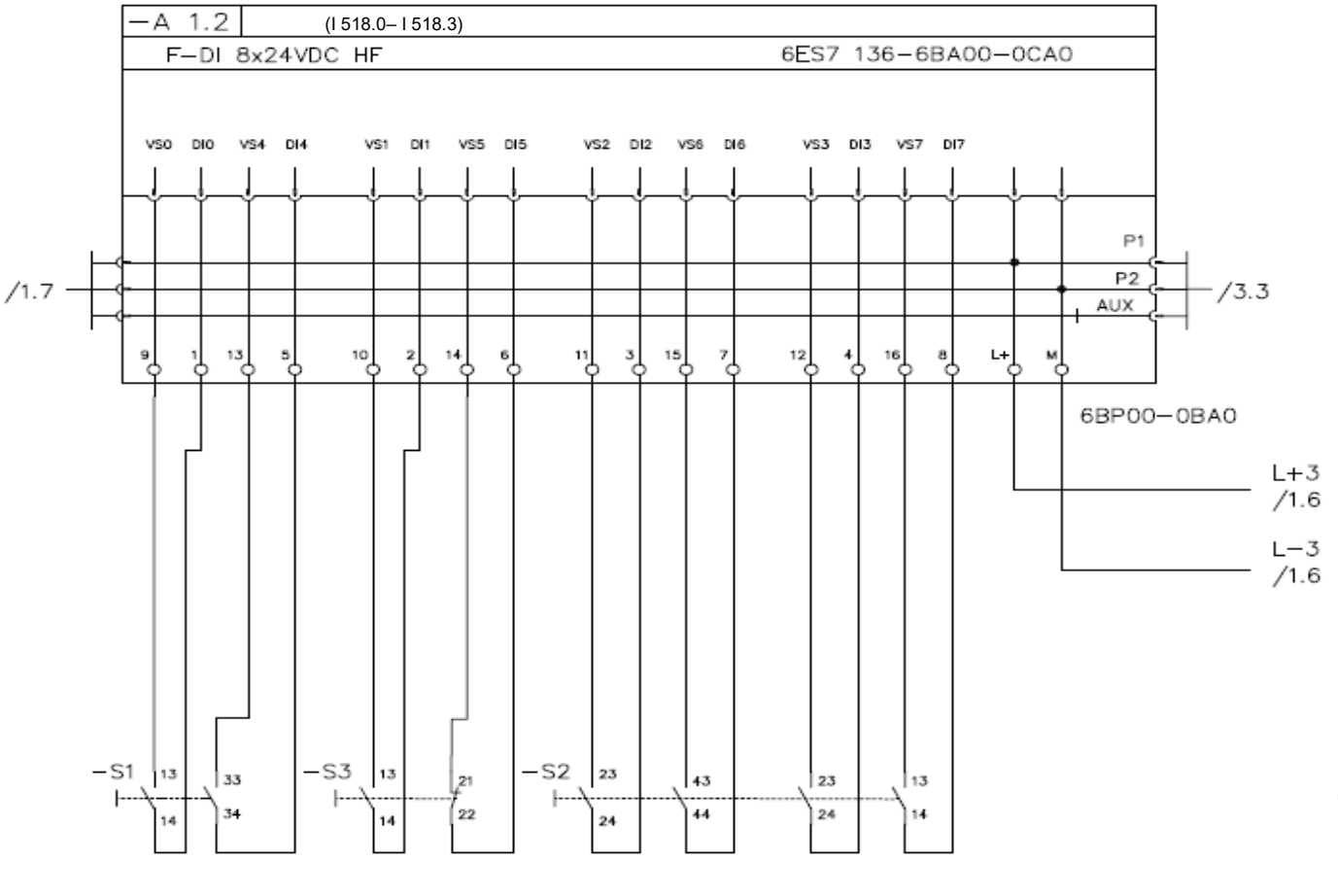
(I 512.0– I 512.3)

Locked Closed

Emergency Stop

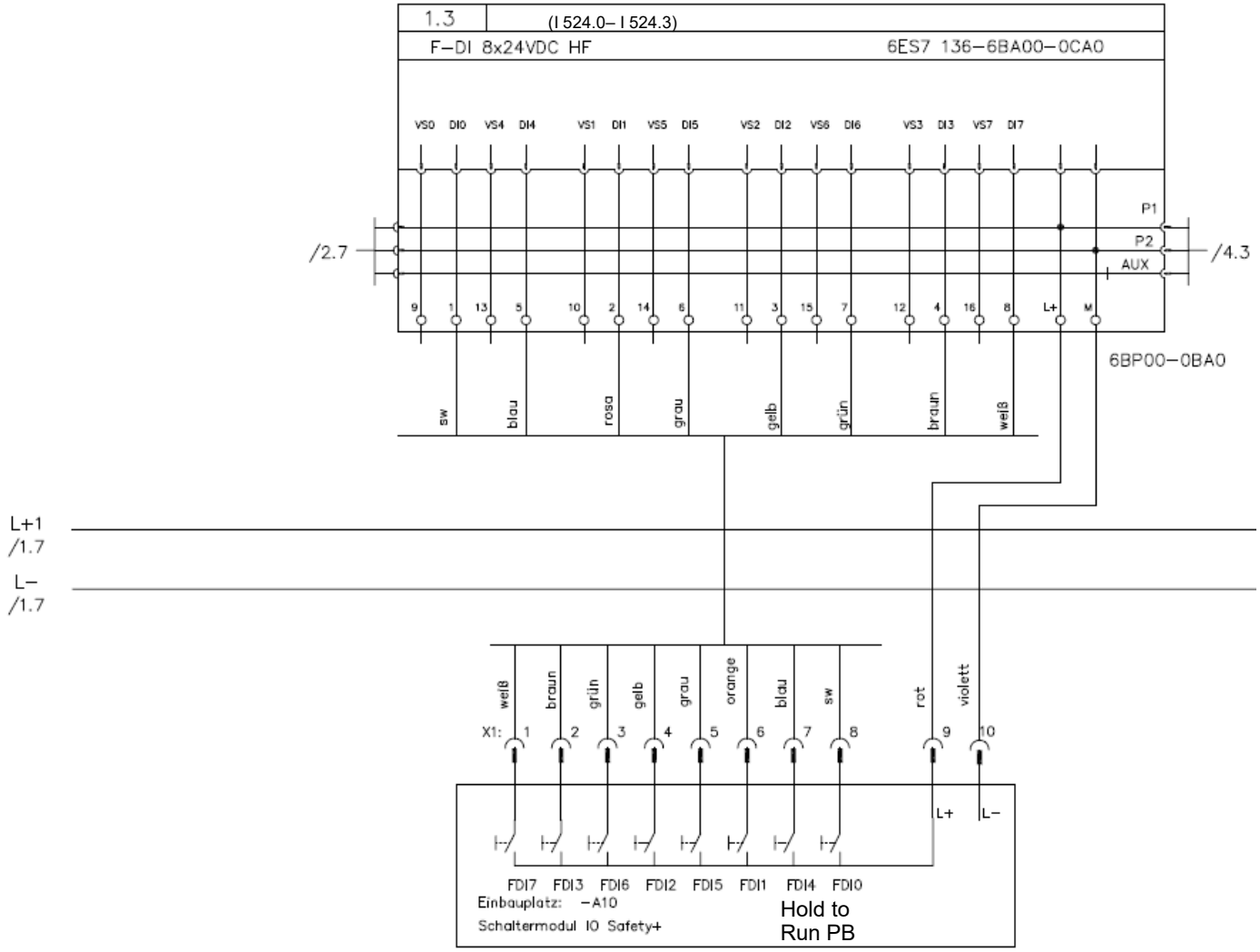
Safety Door

Notes:



Emergency Stop reset Set-up Mode

Notes:


1.3
(I 524.0– I 524.3)
F–DI 8x24VDC HF
6ES7 136–6BA00–0CA0
VS0 DI0 VS4 DI4 VS1 DI1 VS5 DI5 VS2 DI2 VS6 DI6 VS3 DI3 VS7 DI7
P1
P2
AUX
/2.7
/4.3
9 1 13 5 10 2 14 6 11 3 15 7 12 4 16 8 L+ M
6BP00–0BA0
sw blau rosa grau gelb grün braun weiß
L+1
/1.7
L–
/1.7
weiß braun grün gelb grau orange blau sw rot violett
X1: 1 2 3 4 5 6 7 8 9 10
L+ L–
FDI7 FDI3 FDI6 FDI2 FDI5 FDI1 FDI4 FDI0
Einbauplatz: –A10
Schaltermodul IO Safety+
Hold to
Run PB

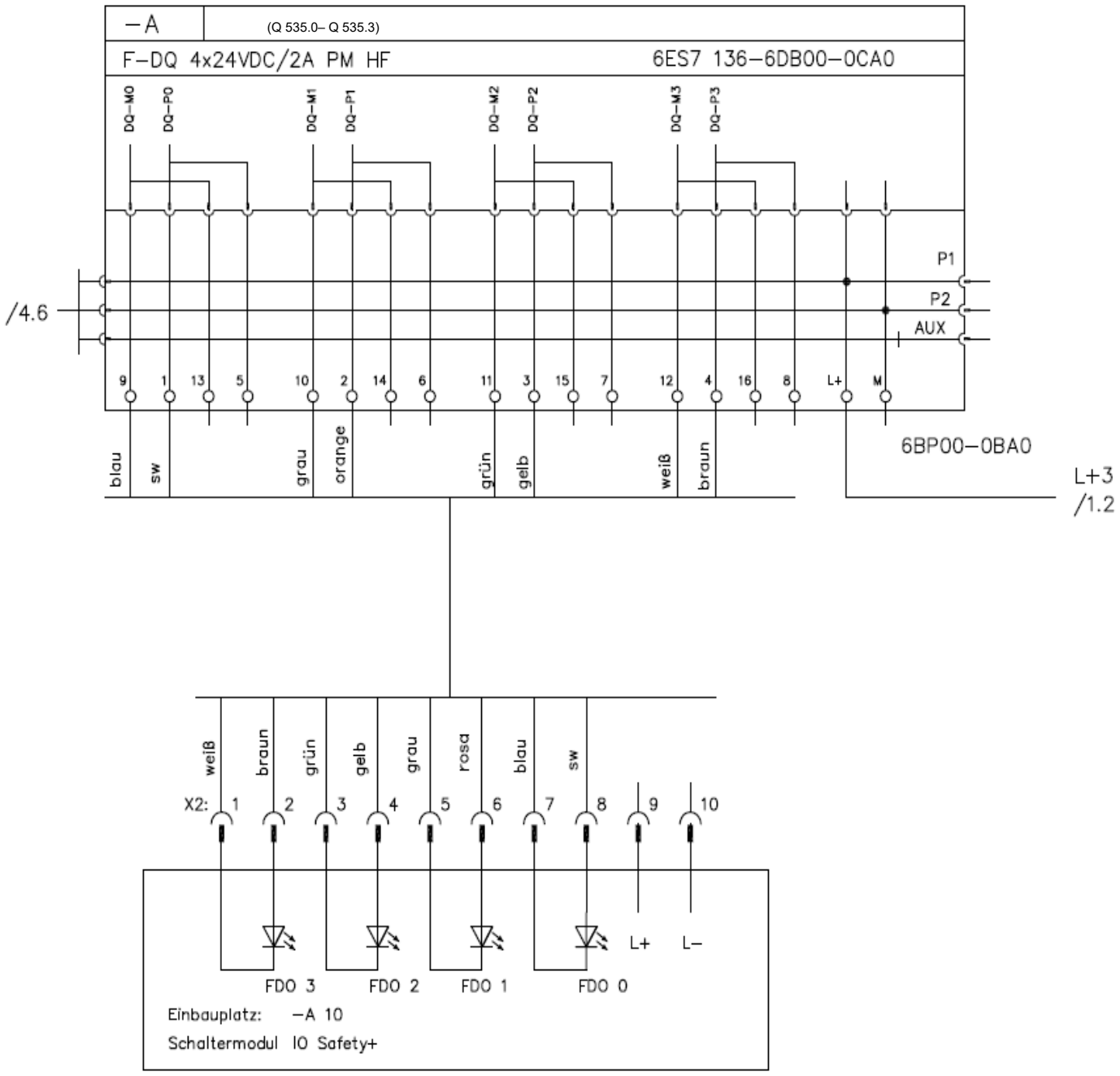
Notes:

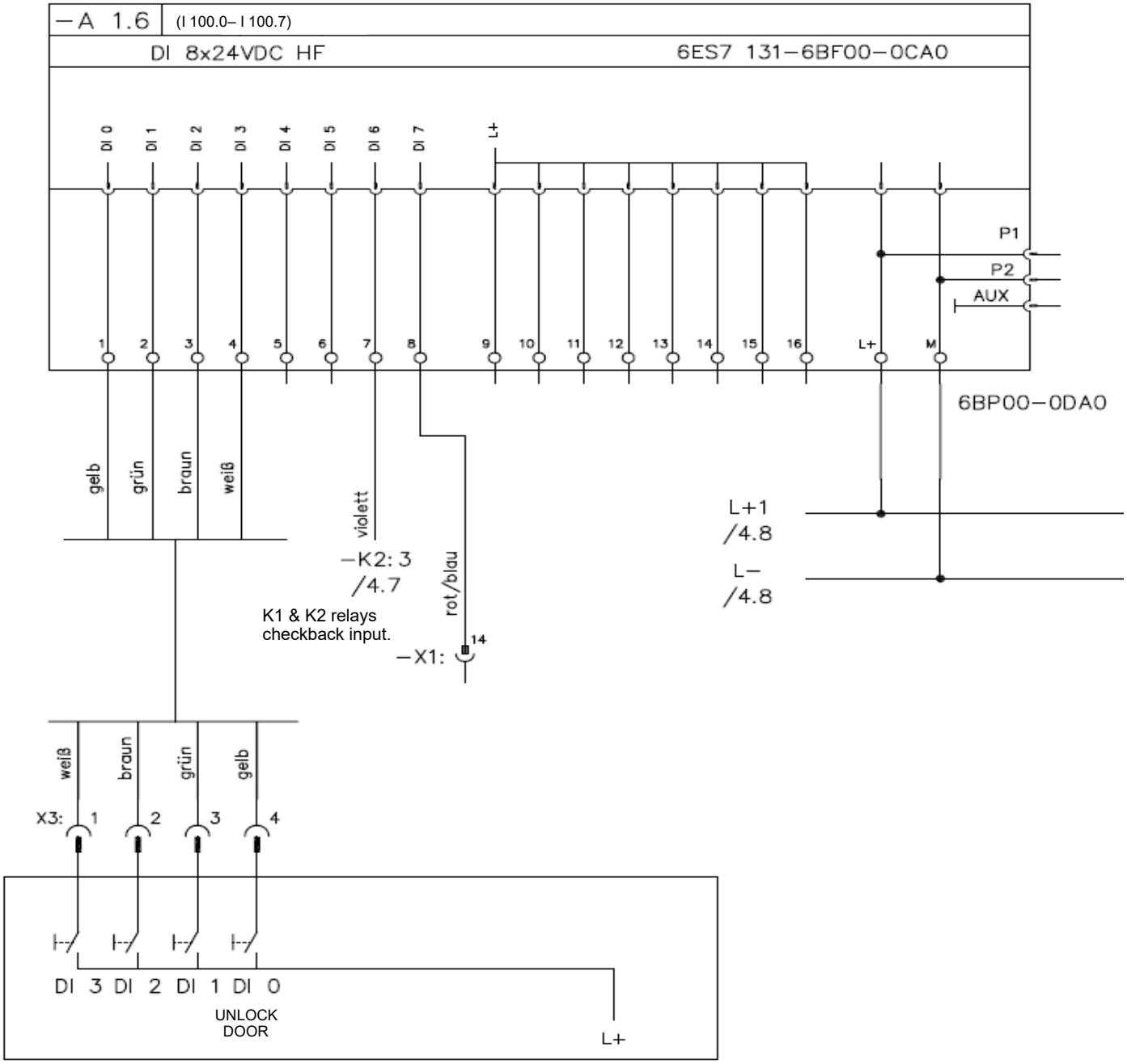
(Q 530.0– Q 530.3)



ESTOP ACK LED
INT EVENT LED
K2 LED
K1 LED

Notes:


−A
(Q 535.0– Q 535.3)
F−DQ 4x24VDC/2A PM HF
6ES7 136−6DB00−0CA0
DQ−M0
DQ−P0
DQ−M1
DQ−P1
DQ−M2
DQ−P2
DQ−M3
DQ−P3
/4.6
P1
P2
AUX
9 1 13 5 10 2 14 6 11 3 15 7 12 4 16 8 L+ M
blau
sw
grau
orange
grün
gelb
weiß
braun
6BP00−0BA0
L+3
/1.2
weiß
braun
grün
gelb
grau
rosa
blau
sw
X2: 1 2 3 4 5 6 7 8 9 10
L+
L−
FDO 3
FDO 2
FDO 1
FDO 0
Einbauplatz: −A 10
Schaltermodul IO Safety+


–A 1.6
(I 100.0– I 100.7)
DI 8x24VDC HF
6ES7 131–6BF00–0CA0
DI 0
DI 1
DI 2
DI 3
DI 4
DI 5
DI 6
DI 7
L+
P1
P2
AUX
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
L+
M
6BP00–0DA0
gelb
grün
braun
weiß
violett
–K2: 3
/4.7
K1 & K2 relays
checkback input.
rot/blau
14
–X1:
L+1
/4.8
L–
/4.8
weiß
braun
grün
gelb
X3:
1
2
3
4
DI 3 DI 2 DI 1 DI 0
UNLOCK
DOOR
L+

Notes:

Q540.0

QB100

Safety door lock/unlock

Q100.7 Door Unlock

Notes:

Notes:

Notes:

A207: END

Notes:

A219

SINUMERIK ONE

SINUMERIK ONE Safety Integrated Acceptance Test

Module Description:

The requirements associated with an acceptance test can be derived from the EU Machinery Directive. Accordingly, the machine manufacturer (OEM) is responsible for the following:

To carry out an acceptance test for safety--related functions and machine parts, and….
To issue an ”Acceptance certificate” that includes the results of the test.

When using the Safety Integrated function, the acceptance test is used to check the correct configuring of the SI monitoring functions used in the NCK, PLC and drive. The test objective is to verify proper implementation of the defined safety functions, to check test mechanisms (forced checking procedure measures) and to examine the response of individual monitoring functions by specifically violating tolerance limits. This should be carried out for all safety functions, i.e. for the axial monitoring functions, the safety-related communication relationships, the safety--related I/O etc.

Module Objective:

After completing the module, you are capable of the following:

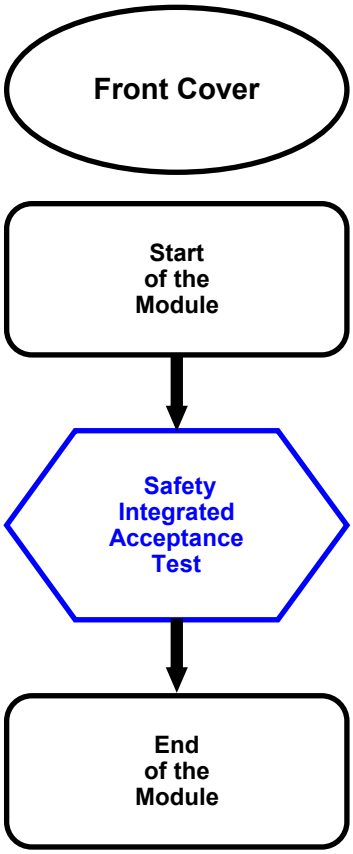
- Completing and understanding the requirements of the Safety Integrated Acceptance Test

Content:

- Safety Integrated Acceptance Test

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Safety Integrated Acceptance Test
End of the Module

Safety Integrated Acceptance Test (SIAT)

Description

The requirements associated with an acceptance test can be derived from the EU Machinery Directive. Accordingly, the machine manufacturer (OEM) is responsible for the following:

- To carry out an acceptance test for safety--related functions and machine parts, and….
- To issue an ”Acceptance certificate” that includes the results of the test.

When using the Safety Integrated function, the acceptance test is used to check the correct configuring of the SI monitoring functions used in the NCK, PLC and drive. The test objective is to verify proper implementation of the defined safety functions, to check test mechanisms (forced checking procedure measures) and to examine the response of individual monitoring functions by specifically violating tolerance limits. This should be carried out for all safety functions, i.e. for the axial monitoring functions, the safety-related communication relationships, the safety--related I/O etc.

Warning:
If machine data for SI functions is changed, a new acceptance test must be carried out for the modified SI function and recorded in the acceptance report.

Note: The previously used Test Stop Procedure is used during part of the SIAT.

Functionality of the SIAT:

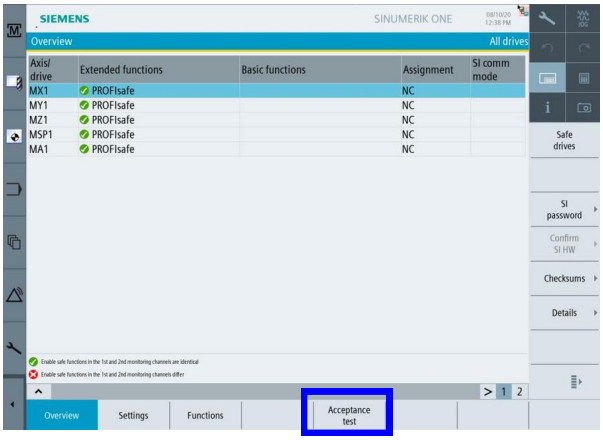
- Switch off signal paths
- Functional Relationships
- Position adjustment
- STO - Safe Torque Off
- SS1 - Safe Stop 1
- SS2 - Safe Stop 2
- SOS - Safe Operational Stop
- SLS - Safe Limited Speed
- SSM - Safe Speed Monitoring
- SLP - Safe Limited Position
- SBT (Sinamics) - Safe Brake Test
- SCA - Safe Cam
- SDI - Safe Direction

Procedure:

The SIAT is carried out using the Sinumerik Operate software of the controller.

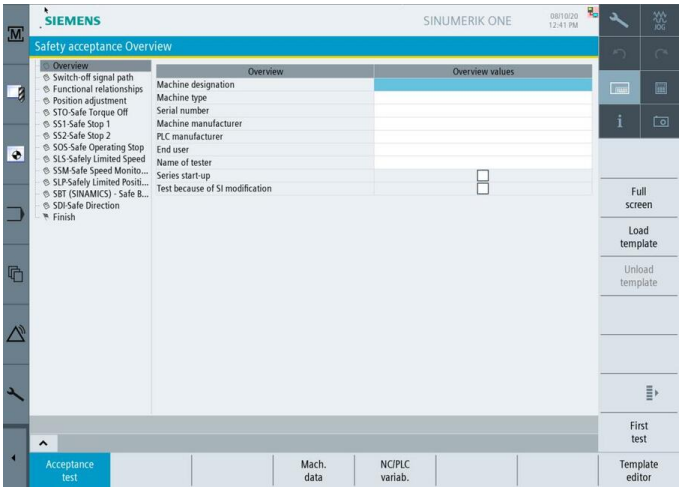








A list available tests is displayed to the left. Machine information etc. can be entered in the area to the right.



Notes:

Example: Machine information.

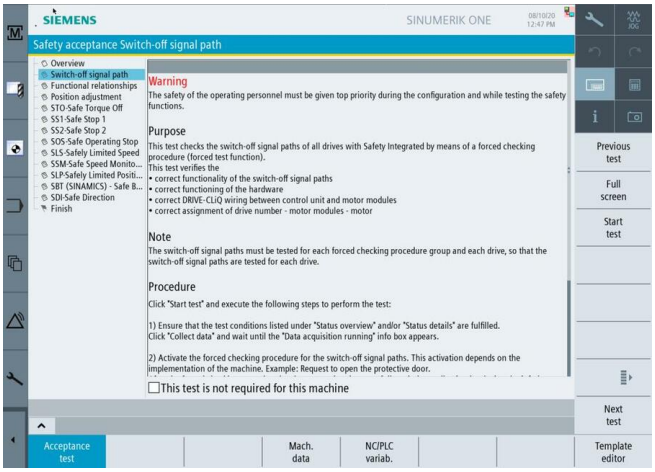
Overview	Overview values
Machine designation	TTT Sinumerik ONE dbSI
Machine type	Milling Machine
Serial number	123456789
Machine manufacturer	ABC Machine Tools
PLC manufacturer	A1 PLC Company
End user	ACME Products
Name of tester	John Smith
Series start-up	☐
Test because of SI modification	☐

There are two options available:

- **Series Start-up** - if the machine is part of a machine series, but not the prototype/first in the series.
- **Test because of SI modification** - If the machine is being re-tested because of a modification to the Safety Integrated set-up.

Select the **First test** soft-key:

First test

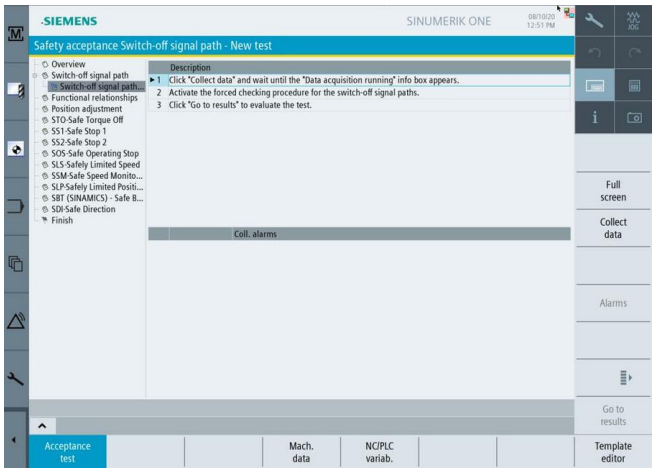


Switch off signal path:
The first test is the **Switch off signal path.**
The pulses to the drives will be disabled by running the ***Test Stop*** routine. The status of the drives will be monitored to check that they reacted in the correct manner.

A detailed description is available on the screen.

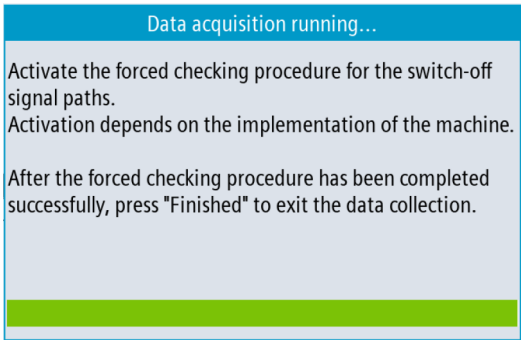
The machine should be in a runnable condition i.e. production mode, doors closed, no apparent faults etc.

Start test



Collect data

Notes:



- Initiate the Test Stop Procedure via input 100.1 on the switch box.
- Allow the Test Stop Procedure to complete.
- Select the ***Finished*** soft-key.



Messages which were present when the test was in progress can be selected/de-selected where relevant. These will be included in the SIAT report.

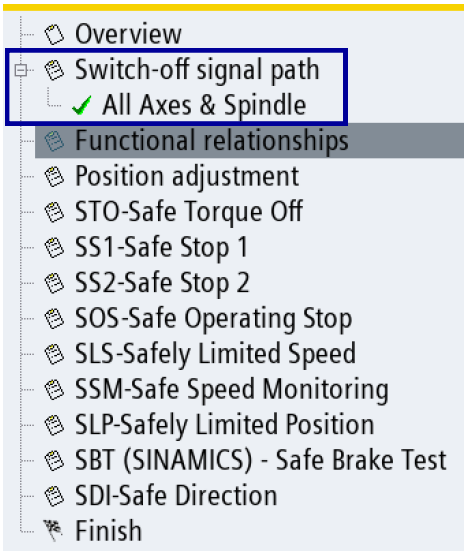
- Enter a name for the test e.g. All axes & Spindle.
- The conditions for running the test and the result of the test also need to be entered.

The messages seen in the example below are part of the Test Stop routine and are expected.





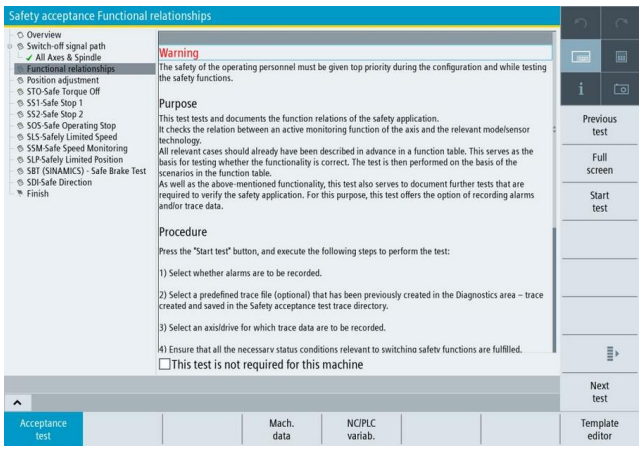
The previous test with can be seen marked with a green tick, to the left of the screen:



Functional relationships

The Functional Relationship test is used to confirm that the Safety Integrated functionality adheres to the function table.





Note: the Functional Relationship test does not require the Test Stop Procedure.

Notes:

Example function table:

Operating mode	*Protective doors*	*Axis / spindle*	*Monitoring*
Production	Closed	X1, Y1, Z1, A1 SP1	SLS4 SLS4
	Open	X1, Y1, Z1, A1 SP1	SOS STO
Setup	Closed	X1, Y1, Z1, A1 SP1	SLS4 SLS4
	Open	X1, Y1, Z1, A1 SP1	SOS or SLS1 With Hold-To-Run PB STO

Ensure that the machine is operational.

Create the conditions for the first test:

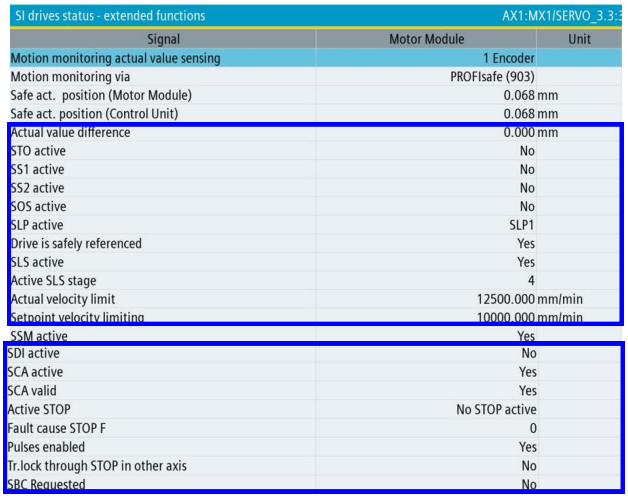
- Production Mode.
- Doors closed.

Locate the Safety Integrated diagnostics screens:

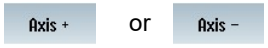


Check that the correct status with respect to the function table.

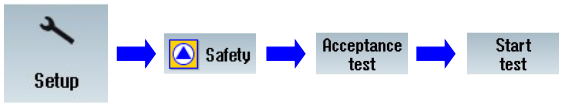
Note: the diagnostic display will differ depending on the enabled safety functions.



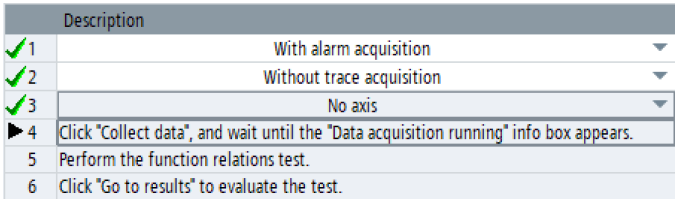
Check that the correct status is active for each axis.



Return to the Safety Integrated Acceptance Test screen:

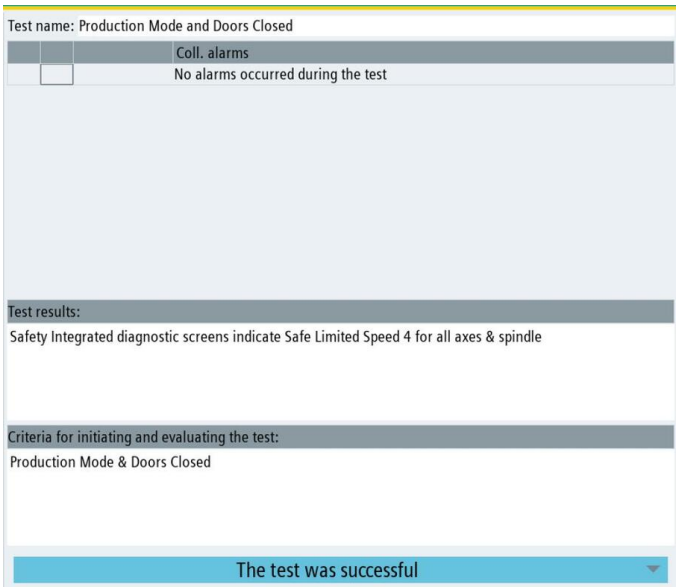


There is no requirement for alarm acquisition, trace acquisition or axis selection





Enter the required information:





Notes:

Create the conditions for the second test:

- Production Mode.
- Doors Open .

Check the actual status via the diagnostic screen.

SI drive status for all the X, Y, Z & A axes:

SI drives status - extended functions AX1:MX1/SERVO_3.3:3

Signal	Motor Module	Unit
Safe act. position (Control Unit)	0.069	mm
Actual value difference	0.000	mm
STO active	No	
SS1 active	No	
SS2 active	No	
SOS active	Yes	
SLP active	SLP1	
Drive is safely referenced	Yes	
SLS active	No	
Active SLS stage	-	
Actual velocity limit	0.000	mm/min
Setpoint velocity limiting	0.000	mm/min
SSM active	Yes	
SDI active	No	
SCA active	Yes	
SCA valid	Yes	
Active STOP	No STOP active	
Fault cause STOP F	0	
Pulses enabled	Yes	
Tr.lock through STOP in other axis	No	
SBC Requested	No	

SI drive status for all the Spindle:

SI drives status - extended functions AX4:MSP1/SERVO_3.3:2

Signal	Motor Module	Unit
Motion monitoring actual value sensing	2 Encoder	
Motion monitoring via	PROFIsafe (903)	
Safe act. position (Motor Module)	359.988	°
Safe act. position (Control Unit)	359.988	°
Actual value difference	0.000	°
STO active	Yes	
SS1 active	No	
SS2 active	No	
SOS active	No	
SLS active	No	
Active SLS stage	-	
Actual velocity limit	0.000	rpm
Setpoint velocity limiting	0.000	rpm
SSM active	Yes	
Active STOP	STOP A active	
Fault cause STOP F	0	
Pulses enabled	No	
Tr.lock through STOP in other axis	No	
SBC Requested	No	

Enter the required information:

Test name: Production Mode and Doors Open

		Coll. alarms
		No alarms occurred during the test

Test results:

Safety Integrated diagnostic screens indicate Safe Operational Stop for X, Y, Z & A Axes and Safe Torque Off f or the Spindle

Criteria for initiating and evaluating the test:

Production Mode and Doors Open

The test was successful

Create the conditions for the third test:

- Set-up Mode.
- Doors closed.

Check the actual status via the diagnostic screen:

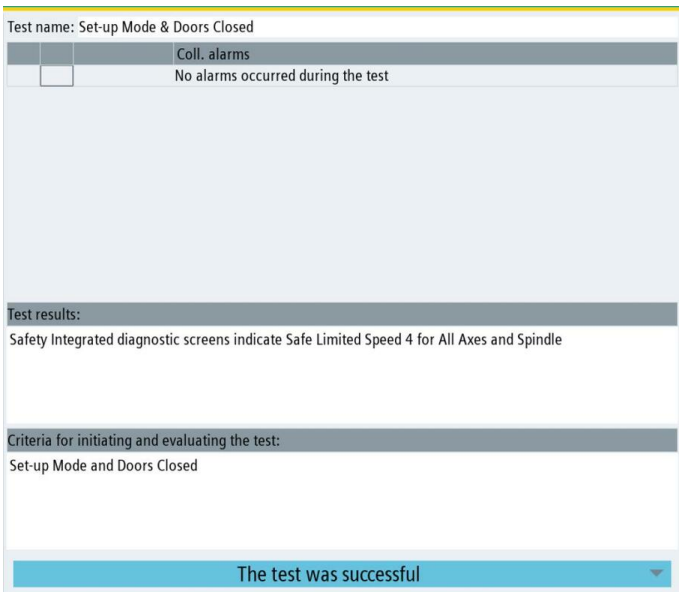
SI status for all the axes & spindle:

SI drives status - extended functions AX1:MX1/SERVO_3.3:3

Signal	Motor Module	Unit
Motion monitoring actual value sensing	1 Encoder	
Motion monitoring via	PROFIsafe (903)	
Safe act. position (Motor Module)	0.068	mm
Safe act. position (Control Unit)	0.068	mm
Actual value difference	0.000	mm
STO active	No	
SS1 active	No	
SS2 active	No	
SOS active	No	
SLP active	SLP1	
Drive is safely referenced	Yes	
SLS active	Yes	
Active SLS stage	4	
Actual velocity limit	12500.000	mm/min
Setpoint velocity limiting	10000.000	mm/min
SSM active	Yes	
SDI active	No	
SCA active	Yes	
SCA valid	Yes	
Active STOP	No STOP active	
Fault cause STOP F	0	
Pulses enabled	Yes	
Tr.lock through STOP in other axis	No	
SBC Requested	No	

Notes:

Enter the required information:

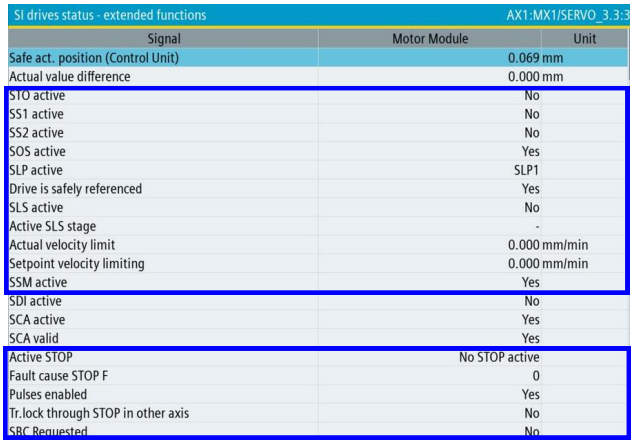


Create the conditions for the fourth test:

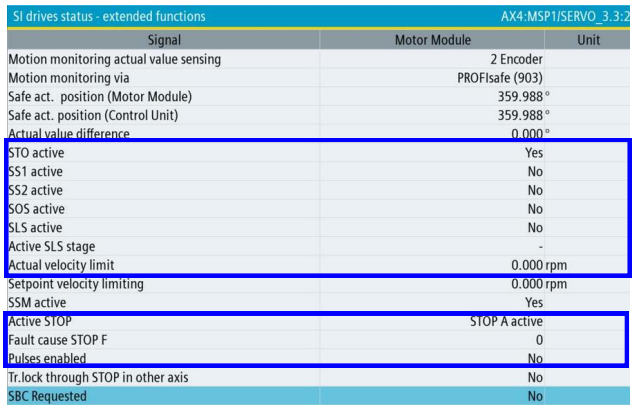
- Set-up Mode.
- Doors open.

Check the actual status via the diagnostic screen:

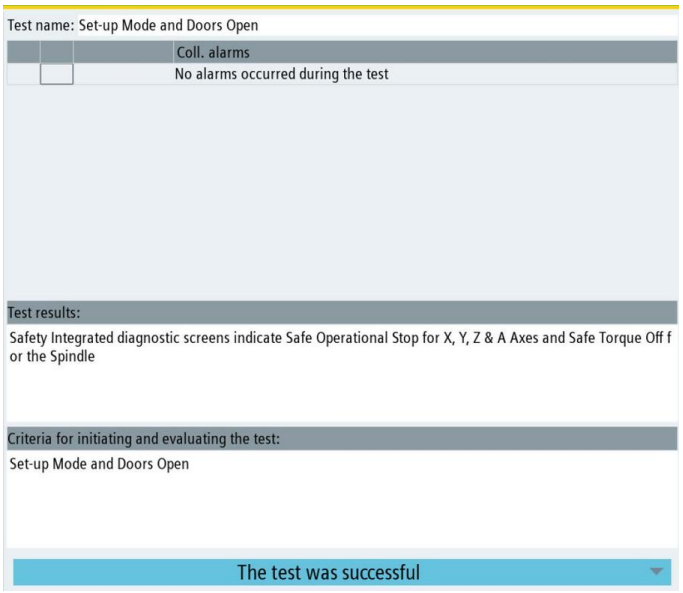
SI status for the X Y Z & A:



SI status for the Spindle:



Enter the required information:



Notes:

Create the conditions for the fifth test:

- Set-up Mode.
- Doors open.
- Hold-To-Run PB pressed.

Check the actual status via the diagnostic screen.

SI status for the all axes: Spindle in STO as on previous page

SI drives status - extended functions AX1:MX1/SERVO_3.3:3

Signal	Motor Module	Unit
Motion monitoring actual value sensing	1 Encoder	
Motion monitoring via	PROFIsafe (903)	
Safe act. position (Motor Module)	0.069	mm
Safe act. position (Control Unit)	0.069	mm
Actual value difference	0.000	mm
STO active	No	
SS1 active	No	
SS2 active	No	
SOS active	No	
SLP active	SLP1	
Drive is safely referenced	Yes	
SLS active	Yes	
Active SLS stage	1	
Actual velocity limit	2500.000	mm/min
Setpoint velocity limiting	2000.000	mm/min
SSM active	Yes	
SDI active	No	
SCA active	Yes	
SCA valid	Yes	
Active STOP	No STOP active	
Fault cause STOP F	0	

Enter the required information:

Test name: Set-up Mode and Doors Open & Hold-to-run PB pressed

	Coll. alarms
☐	No alarms occurred during the test

Test results:

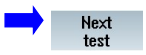
Safety Integrated diagnostic screens indicate Safe Limited Speed 1 for all Axes and Safe Torque Off for the Spindle.

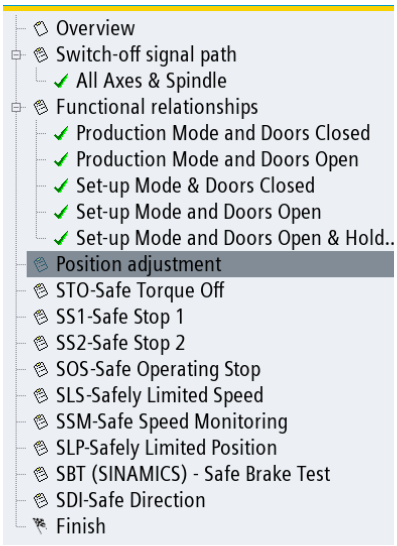
Criteria for initiating and evaluating the test:

Set-up Mode and Doors Open & Hold-to-run PB pressed

The test was successful

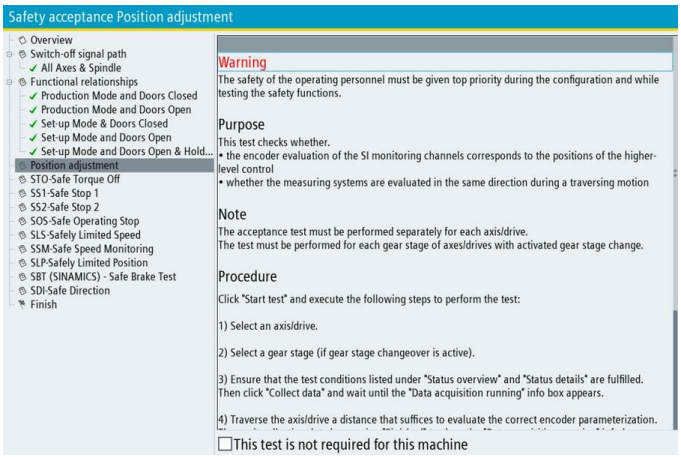
The functional Relationship test is now complete. Select the next ***Next test*** soft-key to advance to the next test:





Position Adjustment

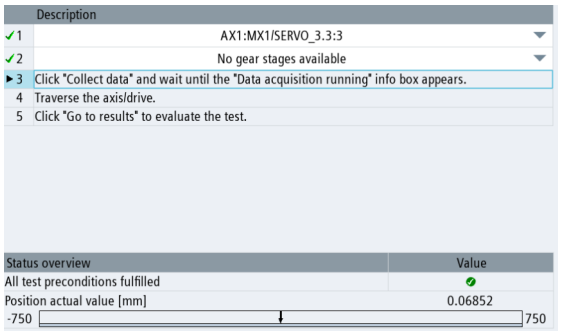
Checks for the correct position direction between the controller and the Safety Integrated monitoring channels.



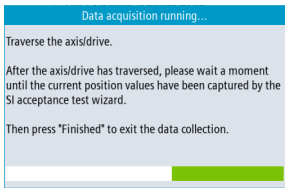
Notes:

Start test

Select the axis to be tested:



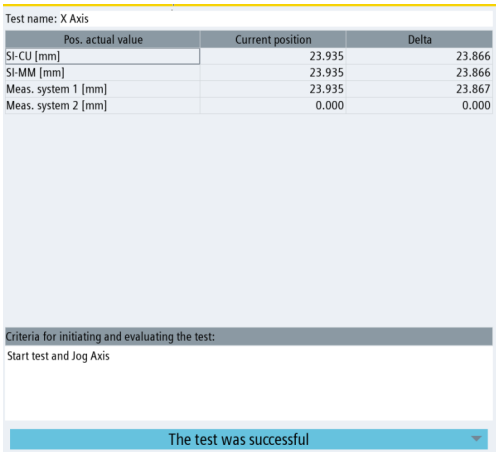
Collect data



- Jog the axis

Finished → Go to results

The positions can be seen to be Identical:



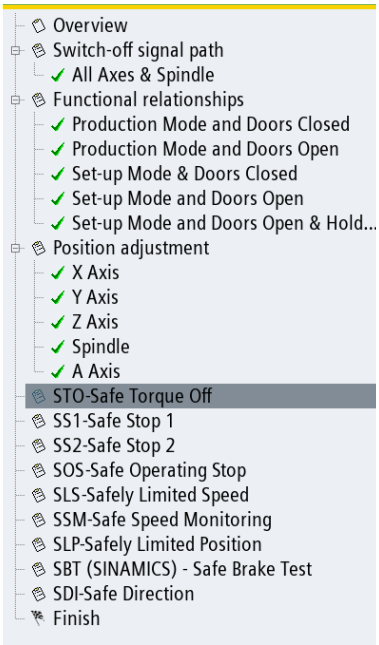
Start new test

- Repeat for the remaining Axes

Note: the training equipment has only the X axis set-up for safe referencing. This means that the SI-CU & SI-MM positions will match each other but not that of Meas. System 1.

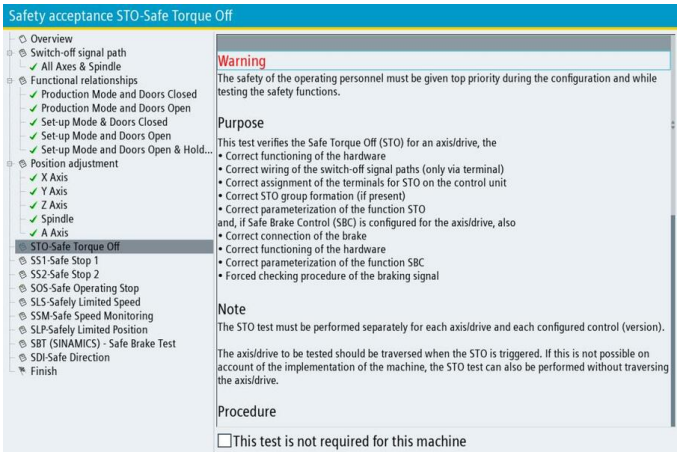
Select the ***Next test*** soft-key:

Next test



Safe Torque Off Test

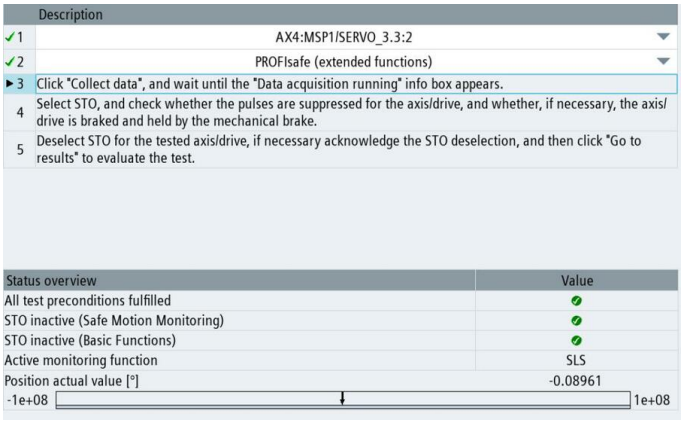
Note: for the training example only the spindle is tested for STO functionality.



Start test

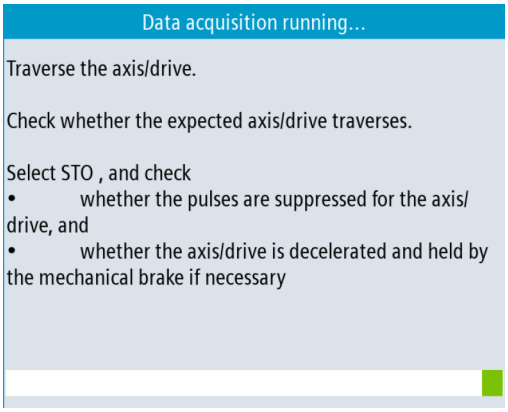
Notes:

Select the axis to be tested and check that the conditions are correct:



Collect data

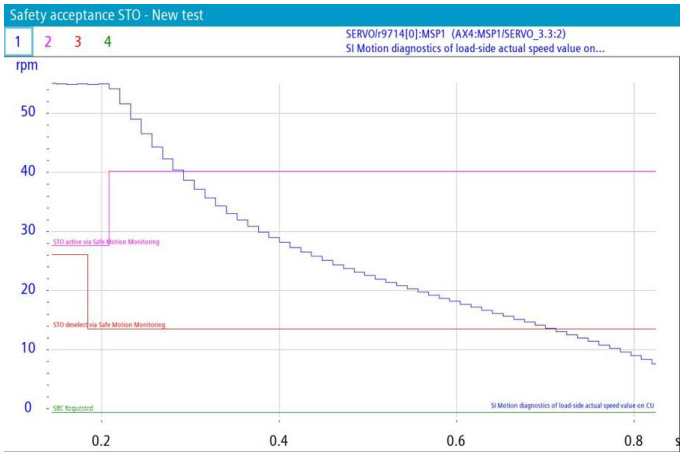
Jog the axis select STO by unlocking/opening the door.



Go to results

View trace

The results show the STO function reacting to the request bringing the axis to a standstill:



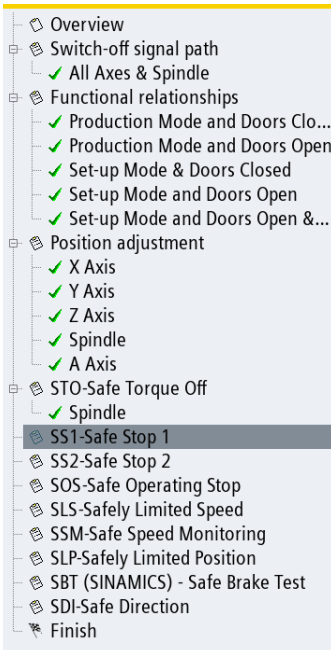


Back

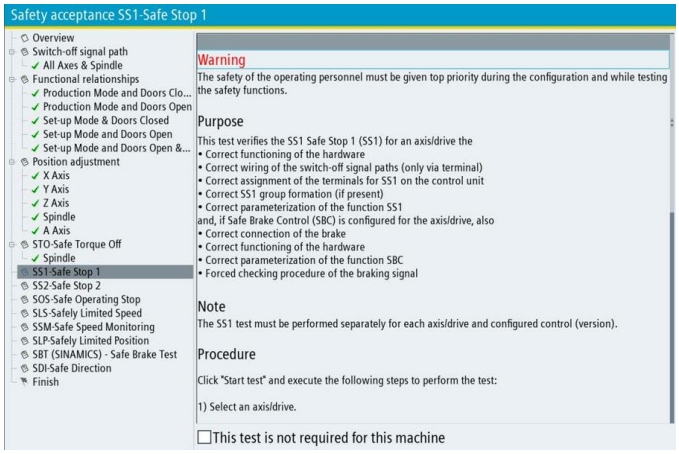
Notes:

Select the ***Next test*** soft-key:





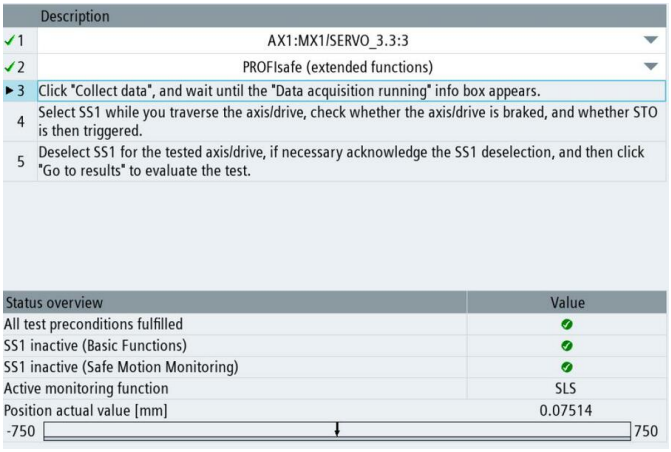
Safe Stop 1 Test

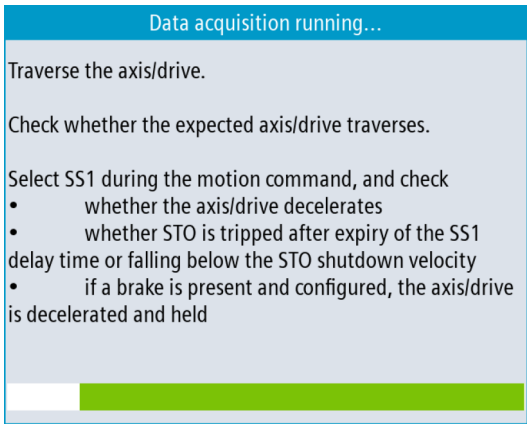




Select the axis to be tested:

Check the conditions are correct:





Press the Emergency Stop button while jogging the axis



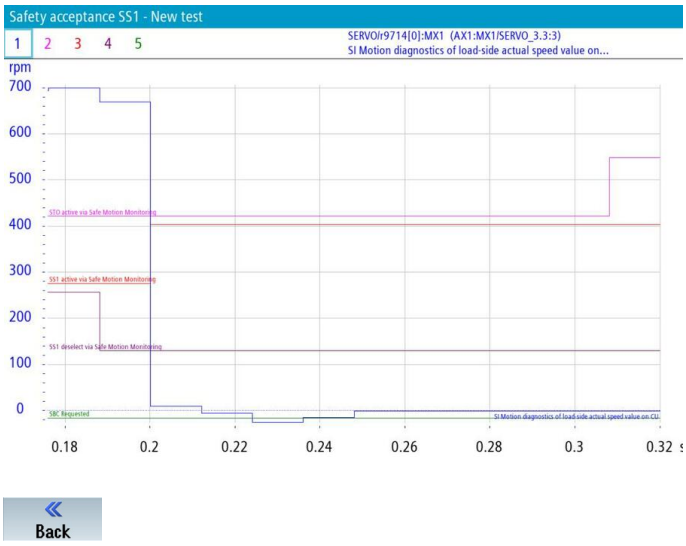




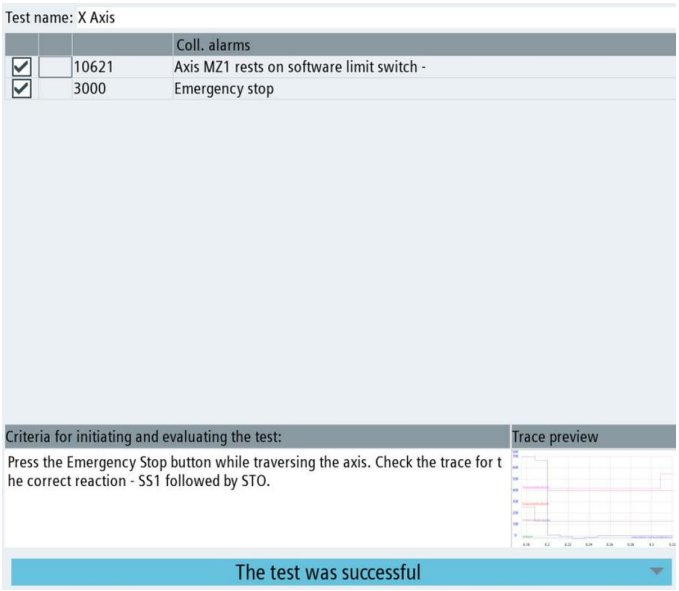


Notes:

SS1 can be seen being activated followed by STO.

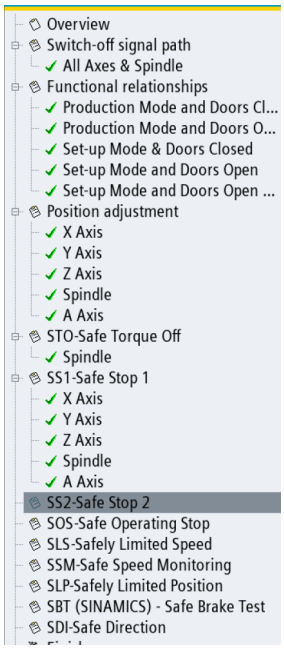


Enter the required information:

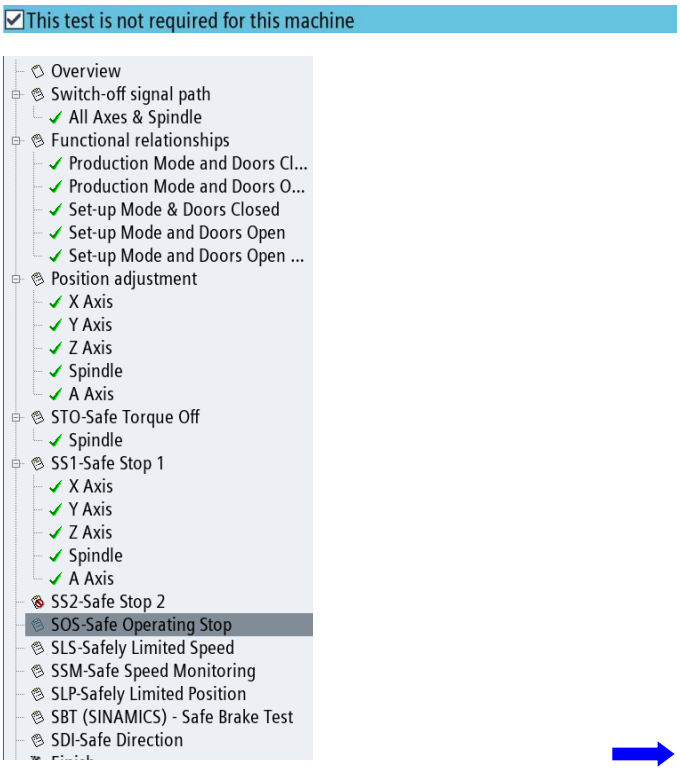


Select the ***Next test*** soft-key:



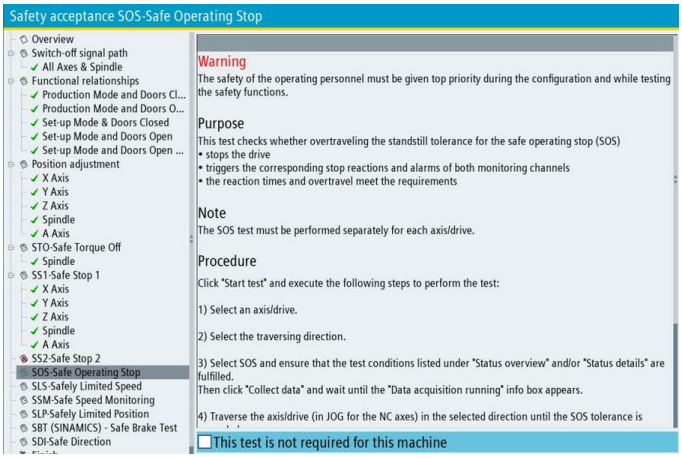


SS2 is not in use in this example



Notes:

Safe Operating Stop Test



Start test

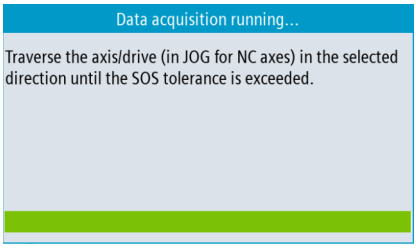
Select the axis to be tested:

Select the axis to be tested, a direction of travel, and check that the conditions are correct:

	Description
✓1	AX1:MX1/SERVO_3.3:3
✓2	Traversing direction positive
▶3	Select SOS, click "Collect data", and wait until the "Data acquisition running" info box appears.
4	Traverse the axis/drive (in JOG for NC axes) in the selected direction until the SOS tolerance is exceeded.
5	Click "Go to results" to evaluate the test.

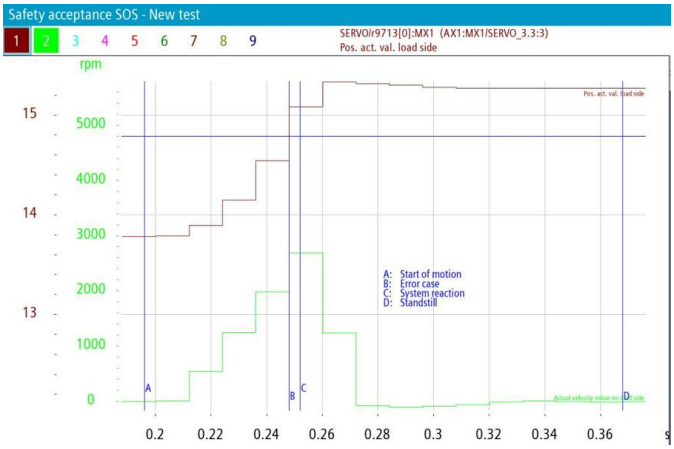
Status overview	Value
All test preconditions fulfilled	✓
SOS active	✓
Active monitoring function	SOS
Position actual value [mm]	13.78572
-750	750

Collect data



Go to results → View trace

The axis was stopped after movement of approximately 1mm.



Back

Enter the required information:

Test name: X Axis

Description	Data
Speed overshoot	2695.000 mm/min
Reaction time	0.004 s
Overtravel from start	1.547 mm
Overtravel from overshoot	0.547 mm

		Coll. alarms
☑	230700	Axis MX1 SERVO_3.3:3 (3), Component Motor_Module_3 (3): SI Motion P2: STOP A initiated.
☑	201700	Axis MX1 SERVO_3.3:3 (3): SI Motion P1 (CU): STOP A initiated.
☑	230701	Axis MX1 SERVO_3.3:3 (3), Component Motor_Module_3 (3): SI Motion P2: STOP B initiated.
☑	230707	Axis MX1 SERVO_3.3:3 (3), Component Motor_Module_3 (3): SI Motion P2: Tolerance for safe operating stop exceeded.
☑	201707	Axis MX1 SERVO_3.3:3 (3): SI Motion P1 (CU): Tolerance for safe operating stop exceeded.
☑	201701	Axis MX1 SERVO_3.3:3 (3): SI Motion P1 (CU): STOP B initiated.

Criteria for initiating and evaluating the test:	Trace preview
Open the door in production mode. Start the test and jog in the direction specified. Check the trace for the correct result - the axis was stopped after the movement was detected.	

The test was successful

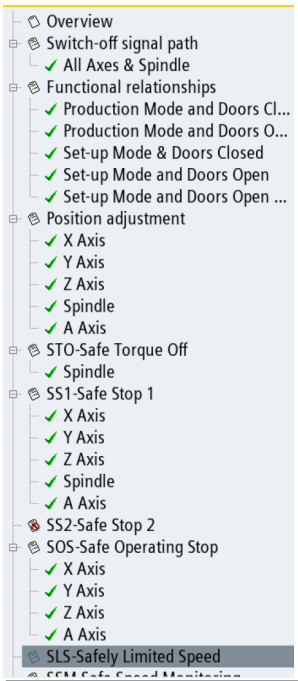
Start new test

- Repeat for all axes

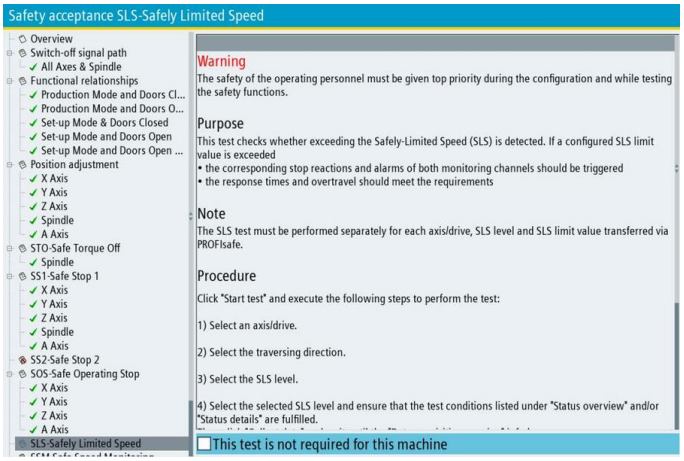
Notes:

Select the ***Next test*** soft-key:









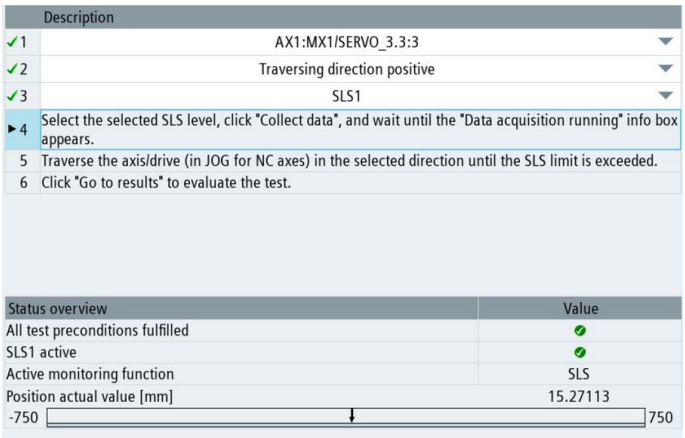
- **SLS - Safely Limited Speed:**

The training example uses SLS1 & SLS4.

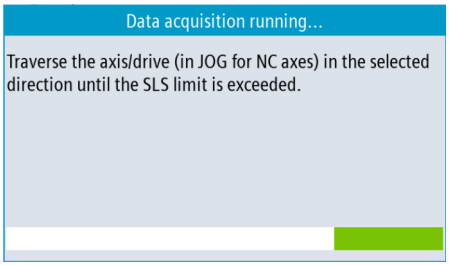
	SLS1	SLS4
X, Y, Z	2500 mm/min	12500 mm/min
A	12.5 rpm	125 rpm
Spindle		1250 rpm

- SLS 1

Set the conditions for SLS1 - refer to the function table.



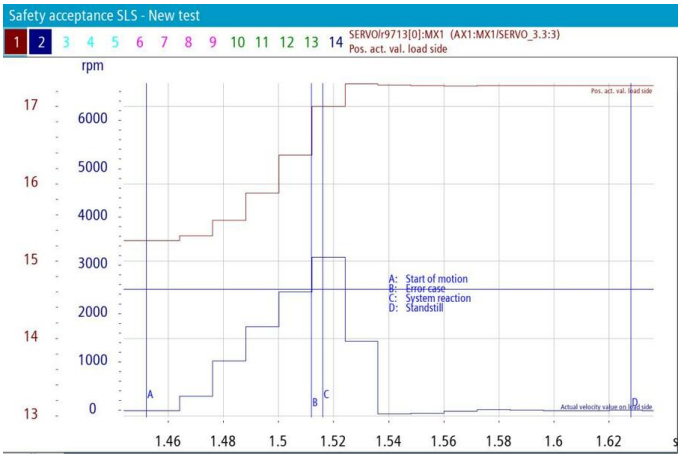


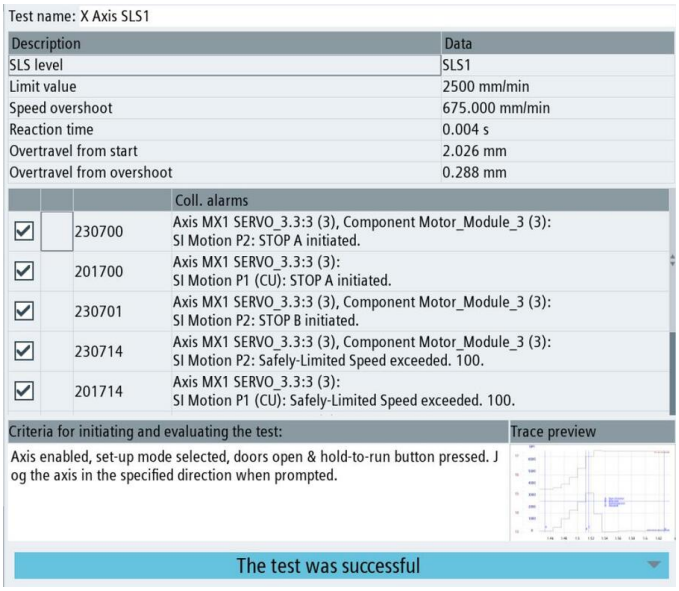




Notes:

From the trace it can be seen that the axis was halted when the velocity (blue trace) reached the SLS1 limit (2500 mm/min).




Test name: X Axis SLS1

Description	Data
SLS level	SLS1
Limit value	2500 mm/min
Speed overshoot	675.000 mm/min
Reaction time	0.004 s
Overtravel from start	2.026 mm
Overtravel from overshoot	0.288 mm

		Coll. alarms
☑	230700	Axis MX1 SERVO_3.3:3 (3), Component Motor_Module_3 (3): SI Motion P2: STOP A initiated.
☑	201700	Axis MX1 SERVO_3.3:3 (3): SI Motion P1 (CU): STOP A initiated.
☑	230701	Axis MX1 SERVO_3.3:3 (3), Component Motor_Module_3 (3): SI Motion P2: STOP B initiated.
☑	230714	Axis MX1 SERVO_3.3:3 (3), Component Motor_Module_3 (3): SI Motion P2: Safely-Limited Speed exceeded. 100.
☑	201714	Axis MX1 SERVO_3.3:3 (3): SI Motion P1 (CU): Safely-Limited Speed exceeded. 100.

Criteria for initiating and evaluating the test: | Trace preview

Axis enabled, set-up mode selected, doors open & hold-to-run button pressed. J og the axis in the specified direction when prompted.

The test was successful

Back Review the data and enter the results.

Start new test

- Repeat the SLS1 test for each axis (except for the spindle).

Machine data	Axis	Original	SLS 4 Test value
32000	X, Y, Z	10,000mm/min	14,000mm/min
	Spindle	1000rpm	1400rpm
	A	100rpm	140rpm
32010	X, Y, Z	5,000mm/min	14,000mm/min
	Spindle	50rpm	1400rpm
	A	25rpm	140rpm
36200	X, Y, Z	11,150	15,000mm/min
	Spindle [0]&[1]	1,150	1,500rpm
	A	115	150rpm

SLS 4

Note: the training example has SLS4 levels of 12500mm/min for the linear axes, 1250rpm for the spindle and 125rpm for the A axis.

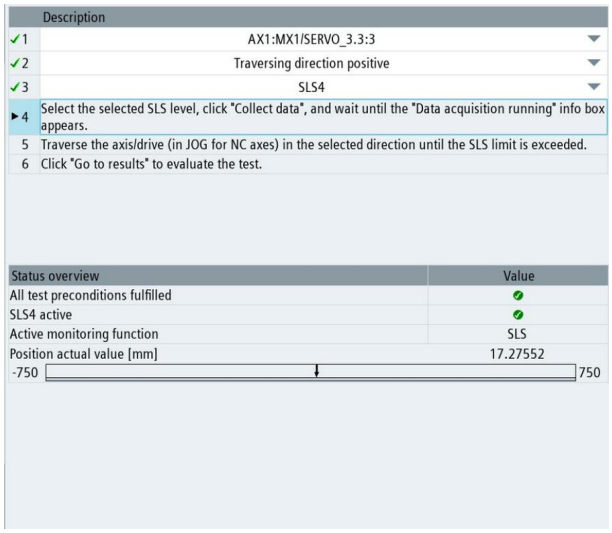
On an actual machine SLS4 should be tested with care due to the possibility of breaching the capabilities of the machine/ motor/drive.
Assuming that it is safe to test the SLS4 it will be necessary to change certain machine data to allow the axis to achieve the SLS 4 limit.

Training equipment machine data changes:

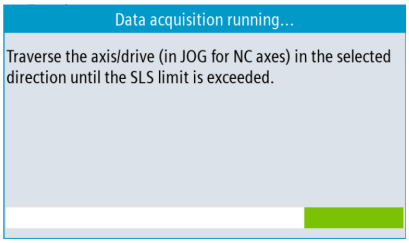
Mach. data → Axis MD

Notes:

- Select the required SLS4 test parameters and ensure that the status is correct for the test to be started.



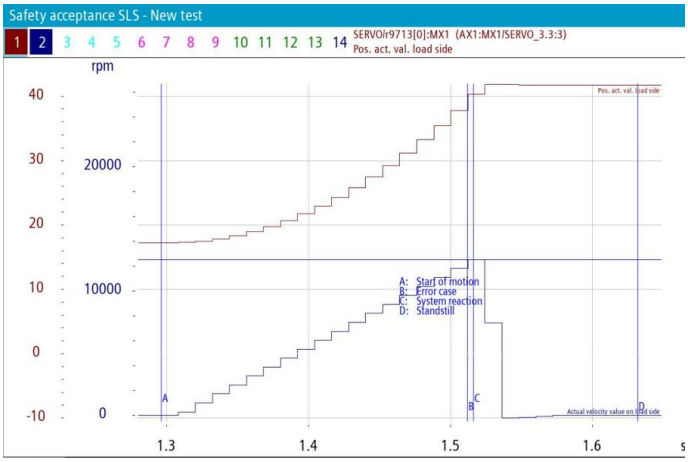
Collect data



Go to results → View trace

- Move the axis in the direction specified using the rapid-jog function.

The trace shows that the axis was halted when the velocity (blue trace) reached the SLS4 limit (12500 mm/min).

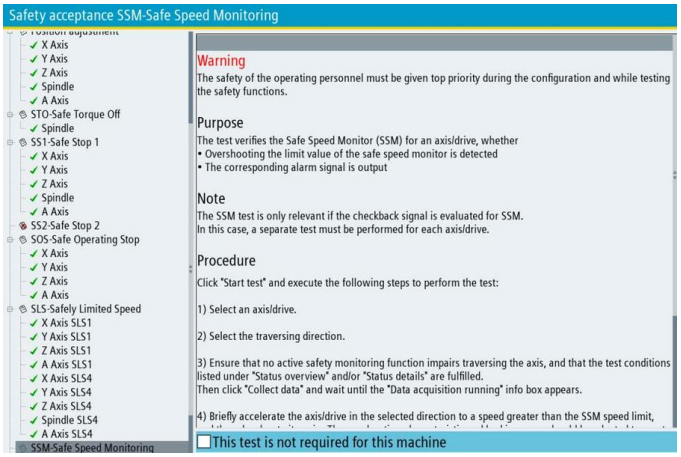


- Repeat the SLS4 test for each axis and the spindle.

Start new test

- Set any changed machine data back to their original values..

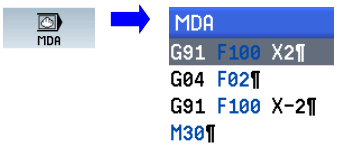
Next test



- **SSM - Safe Speed Monitoring:**

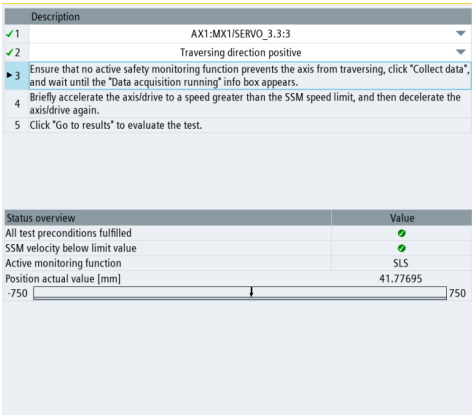
A simple MDA program can be used to move the axis in order to check the switching of the SSM signal.

Notes:

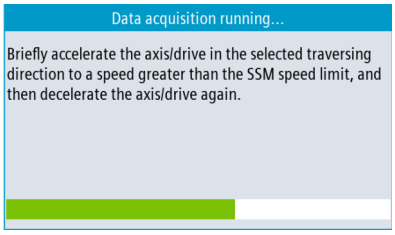


Select the axis to be tested and check that the status is correct etc.

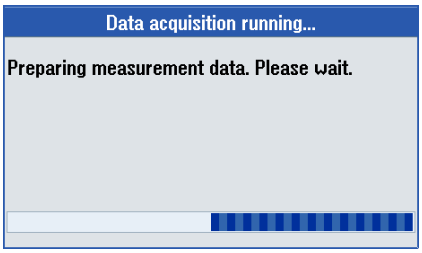






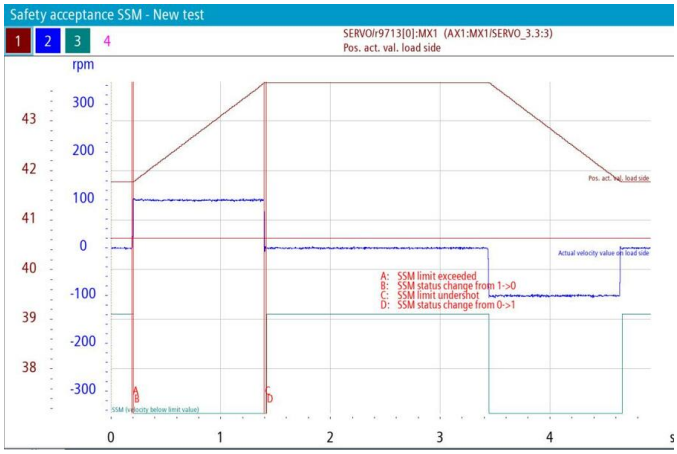


- Press the ***Cycle start*** button to run the axis.
- Allow the test to finish

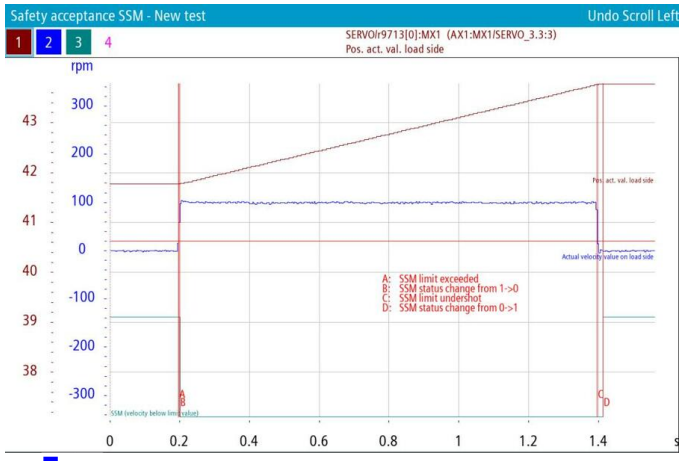




The trace shows the switching of the SSM signal (green trace) as the feed-rate (blue trace) goes above/below 20 mm/min



The zoom feature is useful for a more detailed view.



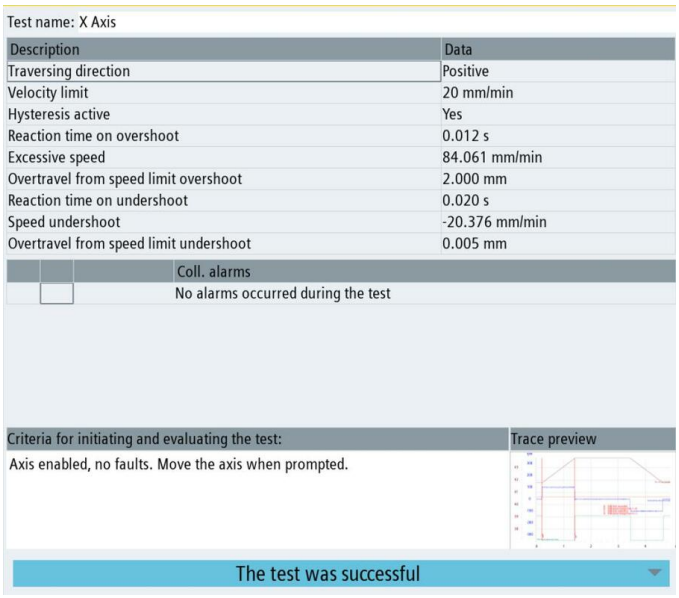


Review the data and enter the results.

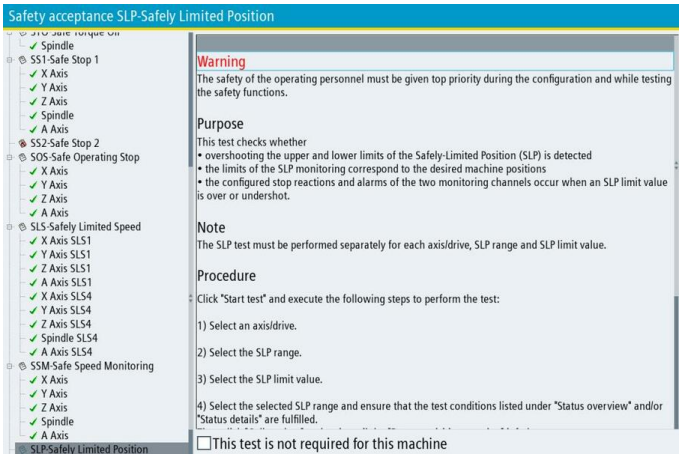


Notes:

Review the data and enter the results.



Next test



- **SLP - Safe Limited Position:**

Notes: for the training course, the safe limits have been enabled only on the X axis.

The axis must be referenced and the user agreement selected.

Start new test

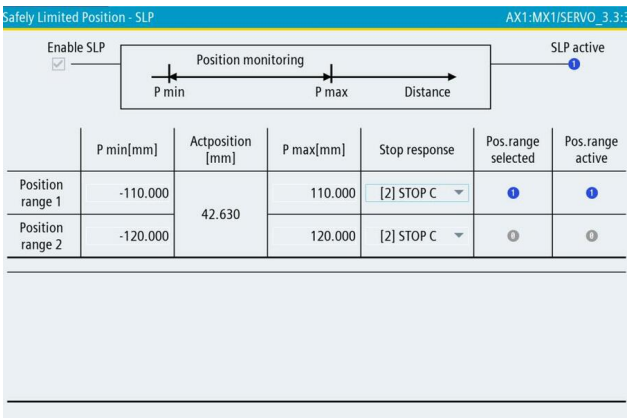
- Repeat for each axis.

The feed-rate used during the test is the G00 setting.
This can lead to unwanted Safety Integrated messages.

It is recommended that axis machine data, **32000 $MA_MAX_AX_VELO**, be set to 2000mm/min for the test to be able to continue error free.

Check the active SLP range. The first and second ranges are selected by the End Limit Override switch.

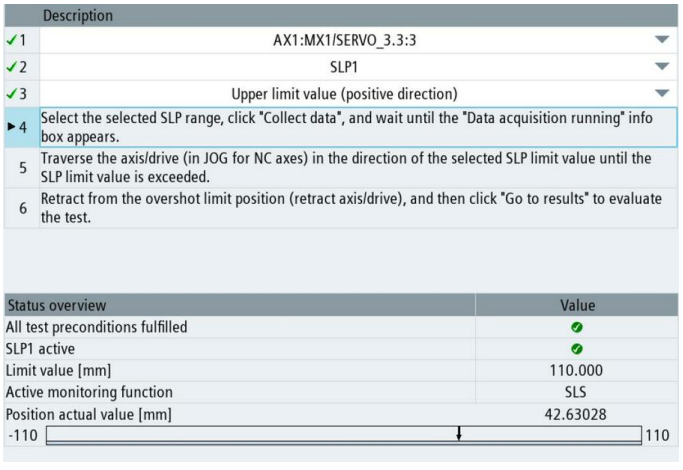
Functions → > → SLP/SP



Acceptance test → Start test

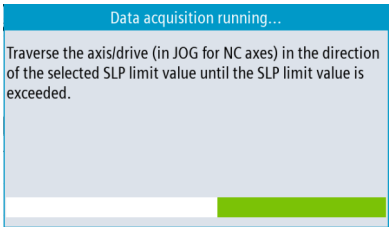
Select the required test options e.g.:
1. X axis
2. SLP1
3. Positive direction

Also, check that the conditions for the test are ok.



Notes:

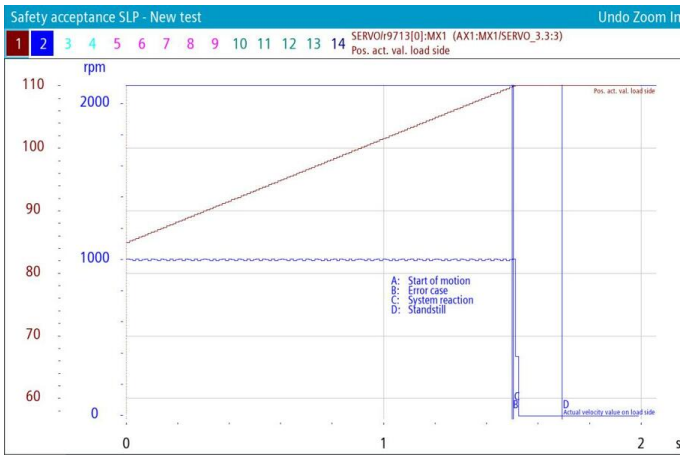
Collect data



- Jog the axis in a positive direction towards until the first SLP is reached (+110 mm)

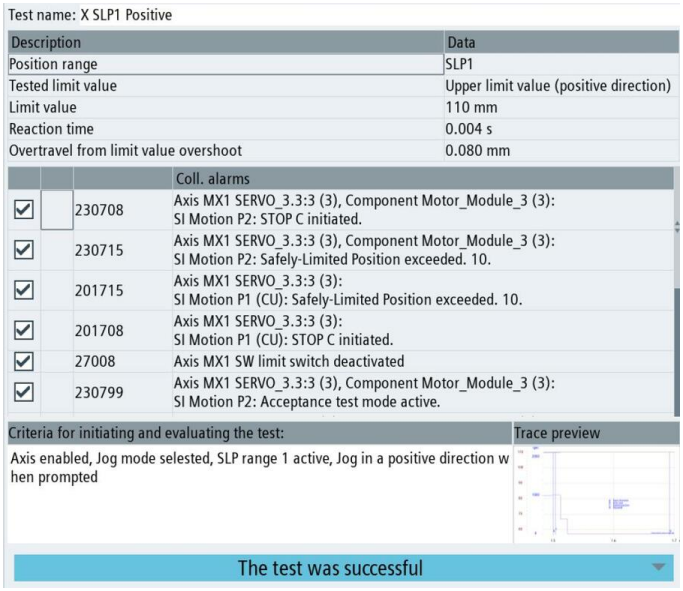
Go to results → View trace

The trace shows that the axis was stopped when the 110mm limit (brown trace) is reached:



Back

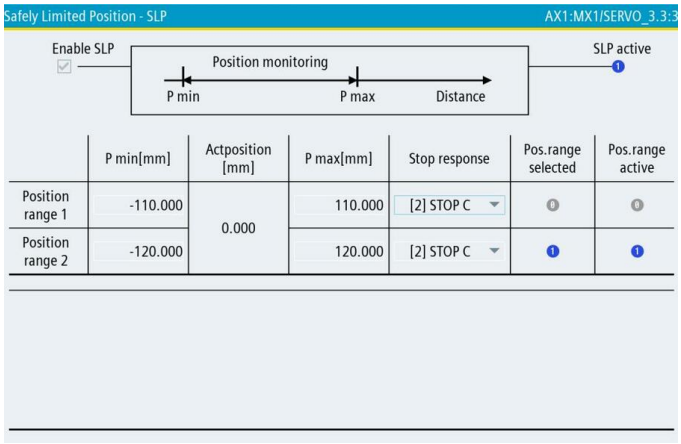
Review the data and enter the results.



- Jog the axis away from the limit
- Repeat the SLP1 test in the negative direction.

SLP range 2

The second SLP range now needs to be selected via the End Limit Override switch



Start new test

Set the parameters for the SLP2 positive direction.

Notes:

	Description
✓1	AX1:MX1/SERVO_3.3:3
✓2	SLP2
✓3	Upper limit value (positive direction)
▶4	Select the selected SLP range, click "Collect data", and wait until the "Data acquisition running" info box appears.
5	Traverse the axis/drive (in JOG for NC axes) in the direction of the selected SLP limit value until the SLP limit value is exceeded.
6	Retract from the overshot limit position (retract axis/drive), and then click "Go to results" to evaluate the test.

Status overview	Value
All test preconditions fulfilled	✓
SLP2 active	✓
Limit value [mm]	120.000
Active monitoring function	SLS
Position actual value [mm]	0
-120	120

Collect data

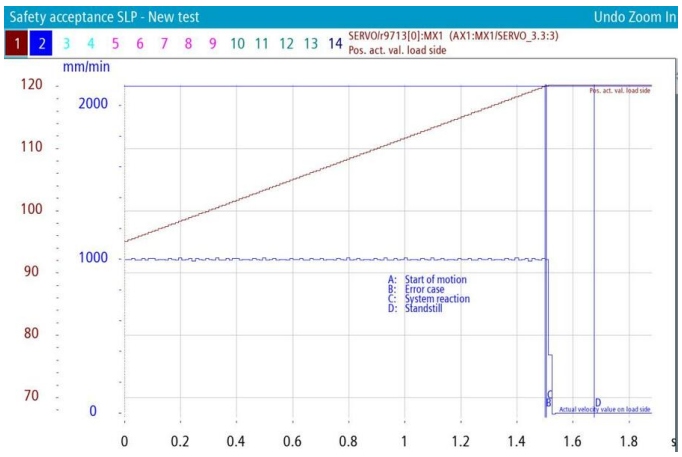
Data acquisition running...

Traverse the axis/drive (in JOG for NC axes) in the direction of the selected SLP limit value until the SLP limit value is exceeded.

- Jog the axis in a positive direction towards until the second SLP is reached (+120 mm)

Go to results → View trace

The trace shows that the axis was stopped when the +120mm limit is reached (red trace):



Back — Review the data and enter the results.

Next test

Test name: X SLP2 Positive

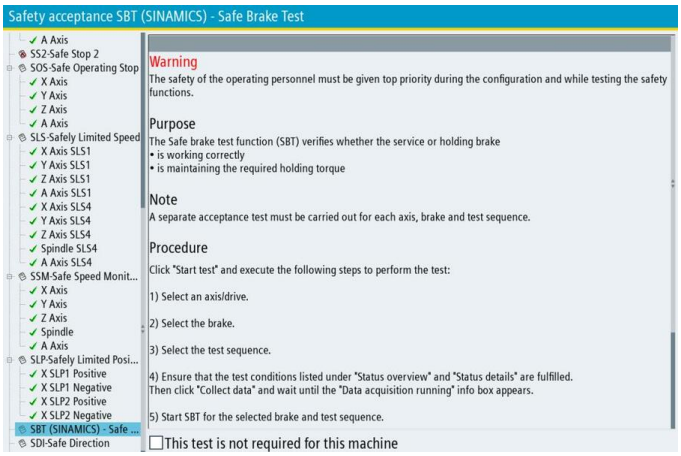
Description	Data
Position range	SLP2
Tested limit value	Upper limit value (positive direction)
Limit value	120 mm
Reaction time	0.004 s
Overtravel from limit value overshoot	0.216 mm

		Coll. alarms
☑	230708	Axis MX1 SERVO_3.3:3 (3), Component Motor_Module_3 (3): SI Motion P2: STOP C initiated.
☑	230715	Axis MX1 SERVO_3.3:3 (3), Component Motor_Module_3 (3): SI Motion P2: Safely-Limited Position exceeded. 20.
☑	201715	Axis MX1 SERVO_3.3:3 (3): SI Motion P1 (CU): Safely-Limited Position exceeded. 20.
☑	201708	Axis MX1 SERVO_3.3:3 (3): SI Motion P1 (CU): STOP C initiated.
☑	27008	Axis MX1 SW limit switch deactivated
☑	230799	Axis MX1 SERVO_3.3:3 (3), Component Motor_Module_3 (3): SI Motion P2: Acceptance test mode active.

Criteria for initiating and evaluating the test: Axis enabled, Jog mode selected, SLP range 2 active, Jog in a positive direction when prompted.

Trace preview

The test was successful



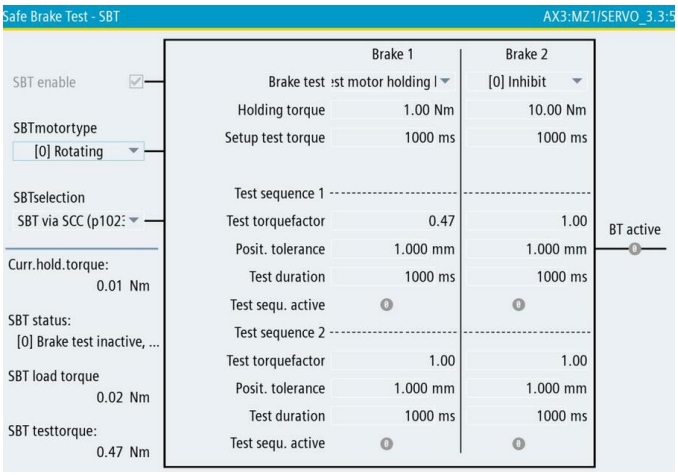
- Remove the User Agreement
- Jog the axis away from the limit
- Set the User Agreement
- Repeat the SLP2 test in the negative direction.

Notes:

- **SBT - Safe Brake Test:**

On the training equipment, the Z axis motor is fitted with a brake.

Functions → SBT



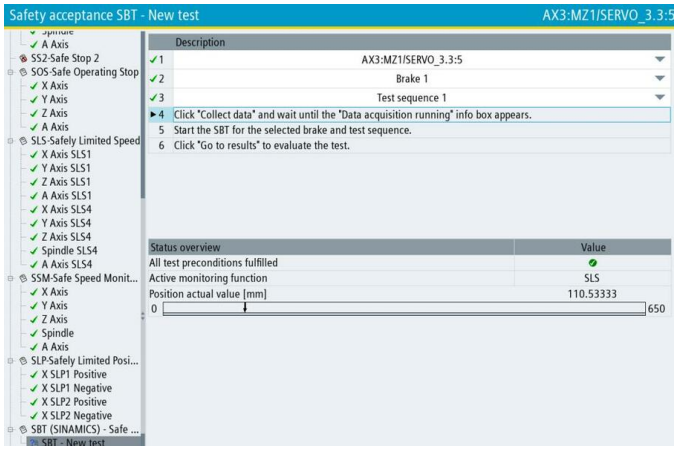
The Safe Brake Test is part of the Test Stop routine.

Start test

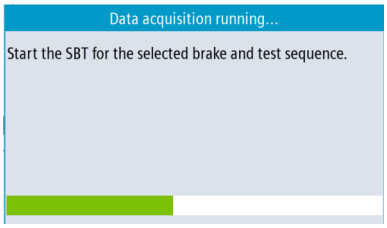
Choose the required options for the SBT:

- Z axis
- Brake 1
- Test sequence 1
- Positive direction

Check the ***Status details*** to confirm the correct axis state



Collect data



Starting the Safe Brake Test:

The Safe Brake Test is started at the beginning of the Test Stop routine.
The Test Stop routine is started with I100.1

- Switch input I100.1 on to start the Test Stop routine.
- Allow the test to complete and switch the input off.

Go to results → View trace

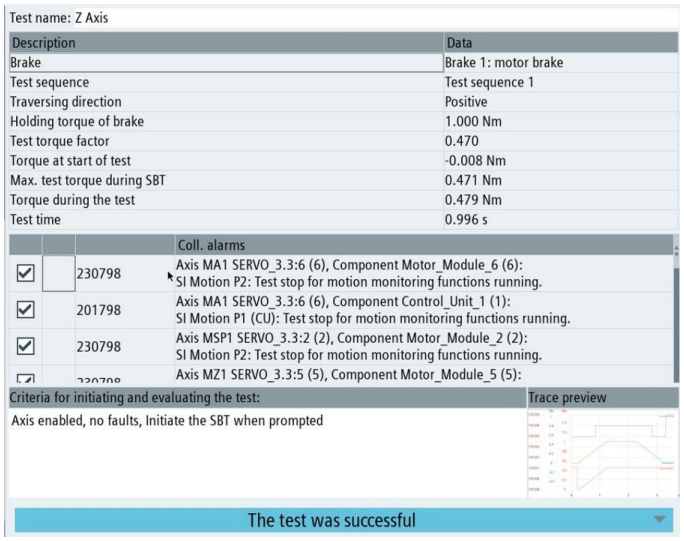
The trace shows that the test torque of 0.47 Nm was applied for 1 second (green trace).
The axis moved slightly (0.001 mm) during the test (brown trace).

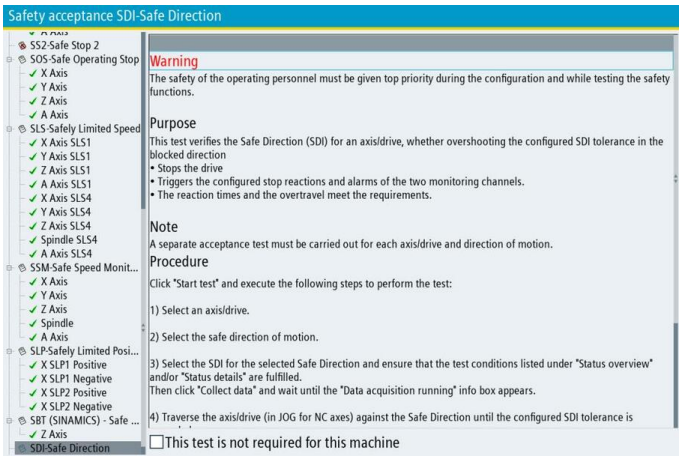


Notes:

Back

Review the data and enter the results.





- **SDI - Safe Direction:**

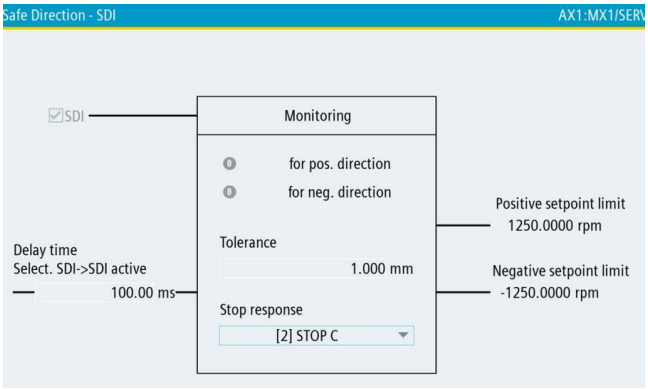
Note: for the training course, the safe direction has been enabled only on the X axis.

The training example uses
I100.2 for SDI+
I100.3 for SDI-

Next test

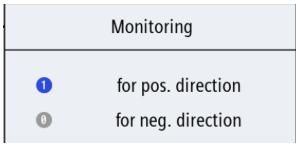
The SDI status can be observed:

Functions > SDI



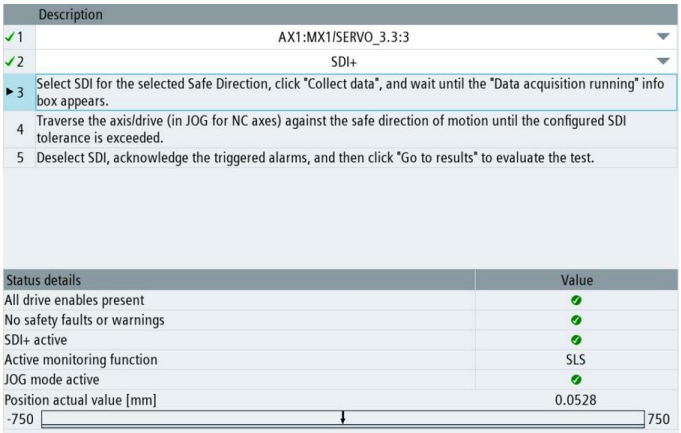
SDI+ acceptance test:

- Switch on SDI+ with I100.3 = ON, Switch off SDI– with I100.2 = OFF



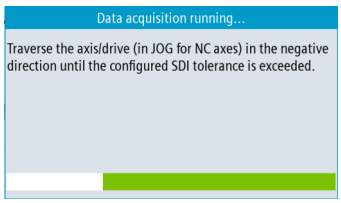
Start test

Set the parameters for the SDI+ direction and check that the conditions for the test are ok:



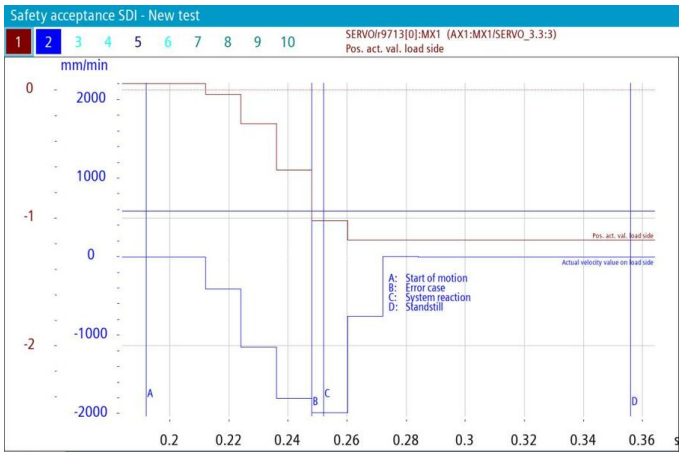
Collect data

Notes:



- Jog the axis in the ***negative*** direction.

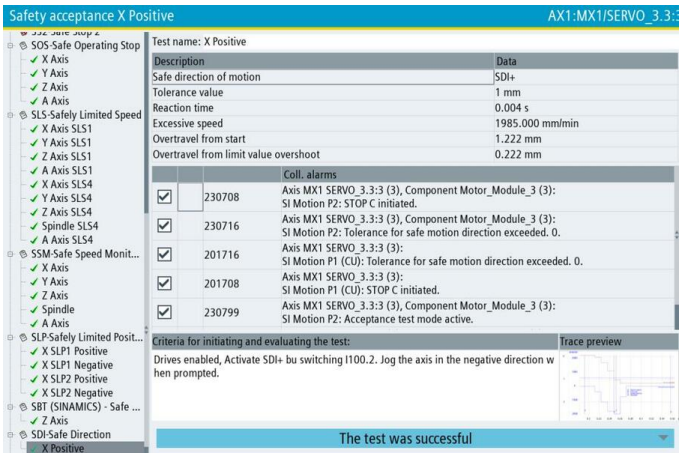
Go to results → View trace



The trace shows the axis was stopped (blue trace) when the SDI tolerance (brown trace) of -1mm was reached.

Back

Review the data and enter the results.

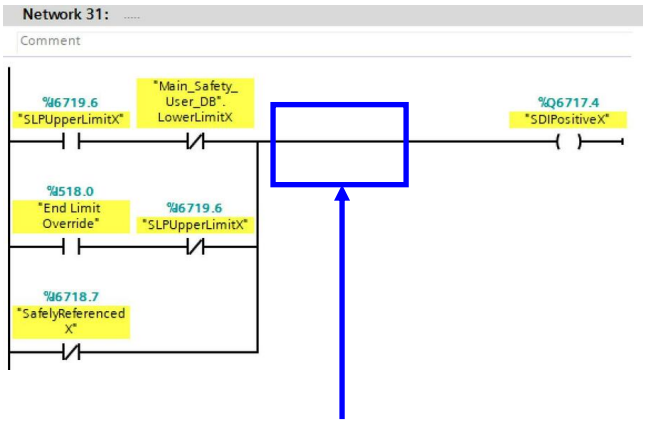


SDI- acceptance test:

The SDI PLC logic was entered earlier in the document. The machine manufacturer will need to facilitate the overriding of the positive and negative limits to allow the SDI function to be tested as part of the SIAT.

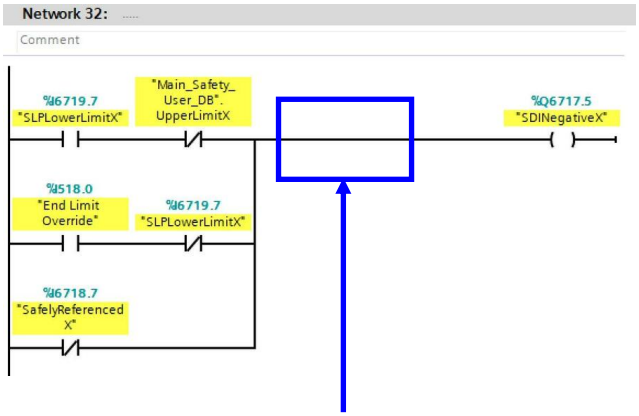
Example - The SDI PLC logic (added earlier)

SDI Positive:



Signal used to override the SDI positive signal. E.G. I100.2

SDI Negative:

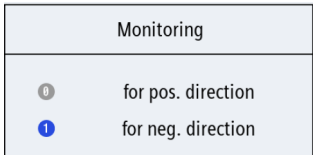


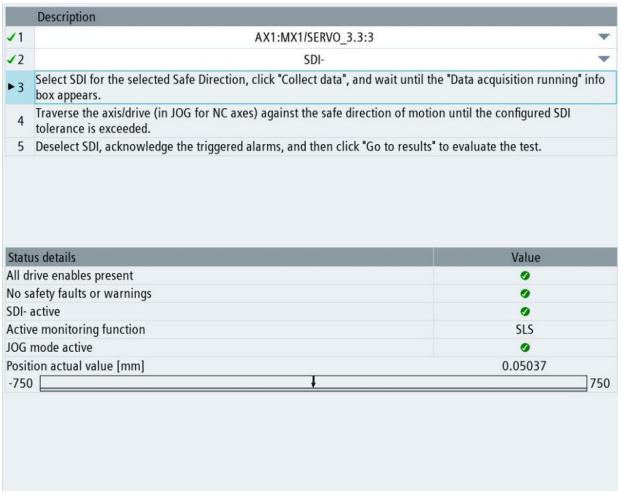
Signal used to override the SDI negative signal. E.G. I100.3

Notes:

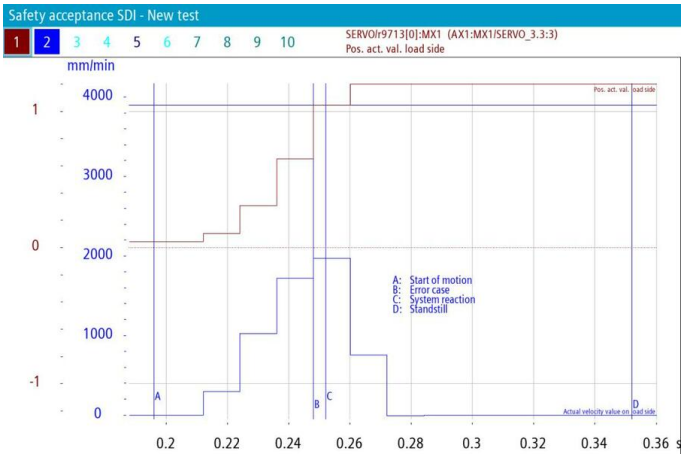
SDI- acceptance test:

- Repeat the test (jog in the ***positive*** direction. I100.2 = ON I100.3 = OFF)

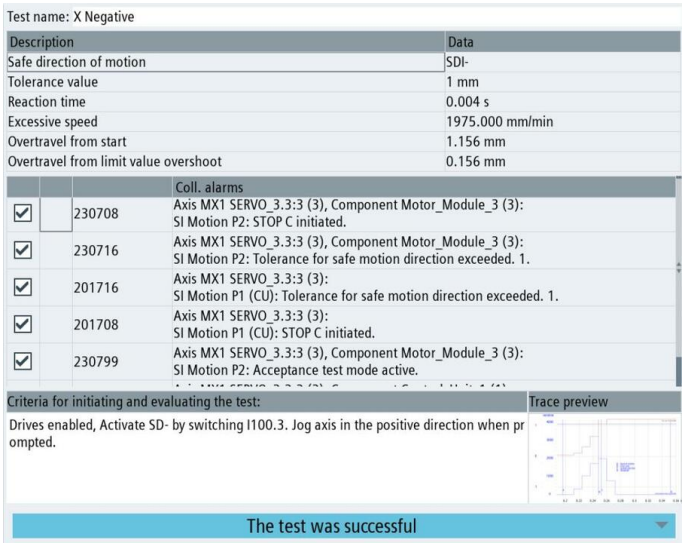




The trace shows the axis was stopped (blue trace) when the SDI tolerance (brown trace) of +1mm was reached.



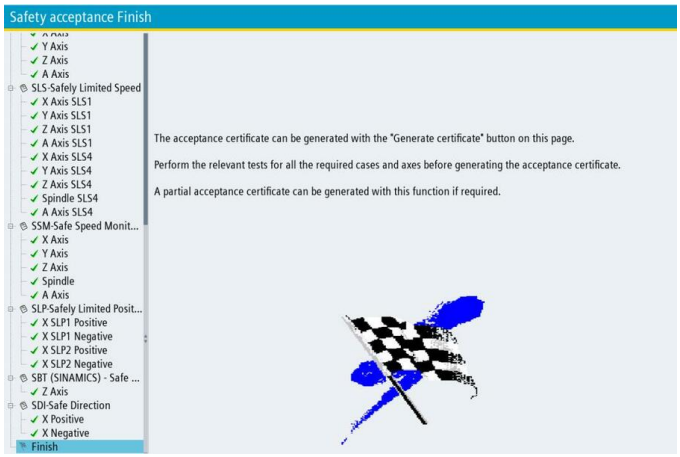
Review the data and enter the results.



Acceptance Certificate:

The SDI test is the final test.
When the Next test soft-key is selected the opportunity to generate the certificate is available.

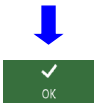




Notes:









The Certificate file should now be edited to include machine specific information:

- Section 1.1.1 Machine diagram detailing the Safe Axes/ Spindles and the location of all safety related devices (Emergency Stops, Guard Door switches….etc).
- Section 2.1 Function Table description.
- Section 2.3 Description of all Safety Related Devices.
- Section 4.4 Data Archiving.
- Section 4.5 Counter Signatures.

Important:

Enter the original values in machine data: that were changed. Return the PLC logic to it's original state.

A219: END

Notes:

Notes:

Appendix G

SINUMERIK ONE

SINUMERIK Safety Integrated F_PLC Standards and Theory

Module Description:

This documentation provides an overview of the regulations and standards as well as the basic safety information.
The basics are described for the "SINUMERIK Safety Integrated F_PLC" functionality for machine tools with safe motion monitoring in the SINAMICS (Drive-Based Safety), the activation by means of a failsafe program in the F-PLC of the SINUMERIK based on TIA Portal V19 and the PROFIsafe communication.
Using the training control, an example is displayed of the documentation together with the function table to be created by the machine manufacturer.

Module Description:

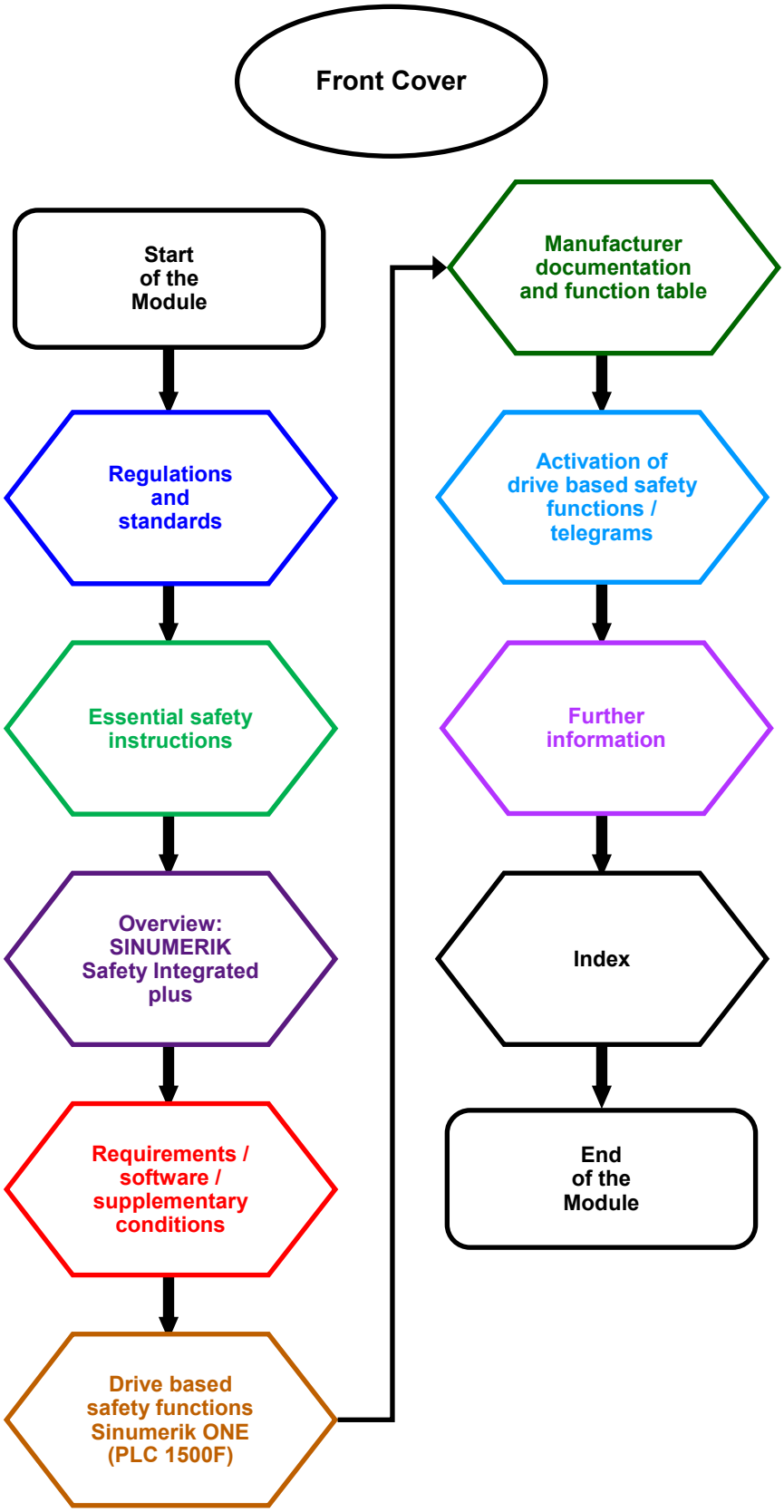
At the end of the module, you will be familiar with the basics of the "SINUMERIK Safety Integrated F_PLC" functionality as prerequisite for the creation of an F-PLC project and the commissioning of the safe
motion monitoring in the SINAMICS.

Content:

- Regulations and standards
- Basic safety information
- Overview: SINUMERIK Safety Integrated F_PLC
- Requirements / software / supplementary conditions
- Drive-integrated safety functions for Sinumerik ONE with PLC 1500-F
- Manufacturer documentation and function table
- Activation of the drive-integrated SI functions / PROFIsafe/PROFIdrive telegrams

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Regulations and standards
Essential safety instructions
Overview: SINUMERIK Safety Integrated plus
Requirements / software / supplementary conditions
Drive based safety functions Sinumerik ONE (PLC 1500F)
Manufacturer documentation and function table
Activation of drive based safety functions / telegrams
Further information
Index
End of the Module

Regulations and standards:

Introduction:

Manufacturers and operators of technical equipment are responsible for **minimizing the risk to plants, machines and other technical equipment according to the latest state of technology**. The safety technology is intended to keep the risk to man and the environment through technical equipment as low as possible, **without restricting the industrial production and use of machines more than absolutely necessary**.

Standards:

- Standards specifiy the abstractly formulated protection objectives of the directives.

Harmonized standards:

- Harmonized standards are required in order to prove

Complete application	⇔	Presumption of conformity

conformity.

- Through the use of harmonized standards, the machine manufacturer can prove that the requirements of the harmonized standards have been satisfied. The presumption of conformity then applies.
- Without the use of harmonized standards, the machine manufacturer has to prove in detail that the requirements of the directive have been satisfied. This results in increased costs for the validation of the machine.
- The relevant authorities are obliged to assume compliance with the requirements of the directive when harmonized standards are used.
- They are taken into the national standards without change by all the European Countries and published in the Official Journal of the European Community as harmonized standards for a specific directive.

International safety standards:

- The valid regulations and standards are decisive at the location of use of the machine or plant
- International standards such as the IEC and ISO standards that have been recognized by numerous countries throughout the world (e.g. ISO 13849-1 or IEC 62061)
- European standards such as the EN standards in the EU (e.g. EN ISO 13849-1 or EN 62061)
- National standards such as the DIN standards in Germany (e.g. DIN EN ISO 13849-1 or DIN EN 62061)
- Overview of European and international standards



Overview:

There are directives, standards and regulations for nearly all areas of technology. They help to ensure that technical equipment has an adequate level of protection for users and safety for operation.
The observance of standards on its own is not a safety measure. It means that the minimum requirements are satisfied and the latest state of technology and knowledge are observed. In order to achieve functional safety on a machine or plant, the safety-relevant components of the protection and control equipment must function correctly and **in the event of an error** respond is such a way that the plant **remains in a safe state or is brought into a safe state.**
Functional safety means that all components of the machine function correctly and there are sufficient devices to reduce risk.
The deployment of specially qualified technology that satisfies the requirements of the relevant standards is also required.

Notes:

The requirements to achieve **functional safety** are based on the following basic objectives:

- **Avoidance of systematic errors**
- **Control of systematic errors**
- **Control of random errors or failures**

The **level of achieved functional safety** is expressed in the standards through different terms.

- In **IEC 61508, EN 62061, EN 61800-5-2** through the **"Safety Integrity Level" (SIL)**
- In **EN ISO 13849-1** through the **"Performance Level" (PL).**

The safety-relevant standards are divided into types A, B and C.

Type A standards:

IEC 61508	Functional safety
EN ISO 12100	Risk analysis
EN ISO 14121	Risk analysis

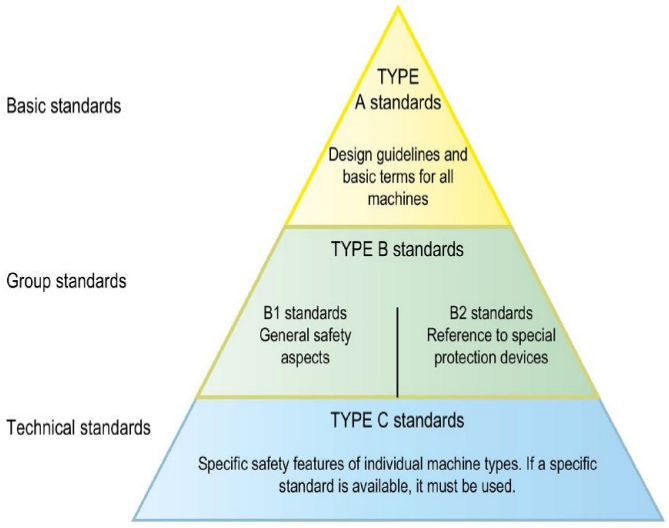
Type B standards:

DIN EN 61800-5-2	Electrical power drive systems with variable speed
IEC 62061	Safety of machines
EN ISO 13849	Safety of machines
IEC 60204	Electrical equipment of machines

Type C standards:

C standards are technical machine safety standards that contain detailed safety requirements for a specific machine or group of similar machines.
If a C standard is available for a relevant machine type, it takes precedence over an A or B standard. If no special C standards exist for the planned machine, the risk reduction must be made in accordance with the A and B standards.

DIN EN ISO 16089	Machine tools – Safety – Stationary grinding machines
ISO 16090	Machine tools – Safety – Machining centers – Milling machines (including drilling machines). (in Europe not just yet valid, but used by OEMs)
DIN EN ISO 23125	Machine tools – Safety - Turning machines
DIN EN 12717	Safety of machine tools and drilling machines



Machinery Directive (2006/42/EC):

- The Machinery Directive is a **European Directive**. Its scope of validity covers the 28 member states (status February 2015) of the European Union. The Machinery Directive is also legally vaid in Switzerland, Lichtenstein, Norway, Iceland and Turkey. In all other countries the Machinery Directive has no legal effect.
- In order that European directives, such as the Machinery Directive, are legally effective in the European member states, they have to be transposed into national law in each member state.
- The Machinery Directive is intended for manufacturers that manufacture products within the Europen Union as well as manufacturers and importers that import machines from outside the European Union.
- **The manufacturer or importer of a machine must provide verification of compliance with the basic requirements of the Machinery Directive.** The following relevant standards for machine safety can be used to meet the requirements of the Machinery Directive.

Notes:

- **The fulfillment of the EU Machinery Directive MRL 2006/42/EU is a prerequisite for the CE marking.**

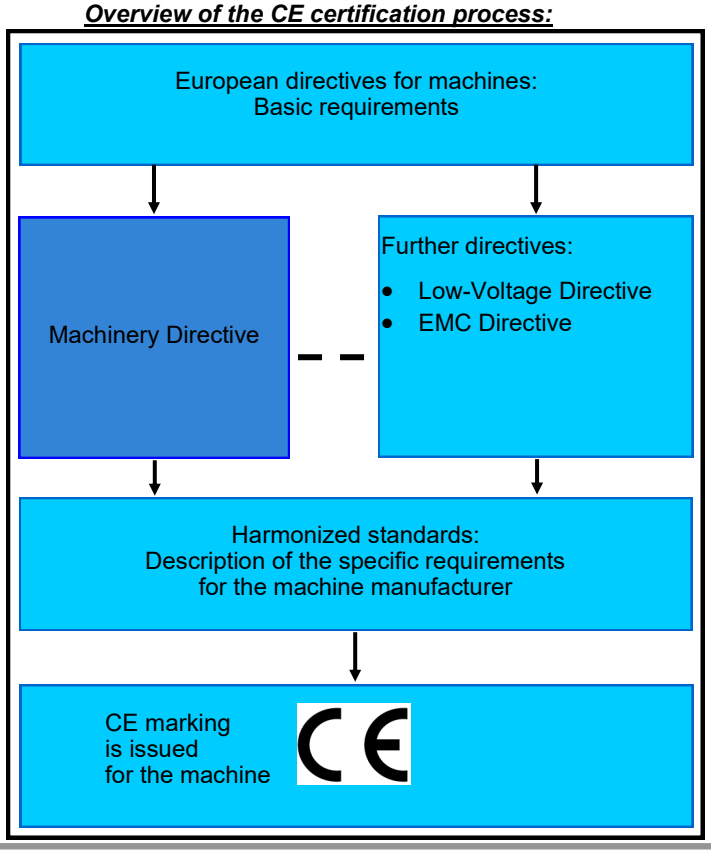
Before placing a machine on the market or commissioning it, the manufacturer must:

- Ensure that the machine satisfies the basic valid health and safety requirements.
- Ensure that the technical documentation is available.
- Ensure in particular that the required information, such as the Operating Instructions, are available.
- Peform the relevant conformity evaluation process.
- Provide the EC Declaration of Conformity and ensure that it is supplied with the machine.
- Apply the CE marking.

The phases required for the ***Implementation of the Machinery Directive*** can be displayed in the following process chain:



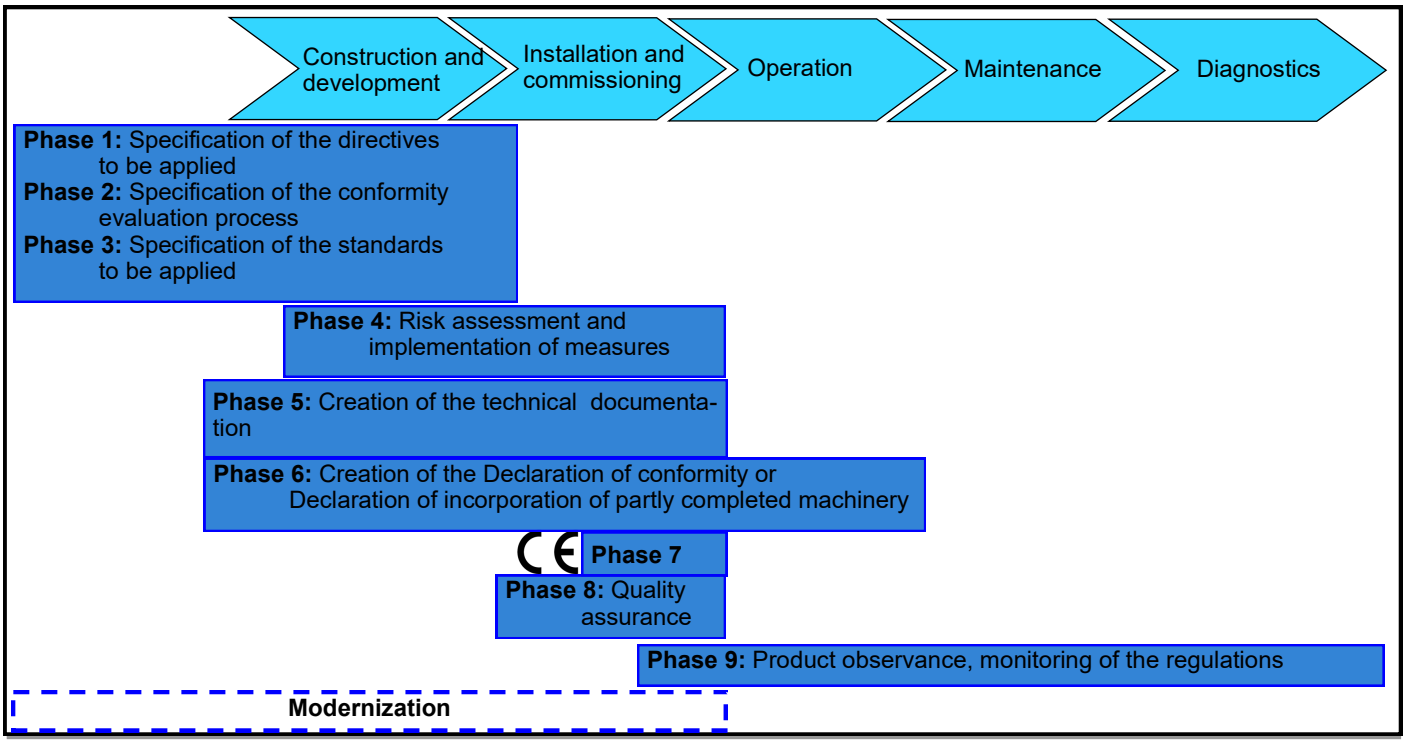
Overview of the CE certification process:



Notes:

CE marking:

The CE conformity process is divided into a series of phases in accordance with the Machinery Directive and describes the individual steps that have to be performed before the CE marking is issued.



Performing a risk assessment:

The EU Machinery Directive requires complete documentation of the risk assessment. Only in this way, can the regulations, conformity evaluations and standards used in phases 1-3 be applied sensibly to the machine.

The first step on the way to a safe machine is therefore the risk assessment – which should already be carried out during the planning phase of the machine.

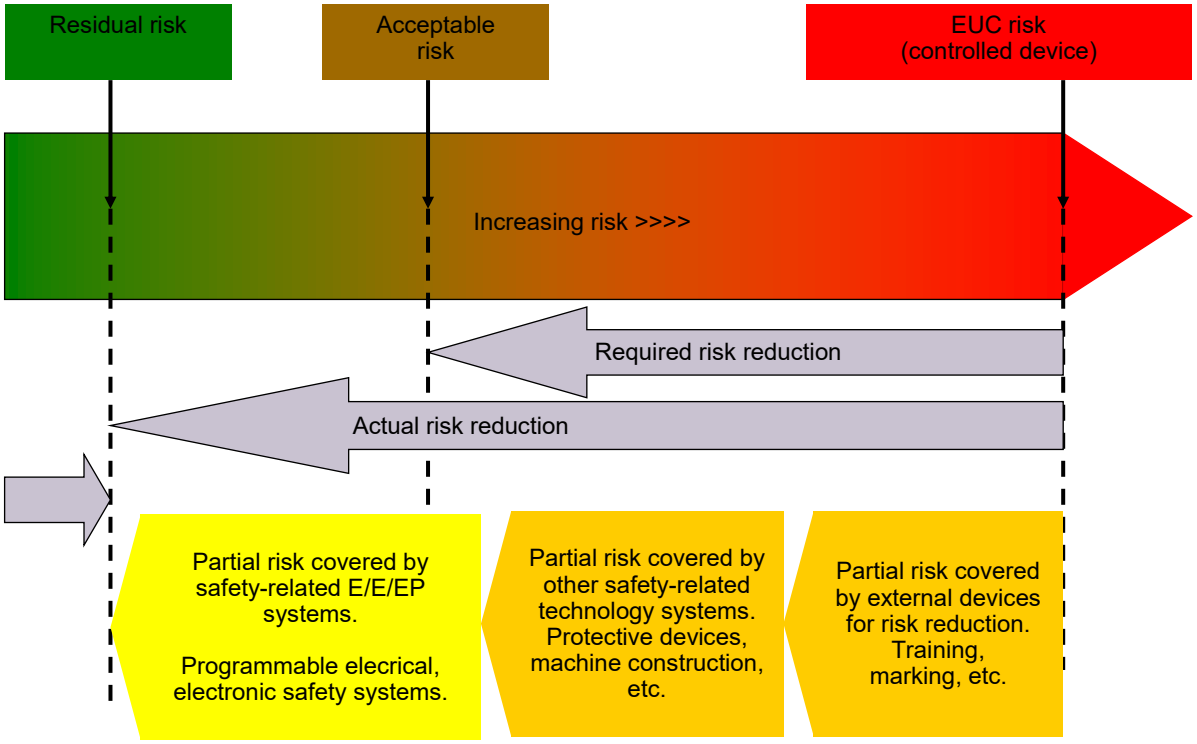
The performance of the risk assessment is a **construction-accompanying process** and should be performed by experts from various fields. In this respect Standard EN ISO 12100 which describes an iterative process for risk assessment is helpful.
A further example: This is covered by standard ANSI B11.0 in the USA.

After the risk assessment, a decision has to be made as to whether measures have to be taken to reduce the risk.
Such a **risk reduction** includes:

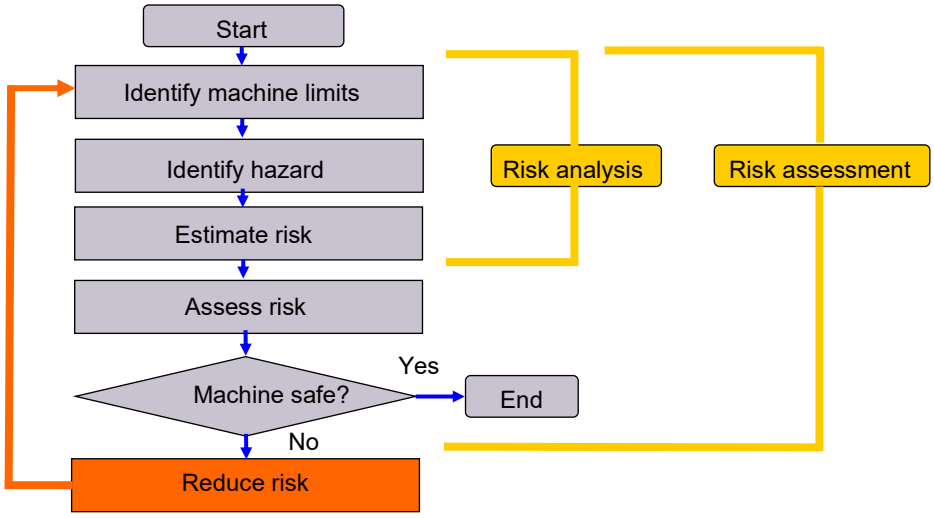
- **Constructive measures**
- **Technical protective devices and**
- **Training for users**

Notes:

Risk reduction:



Iterative risk reduction process according to EN 1050:



Risk reduction and selection of the appropriate safety measures are not part of the risk assessment process. For further information, see EN ISO 12100-1/-2.

Notes:

Relevant standards for machine safety:

IEC 61508:

Functional safety of safety-related electrical / electronic / programmable electronic systems (PLC).

- IEC 61508 is not harmonized in an EC directive. There is therefore no automatic presumption of conformity to satisfy the protection objectives of a directive. However, the manufacturer of a safety technology product can use IEC 61508 to satisfy the basic requirements from European directives according to the new conception.

EN ISO 13849: (Performance Level)

Safety of machines – safety-related components of controls

- **Part 1: General design principles**
- **Part 2 - Validation**
- Its field of application encompasses safety-related components of controls and all types of machines, irrespecive of the technology used (electric, hydraulic, pneumatic, mechanical, etc.).

The following safety-related characteristic values are required for components/devices:

- Category (structural requirement)
- PL: Performance Level
- MTTFd: Mean time to dangerous failure
- DC: Diagnostic coverage
- CCF: Common cause failure

The standard describes the calculation of the Performance Level (PL) based on the PFH value (probability of dangerous failure per hour) for safety-relevant components of controls and complete safety functions based on intended architectures (categories B - 4). The five Performance Levels (a, b, c, d, e) represent the different probability values of a dangerous failure per hour.

For differences, EN ISO 13849-1 refers to EN 61508.

As ISO 13849 considers all types of machines, irrespective of the technology used, this standard is preferred by the manufacturers of machine tools.

IEC 62061: (Safety Integrity Level)

Safety of machines – Functional safety of safety-related electrical and electronic and programmable electronic control systems.

- EN 62061 (identical to IEC 62061) is a sector-specific standard under IEC 61508. It describes the implementation of safety-relevant electrical control systems of machines and considers the entire life cycle from the concept phase to the decommissioning. The basis are the quantitative and qualitative considerations of safety functions.

- Sector-specific standards should take the basic approaches of IEC 61508 and adapt them for the respective field of application, e.g. machine sector, power plant sector, medical sector, etc.

- The IEC 62061 standard describes the risk to be reduced and the capability of a control system to reduce this risk in accordance with the SIL (Safety Integrity Level). Three SILs are used in the machine sector, whereby SIL 1 is the lowest and SIL 3 the highest Safety Integrity Level.

EN 61800-5-2:

Electrical power drive systems with variable speed
Part 5-2: Requirements for the safety – Functional safety.

- This standard specifies requirements and makes recommendations for the design and development, the integration validation of safety-related power drive systems with variable speed (PDS(SR)) with regard to their functional safety.

EN 60204-1:

Safety of machines - Electrical equipment of machines - Part 1: General requirements.

- As a substandard of EN 60204, EN 60204-1 regulates the general specifications and recommendations for the safety, functional capability and maintenance of the electrical equipment on the machine.
- The purpose of the regulation is to avoid hazardous situations and their risks, and to consider safety measures during the construction. It contains general requirements and recommendations for electric, electronic and programmable electronic equipment of machines.



Notes:

Safety Integrity Level:

The result of a risk assessment for a safety function with the aid of IEC 62061 is the Safety Integrity Level (SIL). Depending on the determined SIL, the safety function must have the appropriate PFH value.

Performance Level:

The result of a risk asseessment for EN ISO 13849 is expressed as the Performance Level for the respective safety function. Depending on the determined Performance Level, the safety function must have the appropriate PFH value.

Relationsship between SIL, PL and PFH value:

The following table shows the relationship between Performance Level, PFH value and Safety Integrity Level:

Performance Level (PL) (EN ISO 13849)	PFH value (probability of dangerous failure per hour)	Safety Integrity Level (SIL) (IEC 62061)
a	$\geq 10^{-05}$ to $< 10^{-04}$	-
b	$\geq 3 \times 10^{-06}$ to $< 10^{-05}$	SIL 1
c	$\geq 10^{-06}$ to $< 3 \times 10^{-06}$	SIL 1
d	$\geq 10^{-07}$ to $< 10^{-06}$	SIL 2
e	$\geq 10^{-08}$ to $< 10^{-07}$	SIL 3

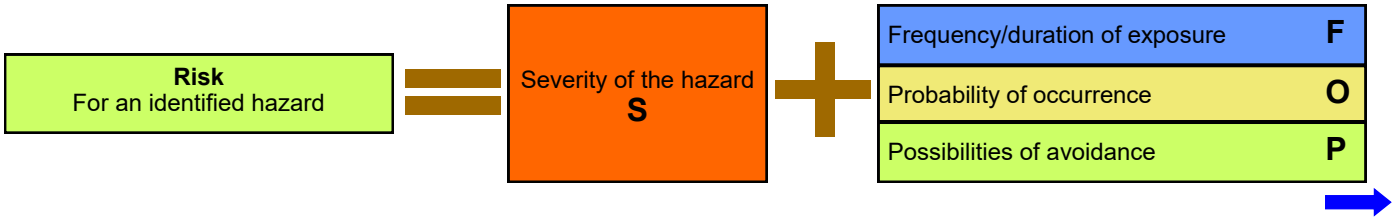
Use of the EN ISO 13849-1 and EN 62061 safety standards:

Conformity with the new Machinery Directive and therefore export capabilty and liability can be achieved by machine manufacturers through the deployment of the EN ISO 13849-1 and EN 62061 standards. They have also introduced quantative aspects in addition to the qualitative considerations, in which "Functional safety" and "Safety function" play an important role. Protective measures for the risk reduction that are described by safety functions can be derived from the risk assessment process. The solution of the safety function with hardware and, if required, software components is then checked and assessed, until the Safety Integrity Level required in the risk assessment is achieved.

After completing the risk assessment, the manufacturer selects the appropriate standard for the machine:

EN 62061 - based on the assignment of a SIL (Safety Integrity Level).
or
EN ISO 13849-1 - based on a PL (Performance Level) through a risk graph.

The ***risk elements S, F, O and P*** serve as input variables for both standards, but are evaluated differently. A required Safety Integrity Level is determined according to EN 62061, a Performance Level (PL) according to EN ISO 13849-1.



Notes:

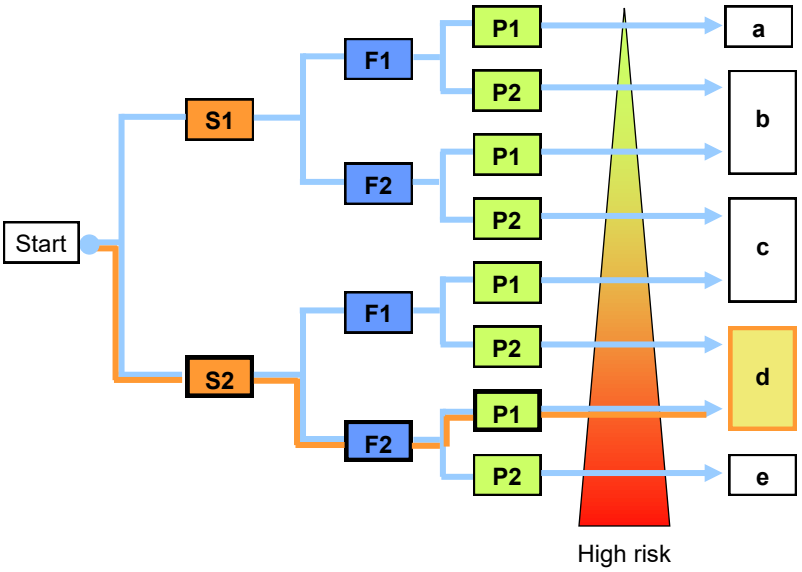
Determination of the required Performance Level (PL) for EN ISO 13849 (PL a - e):

Risk parameters

S = Severity of the injury
- **S1** = slight (usually reversible) injury
- **S2** = severe (usually irreversible) injury, including death

F = Frequency and/or duration of the exposure to the hazard
- **F1** = seldom to often and/or short duration of exposure to the hazard
- **F2** = frequent or continuous and/or long exposure to the hazard

P = Possibility of avoiding or limiting the damage
- **P1** = possible under certain circumstances
- **P2** = hardly possible



Determination of the required Safety Integrity Level (SIL) for EN 62061 (SIL 1, 2, 3):

Frequency and/or duration of the exposure **F**		Probability of occurrence of the hazardous situation **O**		Possibilities of avoidance **P**	
=< 1 hour	5	Frequent	5		
> 1 hour =< 1 day	5	Probable	4		
> 1 day =< 2 weeks	4	Possible	3	Impossible	5
> 2 weeks =< 1 year	3	Seldom	2	Possible	3
> 1 year	2	Negligible	1	Probable	1

Effects	Severity **S**	Class **Cl = F + O + P**				
		3-4	5-7	8-10	**11-13**	14-15
Death, loss of an eye or arm	4	SIL 2	SIL 2	SIL 2	SIL 3	SIL 3
Irreversible, loss of a finger	**3**	Other measures		SIL 1	**SIL 2**	SIL 3
Reversible, medical treatment	2				SIL 1	SIL 2
Reversible, first aid	1					SIL 1

Example:

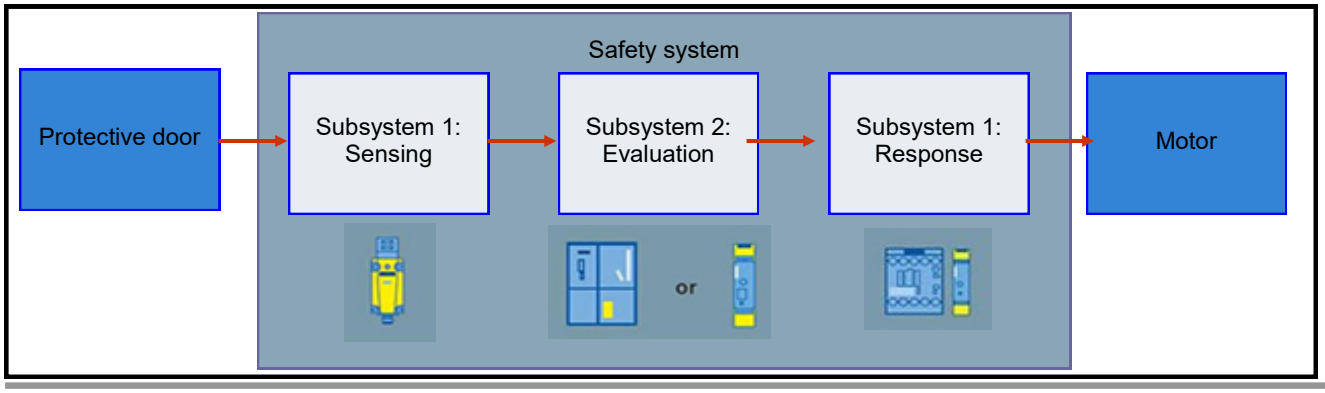
Hazard	S	F		O		P	=	Cl	Safety measures	Safe
Turning spindle	3	5	+	4	+	3	=	12	Monitor protective cover with SIL 2	Yes

Notes:

Evaluation of the safety-relevant equipment:

After determining the respective 'SIL' or 'PL' for the machine, the various safety-relevant equipment that form the different 'Safe systems' of the machine, i.e. emergency stop, protective door, etc. must be selected by the machine manufacturer.
The machine manufacturer is therefore responsible for verifying and documenting that the combination and integration of the various safety-relevant equipment meet the required 'SIL' or 'PL'.

The ***Safety Evaluation Tool*** can be used for this purpose.
The Safety Evaluation Tool for the IEC 62061 and ISO 13849-1 standards leads you directly to your goal.
This TÜV-tested online tool supports you in the quick and reliable assessment of the safety functions of your machine.
The result is a standard compliance report that can be integrated in the documentation as a safety verification.



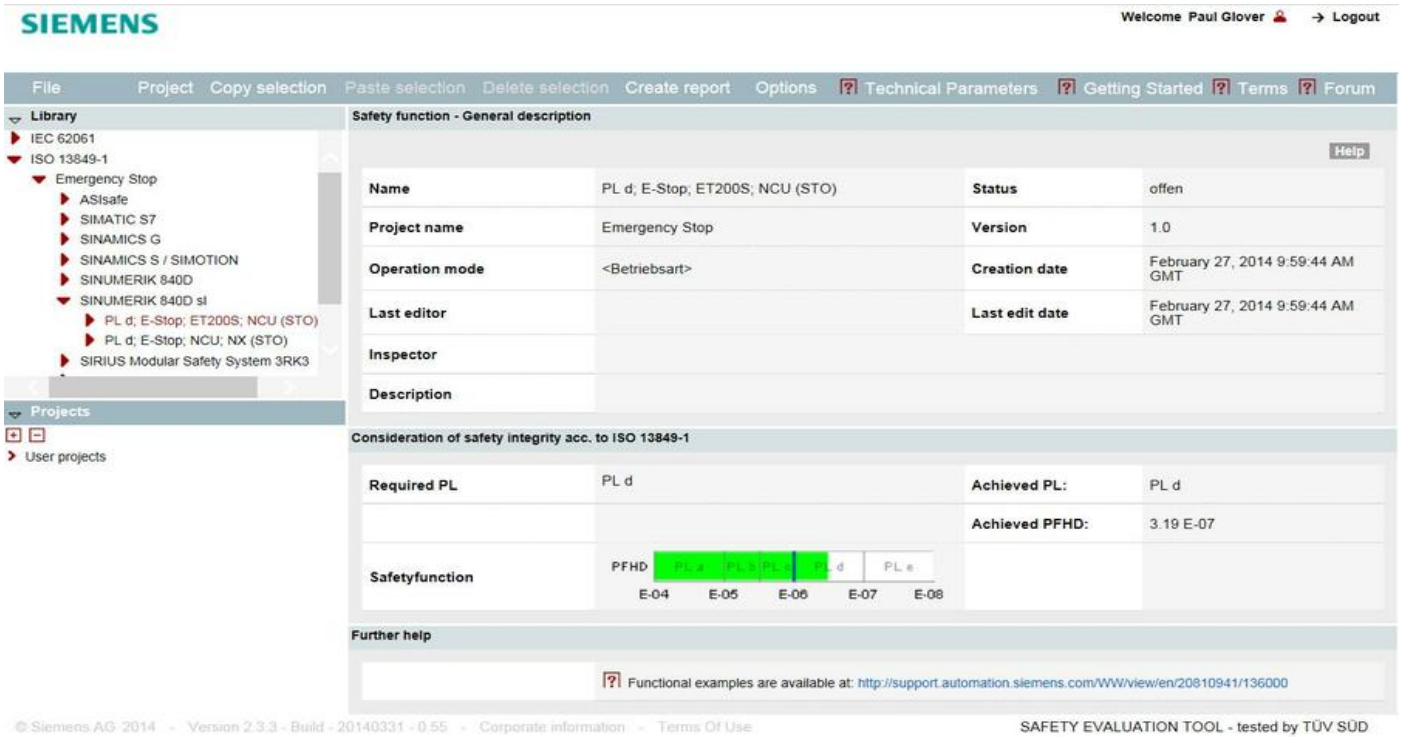
Advantages of using the Safety Evaluation Tool:

- Assurance with regard to the compliance with the standard: TÜV-tested tool
- Free-of-charge use of the online tool
- Automatic calculation in accordance with the current standards
- Quick result: Standard compliance report
- Time saving during the assessment of the safety functions
- Quick access to current product data
- Easy-to-use archiving: Projects can be stored and recalled when required
- Quick and simple handling: Comprehensive predefined example libraries
- Selection menus to determine the DC and the CCF
- Input of different operating cycles in a two-channel structure
- Calculation of the failure rate
- Selection wizard for drive components

For the link to the **Safety Evaluation Tool**, see "Section: Further information > Help for standards"

Notes:

Online Safety Evaluation Tool (SET):



- The Safety Evaluation Tool contains useful examples based on the SINUMERIK ONE. These examples cover IEC 62061 as well as ISO EN 13849-1.
- The tool can create a multi-page report that confirms the Safety Integrity of the safety-relevant components in you project. This report can form a part of the technical file for your machine.
- SET projects can be stored locally and loaded again in the future.
- The tool provides support with online help and tutorials.

Links and further information on the Section "Regulations and standards"
can be found in "Section: Further information > Help for standards"

Notes:

Essential safety instructions:

General safety instructions:

The **"General safety instructions"** of the **"Safety Integrated plus Function Manual"** must be observed.

Equipment damage caused by electric fields or electro-static discharge:

The notes with regard to **"Equipment damage caused by electric fields or electro-static discharge"** of the **"Safety Integrated plus Function Manual"** must be observed.

Warranty and liability for sample applications:

The sample applications described/used in this documentation and in the associated course are not binding and make no claim to be complete with regard to the configuration and equipment as well as any eventualities.

- Sample applications do not show any customer-specific solutions, but rather provide help for typical tasks.
- YOU are responsible for the proper use of the described products.
- The sample applications do not relieve you from the responsibility of safe and professional using, installing, operating and servicing.

Industrial Security:

Note:

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
To protect plants, systems, machines and networks against cyber threats, a holistic, state-of-the-art, industrial security concept must be implemented (and sustained continuously). The products and solutions from Siemens form only part of such a concept.
Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Systems, machines and networks should be connected with the enterprise network or the Internet only when necessary and only to the extent required, and the appropriate protective measures (e.g. use of firewalls and network segmenting) have been adopted.

In addition, the recommendations from Siemens for the appropriate protective measures should be observed. Further information about industrial security can be found at:
Industrial Security
(http://www.siemens.com/industrialsecurity).

The products and solutions from Siemens are being continually enhanced to make them even more secure. Siemens recommends expressly that updates should be performed once the appropriate updates are available and that the current product versions are always used. Using versions that are obsolete or no longer supported can increase the risk of cyber threats.
To be kept informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at:
Industrial Security
(http://www.siemens.com/industrialsecurity).

Notes:



WARNING

Insecure operating states caused by software manipulation:

Manipulations of the software, such as viruses, Trojan horses, malware or worms, can cause insecure operating states in your plant that can lead to death, severe injury and material damage.

- Keep the software up-to-date.
- Integrate the automation and drive components in a holistic industrial security concept of the plant or machine in accordance with the current state-of-the-art.
- Consider all deployed products for your industrial security concept.
- Protect the files on removable storage media against malware by adopting the appropriate protective measures, such as virus scanners.

Residual risks for drive systems (Power Drive Systems):

In accordance with the appropriate local regulations (e.g. EU machine directive), machine manufacturers or plant planners must consider the residual risks that result from the components for the control and drive of a drive system by evaluating the risk for their machine or plant:

1. Uncontrolled movements of driven machine or plant parts during commissioning, operating, servicing and repairing, such as those caused by:
   - – Hardware and/or software errors in the sensory system, control, actuators and connection technology
   - – Response times of the control and the drive
   - – Operational and/or environmental conditions outside the specification
   - – Condensation / conducting contamination
   - – Errors with the parameterization, programming, wiring and assembly
   - – Use of radio equipment / cellular phones in the immediate vicinity of electronic components
   - – External influences / damage
   - – X-ray, ionizing and cosmic radiation
2. In the event of a fault, unusually high temperatures can arise inside and outside components, including naked flames, as well as emissions of light, noise, particles, gases, etc., such as those caused by:
   - – Component failures
   - – Software errors
   - – Operational and/or environmental conditions outside the specification
   - – External influences / damage
3. Dangerous contact voltages, such as those caused by:
   - – Component failures
   - – Influence during electrostatic charging
   - – Induction of voltages for moving motors
   - – Operational and/or environmental conditions outside the specification
   - – Condensation / conducting contamination
   - – External influences / damage
4. Operational electric, magnetic and electromagnetic fields that can be dangerous for persons with heart pacemakers, implants or metallic objects, etc. too near the equipment
5. Release of environmentally polluting materials and emissions in the event of improper operation and/or improper disposal of components
6. Effect of network-based communications systems, e.g. ripple-control transmitters or data communication over the network

Further information about the residual risks that emanate from the components of a drive system can be found in the appropriate sections of the technical user documentation.



Notes:

Overview: SINUMERIK Safety Integrated plus:

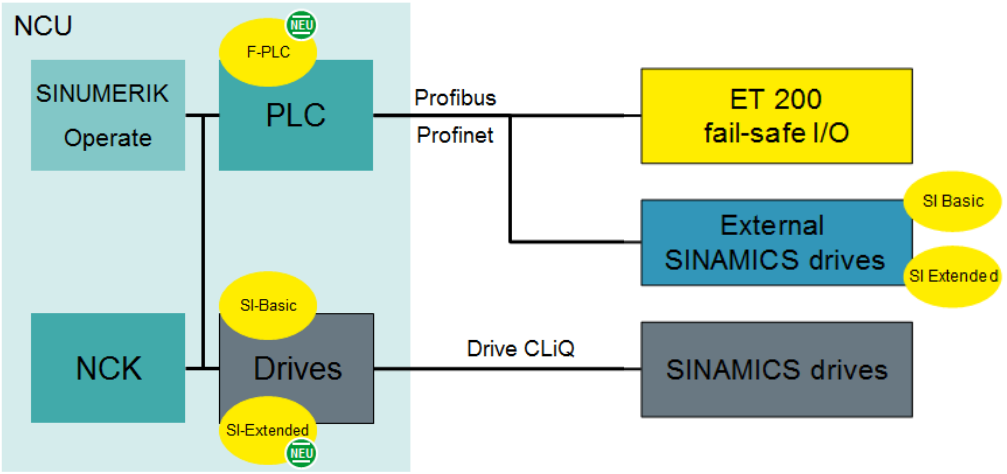
As of SINUMERIK NC software version V6.21, **SINUMERIK Safety Integrated F_PLC for the Sinumerik ONE with integrated F-PLC** can be used as a future-oriented alternative to the previous "Safety Integrated SPL (Safe Programmable Logic)". This means that, amongst other things

- The motion monitoring is performed exclusively through **drive-based safety** with motion control functionality in the SINAMICS.
- This concept corresponds to the motion monitoring for the SINUMERIK 828.
- The Safe Programmable Logic (SPL) is omitted and only a failsafe program has to be written in the F-PLC.
- The same safety concept is used for axes that are "assigned directly to the NCU" and for "external axes".

- SI-Basic = Basic Safety Functions of the SINAMICS (STO; SBC; SS1)
- SI-Extended = Extended Safety Functions of the SINAMICS

Licensing
- **SINAMICS drives**
- **External SINAMICS drives**

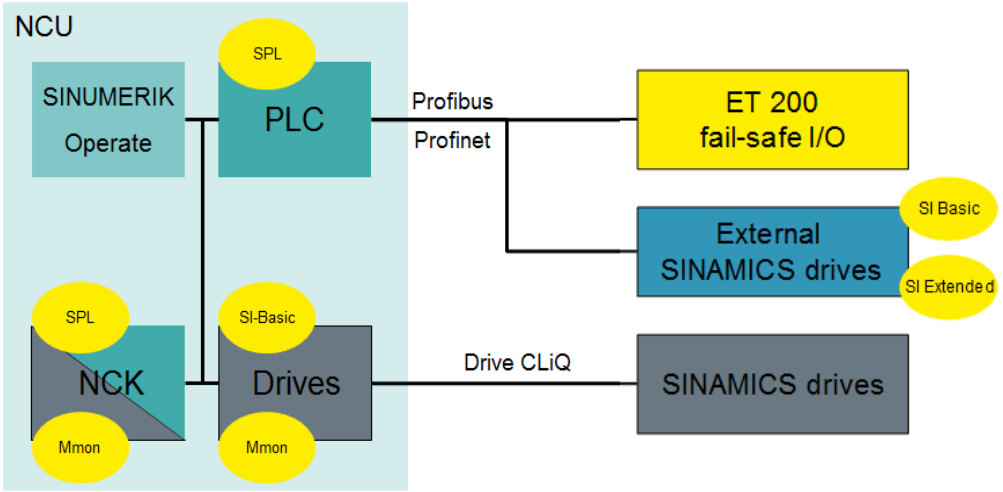
see Licensing of axes/spindles and F-PLC



As a comparison, the previous system-integrated Safety Integrated **concept with "Safe Programmable Logic (SPL)"** is displayed with the following properties:

- With Safety SPL, the motion monitoring for NC axes is distributed between the NCK and the drive
- Two failsafe programs are required:
  - Asynchronous subprogram in the NCKSTEP
  - STEP 7 program in the PLC (no failsafe program)
- Different safety concepts for axes that are "assigned directly to the NCU" and for "external axes".

- SI-Basic = Basic Safety Functions of the SINAMICS (STO; SBC; SS1)
- SI-Extended = Extended Safety Functions of the SINAMICS
- Mmon = Motion Monitoring

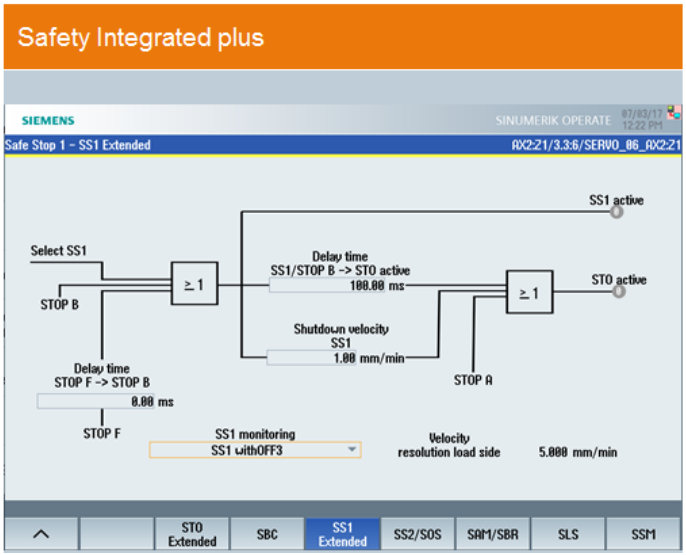


Notes:

Characteristics of SINUMERIK Safety Integrated F_PLC with integrated F-PLC:

- The safety technology does not run in the NCK of the SINUMERIK. Consequently, no "Safe.spf" NC program is required.
- In the integrated F-PLC, not only the "non-safety-relevant cyclical Step 7 program", but also the failsafe program is processed as a safe Step 7 program.
- The safety functions in SINAMICS are activated by PROFIsafe telegrams. This means, with regard to the motion control, the same safety concept for axes assigned directly to the NCK, as well as for the axes coupled externally via PROFIBUS/PROFINET, is applicable.
- SINUMERIK Safety Integrated plus" can be configured only with TIA Portal.
- As of TIA Portal V14 SP1, "Safety Integrated SPL" can also be configured with TIA Portal, so there is a consistent solution for "Safety Integrated SPL" and "SINUMERIK Safety Integrated plus". "Safety Integrated SPL" can still also be configured with the Step 7 Classic V5.x and Step 7 F configuration pack.

- For SINUMERIK Safety Integrated plus, an easily understandable screen form for the graphically supported commissioning in which the commissioning engineer can enter the individual parameters is available for each safety function. These screen forms largely correspond to the Safety Integrated screen forms of the STARTER software for commissioning external SINAMICS drives.

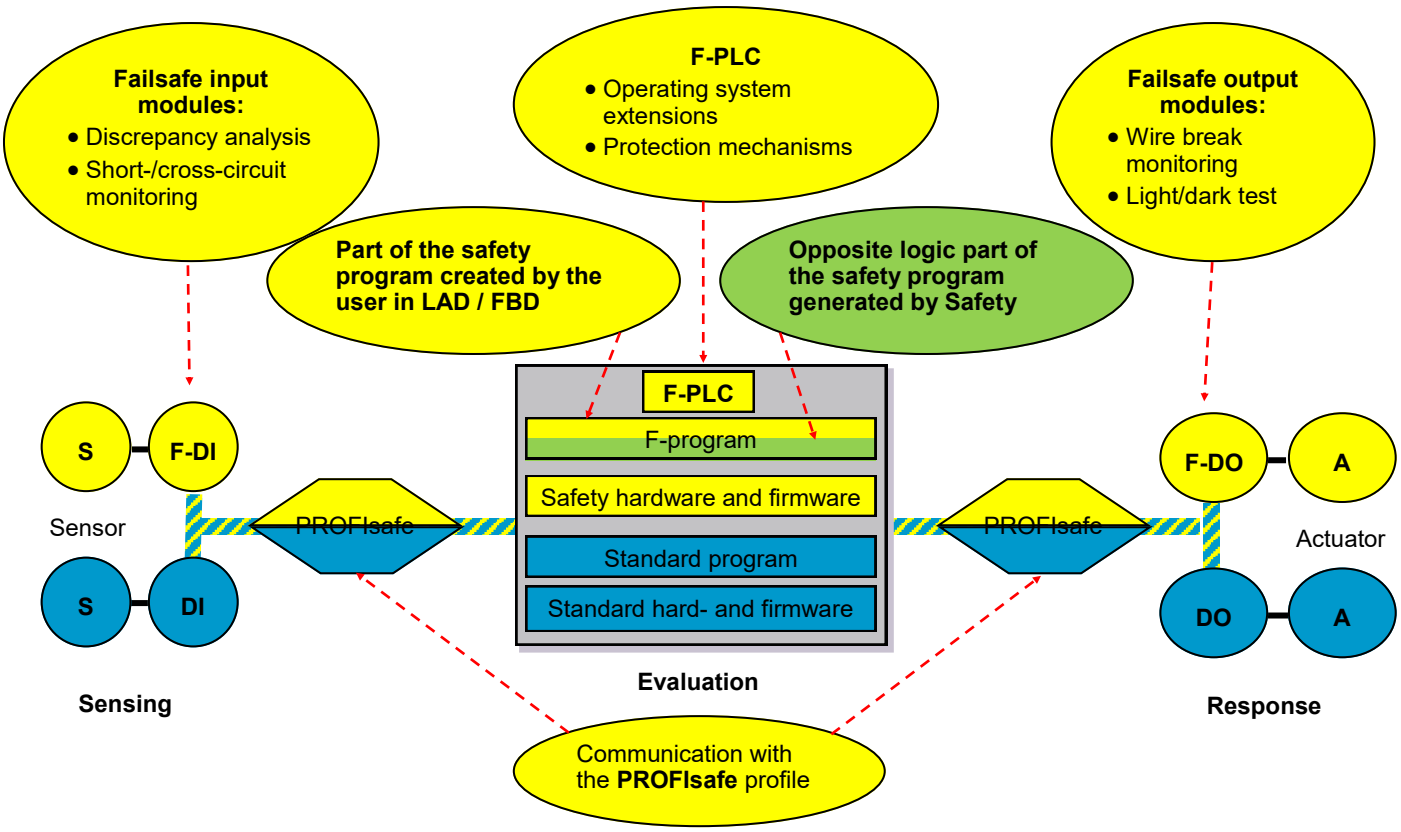


- The "Safety Integrated acceptance test" to be performed prior to the delivery is available in SINUMERIK Operate.
- Certified SIMATIC F-blocks are available, for example, for emergency stop.
- Depending on the deployed PLC, SINUMERIK Safety Integrated plus allows as many as 32 secure connections to be implemented.
- SINUMERIK Safety Integrated F_PLC provides new functions. Safe rotational direction, e.g. for tools that can machine in just one direction, encoderless safe speed monitoring, e.g. for accelerating spindles, and safe position, e.g. for a large number of safe cams.
- Compatibility and migration:
  Because of the new technology, the new motion control and the new failsafe program, a migration of the previous "Safety Integrated SPL" concept to "SINUMERIK Safety Integrated plus" is not possible. The user must reconfigure and reprogram. A conversion to "SINUMERIK Safety Integrated plus", however, is not essential, because the previous SPL concept will remain available while the SINUMERIK 840D sl supply is continued. For the future control generation, there will then be only the "SINUMERIK Safety Integrated plus" concept and a reconfiguring/reprogramming is unavoidable.

Safety concept: (see following representation)

- ***Hardware and firmware extensions:***
  The safety-oriented F-PLC combines the functionality of a standard PLC with that of a safety PLC. The operating system extended with protective mechanisms allows standard and safety-oriented user programs to be executed on a PLC.
- ***F-program***:
  The safety program for controlling safety-oriented functions consists of the part created by the user in FBD or LAD and a part generated by Safety Advanced that contains the diverse logic (with diverse operands/operations see "Further detailed information") for the user part, etc. The standard and safety-oriented user programs are created in the same programming environment. TÜV-certified function blocks for all common safety functions further simplify the programming.

Notes:



Safety-relevant processing with NCU components:

Safety-oriented internal and external signals in the safety-oriented user program (safety program) of the F-PLC are processed in SINUMERIK Safety Integrated F_PLC (F-PLC) mode:

- The F-I/O are connected to the F-PLC and activated via PROFIsafe:
  The safety functions are contained principally in the safety program (F-PLC) and in the fail-safe inputs and outputs (F-I/O).

- Drive-independent safety functions are also activated by PROFIsafe:
  For this purpose, the PROFIsafe telegrams (SIEMENS telegram 902) are transferred on the DP Integrated interface from the PLC to the SINAMICS Integrated (DRIVE).

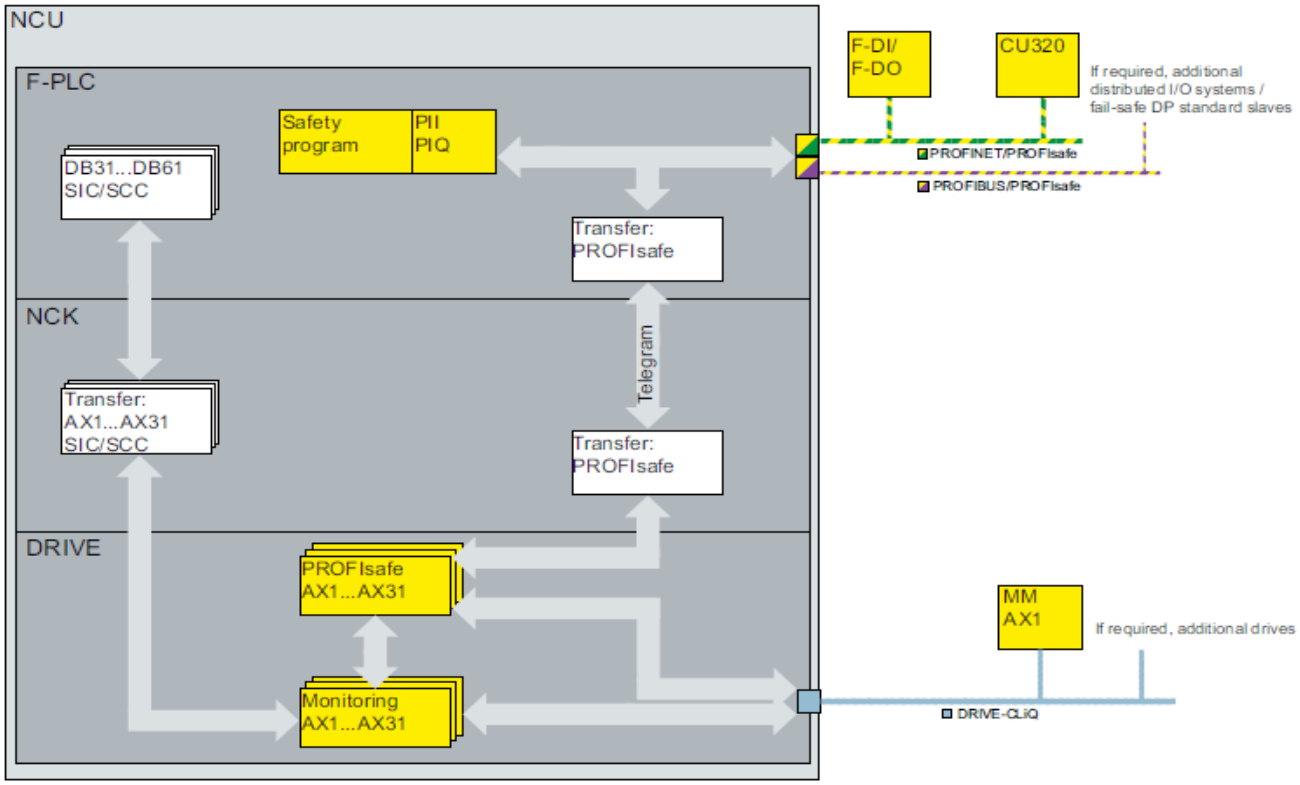
Status and control information is transferred with SIC/SCC (Safety Info Channel / Safety Control Channel) between integrated subcomponents of the NCU (PLC, NCK and SINAMICS Integrated).

PROFIsafe communication is possible only between F-components.
Consequently, PROFIsafe communication for the SINUMERIK NCU is performed between the following components:

- F-PLC as PROFIsafe host
- SINAMICS Integrated (DRIVE) as PROFIsafe slave/device
- F-I/O as PROFIsafe slave/device

Notes:

Processing and communication of safety-relevant information through the components of the NCU:



Subcomponents of NCU:

F-PLC **PROFIsafe Host**	• Processing of safety-related internal and external signals in the safety-related user program (safety program). See "Section: Safety programming of SINUMERIK F-PLC" • Processing of status and control signals (SIC/SCC) via the appropriate axis-DB (Axis1 [DB31] ... Axis 31 [DB61]). See "Section: Control of drive-integrated SI functions / telegrams > Signals to the axis / signals from the axis"
NCK	• Communication of status and control information (SIC/SCC, axis-DB) between F-PLC and DRIVE. • Communication of PROFIsafe telegrams (safety program, PII/PIQ) between F-PLC and DRIVE. • Read-in of safety-relevant messages from the message buffers of the SINAMICS Integrated to the alarm buffer of the NCK.
HMI	• Display of the safety-related messages/alarms from the NCK alarm buffer in a Safety Integrated diagnostics overview. See HMI Operate > Diagnostics > Safety • Read-out and display of further safety-relevant diagnostics information directly from the drive. • Display of a Safety deletion symbol for pending Safety messages.

Notes:

DRIVE **(SINAMICS Integrated)** **PROFIsafe Slave**	• Safety-related motion monitoring of NC-controlled drives (on the DRIVE-CLiQ-interface). See "Section: Commissioning, F-PLC, diagnostics: STO, SS1, SS2, SS2e, SOS, SAM, SSM" • Execution of the error response functions or user safety functions.

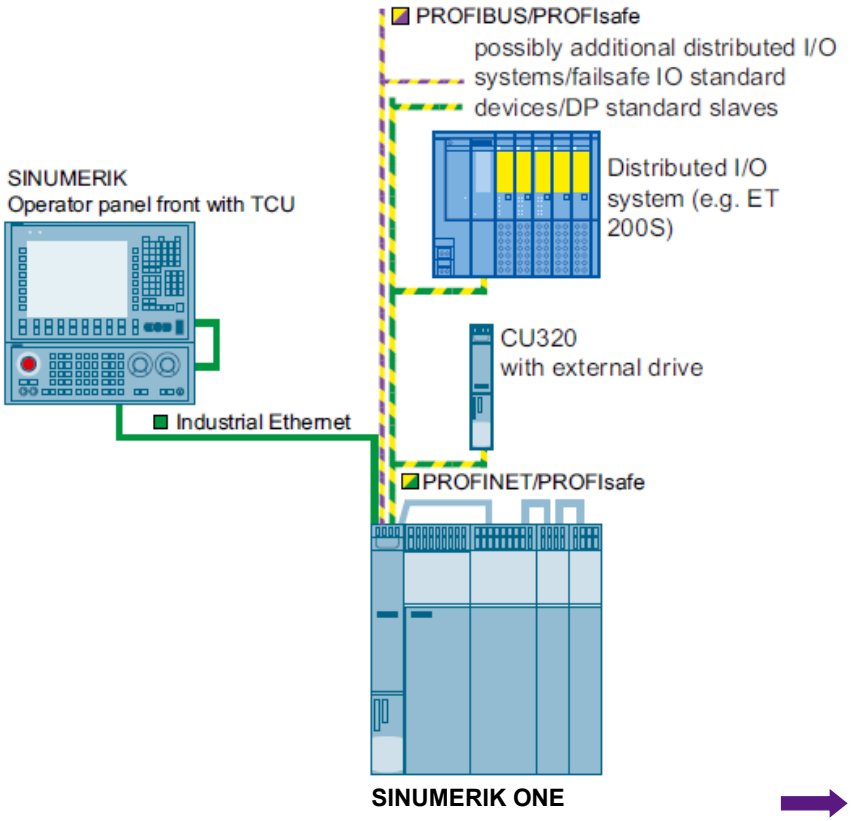
I/O and drives:

F-I/O **PROFIsafe Slave**	• The F-I/O ensures the safe processing of the field information (sensors: e.g. EMERGENCY STOP button, light barriers; actuators, e.g. motor activation). The safety function for the process can be provided by a user safety function or an error response function. See "Section: Safety programming of SINUMERIK F-PLC"
NC-controlled drives	• Drive-independent safety functions that communicate with the process via safety-related input/output signals, can be implemented for each individual axis and spindle. See "Section: Commissioning, F-PLC, diagnostics: STO, SS1, SS2, SS2e, SOS, SAM, SSM"

Hardware components:

The SINUMERIK Safety Integrated F_PLC (F-PLC) safety solution places specific demands on the hardware components to be used.
Failsafe SIMATIC and standard I/O for PROFIBUS DP or PROFINET IO can be used irrespective of the deployed scope of Safety Functions, as known from SIMATIC Safety. Which drive types can be used depends on whether a safety-related encoder is used and which axis type is to be used.

A possible structure with failsafe hardware components is shown in an overview in the adjacent figure.



Notes:

F-I/O for PROFIBUS DP:

The following failsafe components can be deployed on the PROFIBUS DP in SIMATIC Safety failsafe systems:

- F-CPUs with DP interface
- Failsafe Inputs/outputs (F-I/O), e.g.:
  – S7-300 failsafe signal modules in ET 200M
  – ET 200MP failsafe modules
  – ET 200SP failsafe modules
  – ET 200S failsafe modules
  – ET 200pro failsafe modules
  – ET 200iSP failsafe modules
  – ET 200eco failsafe I/O modules
  – Failsafe DP standard slaves (light grid, laser scanner etc.)

The structure can be extended with standard I/O. The following CPs can be deployed on the PROFIBUS DP in a SIMATIC Safety failsafe system:

- CP 443-5 Basic
- CP 443-5 Extended
- CM 1542-5

F-I/O for PROFINET IO:

The following failsafe components can be deployed on the PROFINET IO in SIMATIC Safety failsafe systems:

- F-CPUs with PN interface
- Failsafe Inputs/Outputs (F-I/O), e.g.:
  – S7-300 failsafe signal modules in ET 200M
  – ET 200MP failsafe modules
  – ET 200SP failsafe modules
  – ET 200S failsafe modules
  – ET 200pro failsafe modules
  – ET 200iSP failsafe modules
  – Failsafe IO standard devices (light grid, laser scanner etc.)

The structure can be extended with standard I/O.
The following CPs can be deployed on the PROFINET IO in a SIMATIC Safety failsafe system:

- CP 443-1
- CP 443-1 Advanced-IT

Drive components:

Which drive types can be used depends on whether a safety-related encoder is used and which axis type is to be used.

The following table shows the supported axis types depending on the drive type and the safety-related encoder.

	Spindle	Rotary axis	Linear axis
Asynchronous motor with encoder	**Yes**	**Yes**	**Yes**
Asynchronous motor without encoder	**Yes**	**No**	**No**
Synchronous motor with encoder	**Yes**	**Yes**	**Yes**
Synchronous motor without encoder	**No**	**No**	**No**
SINAMICS HLA with encoder	**No**	**No**	**Yes**[1)]
SINAMICS HLA without encoder	**No**	**No**	**No**

[1)] Not all Basic or Extended Functions are supported. See "SINAMICS HLA"

Notes:

Drive monitoring with or without encoder:

In the SINUMERIK Safety Integrated F_PLC system **encoderless Safety operation only for spindles with induction motors** is supported.

If motors without (Safety-capable) encoders are deployed, not all Safety Integrated Functions can be used.

Definition: "Without encoder"
In the "**SINUMERIK ONE Safety Integrated plus Commissioning Manual**" as well as in this documentation, "Without encoder" always means that either **no encoder** or **no Safety-capable encoder** is used.

During operation without an encoder, the actual velocity values are calculated from the measured actual electrical values. In this way, velocity monitoring is also possible during operation without an encoder.

The configuration of the Safety Integrated Functions and the selection of the monitoring with or without an encoder is performed in the commissioning area of SINUMERIK Operate.

The following table shows the dependency of the drive-independent safety functions on the drive components used (drive and axis type, if required, use of a safety-related encoder).

Function	Basic	Extended	HLA	Electric drive		Brief description
				With encoder	Without encoder	
STO/SS1 Basic **Safe Torque Off / Safe Stop 1**	X	—	X[3]	X	X[1]	Safe Torque Off / Safe Stop according to Stop category 1 of DIN EN 60204-1
STO Extended **Safe Torque Off**	—	X	X[3]	X	X[1]	Safe Torque Off
SBC **Safe Brake Control**	X	X	—	X	X[1]	Safe Brake Control
SS1 Extended **Safe Stop 1**	—	X	X[3]	X	X[1]	Safe Stop according to Stop category 1 of DIN EN 60204-1
SS2/SOS **Safe Stop 2 / Safe Operating Stop**	—	X	X[3]	X	X[1]	Safe Stop according to Stop category 2 of DIN EN 60204-1/ Safe Operating Stop
SLS **Safely-Limited Speed**	—	X	X[3]	X	X[1]	Safely-Limited Speed

Notes:

Function	Basic	Extended	HLA	Electric drive		Brief description
				With encoder	Without encoder	
SSM **Safe Speed Monitor**	—	X	X[3]	X	X[1]	Safe Speed Monitor
SAM **Safe Acceleration Monitor**	—	X	—	X	X[1,2]	Safe Acceleration Monitor
SBR **Safe Brake Ramp**	—	X	—	X	X[1]	Safe Brake Ramp
SDI **Safe Direction**	—	X	X[3]	X	X[1]	Safe Direction
SLP/SP **Safely-Limited Position / Safe Position**	—	X	X[3]	X	X[1]	Safely-Limited Position / Safe Position
SBT **Safe Brake Test**	—	X	—	X	—	Safe Brake Test
SCA **Safe Cam**	—	X		X	—	Safe Cam

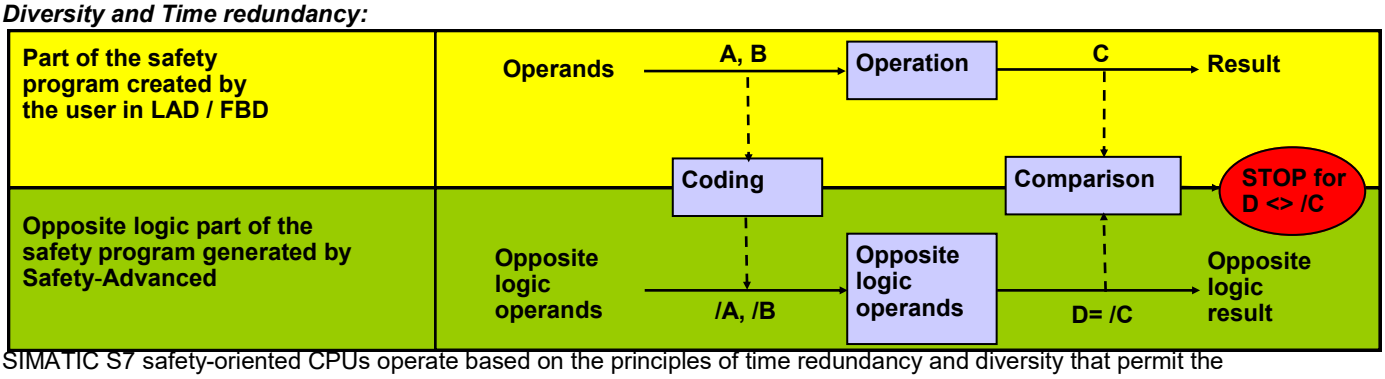
- [1] Encoderless Safety operation is only supported for spindles with induction motors, see "Drive components"
- [2] Only for function setting "Safety without encoder with acceleration monitoring"
- [3] Encoderless Safety operation SINAMICS HLA is not supported for Basic Functions or Extended Functions, see "SINAMICS HLA"

Encoders permitted for asynchronous motors and synchronous motors:

Motors with sin/cos encoders and encoder evaluation with DRIVE-CLiQ interface or via SMC20, SME20/25/120/125 Sensor Modules.
See "Section: Further information
> Overview: SINUMERIK Safety Integrated plus"

Notes:

Diversity and Time redundancy:



SIMATIC S7 safety-oriented CPUs operate based on the principles of time redundancy and diversity that permit the **implementation of F-systems with only one CPU and only one processor**.
The "Safety-Advanced" programming tool generates additional F-blocks (F-FC/F-FB) for the safety program programmed by the user.
These F-blocks are based on the **"diverse redundance" logic** for the user program that uses **"diverse redundance" operands and operations**.

Both parts of the safety program are **processed time redundant, i.e. successively** and the results compared. If an error occurs, the F-CPU responds accordingly and places the plant in the safe state.

Safety Advanced also generates F-system blocks that handle the safety-oriented PROFIsafe communication with the F-I/O, etc.

Approvals:

The classification of the functional safety always applies to the complete safety system. Whereby, the maximum achievable safety requirements of a complete system result from the lowest achievable safety requirement of the individual safety-relevant components.
The SINUMERIK Safety Integrated F_PLC safety system with the F-PLC and SINAMICS Integrated safety-relevant components can be deployed up to the following safety requirements:

- ***SIL2*** - Safety Integrity Level in accordance with IEC 61508
- ***PL d*** - Performance Level in accordance with DIN EN ISO 13849-1
- ***Category 3*** in accordance with DIN EN ISO 13849-1

Furthermore, the safety functions of the SINAMICS Integrated or SINAMICS S120 are normally certified by independent institutes. A list of the appropriate currently certified components is available on request from your local Siemens branch office.

WARNING

The SIMATIC Safety F-system controls processes with the achievable safe state by immediate shutdown.
You may deploy SIMATIC Safety to control only those processes that do not endanger persons or the environment when shutdown immediately.

Further information:
Information about the maximum achievable safety requirements of the individual safety-relevant components can be found in the associated documentation:

- SIMATIC Safety Programming and Operating Manual
- SINAMICS Safety Function Manual

Notes:

Further detailed information:

SINAMICS HLA:

Encoderless safety operation with HLA is supported neither for the Basic Function nor for the Extended Function.
An encoder type approved for SINAMICS HLA is required for safety operation with SINAMICS HLA.

Comparison of electric ↔ hydraulic drives:

The "**SINUMERIK ONE Safety Integrated plus Function Manual**" describes the Safety Integrated Functions from the viewpoint of an electric drive. These descriptions, however, also apply analogously to the "hydraulic" environment. Parameters and messages for the HLA drive object can be found in the **SINAMICS S120/S150 List Manual**.

Encoder types supported by SINAMICS HLA:

- **1-encoder systems:**
  - Safety-conform DRIVE-CLiQ encoder
  - sin/cos encoder connected via SME20/25, SME120/125 or SMC20 (1Vpp, pure analog signal processing)
- **2-encoder systems:**
  - Encoder with DRIVE-CLiQ connection
  - sin/cos encoder connected via SME20/25, SME120/125 or SMC20 (1Vpp, pure analog signal processing)

Basic Functions supported by SINAMICS HLA:

- **Basic Functions:**
  These functions belong to the **standard scope** of the drive and can be deployed without requiring an additional license. They are **always available**.
  They **do not** require an encoder and **do not** place any special requirements on the associated encoder.

Safe Torque Off (STO):
Safe Torque Off is a safety function to prevent unexpected starting in accordance with EN 60204-1. STO prevents energy from being supplied to the valve that can produce a force and corresponds to Stop category 0.

- **Safe Stop 1 (SS1, time controlled):**
  Safe Stop 1 is based on the "Safe Torque Off" function. This allows a stoppage in accordance with EN 60204-1 of Stop category 1.

Extended Functions supported by SINAMICS HLA:

- **Safe Torque Off (STO):**
  Safe Torque Off is a safety function to prevent unexpected starting in accordance with EN 60204-1.
- **Safe Stop 1 (SS1, time and acceleration controlled):**
  Safe Stop 1 is based on the "Safe Torque Off" function. This allows a stoppage in accordance with EN 60204-1 of Stop category 1.
- **Safe Operating Stop (SOS):** Safe Operating Stop protects against unwanted movements. The drive is in closed-loop control and is not disconnected from the energy supply.

Safe Stop 2 (SS2):
Safe Stop 2 safely brakes the valve with subsequent transition to the "Safe Operating Stop" (SOS) state. This allows a stoppage in accordance with EN 60204-1 of Stop category 2.

- **Safely-Limited Speed (SLS):**
  Safely-Limited Speed monitors the drive to ensure that it does not overshoot a preset speed limit value.
- **Safe Speed Monitor (SSM):**
  Safe Speed Monitor serves to safely detect any undershooting of a speed limit value in both movement directions, e.g. for standstill detection. A safe output signal is provided for further processing.
- **Safe Direction (SDI):**
  Safe Direction serves to safely monitor the movement direction.
- **Safely-Limited Position (SLP):**
  Safely-Limited Position ensures that a freely defined traversal area is not exited.

Notes:

- **Transfer safe position values (SP):**
  The "Transfer safe position values (SP)" function transfers a safe position via PROFIsafe to the higher-level control.

For a detailed description of the safety functions, see
"Section: Drive-integrated safety functions of Sinumerik ONE"

Notes:

System Requirements & Information:



SINUMERIK ONE:

SINUMERIK Safety Integrated F_PLC is offered for **NCU/PPU 1740/1750/1760** as of **SINUMERIK ONE NC SW V6.23**. The performance of the PLC must be taken into account depending upon on the application.

- NCU 1750 with PLC 1500F 6FC5317-5AA00-0AA0
- NCU 1760 with PLC 1500F 6FC5317-6AA00-0AA0
- PPU 1740 with PLC 1500F 6FC5317-4AA00-1xA0 (x =C or D)

Typical PLC cycle times 1740/1750/1760 15...20ms/10...15ms/5...7ms

(Please refer to the NC63 catalogue for more information)

SINUMERIK software: (general release)

- NCU firmware as of V6.23
- HMI software SINUMERIK OPERATE as of V6.23

Engineering system software:

SINUMERIK Safety Integrated F_PLC can only be configured with:

- TIA Portal as of V19 Update 1 (SIMATIC STEP 7 Professional)
- SINUMERIK STEP 7 Toolbox as of V19

Option Description	Option Number	Course Requirement
Safety Integrated /SI Axis/Spindle • Safety Integrated Axis / Spindle Safe motion monitoring function 1 axis / spindle. Prerequisite: Safety Integrated F -PLC	**6FC5800-0BK00-0YB0**	**5**
Safety Integrated /SI Logic • Safety Integrated F-PLC Enables the F-PLC integrated in the SINUMERIK ONE for failsafe programming, contains no axis	**6FC5800-0BS60-0YB0**	✓

Notes:

Optional TIA Selection Tool software:
The TIA Selection Tool can be downloaded without charge (link see "Section: Further information > Requirements, software and supplementary conditions") allows devices for Totally Integrated Automation (TIA) to be selected, configured and ordered.

The TIA Selection Tool is available in two variants:
- To download and execute on Windows computers
- As Cloud variant that is started from mobile devices directly in the browser (for which we recommend Safari, Chrome and Firefox)

The TIA Selection Tool provides wizards for selecting individual devices and complete plants. The TIA Selection Tool supports the selection of modules and accessories, checks slot rules and prevents selection errors.
The TIA Selection Tool creates a complete order list from the product selection or product configuration. This order list can be exported directly to the shopping cart of the Industry Mall or in a general standard format.
Controls, IO systems, HMI operator panels, industrial PCs, drive systems, switchgear, software, communication components, power supplies and industrial identification systems can be selected and configured. PROFIBUS and PROFINET networks, including the associated cables and plugs, can also be configured.

Migration:
The migration wizard in the TIA Selection Tool permits with just a few clicks, the configurations of the S7-300 or S7-400 to be migrated to S7-1500 and those of the ET200S or ET200M to be migrated to ET200SP or ET200MP.

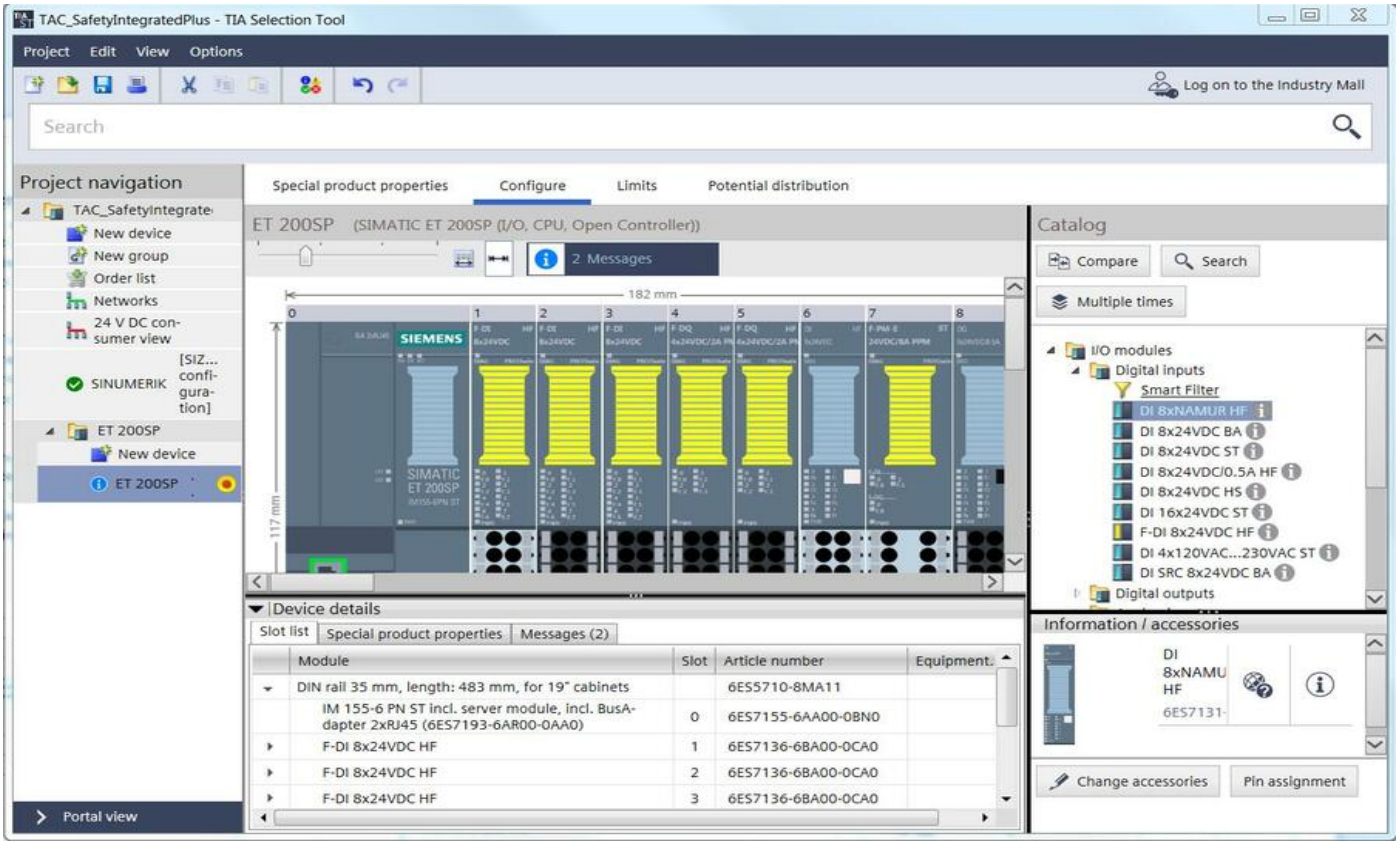
Plant configuration:
The plant configuration allows a SIMATIC controller, as well as decentralized peripheral devices, to be created quickly or easily with SIMATIC ET 200. Whereby, the number and type of the signals must be specified, from which the required modules and devices, including any necessary accessories, are created automatically.

Configuring drives:
With the interface to "SIZER for Siemens Drives" (link see "Section: Further information > Requirements, software and supplementary conditions") you can switch from the TIA Selection Tool to SIZER to configure your drive solution, including the control and motion control. You can then accept and complete the configuration in the TIA Selection Tool.

It is desirable to use the TIA Selection Tool for configuring and ordering, e.g. the ET200SP peripherals with the F-modules.

The TIA Selection Tool is shown at the side in the project view with the opened "TAC SINUMERIK Safety Integrated plus" project:
The "look and feel" of the tool corresponds to that of the TIA Portal.
- The ET200SP configuration is shown in the work area, inspector window and TaskCards.
- The SINUMERIK with Sinamics Integrated is imported automatically by the SIZER configuration.

Notes:



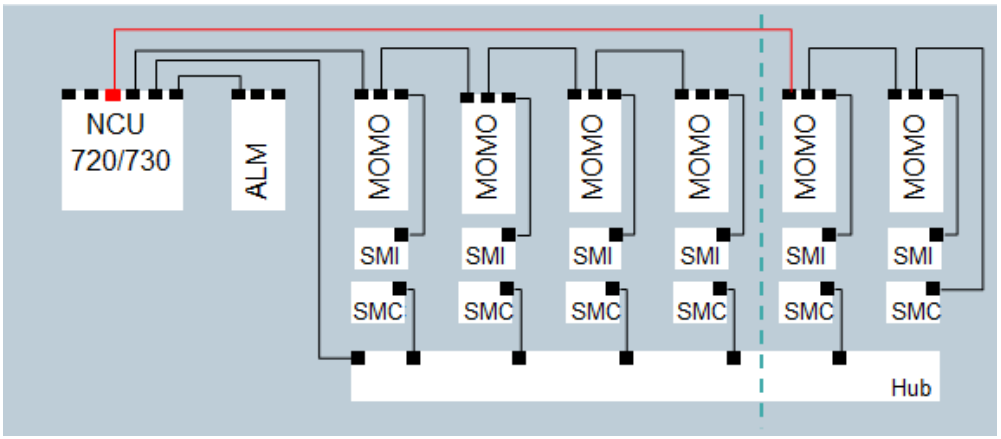
- ***Supplementary conditions:***

- Only **explicitly released encoder systems** may be used.
  A list of the approved Siemens encoders and motors for the Safety Integrated functions can be obtained from your Siemens partner.
- Only **1v Peak to Peak ENDat V2.1** (ENDat 2.2 with additional incremental track - order designation 02) or encoders connected **with incremental DRIVE-CLiQ** are permitted, i.e. SMI or SMC, SME, etc. The use of TTL, HTL, SSI, Resolver, ADI4, etc. is not possible for Safety Integrated functions.
- S120 Booksize and a maximum of two Chassis drives can be used. As of V4.4 software, the S120 Combi drive may also be used.
- ***Performance and number of axes for the PLC 1500:*** If a SINUMERIK ONE with PLC 1500 is deployed for SINUMERIK Safety Integrated plus, it must be clarified in specific situations whether the performance of this PLC suffices to satisfy the requirements of this application. It is planned to prepare recommendations for the migration.
- ***Encoder as "Single Channel Safety Encoder":***
  As of SINUMERIK software release 4.7 SP2, a single-channel encoder (initially Fagor Corp.) can be deployed as second measurement system for Safety Integrated.
  For Safety Integrated plus, a two-channel encoder must also be deployed for the second measurement system.
- ***Stop E*** (brake along the contour after previous separation of tool and workpiece) is implemented as internal fault response with SINUMERIK Safety Integrated plus.
  The external Stop E selection (e.g. for emergency stop) is not currently available (implementation date open).
- ***Dynamic stop D***:
  The acceleration can be changed by program with the "ACC" NC program command. This could cause the drives not being brought to a standstill in time for a Stop D; this would cause a pulse suppression (coasting of the drives).
- ***Safe cams***:
  As of SINUMERIK software release 6.13 (SINAMICS software release 4.9), 32 safe cams are available for SINUMERIK Safety Integrated plus.
  Note: The "Safe cams" and "Safe position" functions mutually exclude each other.

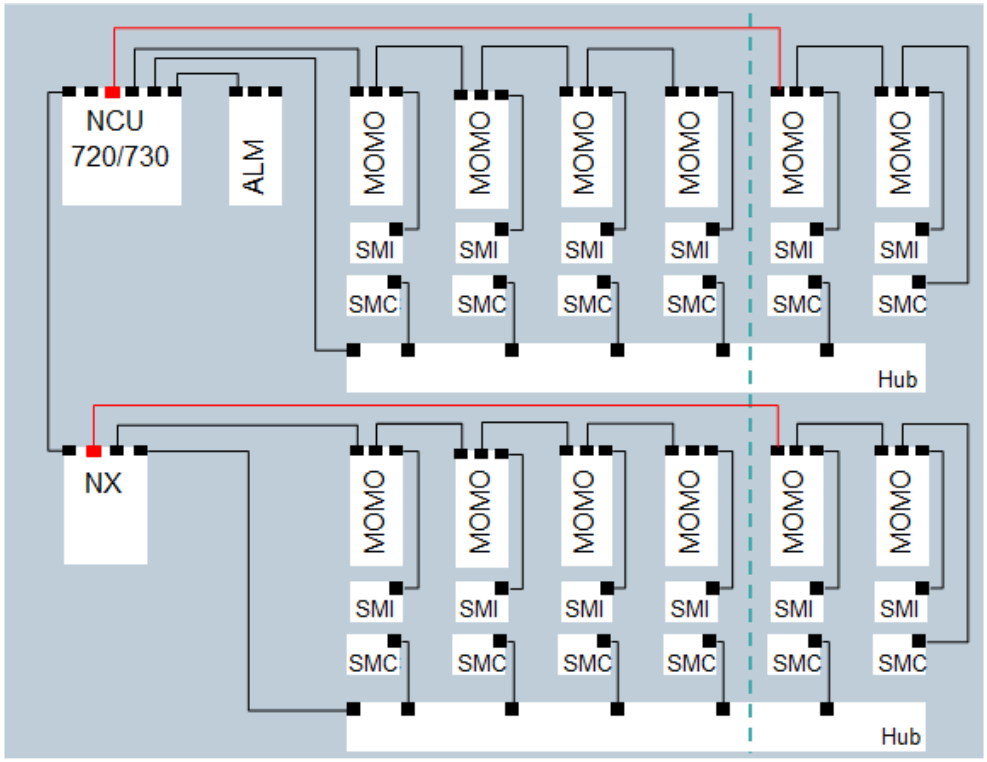
Notes:

- ***Maximum of four drives on a DRIVE-CLiQ line:***
  Because of the increased volume of communication data for SINUMERIK Safety Integrated plus, the maximum number of drives on a DRIVE-CLiQ line is limited to four. If more drives are operated they must be distributed over different DRIVE-CLiQ lines.

Topology example constellation without NX:

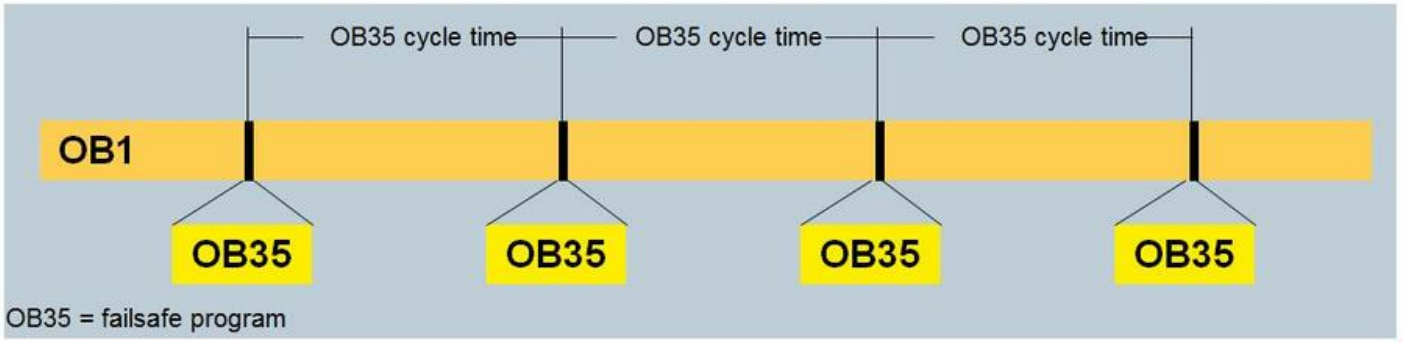


Topology example constellation with NX:



Notes:

Performance and number of axes depends on the desired response time:



The default OB35 interrupt time is 100ms. This can be optimised to a lower value and will be dependant on the number of Safety Integrated axes/inputs/outputs/functions in use.

Notes:

Drive-integrated safety functions of Sinumerik ONE:

Integrated safety functions are available with **SINUMERIK Safety Integrated F_PLC** with which a highly effective man and machine protection can be implemented. The safety functions satisfy the requirements of **Category 3** as well a **Performance Level d**
according to DIN EN ISO 13849-1 and **Safety Integrity Level SIL 2** according to DIN EN 61508.
In this way, the essential requirements for functional safety can be implemented simply and economically.

SINUMERIK Safety Integrated withsystem-integrated safety functions and Safe Programmable Logic (SPL)	**SINUMERIK Safety Integrated F_PLC with Drive-based Safety (Basic Functions)**	**SINUMERIK Safety Integrated F_PLC with Drive-based Safety (Extended Functions)**
Stop A external: SH: Safe standstill	**STO:** Safe Torque Off	**STO:** Safe Torque Off
Stop B:	**SS1**: Safe Stop 1 (Safe Stop according to Stop Category 1)	**SS1:** Safe Stop 1 (Safe Stop according to Stop Category 1)
Stop C ext.:		**SS2:** Safe Stop 2 (Safe Stop according to Stop Category 2)
Stop D ext.:		**SS2e:** Safe Stop 2e I)
Stop E ext.:		**Stop E:** II)
Stop F:		System response to errors
SBH: Safe operating stop		**SOS:** Safe Operating Stop
SBC: Safe brake control	**SBC:** Safe Brake Control	**SBC:** Safe Brake Control
SBT: Safe brake test		**SBT:** Safe Brake Test
SBM: Safe brake management		**SBM:** Safe Brake Management comprises SBC and SBT
SBR: Safe braking ramp		**SAM:** Safe Acceleration Monitor
		SBR: Safe Brake Ramp
n < nx: Safe speed monitoring		**SSM:** Safe Speed Monitor
SG (1...4): Safely reduced speed		**SLS (1...4):** Safely-Limited Speed
SG-Override Bit 0 - 4: SG-Override for SG2 + SG4		**SLS-Override via PROFIsafe:** 0 – 100% (1...32767) for SLS 1
SN: Safe software cam, Safe cam track		**SCA:** Safe Cam
SE: Safe software limit switch		**SLP:** Safely-Limited Position
		SP: Safe Position
		SDI: Safe Direction
		SLSsensorless: Safely-Limited Speed
SGA: Safety-related output		**F--DO:** Failsafe Digital Output
SGE: Safety-related input		**F--DI:** Failsafe Digital Input
Safe software relays		
SPL: Safe Programmable Logic (NCK & PLC)		Safe program in the F-PLC

The following table shows a ***Comparison of the function designations for SINUMERIK***:

I) For the drive-based safety functions of the SINAMICS, this is a delayed SOS.

II) Stop E (braking on the contour after previous separation of tool and workpiece) is currently not available for SINUMERIK Safety Integrated F_PLC "with selection from external" (if requried, it can be resolved in the application). For the drive-based safety functions of the SINAMICS, this is a delayed SOS with previous triggering of ESR (Extended

System-based stop variants for SINUMERIK Safety Integrated (SPL) and SINUMERIK Safety Integrated plus:

Safe Stops bring a drive in motion down to standstill. The type of stop response in the event of an error can be permanently specified by the system or configured by the machine manufacturer. The axes can remain ready for operation and are safely monitored at standstill. The axes can be braked on the current acceleration profile or for an axis-specific current limit. Stops A, C, D and E can be directly controlled via the safely programmable logic.
STO, SS1, SS2 and SOS refer to naming conventions from IEC61800-5-2 and are used for Drive-based Safety.

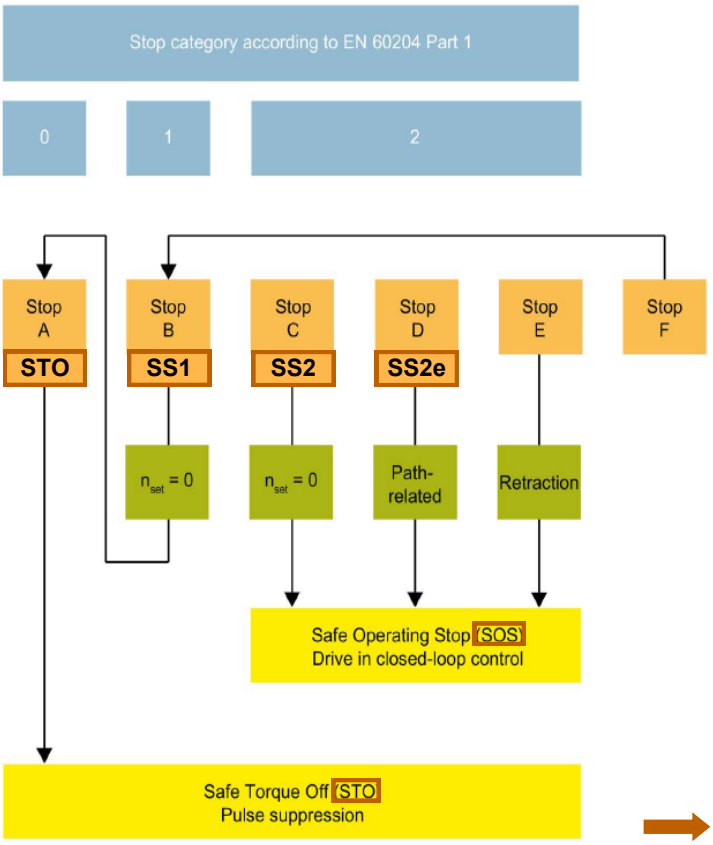
SINUMERIK Safety Integrated SPL (EN60204 Part 1)

- **Stop A** → **SH:** Safe standstill
- **Stop B** → **Stop A** → **SH:** Safe standstill
- **Stop C** → **SBH:** Safe operating
- **Stop D** → **Path-related** → **SBH:** Safe operating
- **Stop E** → **Retract** → **SBH:** Safe operating
- **Stop F** → **Stop B** → **Stop A** → **SH:** Safe standstill

Drive-based Safety (IEC61800-5-2)

(Select **STO** → **STO:** Safe torque off)
(Select **SS1** → **STO:** Safe torque off)
(Select **SS2** → **SOS:** Safe operating stop)
(Select **SS2e** → **SOS:** Safe operating stop)
Delayed SOS with previous ESR
System response to errors

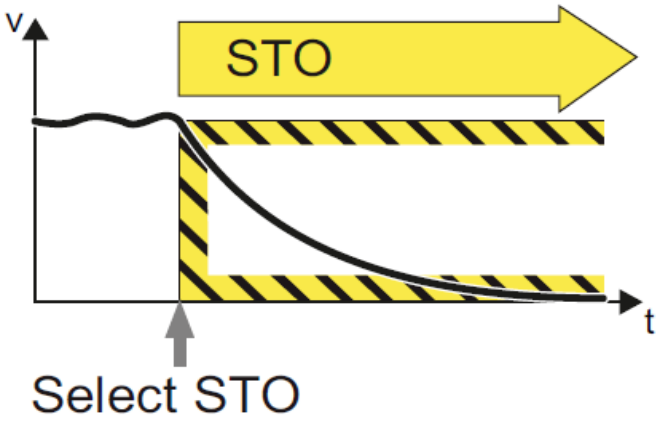
- **Stop A:** With Stop A (without galvanic separation), the drive is switched directly torque-free via the Safe standstill function. A drive that is still in motion coasts to a standstill. A drive at standstill can no longer be unintentionally started.
- **Stop B:** The drive is braked at the current limit in speed control and is brought to a safe standstill.
- **Stop C:** The drive is braked at the current limit in speed control and is brought to a safe standstill.
- **Stop D:** The drives are braked as a group path-related (interpolatory) on the contour and are brought to a safe standstill.
- **Stop E:** The drives are braked as a group including a retraction movement, in which the tool and the workpiece are separated from one another, path-related and are brought to a safe standstill.
- **Stop F:** Stop F is permanently asssigned to the cross-check result and data comparison (DCC) and connot be influenced by the user. A Stop F is triggered when a discrepancy is determined in the montioring channels of Safety Integrated. A Stop A or Stop B is then triggered depending on the parameterization.



Notes:

SINUMERIK Safety Integrated F_PLC functions:

Safe Torque Off (STO) Basic/Extended) (Parameter)
Safe Torque Off (STO) is used in the event of an error or in conjunction with a machine function for the safe electronic separation of the power supply to the motor, which can generate a torque. This is performed axis-specifically and without contacts. There is no galvanic isolation of the motor from the drive module required.



Protection against unexpected start up:
- The pulses of the drive are suppressed
- The power supply is separated electronically
- The drive is safely torque-free
- A restart is prevented by the two-channel pulse suppression. The switch-on inhibit prevents an autonomous restart after deselection of STO.
- The basis for this function is the two-channel pulse suppression integrated in the Motor Modules / Power Modules.

Application cases of STO:
- Always active after emergency stop
- Manual turning of a spindle when the protective door is open
- A protective door must only be opened when the torque of a motor is off.

Functional characteristics of STO:
- Selection of STO by means of a failsafe input or PROFIsafe safe communication.
- The function must be enabled via parameters.
- With the selection/deselection of the STO function, the **Safety errors are acknowledged** when the Basic Functions are used. The standard acknowledgment mechanisim must also be performed.

- ***Extended acknowledgment:***
  Safety messages of the Extended Safety Functions can also be acknowledged through the selection/deslection of STO. You must configure the extended message acknowledgment for this purpose **(p9507.0 = 1)**.
  If in addition to the "Extended Functions", the "Basic Functions via terminals" are also enabled, acknowledgment through the selection/deslection of STO via terminals is also possible in addition to the acknowledgment through the selection/deslection of STO via PROFIsafe.

WARNING

Unplanned movements of the motor:
Unwanted movements of the motor can occur after separation of the power supply (STO active) (e.g. coast down of the motor), and therefore endanger persons.
- Prevent unwanted movements by using a brake with safe montoring (SBC), for example.

WARNING

Unplanned movements of the motor:
The simultaneous breakdown of the depletion layer of two power transistors can cause an unwanted movement of the motor [1)]. This can result in accidents causing severe injuries or death.
- Prevent unwanted movements by using a brake with safe montoring (SBC), for example.

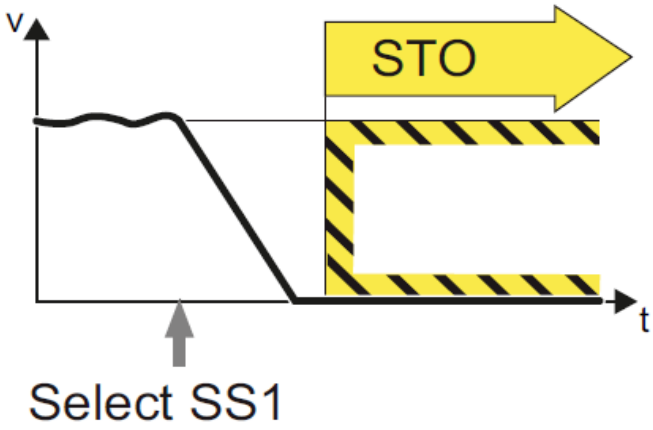
[1)] The maximum movement can be:
- For synchronous rotary motors:
  Maximum movement = 180° / pole pair number
- For synchronous linear motors:
  Maximum movement = pole width

Further details on the STO function can be found in the **"SINUMERIK ONE Safety Integrated plus Function Manual"**.

Notes:

Safe Stop 1 (SS1): (Parameter)
A shutdown of the drive according to EN 60204-1: 2006 of Stop category 1 can be implemented with the Safe Stop 1 (SS1) function. The SS1 function brakes the motor along a settable ramp (OFF3 ramp p1135), monitors the motor deceleration within specified limits and triggers the **Safe Torque Off (STO)** function

- After a **delay time** (p9652 Basic /p9556 Extended)
- Or when **a speed is below the threshold** (**Extended Functions only,** see Monitoring modes)



Application cases of SS1:

- After an emergency stop button is pressed, a drive must be brought into the STO state as quickly as possible.
- A central emergency stop button ensures that several drives are braked as quickly as possible and brought into the STO state.

Procedure at SS1 selection (Basic Functions):

- If the **motor** has already been **switched off** when SS1 is selected, then there is no response until the SS1 delay time has expired.
  STO takes effect after the time has expired.
- If the **motor** has been **switched on** when SS1 is selected, the converter brakes the motor with the OFF3 ramp-down time. When the delay time has expired, STO is triggered automatically.

Procedure at SS1 selection (Extended Functions):

- If the **motor** has already been **switched off** when SS1 is selected, then the converter directly switches the motor torque off safely (STO).

- If the **motor** has been **switched on** when SS1 is selected, the converter brakes the motor with the OFF3 ramp-down time. **At the latest** (see Monitoring modes) after the delay time has expired, STO is triggered automatically.

Functional characteristics of Safe Stop 1:
SS1 is enabled by p9652 (delay time) ≠ 0:

- **p9652/p9556 = 0: SS1 is not enabled**. Only STO can be selected via the onboard terminals and PROFIsafe.
- **p9652/p9556 > 0: SS1 is enabled**. Only SS1 can be selected via the onboard terminals; SS1 and STO can be selected with PROFIsafe.

Setting the delay time:
In order that the drive can fully traverse the OFF3 ramp and an existing motor holding brake can close before the pulses are safely suppressed, a delay time must be set as follows:

- Motor holding brake parameterized:
  Delay time p9652/p9556 ≥ p1135 + p1228 + p1217
- Motor holding brake not parameterized:
  Delay time p9652/p9556 ≥ p1135 + p1228
- The setting of parameter p1135 must be oriented towards the actual braking capability of the drive.

Monitoring modes:
There are two different monitoring modes of the SS1 function for the **Extended Functions with or without encoder**:

- Braking ramp monitoring (Safe Brake Ramp, SBR):
  STO is activated as soon as the speed is below the threshold.
- Acceleration monitoring (Safe Acceleration Monitor, SAM):
  STO is activated as soon as the SS1 shutdown speed is below the threshold.

SBR and SAM are explained in detail in the following.

Notes:

Safe Stop 1 with external stop (SS1E): (Parameter)
For drive line-ups (e.g. drives that are connected mechanically), the drive-independent braking on the respective OFF3 ramp can be distruptive. When the SS1E is selected, the respective safe delay time (p9562) is started, but **no braking on the OFF3 ramp** is triggered.
The setpoint control remains at the higher-level control.
The control receives information that SS1E has been selected via the Safety Information Channel.

When "SS1 with external stop" is used, **neither of the two monitoring functions (SBR, SAM) is active**. With SS1E, the drive must be shut down within the delay time by the higher-level control. STO/SBC is triggered automatically after the delay time has expired.

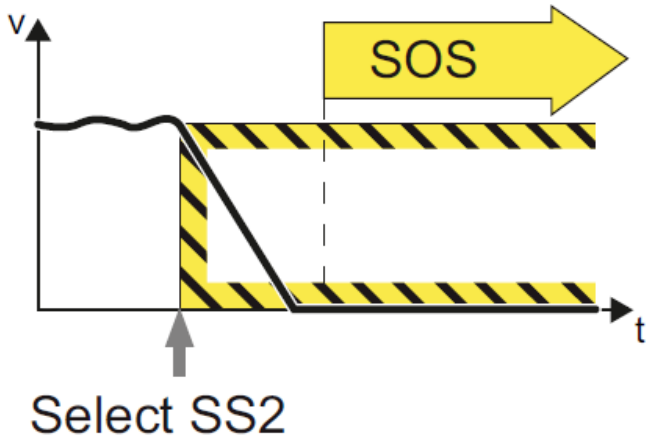
Activation of Safe Stop 1 with external stop:
- With the **Basic Functions**, **p9653 = 1** must also be set
- With the **Extended Functions**, **p9507.3 = 1** must also be set

WARNING

Non-monitored axis movements:
Non-monitored random axis movements that can cause death or serious injury can occur with "Safe Stop 1 (time controlled) with external stop" during the non-monitored delay time.
- If there is a danger of unwanted movements in your application, take the appropriate measures, e.g. by using a brake with safe monitoring.
  For further information, see SBC

Notes:

Safe Stop 2 (SS2): (Parameter)
A shutdown of the drives according to EN 60204-1: 2006 of Stop category 2 can be implemented with the Safe Stop 2 function.
After selecting Safe Stop 2, the drive brakes along the OFF3 ramp with monitoring of the load speed and after expiry of the delay time or undershoot of a configured speed limit, goes into the **Safe Operating Stop (SOS)** state. The Safe Stop 2 (SS2) function can only be used with an encoder.
.



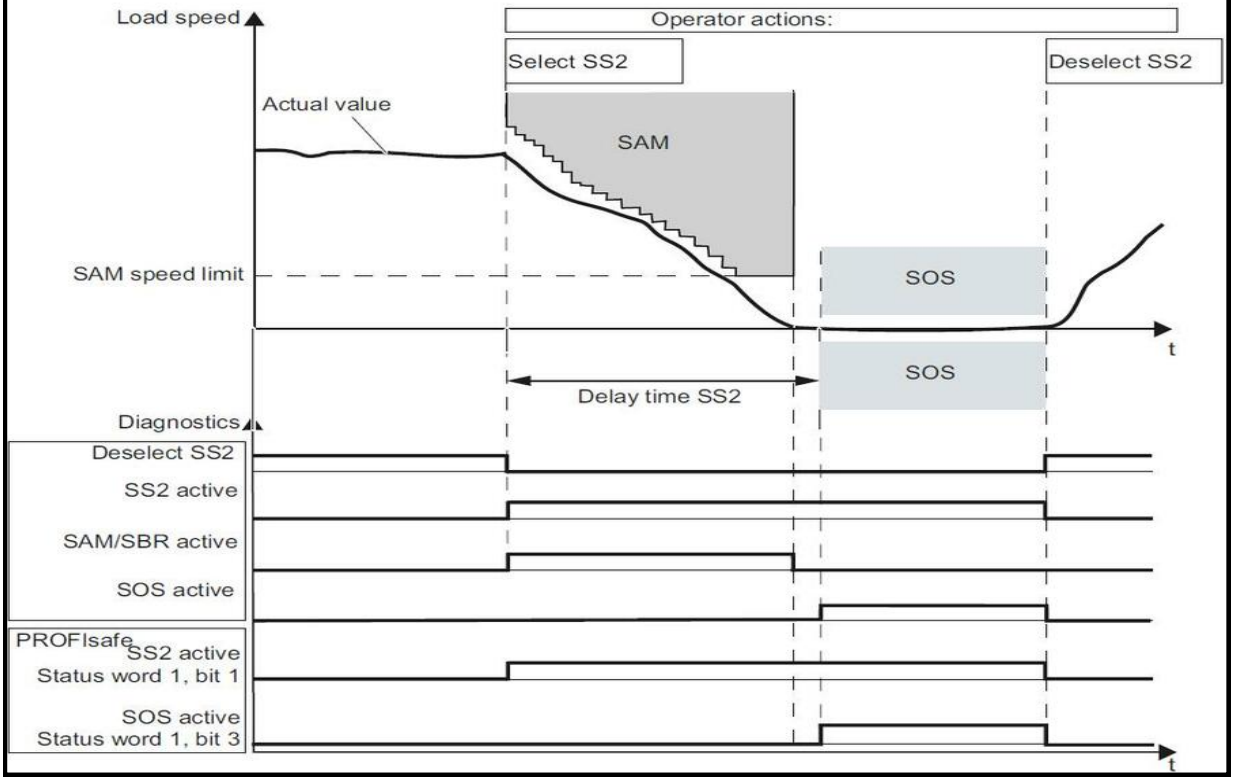
Application cases of SS2:
- A protective door must only be opened when the motor is at safe standstill.

Procedure at SS2 selection :
- If the motor is already at standstill when SS2 is selected, the converter activates the **Safe Operating Stop (SOS)** function after a delay time.
- If the motor is not at standstill when SS2 is selected, it is **braked along the OFF3 ramp**. The braking operation is monitored with one of the following functions depending on the **setting in p9506**:

1. **Safe Acceleration Monitor (SAM)**: A faulty acceleration is detected.
2. **Safe Brake Ramp (SBR):** A violation of the braking ramp is detected.

After a delay time, the converter activates the **Safe Operating Stop (SOS)** function.
The standstill of the drive is monitored safely with this function.

Braking response and diagnostics for SS2 with SAM:



Notes:

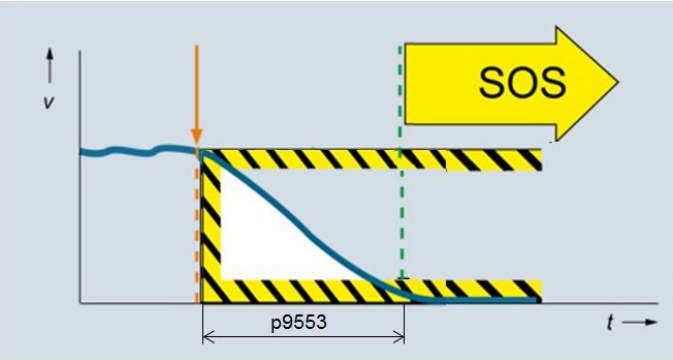
Functional characteristics of Safe Stop 2:

- Shutdown of the drive according to Stop category 2 (EN 60204-1).
- Braking on a settable ramp (OFF3, p1135).
- With SS2, the braking operation on the OFF3 ramp can be monitored by means of **SAM and SBR**. A faulty acceleration is then detected and the drive shut down with STO.
- Transition to SOS after a settable time (p9552) or undershoot of a configurable speed.
- The delay time (p9552) for safe braking of the motor on the OFF3 braking ramp (p1135) must be dimensioned so that the drive can brake to standstill within this time from any speed of the working process.
- The standstill tolerance (p9530) must not be viloated after the transition to the SOS state.
- After the braking operation, the drives remain in speed control with the speed setpoint n = 0. The full torque is available.
- The setpoint specification (e.g. from the setpoint channel or from a higher-level control) remains disabled during the selection of SS2.
- One of the following functions is active during braking:

1. With **p9506 = 0**, the following applies:
   The braking is monitored with the **"Safe Acceleration Monitor"** function (see SAM).
2. With **p9506 = 2**, the following applies:
   The braking is monitored with the **"Safe Brake Ramp"** function (see SBR).

Notes:

Safe Stop 2e with external stop (SS2e): (Parameter)
A safe **shutdown of the drives on the contour** by means of an external stop can be implemented with the **Safe Stop 2 external** function.



Application cases of SS2e:

- Machine protection

Note:
The SS2e function is called Dynamic Stop D in the previous supplementary conditions.

WARNING

Unexpected axis movement:
When the "Safe Stop 2 with external stop" (SS2E) function is active, the speed follows the setpoint of the higher-level control during the deceleration time (p9553). Unexpected axis movements that can result in serious injury or death are therefore possible.

- During the deceleration time (p9553), keep persons out of the hazardous area of the machine or plant, e.g. by closing protective devices.

Safe shutdown of the drives on the contour:

- Braking of the drives on the contour.
- **During the deceleration time p9553, there is no monitoring of the brake ramp (SBR) or the acceleration (SAM) and no standstill detection.**
- **No persons must be in the hazardous area during the deceleration time.**
- With SS2E, the **drives** involved on the countour must be shutdown **within the deceleration time** by the setpoint of the higher-level control**.**
- When SS2 with external stop is selected, the drive does not brake the motor itself, but follows the specified setpoint of the higher-level control.
- Transition to **SOS** after the deceleration time expires. When the SS2E function is active, the higher-level control must specify the speed setpoint in such a way that the motor is at standstill at the latest after the deceleration time.
- To enable "Safe Stop 2 with external stop", p9501.18 = 1 must be set.
- The PROFIsafe control word S_STW2.28 selects the SS2E function. PROFIsafe S_STW2.28 is contained in telegrams 31, 901, 902 and 903.
- The PROFIsafe status word S_ZSW2.28 shows whether the SS2E function is active. The PROFIsafe status word S_ZSW2.28 is contained in telegrams 31, 901,902 and 903. The associated diagnostics parameter is r9722.28. In the "Safety Information Channel", the staus word S_ZSW3B.11 shows whether the SS2E function is active. The associated diagnostics parameter is r10234.11. The diagnostics parameters p9722.28 and p10234.11 are also set for the internal STOP D.

Interruption of the active SS2E function through the SS1 and SS2 functions:

- When SS1 is selected, the drive brakes the motor along the OFF3 ramp and monitors the speed with the SAM function. The STO function is active when the motor is at standstill.
- When SS2 is selected, the drive also brakes the motor along the OFF3 ramp and monitors the speed with the SAM function. The SOS function is active after the p9552 time.

Notes:

Stop E:

Braking along the contour with previous retraction movement

Activation of the drive-independent ESR functionality:

- Braking delayed
- Axes retracted (not path-related) by a parameterized amount
- Triggered for: communications failure, SLS, SDI, SLP

Retraction during the delay time by the control

- Braking delayed
- Axes retracted path-related by the control
- Triggered for communications failure, SLS, SDI, SLP

ESR functionality for an external stop for example, emergency stop is not currently supported.

Detailed information about ESR:

ESR behavior for communications failure:

The response of the SINAMICS drive in the event of a communications failure for the concurrently enabled "Extended stop and retract (ESR)" function module is described below.

Prerequisite:

- The Safety Integrated Extended Functions are controlled via PROFIsafe
- The "Extended stop and retract (ESR)" function module is activated and enabled

Communications failure:

In this context, a communications failure is considered to be one of the following possibilities:

- Interruption/fault of the PROFIBUS or PROFINET connection for activation via PROFIsafe
- STOP response of the F-PLC

Drive response:

In the event of a communications failure, SINAMICS differentiates between the following responses:

- If p9580 ≠ 0 and ESR is enabled, the parameterized ESR response is performed in the event of a communications failure.
- If p9580 ≠ 0 and SLS is active, the parameterized ESR response is performed in the event of a communications failure only when as SLS response, a STOP with delayed pulse suppression is parameterized for bus failure (p9563 [0...3] ≥ 10).
- If p9580 ≠ 0 and SDI is active, the parameterized ESR response is performed in the event of a communications failure only when as SDI response, a STOP with delayed pulse suppression is parameterized for bus failure (p9566 [0...3] ≥ 10).
- If p9580 ≠ 0 and SLP is active, the parameterized ESR response is performed in the event of a communications failure only when as SLP response, a STOP with delayed pulse suppression is parameterized for bus failure (p9562 [0...1] ≥ 10).

Maximum response time for communications interruption:

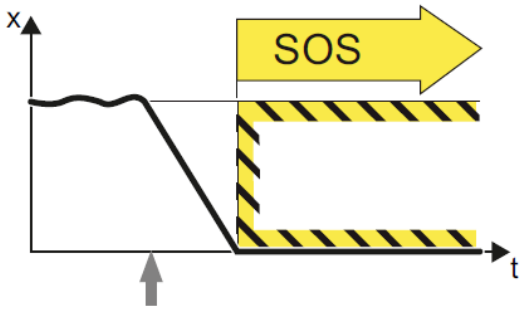
Safety Integrated for the interruption of a safety-relevant communication permits only a response time (p9580) of maximum 800 ms. After expiration of this time, the drive switches to the "failsafe status". In this case, the S_STW1/2 is assumed to be 0. This means that all safety functions are selected. Whereby, STO has the highest priority; i.e. the pulses are cleared safely.

Safety Integrated plus supports STOP E only as parameterized fault response of the SLS and SLP safety functions:

All safety functions can be parameterized in dedicated SINUMERIK Operate screen forms. The fault response, the delay time, etc. can be set in these screen forms.

Notes:

Safe Operating Stop (SOS): (Parameter)
According to EN 61800-5-2, the Safe Operating Stop (SOS) function is used for the safe monitoring of the standstill position of an axis or spindle. The drives are fully functional in positon control or speed control.
The standstill of the drive is monitored via an SOS tolerance window. At the time this function takes effect, the current actual position is saved as comparison position until SOS is deselected.
After the deselection of SOS, there is no delay time, the drive can be traversed immediately.
The drive is shut down with SS1 when the safe standstill tolerance window is violated.



SOS selection

WARNING

A drive can be pushed out of the SOS position through mechanical forces:
A drive in position control can be pushed out of the Safe Operating Stop (SOS) position by forces that are greater than the maximum torque of the drive. This unwanted movement of the drive triggers a Stop function category 1 according to EN 60204-1 (fault response function STOP B). The warnings for SS1 and STO must be taken into account here.
- If there is a danger of unwanted movement in your application, take the appropriate measures, e.g. by using a brake with safe monitoring.
  See "Safe Brake Control (SBC)"

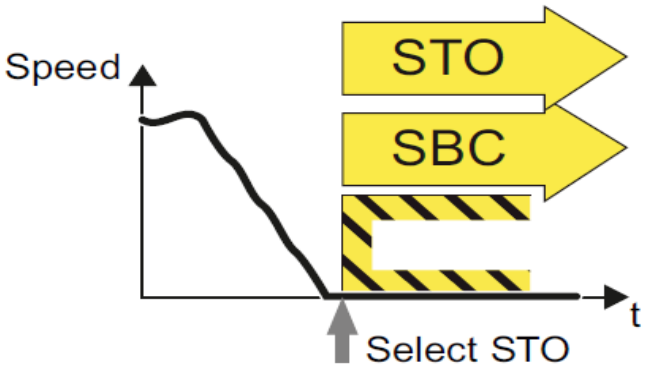
Safe position monitoring:
- The drive remains in position control
- The standstill of the drive is monitored via an SOS tolerance window (p9530)
- The drive has the full torque

Application cases of SOS:
- A protective door must only be opened when the motor is in safe standstill
- Working in a hazardous area with opened protective doors
- Axes with assymmetric tools are held in position
- Axes subject to gravity are held in position

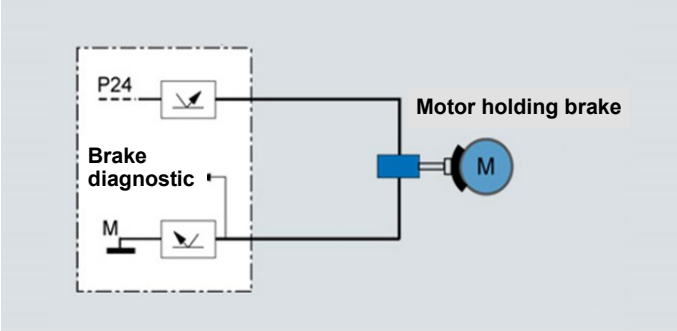
Notes:
- ***In contrast to SS1 and SS2, SOS does not brake the drive independently:***
  The setpoint control remains with the higher-level control. Therefore, the control must react to the "SOS selected" bit in the user program in such a way that the drive is brought into standstill within the deceleration time.
- Particularly in position control of the SOS function, the motor is under current:
  Prevent the motor being moved in the SOS state.
- Effect on the "Speed setpoint limitation active" (r9733[0...2]):
  When SOS is selected, the setpoint 0 is specified in r9733[0...2].
- Size of the tolerance window:
  The tolerance window should be slightly larger than the standard standstill monitoring limit, otherwise, the standard monitoring functions have no effect.
  Parameter r9731 shows the safe position accuracy (load-side) that can be achieved as maximum for the safe motion monitoring functions based on the sensing of the actual value.

Notes:

Safe Brake Control (SBC): (Parameter)
The Safe Brake Control (SBC) function controls holding brakes that work on the quiescent current principle (e.g. motor holding brake), by means of a safe output signal.



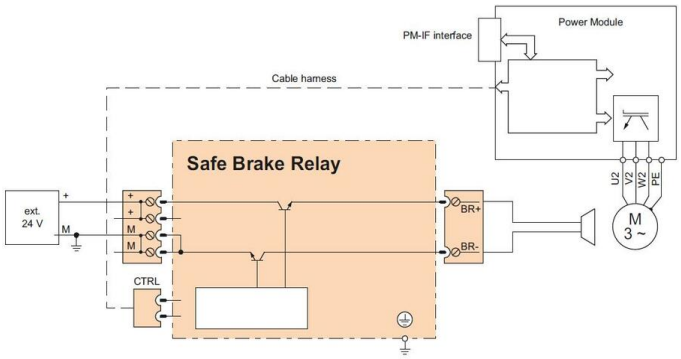
Safe control of a brake:
- SBC is executed when "Safe Torque Off" is selected.
- The SBC function can only be used together with STO. SBC cannot be selected separately.
- Short-circuit of the brake winding and wire break are detected.
- The "Safe Brake Control" function is enabled via parameter p9602.
- SBC is executed irrespective of the operating mode of the brake control set in p1215. However, SBC does not make sense for p1215 = 0 or 3.
- The brake control is performed via a two-channel output of the SINAMICS Motor Module.



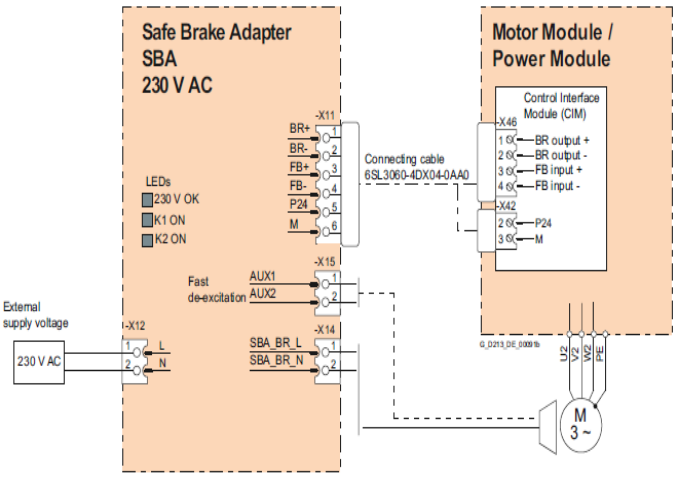
Application cases of SBC:
- Control of a brake for axes subject to gravity.
- The safe control of a motor holding brake must be ensured in order to guarantee the standstill of the motor.

Procedure at SBC selection:
The converter detects the selection of STO via a failsafe input or via the PROFIsafe safe communication. The converter then safely switches off the torque of the connected motor.
SBC is triggered together with STO (if configured). The Motor Module / Safe Brake Relay / Safe Brake Adapter then perform the action and safely control the outputs for the brake.

Additionally required hardware for other formats:
- A **Safe Brake Relay** is required for the Blocksize format for the "Safe Brake Control".



- With the Chassis format (as of an article number with the ending ...3), the connection terminals are only dimensioned for 24 VDC with 150 mA. The Safe Brake Adapter is therefore required for higher currents and voltages. The **Safe Brake Adapter** is available for 230 VAC brake control voltage.



Notes:

Notes:

- ***No SBC for SINAMICS HLA:*** Safe Brake Control is not supported by SINAMICS HLA.
- ***Control of the brake via a relay for "Safe Brake Control":*** When the "Safe Brake Control (SBC)" function is used, the use of relays/contactors for switching brakes can cause errors in the brake control. For this reason, this type of control is not generally enabled.



WARNING

Unwanted movements of the motor due to a defective brake:

The "Safe Brake Control" function does not detect any mechanical defects of the brake.
A cable break or short-circuit in the brake winding is only detected at a state change, i.e. when opening or closing the brake.
With the SINAMICS S120M, a cable break is only detected when opening the brake.
For devices in Chassis format with connected Safe Brake Adapter, the connection cable between the Safe Brake Adapter and the motor brake is not monitored for cable break or short-circuit.
The defects described above can trigger unwanted movements of the motor which can result in serious injury or death:

- Make sure the brake does not have an external power supply. Information on this can be found in EN 61800-5-2:2007, Appendix D.
- During the commissioning, test the brake with the Safety Integrated Extended Function "Safe Brake Test (SBT)". See "Safe Brake Test (SBT)"

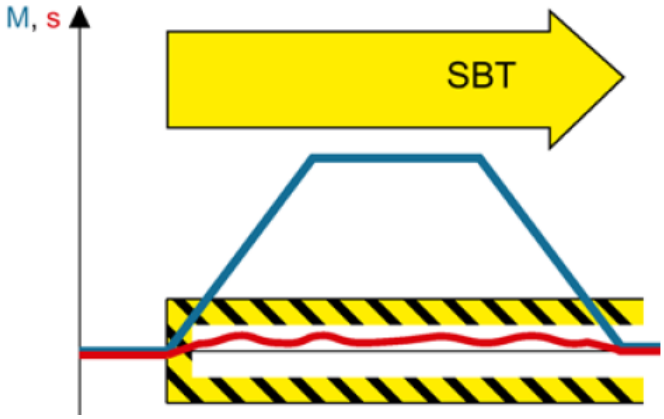
Safe brake management (SBM):

For the SINUMERIK Safety SPL, Safe brake management is the **combination** of the previously described functions:

- **Safe Brake Control**
  for safe two-channel brake control
- **Safe Brake Test**
  for a two-channel cyclic brake test

Notes:

Safe Brake Test (SBT): (Parameter)
The Safe Brake Test (SBT) **diagnostic function**, which satisfies the requirements for Category 2 in accordance with EN ISO 13849-1, tests the required holding torque of a brake (service brake or holding brake). Linear and rotary brakes can be tested. To do this, the drive establishes a configured torque that opposes the closed brake and tests whether the axis movement exceeds the parameterized tolerance window. In parallel, monitoring is performed whether the speed lies below 10% of the configured maximum speed of the drive. When the brake operates correctly, the axis movement and the speed remain within a parameterized tolerance.
If a larger axis movement is determined by the actual encoder values, the brake cannot provide the required holding torque. The brake must now be serviced or replaced.
Two test sequences can be selected for each brake. Within these test sequences, for example, different holding torques can be parameterized.
When selected, the function determines the currently applied load torque and uses this to specify the test torque.



Functional characteristics of SBT:
- The SBT function parameters are protected with the Safety password and can be changed only in Safety commissioning mode.
- This function can be used to test brakes not only operated directly on the SINAMICS S120 (integrated brake control), but also externally activated brakes (e.g. via a PLC).

- The SBT function parameters are protected with the Safety password and can be changed only in Safety commissioning mode.
- A maximum of two brakes can be tested:
  - One motor holding brake activated by the integrated brake control of the SINAMICS and, in addition, one externally activated brake.
  - Two externally activated brakes.
  - One motor holding brake activated by the integrated brake control of the SINAMICS.
  - One externally activated brake.
- The function satisfies SIL 1 in accordance with IEC 61508 or PL d / Cat. 2 in accordance with EN ISO 13894-1.

Integration with NCK and PLC programs:
The SBT drive-independent function does not require any setpoint specification for PLC or NC, but rather only enables from the NC and activation via the PLC user program. The relevant signals for performing the Safe Brake Test (SBT) are transferred via the Safety Info Channel (SIC) and the Safety Control Channel (SCC) between the PLC and the drive. See "Section: Drive-integrated safety functions activation > Safety Info Channel, Safety Control Channel"

- The SIC/SCC communication is transferred in telegram 701. Telegram 701 is added automatically in the TIA Portal configuration when the SINUMERIK Safety Integrated F_PLC (F-PLC) mode is activated.
- In the axis DB (Axis 1 [DB31] ... Axis 31 [DB61]), the SIC/SCC signals (for activation of SBT) are represented. SBT is activated with bits (DBX11.0 and DBX71.0) in the associated axis DB. This makes it possible to perform the SBT via the PLC user program.
- The PLC block "SI_BT_AWL_V3.0" is used to activate the Safe Brake Test of an axis. This block is added to a sample application that is explained in more detail during the commissioning and programming of the SBT.

Information for switching: Block required for SBT (Safe Brake Test):
The SBT is activated via the associated axis DB. The SI_BrakeTest [FB11] block familiar from the Safety Integrated (SPL) operating mode is no longer used in the SINUMERIK Safety Integrated F_PLC (F-PLC) operating mode.

Notes:

Prerequisites for enabling the Safe Brake Test by the NCK:

- The axis must be at standstill and have a safe axis position.
- The controller enable and the pulse enable from the NC must be present.

Prerequisites for enabling the Safe Brake Test by the DRV (drive):

- Enable the Safety Integrated Extended Functions; also available for Safety Integrated Extended Functions without selection.
- Safe Brake Control must have been enabled for testing a brake (motor holding brake) controlled by SINAMICS.
- Safety Integrated Extended Functions with encoder have been enabled (the "Safe Brake Test" (SBT) safety function can be deployed only with encoder). For information about the possible encoder concepts, see "Safe actual value acquisition".
- Speed closed-loop control with encoder (p1300 = 21): SBT is not possible for encoderless speed closed-loop control (e.g. vector-U/f closed-loop control) and torque closed-loop control. In this case, alarm A01784 is issued.
- The pulses must have been enabled when the SBT is selected. The actual speed value must not exceed 1% of the maximum speed (p1082) when the SBT is selected and during its complete execution.
- The brake(s) must be open.
- SBT and HLA: The "Safe Brake Test" (SBT) function is not available for SINAMICS HLA.

General execution of the Safe Brake Test:

- After selecting the brake test by the user (0/1 edge in r10231.0), the static hanging load is determined. Consequently, all brakes must be open and the pulses enabled when the brake test is selected.
  - For the motor holding brake test activated directly by SINAMICS, the opening is performed automatically for pulse enable and p1215=1.
  - For the test of the external brake via p10234.6 or the SIC/SCC via S_ZSW3B.6, the value 0 indicates that the external brake must be opened. This must occur within 11 seconds, otherwise the test is canceled and a fault issued.
- The brake, the test sequence and the test direction are then selected by the user.
- Only when with the start of the brake test / brake test sequence by the user (0/1 edge in r10231.1) the brake test becomes active, is the motor holding brake (MHB) closed or the request to close the external brake issued. The request to close the brake is indicated again via p10234.6 = 1 or S_ZSW3B.6 = 1. A fault is issued after a maximum of 11 seconds.
- The test torque (test torque ± load torque for hanging axis) is specified during the SBT. For specified n = 0, the control establishes an appropriate test torque opposing the closed brake. Whereby, the test torque is applied as a ramp. The ramp is defined by time p10208.
- At the end of the test sequence, the brake is closed or the request to open the brake issued.
- After deselection of the test sequence (test sequence shutdown), a different test sequence, such as with a different brake in a different direction, can be selected, provided the brake test is still selected.
- For an active test sequence, the brake not currently being tested must remain open.
- After deselection of the SBT, the original speed setpoint becomes active again.

Cancellation/deselection and acknowledgment (SBT):

- An active brake test is canceled by deselecting the test sequence or brake test. Alarm A01782 is issued.

Acknowledging alarms:

- The alarms that affect the brake test can only be safely acknowledged (Failsafe Acknowledge), and possibly only when the brake test is deselected. For "motion control without selection", a POWER ON or the selection/deselection of STO/SS1 (for configured extended alarm acknowledgment) is required for acknowledging.

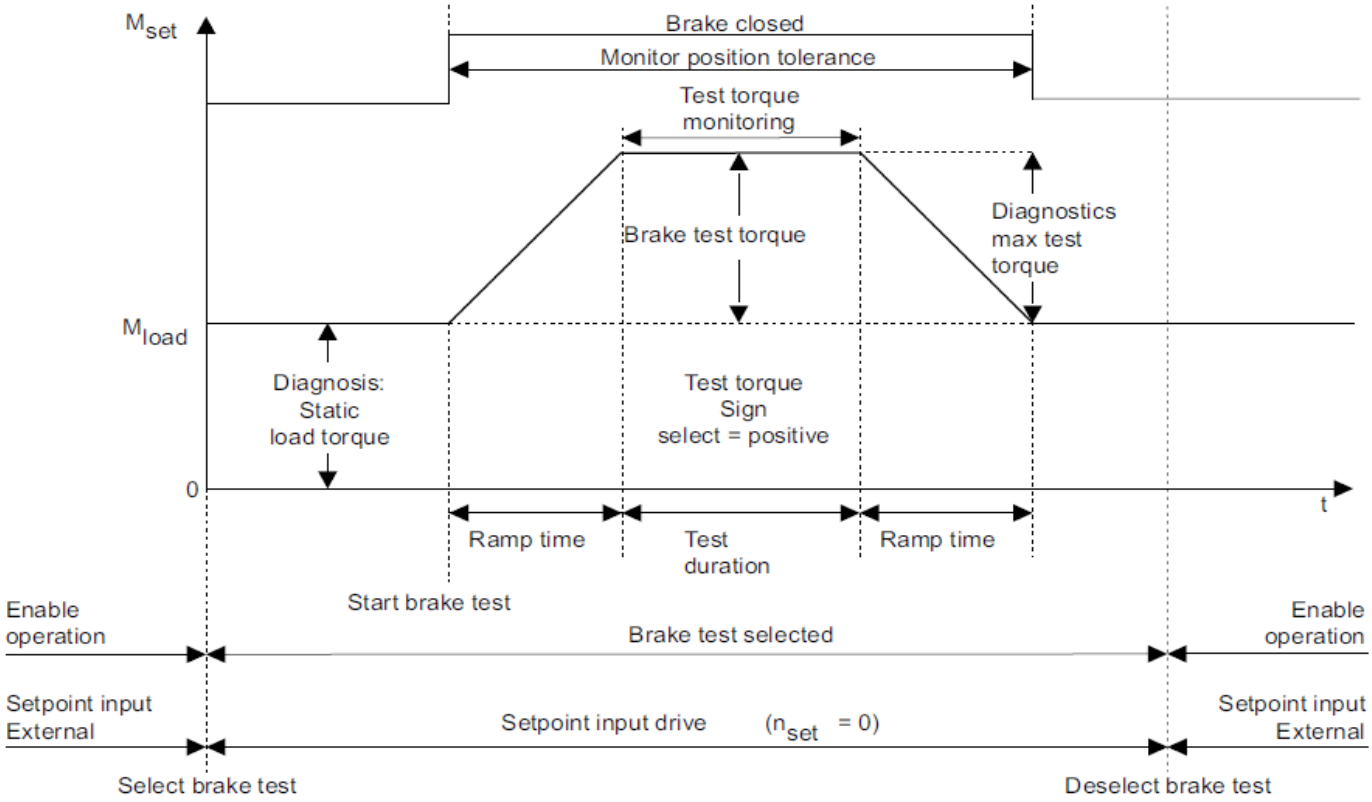
Further details on the SBT function can be found in the **"SINUMERIK ONE Safety Integrated plus Function Manual"**.

The general execution is shown graphically below.



Notes:

Safe Brake Test curve:

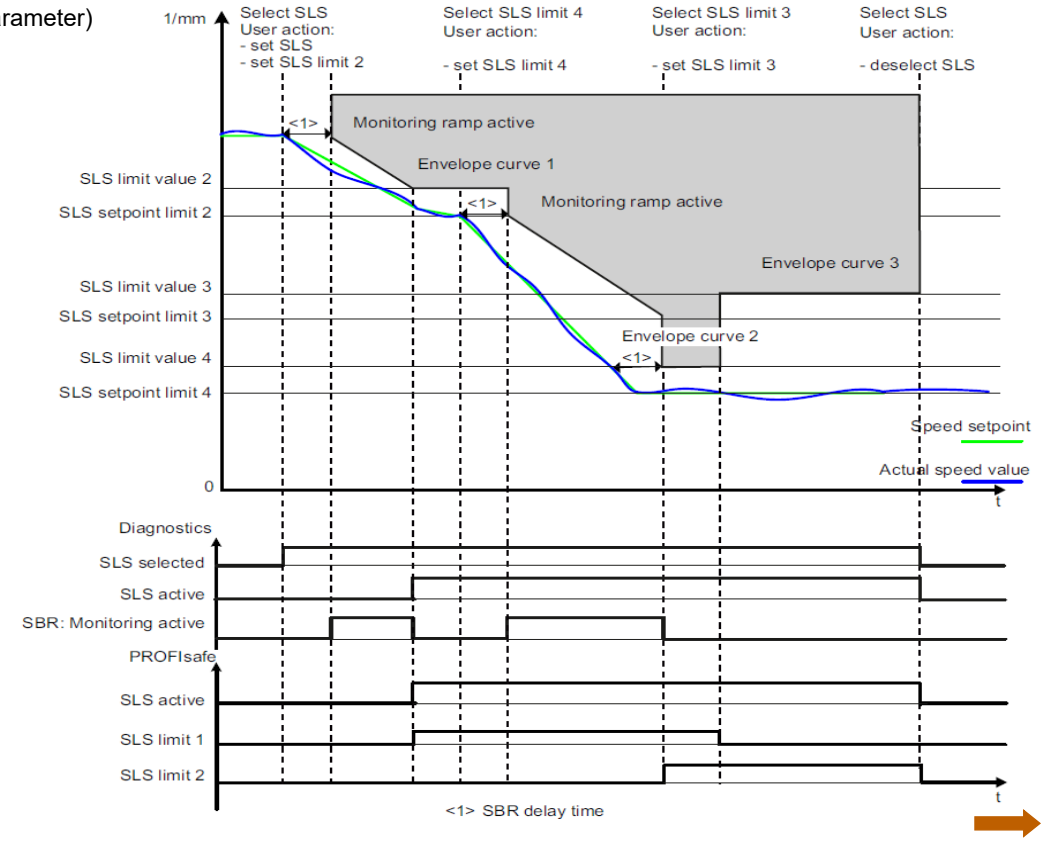


Safe Brake Ramp (SBR): (Parameter)

The Safe Brake Ramp (SBR) function is a **safe monitoring of the brake ramp**.

The Safe Brake Ramp function is used for the **"SS1 with/ without encoder" and "SLS without encoder" functions as well as for STOP B / STOP C (for Safety with encoder) for monitoring the braking operation**.

However with SLS, the setpoint limitation of the Safety Integrated Functions (r9733) must be connected to the ramp-function generator (p1051/p1052).



Notes:

Functional characteristics of SBR:
After triggering SS1, SS2 or SLS, the motor is braked immediately with the OFF3 ramp. After the deceleration time p9582 has expired, the monitoring of the brake ramp is activated. Monitoring is performed to ensure that the motor does not overshoot the set brake ramp (SBR) during the braking operation.

The **safe monitoring of the brake ramp is deactivated:**

- **For SS1:**
  – As soon as the shutdown speed (p9560) is below the threshold.
  – As soon as the SS2 deceleration time (p9552) has expired.

- **For SLS:**
  – As soon as the set brake ramp has reached the new SLS level.
  – As soon as the actual speed has fallen below the new selected SLS level and has remained below this level for the time parameterized in p9582.

Parameterization of the brake ramp:
The rate of rise of the brake ramp is set with p9581 (SI Motion brake ramp reference value) and p9583 (SI Motion brake ramp monitoring time).
Parameter p9581 determines the reference speed and parameter p9583 the ramp down time.
In parameter p9582, the time is set between the triggering of SS1, selection of SLS or SLS level changeover and when the monitoring of the brake ramp takes effect.

Notes:

- ***SBR- and OFF3 curve:***
  The SBR curve should be aligned with the OFF3 curve. In addition, a check should be made as to whether the drive can follow this OFF3 ramp under every load condition.

- ***Limitation of the SBR deceleration time:***
  The SBR deceleration time (p9582) is limited to a minimum value of two SI Motion monitoring cycles (2 x p9500), i.e. even when a time less than 2 x p9500 has been parameterized for the deceleration time (p9582), SBR does not take effect until two safety cycles after the active SS1.

If a value greater than 2 x p9500 is parameterized for the deceleration time (p9582), SBR takes effect after the active SS1 after the p9582 time. Make sure to round off the SBR deceleration time to an integer multiple of the Safety cycle (p9500).

Responses for brake ramp violated (SBR):

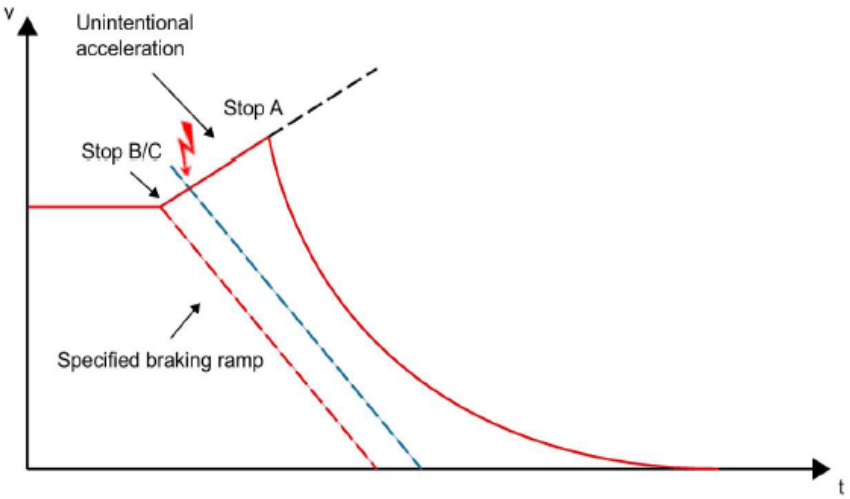
- Safety message C01706 (SI Motion: SAM/SBR limit exceeded).
- Shutdown of the drive with STOP A.
- For an encoder error in a single-encoder system, the "Safe Acceleration Monitor" function is not active. A stop response of Categorie 0 or 1 (EN 60204-1) can be set via parameter p9516.4.

Notes:

Safe Acceleration Monitor (SAM): (Parameter)
The Safe Acceleration Monitor (SAM) function is a safe monitoring of the braking operation on the OFF3 ramp.
The function is active for SS1 and SS2.
As long as the speed decreases, the converter continuously adds the settable tolerance p9548 to the current speed and therefore the monitoring follows the speed.

If the speed temporarily increases,
the monitoring remains at the last value.
The converter reduces the monitoring
until the "Shutdown speed" is reached.

If the speed increases during the braking
operation by the tolerance in p9548,
an STO is triggered immediately and the
relevant drive coasts to a standstill.
Further acceleration is therefore prevented.
The monitoring ends when the speed falls
below the value in p9568.



Functional characteristics of SAM:
- Active for SS1 and SS2.
- SAM is part of the Safety Integrated Extended Functions SS1 and SS2 and therefore cannot be selected separately.
- Monitoring of the speed for increase during the braking operation.
- An STO is triggered as soon as the monitoring responds.

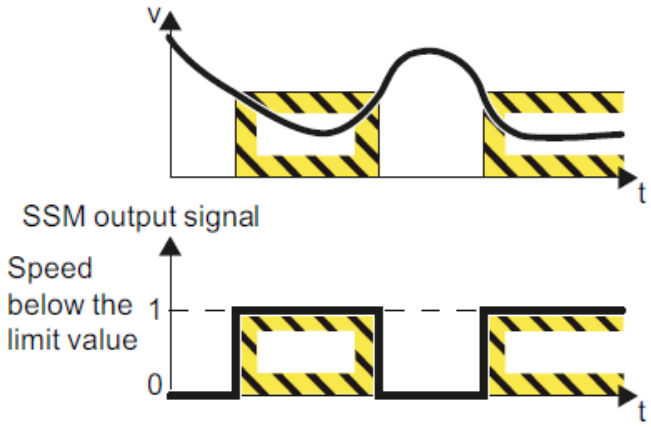
Application cases of SAM:
- Monitoring of the braking operation.

Relationship of SAM and SSM:
If the value 0 is entered for p9568, the speed limit value of the SSM (p9546) function serves simultaneously as the minimum limit value for the SAM (Safe Acceleration Monitor) function.
If the speed is below this limit value, SAM no longer triggers any response of the drive.
In this case therefore, with a relatively high SSM speed limit during the use of the SS1 and SS2 stop functions, the effect of the Safe Acceleration Monitor is severely limited.

Setting of the SAM tolerance of the actual speed:
Set the tolerance so that the "undershoot" that inevitably occurs when standstill is reached during braking on the OFF3 ramp, is tolerated. However, the size of the "undershoot" cannot be calculated.
Further details on the **Calculation of the SAM tolerance** can be found in the "**SINUMERIK ONE Safety Integrated plus Function Manual**".

Notes:

Safe Speed Monitor (SSM): (Parameter)
The Safe Speed Monitor (SSM) function provides a safe output signal when the motor speed (n) is below a specified limit value (n_x) in both directions of rotation.
The SSM safety function cannot be selected or deselected through external control signals. SSM is active when a monitoring speed > 0 is set for SSM.
This safe signal can be used, for example, to unlock a protective door.



Application cases of SSM:
- Release of the protective door only when all drives are stationary.
- A centrifuge may only be filled below a speed defined by a user.

Relationship of SSM and SAM:
If the value 0 is entered for p9568 (SAM shutdown threshold), the speed limit value of the SSM (p9546) function serves simultaneously as the lower limit for the Safe Acceleration Monitor (SAM) function.
In this case therefore, with a relatively high SSM speed limit during the use of the SS1 and SS2 stop functions, the effect of the Safe Acceleration Monitor is severely limited.

Functional characteristics of SSM:
- SSM is a pure signaling function:
  In contrast to other drive-integrated safety functions, overshoot of the SSM limit value does not result in a drive-independent stop response.
- SSM provides a safe output signal when the speed limit value has been reached.

- Parameterizable hysteresis:
  The parameterizable hysteresis ensures that the SSM output signal in the limit range does not jump between the values "0" and "1".
- Settable filter time (PT1 filter).
- Processing of the safe output signal in the safety logic.
- Requirements:
  The SSM safety function cannot be selected or deselected through external control signals. SSM is active when the following settings are made for the relevant drive:
  - Extended Functions are selected as the Safety scope of functions.
  - The speed limit for SSM is parameterized with > 0.

Differences in operation without encoder:
The SSM safety function differs in operation without encoder in the following ways:
- Because of the less precise speed measurement, "Safe Speed Monitor without encoder" requires a greater hysteresis (p9547) and, if required, a filter time (p9545) in comparison to the function with encoder
- With Safe Speed Monitor without encoder, the drive cannot determine the current speed after the pulse suppression. The additional setting "SSM feedback at pulse suppression" is available.

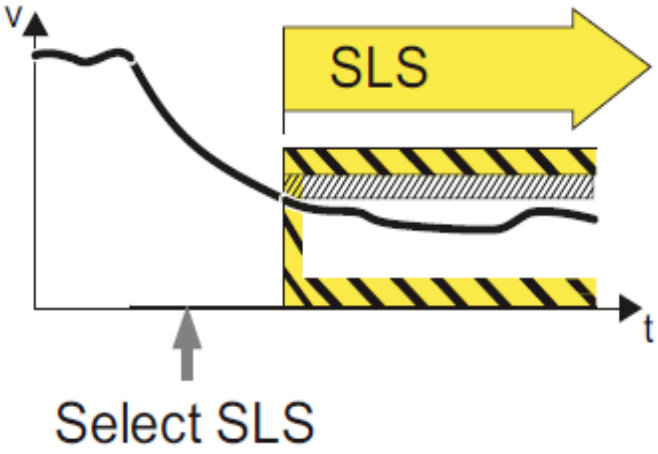
Further details on the SSM function can be found in the **"SINUMERIK ONE Safety Integrated plus Function Manual"**.

WARNING

Danger through unusual behavior of the STOP F for SSM:
A STOP F is displayed by Safety message 201711. STOP F only results in the following STOP B / STOP A response when one of the Safety functions is active. If only the SSM without hysteresis function (i.e. p9501.16 = 0) is active, a STOP F cross-check error does not result in the following STOP B / STOP A response.
- SSM is therefore only considered as an active monitoring function when "SSM with hysteresis" has been activated.

Notes:

Safely-Limited Speed (SLS 1...4): (Parameter)
The Safely-Limited Speed (SLS) function is used for the safe monitoring of the load-side speed of an axis or spindle. Safely-Limited Speed prevents a parameterized speed limit value from being exceeded. The limit values must be specified depending on the risk analysis. Up to four different SLS speed limit values can be parameterized with parameter p9531[0..3], between which it is also possible to switch when SLS is activated (SLS1, SLS2, SLS3, SLS4).



Application cases of SLS:
- The machine operator must be able to access the machine after opening the protective door and slowly traverse a horizontal conveyor in the hazardous area with the aid of an enabling button. > Operator protection
- A spindle drive must not exceed a specific maximum speed depending on the machining tool selection. > Machine protection

Functional characteristics of SLS:
- Safe monitoring of the load-side speed and protection against unwanted high speeds of a drive in both directions of rotation.
- Four different speed values can be parameterized for each drive with parameter p9531[0..3] 4
- An override can also be applied to SLS limit value1. This override can be varied during operation via a PROFIsafe telegram
- SLS sensorless for induction motors.

Procedure at SLS selection:
1. The converter detects the selection of SLS via a failsafe input or via the PROFIsafe safe communication.
2. SLS alows the motor to reduce its speed, which may be too high, within a specified time.
3. SLS monitors the current speed value. The SLS setpoint limit can be transferred to the F-PLC in order to enable a limitation of the speed setpoint there.
4. In order to avoid a limit value violation, the setpoint limit provided by SLS can also be configured as maximum speed in the ramp-function generator. In this case, the speed setpoint is limited through SLS. If the speed setpoint limitation is connected to the ramp-function generator, the converter limits the speed to a value below the SLS monitoring.
5. For SLS without encoder, you can select whether the converter is to monitor the braking of the motor with the SBR (Safe Brake Ramp) or the SAM (Safe Acceleration Monitor) function or not. For SLS with encoder, the SBR function cannot be selected.

SLS deselection:
If the higher-level control deselects SLS, the converter deactivates the limitation and monitoring.

Switchover of the monitoring limits:
With active SLS, you can switch between four different speed levels. The exception is "SLS without selection": In this case there is only one limit.

SLS without selection:
As an alternative to activation via terminals and/or PROFIsafe, you can configure the SLS without selection function. In this case, the SLS function is permanently active after POWER ON.
Details on this can be found in the
"SINUMERIK ONE Safety Integrated plus Function Manual
> Section: Safely-Limited Speed without selection".

Notes:

Safely-Limited Speed (SLS) via PROFIsafe:
An override can also be applied to SLS limit value1.
This override can be varied during operation via a PROFIsafe telegram.
- The first SLS limit value(SLS1) can be influenced via PROFIsafe.
- Value range 1…..32767 (32767 = 100%)
- Speed levels 2, 3 and 4 can also be parameterized and selected in this case.

Application cases of SLS via PROFIsafe:
- Shatter protection, e.g. for grinding wheels
- lathe chucks > Machine protection

Notes:
Deviation of the displayed speed limit:
The SLS speed limit displayed in r9714[2] can deviate slightly from the specified SLS speed limit. The reason for this is the internal resolution (r9732) of the speed values.

Behavior at communication failure:
If p9580 ≠ 0
- p9580 = wait time after which STO is performed after a bus failure > ESR (Extended Stop and Retract) can be performed within the wait time

and SLS is active, the parameterized ESR response is only performed when a STOP with delayed pulse suppression for bus failure (p9563[0...3] ≥ 10) has been parameterized as SLS response.

Speed setpoint limitation and SLS:
- At the same time as the parameterization of SLS, it is useful to configure the speed setpoint limitation. This configuration is performed, e.g. in a higher-level control which evaluates the Safety Information Channel, or by wiring r9733[0/1] to the ramp-function generator speed limits (p1051/p1052).
- The use of the positive and negative setpoint limitation for SLS does not make sense in conjunction with the standard telegram 105 and others: In this combination, the speed setpoint of the standard telegram does not take effect until after the setpoint limitation.

WARNING

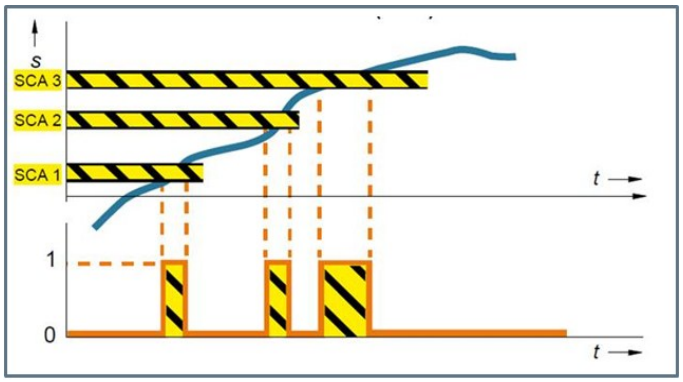
Parameterization of SLS with STOP D or STOP E requires additional acknowledgment via RESET:
For a parameterization of SLS with STOP D or STOP E, the safe acknowledgment and a RESET on the machine control panel are required in the event of an error.
If in the event of an error, only the safe acknowledgment is performed, the spindle restarts immediately. This does not comply with standard IEC61800-5-2. The safe acknowledgment must be linked to a PROFIdrive RESET.

Connection to the motion control in the NCK:
In order that drive-independent Safety functions do not cause follow-on reactions or alarms in the NCK, the following status bits or status values from the SIC/SCC send data are connected to the NCK:
- SI Motion Safety Info Channel status word: S_ZSW1B
  - – Bit 4 SLS_ACTIVE (SLS active)
  - – Bit 6 SLS_SELECTED (SLS selected)
- Speed setpoint limitation in 32-bit resolution with sign bit: S_V_LIMIT_B
  - – If the SLS function has been selected or is active, the corresponding bits are set in S_ZSW1B. The currently selected speed limit value of the drive-independent SLS function is then entered in S_V_LIMIT_B.
  - – After rescaling, SV_LIMIT_B is multiplied by the value from MD36933 $MA_SAFE_DES_VELO_LIMIT[0...3], and forwarded to the motion control in the NCK as speed setpoint limitiation.
  - – If a value of 0% or 100% is entered in MD36933, the value from S_V_LIMIT_B is forwarded as speed setpoint limitation.
  - – With SOS selection or a STOP A to F, S_V_LIMIT_B contains the setpoint 0. If the "SLS" or "SOS" function in the drive is not selected, the speed limit value in S_V_LIMIT_B corresponds to the maximum velocity / maximum speed (Vmax).

Further details on the SLS function can be found in the **"SINUMERIK ONE Safety Integrated plus Function Manual"**.

Notes:

Safe Cam (SCA): (Parameter)
According to the EN 61800-5-2 definition, the "Safe Cam" (SCA) function provides a safe output signal to show whether the position of the motor shaft is within a specified range.



Functional characteristics of SCA:

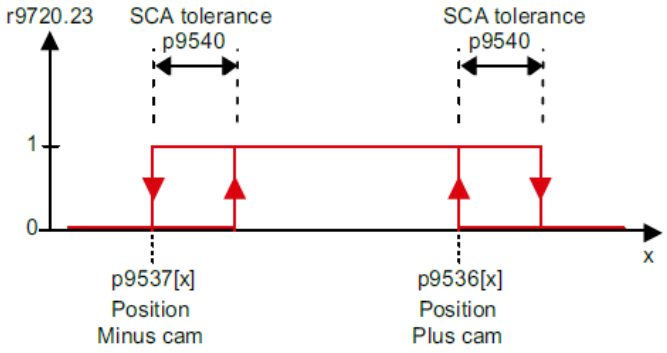
- Provides a safe signal.
- Processing of the signal in the failsafe pogram.
- 30 safe cam pairs (plus/minus) available for each axis.
- The SCA safety function can only be used with an encoder.
- Telegram 903 must be used for SCA.

Application cases of SCA:

- Implementation of safe electronic cams.
- Implementation of a safe range identification.
- Definition of working areas and protection areas.
- Safe switchover of safety functions (e.g. switchover of SG levels).

Specifying cam positions:

- The cam positions to be monitored are permanently specified with parameters p9536[x] and p9537[x] (with x = 0 ... 29).
  The defined cams must have a certain minimum length:
  p9536[x] - p9537[x] ≥ p9540 + p9542
  If this rule is violated, the drive outputs the message F01686 ("SI Motion: Illegal parameterization cam position").
- Due to the different cycle times and runtimes, the cam signals of the two monitoring channels do not switch simultaneously or not precisely at the same position. Therefore, when required, a tolerance band must be specified for all cams via parameter p9540. Within this tolerance band, the monitoring channels can have different signal states of the same cam:.



- The tolerance band for the SCA function should be selected as small as possible (<5 ... 10 mm). The cam tolerance should be parameterized greater than or equal to the actual value tolerance.
- The axis is referenced with the "Safe Referencing" function.

Enabling SCA:

- The SCA function is enabled with p9501.28 = 1.
- Each cam must be enabled individually with p9503.x = 1 (with x = 0 ... 29).

WARNING

Safe referencing:

The enabled cam signals are output immediately after POWER ON. However, this output is only safe after the safe referencing; i.e. the cams are only considered as safe when they have been safely referenced.

- The axis is referenced with the "Safe Referencing" function. See "Safe Referencing"

Selecting SCA:

The SCA function is on with the PROFIsafe control word S_STW2.23. Telegram 903 in which control word S_STW2 and status word S_ZSW_CAM1 are available for SCA, must be used.

Notes:

Cam synchronization:

The cam signals of the two monitoring channels are synchronized for the transfer of the cam status word via PROFIsafe to the F-host. Whether a different cam signal from the second channel is plausible, is also monitored. If the drive detects an error, it outputs message C01711 with fault value 1014.

The setting of the tolerance for the cross-check comparison of the actual position between the two monitoring channels in p9542 ("Actual value comparison tolerance") is used as position tolerance for the monitoring of the cam positions.

Transfer via PROFIsafe:

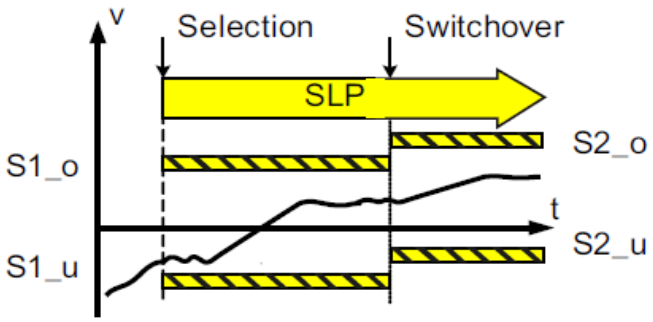
After SCA has been parameterized and selected, the results of the monitoring are transferred in status word S_ZSW_CAM1

(see "Section: Control of drive-integrated safety functions > S_ZSW_CAM1)

Notes:

Safety Limited Position (SLP 1...2): (Parameter)
An axis-specific traversing range limitation can be implemented with the Safety Limited Position (SLP) safety function. Unreliable and expensive hardware limit switches are therefore not needed.
Two different safe software limit switch pairs (SLP1, SLP2) are available between which switching is possible.
If one of the safe software limit switches is traversed over, Safety Integrated triggers a stop function parameterized by the machine manufacturer.
Definition in accordance with EN 61800-5-2:

- The SLP function prevents the motor shaft from exceeding the specified position limitation(s).



Functional characteristics of SLP 1...2:

- Selection via safe terminals (onboard F-DI) or PROFIsafe.
- Two position ranges, each defined by a limit switch pair (p9534 and p9535).
- Safe switching between the two position ranges (not available for PROFIsafe telegram 30).
- When the position of one of the two limit switches is exceeded, a configurable stop reaction (p9562; Stop A, stop B, stop C, stop D or stop e) and the safety message C01715 output.
- To acknowledge this fault, either a switch must be made to a range whose limits are not violated or the SLP function deselected. After the acknowledgment, the drive can traverse again in the permitted range.
  Safety-oriented traversal in the permitted range is possible with the "Retract" function, see "Retract".
- The drive must be safely referenced.
- Replacement for hardware limit switches.

Application cases of SLP:

- Traversing range limitations for axes.
- The drive must not exit the specified position ranges.
- Safe software limit switches.

Prerequisites for the SLP function:

- Usage of one or two suitable encoders for the extended safety functions with encoder.
  See "Safe actual value acquisition"
- Determination of the absolute position of the drive by referencing during commissioning and following all actions after which a safe absolute referencing can no longer be guaranteed (POWER ON, parking).
  See "Safe referencing"

Enabling the SLP function:

- The SLP function is enabled with p9501.1 = 1.
- A POWER ON at the converter must be performed after the enabling.

No actual value synchronization for SLP:
The concurrent enabling of the SLP function and the actual value synchronization (p9501.3 = 1) is not permitted:
In this case, the drive issues fault F01688.

Control and status signals for SLP:
The selection of SLP and the switching between the position range is made via an F-DI or a PROFIsafe activation bit. The SLP selection can be tested via parameter r9720.6. The selected position range can be checked via parameter r9720.19. If SLP is active, status bit r9722.6 is set. The active position range is indicated by r9722.19. The observance of the upper and lower active SLP limits can be tested via r9722.30 and r9722.31, respectively.

Jumping in the display:
No hysteresis is provided for r9722.30 and r9722.31. This means small fluctuations around the range limit can cause jumping in the display.

Notes:

Control of the Safety Limited Position function using PROFIsafe:

- The SLP is selected or deselected via the control word S_STW1.6 or S_STW2.6, respectively.
- The limit switch pairs are switched via control word S_STW2.19.
- S_ZSW2.23 indicates whether the current position is "safe"; the bit is set, for example, only when "safe referencing" has been performed.
- Bit 6 of status word S_ZSW1.6 or S_ZSW2.6 indicates whether SLP is active. The bit is set only when SLP is selected and the axis has the "safely referenced" status.
- Status word S_ZSW2.19 indicates which SLP limit switch pair is active. This indicator is valid only when SLP itself is active.
- S_ZSW2.30 and S_ZSW2.31 indicate whether the upper limit and lower limit of the active position range are observed.

Extended functions via PROFIsafe:

- The status signal "SLP active" (S_ZSW1.6 or S_ZSW2.6) is not identical with the diagnostic signal "SLP active" (r9722.6), but rather identical with the AND logic operation between "SLP active" (r9722.6) and "safely referenced" (r9722.23).
- The other SLP status signals S_ZSW2.19 "SLP active position range", S_ZSW2.30 "SLP upper limit observed" and S_ZSW2.31 "SLP lower limit observed" match the corresponding bits in r9722.

Restrictions for PROFIsafe telegram 30:

The use of PROFIsafe telegram 30 (with the 16-bit words S_STW1 and S_ZSW1) involves the following restrictions:

- Only position range 1 is available.
- Switching to position range 2 is not possible.
- The status feedbacks "safely referenced", "active position range", "upper SLP limit observed" and "lower SLP limit observed" are not available.

"Extended functions via F-DI", see "**SINUMERIK ONE Safety Integrated plus Function Manual"**

Retract:

After a limit of the active traversal range has been exceeded, the drive must be returned to the permitted range. In this case, a safe acknowledgment would only cause the safety messages to be reissued, the movement of the drive remains suppressed. If switching to another traversal range is not permitted, only deselection of SLP remains. This, however, would have the disadvantage that there would be no monitoring as to whether the drive moves in the direction of the permitted traversal range.

Consequently, it is better to implement a retract function with the following procedure:

Safety commissioning of the retract functionality:

1. Parameterize SLP completely
2. Parameterize SDI completely
3. Perform an acceptance test for both functions

Activate retract via PROFIsafe:

Implement a **user program in the F-CPU** that performs the retract function with the following steps:

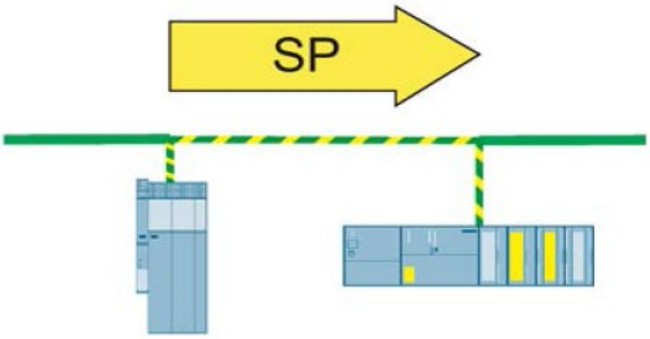
- Select **SDI positive** for violation of the lower SLP limit or **SDI negative** for violation of the upper SLP limit .
- Wait until the selected SDI is active, then **deselect SLP**.
- Safe **acknowledge** of the limit value violation
- **Move the drive** with appropriate setpoint specifications in the enabled range.
- **Select SLP.**
- Wait until the SLP is active, then **deselect SDI**.

A **"Retract FAQs" link** is provided in "Section: Further information > Overview of the safety functions for SINUMERIK ONE"

"Activation via F-DI (onboard terminals)", see **"SINUMERIK ONE Safety Integrated plus Function Manual"**.

Notes:

Safe Position (SP): (Parameter)
The function enables safe position values to be transferred via PROFIsafe (telegrams 901 or 902) to the higher-level failsafe control (F-PLC).
The F-PLC can also calculate the velocity from the change of position over time. In telegram 902, the values are transferred in 32-bit format and in telegram 901 in 16-bit format.
The actual position value is not transferred in PROFIsafe telegram 903, which is the default setting for SINUMERIK.
See: "Section: Control of drive-integrated SI functions / telegrams"



Application cases of SP:
- Implementation of more than 30 safe cams
- Calculation of the current velocity

After the parameterization, enabling and POWER ON, the function is selected automatically and the values are transferred. The following must be taken into account:
- **The "Absolute position" must also be enabled and then safely referenced for use as safe absolute position.**
  This means that the reference to the absolute position must be established at the control level.
  The "safely referenced" bit of the SINAMICS S120 (r9722.23) cannot be used in this case.
- **The actual position value must be valid so that the transferred position can be used.**

The velocity can also be calculated from the position values with the aid of the time stamp that is also transferred.
If only the velocity is to be calculated, it is sufficient to enable the "Transfer of safe position values" without the "Absolute position".

Enable of the "Transfer of safe position values" by means of the following steps:
- **Enable of the Safety Integrated Extended Functions:**
– p9601 = 12 = C hex (= Extended Functions via PROFIsafe)
– p9601 = 13 = D hex (= Extended Functions via PROFIsafe and Basic Funtions via onboard terminals).
- Enable of the "Transfer of the **safe absolute position** with the option for calculation of the velocity by the control":
– Selection of one of the PROFIsafe telegrams 901 or 902 (p60022, p9611, p9811)
– p9501.2 = 1 (= Enable of the absolute position)
– p9501.25 = 1 (= Enable of the transfer of the safe position via PROFIsafe)
- Enable of the "Transfer of a **safe relative position**" only for the calculation of the velocity by the control:
– Selection of one of the PROFIsafe telegrams 901 or 902
– p9501.25 = 1
- A **POWER ON** of the converter must be performed after the enable.

Functional characteristics of SP:
- **Transfer of safe absolute position values:**
  - If the transfer of the safe relative position through p9501.25 = 1 and p9501.2 = 0 has been enabled, the validity of the safe relative position is displayed by the set bit S_ZSW2.22.
  - If the transfer of the safe absolute position through p9501.25 = 1 and p9501.2 = 1 has been enabled, S_ZSW2.22 is only set when the drive has also been safely referenced.
- **Transfer of safe relative position values**
  (e.g. for the calculation of the velocity):
  - For the calculation of the velocity, only S_ZSW2.22 (r9722.22, actual position value valid) has to be set.

Setting of the modulo value for rotary axes:
- p9505 **defines the modulo range of a safety rotary axis (p9502 = 1)** when the transfer of a safe absolute position has been enabled (p9501.2 = 1 and p9501.25 = 1).
  The parameterization of the modulo value can result in a jump of the actual position value when the displayable range overflows. For this reason, p9505 must only be parameterized in steps of 2n × 360° (n = 1, 2, 3, …).

Notes:

In all other cases, the converter outputs alarm A01794. This alarm can be hidden for the case that the possible jump of the actual position value can be tolerated for the relevant application, or does not represent a problem.

- **With p9505 = 0, the modulo function is disabled**. This parameter is not relevant for a safety linear axis (p9502 = 0) or for the enabled transfer of a safe relative position (p9501.2 = 0 and p9501.25 = 1).
- **If SLP has been enabled** (p9501.1 = 1), **the modulo function must be disabled** (p9505 = 0).

Transfer formats, value range and resolution for telegram 901 (16-bit format):
see **"SINUMERIK ONE Safety Integrated plus Function Manual"**

Transfer formats, value range and resolution for telegram 902 (32-bit format):

- Transferred value ranges.
- Display of the diagnostics information in parameter r9708 with the following properties according to the table:

	Linear axis	**Rotary axis**
Position values	±737280000	±737280000
Unit	1 µm	0.001°
Comment	Monitoring ±737.280 µm with accuracy of 1 µm	= 2048 revolutions

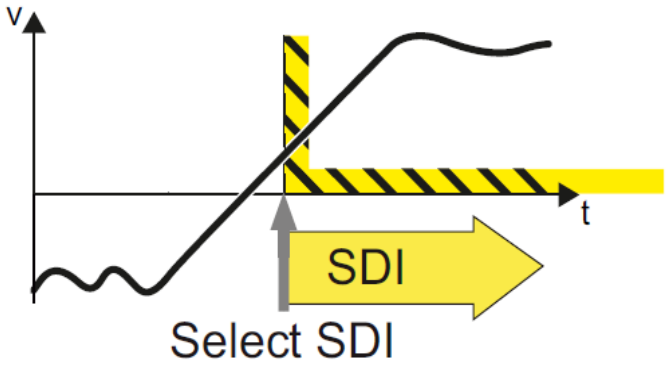
Velocity calculation:
The velocity must be calculated as follows by the control from the change in position:

- Posdiff = Posnew - Posold
- Clockdiff = Clockcountnew - Clockcountold
- Timediff = Clockdiff × Safety cycle (the velocity calculated last must be used for Clockdiff = 0)
- v = Posdiff/Timediff
- Format v

An ***Acceptance test*** is not required for the "Transfer safe position values“ function, but the function that was implemented with SP must be accepted in the higher-level control.

Notes:

Safe Direction (SDI): (Parameter)
The Safe Direction function enables the safe monitoring of the direction of motion of the drive. When this function is activated, the drive can only move in the enabled direction.



After the selection of SDI via terminals or PROFIsafe, the delay time p9565 is started. During this time, you can make sure that the drive moves in the enabled direction.
Thereafter, the Safe Direction function is active and the direction of motion is monitored.

If the drive now moves in the blocked direction by more than the configured tolerance (p9564), message C01716 is output and the stop response specified in p9566 is started.
To acknowledge the messages, SDI must be deselected and the cause of the error corrected, and then the messages safely acknowledged. SDI can then be selected again.

SDI selection and deselection:
As soon as the converter has detected the SDI selection:
- A settable delay time starts, within which the control must ensure that the converter moves in the enabled (safe) direction.
- If a settable tolerance is active, within which the converter tolerates a movement in a non-enabled (unsafe) direction, the triggering of faults during braking (overshoot) as well as during controlled standstill, can be avoided.
- The converter monitors the direction of motion of the drive after the delay time has expired.
- The converter outputs a message when it now moves in the disabled direction by more than the configured tolerance. The specified stop response is then started.

Functional characteristics of SDI:
- SDI monitors the current direction of motion.
- The SDI setpoint limit can be transferred to the higher-level control in order to allow a limitation of the speed setpoint there.
- In addition, the setpoint limit provided by SDI can also be configured in the ramp-function generator as maximum speed. In this case, SDI limits the speed setpoint to the permissible direction.
- The disabling of the positive or the negative direction of rotation can be selected via two failsafe signals (F-DIs or PROFIsafe).
- The "Safe Direction" function is enabled with p9501.17 = 1.
- Parameters r9720.12 and r9720.13 show whether the SDI function has been selected.
- Parameters r9722.12 and r9722.13 show whether the SDI function is active.
- The tolerance is set with parameter p9564 within which the converter tolerates a movement in a non-enabled (unsafe) direction.
- The stop response in the event of an error is specified with parameter p9566.
- The following value is set automatically through the selection of "SDI positive":
  r9733[1] = 0 (negative setpoint limitation)
- The following value is set automatically through the selection of "SDI negative":
  r9733[0] = 0 (positive setpoint limitation)
- The absolute speed setpoint limitation is available in r9733[2].

SDI without selection:
As an alternative to activation via terminals and/or PROFIsafe, you can configure SDI without selection. In this case, SDI is permanently active after POWER ON.

Application cases of SDI:
- A protective door must only be opened when the drive moves in a safe direction (away from the operator).
- At the plate change of impression cyclinders, the drive must only move in the safe direction.
- A rolling door must only run in one direction when the anti-pinch protection is triggered.
- A crane trolley must only start in the opposite direction at the operational limit switch.

Notes:

Safe Referencing: (Parameter)
The Safe referencing function enables **a safe absolute position to be specified by means of user agreement**.
This safe position is required for the following functions:
- SLP (**Safely-Limited Position**)
- SP (**Safe Position**)
- SCA (**Safe Cam**)

In most cases, the referencing of an absolute position is performed by an external control. The converter only performs this task in special cases (e.g. EPOS).

- ***Referencing through SINUMERIK:***
  Requirement:
  p9501.27 = Referencing enable via SCC = 1
  (Otherwise it is not possible to set the user agreement in the HMI screen)

- ***Referencing through external control:***
  Requirement: No movement of the drive.
  The referencing position determined by the control is entered in parameter p9572 and declared as valid by p9573 = 89.

- ***Referencing through EPOS:***
  The SINAMICS function EPOS transfers the determined position directly to Safety Integrated during referencing. This can also be performed during a movement.

- ***User agreement:***
  Within a certain time (see below), the user agreement must be set after the referencing (p9726 = p9740 = AC hex).

Safety Integrated only evaluates the referencing position when this is required by an enabled function (e.g. SLP).
With the diagnostics bit r9723.17, Safety Integrated signals whether the drive has been referenced.
Safety Integrated displays the position of the drive in the diagnostics parameters r9708 and r9713. Bit r9722.23 is set when the axis is safely referenced.

Referencing types:
SINAMICS distinguishes between two types of referencing:

- ***Initial referencing:***
  The following steps are required for the first Safe Referencing, or in the event of an error, for the follow-up referencing:
  – The **referencing position** determined by the control **is entered in p9572 and declared as valid with p9573 = 89.** This step is not required for position control via EPOS.
  – **Referencing has been performed correctly (r9723.17 = 1)**
  – **Confirm the actual position value:**
  Set parameter **p9726 = p9740 = AC** hex
  – If after setting p9726, you do not **set p9740 within 4 s**, the converter outputs the messages C01711 (value: 1002), C30711(value: 0) and possibly additional messages. The user agreement is deleted in both safety channels.
  – If after setting p9740, you do not set p9726 within 2 s, the converter outputs the messages C01711 (value: 0), C30711 (value: 1002) and possibly additional messages. The user agreement is deleted in both safety channels.
  After this "User agreement", the drive is "safely Referenced" (r9722.23 = 1).

Note:
No automatic user agreement permitted:
Please note that the operator must be able to assign the real position of the axis to the determined position before setting the user agreement. This can be performed, for example, by a visual check of the axis position. These parameters must not be set fully automatically by the control without the user agreement.
This would only be permitted when the referencing position can be reliably acquired with a failsafe sensor.



Notes:

- ***Follow-up referencing:***

  Follow-up referencing is referencing with a safe previous history (i.e. with an internally buffered user agreement), after a POWER ON or after the deselection of "Parking axis".

  – The **position** determined by the control **is entered in p9572 and declared as valid with p9573 = 89.** This step is not required for position control via EPOS and use of an absolute encoder.

  – After referencing the drive, Safety Integrated automatically performs a **plausibility check**.

  – If the difference between the current absolute position and the standstill position stored previously by Safety Integrated in the NVRAM is within the tolerance p9544, the drive goes into the "safely referenced" state via (r9722.23 = 1).



Notes:

Safe actual value sensing:

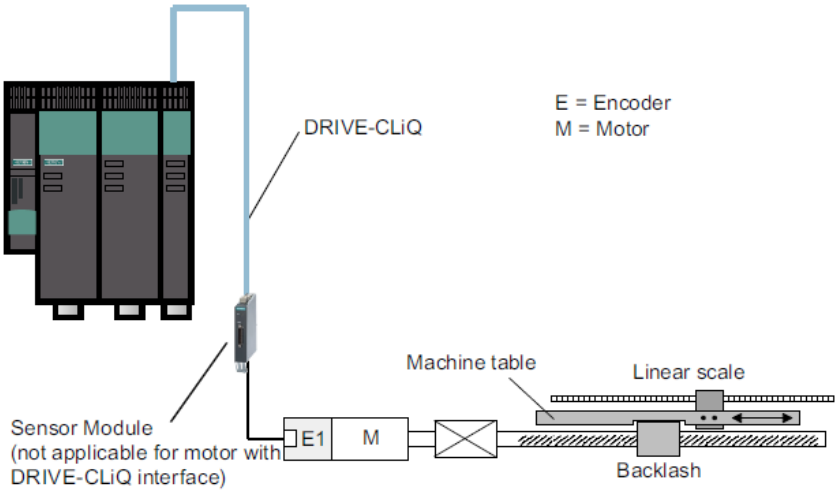
Rules for connecting an encoder:
The valid rules must be observed when connecting an encoder:
(See SINAMICS S120 Drive Functions Function Manual)

In principle, the following can be used for safe speed/position sensing:
- 1-encoder systems
- 2-encoder systems

1-encoder systems:
In a 1-encoder system, only the motor encoder is used for the safe sensing of the drive actual values. This motor encoder must be suitable (see "Section: Further information > Drive-integrated safety functions of Sinumerik ONE").
The actual values are directly safely generated in the encoder or in the Sensor Module and supplied to the Control Unit via DRIVE-CLiQ.
For motors without a DRIVE-CLiQ interface, the connection is via additional Sensor Modules.

Note:
No "Safe Acceleration Monitor" for encoder errors in a 1-encoder system:
The "Safe Acceleration Monitor" (p9506 = 3) function is not active for an encoder error in a 1-encoder system.
A stop response of Category 0 or 1 (EN 60204-1) can be set via parameter p9516.4.

Special features of linear motors:
For linear motors, the motor encoder (linear scale) corresponds to the measuring system on the load.
Therefore only one measuring system is required.
Connection is via a Sensor Module or directly via DRIVE-CLiQ.

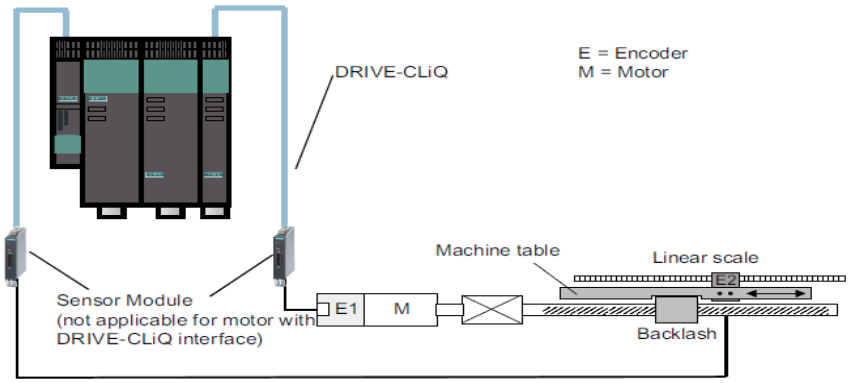


2-encoder system:
The safe actual values for a drive are supplied by two separate encoders in this system. The actual values are transferred to the Control Unit via DRIVE-CLiQ. For motors without a DRIVE-CLiQ interface, the connection is via addiitonal Sensor Modules (see "Section: Further information > Drive-integrated safety functions of Sinumerik ONE")

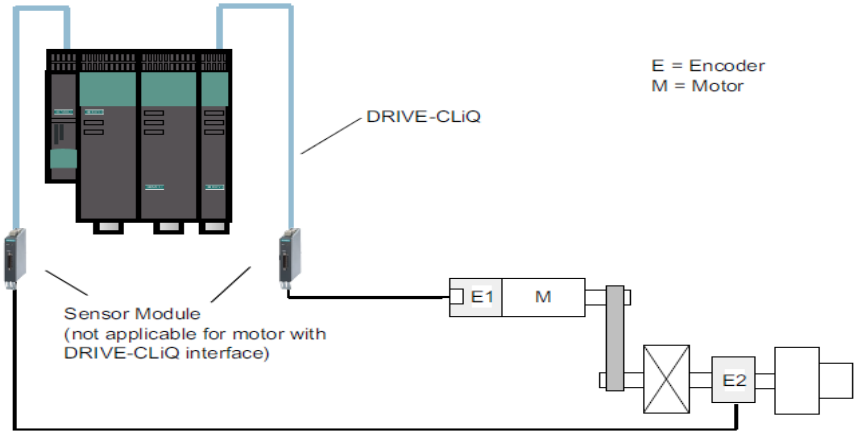
Notes:

Here are two examples:

- ***2-encoder system on a linear axis via a ball screw:***



- ***2-encoder system on a rotary axis:***



For the parameterization of a 2-encoder system with Safety Integrated,
parameters p9315 to p9329 must be aligned with parameters r0401 to r0474.

Notes:

- ***Assignment of the encoder parameters:***
  Parameters **p95xx** are assigned to the **1st encoder**, parameters **p93xx** to the **2nd encoder**.

- ***Transfer of the values from the encoder commissioning:***
  To transfer the parameters ascertained in the encoder commissioning to the Safety parameterization, parameter **p9700 = 46 (2E hex)** must be set. This copy function is **only possible online**.

The following table shows the ***encoder parameters and the corresponding Safety parameters for 2-encoder systems:***

Safety parameter	Designation	Encoder parameter
p9315 / p9515	**SI Motion coarse position value configuration:**	
p9315.0 / p9515.0	**Up-counter**	r0474[x].0
p9315.1 / p9515.1	**Encoder CRC least significant byte first**	r0474[x].1
p9315.2 / p9515.2	**Redundant coarse position value most significant bit left-justified**	r0474[x].2
p9315.16 / p9515.16	**DRIVE-CLiQ encoder**	p0404[x].10

Notes:

p9316 / p9516	SI Motion encoder configuration safe functions:	
p9316.0 / p9516.0	**Motor encoder rotary/linear**	p0404[x].0
p9316.1 / p9516.1	**Actual position value sign change**	p0410[x]
p9317 / p9517	**SI Motion linear scale lattice pitch**	p0407
p9318 / p9518	**SI Motion encoder pulses per revolution**	p0408
p9319 / p9519	**SI Motion fine resolution G1_XIST1**	p0418
p9320 / p9520	**SI Motion leadscrew pitch**	Encoder parameterization mask
p9321 / p9521	**SI Motion gearbox encoder**	Encoder parameterization mask
p9322 / p9522	**SI Motion gearbox encoder**	Encoder parameterization mask
p9323 / p9523	**Redundant coarse position value valid bits**	r0470
p9324 / p9524	**Redundant coarse position value fine resolution bits**	r0471
p9325 / p9525	**Redundant coarse position value relevant bits**	r0472
p9326 / p9526	**SI Motion encoder assignment**	Encoder parameterization mask
p9328 / p9528	**SI Motion Sensor Module node identifier**	
p9329 / p9529	**Gx_XIST1 coarse position safe most significant bit**	p0415 = r0470 – r0471

Encoder types for 1- and 2-encoder systems:

Incremental encoders or absolute encodrs can be used for the safe acquisition of position values on a drive.
The absolute position values can be transferred to the control via the serial EnDat interface or the SSI interface. However, they are not evaluated by the safety functions.

In encoder systems with SINAMICS Safety Integrated (1- and 2-encoder systems), **the following encoders are permitted** for the safe actual value sensing:

- ***Encoders with sin/cos 1 Vpp signals:***
  - 1- and 2-encoder systems
  - Connected to the SINAMICS Sensor Modules SME20/25, SME120/125 and SMC20
  - The encoders must have a pure analog signal processing and generation. This is required in order to avoid the A/B track signals with valid levels becoming static (freezing).
- ***HTL/TTL encoders:***
  - Can only be used in 2-encoder systems. Whereby, one encoder must be an HTL/TTL encoder. The other encoder can be a sin/cos encoder or an HTL/TTL encoder.
  - Connected to a Sensor Module Cabinet SMC30 or the onboard interface of the CU310-2, SINAMICS HLA or SINAMICS S120 Combi.
  - An HTL/TTL encoder connected to the onboard interface of the CU310-2, SINAMICS HLA or SINAMICS S120 Combi, must not be operated as first encoder.
  - Note the lowest possible velocity resulotion for an HTL/TTL encoder system (r9732[1]).

See also "Section: Further information
> Overview: SINUMERIK Safety Integrated plus"

Notes:

Encoders with integrated DRIVE-CLiQ interface:

These encoders must be certified at least according to IEC 61800-5-2 (SIL2) or ISO 13849-1 (Performance Level d /Category 3).

Notes:

Failure Mode Effects Analysis (FMEA):
A Failure Mode Effects Analysis (FMEA) must be performed **for the mounting of the encoder on the motor shaft or on the linear drive**. The result must be that a failure of the encoder mounting can be excluded (refer to DIN EN 61800-5-2, 2008, Table D.16). If the encoder mounting were to fail, the encoder could no longer measure the motion correctly.
Note that the fulfillment of the above requirments is the sole responsibility of the machine manufacturer.
The information about the internal implementation of the encoder must be supplied by the encoder manufacturer.
The FMEA must be performed by the machine manufacturer.

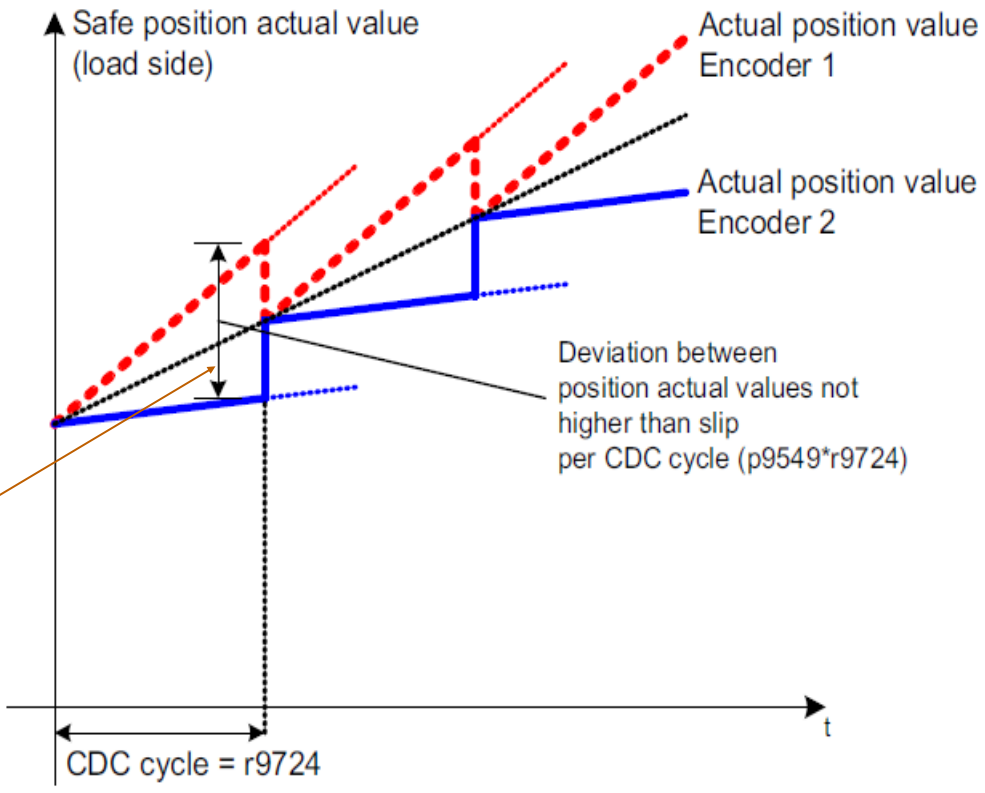
Siemens motors with and without DRIVE-CLiQ connection that can be used for Safety Integrated Functions.
With these motors, the mounting of the encoder on the motor shaft can be considered as safe and failure of this mounting can be excluded:
see "Section: Further information > Drive-integrated safety functions of Sinumerik ONE"

Basic absolute encoders with EnDat interface and additional sin/cos tracks are not permitted:
Basic absolute encoders (e.g. EQI) that have an EnDat interface with additional sin/cos tracks, but work internally according to an inductive measuring principle, are not permitted for SINAMICS Safety Integrated.

Encoder types for SINAMICS HLA:
The following encoder types are permitted for SINAMICS HLA:
- **1-encoder systems:**
  – Safety-capable DRIVE-CLiQ encoders
  – sin/cos encoders connected via SME20/25, SME120/125 or SMC20 (1 Vpp, pure analog signal processing)
- **2-encoder systeme:**
  – Encoders with DRIVE-CLiQ connection
  – sin/cos encoders connected via SME20/25, SME120/125 or SMC20 (1 Vpp, pure analog signal processing)
  – HTL/TTL encoders connected via SMC30 (not in conjunction with SINUMERIK)
  – TTL encoders connected via onboard interface of the HLA module (not in conjunction with SINUMERIK)

Actual value synchronization:
With the activation of the actual value synchronization (p9301.3 = 1), e.g. for systems or machines with slip, the actual values of both channels are averaged cyclically.
The maximum slip in p9549 is monitored in the cross-check cycle (r9724).
If the "Actual value synchronization" has not been enabled, the value parameterized in p9542 is used as tolerance in the cross-check.

This difference must not become greater than the position difference that can occur with maximum slip (p9549) during a cross-check cycle (r9724).



Notes:

Safe motion control:
The properties of the actual value acquisition determine (together with the deployed encoders) the optimum values that must be observed for safe motion control:

- ***Safe maximum speed (r9730):***
  The maximum speed (load side) permitted by the acquisition of the actual values for the safe motion control functions is indicated in r9730. The parameter indicates to which load speed, the safe actual encoder values (redundant encoder coarse position) can still be acquired correctly by the associated encoder parameterization.
  The actual value acquisition cycle (p9511) determines the frequency with which the actual values can be acquired. The faster the selected cycle, the faster the "safe maximum speed". However, a faster actual value acquisition cycle increases the load on the Control Unit. This situation must be considered when setting the optimum for the associated application.
- ***Position accuracy (r9731):***
  This position accuracy is best achieved by acquisition of the actual values. In the case of a 2-encoder system, the accuracy of the worst encoder is indicated by the number of encoder marks.

Safe gearbox switching. See "**SINUMERIK ONE Safety Integrated plus Function Manual**

Forced dormant error detection:
To satisfy the requirements from EN ISO 13849-1 and IEC 61508 for timely fault detection, the **functions and the shutdown paths within a time interval** must be **tested** for correct operation at least once.

The **maximum permitted interval** for test stop / forced dormant error detection for the **basic and extended functions** is **8760 hours**; i.e. the forced dormant error detection must be performed **at least once per year**.

This must be implemented by the **cyclical manual or process-automated triggering** of test stop / forced dormant error detection.

The test stop cycle is monitored; on expiration of the parameterized timer (also after POWER ON / warm start), **alarm A01697: "SI Motion: Test of the motion monitoring required"** is triggered and a status bit is set that can be applied to an output or a PZD bit via BICO. This alarm does not impair the operation of the machine.

Examples for the time of execution:
When a machine is running, it can be assumed that the appropriate safety equipment (e.g. protective doors) ensures there is no danger for persons. Consequently, the user is informed of the required forced dormant error detection only with an alarm and so requested to perform the forced dormant error detection at the next possible opportunity (some examples are shown below).

- For stationary drives after switching on the plant (POWER ON).
- Before opening the protective door.
- In a specified time period (e.g. in an 8-hour time period).
- In automatic operation, time- and event-dependent.

Notes:

Options and activation:

Prerequisites:

- A test stop of the safety functions triggers an STO.
  STO must not be selected before selecting the test stop.
- When using Blocksize Power Modules, the test stop must be triggered in controlled standstill
  (speed specification 0, motor supplied with current).

The forced dormant error detection (test stop) can either be triggered applicative at any time or automatically at POWER ON. The scope of the forced dormant error detection / test stop function is identical in all cases.
The parameterization and activation are different:

- ***Applicative execution with activation via Safety Control Channel (SCC):***
  (integrated drives, external drives):
  The following parameters are relevant:
  – p9559 SI Motion forced dormant error detection timer (Control Unit)
  – p9705 BI: SI Motion test stop signal source
  – r9723.0 CO/BO: SI Motion drive-integrated diagnostic signals

  See: "Section: Drive-integrated SI functions / telegrams activation"
  - If the test stop is performed as described, the action does not require a POWER ON. The acknowledgment is made by deselecting the test stop request.

- ***Applicative execution with activation via F-DI/DO (only external drives):***
  – The forced dormant error detection (test stop) for external drives can be activated applicative via F-DI/DO.
  – The forced dormant error detection for integrated drives of the NCU (SINAMICS Integrated, NX modules) can not be activated applicative via F-DI/DO, but only via SCC.

  To perform the forced dormant error detection for external drives applicative via F-DI/DO, the single-channel parameter p9705 is wired via BICO either to the input terminal on the drive unit (Control Unit) or to a bit of any PZD. If the test stop is performed as described, the action does not require a POWER ON. The acknowledgment is made by deselecting the test stop request.

- ***Automatic execution for POWER ON:***
  – If an automatic test stop of the Safety Integrated Extended Functions as well as the automatic test of the F-DO should be performed for CU310-2, p9507.6 = 1 must be set.
  – For the test of the F-DO of the CU310-2, p10042 must be parameterized and the test activated in p10046.
  Note: Automatic test stop and SBT:
  The automatic test stop of the Safety Integrated Ex tended Functions together with the function "Select brake test for test stop" (p10203 = 2) is possible.
  – If the test stop was parameterized for POWER ON, a test stop can nevertheless still be triggered applicative at any time.
  – If the automatically triggered function cannot be completed correctly because of a problem (e.g. communications failure), the function is restarted automatically after rectification of the problem.
  – After successfully running forced dormant error detection / test stop, the converter enters the "operational" status.
  – The automatic forced dormant error detection / test stop resets timer p9559.
  – The automatic test stop for POWER ON does not affect the Safety Integrated Functions.



Notes:

Overview of important parameters:
(see SINAMICS S120/S150 List Manual)

STO, SS1, SS1e:
(Safe Torque Off, Safe Stop 1, Safe Stop 1 external)

- p9601 SI enable drive-integrated functions (Control Unit)
- p1135[0...n] OFF3 ramp-down time
- p1217 Motor holding brake closing time
- p1228 Pulse suppression delay time
- p9652 SI Safe Stop 1 delay time (Control Unit)
- p9506 SI Motion function specification (Control Unit)
- p9556 SI Motion STOP A delay time Extend. Fct
- p9560 SI Motion STO shutdown speed (Control Unit)
- Only for "Safe Stop 1 (time-controlled) with external stop"
  p9653 SI Safe Stop 1 drive-based braking response
- Only for SS1 (Extended Functions) with external stop:
  p9507 SI Motion function configuration (Control Unit)
- r9720 CO/BO: SI Motion drive-integrated control signals
- r9722 0...15 CO/BO: SI Motion drive-integrated status signals
- r9772 0...23 CO/BO: SI status (Control Unit)
- r9773 0...23 CO/BO: SI status (Control Unit + Motor Module)
- r9774 CO/BO: SI status (group STO)
- r9780 SI monitoring cycle (Control Unit)
- r9774.0...23 CO/BO: SI status (group STO)

HLA parameters:

- p9625[0...1] SI HLA shutoff valve wait time (CU)
- p9626 SI HLA shutoff valve fedback contact configuration (CU)

SS2, SS2e: (Safe Stop 2, Safe Stop 2 external)

- p9501 SI Motion enable safety functions (Control Unit)
- p1135[0...n] OFF3 ramp-down time
- p9530 SI Motion standstill tolerance (Control Unit)
- p9548 SI Motion SAM actual speed tolerance (Control Unit)
- p9552 SI Motion transition time STOP C to SOS (SBH) (Control Unit) (STOP C corresponds to SS2)
- r9722 0...15 CO/BO: SI Motion drive-integrated status signals

SOS: (Safe Operating Stop)

- p9501 SI Motion enable safety functions (Control Unit)
- p9530 SI Motion standstill tolerance (Control Unit)
- p9551 SI Motion SLS(SG) switchover SOS (SBH) delay time (CU)
- r9722 0...31 CO/BO: SI Motion drive-integrated status signals
- r9731 SI Motion safe position accuracy

SBC: (Safe Brake Control)

- p0799 CU inputs/outputs sampling time
- p1215 Motor holding brake configuration
- p7015 Parallel circuit holding brake, power unit data set
- p9602 SI enable Safe Brake Control (Control Unit)
- p9621 BI: SI Safe Brake Adapter signal source (Control Unit)
- p9622[0...1] SI SBA relay wait times (Control Unit)
- r9771.14 SI Common functions (Control Unit): SBC supported for parallel circuit
- r9780 SI monitoring cycle (Control Unit)



Notes:

SBT: (Safe Brake Test)

- p1215 Motor holding brake configuration
- p1216 Motor holding brake opening time
- p1217 Motor holding brake closing time
- p9501 SI Motion enable safety functions (Control Unit)
- p9601 SI enable drive-integrated functions (Control Unit)
- p9602 SI enable Safe Brake Control (Control Unit)
- p10201 SI Motion SBT enable
- p10202[0...1] SI Motion SBT brake selection
- p10203 SI Motion SBT control selection
- p10204 SI Motion SBT motor type
- p10208[0...1] SI Motion SBT test torque ramp time
- p10209[0...1] SI Motion SBT brake holding torque
- p10210[0...1] SI Motion SBT test torque factor sequence 1
- p10211[0...1] SI Motion SBT test duration sequence 1
- p10212[0...1] SI Motion SBT position tolerance sequence 1
- p10218 SI Motion SBT test torque sign
- p10220[0...1] SI Motion SBT test torque factor sequence 2
- p10221[0...1] SI Motion SBT test duration sequence 2
- p10222[0...1] SI Motion SBT position tolerance sequence 2
- p10230[0...5] BI: SI Motion SBT control word
- r10231 SI Motion SBT control word diagnostics
- r10234.0...15 CO/BO: SI Safety Info Channel status word S_ZSW3B
- p10235 CI: Safety Control Channel control word S_STW3B
- r10240 SI Motion SBT test torque diagnostics
- r10241 SI Motion SBT load torque diagnostics
- p60122 IF1 PROFIdrive SIC/SCC telegram selection

SBR: (Safe Brake Ramp)

- p9560 SI Motion STO shutdown speed (CU)
- p9581 SI Motion brake ramp reference value (CU)
- p9582 SI Motion brake ramp delay time (CU)
- p9583 SI Motion brake ramp monitoring time (CU)

SAM: (Safe Acceleration Monitor)

- p9546 SI Motion SSM (SGA n < nx) speed limit (CU)
- p9548 SI Motion SAM actual speed tolerance (Control Unit)
- p9568 SI Motion SAM speed limit (Control Unit)

SSM: (Safe Speed Monitor)

- p9501 SI Motion enable safety functions (Control Unit)
- p9501.16 Enable SSM (n < nx) hysteresis and filtering (SSM with hysteresis)
- p9506 SI Motion function specification (Control Unit)
- p9509 SI Motion behavior during pulse suppression (Control Unit)
- p9509.1 SSM during pulse suppression and encoderless (feedback SSM active during pulse suppression
- p9545 SI Motion SSM (SGA n < nx) (Control Unit) (filter time)
- p9546 SI Motion SSM (SGA n < nx) speed limit (CU)
- p9547 SI Motion SSM (SGA n < nx) speed hysteresis (CU)
- r9722.0...14 CO/BO: SI Motion drive-integratd status signals
- r9722.15 SSM (speed below limit value) (speed below SSM limit value)

Notes:

SLS: (Safely-Limited Speed)

- p5901.0 SI Motion enable safety functions (Control Unit)
- p9512 Select SI Motion safety functions without selection (CU)
- p9531[0...3] SI Motion SLS (SG) limit values (Control Unit)
- p9533 SI Motion SLS speed setpoint limitation override factor (Control Unit)
- p9551 SI Motion SLS(SG) switchover SOS (SBH) delay time (CU)
- p9563[0...3] SI Motion SLS(SG)-specific stop response (Control Unit)
- p9580 SI Motion STO delay bus failure (Control Unit)
- p9581 SI Motion brake ramp reference value (Control Unit)
- p9582 SI Motion brake ramp delay time (Control Unit)
- p9583 SI Motion brake ramp monitoring time (Control Unit)
- p9601 SI enable drive-integrated functions (Control Unit)
- r9707[0...2] CO: SI Motion diagnostics encoder actual position value GX_XIST1
- r9714[0...2] CO: SI Motion diagnostics speed
- r9720.0...27 CO/BO: SI Motion drive-integrated control signals
- r9721.0...15 CO/BO: SI Motion status signals (Control Unit)
- r9722.0...31 CO/BO: SI Motion drive-integrated status signals (Control Unit)

SCA: (Safe Cam)

- p5901 SI Motion enable safety functions (Control Unit)
- p9503 SI Motion SCA (SN) enable (Control Unit)
- p9505 SI Motion SP modulo value (Control Unit)
- p9536[0...29] SI Motion SCA (SN) plus-cam position (Control Unit)
- p9537[0...29] SI Motion SCA (SN) minus-cam position (Control Unit)
- P9540 SI Motion SCA (SN) tolerance (Control Unit)
- P9542 SI Motion actual value comparison tolerance (cross-check) (Control Unit)
- r9703.0...31 CO/BO: SI Motion SCA status signal (Control Unit)
- r9708[0...5] SI Motion diagnostics safe position
- 9720.23 CO/BO: SI Motion drive-integrated control signals: Deselect SCA
- 9727 SI Motion User agreement drive internally
- 9771.22 SI common functions: SCA supported

SLP: (Safely-Limited Position)

- P9501 SI Motion enable safety functions (Control Unit CU)
- p9534[0...1] SI Motion SLP (SE) upper limit value CU
  p9535[0...1] SI Motion SLP (SE) lower limit value CU
- P9544 SI Motion actual value comparison tolerance (referencing) (CU)
- p9562[0...1] SI Motion SLP (SE) stop response CU
- P10009 SI Motion SLP retraction F-DI (processor 1)
- p10032[0...3] SI Motion SLP select input terminal (processor 1)
- P10033 [0...3] SI Motion SLP position range input terminal (processor 1)
- p10039[0...3] SI Motion Safe State signal selection (processor 1)
- P10109 SI Motion SLP retraction F-DI (processor 2)
- p10132[0...3] SI Motion SLP select input terminal (processor 2)
- p10133[0...3] SI Motion SLP select input terminal (processor 2)
- P10139 SI Motion Safe State signal selection (processor 2)

Notes:

SP: (Safe Position)

- p9501 SI Motion enable safety functions (Control Unit)
- p9505 SI Motion SP modulo value (Control Unit)
- p9542 SI Motion actual value comparison tolerance (cross-check) (Control Unit)
- p9601 SI enable drive-integrated functions (Control Unit)
- r9708[0...5] SI Motion diagnostics safe position
- r9713[0...5] CO: SI Motion diagnostics actual position value load-side

SDI: (Safe Direction)

- p1820[0...n] Reverse output phase sequence
- p1821[0...n] Direction of rotation
- p9501.17 SI Motion enable safety functions (Control Unit): Enable SDI
- P9506 SI Motion function specification (Control Unit)
- P9509 SI Motion behavior during pulse suppression (Control Unit)
- P9564 SI Motion SDI tolerance (Control Unit)
- P9565 SI Motion SDI delay time (Control Unit)
- P9580 SI Motion pulse suppression delay bus failure (Control Unit)
- r9720.0...27 CO/BO: SI Motion drive-integrated control signals
- r9722.0...31 CO/BO: SI Motion drive-integrated status signals (Control Unit)
- 9733[0...2] CO: SI Motion speed setpoint - limitation active
- P10017 SI Motion digital inputs debounce time (processor 1)
- p10030[0...3] SI Motion SDI positive input terminal (processor 1)
- p10031[0...3] SI Motion SDI negative input terminal (processor 1)
- p10039[0...3] SI Motion Safe State signal selection (processor 1)
- p10042[0...5] SI Motion F-DO signal sources (processor 1)
- p10043[0...5] SI F-DO 1 signal sources
- p10044[0...5] SI F-DO 2 signal sources
- p10045[0...5] SI F-DO 3 signal sources

Notes:

Safe Referencing:

- p9501.27 Enable referencing via SCC
- p9572 SI Motion referecing position (Control Unit)
- p9573 SI Motion accept referencing position (Control Unit)
- r9708[0...5] SI Motion diagnostics safe position
- 9713[0...5] CO: SI Motion diagnostics actual position value load side
- 9722.0...31 CO/BO: SI Motion drive-integrated status signals (Control Unit)
- r9723.0...17 CO/BO: SI Motion drive-integrated diagnostics signals
- p9726 SI Motion User Agreement selection/deselection
- p9740 SI Motion User Agreement selection/deselection MM

Safe actual value sensing:

- p9501.3 SI Motion enable safety functions (Control Unit), enable actual value synchronization
- p9502 SI Motion axis type (Control Unit)
- p9511 SI Motion actual value sensing cycle (Control Unit)
- p9515 SI Motion encoder coarse position value configuration (Control Unit)
- p9516 SI Motion encoder configuration safety functions (Control Unit)
- p9517 SI Motion linear encoder lattice pitch (Control Unit)
- p9518 SI Motion encoder pulses per revolution (Control Unit)
- p9519 SI Motion fine resolution G1_XIST1 (Control Unit)
- p9520 SI Motion leadscrew pitch (Control Unit)
- p9521[0...7] SI Motion gearbox encoder (motor)/load denominator (Control Unit)
- p9522[0...7] SI Motion gearbox encoder (motor)/load numerator (Control Unit)

- p9523 SI Motion redundant coarse position value valid bits (Control Unit)
- p9524 SI Motion redundant coarse position value fine resolution (Control Unit)
- p9525 SI Motion redundant coarse position value relevant bits (Control Unit)
- p9526 SI Motion encoder assignment second channel
- p9542 SI Motion actual value comparison tolerance (cross-check) (Control Unit)
- p9549 SI Motion slip velocity tolerance (Control Unit)
- p9585 SI Motion actual value sensing sensorless fault tolerance (CU)
- p9586 SI Motion actual value sensing sensorless delay time (CU)
- p9587 SI Motion actual value sensing sensorless filter time (CU)
- p9588 SI Motion actual value sensing sensorless minimum current (CU)
- p9589 SI Motion actual value sensing sensorless acceleration limit (CU)
- p9700 SI Motion copy function
- p9724 SI Motion cross-check comparison cycle
- p9730 SI Motion safe maximum velocity
- p9731 SI Motion safe position accuracy
- r9713[0...5] SI Motion diagnostics actual position value load side
- r9714[0...2] SI Motion diagnostics velocity
- r9732[0...1] SI Motion velocity resolution

Notes:

Manufacturer documentation and function table:

Manufacturer documentation:

The machine manufacturer is obligated to provide documentation with regard to the machine requirements for functional safety.

This documentation must contain the following descriptions

Plant description

1.1 **Machine description**

1.1.1 **Overvew of the machine**

1.2 **Specifications of the control**

1.3 **Configuration plan**

Description of the safety functions

1.4 **Function table**

1.5 **Safety functions per axis**

1.6 **Specifications of the safety devices**

1.7 **Circuit diagrams for all safety-relevant devices**

1.8 **Safety summary of the F-PLC**

An example of this documentation is created for the TAC training control in the following.

A PLC project of the **TIA Portal Integration** is used as the basis and
supplemented in the following sections:
The hardware of the training control is expanded by:

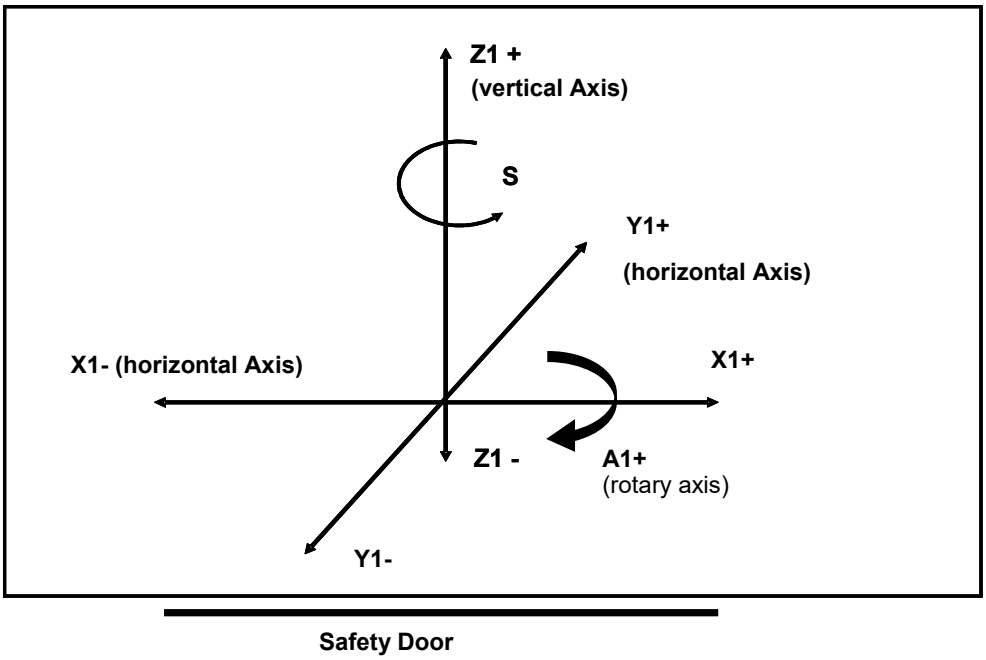
- SINAMICS drive rack with five drives
- Safety I/O rack with ET200SP

Plant description:

1.1 ***Machine description:***

Designation	TAC SINUMERIK Safety Integrated F_PLC
Type	Machining center
Serial no.	A019
Electronic axes	4
Other axes	None
Spindles	1

1.1.1 ***Overview of the machine:***



Notes:

1.2 *Specifications of the control:*

Type and SW version	SINUMERIK ONE SW 31-3 Export V06.23			
NC type	NCU 1760			
PLC type	S7 1500-F			
Operator panel	TOP22			
SW version	NCK version	Drive version	PLC version	GP version
	145.01.00	05.20.73.49	06.23.	06.23.
Values from the relevant machine data	Basic system clock cycle		IPO cycle	Monitoring cycle
	0.0020 s		0.0080 s	0.012 s

1.3 *Configuration plan:*

Channel	Axis/ spindle	Name	Drive control		Module no.	Measuring systems / motor holding brake
			Control	Module type		
1	Horizontal	**X1**	SINAMICS	Double Motor Module	2	Incremental motor measuring system without brake
1	Horizontal	**Y1**	SINAMICS	Double Motor Module	2	Incremental motor measuring system without brake
1	Vertical	**Z1**	SINAMICS	Double Motor Module	3	Incremental motor measuring system with brake
1	Spindle	**SP1**	SINAMICS	Single Motor Module	1	Incremental motor measuring system 2nd incremental motor measuring system without brake
1	Rotary axis	**A1**	SINAMICS	Double Motor Module	3	Absolute motor measuring

Notes:

Description of the safety functions:

1.4 *Function table:*

The function table must be created before the configuration/programming of the machine. It describes the active monitoring functions depending on the operating mode, the protective door, the Hold to Run enabling button, the emergency stop button and the "End Limit Override" keyswitch and can be inserted later in the Safety Integrated acceptance test.

Machine-control drives	**Off**		**Axis/spindle X1, Y1, Z1, A1, SP1**	**Monitoring STO**
Machine-control drives	**On**			
Operating mode	**Protective door with locking Enabling button**	**End Limit Override Switchover SLP1 / SLP2 (SLP and SDI: Not for axis A1 and spindle SP1)**	**Axis/spindle**	**Monitoring**
Production (AUTO) • **AUTO** • **(MDI)**	Closed and locked	Active / not active	X1, Y1, Z1, A1, SP1	SLS4 active SLP1 active
	Unlocked	Active / not active	X1, Y1, Z1, A1	SOS active SLP1 active
			SP1	STO active
Mode 3 • **AUTO** • **MDI** **(Program start for open protective door with enabling button and cycle start button)**	Closed and locked	Active / not active	X1, Y1, Z1, A1, SP1	SLS4 active SLP1 active
	Unlocked	Active / not active	X1, Y1, Z1, A1	SOS active SLP1 active
			SP1	STO active
	Unlocked and enabling button actuated	Active / not active	X1, Y1, Z1, A1, SP1	SLS3 active SLP1 active
Setup • **JOG** • **INC**	Closed and locked	Active (retract in opposite direction from SLP1/2 with SDI)	X1, Y1, Z1, A1, SP1	SLS2 active SDI active
	Unlocked	Active / not active	X1, Y1, Z1, A1	SOS active SLP2 active
			SP1	STO active
	Unlocked and enabling button actuated	Not active	X1, Y1, Z1, A1, SP1	SLS1 active SLP2 active
		Active (retract in opposite direction from SLP1/2 with SDI)	X1, Y1, Z1, A1, SP1	SLS1 active SDI active
Operating mode-independent	**Emergency stop actuated:** -ESTOP1 / Axes instance -ESTOP1 / Spindle instance	**Emergency stop (Operating mode: Setup)**	X1, Y1, Z1, A1, *SP1 shutdown of ext. devices	SS1 > STO
		Emergency stop (Operating mode: Production / Mode 3)		SS2e / *SS2 > SOS
		Emergency stop delayed		STO

Notes:

Operating mode concept in accordance with DIN EN ISO 16089 or ISO 16090:

Some of the safe operating modes described in the following table are used for Powertrain projects.

In Section 1.6 **Safety equipment**
you will find the Safety Integrated test rack used in this documentation
represented with the **operating mode keyswitch** (Production / Setup / Mode 3).

	Requirements from European Standards, e.g. DIN EN ISO 16089 or ISO 16090
Mode 1: • **Setup** • **Production**	The machine runs only for **active or closed protective equipment**.
Mode 2: • **Setup (JOG, INC)**	The machine runs for **revoked protective effect** of a locked, movable, separating protective equipment item (e.g. operator door), however, only for the following prerequisites: • Individual **axis movements reduced to 2 m/min** or in steps of maximum 10 mm - **With Jogging switch** or - **Electronic handwheel** or - Manual data input, incl. enabling • Spindle speed limited by the braking power (**brake within two revolutions**) • Machine-integrated **chip conveyor with Enabling key and Jogging switch** • **Upper limits for speeds and feeds** must be monitored. A controlled stop (Cat. 1) must be triggered when these limits are exceeded. • If several command equipment items (operator panels or hand-held terminals) are present, only one device may be functional.
Mode 3: • **MDA single step**	**Manual intervention under restricted operating conditions** The machine runs for **revoked protective effect** of a locked, movable, separating protective equipment item (e.g. operator door), however, only for the following prerequisites: • Limitation of single or multiple axes to **5 m/min** • Spindle speed limited by the braking power (**brake within five revolutions**) (Powertrain: **Spindle speed limited to a minimum value necessary for reliable stock removal** (additional technical protective measures may be required for the monitoring, such as divisible door). • **Program start and spindle start with Enabling device + Cycle start key** • **Upper limits for speeds and feeds** must be monitored. If not in Cat. 3, then a combination of this function **with enabling equipment**. A controlled stop must be triggered when these limits are exceeded. (Powertrain: **Axes 6 m/min**; robots 250 mm/s) • If for ergonomic reasons when using operating mode 3, the deployment of an enabling device is not practicable (e.g. because the duration of a necessary process monitoring and/or of an intervention exceeds an acceptable fatigue time by the machine operator for actuating the Enabling switch or setting multiple parameters on the control device prevents the continuing activation of the enabling equipment), a combination of other technical control measures, other than the enabling equipment, must be used to reduce dangers caused by being crushed or drawn in.
Mode 4: • **MDA single step**	**Operating mode without continuous activation of an enabling device** For applications for which the settings to be performed or for ergonomic reasons preclude the continuous activation of an enabling device.
Mode 5: • **AUTO**	In accordance with MRL 2006/42/EU Annex 1 No. 1.2.5, **automatic operation with opened protective door when technically required is generally permitted.** Dangers from chained commands must be precluded.

Notes:

1.5 ***Safety functions per axis:***

Axis	SI function	Limit value	Active when
X1 Y1 Z1 A1	**SS1** Safe Stop1		• Emergency stop is actuated • Machine / control / drives are switched off
	STO Safe Torque Off		• After the "ESTOP axis delay time" for emergency stop has expired • Machine / control / drives off
	SBC (only Z1) Safe Brake Control		• STO active
	SS2 Safe Stop2		• Not active
	SS2e Safe Stop2 external		• Selection of Emergency stop in "Production" or "Mode 3" with transition to SOS; After the "ESTOP axis delay time" STO is active.
	SOS Safe Operating Stop		• Protective door unlocked and not (protective door open and enabling button actuated) for Setup mode or Mode 3
	SAM Safe Acceleration Monitor		• SS1 active and SAM shutdown threshold p9568 > 0 • SS2 active and SAM shutdown threshold p9568 > 0
	SSM Safe Speed Monitor		• Monitoring velocity p9546 > 0 → Always
	SLS1 Safety-Limited-Speed 1	2 m/min	• Protective door open and Setup mode and enabling button actuated
	SLS2 Safely-Limited-Speed 2	7 m/min	• Protective door closed and locked and Setup mode
	SLS3 Safely-Limited-Speed 3	5 m/min	• Protective door open and Mode 3 and enabling button actuated
	SLS4 Safely-Limited-Speed 4	10 m/min	• Protective door closed and locked and (Production mode or Mode 3)
	SLS1 override	0...100% of SLS1	• Not active
	SLP1+ Safely-Limited Position 1 plus	1010 mm Not axis A1	• "Production" mode or "Mode 3" and "End Limit Override" not active
	SLP1- Safely-Limited Position 1 minus	-10 mm Not axis A1	• "Production" mode or "Mode 3" and "End Limit Override" not active
	SLP2+ Safely-Limited Position 2 plus	1020 mm Not axis A1	• "Setup" mode and "End Limit Override" not active and protective door unlocked and enabling button actuated
	SLP2- Safely-Limited Position 2 minus	-20 mm Not axis A1	• "Setup" mode and "End Limit Override" not active and protective door unlocked and enabling button actuated
	SBT (only Z1) Safe Brake Test		• PLC block "DBSI_SBT_X_Axis" is activated
	SP Safe Position	Not axis A1	
	SDI Safe Direction	Not axis A1	"End Limit Override" keyswitch active and position > SLP1/2+ → SDI- and position < SLP1/2- → SDI+
	SCA Safe Cam	1:0..500mm 2:500..1000mm	• Cam position approached: Only axis X1

Spindle	SI function	Limit value	Active when
SP1	**SS1** Safe Stop1		• Emergency stop is actuated • Machine / control / drives are switched off
	STO Safe Torque Off		• After the "ESTOP spindle delay time" for emergency stop has expired • Machine / control / drives off
	SS2 Safe Stop2		• Selection of Emergency stop in "Production" or "Mode 3" with transition to SOS; After the "ESTOP axis delay time" STO is active.
	SS2e Safe Stop2 external		• Not active
	SOS Safe Operating Stop		• Is not used for the spindle to in order to turn the spindle manually when the door is unlocked/open (manual tools)
	SAM Safe Acceleration Monitor		• SS1 active and SAM shutdown threshold p9568 > 0 • SS2 active and SAM shutdown threshold p9568 > 0
	SSM Safe Speed Monitor		• Monitoring velocity p9546 > 0 → Always
	SLS1 Safety-Limited-Speed 1	50 rpm (braking within two revolutions must be ensured)	• Protective door open and Setup mode and enabling button actuated
	SLS2 Safety-Limited-Speed 2	500 rpm	• Protective door closed and locked and Setup mode
	SLS3 Safety-Limited-Speed 3	125 rpm (braking within five revolutions must be ensured)	• Protective door open and Mode 3 and enabling button actuated
	SLS4 Safety-Limited-Speed 4	3000 rpm	• Protective door closed and locked and (Production mode or Mode 3)
	SLS1 override	0...100% of SLS1	• SLS1 active
	SP Safe Position		• Not active
	SDI Safe Direction		• Not active
	SCA Safe Cam		• Not active

Notes:

1.6 *Specifications regarding safety equipment:*

1.6.1 *Emergency stop: (MCP push-button)*

With the dual-channel emergency stop signal, the F-PLC activates a **standstill of the drives** X1, Y1, Z1, A1, SP1 in accordance with EN 60204-1: 2006 of

- stop category 1 (**SS1** → **STO**) in “Setup” active.
- stop category 2 (**SS2 / SS2e** → **SOS**) in “Production” and “Mode 3” active.

After a **delay time SS1** → **STO in Setup** respectively

- **ESTOP1-Delay-Time-Axis** in **Production / Mode 3** an **STO** is triggered for axes X1, Y1, Z1, A1
- **ESTOP1-Delay-Time-Spindle** in **Production / Mode 3** an **STO** is triggered for spindle SP1

The **hydraulic system, refrigerant pump, tool magazine** and the **chip conveyor** are shutdown safely using **safety-oriented shutdown of a standard DO module** (DQ 8x24 VDC / 0.5 A HF) **by the F-Power Module** (F-PM-E 24 VDC / 8 A PPM) of the ET200SP in the event of an emergency stop. This safety-oriented shutdown achieves SIL 2 / PL d.

1.6.2 *Emergency Stop Reset: (SI test rack)*

The dual-channel key is used for:

- Emergency stop acknowledgment
- Acknowledgment/reintegration for F-I/O faults
- Acknowledgment of the Safety Integrated monitoring functions in accordance with the function table
- Connection/disconnection of the power (machine/control/drives: On/Off)

Meaning of the light signal:

- Off: Power disconnected
- On: Power connected
- Flashing: Emergency stop acknowledgment or monitoring functions required and possible

1.6.3 *Unlock protective door: (SI test rack Request key)*

The protective door is unlocked by a single-channel Request key, and if the associated safety-relevant requirements are satisfied (**SSM** of the X1, Y1, Z1, A1 and SP1 drives; no test stop required).

1.6.4 *Protective door switch: (SI test rack)*

The protective door is equipped with a protective door switch with tumbler.
The protective door switch provides the dual-channel F-PLC input signals:

- **Protective door closed**
- **Protective door locked**

The tumbler is **unlocked** with a single-channel PLC output signal. The switching and selection of the safety functions depending on the protective door switch are described in the function table in Section 2.1.

1.6.5 *Mode selector switch: (SI test rack keyswitch)*

The following operating modes are selected with the Mode selector switch:

- **Setup**
- **Production**
- **Mode 3**

The keyswitch has the two dual-channel **Setup and Mode 3 contact levels**. The switching and selection of the safety functions depending on the protective door switch are described in the function table in Section 2.1.

1.6.6 *Hold to Run Switch: (Safety Integrated Testrack key button)*

For an opened protective door, the dual-channel Permission switch,

- are deselected in the SOS "Setup" operating mode for X1, Y1, Z1, A1 as well as STO of the SP1 and SLS1 selected
- are deselected in the SOS "Mode 3" operating mode for X1, Y1, Z1, A1 as well as STO of the SP1 and SLS3 selected

1.6.7 *Monitoring of the max. axis speeds or spindle speed: (SLS1...SLS4)*

To protect the machine, the maximum permitted feed speed of the axes or the maximum permitted speed of the spindles is monitored for closed protective equipment. To do this, SLS4 is active for closed protective equipment (no trace for documentation necessary/possible, because the limit values for the SLS4 are higher than the maximum axis speeds).

1.6.8 ***Safe software limit switches: (SLP1...SLP2)***

To limit the operating range, the X1, Y1 and Z1 linear axes are equipped with SLP1 and SLP2 safe software limit switches with the following functionality:

- The SLP1 function is always active in the "Production" and "Mode3" operating modes, provided the "End Limit Override" keyswitch is NOT actuated.
- The SLP2 function is always active in the "Setup" operating mode, provided the "End Limit Override" key-operated switch is NOT actuated.
- The software limit switches are safe only when the "User agreement" is also set.

1.6.9 ***Z1 axis holding brake: (SBC and SBT)***

The holding brake is connected via a Sinamics Motor Module and switched via **STO active**.
The **SBT safety function** is deployed to monitor the correct operation of the holding brake for the Z1 axis.
The SBT safety function is performed with the "Unlock protective door" request signal. The protective door is unlocked only after a brake test has been performed successfully. If the SBT detects a fault, the axis must be traversed to a safe position so that the protective door can be unlocked.

1.6.10 ***Output for K1 and K2 actuators***:

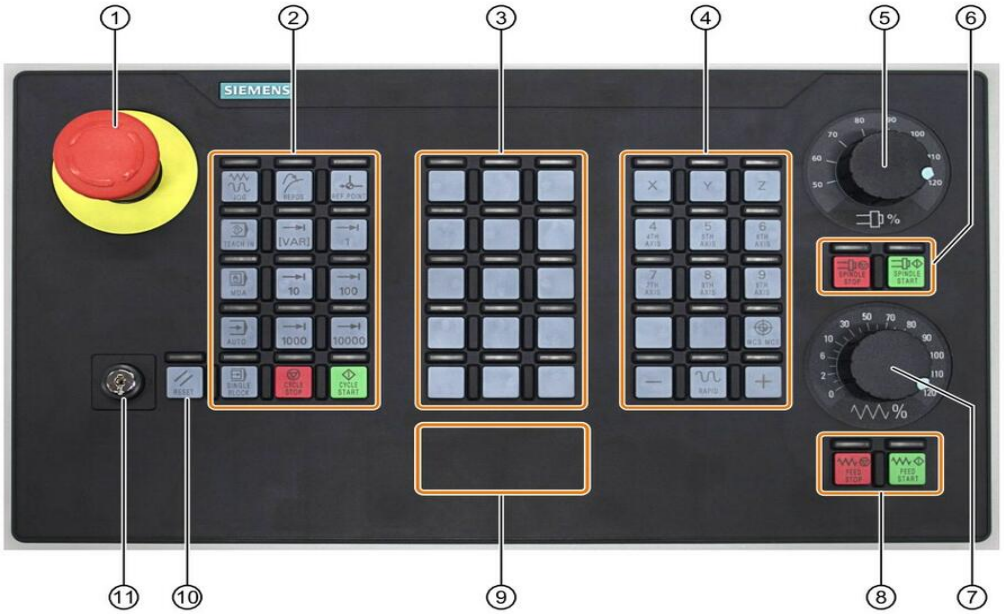
K1 and K2 are relays, each controlled from a safe output signal.
They are energized as long as the associated F-DO is activated. A de-energized closed (NC contact) feedback circuit is connected to I32.6. (On a real machine, the correct operation of this actuator must be monitored with the PLC test stop.)

1.6.11 ***Test stop***

The test stop is performed generally after the complete switch on/off of the control, as well as after the time interval has elapsed (p9559 forced dormant error detection timer).
User key T1 is used to start the test stop routine

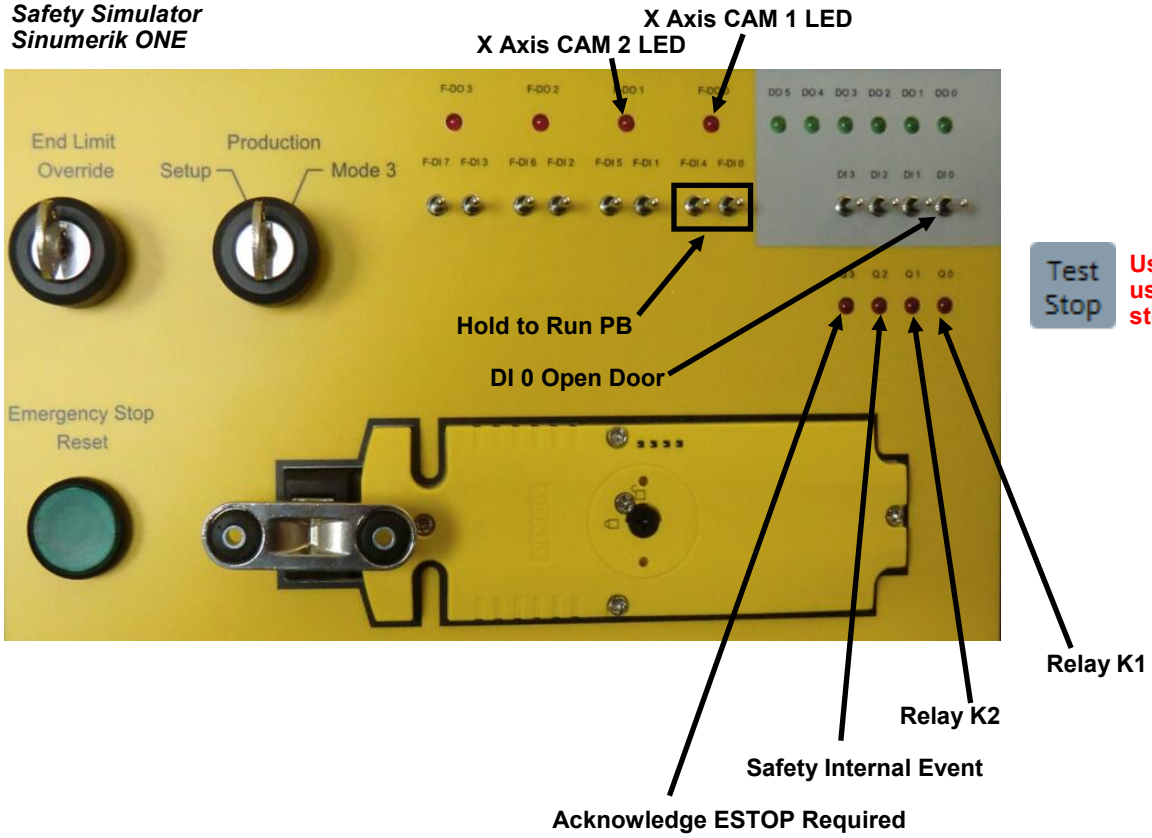
To illustrate the safety equipment used in this training documentation, the MCP398C as well as the ***Safety Integrated test rack*** on the following page are described below.

(1) **Emergency stop butto**n
(2) Operating modes, machine functions
(3) User keys T1 to T15
(4) Direction keys with rapid traverse override (R1 to R15)
(5) Override switch for the spindle control
(6) Spindle control
(7) Override switch for the feed control
(8) Feed control
(9) Slots for two control devices (d = 16 mm or 22 mm)
(10) Reset key
(11) Keyswitch (four positions)



Notes:

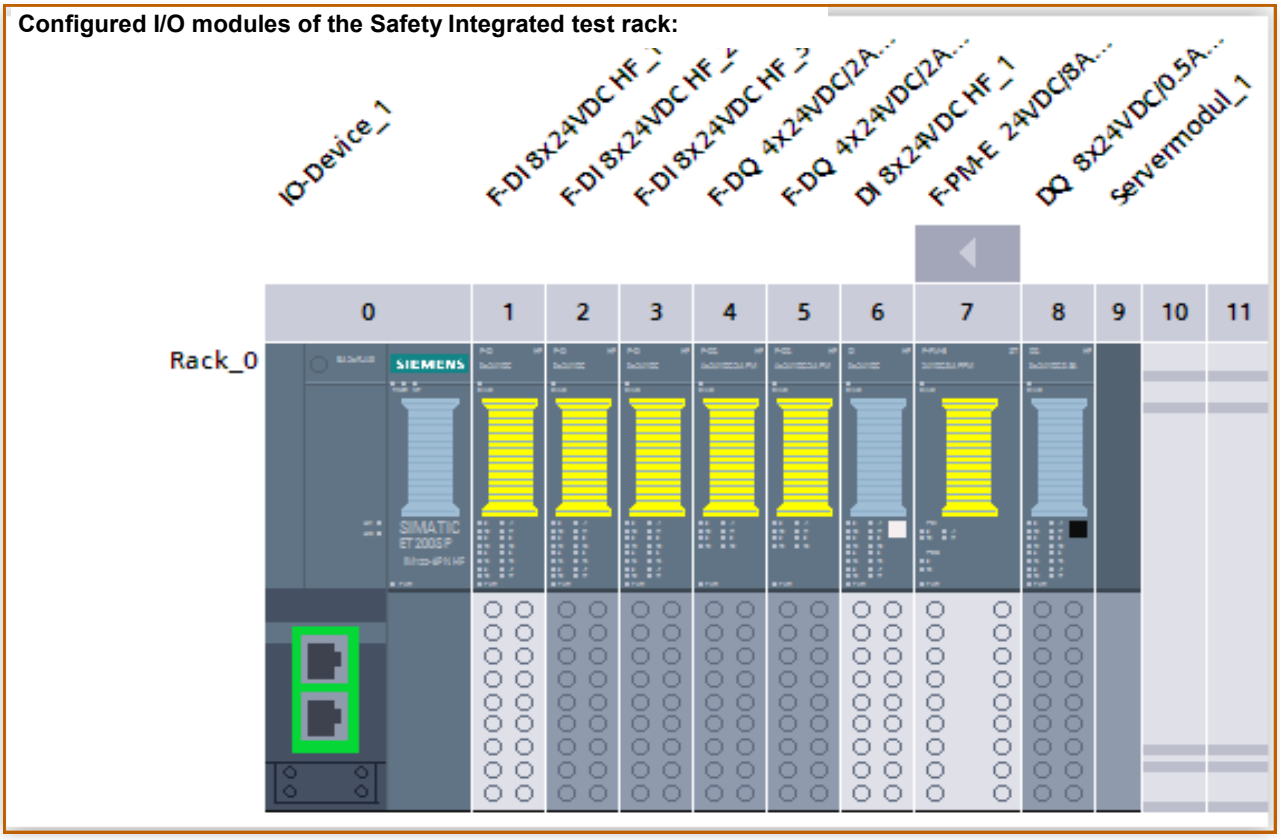
Safety Simulator
Sinumerik ONE



Test Stop

User key Test Stop is used to start the test stop routine

Configured I/O modules of the Safety Integrated test rack:



Notes:

The **I/O addresses** and the **Article numbers** of the configured modules in the ***Device overview***, as well as the assignment of the addresses to the configured **Variable name** in the ***"Safety" PLC variable table*** must also be used.

Device overview

Module	...	Rack	Slot	I address	Q address	Type	Article number
		0	300				
IO device_1		0	0			IM 155-6 PN HF	6ES7 155-6AU00-0CN0
PROFINET interface		0	0 X1			PROFINET interface	
F-DI 8x24VDC HF_1		0	1	512...517	512...515	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0
F-DI 8x24VDC HF_2		0	2	518...523	518...521	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0
F-DI 8x24VDC HF_3		0	3	524...529	524...527	F-DI 8x24VDC HF	6ES7 136-6BA00-0CA0
F-DQ 4x24VDC/2A PM HF_1		0	4	530...534	530...534	F-DQ 4x24VDC/2A ...	6ES7 136-6DB00-0CA0
F-DQ 4x24VDC/2A PM HF_2		0	5	535...539	535...539	F-DQ 4x24VDC/2A ...	6ES7 136-6DB00-0CA0
DI 8x24VDC HF_1		0	6	100		DI 8x24VDC HF	6ES7 131-6BF00-0CA0
F-PM-E 24VDC/8A PPM ST_1		0	7	540...546	540...544	F-PM-E 24VDC/8A P...	6ES7 136-6PA00-0BC0
DQ 8x24VDC/0.5A HF_1		0	8		100	DQ 8x24VDC/0.5A ...	6ES7 132-6BF00-0CA0
Server module_1		0	9			Server module	6ES7 193-6PA00-0AA0

Safety

	Name	Data type	Address
1	SafeSpeedMonitorA	Bool	%I6765.7
2	AckIntEventA	Bool	%Q6764.7
3	AckIntEventSP	Bool	%Q6700.7
4	AckIntEventX	Bool	%Q6716.7
5	AckIntEventY	Bool	%Q6732.7
6	AckIntEventZ	Bool	%Q6748.7
7	DeselectSOSA	Bool	%Q6764.3
8	DeselectSS1A	Bool	%Q6764.1
9	DeselectSS2A	Bool	%Q6764.2
10	DeselectSTOA	Bool	%Q6764.0
11	DeselectSOSSP	Bool	%Q6700.3
12	DeselectSS1SP	Bool	%Q6700.1
13	DeselectSS2SP	Bool	%Q6700.2
14	DeselectSTOSP	Bool	%Q6700.0
15	DeselectSOSX	Bool	%Q6716.3
16	DeselectSS1X	Bool	%Q6716.1
17	DeselectSS2X	Bool	%Q6716.2
18	DeselectSTOX	Bool	%Q6716.0
19	DeselectSOSY	Bool	%Q6732.3
20	DeselectSS1Y	Bool	%Q6732.1
21	DeselectSS2Y	Bool	%Q6732.2
22	DeselectSTOY	Bool	%Q6732.0
23	DeselectSOSZ	Bool	%Q6748.3
24	DeselectSS1Z	Bool	%Q6748.1
25	DeselectSS2Z	Bool	%Q6748.2
26	DeselectSTOZ	Bool	%Q6748.0
27	HoldToRunButton	Bool	%I524.0
28	InternalEventActiveA	Bool	%I6764.7
29	InternalEventActiveSP	Bool	%I6700.7
30	InternalEventActiveX	Bool	%I6716.7
31	InternalEventActiveY	Bool	%I6732.7
32	InternalEventActiveZ	Bool	%I6748.7
33	SLSBit0A	Bool	%Q6765.1
34	SLSBit0SP	Bool	%Q6701.1
35	SLSBit0X	Bool	%Q6717.1
36	SLSBit0Y	Bool	%Q6733.1
37	SLSBit0Z	Bool	%Q6749.1
38	SLSBit1A	Bool	%Q6765.2

Safety

	Name	Data type	Address
39	SLSBit1SP	Bool	%Q6701.2
40	SLSBit1X	Bool	%Q6717.2
41	SLSBit1Y	Bool	%Q6733.2
42	SLSBit1Z	Bool	%Q6749.2
43	SafeSpeedMonitorSP	Bool	%I6701.7
44	SafeSpeedMonitorX	Bool	%I6717.7
45	SafeSpeedMonitorY	Bool	%I6733.7
46	SafeSpeedMonitorZ	Bool	%I6749.7
47	SOSActiveX	Bool	%I6716.3
48	SOSActiveA	Bool	%I6764.3
49	SOSActiveSP	Bool	%I6700.3
50	SOSActiveZ	Bool	%I6748.3
51	SOSActiveY	Bool	%I6732.3
52	SetupSwitch	Bool	%I518.2
53	EmergencyStopResetPB	Bool	%I518.1
54	Q PowerModule	Bool	%Q540.0
55	Mode3Switch	Bool	%I518.3
56	DoorLocked	Bool	%I512.2
57	DoorClosed	Bool	%I512.3
58	End Limit Override	Bool	%I518.0
59	Safe Cam1 DO 0 LED	Bool	%Q535.0
60	Safe Cam2 DO 0 LED	Bool	%Q535.1
61	Q1 Relay K2 Door Closed	Bool	%Q530.1
62	Q0 Relay K1 Door Closed and L...	Bool	%Q530.0
63	Internal Event Lamp	Bool	%Q530.2
64	EmergencyStopPB	Bool	%I512.0
65	SPDeselectSLS	Bool	%Q6700.4
66	SelectSLPRange2X	Bool	%Q6718.3
67	SLPUpperLimitX	Bool	%I6719.6
68	SLPLowerLimitX	Bool	%I6719.7
69	SDIPositiveX	Bool	%Q6717.4
70	SafelyReferencedX	Bool	%I6718.7
71	SDINegativeX	Bool	%Q6717.5
72	SafeCamsActivatedX	Bool	%Q6718.7
73	SafeCamValid	Bool	%I6723.6
74	SafeCam1X	Bool	%I6720.0
75	SafeCam2X	Bool	%I6720.1
76	ESTOPResetRequired	Bool	%Q530.3

The Tag tables for the F-PLC can be prepared as shown here, or the tag names can be entered individually as the PLC logic is created.

Safety Other

	Name	Data type	Address
1	DoorUnlockButton	Bool	%I100.0
2	K1Relaytest	Bool	%M201.2
3	K2RelayTest	Bool	%M201.3
4	DoorUnlockRequest	Bool	%Q100.7

1.7 ***Circuit diagrams for all safety-relevant devices:***

Protection door

HT2 permission

MCP Emergency Stop





Drive-integrated SI functions activation / telegrams:

SINUMERIK Safety Integrated F_PLC provides the following means of activating the Safety Integrated functions.
Some of them can be combined with each other for the same drive.

Activation type:	Basic Functions	Extended Functions
Activation without selection	No	SLS, SDI
Activation via terminals (on the Control Unit and On the Motor/Power Module)	Yes	No
Telegrams to activate process data via PROFIsafe and SIC/SCC (PROFIdrive)	Yes	Yes

Motion monitoring without selection:

As an alternative to activation via terminals and/or PROFIsafe, it is possible to parameterize some safety functions without selection. For this mode, these functions are selected permanently after the parameterization and a POWER ON.

See "Section: Further information > Drive-integrated SI functions activation / telegrams"

Activation via terminals on the NCU and the Motor Module / Power Module:

Characteristics:

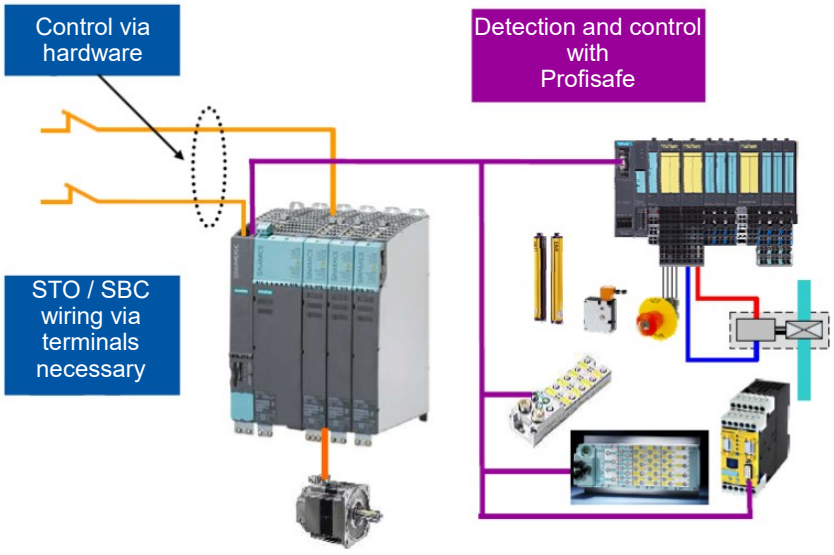
- Only for the Basic Functions
- Dual-channel structure via two digital inputs (e.g. NCU / power section)
- To prevent spurious tripping caused by signal faults or asymmetric test signals, the signals of the components (keys, switches, etc.) attached to the terminals can be debounced.
- These debouncing times can be specified in the following parameters:
  – p9651 SI STO/SBC/SS1 debouncing time (Control Unit) / SI STO t_Entpr CU
  – p9851 SI STO/SBC/SS1 debouncing time (Motor Module) / SI STO t_Entpr MM
- The EP terminals on the Motor Module are evaluated only when the Safety Integrated Basic Functions are enabled via onboard terminals.
- Terminals can be grouped for the concurrent triggering of SI functions for multiple drives.

Description of the dual-channel structure:

The functions are selected/deselected separately for each drive via two terminals:

- **NCU shutdown path:**
  The desired input terminal is selected via BI-CO interconnection.
- **Motor Module shutdown path:**
  The input terminal is the EP (Enable Pulses) terminal.

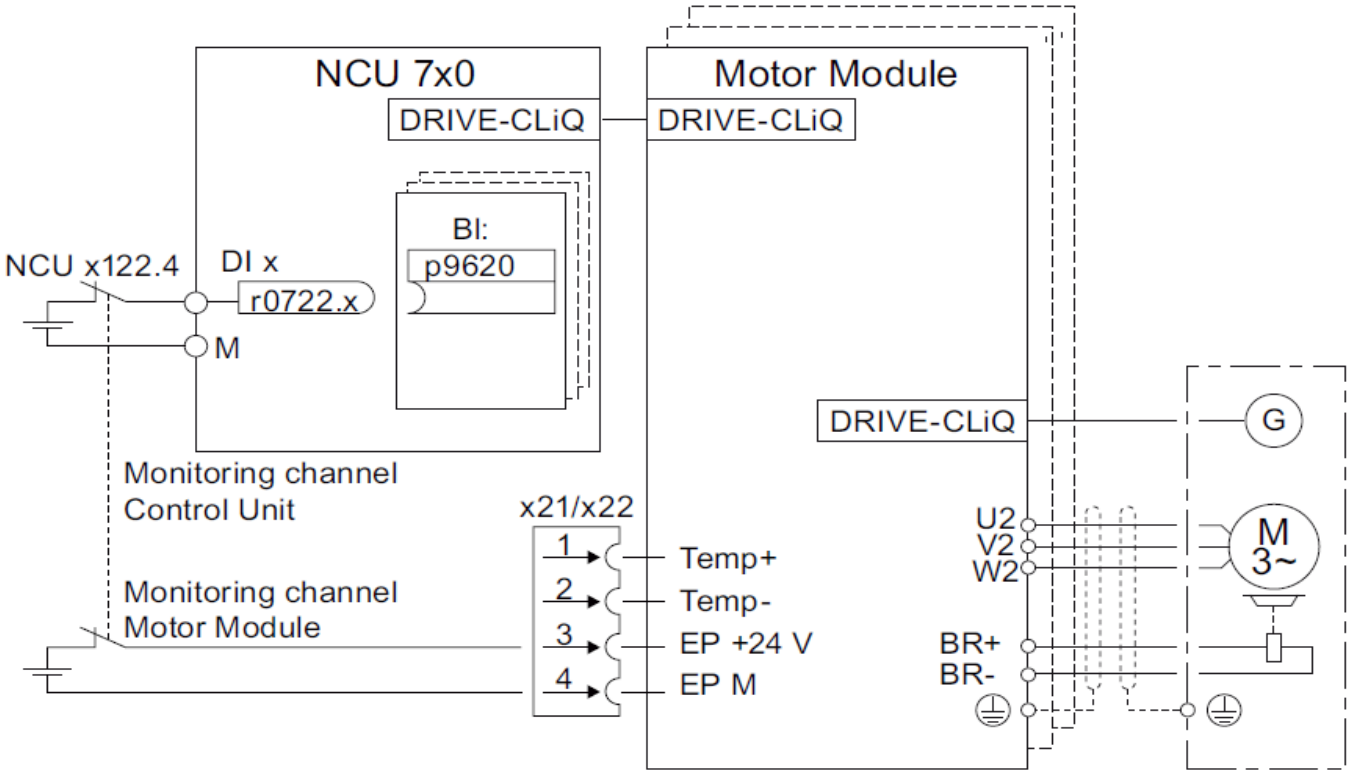
The activation via terminals requires an external hardware interface for controlling the Safety Functions.



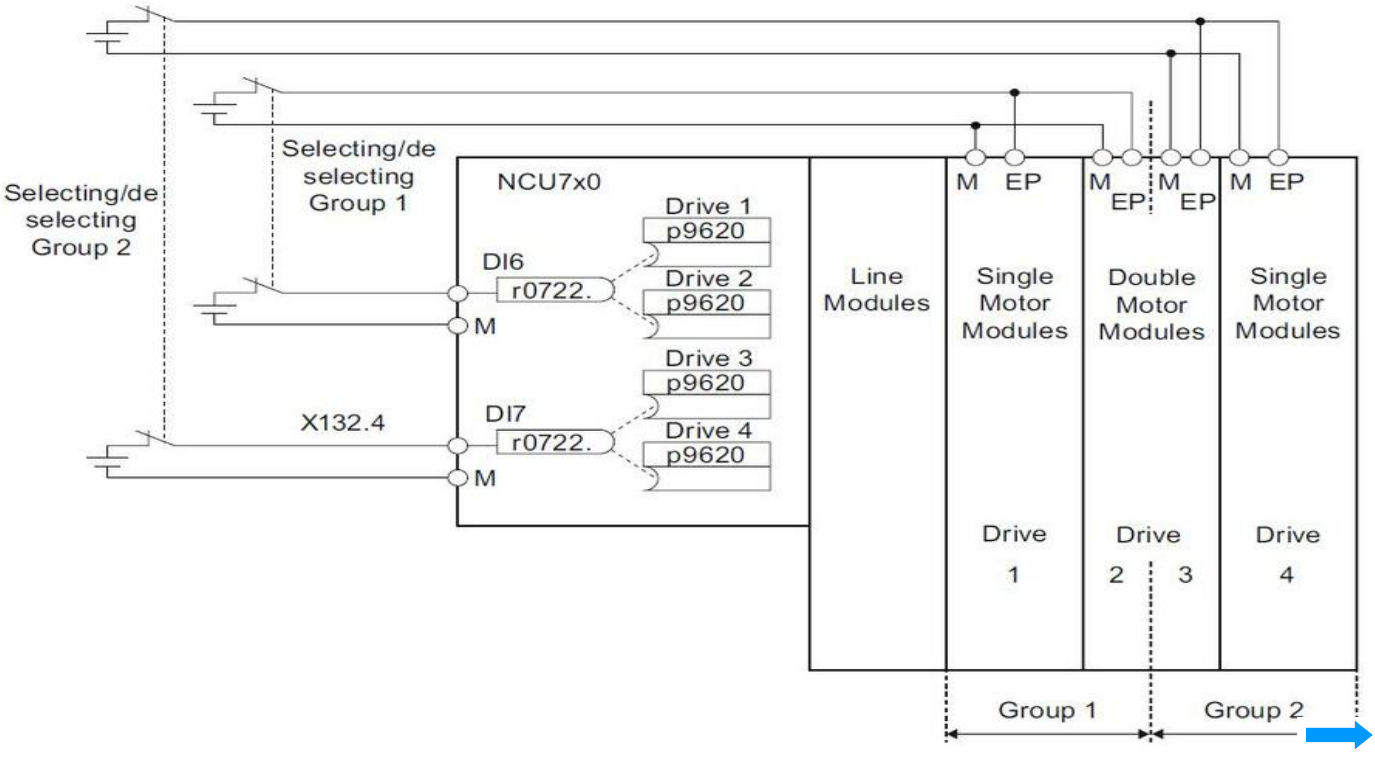
See "Section: Further information > Drive-integrated SI functions activation / telegrams"

Notes:

Example: Terminals for STO for an NCU7x0 and a Motor Module Booksize:



With the aid of a second NCU input, e.g. X122.3, a second group of axes/spindles can be activated by means of terminals different to the first group.



Notes:

Activation via PROFIsafe:

Activation via PROFIsafe is available for the Safety Integrated Basic Functions as well as for the Safety Integrated Extended Functions.
Control via PROFIsafe is also used for SINUMERIK Safety Integrated plus, whereby the advantages of the integrated F-PLC together with the communication via PROFIBUS and PROFINET can be fully utilized.
The activation of this variant is also part of this course documentation and is described in detail in the following.
Further information on the supported documents:
See "Section: Further information > Activation of drive-integrated SI Functions / telegrams "

Overview of telegrams and process data:

The following table shows the ***Telegrams for the activation via PROFIsafe and PROFIdrive SIC/SCC:***
(Safety Info Channel / Safety Control Channel).

- **The default setting for internal NC drives with firmware V4.8 is SIEMENS telegram 903**
- **The other telegrams 30 to 902 can also be used for internal NC drives**
- **Telegrams 30 to 903 can also be used for external drives (CU320)**

Activation	Purpose	Designation of the process data (status word / control word)	
PROFIsafe telegram 30	Standard telegram (Basic Functions)	• S_STW1 • S_ZSW1	• Safety control word 1 • Safety status word 1
PROFIsafe telegram 31	Extended standard telegram (Extended Functions)	• S_STW2 • S_ZSW2	• Safety control word 2 • Safety status word 2
SIEMENS telegram 901	Extended manufacturer-specific telegram with **16-bit position value**.	• S_STW2 • S_SLS_LIMIT_A • S_ZSW2 • S_SLS_LIMIT_A_ACTIVE • S_CYCLE_COUNT • S_XIST16	• Safety control word 2 • SLS limit value specification • Safety status word 2 • Active SLS limit value • Counter for the Safety cycle • Current actual position value (16-bit)
SIEMENS telegram 902	Extended manufacturer-specific telegram with **32-bit position value**. Default setting for internal NC drives with **firmware V4.7**.	• S_CYCLE_COUNT • S_XIST32 • S_STW2 • S_SLS_LIMIT_A • S_ZSW2 • S_SLS_LIMIT_A_ACTIVE • S_ZSWCAM1	• Counter for the Safety cycle • Current actual position value (32-bit) • Safety control word 2 • SLS limit value specification • Safety status word 2 • Active SLS limit value • Safe Cam status word
SIEMENS telegram 903	Extended manufacturer-specific telegram with **32-bit Safe Cam (S_ZSWCAM1).** Default setting for internal NC drives as of **firmware V4.8.**		
SIEMENS telegram 701	Manufacturer-specific telegram for SIC/SCC communication. For **SINUMERIK,** the **SIC/SCC communication** is exclusively via telegram 701.	• S_STW1B • S_STW3B • S_ZSW1B • S_ZSW2B • S_V_LIMIT_B • S_ZSW3B	• Safety Control Channel control word 1 • Safety Control Channel control word 3 • SI Motion / Safety Info Channel status word 1 • Safety Info Channel status word 2 • Safety Info Channel speed setpoint limitation • Safety Info Channel status word 3

Notes:

The following tables show the *PROFIsafe process data PZD 1...6 used depending on the telegrams:*

The PROFIsafe process data used are listed in detail in the following and can be selected in the table by means of a link.

Process data / Telegram:		903	902	901	31	30
Safety control word 1 Basic Functions	S_STW1	—	—	—	—	PZD1
Safety status word 1 Basic Functions	S_ZSW1	—	—	—	—	PZD1
Safety control word 2 Extended Functions	S_STW2	PZD1...2	PZD1...2	PZD1...2	PZD1...2	—
Safety status word 2 Extended Functions	S_ZSW2	PZD1...2	PZD1...2	PZD1...2	PZD1...2	—
SLS limit value specification	S_SLS_LIMIT_A	PZD3	PZD3	PZD3	—	—
Active SLS limit value	S_SLS_LIMIT_A_ACTIVE	PZD5	PZD3	PZD3	—	—
Counter for the Safety cycle	S_CYCLE_COUNT	—	PZD4	PZD4	—	—
Current actual position value (16-bit)	S_XIST16	—	—	PZD5	—	—
Current actual position value (32-bit)	S_XIST32	—	PZD5...6	—	—	—
Safe Cam status word	S_ZSW_CAM1	PZD3...4	—	—	—	—

PROFIsafe telegram 30		
	Send data	**Receive data**
PZD1	S_STW1	S_ZSW1

PROFIsafe telegram 31		
	Send data	**Receive data**
PZD1	S_STW2	S_ZSW2
PZD2		

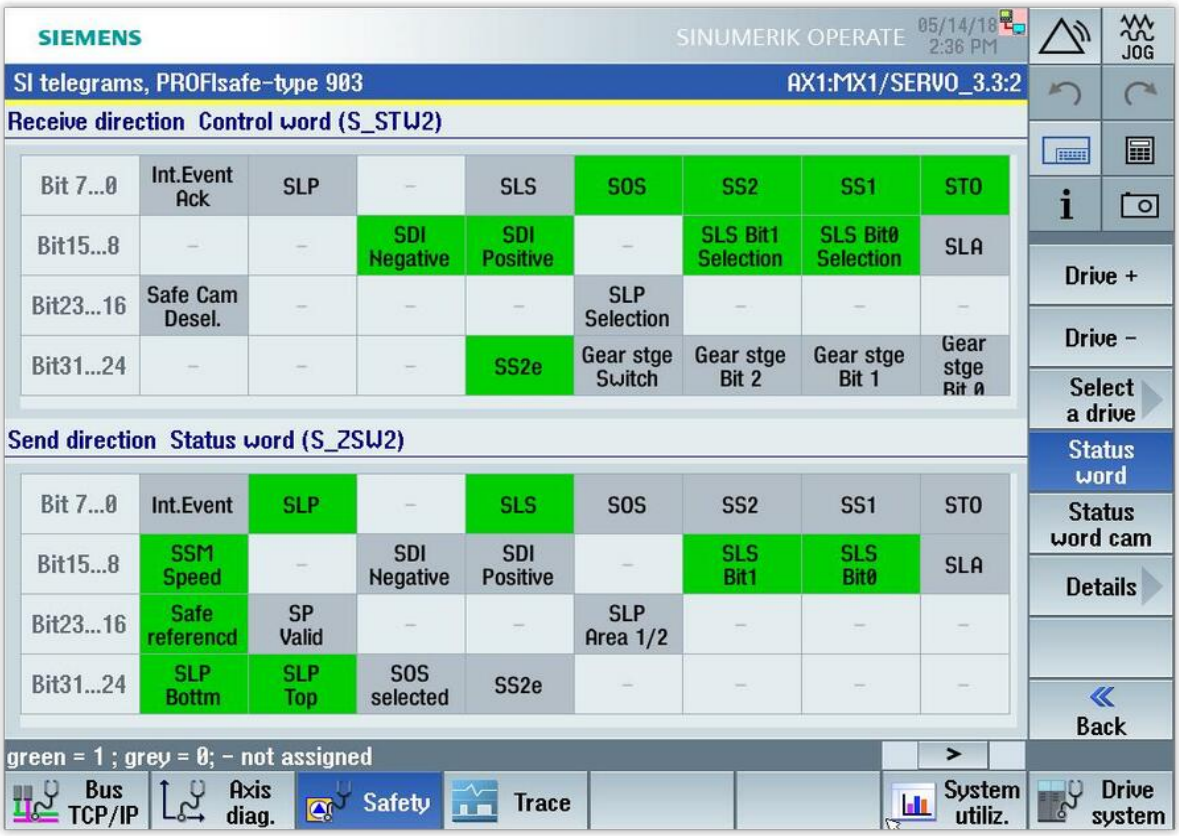
PROFIsafe telegram 902		
	Send data	**Receive data**
PZD1	S_STW2	S_ZSW2
PZD2		
PZD3	S_SLS_LIMIT_A	S_SLS_LIMIT_A_ACTIVE
PZD4	—	S_CYCLE_COUNT
PZD5	—	S_XIST32
PZD6		

PROFIsafe telegram 901		
	Send data	**Receive data**
PZD1	S_STW2	S_ZSW2
PZD2		
PZD3	S_SLS_LIMIT_A	S_SLS_LIMIT_A_ACTIVE
PZD4	—	S_CYCLE_COUNT
PZD5	—	S_XIST16

PROFIsafe telegram 903		
	Send data	**Receive data**
PZD1	S_STW2	S_ZSW2
PZD2		
PZD3	S_SLS_LIMIT_A	S_ZSW_CAM1
PZD4	—	
PZD5	—	S_SLS_LIMIT_A_ACTIVE

Notes:

HMI diagnostics Safety Integrated plus PROFIsafe telegrams:
At Diagnostics > Safety > SI telegrams > PROFIsafe, the safe drive telegrams can be diagnosed with receive and send directions.



Notes:

Safety Info Channel and Safety Control Channel (SIC/SCC):

In the SINUMERIK Safety Integrated F_PLC (F-PLC) mode, the status and control signals of the drive-based motion control are connected to the PLC and the NCK:

- Via the **Safety Control Channel (SCC)**, **control signals are sent from the PLC and the NCK to the drive**: **S_STW1B and S_STW3B** , e.g. to start and influence the "Safe Brake Test (SBT)" function in the drive.
- Via the **Safety Info Channel (SIC)**, the following **status signals of the drive are sent to the PLC:** Status information of the Safety Integrated functionality of the drive: **S_ZSW1B, S_ZSW2B, S_ZSW3B, S_V_LIMIT_B**, such as for the brake test.
- **For a STOP response of the drive, status signals are transferred to the NCK and to the PLC**. The motion control in the NCK responds automatically to the STOP response of the drive, for example, by not specifying any further setpoints for the path. This avoids alarms that need to be acknowledged, such as "contour monitoring" or "position monitoring".

SIC/SCC parameterization and access to the signals:

- SIC/SCC must be activated while commissioning the drive in SINUMERIK Operate. The SIC/SCC signals are represented in the **PLC user interface** and so can be accessed via the axis DB DB31...DB61.

Notes:

- SIC and SCC can only be configured and parameterized together.
- Axis-specific mixed operation is not possible with "Safety Integrated (SPL)" in "SINUMERIK Safety Integrated F_PLC (F-PLC)" mode.
- SIC/SCC communication can generally be transferred via the predefined PROFIdrive 700 or 701 telegram.
  For SINUMERIK, however, only the 701 telegram can be used. If a different telegram is configured for SIC/SCC, the cyclical SIC/SCC communication is suppressed and Alarm 27811 issued.

The following table shows the *PROFIdrive SIC / SCC process data of telegram 701:*

PROFIdrive Telegram 701				
	Send data	**Parameter**	**Receive data**	**Parameter**
PZD1	Safety Info Channel Status Word 1 S_ZSW1B	**r9734**	Safety Control Channel Control Word 1 S_STW1B	**p10250**
PZD2	Safety Info Channel Status Word 2 S_ZSW2B	**r9743**	Safety Control Channel Control Word 3 S_STW3B	**p10235**
PZD3	Safety Info Channel- Setpoint velocity limitation S_V_LIMIT_B	**r9733[2]**		
PZD4				
PZD5	Safety Info Channel Status Word 3 S_ZSW3B	**r10234**		

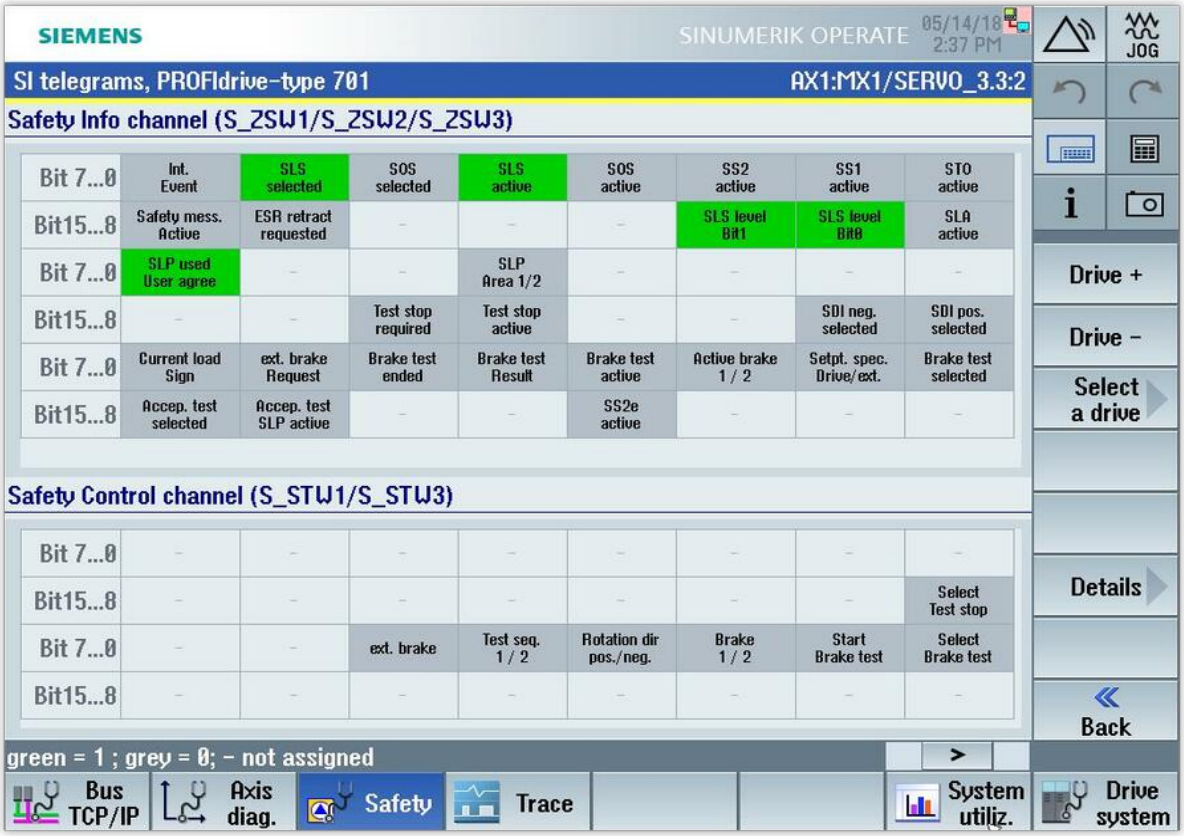
PLC user interface (DB3x/DB3x00):

The previously listed PROFIdrive **SIC/SCC** process data in the SINUMERIK Safety Integrated F_PLC (F-PLC) is also represented in the **axis-specific data of the axis DBs (Axis 1 [DB31] ... Axis 31 [DB61]).**

The **assignment of the SIC/SCC** process data **in the axis DB** is represented as follows in the interface signals to axis/spindle / signals from axis/spindle.

Notes:

HMI diagnostics Safety Integrated plus PROFIsafe telegrams:
At Diagnostics > Safety > SI telegrams > PROFIsafe, the "non-safe" control and status words can be diagnosed.



Notes:

DB31...DB61	*Signals to axis/spindle*							
/FBSIsl/	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte	SCC (PLC → drive)							
S_STW1B: Safety Control Channel control word 1								
DBB140 Safety Integrated								Test stops for Extended Functions
DBB141 Safety Integrated								
DBB142 Safety Integrated								
S_STW3B: Safety Control Channel control word 3								
DBB143 Safety Integrated			External brake closed	Torque 1 or 2	Direction of rotation	Test with brake 1 or 2	Start brake test	Select brake test
DBB144… DBB163 Safety Integrated								

Notes:

DB31...DB61	*Signals from axis/spindle*							
/FBSIsl/	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte	SIC (Drive → PLC)							
S_ZSW1B: SI Motion / Safety Info Channel status word 1								
DBB164 Safety Integrated	Safety fault active	ESR requested	SDI negative selected	SDI positive selected		SLS bit 1 limit value Safely-Limited-Speed	SLS bit 0 limit value Safely-Limited-Speed	
DBB165 Safety Integrated	Safety fault with Stop A	SLS Safely-Limited-Speed selected	SOS Safe Operating Stop selected	SLS Safely-Limited-Speed active	SOS Safe Operating Stop active	SS2 Safe Stop 2	SS1 Safe Stop 1	STO Safe Torque Off
S_ZSW2B: Safety Info Channel status word 2								
DBB166 Safety Integrated			Test stop required	Test stop active			SDI Safe Direction negative	SDI Safe Direction positive
DBB167 Safety Integrated	SLP Safely-Limited-Position is selected and User agreement set			SLP range 2 for Safely-Limited-Position selected				
S_V_LIMIT_B: Safety Info Channel speed setpoint limitation								
DBD168 Safety Integrated	SLS speed limit							
S_ZSW3B: Safety Info Channel status word 3								
DBB172 Safety Integrated	Acceptance test mode selected	Acceptance test SLP active				REPOS delay acknowledgment	REPOS offset valid	REPOS offset
DBB173 Safety Integrated	Sign of load torque negative	Close external brake	Brake test completed	Brake test OK	Brake test active	Brake test with brake 2	Setpoint specification during SBT in the drive	Safe brake test selected

Notes:

Further information:

Enabling of the PROFIsafe activation:
Activation via PROFIsafe is available not only for Safety Integrated Basic Functions, but also for Safety Integrated Extended Functions, and must be enabled in p9601.
p9601 SI enable drive-integrated functions (Control Unit) / SI enable CU function:
- 0008 hex: Basic Functions via PROFIsafe are enabled (permitted for r9771.6 = 1).
- 0009 hex: Basic Functions via PROFIsafe and online terminals are enabled (permitted for r9771.6 = 1).
- 000C hex: Extended Functions via PROFIsafe are enabled (permitted for r9771.4 = 1).
- 000D hex: Extended Functions via PROFIsafe and Basic Functions via online terminals are enabled (permitted for r9771.4 = 1).
- p9601.0 = 1: STO (SH) enabled via terminals (CU)

r9771 SI shared functions (Control Unit) / SI CU shared functions:
- r9771.4 = 1: PROFIsafe Extended Functions supported
- r9771.6 = 1: PROFIsafe Basic Functions supported

Note: Safety Integrated Basic Functions via PROFIsafe and via terminals:
The activation of the Basic Functions via terminals on the Control Unit and on the Motor Module / Power Module (parameter p9601.0 = 1) may also be enabled.
To make SS1 selectable, an SS1 delay time p9652 > 0 must be configured.
SS1 and STO can then be selected with PROFIsafe.
In this case, only SS1 is available for activation via a terminal.
STO has priority over SS1, i.e. if SS1 and STO are selected simultaneously, STO becomes active.

Selection of the PROFIsafe telegram:
Telegram 902, 901, 31 or 30 can be used for PROFIsafe communication via PROFIBUS and PROFINET. Siemens telegram 902 is the default setting for internal NC drives.
- The settings for the F-PLC (F-Host) are made in the TIA Portal.
- The settings for the individual drives (F-Slaves or F-Devices) with SINUMERIK Operate
  - **p60022** PROFIsafe telegram selection
  - **p9611** SI PROFIsafe telegram selection (Control Unit)
  - **p9811** SI PROFIsafe telegram selection (Motor Module)

Time separation between switching operations:
If switching operations occur too often, message F01611 with fault value 1000 is issued. The cause depends on the activation type:
- Continuous signal changes occurred at F-DI.
- A continuous STO (also as secondary response) was triggered via PROFIsafe.

Within a time of 5 × p9650, at least two switching operations must have a separation of at least p9650 at the terminals or via PROFIsafe.

ESR behavior in the event of communication failure:

See "Section: Further information > Drive-integrated SI functions activation / telegrams"

Notes:

Acknowledgment via PROFIsafe:
The PROFIsafe acknowledgment must be programmed in the F-PLC safety program by setting the "Internal Event ACK" signal separately for each drive object via the PROFIsafe telegram (STW bit 7). A falling edge in the "Internal Event ACK" signal resets the "Internal event" status in the associated drive and so acknowledges the fault.

Acknowledgment via POWER ON:
Like all other faults, the acknowledgment of the safety faults also functions by switching the drive device off and on (POWER ON).
If the fault cause is not yet rectified, the fault reappears immediately after startup.

Additional acknowledgment via PROFIdrive RESET:



Unexpected restart for safe acknowledgment without MCP Reset!

- If a STOP response from the machine operator is acknowledged with a safe acknowledgment, but is acknowledged without PROFIdrive RESET, the part program is not canceled and continues with the acknowledgment!
- Consequently, integrate the safe acknowledgment (PROFIsafe IEACK) in the safety program with a RESET at the machine control panel (PROFIdrive RESET).

Various fault situations also require an acknowledgment with PROFIdrive RESET for the safe acknowledgment.

This affects the following STOP responses and safety functions:

- **Extended Safety Functions:**
  - **All fault situations, safety functions and STOP responses**
- Basic Functions or inactive safety message buffer:
  - Parameterization of SLS with STOP D or STOP E

If in case of a fault, the safe acknowledgment **without PROFIdrive RESET** is made, although the part program remains stopped, the **spindle resumes running**.
Such a configuration does not conform with Standard IEC 61800-5-2. In accordance with the Standard, a restart lock is required for all STOP functions and the spindle may restart only after RESET.

Consequently, integrate the safe acknowledgment (PROFIsafe IEACK) in the safety program with a RESET at the machine control panel (PROFIdrive RESET).
Thus, configure the machine control panel RESET (MCP reset or channel/BAG RESET) so that the actuation triggers a PROFIdrive RESET and a safe acknowledgment (e.g. PROFIsafe IEACK).

For the following items, see
"Section: Further information
> Drive-integrated SI functions activation / telegrams"

- Additional acknowledgment via PROFIdrive RESET for Basic Functions
- Acknowledgment via F-DI
- Extended acknowledgment (deselect STO/SS1)
- Additional acknowledgment for an inactive safety message buffer

Notes:

S_STW1: Safety control word 1 (Basic Functions)

[1] Non-relevant signals for Basic Functions should be set to "0".

Byte	Bit	Meaning	Remarks / r9720	
0	0	STO	1	Deselect STO
			0	Select STO
	1	SS1	1	Deselect SS1
			0	Select SS1
	2	SS2	0	— [1]
	3	SOS	0	— [1]
	4	SLS	0	— [1]
	5	Reserved	—	—
	6	SLP	0	— [1]
	7	Internal event ACK	1/0	Acknowledgment
			0	No acknowledgment
1	0	Reserved	—	—
	1	Select SLS bit 0	0	— [1]
	2	Select SLS bit 1	0	
	3	Reserved	—	—
	4	SDI positive	0	— [1]
	5	SDI negative	0	
	6, 7	Reserved	—	—

S_ZSW1: Safety status word 1 (Basic Functions)

[1] Non-relevant signals for Basic Functions must not be evaluated.

Byte	Bit	Meaning	Remarks / r9722	
0	0	STO active	1	STO active
			0	STO not active
	1	SS1 active	1	SS1 active
			0	SS1 not active
	2	SS2 active	0	— [1]
	3	SOS active	0	— [1]
	4	SLS active	0	— [1]
	5	Reserved	—	—
	6	SLP active	0	— [1]
	7	Internal event	1	Internal event
			0	No internal event
1	0	Reserved	—	—
	1	Active SLS level bit 0	0	— [1]
	2	Active SLS level bit 1	0	
	3	SOS selected	0	— [1]
	4	SDI positive active	0	— [1]
	5	SDI negative active	0	— [1]
	6	Reserved	—	—
	7	SSM (speed below limit value)	0	— [1]

Notes:

S_STW2: Safety control word 2 (Extended Functions)				
Byte	**Bit**	**Meaning**	**Remarks / r9720**	
0	0	STO	1	Deselect STO
			0	Select STO
	1	SS1	1	Deselect SS1
			0	Select SS1
	2	SS2	1	Deselect SS2
			0	Select SS2
	3	SOS	1	Deselect SOS
			0	Select SOS
	4	SLS	1	Deselect SLS
			0	Select SLS
	5	Reserved	—	—
	6	SLP	1	Deselect SLP
			0	Select SLP
	7	Internal event ACK	1/0	Acknowledgment
			0	No acknowledgment
1	0	Reserved	—	—
	1	Select SLS bit 0	—	Select the speed limit for SLS (2 bits)
	2	Select SLS bit 1	—	
	3	Reserved	—	—
	4	SDI positive	1	Deselect SDI positive
			0	Select SDI positive
	5	SDI negative	1	Deselect SDI negative
			0	Select SDI negative
	6, 7	Reserved	—	—

S_ZSW2: Safety status word 2 (Extended Functions):				
Byte	**Bit**	**Meaning**	**Remarks / r9722**	
0	0	STO active	1	STO active
			0	STO not active
	1	SS1 active	1	SS1 active
			0	SS1 not active
	2	SS2 active	1	SS2 active
			0	SS2 not active
	3	SOS active	1	SOS active
			0	SOS not active
	4	SLS active	1	SLS active
			0	SLS not active
	5	Reserved	—	—
	6	SLP active	1	SLP active
			0	SLP not active
			—	The "SLP active" status signal is not the same as the "SLP active" diagnostics signal (r9722.6), but rather the same as the AND logic operation from "SLP active" (r9722.6) and "safely referenced" (r9722.23).
	7	Internal event	1	Internal event
			0	No internal event
1	0	Reserved	—	—
	1	Active SLS level bit 0	—	Display of the speed limit for SLS (2 bits)
	2	Active SLS level bit 1	—	
	3	SOS selected	1	SOS selected
			0	SOS deselected
	4	SDI positive active	1	SDI positive active
			0	SDI positive not active
	5	SDI negative active	1	SDI negative active
			0	SDI negative not active
	6	Reserved	—	—
	7	SSM (speed)	1	SSM (speed below limit value)
			0	SSM (speed above or the same as the limit value)

Notes:

S_STW2: Safety control word 2 (Extended Functions)				
Byte	**Bit**	**Meaning**	**Remarks / r9720**	
2	0...2	Reserved	—	—
	3	Select SLP position range	1	Select SLP range 2 (SLP2)
			0	Select SLP range 1 (SLP1)
	4...6	Reserved	—	—
	7	Deselect Safe Cam	1	Deselect Safe Cam
			0	Select Safe Cam
3	0	Select gear stage bit 0	—	Selection of the gear stage (3 bits)
	1	Select gear stage bit 1	—	
	2	Select gear stage bit 2	—	
	3	Gear stage change		
	4	Deselect Safe Stop 2 external	1	Deselect SS2e
			0	Select SS2e
	5...7	Reserved	—	—

S_ZSW2: Safety status word 2 (Extended Functions)				
Byte	**Bit**	**Meaning**	**Remarks / r9722**	
2	0...2	Reserved	—	—
	3	SLP active position range	1	SLP range 2 (SLP2) active
			0	SLP range 1 (SLP1) active
	4, 5	Reserved	—	—
	6	Safe Position valid	1	Safe Position valid
			0	Safe Position invalid
	7	Safely referenced	1	Safe Position is considered as "safely referenced"
			0	Safe Position is not considered as "safely referenced"
3	0	F-DI 0 (only valid for CU310-2)	1	F-DI 0 inactive
			0	F-DI 0 active
	1	F-DI 1 (only valid for CU310-2)	1	F-DI 1 inactive
			0	F-DI 1 active
	2	F-DI 2 (only valid for CU310-2)	1	F-DI 2 inactive
			0	F-DI 2 active
	3	Reserved	—	—
	4	Safe Stop 2 external selected	1	SS2e selected
			0	SS2e deselected
	5	SOS selected	1	SOS selected
			0	SOS deselected
	6	SLP upper limit observed	1	SLP: Upper limit observed
			0	SLP: Upper limit not observed
			—	The "SLP upper limit observed " status siganl always corresponds to the "SLP upper limit observed " diagnostics signal (r9722.30).
	7	SLP lower limit observed	1	SLP: Lower limit observed
			0	SLP: Lower limit not observed
			—	The "SLP lower limit observed " status siganl always corresponds to the "SLP lower limit observed " diagnostics signal (r9722.31).

S_SLS__LIMIT_A: SLS limit value specification:
- PZD3 in telegrams 901, 902 and 903, output signals
- SLS limit value specification
- Value range 1 ... 32767; 32767 ≙ 100% of the 1st SLS level

S_CYCLE_COUNT: Counter for the safety cycle:
- PZD4 in telegrams 901 and 902, input signals
- Counter for the safety cycle
- Value range -32768 ... +32767
- May be evaluated only when the transfer of safe position values is active (p9501.25 = 1) and the position value is valid (r9722.22 = r9722.23 = 1).

S_SLS_LIMIT_A_ACTIVE: Active SLS limit value:
- PZD3 in telegrams 901 and 902, input signals
- PZD5 in telegram 903, input signals
- Active SLS limit value
- Value range 1 ... 32767; 32767 ≙ 100%
- May be evaluated only when SLS 1 is active and p9501.24 = 1.

S_XIST16: Current actual position value (16-bit):
- PZD5 in telegram 901, input signals
- Current actual position value (16-bit)
- Value range ±32767
- May be evaluated only when the transfer of safe position values is active (p9501.25 = 1) and the position value is valid (r9722.22 = r9722.23 = 1).
- **Scaling by p9574:** The transferred position value in S_XIST16 must not exceed the value range that can be represented. For this reason, the safe position value of the drive (r9713[0]) can be given a scaling factor. Before the transfer, the position value is divided by this factor. Reducing the accuracy therefore allows a larger value range to be transferred.
- **Example:** For a position of -29,999 mm reported in r9708[0] and r9708[1], and a scaling factor of p9x74 = 1000, a numeric value of -29 is reported to the control.

S_XIST32: Current actual position value (32-bit):
- PZD5 and PZD6 in telegram 902, input signals
- Current actual position value (32-bit)
- Value range ±737280000
- Unit: 1 µm (linear axis), 0.001° (rotary axis)
- May be evaluated only when the transfer of safe position values is active (p9501.25 = 1) and the position value is valid (r9722.22 = r9722.23 = 1).



Notes:

S_ZSW_CAM1: Safe Cam

Description of Safe Cam Safety status word

Byte	Bit	Meaning	Remarks / r9703	
0	0	Position at cam 1	1	Position is at cam 1
			0	Position is not at cam 1
	1	Position at cam 2	1	Position is at cam 2
			0	Position is not at cam 2
	2	Position at cam 3	1	Position is at cam 3
			0	Position is not at cam 3
	3	Position at cam 4	1	Position is at cam 4
			0	Position is not at cam 4
	4	Position at cam 5	1	Position is at cam 5
			0	Position is not at cam 5
	5	Position at cam 6	1	Position is at cam 6
			0	Position is not at cam 6
	6	Position at cam 7	1	Position is at cam 7
			0	Position is not at cam 7
	7	Position at cam 8	1	Position is at cam 8
			0	Position is not at cam 8

Byte	Bit	Meaning	Remarks / r9703	
1	0	Position at cam 9	1	Position is at cam 9
			0	Position is not at cam 9
	1	Position at cam 10	1	Position is at cam 10
			0	Position is not at cam 10
	2	Position at cam 11	1	Position is at cam 11
			0	Position is not at cam 11
	3	Position at cam 12	1	Position is at cam 12
			0	Position is not at cam 12
	4	Position at cam 13	1	Position is at cam 13
			0	Position is not at cam 13
	5	Position at cam 14	1	Position is at cam 14
			0	Position is not at cam 14
	6	Position at cam 15	1	Position is at cam 15
			0	Position is not at cam 15
	7	Position at cam 16	1	Position is at cam 16
			0	Position is not at cam 16

Byte	Bit	Meaning	Remarks / r9703	
2	0	Position at cam 17	1	Position is at cam 17
			0	Position is not at cam 17
	1	Position at cam 18	1	Position is at cam 18
			0	Position is not at cam 18
	2	Position at cam 19	1	Position is at cam 19
			0	Position is not at cam 19
	3	Position at cam 20	1	Position is at cam 20
			0	Position is not at cam 20
	4	Position at cam 21	1	Position is at cam 21
			0	Position is not at cam 21
	5	Position at cam 22	1	Position is at cam 22
			0	Position is not at cam 22
	6	Position at cam 23	1	Position is at cam 23
			0	Position is not at cam 23
	7	Position at cam 24	1	Position is at cam 24
			0	Position is not at cam 24

Byte	Bit	Meaning	Remarks / r9703	
3	0	Position at cam 25	1	Position is at cam 25
			0	Position is not at cam 25
	1	Position at cam 26	1	Position is at cam 26
			0	Position is not at cam 26
	2	Position at cam 27	1	Position is at cam 27
			0	Position is not at cam 27
	3	Position at cam 28	1	Position is at cam 28
			0	Position is not at cam 28
	4	Position at cam 29	1	Position is at cam 29
			0	Position is not at cam 29
	5	Position at cam 30	1	Position is at cam 30
			0	Position is not at cam 30
	6	Safe Cam active	1	Safe Cam is active
			0	Safe Cam is not active
	7	Validity of the Safe Cam values	1	Safe Cam values are valid
			0	Safe Cam values are not valid

Notes:

SIC/SCC (PROFIdrive) process data in the "Axis/spindle signals" interface DB31...DB61:

S_STW1B: Safety Control Channel control word 1				
Bit	**Meaning**	**Remarks**		**Parameter**
0...7	Reserved	—	—	—
8	Extended Functions test stop / forced checking procedure	1	Extended Functions test stop / forced checking procedure selected	r10251.8
		0	Extended Functions test stop / forced checking procedure deselected	
9...15	Reserved	—	—	—

S_STW3B: Safety Control Channel control word 3:				
Bit	**Meaning**	**Remarks**		**Parameter**
0	Select brake test	1	Brake test selected	r10231.0
		0	Brake test deselected	
1	Start brake test	1	Start brake test requested	r10231.1
		0	Start brake test not requested	
2	Brake selection	1	Test brake 2 selected	r10231.2
		0	Test brake 1 selected	
3	Select direction of rotation	1	Negative direction selected	r10231.3
		0	Positive direction selected	
4	Select test sequence	1	Test sequence 2 selected	r10231.4
		0	Test sequence 1 selected	
5	Status of external brake	1	External brake closed	r10231.5
		0	External brake open	
6...15	Reserved	—	—	—

Notes:

S_ZSW1B: SI Motion-/ Safety Info Channel status word:				
Bit	**Meaning**	**Remarks**		**Parameter**
0	STO active	1	STO active	r9734.0
		0	STO not active	
1	SS1 active	1	SS1 active	r9734.1
		0	SS1 not active	
2	SS2 active	1	SS2 active	r9734.2
		0	SS2 not active	
3	SOS active	1	SOS active	r9734.3
		0	SOS not active	
4	SLS active	1	SLS active	r9734.4
		0	SLS not active	
5	SOS selected	1	SOS selected	r9734.5
		0	SOS not selected	
6	SLS selected	1	SLS selected	r9734.6
		0	SLS not selected	
7	Internal event	1	Internal event	r9734.7
		0	No internal event	
8	Reserved	—	—	—
9	Active SLS level bit 0	—	Display of the speed limit for SLS (2 bits)	r9734.9
10	Active SLS level bit 1	—		r9734.10
11	Reserved	—	—	—
12	SDI positive selected	1	SDI positive selected	r9734.12
		0	SDI positive deselected	
13	SDI negative selected	1	SDI negative selected	r9734.13
		0	SDI negative deselected	
14	ESR retraction requested	1	ESR retraction requested	r9734.14
		0	ESR retraction not requested	
15	Safety message effective		Safety message effective	r9734.15
			No Safety message effective	

Notes:

S_ZSW2B: Safety Info Channel status word 2				
Bit	**Meaning**	**Remarks**		**Parameter**
0...3	Reserved	—	—	—
4	SLP selected position range	1	SLP range 2 selected	r9743.4
		0	SLP range 1 selected	
5...6	Reserved	—	—	—
7	SLP selected and user agreement	1	SLP selected and user agreement set	r9743.7
		0	SLP selected and user agreement not set	
8	SDI positive	1	SDI positive selected	r9743.8
		0	SDI positive deselected	
9	SDI negative	1	SDI negative selected	r9743.9
		0	SDI negative deselected	
10, 11	Reserved	—	—	—
12	Test stop active	1	Test stop active	r9743.12
		0	Test stop not active	
13	Test stop required	1	Test stop required	r9743.13
		0	Test stop not required	
14, 15	Reserved	—	—	—

S_ZSW3B: Safety Info Channel status word 3				
Bit	**Meaning**	**Remarks**		**Parameter**
0	Brake test	1	Brake test selected	r10234.0
		0	Brake test deselected	
1	Setpoint specfication drive/external	1	Setpoint specfication for the drive (The speed setpoint is specified by the SBT function.)	r10234.1
		0	External setpoint specfication (control) (The "normal" speed setpoint is effective.)	
2	Active brake	1	Test brake 2 active	r10234.2
		0	Test brake 1 active	
3	Brake test active	1	Test active	r10234.3
		0	Test inactive	
4	Brake test result	1	Test successful	r10234.4
		0	Test faulty	
5	Brake test completed	1	Test run	r10234.5
		0	Test incomplete	
6	External brake request	1	Close brake	r10234.6
		0	Open brake	
7	Current load sign	1	Sign negative	r10234.7
		0	Sign positive	
8...13	Reserved	—	—	—
14	Acceptance test SLP active	1	Acceptance test SLP active	r10234.14
		0	Acceptance test SLP not active	
15	Acceptance test mode selected		Acceptance test mode selected	r10234.15
			Acceptance test mode deselected	

S_V_LIMIT_B: Safety Info Channel speed setpoint limitation:	
Speed setpoint limitation (SLS speed limit) with a 32-bit resolution with sign bit	
Meaning	**Parameter**
SLS speed limit	r9733[2]
Scaling of the SLS speed limit S_V_LIMIT_B = 4000 0000 hex = speed in p2000	p2000

Overview of important parameters:
(see SINAMICS S120/S150 List Manual)

PROFIsafe activation:

- p9563[0...3] SI Motion SLS(SG)-specific stop response (Control Unit)
- p9566 SI Motion SDI stop response (Control Unit)
- p9580 SI Motion STO delay bus failure (Control Unit)
- p9601 SI enable drive-integrated functions (Control Unit)
- p9610 SI PROFIsafe address (Control Unit)
- p9611 SI PROFIsafe telegram selection (Control Unit)
- p60022 PROFIsafe telegram selection

SIC/SCC:

- r9733[0...2] CO: SI Motion speed setpoint limitation effective
- r9734.0...15 CO/BO: SI Safety Info Channel status word S_ZSW1B
- r9743.4...15 CO/BO: SI Safety Info Channel status word S_ZSW2B
- r10231 SI Motion SBT control word diagnostics
- r10234.0...15 CO/BO: SI Safety Info Channel status word S_ZSW3B
- p10235 CI: SI Safety Control Channel control word S_STW3B
- p10250 CI: SI Safety Control Channel control word S_STW1B
- r10251.8...12 CO/BO: SI Safety Control Channel control word S_STW1B diagnostics
- p60122 IF1 PROFIdrive SIC/SCC telegram selection

Notes:

Further information:

Help for standards:

- **EN 62061:** Siemens function example for 62061
  http://support.industry.siemens.com/cs/document/23996473

- **EN ISO 13849:** BGIA Report 2008
  http://www.dguv.de/bgia/13849

- **EN62061 and EN ISO 13849:** Siemens Standards Brochure, Standards Poster
  http://www.automation.siemens.com/cd/safety/index_00.htm

- **EU Directives:** Directives, activities for the directives, list of harmonized standards, FAQs
  http://www.newapproach.org

- **Standards:**
  http://irc.siemens.de/NORIS

- **Specialist Book:** Functional Safety of Machines and Plants. Implementation of the European Machinery Directive in Practice
  (ISBN 978-3-89578-366-1)

- **SITRAIN Course ST-FASAFN:** CE marking & functional safety in machine and plant construction
  http://siemens.com/sitrain

- **Workshop VM-SIMA:** The way to a safe machine in accordance with the current standards
  http://siemens.com/sitrain

- **Safety Evaluation Tool:**
  http://www.industry.siemens.com/topics/global/en/safety-integrated/machine-safety/safety-evaluation-tool/Pages/Default.aspx

Overview: SINUMERIK Safety Integrated plus:

- **List of approved encoders in Siemens Industry Online Support under Entry ID 33512621:**
  https://support.industry.siemens.com/cs/document/33512621?dti=0&dl=de

Requirements, software and supplementary conditions:

- **SINUMERIK ONE Safety Integrated plus Commissioning Manual.pdf** (CNC V6.13 / Toolbox V16)
  (Appendix to the course documentation)

- **Getting Started SINUMERIK STEP7 Toolbox V16 de/en.pdf** (Appendix to the course documentation)

- **V16_Prof_ReadMe_de/en.pdf** (Appendix to the course documentation)

- **Supplementary_conditions_toolbox_V16** (Toolbox SW > Documents > Readme > English > Readme_TIA_ToolboxenUS.HTML)

TIA Selection Tool:

Notes:

- www.siemens.com/TIA Portal

- **SIZER for Siemens Drives:**
  http://w3.siemens.com/mcms/mc-solutions/de/software/antriebs-projektierungs-tool-sizer/seiten/antriebsprojektierungstool-sizer.aspx

Drive-integrated safety functions of Sinumerik ONE:

- **SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf** (Appendix to the course documentation)

- **SINAMICS S120 Safety Integrated Function Manual de/en.pdf** (Appendix to the course documentation)

- **FAQ retraction:**
  http://support.automation.siemens.com/WW/view/en/65128501

- **Suitable motors and encoders for the Safety Integrated Functions of the SINAMICS S120 drive system and for SINUMERIK Safety Integrated:**
  http://support.automation.siemens.com/WW/view/en/33512621

Activation of drive-integrated SI Functions / telegrams:

- **Motion monitoring without selection:** SINAMICS S120 Safety Integrated Function Manual 072016 de/en.pdf
  (Appendix to the course documentation)

- **Activation via terminals on the Control Unit and the Motor/Power Module:**
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)
  SINAMICS S120 Safety Integrated Function Manual 072016 de/en.pdf (Appendix to the course documentation)

- **Activation via PROFIsafe:**
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)

- **ESR behavior at communication failure:**
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)

- **Additional acknowledgment via PROFIdrive RESET for Basic Functions**:
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)

- **Acknowledgment via F-DI:**
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)

- **Extended acknowledgment (deselection of STO/SS1):**
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)

- **Additional acknowledgment for inactive Safety message buffer:**
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)

Notes:

Configuration of the ET200SP with F-I/O:

- **Application cases of the electronic modules according to the requirements on fail-safety (SIL1/Kat.1/PLc … SIL3/Kat.4//PLe):** (Appendix to the course documentation)

  Manual: et200sp_f_di_8x24vdc_hf_manual_en-US_en-US.pdf
  Manual: et200sp_f_dq_4x24vdc_2a_pm_hf_manual_en-US_en-US.pdf
  Manual: et200sp_f_pm-e_24vdc8a_ppm_st_manual_en-US_en-US.pdf
  Manual: et200sp_di_8x24vdc_hf_manual_en-US_en-US.pdf
  Manual: et200sp_dq_8x24vdc_0_5a_hf_manual_en-US_en-US.pdf

Safety programming SINUMERIK F-PLC:

- **TIA Information System > PLC Programming > SIMATIC Safety > Configuring and Programming > Safety Administration Editor**

- **Benefits of libraries:**
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)
  Safety-related communication:
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)
  Reading of variables from the standard user program that can change during the runtime of an F-runtime group, as well as the use of clock memory in the F-program:
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)

Commissioning, F-PLC, diagnostics: STO, SS1, SS2, SS2e, SOS, SAM, SSM:

- **Configuring sensorless operation:**
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)
  SINAMICS S120 Safety Integrated Function Manual 072016 de/en.pdf (Appendix to the course documentation)

Acceptance test:

- **PFH values of the individual SINAMICS S120 safety components:**
  SIOS article 76254308 (https://support.industry.siemens.com/cs/document/76254308/).

- **PFH and PFDavg values of F-CPUs, F-I/O or SIMATIC components:**
  SIOS article 109481784 (https://support.industry.siemens.com/cs/document/109481784).

- **Identifying and documenting the correctness of the relevant properties or components:**
  SIMATIC Safety - Configuring and Programming - Programming and Operating Manual:
  (https://support.industry.siemens.com/cs/products?search=SIMATIC%20Safety&dtp=Manual&pnid=24471)

- **Observing the requirements in the report for the acceptance certificate:**
  Annex 1 of the Report to the Certificate for SIMATIC Safety System:
  (http://support.automation.siemens.com/WW/view/en/49368678/134200)

- **Safety acceptance by means of SINUMERIK OPERATE:**
  "Help for the Acceptance Test in SINUMERIK Operate Function Manual" (Appendix to the course documentation)
  ”63012568_Anleitung_Acceptance test_DOKU_V11" (SINUCOM NC) (Appendix to the course documentation)

Notes:

Appendix D:

- **Reference to the CMC KillSafety package:** (Download page of the article: SW V4.7 ; Docu SW V4.8 in preparation)
  "109739607_Application_Use_predefined_CMC_packages.pdf"
  https://support.industry.siemens.com/cs/ww/en/view/109739607

- **Hardware/software component replacement:**
  SINUMERIK ONE Safety Integrated plus Commissioning Manual de/en.pdf (Appendix to the course documentation)

Notes:

Index:

Section: Regulations and standards	Page 5
Introduction	5
Standards	5
Harmonized standards	5
International safety standards	5
Overview	5
Type A standards	6
Type B standards	6
Type C standards	6
Machinery Directive (2006/42/EG)	7
Implementation of the Machinery Directive	7
Overview of the CE certification process	7
CE marking	8
Performing a risk assessment	8
Risk reduction	9
Iterative risk reduction process according to EN 1050	9
Relevant standards for machine safety	10
IEC 61508	10
EN ISO 13849	10
EN 61800-5-2	10
IEC 62061	10
EN 60204-1	10
Safety Integrity Level	11
Performance Level	11
Relationsship between SIL, PL and PFH value	11
Use of the EN ISO 13849-1 and EN 62061 safety standards	11
Risk elements S, F, O and P	11
Determination of the required Performance Level (PL) for EN ISO 13849 (PL a - e)	12
Determination of the required Safety Integrity Level (SIL) for EN 62061 (SIL 1, 2, 3)	12
Evaluation of the safety-relevant equipment	13
Safety Evaluation Tool	13

Index:

Section: Basic safety information	Page 15
General safety notes	15
Equipment damage caused by electric fields or electrostatic discharge	15
Industrial security	15
Warranty and liability for sample applications	15
Residual risks for drive systems (Power Drive Systems)	16
Warning: Insecure operating states caused by software	16

Notes:

Index:

Section: Overview SINUMERIK Safety Integrated plus	Page 19
Characteristics of SINUMERIK Safety Integrated F_PLC with integrated F-PLC	20
Safety concept	20
Hardware and firmware extensions	20
F-program	20
Safety-relevant processing with NCU components	21
Processing and communication of safety-relevant information through the components of the NCU	22
Subcomponents of NCU	22
I/O and drives	23
Hardware components	23
F-I/O for PROFINET IO	24
F-I/O for PROFIBUS DP	24
Drive components	24
Drive monitoring with or without encoder	25
Definition: Without encoder	25
Encoders permitted for induction motors and synchronous motors	26
Approbations	27
SIL2	27
PL d	27
Category 3	27
Further detailed information	28
SINAMICS HLA	28
Comparison of electric ↔ hydraulic drives	28
Encoder types supported by SINAMICS HLA	28
Encoder types supported by Basic Functions	28
Encoder types supported by Extended Functions	28

Index:

Section: Requirements, software and supplementary conditions	Page 31
SINUMERIK ONE	31
Engineering system software	31
Engineering system licensing	31
SINUMERIK software	31
Licensing of axes/spindles and F-PLC	31
Optional TIA Selection Tool software	32
Supplementary conditions	33
Encoder as Single Channel Safety Encoder	33
Stop E	33
Dynamic Stop D	33
Performance and number of axes for the PLC 1500	33
Safe cams	33
Maximum 4 drives on a DRIVE-CliQ line	34
Performance and number of axes depends on the desired response time	35

Notes:

Index:

Section: Drive-integrated safety functions of Sinumerik ONE	Page 37
Comparison of the function designations for SINUMERIK	37
System-based stop variants for SINUMERIK Safety Integrated (SPL) and Safety Integrated plus	38
Protection against unexpected start up	39
Safety Integrated plus functions	39
Safe Torque Off (STO) Basic/Extended)	39
Application cases of STO	39
Functional characteristics of STO	39
Extended acknowledgment	39
Unplanned movements of the motor	39
Safe Stop 1 (SS1)	40
Application cases of SS1	40
Procedure at SS1 selection (Basic Functions)	40
Procedure at SS1 selection (Extended Functions)	40
Überwachungsmodi	40
Functional characteristics of Safe Stop 1	40
Setting the delay time	40
Safe Stop 1 with external stop (SS1E)	41
Activation of Safe Stop 1 with external stop	41
Non-monitored axis movements	41
Safe Stop 2 (SS2)	42
Application cases of SS2	42
Procedure at SS2 selection	42
Braking response and diagnostics for SS2 with SAM	42
Functional characteristics of Safe Stop 2	43
Safe Stop 2e with external stop (SS2e)	44
Application cases of SS2e	44
Safe shutdown of the drives on the controur	44
Interruption of the active SS2E function through the SS1 and SS2 functions	44
Unexpected axis movement	44
Stop E	43
Safe Operating Stop (SOS)	46
Application cases of SOS	46
Safe position monitoring	46
In contrast to SS1 and SS2, SOS does not brake the drive independently	46
Drive can be pushed out of the SOS position through mechanical forces	46
Safe Brake Control (SBC)	48
Application cases of SBC	48
Safe control of a brake	48
Procedure at SBC selection	48
Additionally required hardware for other formats	48
No SBC for SINAMICS HLA	49

Index:

Section: Drive-integrated safety functions of Sinumerik ONE	Page 37
Control of the brake via a relay for "Safe Brake Control	49
Unwanted movements of the motor due to a defective brake	49
Safe Brake Management (SBM)	49
Safe Brake Test (SBT)	50
Functional characteristics of SBT	50
Integration with NCK and PLC programs	50
Information for switching: Block required for SBT (Safe Brake Test)	50
Prerequisites for releasing the Safe Brake Test by the NCK	51
Prerequisites for releasing the Safe Brake Test by the DRV (drive)	51
General execution of the Safe Brake Test	51
Cancellation/deselection and acknowledgment (SBT)	51
Acknowledging alarms	51
Safe Brake Ramp (SBR)	52
Functional characteristics of SBR	53
Parameterization of the brake ramp	53
SBR- and OFF3 curve	53
Limitation of the SBR deceleration time	53
Responses for brake ramp violated (SBR)	53
Safe Acceleration Monitor (SAM)	54
Functional characteristics of SAM	54
Application cases of SAM	54
Setting of the SAM tolerance of the actual speed	54
Relationship of SAM and SSM	54
Safe Speed Monitor (SSM)	55
Functional characteristics of SSM	55
Application cases of SSM	55
Relationship of SSM and SAM	55
Danger through unusual behavior of the STOP F for SSM	55
Differences in operation without encoder	56
Safely-Limited Speed (SLS 1...4)	56
Functional characteristics of SLS	56
Application cases of SLS	56
Procedure at SLS selection	56
SLS deselection	56
Switchover of the monitoring limits	56
SLS without selection	56
Behavior at communication failure	57

Notes:

Index:

Section: Drive-integrated safety functions of Sinumerik ONE	Page 37
Safely-Limited Speed (SLS) via PROFIsafe	57
Application cases of SLS via PROFIsafe	57
Speed setpoint limitation and SLS	57
Deviation of the displayed speed limit	57
Parameterization of SLS with STOP D or STOP E requires additional acknowledgment via RESET	57
Connection to the motion control in the NCK	57
Safe Cam (SCA)	58
Functional characteristics of SCA	58
Application cases of SCA	58
Specifying cam positions	58
Enabling SCA	58
Selecting SCA	58
Warning: Safe referencing	58
Cam synchronization	59
Transfer via PROFIsafe	59
Safely-Limited Position (SLP 1...2)	60
Functional characteristics of SLP 1...2	60
Application cases of SLP	60
Prerequisites for the SLP function	60
Enabling the SLP function	60
No actual value synchronization for SLP	60
Control and status signals for SLP	60
Jumping in the display	60
Control of the Safety Limited Position function using PROFIsafe	61
Extended functions via PROFIsafe	61
Restrictions for PROFIsafe telegram 30	61
Retract	61
Safety commissioning of the retract functionality	61
Activate retract via PROFIsafe	61
Safe Position (SP)	62
Functional characteristics of SP	62
Application cases of SP	62
Enable of the "Transfer of safe position values"	62
Setting of the modulo value for rotary axes	62
Velocity calculation	63
Transfer formats, value range and resolution for telegram 901 (16-bit format)	63
Transfer formats, value range and resolution for telegram 901 (32-bit format)	63
Acceptance test	63
Safe Direction (SDI)	64
Functional characteristics of SDI	64
Application cases of SDI	64
SDI selection and deselection	64

Index:

Section: Drive-integrated safety functions of Sinumerik ONE	Page 37
Safe referencing	66
Referencing through SINUMERIK	66
User agreement	66
No automatic user agreement permitted	66
Referencing through EPOS	66
Referencing through external control	66
Referencing types	66
Initial referencing	66
Follow-up referencing	67
Safe actual value sensing	68
No "Safe Acceleration Monitor" for encoder errors in a 1-encoder system	68
Rules for connecting an encoder	68
1-encoder systems	68
Special features of linear motors	68
2-encoder system	68
2-encoder system on a linear axis via a ball screw	69
2-encoder system on a rotary axis	69
Assignment of the encoder parameters	69
Transfer of the values from the encoder commissioning	69
encoder parameters and the corresponding Safety parameters for 2-encoder systems	69
Encoder types for 1- and 2-encoder systems	70
Encoders with sin/cos 1 Vpp signals	70
Encoders with integrated DRIVE-CLiQ interface	70
Failure Mode Effects Analysis (FMEA)	71
Basic absolute encoders with EnDat interface and additional sin/cos tracks are not permitted	71
Encoder types for SINAMICS HLA	71
Actual value synchronization	71
Position accuracy (r9731)	72
Safe maximum speed (r9730)	72
Forced dormant error detection	72
execution with activation via Safety Control Channel (SCC)	73
Automatic execution for POWER ON	73
STO, SS1, SS1e: Overview of important parameters	74
SS2, SS2e: Overview of important parameters	74
SOS: Overview of important parameters	74
SBC: Overview of important parameters	74
SBT: Overview of important parameters	75
SBR: Overview of important parameters	75
SSM: Overview of important parameters	75
SAM: Overview of important parameters	75
SLS: Overview of important parameters	76
SCA: Overview of important parameters	76
SLP Overview of important parameters	76
SP: Overview of important parameters	77
SDI: Overview of important parameters	77
Safe referencing: Overview of important parameters	78

Notes:

Index:

Section: Manufacturer documentation and function table	Page 79
Manufacturer documentation	79
Machine description	79
Overview of the machine	79
Specifications of the control	80
Configuration plan	80
Function table	81
Operating mode concept in accordance with EN 12417 or EN 13128	82
Safety functions per axis	83
Specifications regarding safety equipment	85
Emergency stop: (HT2 key)	85
Emergency Stop Reset: (SI test rack illuminated key)	85
Unlock protective door: (SI test rack Request key)	85
Protective door switch: (SI test rack)	85
Mode selector switch: (SI test rack keyswitch)	85
Permission Switch: (SI- Testrack)	85
Monitoring of the max. axis speeds or spindle speed: (SLS1...SLS4)	85
Safe software limit switches: (SLP1...SLP2)	86
Z1 axis holding brake: (SBC and SBT)	86
Output for K1 and K2 actuators	86
Test stop	86
Machine control panel MCP398C	86
Safety Integrated test rack	86
Device overview	88
"ET200SP" PLC variable table	88
Circuit diagrams for all safety-relevant devices	89

Index:

Section: Drive-integrated SI functions activation / telegrams	Page 97
Activation via terminals on the NCU and the Motor Module / Power Module	97
Motion monitoring without selection	97
Activation via PROFIsafe	99
Overview of telegrams and process data	99
Telegrams for the control via PROFIsafe and PROFIdrive SIC/SCC	99
PROFIsafe process data PZD 1...6 used depending on the telegrams	100
HMI diagnostics Safety Integrated plus PROFIsafe telegrams	101
Safety Info Channel and Safety Control Channel (SIC/SCC)	102
SIC/SCC parameterization and access to the signals	102
PROFIdrive SIC / SCC process data of telegram 701	102
PLC user interface (DB3x/DB3x00)	102
HMI diagnostics Safety Integrated plus PROFIdrive telegrams	103
Signals to axis/spindle	104
Signals from axis/spindle	105
Furher information	106
Release of the PROFIsafe activation	106
Note: Safety Integrated Basic Functions via PROFIsafe and via terminals	106
Selection of the PROFIsafe telegram	106
ESR behavior if communication failure	106
Time separation between switching operations	106
Ack. via PROFIsafe	107
Additional ack. via PROFIdrive RESET	107
Unexpected restart for safe acknowledgment without MCP Reset	107
Acknowledgment via POWER ON	107
S_ZSW1: Safety status word1 (Basic Functions)	109
S_STW1: Safety control word1 (Basic Functions)	109
S_STW2: Safety control word2 (Extended Functions)	110
S_ZSW2: Safety status word 2 (Extended Functions)	110
S_SLS_LIMIT_A_ACTIVE: Active SLS limit value	112
S_SLS_LIMIT_A: SLS limit value specification	112
S_CYCLE_COUNT: Counter for the safety cycle	112
S_XIST32: Current actual position value (32-bit)	112
S_XIST16: Current actual position value (16-bit)	112
S_ZSW_CAM1: Safe Cam:	113
SIC/SCC (PROFIdrive) process data in the "Axis/spindle signals" interface DB31...DB61	114
S_STW1B Safety Control Channel control word1	114
S_STW3B Safety Control Channel control word3	114
S_ZSW1B: SI Motion / Safety Info Channel status word	115

Notes:

Index:

Section: Drive-integrated SI functions activation / telegrams	Page 97
S_ZSW2B: Safety Info Channel status word 2	116
S_ZSW3B: Safety Info Channel status word 3	116
S_V_LIMIT_B: Safety Info Channel speed setpoint limitation	117
Overview of important parameters	117
PROFIsafe activation	117
SIC/SCC	117

Index:

Section: Furher information	Page 119
Help for standards	119
Overview: SINUMERIK with Safety Integrated Plus	119
Requirements, software and supplementary conditions	119
Drive-integrated safety functions of Sinumerik ONE	120
Drive-integrated SI functions activation / telegrams	120
Configuration of the ET200SP with F-I/O:	121
Safety programming SINUMERIK F-PLC:	121
Commissioning, F-PLC, diagnostics: STO, SS1, SS2, SS2e, SOS, SAM, SSM	121
Acceptance test	121
Appendix D	122

Appendix G: END

Notes:

102

SINUMERIK

SSD and SD card - Data Management

Module Description:

In order to work effectively with the Sinumerik system containing the SSD and SD card, an understanding of the data storage system is required. This module describes the data structure of the internal SSD card as well as the user SD card. The tools which are available and how to use those tools are described in detail as well as a description of the available functions and there usage.
The service stick functionality is described in its entirety.

Module objective:

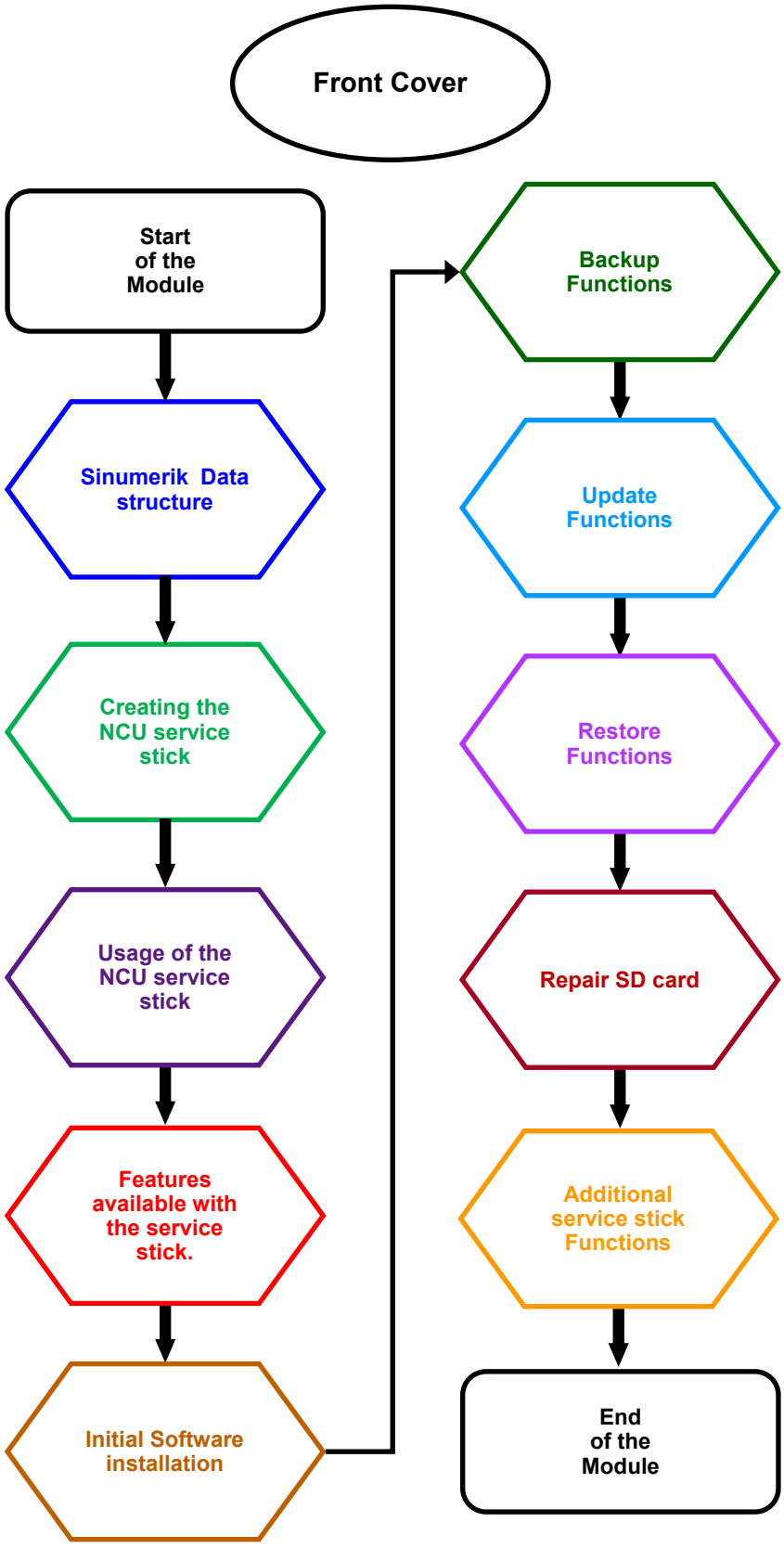
Upon completion of this module you will be able to complete the tasks listed below.

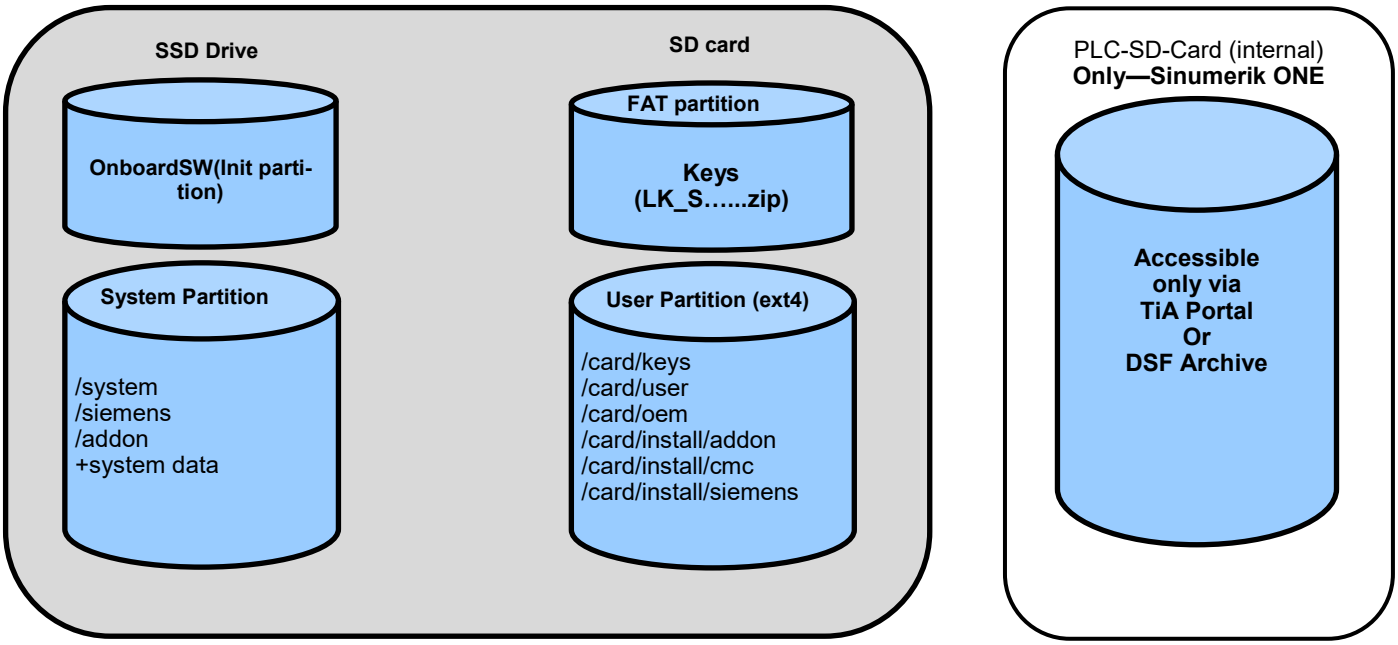
Content:

Sinumerik

- Data structure of NCU
- Creating a service stick
- Usage of the NCU service stick
- Features available with the service stick.
- Initial software installation
- Backup function
- Restoring function
- Update function
- Repair SD card
- Additional service stick Functions

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Sinumerik Data structure
Creating the NCU service stick
Usage of the NCU service stick
Features available with the service stick.
Initial Software installation
Backup Functions
Update Functions
Restore Functions
Repair SD card
Additional service stick Functions
End of the Module



Sinumerik Data structure

The Sinumerik (HMI NCK DRV(PLC 828D only)) runs from an SSD hard drive with a mounted SD card containing user data and system restore points, this user data also includes the license key if the SD card is licensed.
In the diagram above the basic directory structure can be seen on both the SSD and SD data carriers, the Sinumerik system refers to the directories of both mediums with a mounted SD card.

SINUMERIK ONE only:
In addition to the SSD and mounted SD card a second SD card is available for the integrated S7-1500 PLC, this second SD card is only available as PLC memory and can be managed using the TiA Portal and also the DIP switch located on the front part of the Sinumerik, the PLC program can be loaded using the TiA portal or a dsf archive created from within the TiA portal, setup archives made from the controller which include the PLC part can be opened from within the TiA portal.

The Service System is not able to manage or to get access to the PLC SD-Card. All operations done in the Service System will not affect the PLC.

Partition Information

SSD = EXT4 partition
Primary directories

/system
/addon

SD = EXT4 partition
Primary directories

/card/keys
/card/user
/card/oem
/card/install/addon
/card/install/cmc
/card/install/siemens

SD = FAT16 - 24MB

\keys\sinumerik
\user\sinumerik

Notes:

The license is available as a ZIP file on the FAT partition of the SD card and automatically copied to the EXT4 partition at first start-up of the system.

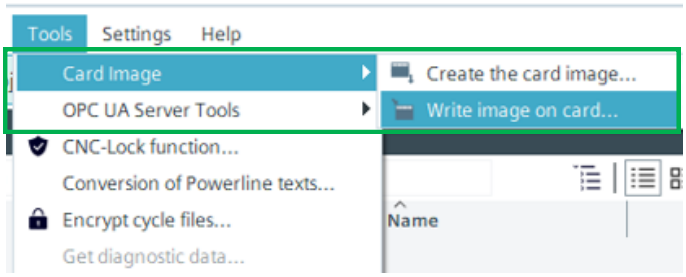
Creating the NCU service stick

Installing the System software is carried out using the Service stick.
A blank =>16GB is required.
This will become a bootable stick containing the service system.
When the system is started from the bootable service stick a selection of menus will be made available for the various service tasks available with the Sinumerik system.

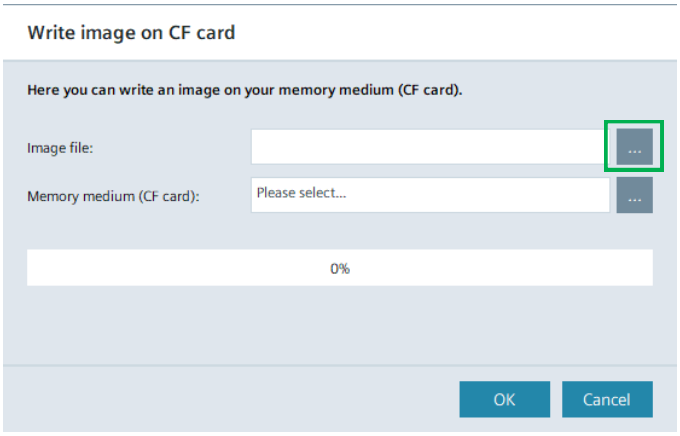
Creating a USB service stick.

Access MyMachine P2P is used to create the service stick.
The image for the stick is delivered in a “zip” file and should be extracted to a suitable location on your computer, once extracted, the file **“service.img”** can be found in the following directory:
SINUMERIK Service System NCU V06_50_00_00
SINUMERIK Service System/service.img

Using “Access MyMachine” ; select “Write image on CF card”, which can be found on the “Tools” menu.

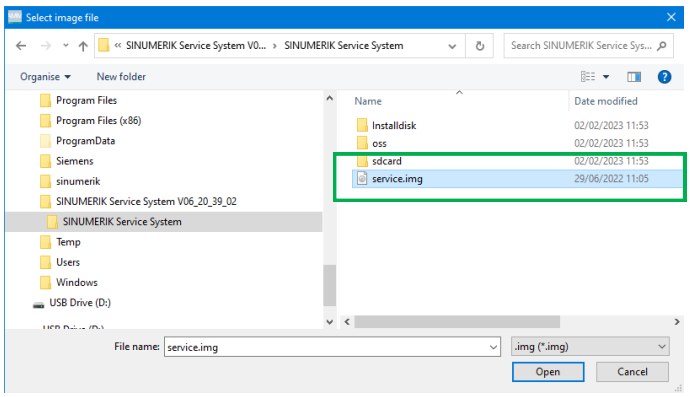


Locate the “service.img” file and select “Open” as follows.



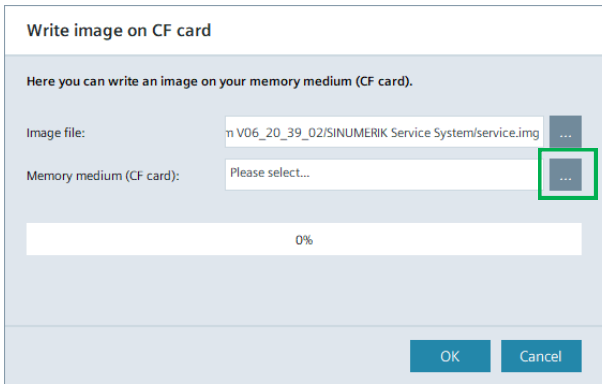








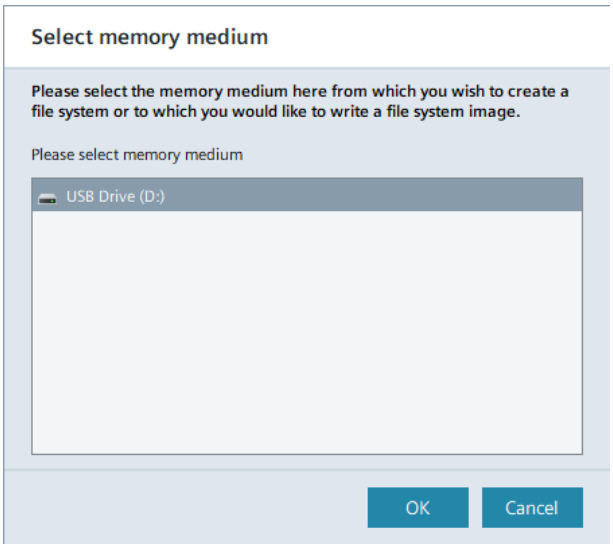
Select the drive letter of the USB stick as follows:



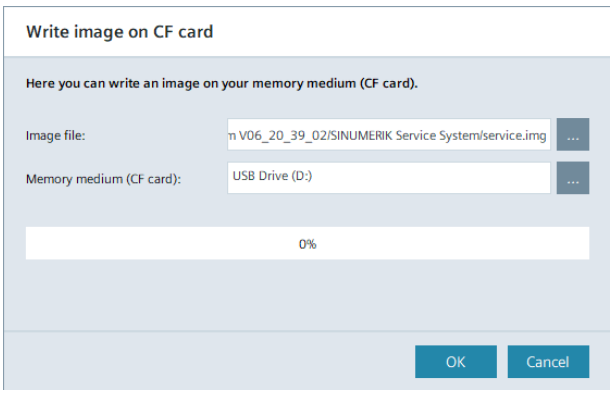




Notes:

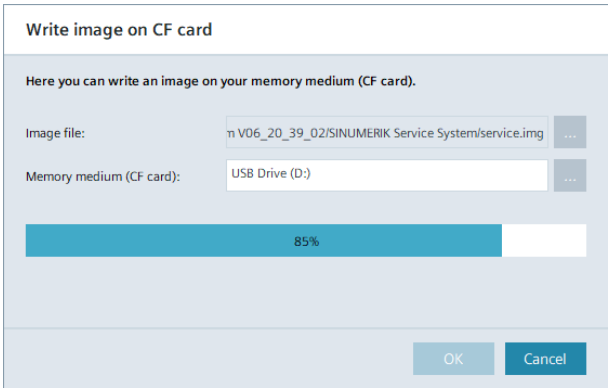


OK

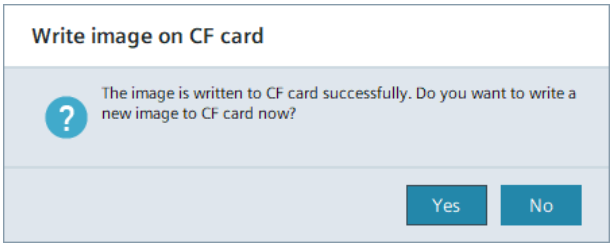


OK

The writing process will begin.



The process can be repeated if required.



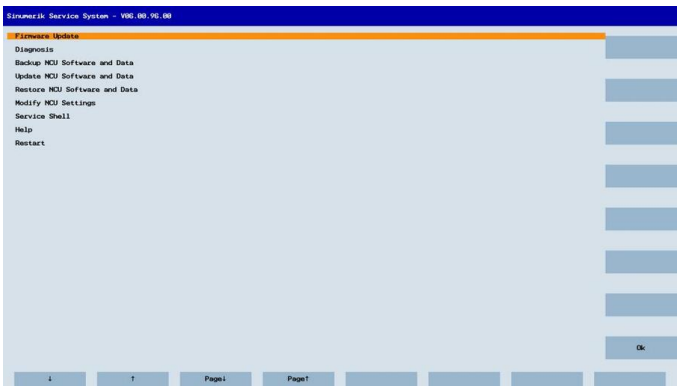
No

Usage of the NCU service stick

The USB service stick needs to have its partition structure expanded in order to use it correctly, this is carried out as follows:

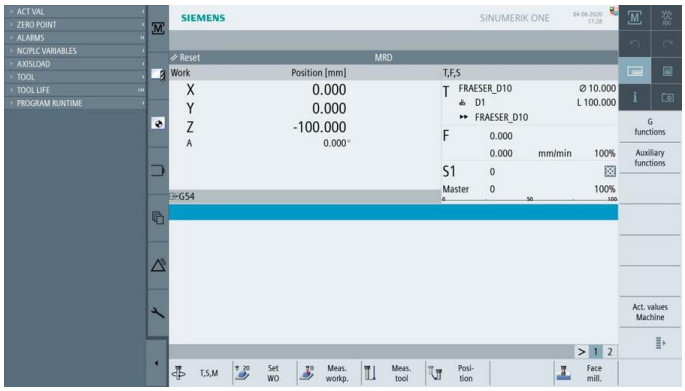
With the NCU switched off, insert the service stick into X125 of the NCU, and place the NC rotary switch in any position (suggestion turn anticlockwise to position “F”) other than 0.

The following screen will be displayed after a successful boot with the service stick.

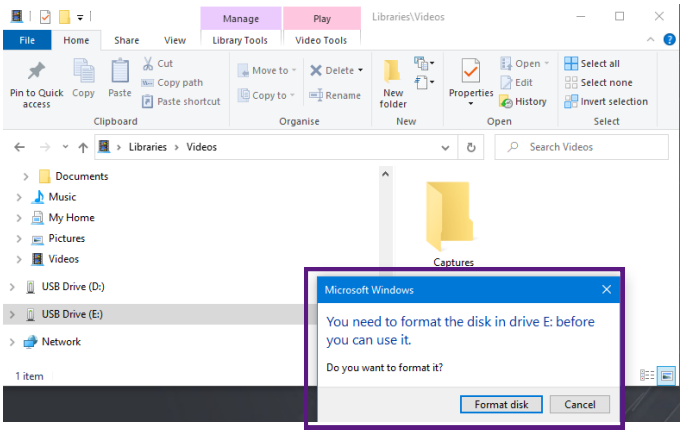


Notes:

Once the service stick has been expanded, switch off the controller, place the NCK rotor switch back to “0” and switch the controller on. The SINUMERIK operate interface should be available, as can be seen below.



The service stick is now having the correct partition sizes and can be used as a fully functional service stick.
If placed in a PC to install system software's be aware that the Linux EXT4 partition will not be recognized and the following message will be seen, on no account should the service stick be formatted or repaired as this will destroy the Linux partition required on the stick for it´s functionality.



Notes:

Rotary switch:
Various positions of the rotary switch are performing various actions, “F” is not having any additional action and is therefore safe to be used for the booting of the service stick.

Functions available with the service stick.

Main Menu Item	Sub menu item	Function (used case)
Firmware Update		Sub-system Bios / FW update
Diagnosis		
	Installed Software versions	
	Check Storage integrity	Scan SSD and SD for errors
Backup NCU Software and Data		
	Backup SSD Storage	
	Backup SD storage	
	Backup User Data	
	Backup Licence key	
	Create System restore point on SD card	
	Connect to Network drive	
Update NCU Software and Data		
	Update System SW and create restore	
	Update System SW without restore point	
	Clear ALL Update System SW and create restore	
	Clear ALL Update System SW without restore point	
	Create System restore point on SD card	
Restore NCU Software and Data		
	Restore Backup from USB stick	
	Restore from System restore point on SD card	
	Clear SD card and install SD image from service sick	
	Restore SD encryption key	Required to unencrypt SD card
	Clear system partition (on SSD)	Init partition to TRUE
	Connect to Network drive	
Modify NCU Settings		
	Change network settings	X120 DHCP Priority
		X130 DHCP Off
	Disable internal HMI	
	Enable SSD Init partition	Init partition to TRUE
	Reset HMI data	Factory settings restored
	Reset NC data	Factors settings restored
	Reset Drive	Factory settings restored
	Secure Disposal	Device end of life - no way back
Service Shell		
Help		
Restart		

Notes:

Initial Software installation

Image SD extract to SSD-SD

In order to install the purchased system software from the SD card to the system (SSD + mounted SD) the following procedure takes place:

The following procedure assumes a new NCU/PU/PPU and a new SD card with an existing License key are being used.
A licensed SD Card is not a prerequisite for this procedure.

With power on:

The “current.tgz” file in SD directory “/card/install/siemens” will be extracted to the SSD and mounted SD card creating the directory structures which can be seen in the diagrams below. The process of extraction can be seen with a cursor rotating around the 7 segment display.

If a license zip file is existing in the FAT16 partition it will be copied to the “/card/keys” directory on the EXT4 partition.

Once the process is complete the Sinumerik operate HMI will be displayed if a HMI is available.

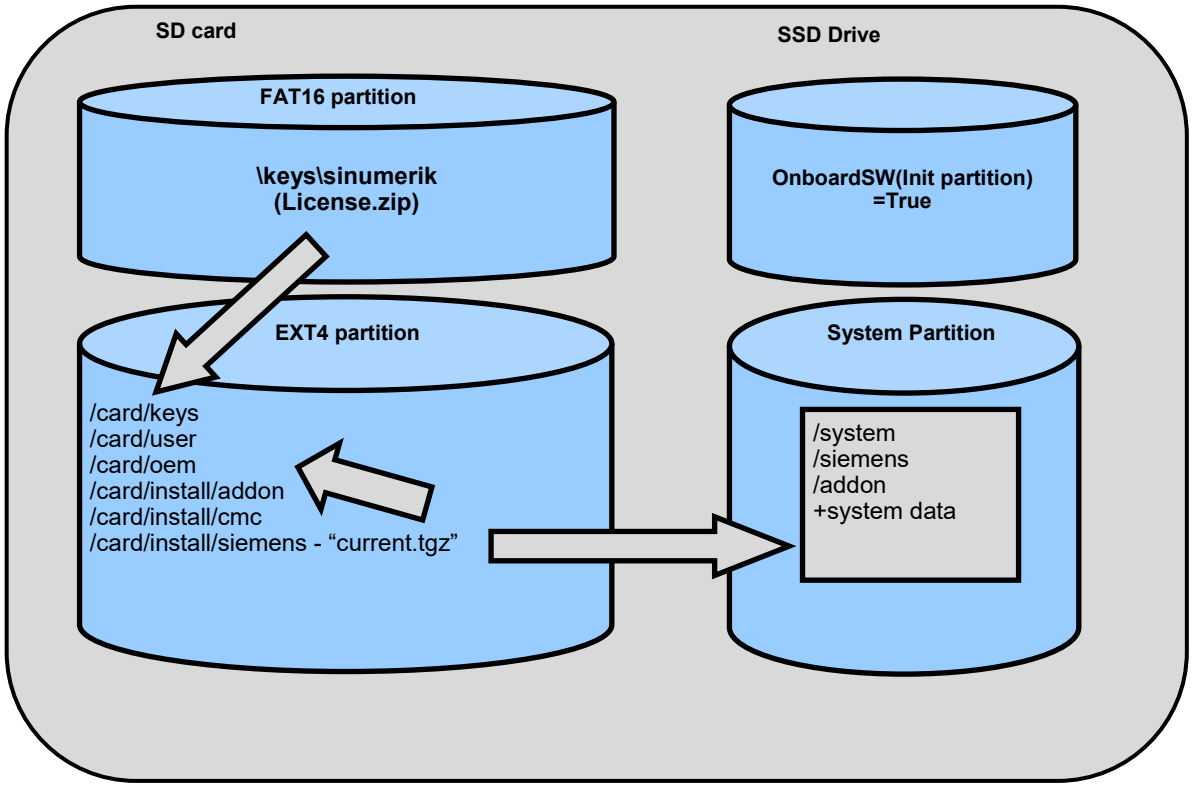
Used case/s / special cases

The following should be carried out when installing the new “current.tgz” SD card image to an already used NCU.

When installing to a previously used NCU it is recommended to Clean the SSD partition first to ensure no errors due to remnant data or partition faults as the update process is a copy/overwrite action and not a clear format action . In order to carry out this action the service menu point: Restore -> clear system partition.

After executing this command you will be advised that a new SSD-Init will take place with the next booting of the NCU where the behaviour will be the same as a new NCU, the current.tgz will be extracted to the SSD from the SD automatically without the necessity to use the service stick.
Upon completion of the action the system should start-up as new.

Forcing a tgz restore from the card can be performed using the SSD-init instruction from the service stick modify menu.



Notes:

Diagnostics

The following features and usage are available on the Diagnostic menu:

Show All network settings
Show network settings for X120
Show network settings for X127
Show network settings for X130

installed software versions
show storage identification
Check storage integrity
Show status
Copy Firmware “ReadmeOSS”

Used case/s / special cases

Check Storage integrity -

The storage integrity of the SD and SDD will be checked and corrected if possible.

Upon completion of the integrity check the partitions are checked and resized if necessary.

Notes:

Backup Functions

Backup SSD Storage

Using the Backup SSD function, a tgz file is saved to the FAT32 partition of the service stick/network drive, this tgz file contains data mainly from the /siemens and /addon directories.

Data which has been saved using this method can be restored using the “restore” menu item of the service stick.

Used case/s / special cases

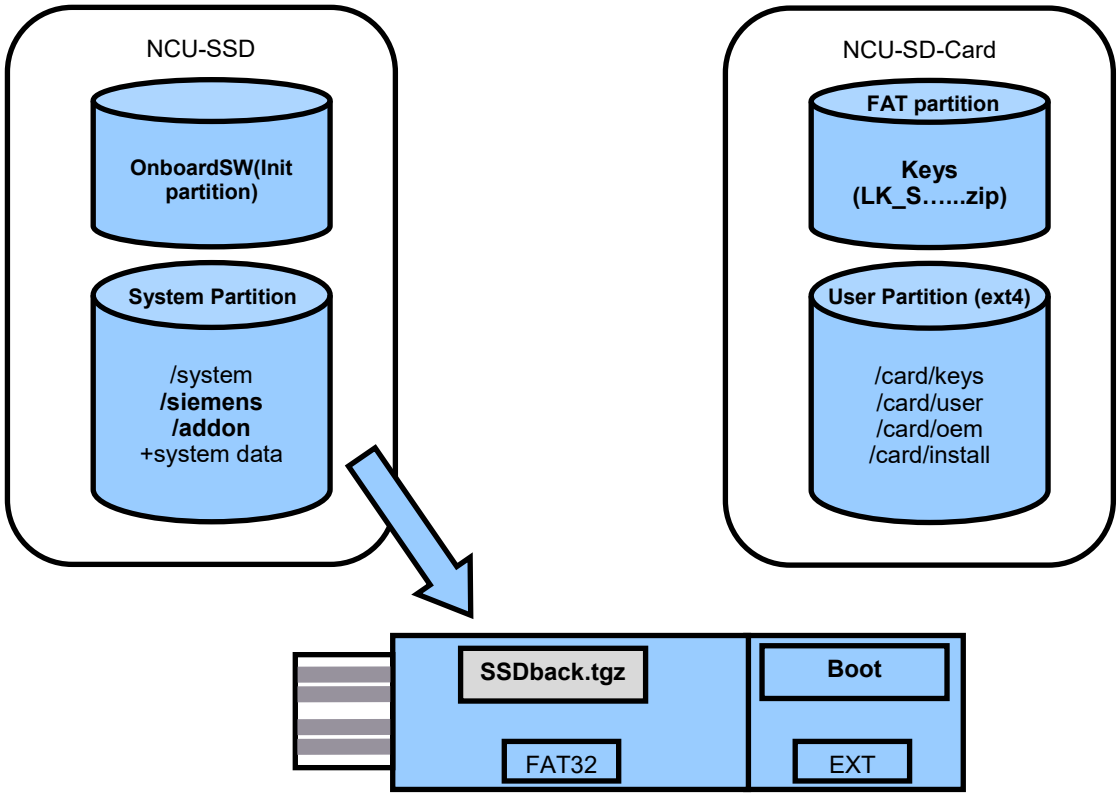
The feature can be used to save a known data set for later use on the same or another SSD drive.

The Backup of the SD card is essential as a data Backup to be left at the machine for later service use cases.

Save DSF/ARD archive of all components in order to completely restore the machine.

Be sure, that the installed System-SW on SSD exists as Restore point on SD-card.

Make an SD card Backup and store it on customer’s data archive at customer site.



Notes:

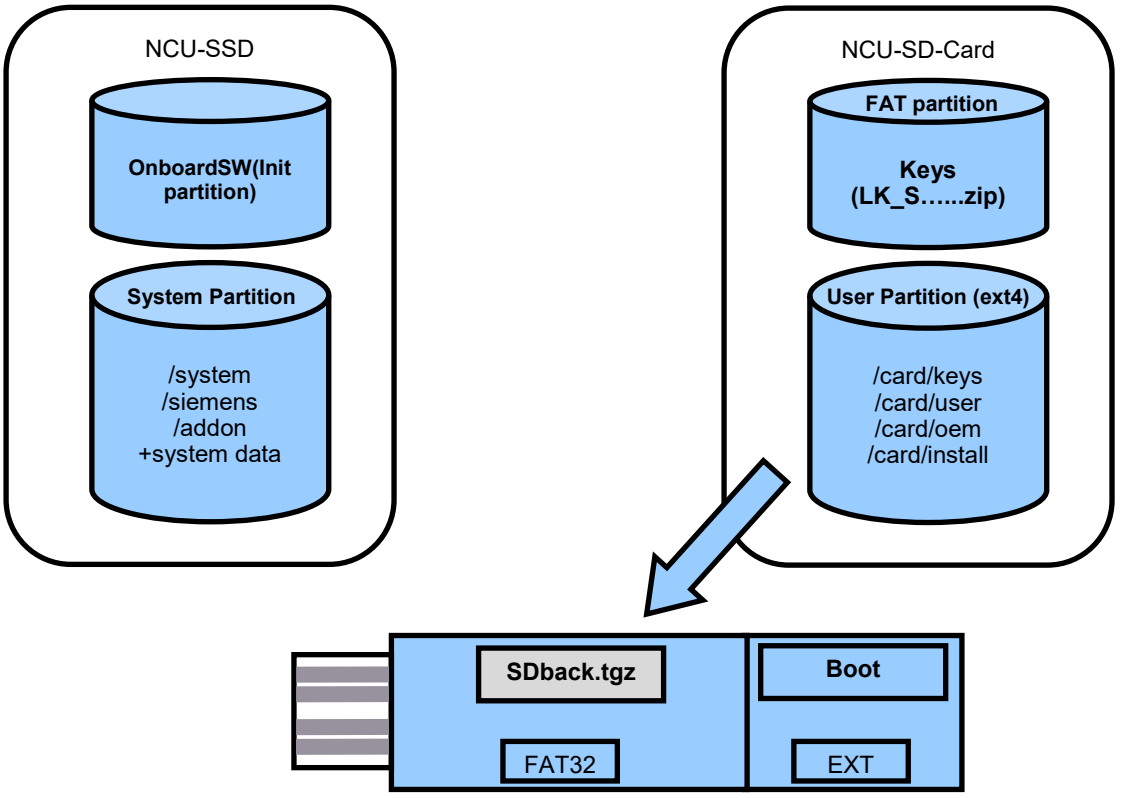
Backup SD Card

Using the Backup SD card function, a tgz file is saved to the FAT32 partition of the service stick or to a network drive, this tgz file contains data mainly from the /oem and /user /install / Keys directories.

Data which has been saved using this method can be restored using the “restore” menu item of the service stick.

Used case/s / special cases

The feature can be used to save a known data set for later use on the same or another SD card.



Notes:

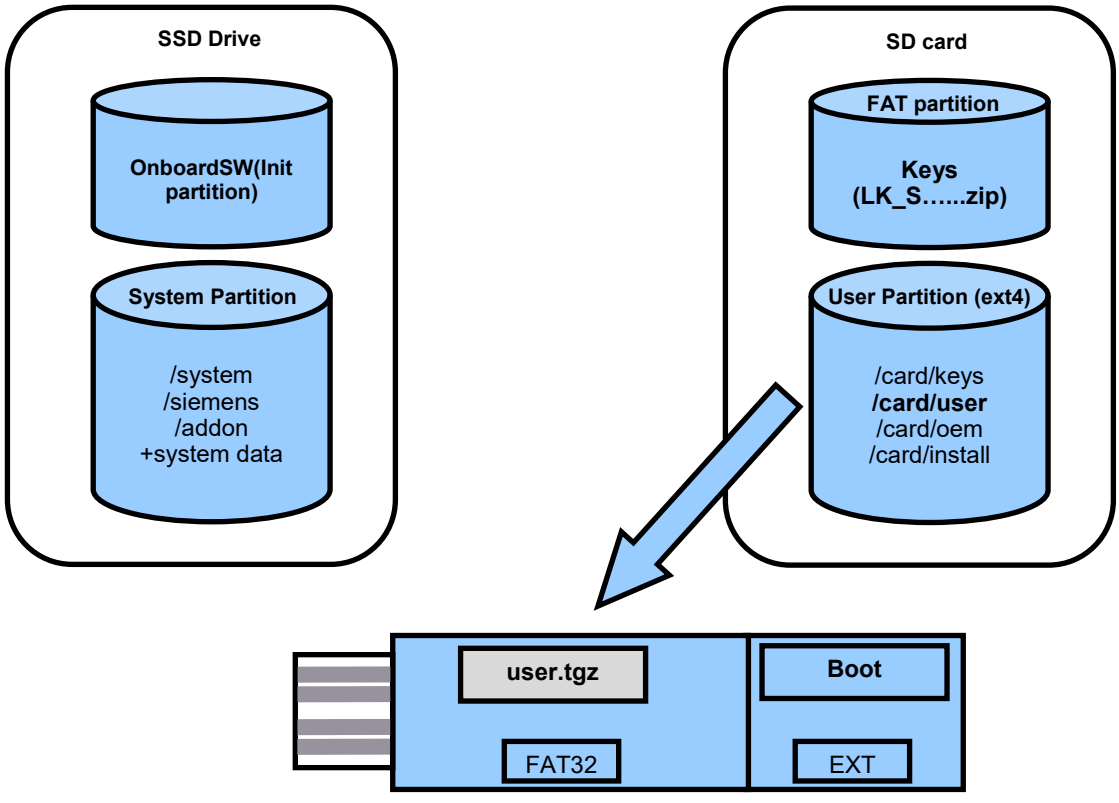
Backup User Data

Using the Backup User function, a tgz file is saved to the FAT32 partition of the service stick or to a network drive, this tgz file contains data mainly from the /card/user directory.

Data which has been saved using this method can be restored using the “restore” menu item of the service stick.

Used case/s / special cases

The feature can be used to save a known data set for later use on the same or another system.



Notes:

Backup License key

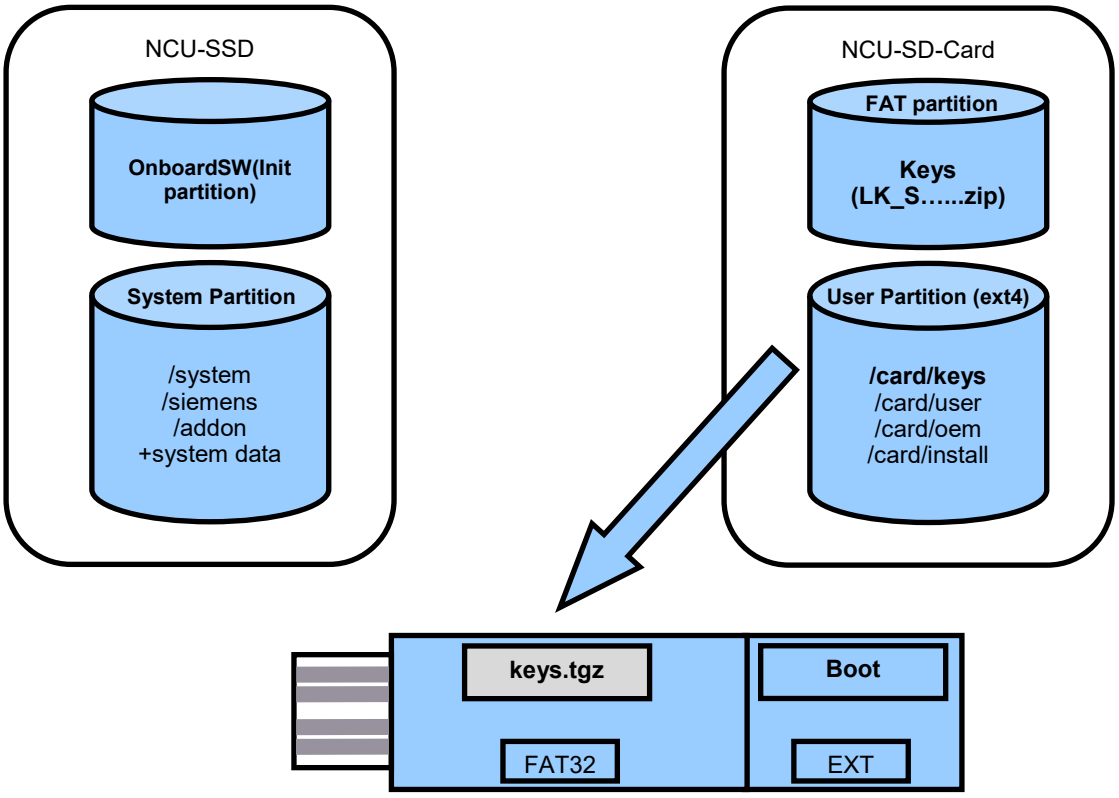
Using the Backup license keys function, a tgz file is saved to the FAT32 partition of the service stick or to a network drive, this tgz file contains the active license key assigned to this SD card.

Data which has been saved using this method can be restored using the “restore” menu item of the service stick.

Used case/s / special cases

The feature can be used to save a known license for later use on the same system.

A license transfer hast to be carried out if the license should be used on another SD card (service SD card replacement).



Notes:

The keys tgz file is stored in the \data\license\SNofSDCard

The license is the associated to the SN of the SD card

Create System restore point on SD Card

Creating a system restore point on the SD card does not save any data to the service stick, only to the SD card.

During the process two tgz files are created and stored on the SD card replacing and previously created restore points or the current.tgz file:

the first tgz is created from the content of the /siemens directory.

The second is created from the content of the /addon

The target storage location is on the mounted SD card in the directories /card/install/siemens and /card/install/addon respectively.

Used case/s / special cases

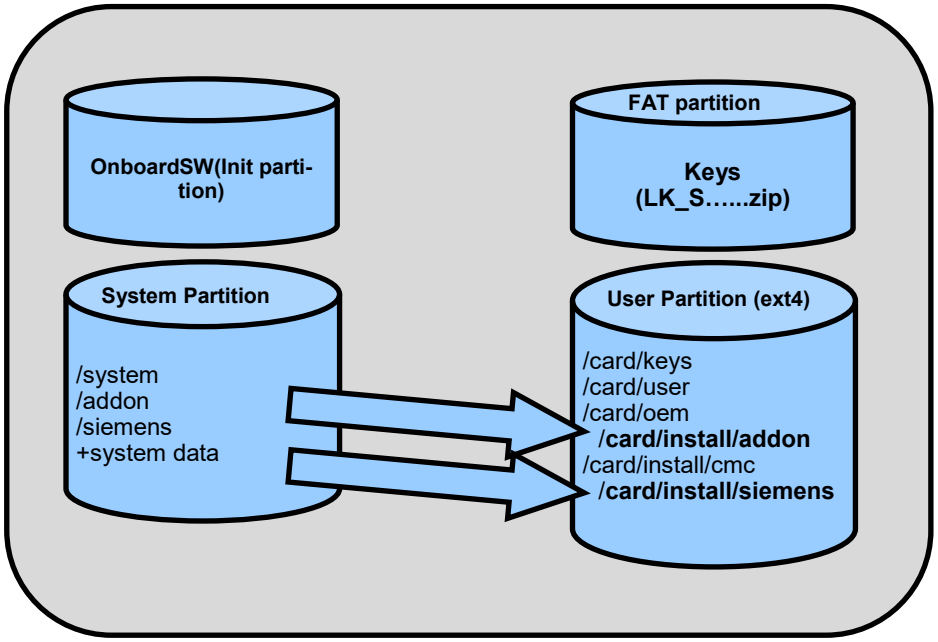
The creation of the System restore point is required when replacing the NCU.

After replacing the NCU:

New NCU will cause the restore file from /card/install/siemens to be extracted to the SSD and mounted SD card followed by the second addon tgz and any other tgz which may be located in the cmc directory from the Machine Tool Builder.

Once this process has finished and the NCU is again running, the complete 2nd commissioning archive has to be read into the controller to ensure a consistent data set.
Prior to the reading in of the backup archive standard data should be loaded by setting switch position 1 on the NCU rotary switch.

The active OnboardSW (init partition) will extract the restore-points into SSD. On a new NCU the init partition is active as default, an NCU which has been already commissioned once, you have to activate using the Sinumerik Service System.



Notes:

Connect to Network drive

Using the Backup / Restore / Update functions the target and source for the data transfer can be defined as the service stick data medium or a connected network drive.

With this menu item you define the :

Hostname/Sharename
Login name

Used case/s / special cases

Please be aware standard passwords may no longer be available.

Notes:

Update Functions

Update System SW and create restore point

Updating the system SW and the creation of a restore point is carried out in two steps.

Prior to the steps taking place the new system software has to be copied to the FAT partition of the service stick.
Once the stick has been prepared the NCU can be started using the service stick connected to USB port X125 with rotary switch at position “F”, once the NCU has started you will see the menu selections and can choose the menu item “update system software and create restore” from the main Update menu.

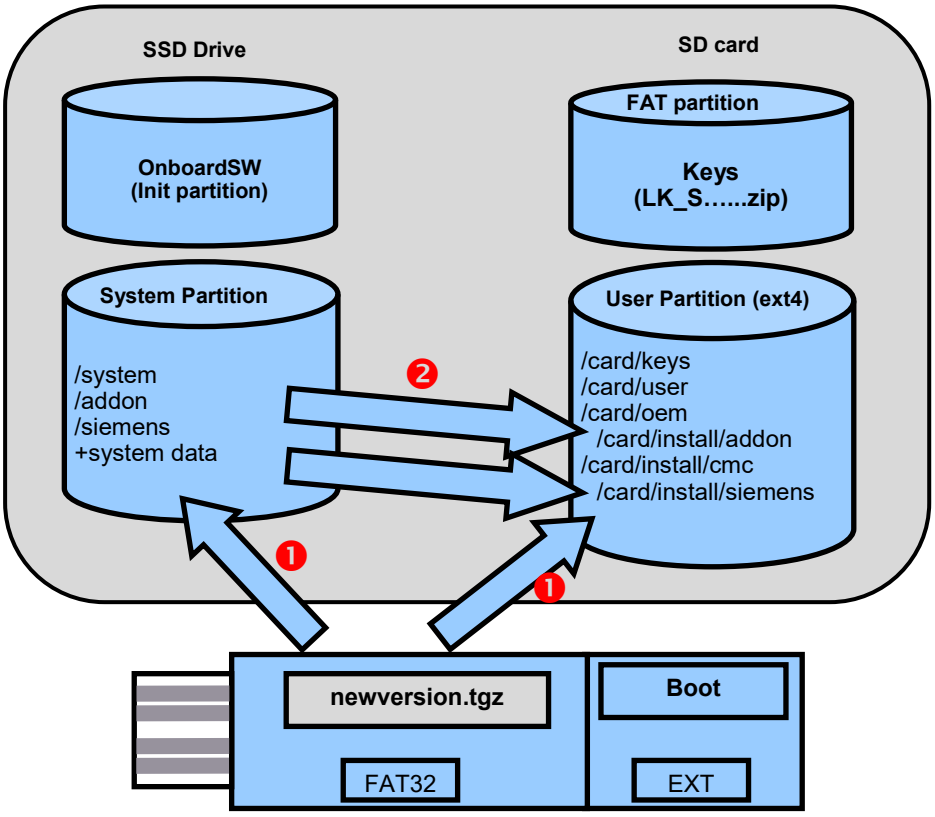
Upon selection you are able to select the new software file (tgz), the installation will follow and can be observed with status information on the screen.

The two steps are shown in the diagram shown below:

1. The new tgz will be extracted to the system SSD+mounted SD, existing files will be overwritten.
2. Restore tgz files are made of the /siemens , /addon and /cmc directories and stored on the SD in the /card/install/siemens, /.../../ addon /../..cmc directories respectively.

Used case/s special cases

- Existing customer data is not affected.
- In order to restore the machine to a running state the 2nd Commissioning archive needs to be read-in to the system.



Notes:

Update System SW without restore point

Updating the system SW without the creation of a restore point is carried out in one step.

Prior to the step taking place the new system software has to be copied to the FAT partition of the service stick.
Once the stick has been prepared the NCU can be started using the service stick connected to USB port X125 with rotary switch at position “F”, once the NCU has started you will see the menu selections and can choose the menu item “update system software without creating a new restore point” from the main Update menu.

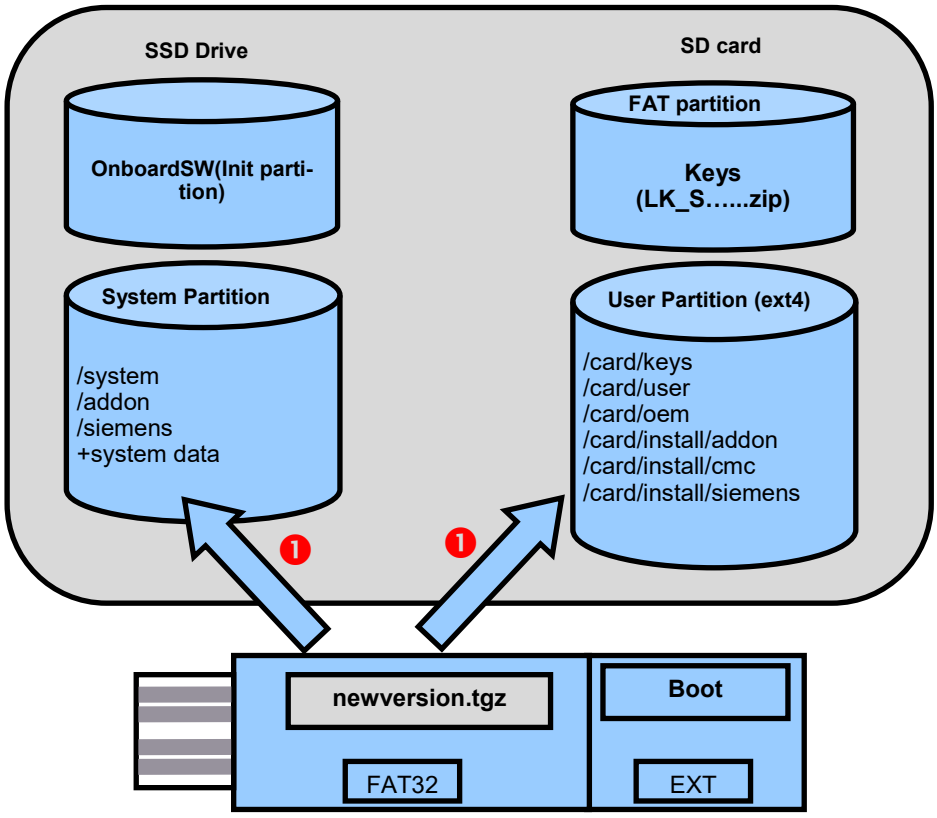
Upon selection you are able to select the new software file (tgz), the installation will follow and can be observed with status information on the screen.

The step required is shown in the diagram shown below:

1. The new tgz will be extracted to the system SSD+mounted SD, existing files will be overwritten.

Used case/s / special cases

- Existing customer data will not be affected.
- No restore files will be created
- In order to restore the machine to a running state the 2nd Commissioning archive needs to be read-in to the system.
  Prior to the reading in of the backup archive standard data should be loaded by setting switch position 1 on the NCU rotary switch.
- The original tgz restores if available can be used to restore the machine to it´s original state prior to the SW update.



Notes:

The existing restore points in the /card/install/siemens and /card/install/addon will not be overwritten.

Old data still available as rollback.

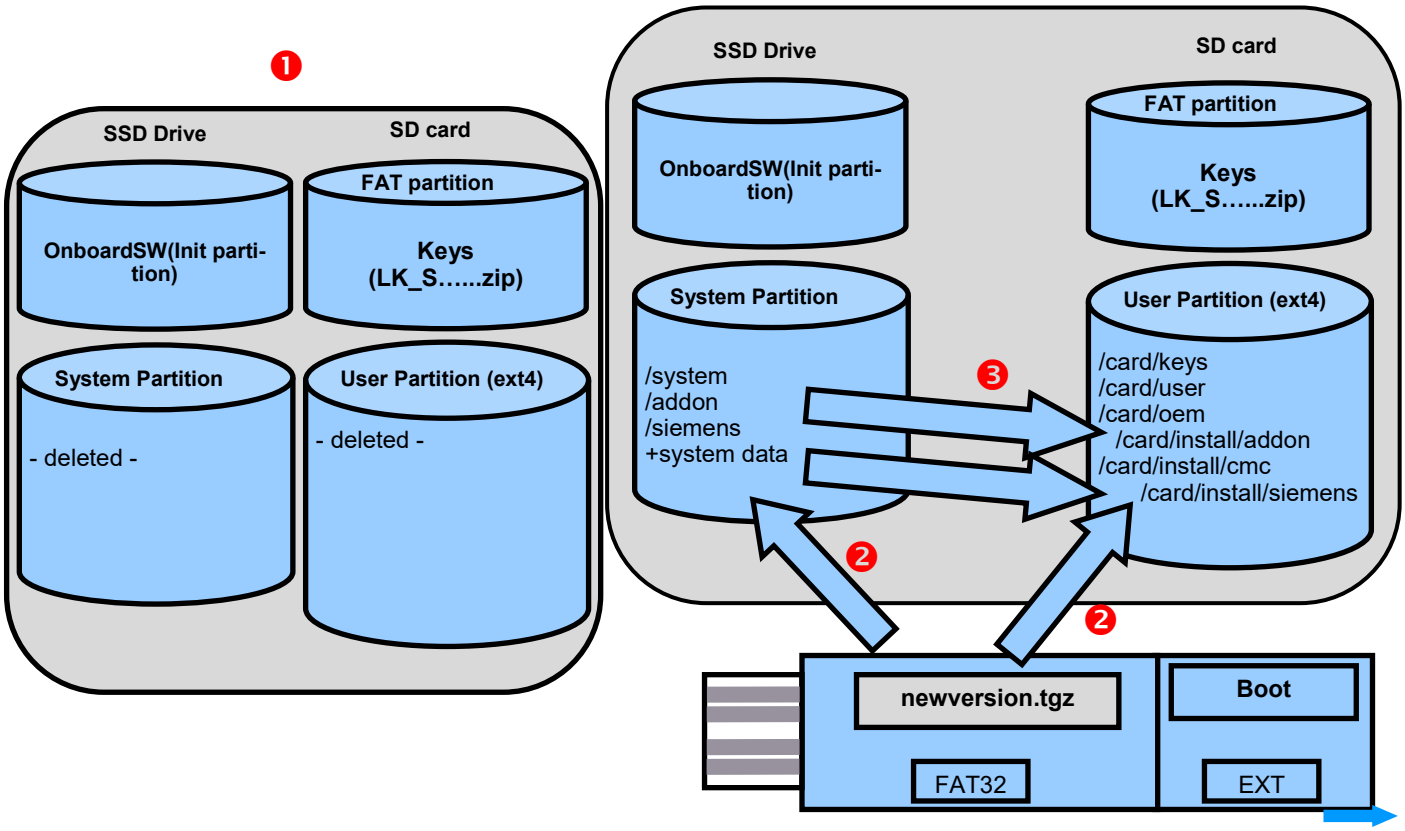
Clear All + reinstall System SW create restore point

Clear All + reinstalling the system SW and the creation of a restore point is carried out in three steps.
Prior to the steps taking place the new system software has to be copied to the FAT partition of the service stick.
Once the stick has been prepared the NCU can be started using the service stick connected to USB port X125 with rotary switch at position “F”, once the NCU has started you will see the menu selections and can choose the menu item “clear all + reinstall system software and create restore” from the main Update menu.
Upon selection you are able to select the new software file (tgz), the installation will follow and can be observed with status information on the screen.
The two steps are shown in the diagram shown below:

1. All content from the SSD and Mounted SD will be deleted.
2. The new tgz will be extracted to the system SSD+mounted SD.
3. Restore tgz files are made of the /siemens and /addon directories and stored on the SD in the /card/install/siemens and /.../../addon direct -ories respectively.

Used case/s / special cases

- Existing customer data will be deleted (OEM and user).
- Two tgz files are created for both the /siemens and / addon directories.
- Due to the Clear ALL the machine may not be runnable after restoration of the archive, due to missing customer files on the data medium.
- Clean system and remove unwanted customer data.



Notes:

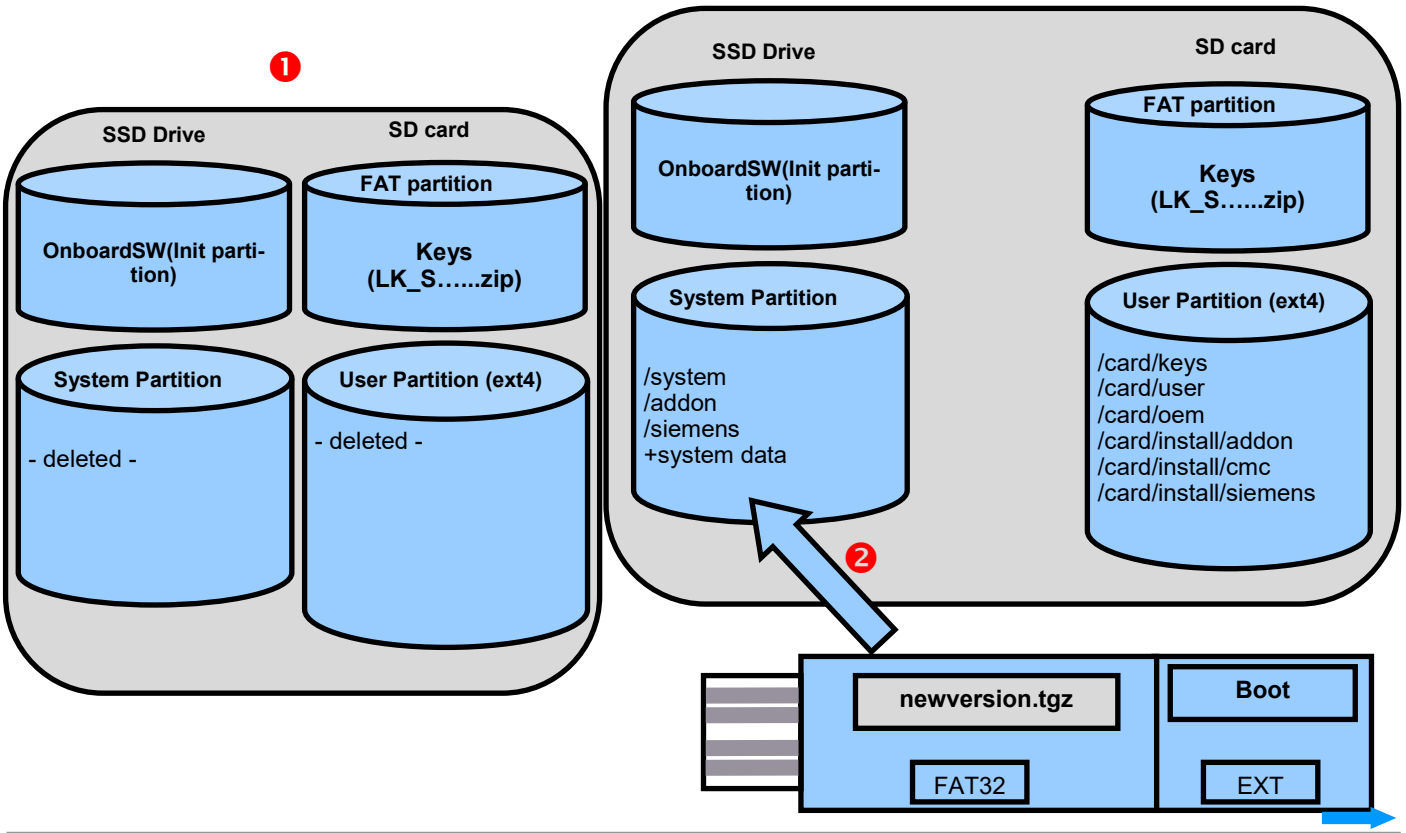
Clear All + reinstall System SW without restore point

Clear All + reinstalling the system SW and without creation of a restore point is carried out in two steps.
Prior to the steps taking place the new system software has to be copied to the FAT partition of the service stick.
Once the stick has been prepared the NCU can be started using the service stick connected to USB port X125 with rotary switch at position “F”, once the NCU has started you will see the menu selections and can choose the menu item “clear all and reinstall system software without restore point” from the main Update menu.
Upon selection you are able to select the new software file (tgz), the installation can be observed with status information on the screen.
The two steps are shown in the diagram shown below:

1. All content from the SSD and Mounted SD will be deleted.
2. The new tgz will be extracted to the system SSD+mounted SD.

Used case/s / special cases

- Existing customer data will be deleted.
- Clean system and remove unwanted customer data.
- In the case of a Clear ALL the existing restore points on the SD card card will be deleted which means that this SD card cannot be used for an initial install with a new NCU due to the missing tgz “current.tgz”!, Due to the missing restore data the SD card is having 1GB more usable space if required.



Notes:

Create System restore point on SD Card

Creating a system restore point on the SD card does not save any data to the service stick, only to the SD card.

During the process two tgz files are created and stored on the SD card replacing and previously created restore points:

the first tgz is created from the content of the /siemens directory.

The second is created from the content of the /addon directory and some additional directories from the SSD drive.

The target storage location is on the mounted SD card in the directories /card/install/siemens and /card/install/addon re-spectively.

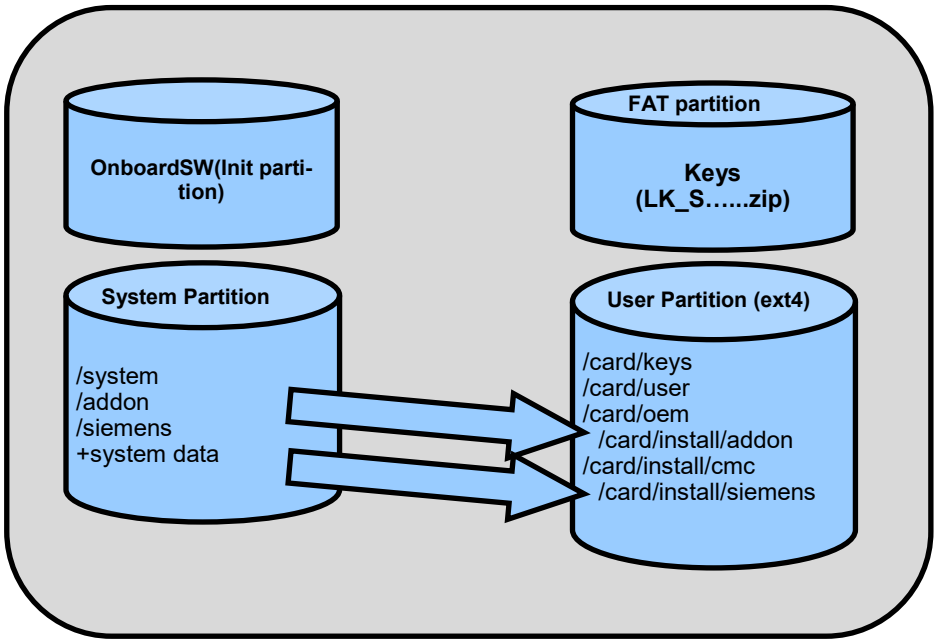
Used case/s

The creation of the System restore point is required when replacing the NCU.

After replacing the NCU:

New NCU will cause the restore from /card/install/ siemens to be extracted to the SSD and mounted SD card followed by the second addon tgz and any other tgz which may be located in the cmc directory from the Machine Tool Builder.

Once this process has finished and the NCU is again running, the complete 2nd commissioning archive has to be read into the controller to ensure a consistent data
Prior to the reading in of the backup archive standard data should be loaded by setting switch position 1 on the NCU rotary switch.



Notes:

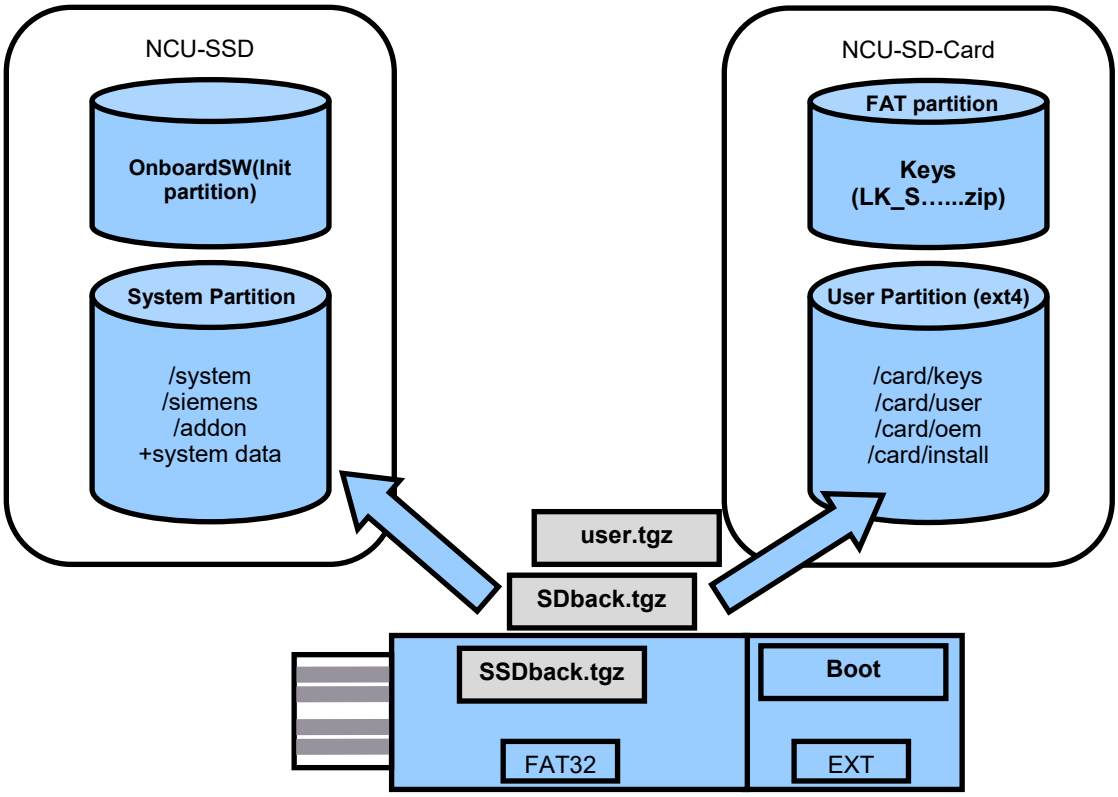
Restore Functions

Restore Backup from USB Stick

Using the Restore backup function, a tgz file on the USB service stick is restored to it´s original location on the data medium (SSD + mounted SD).

Used case/s

The feature can be used to restore a known data set.

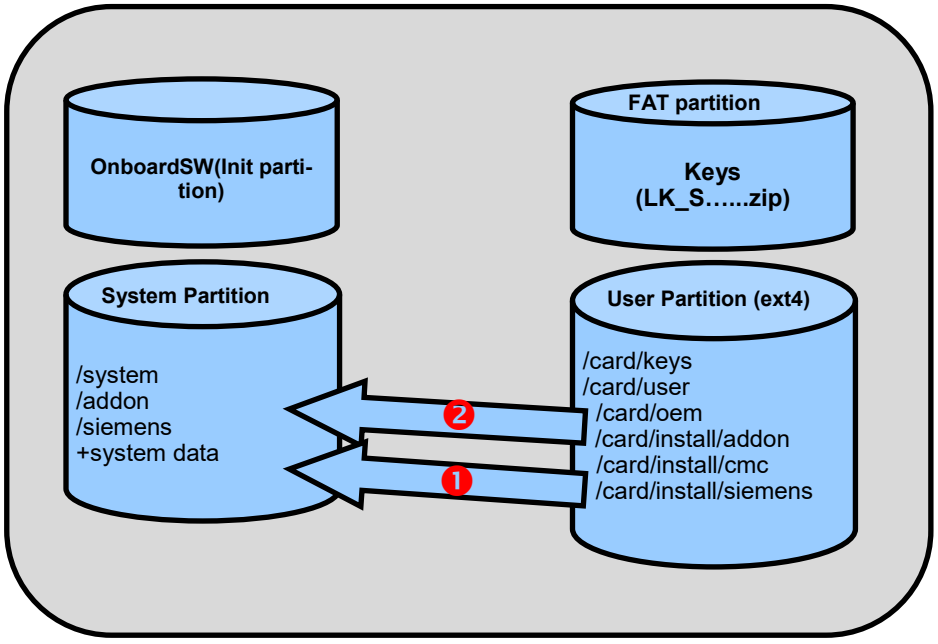


Notes:

Restore from System restore point on SD Card

If restore points are available in the /card/install/siemens and /.../.../addon directories they will be extracted to their original locations on the destination data medium, the extraction will begin with the tgz file located in the siemens directory then followed by the tgz file in the addon directory.
Any existing files will be overwritten.

Used case/s / special cases



Notes:

Clear SD Card and install SDCard image from Service stick.

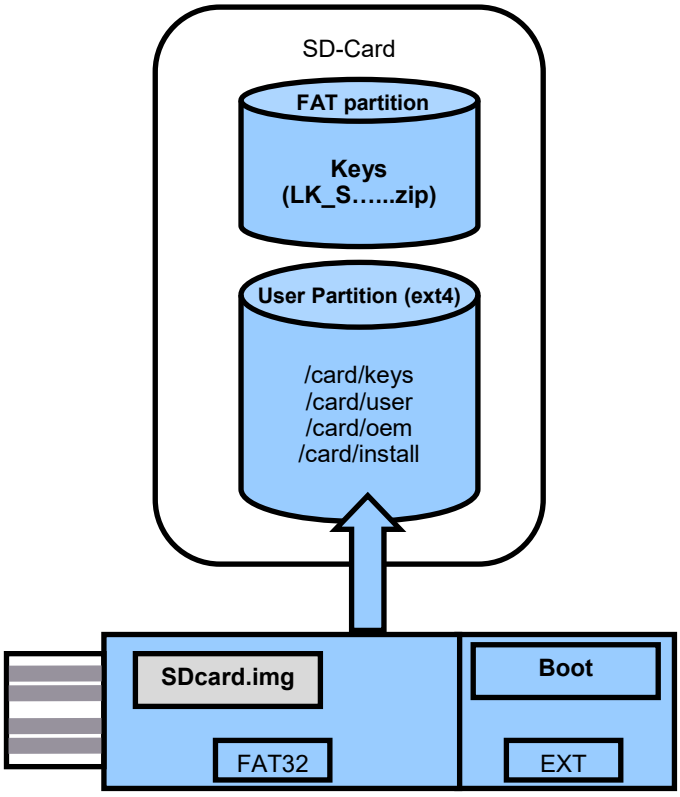
Using this feature the SDCard.img file located on the service stick will be written to the SD Card.

This action is a prerequisite to using the function Restore (SDback.tgz) of a previously made SD card Backup.

Used case/s / special cases

Any existing data on the SD card will be lost.

repairing a corrupt directory structure on the SD card.



Notes:

Additional service stick Functions

Disable internal HMI

When using HMI Operate on an IPC it is recommended to disable the internal (Linux based) HMI to prevent confliction when two HMI systems are running side by side and trying to access the same data areas.

Example

MDA Buffer can only be active for one HMI.

Enable SSD Init partition

Enable SSD init partition is available on the restore menu of the service stick.

When this is enabled the restore from the SD card /card/install/siemens and /.../../addon will be extracted to the system data medium (SSD + mounted SD card)
This process can be monitored at the next NCU start through a cursor rotating around the seven segment display on the NCU.

Restore SD card encryption key

The encryption key used to encrypt the SD card in security settings will be restored in the case of a system restoration.

Clear system partition on SSD and set init partition TRUE

Enable SSD init partition is available on the restore menu of the service stick.

Carrying out this function will put the SSD back to its factory state, Init partition will be set to TRUE.

Reset HMI - NC - Drive data

The relevant data will be deleted from the SD card (Drectories /card/user and /card/oem are affected).

Service Shell

With login name and password you can enter the SC (service commands) directly at the command prompt.

For a list of the available commands you can type “sc help” at the command prompt.

102: END

Notes:

Notes:

A203

SINUMERIK ONE

Preparation of system software for NCU and IPC

Module Description:

Before the commissioning of a machine can take place, the NCU and the IPC need to be in a running state.
This module discribes the nessecery steps to install the system softwares on both items of hardware.

Module objective:

Upon completion of this module you will be able to perform the following task:

- Install the system softwares on both NCU and IPC.

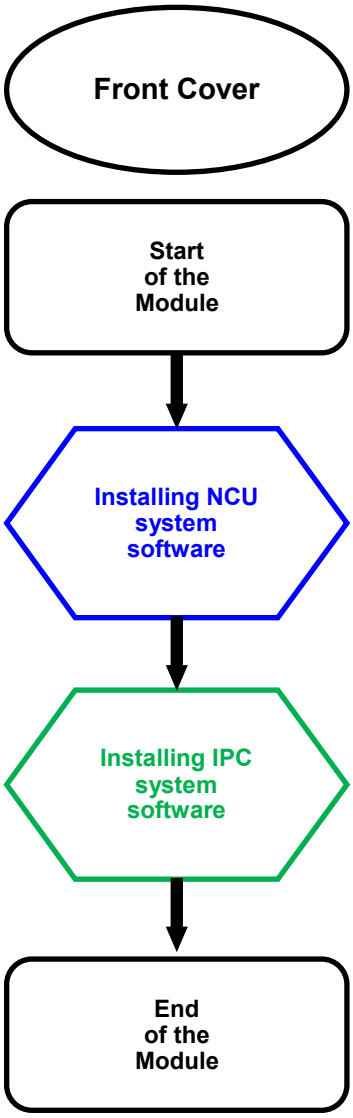
Content:

Installing NCU system software

Installing IPC system software

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

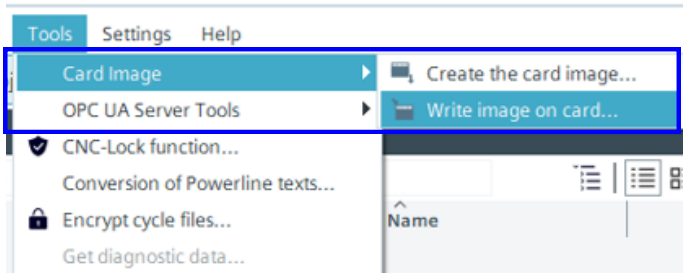

Front Cover
Start of the Module
Installing NCU system software
Installing IPC system software
End of the Module

Installing NCU system software

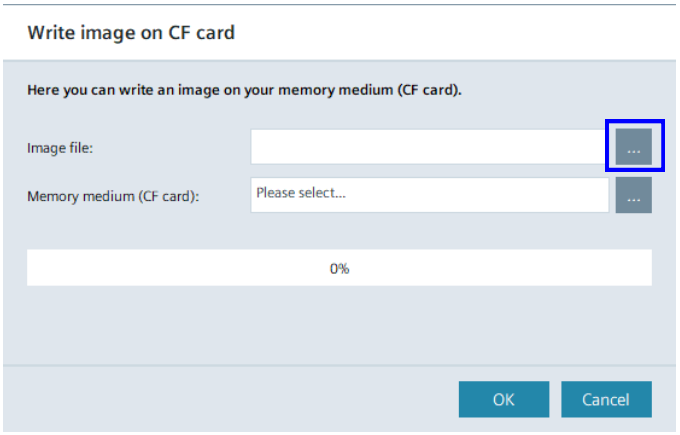
Installing the System software onto SD card is carried out by using the “Service System” software.
A USB service stick is required. This is a bootable stick which contains Service System software.
When plugged into interface X125 the system will boot from the USB device into the “Service” stick system menu.
From this menu it is possible to install the CNC software file.
.

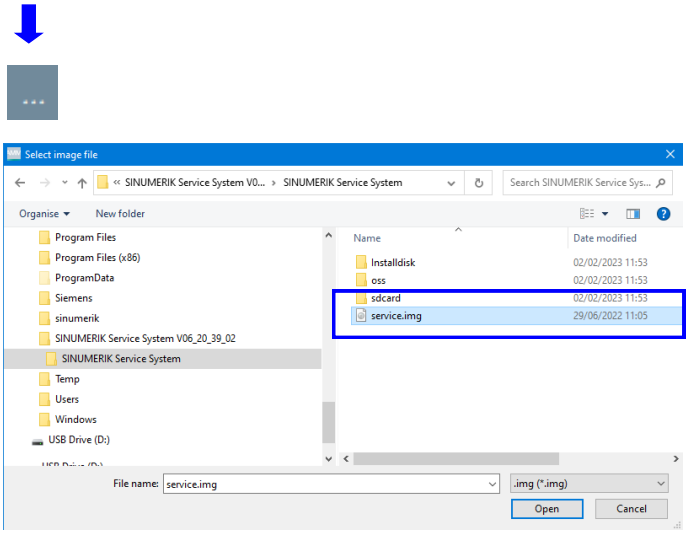
Creating a USB service stick.
Access MyMachine is used for creating a USB service stick.
The software is delivered in “zip” format and should be extracted to a suitable place on a computer.
When extracted, the file “service.img” can be found in the following path:
SINUMERIK Service System NCU V06_60_00_00/
SINUMERIK Service System/service.img

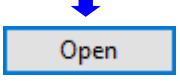
Run the Access MyMachine and select “Writing image to CF card” found on the “Tools” menu.



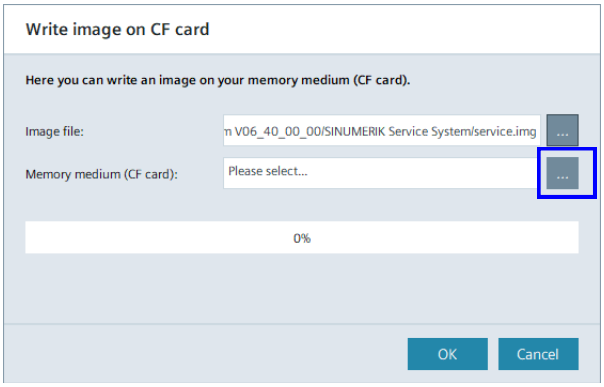
Locate the “service.img” file and select “Open”



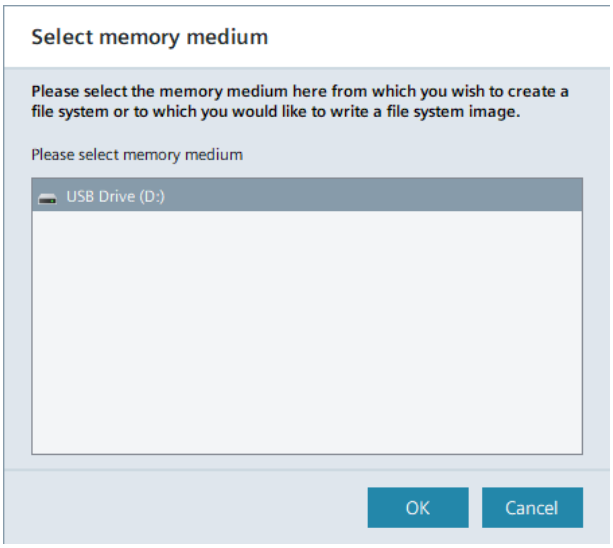




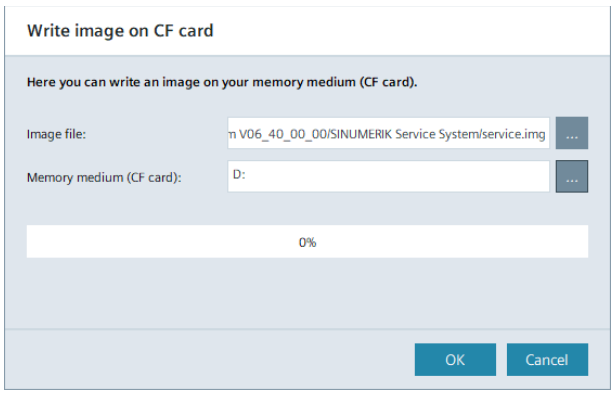
Select the drive letter of the USB stick.



Notes:

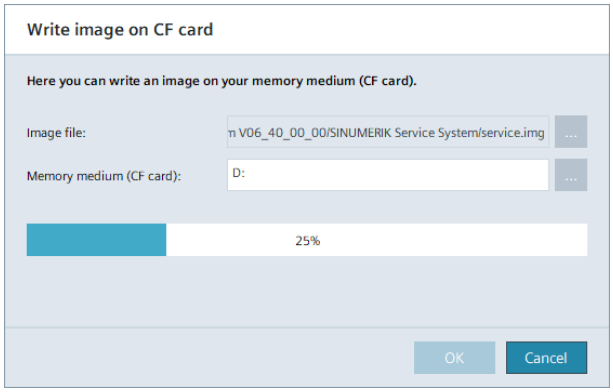


OK

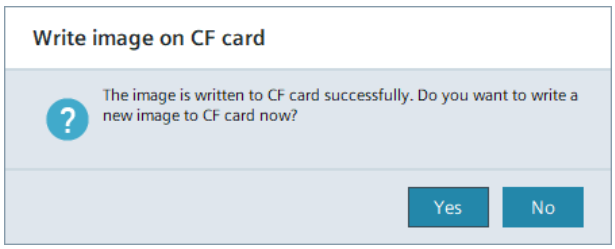


OK

The writing process will begin.



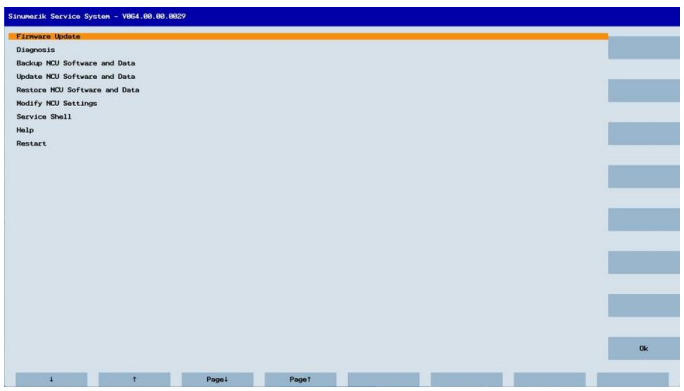
The process can be repeated if required.



No

Starting the Service System.

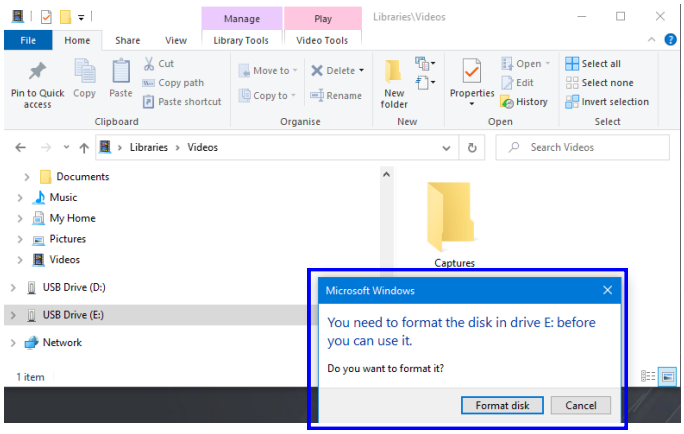
Switch off the controller and insert the bootable USB stick into X125 of the NCU, and place the NCK rotary switch in any unused rotary position eg. F.
Power ON the controller.
The following screen will be displayed:



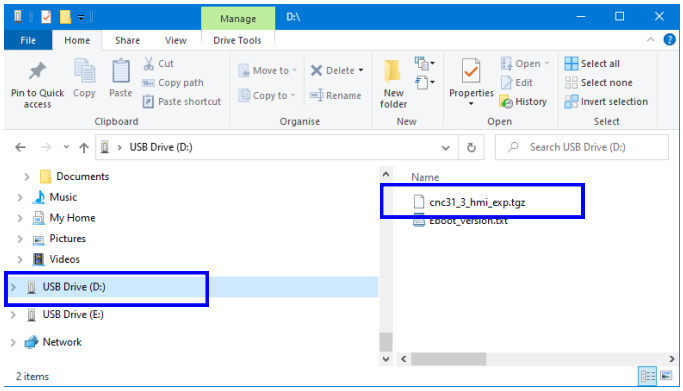
By powering up the boot stick in the NCU, the correct partitions are created .
Power OFF the NCU box, and view the boot stick in Windows Explorer.

Notes:

The boot stick in Windows Explorer, will show two drives. Drive E:/ is the Linux boot partition and MUST NOT be formatted.



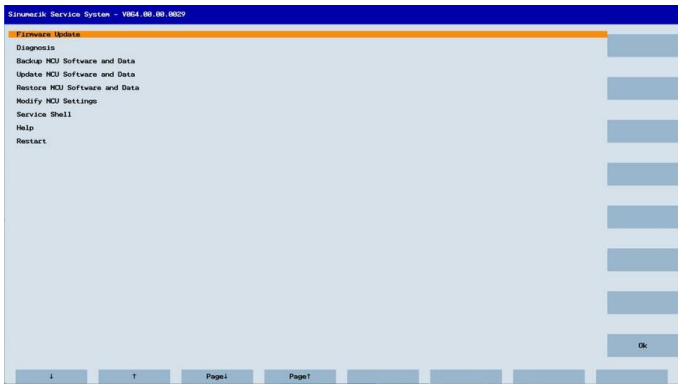
Drive D:/ is the partition where the system software *.tgz file must be transferred to.



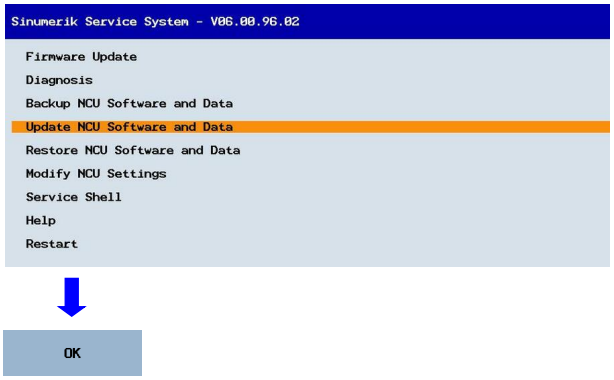
Installing the system software (TGZ file) to the SD card of the NCU

Starting the Service System.

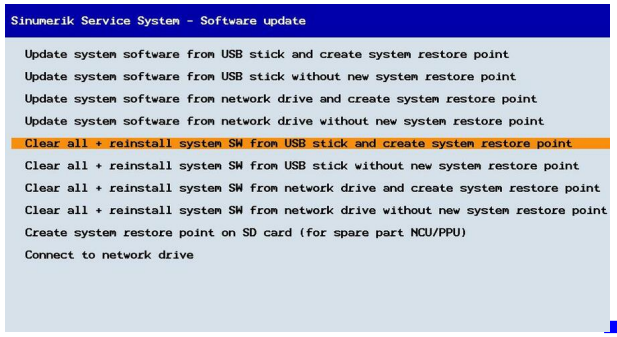
Switch off the controller and insert the bootable USB stick into X125 of the NCU, and place the NCK rotary switch in any unused rotary position eg. F.
Power on the controller
The following screen will be displayed:



Cursor to the “Update NCU Software and Data” menu item and select “Ok”



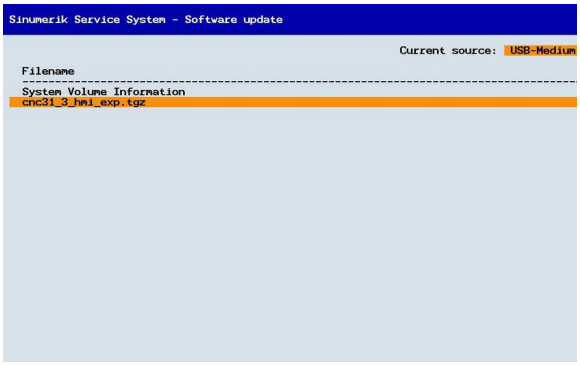
A number of “reinstall” choices are available, select the menu item “Clear all + reinstall system SW from USB stick and create system restore point”



Notes:

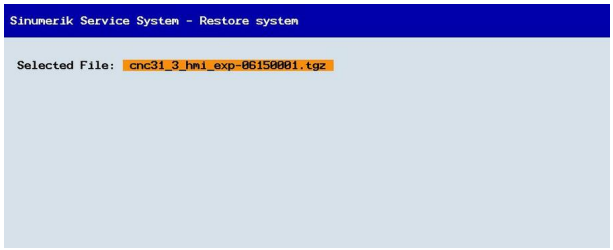
OK

Select the system software file, in this case “cnc31_3_hmi_exp.tgz”.

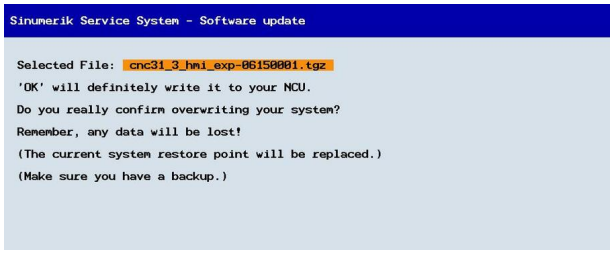


OK

The procedure will begin.

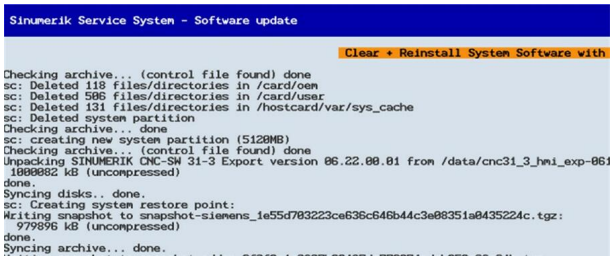


OK

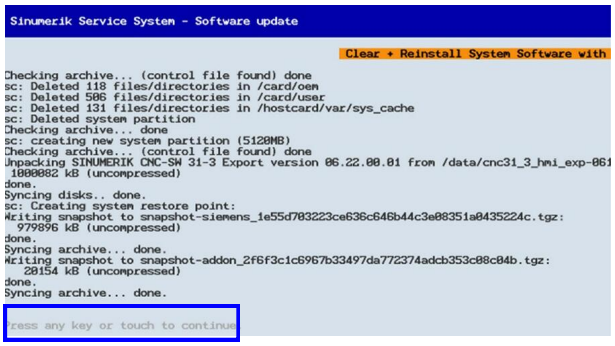


OK

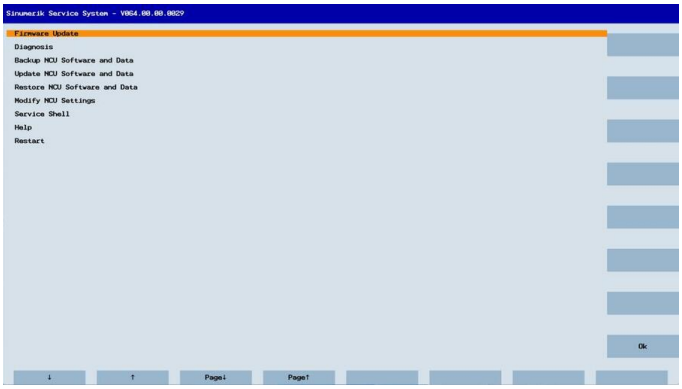
The Installation procedure will begin.



On completion, press any soft-key.

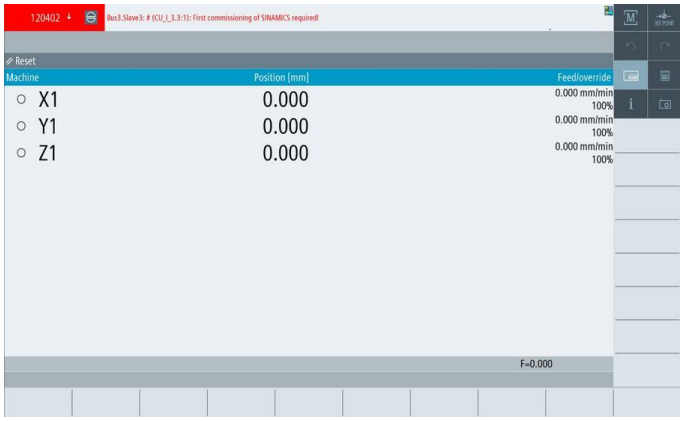


The original screen will be displayed.
The USB stick should be removed and the system rebooted.



Switch off the controller and remove the bootable USB stick from X125 of the NCU, and place the NC rotary switch into position 0, switch on the NCU.

Notes:



The system software running from the NCU,
Before commissioning can take place:

Standard machine data needs to be loaded.
PLC area needs to be cleared.

Set the password.

Set the time and date.

Notes:

Prepare IPC MicroBox

Hardware required:

PC monitor with displayPort cable
(screen with a resolution of 1920 x 1080, this is to match the resolution of the SIEMENS Industrial Thin Client ITC V3)

USB keyboard

USB mouse

Software required:

IPC Boot stick (included in delivery package of IPC)

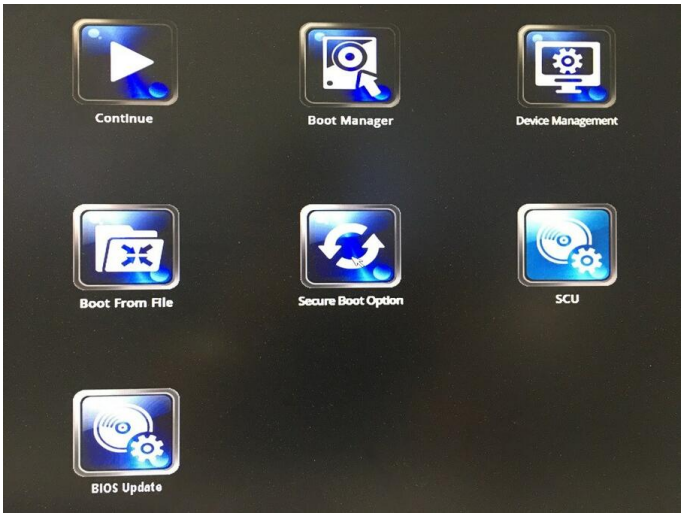
PCU-Basesoftware WIN10 IPC_MCU V15_0 SP3 HF1

SINUMERIK ONE HMI IPC V6.24

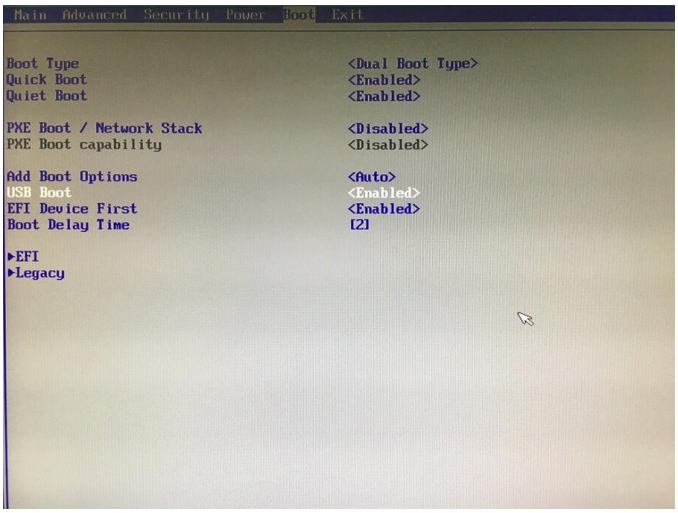
NOTE:

An IPC boot stick can be found in the brown A4 envelope included in the delivery.
The use of this boot stick is mandatory, to update WINDOWS 10 to the latest version of OS required for the IPC, and the installation of SINUMERIK ONE

Connect the PC monitor (using a “DiplayPort cable“), along with a mouse, and keyboard, to the IPC Micro box.
Insert the IPC boot stick into the IPC, and power on the IPC box.
During the self-test, the following message appears:
`Press Esc for Boot Options`
Press Esc on keyboard
select “SCU“

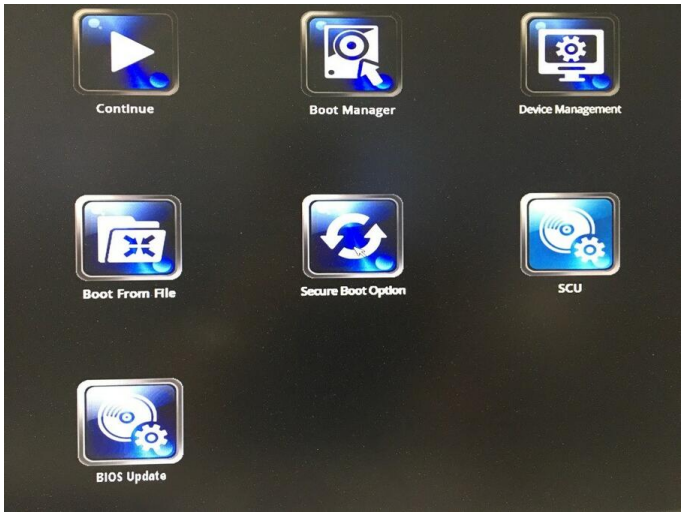


Enable “USB BOOT“
Press F10 (save and exit)
Power off / on

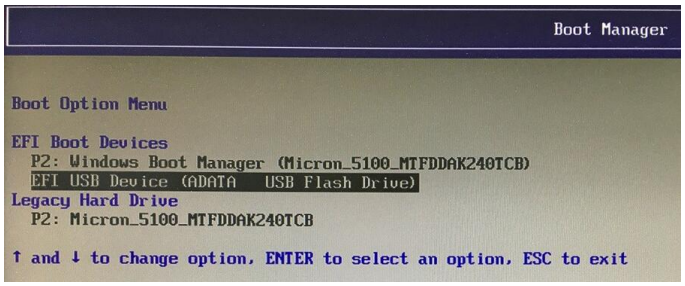


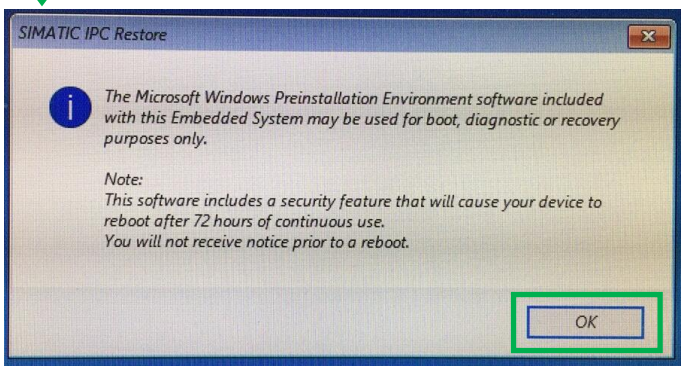
Notes:

During the self-test, the following message appears:
`Press Esc for Boot Options`
Press Esc on keyboard
select “Boot Manager“

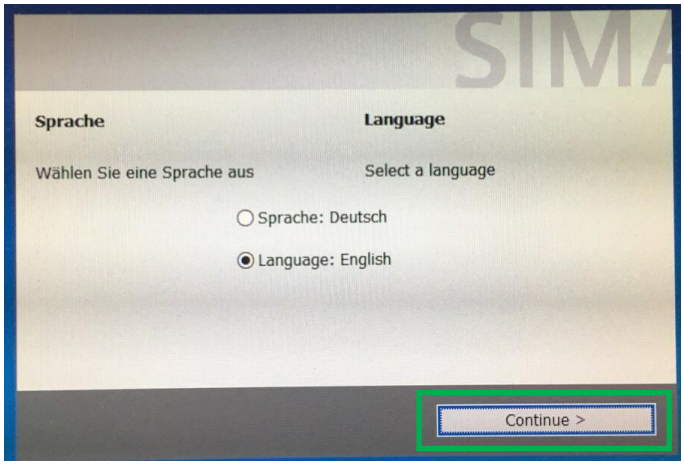


Select “EFI USB Device (ADATA USB Flash Drive)“



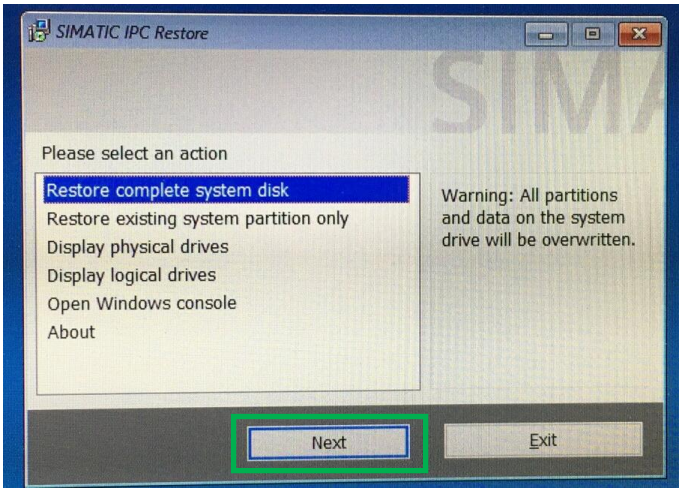


Select “OK“



Select “Language: English“

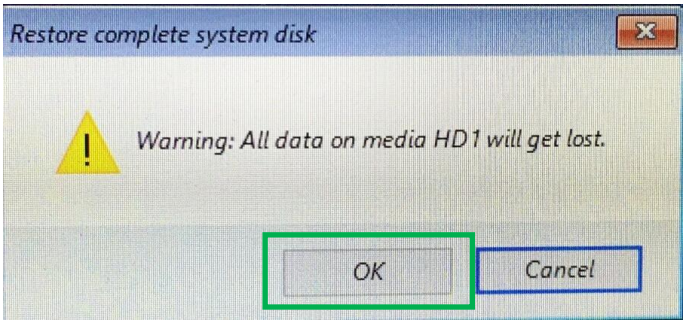
Followed by “Continue“



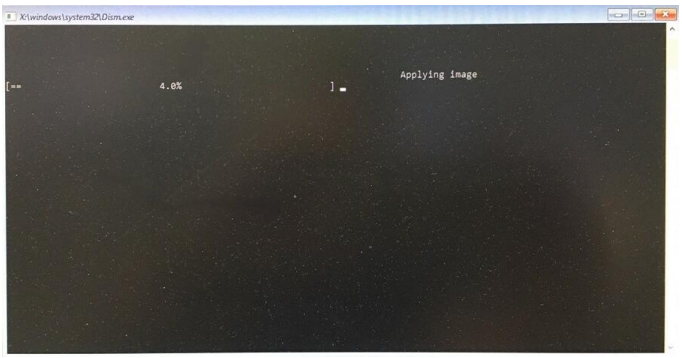
Select “Restore complete system disk“

Followed by “Next“

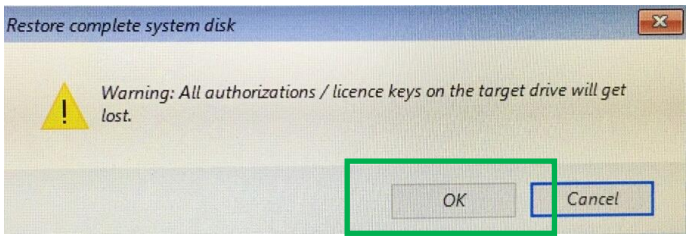
Notes:



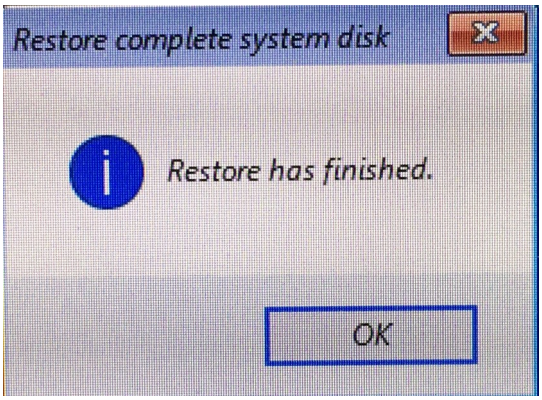
Select “OK“



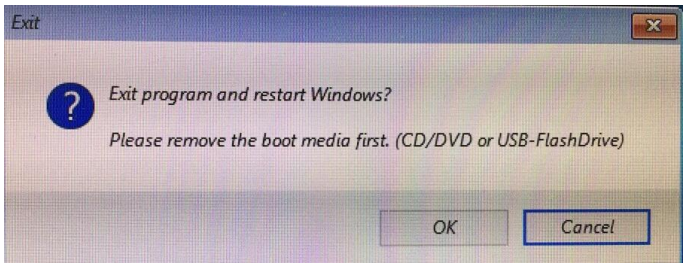
A progress window is shown



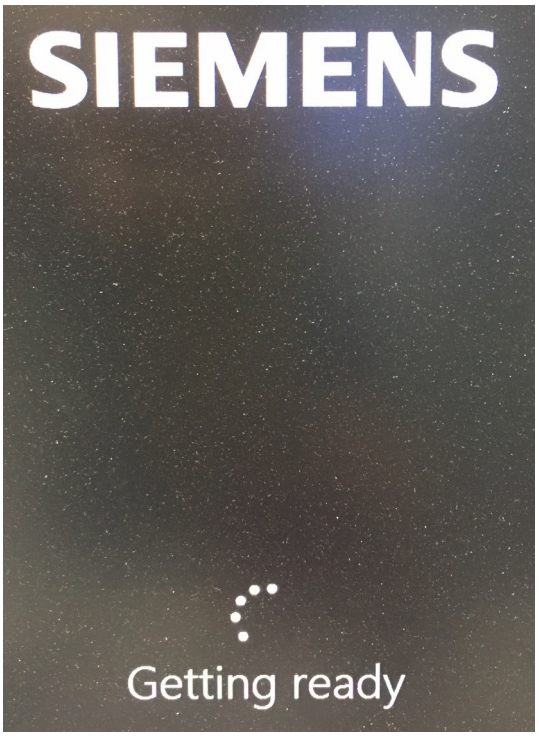
Select “OK“



Select “OK“

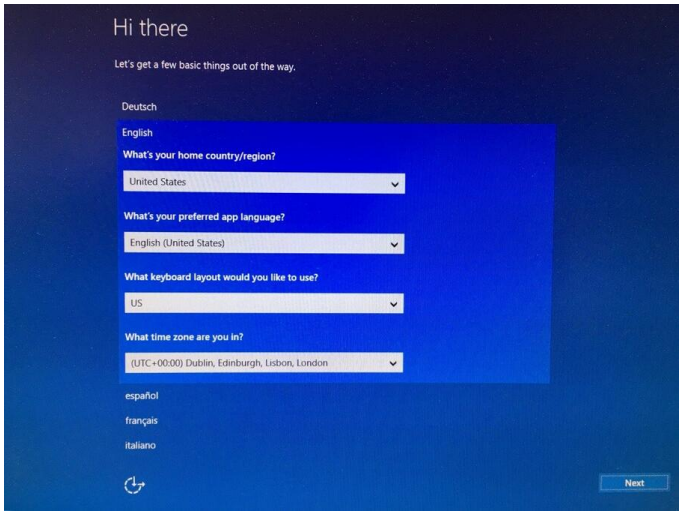


When this window is shown, power off IPC, remove USB boot stick, and power on



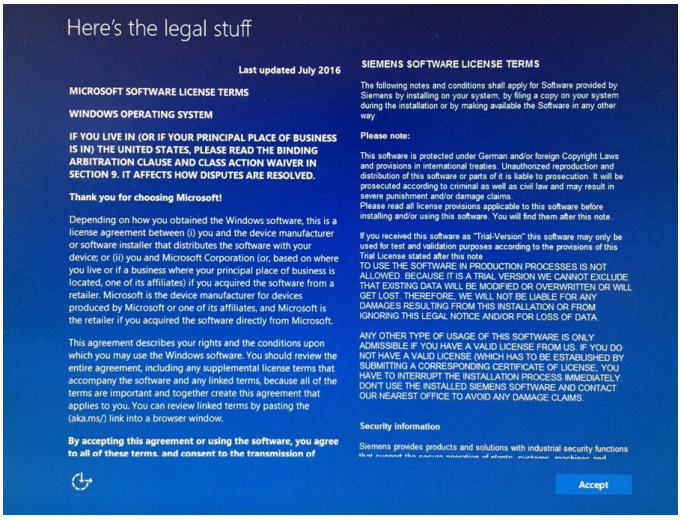
On power up, standard Microsoft progress icon shown

Notes:

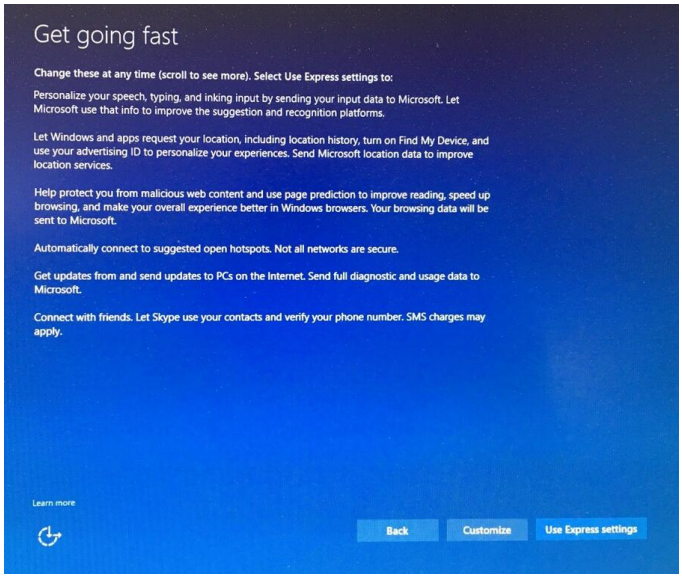


Make sure that the language “ENGLISH“ is selected

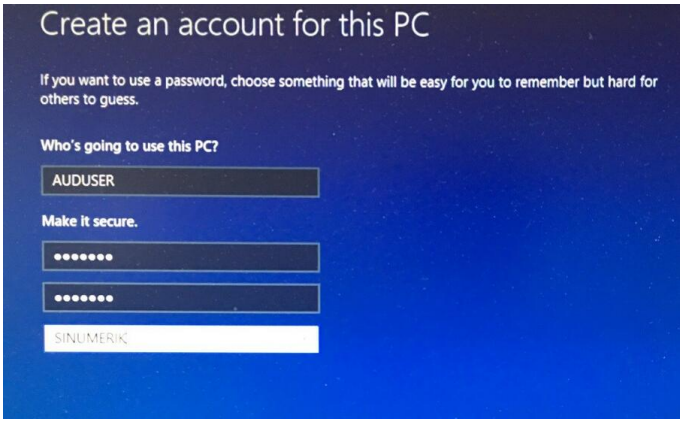
Followed by “next“



Select “next“



Select “Use Express settings“



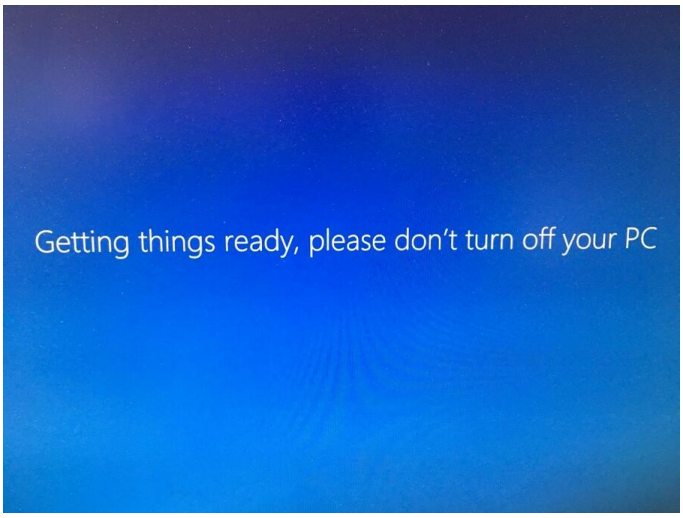
Enter AUDUSER for the account name

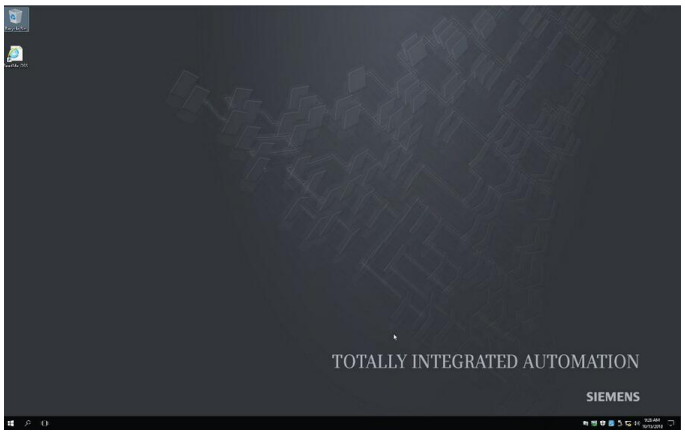
Enter SUNRISE for the password

Enter SINUMERIK for the hint

Followed by “next“

Notes:



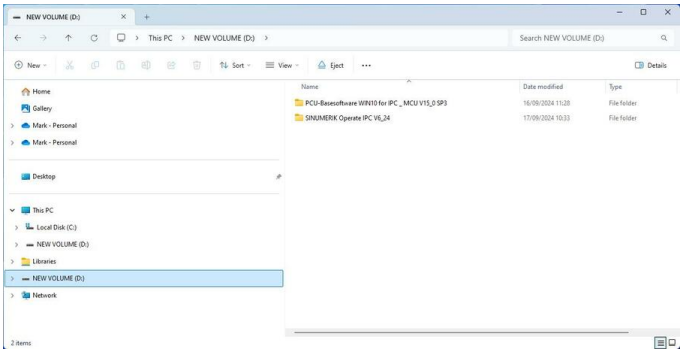


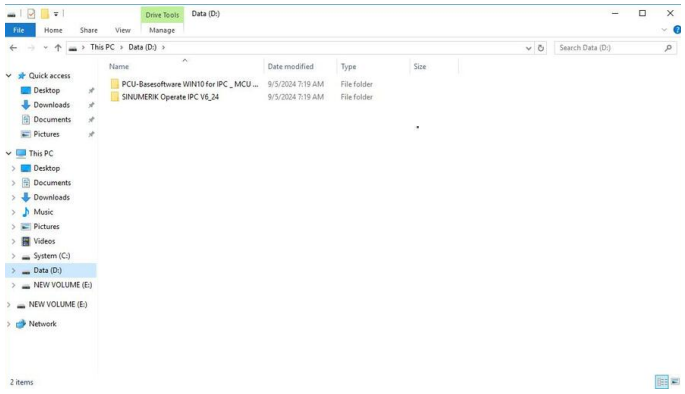
Desk top as shown above.

Windows will setup the account

The IPC will automatically reboot

Install IPC-Basesoftware





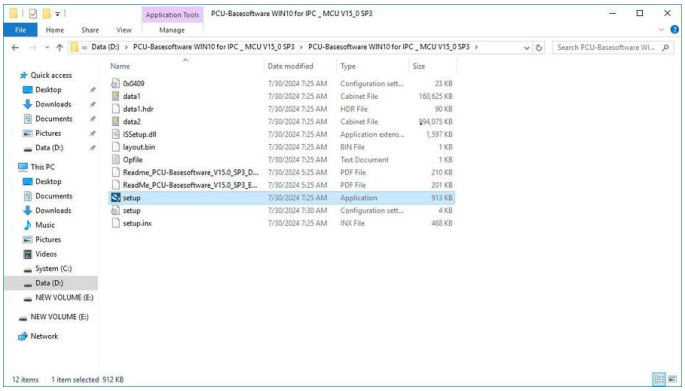
PCU-Basesoftware WIN10 IPC_MCU V15_0 SP3 HF1 must be installed first, this software will setup all the requirements for SINUMERIK Operate BASE including system IP addresses and Ethernet sockets on the IPC.

Using a memory stick, transfer software

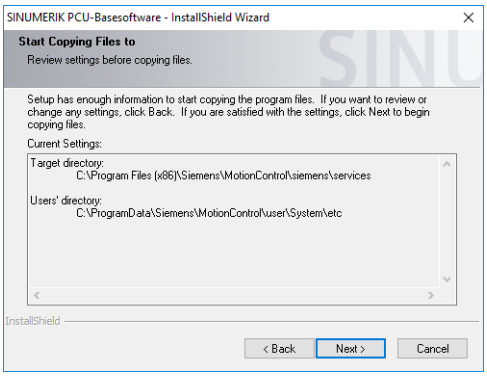
PCU-Basesoftware WIN10 IPC_MCU V15_0 SP3 HF1
SINUMERIK Operate IPC V6_24

using copy and paste into the root of Drive D:/ of the IPC

Notes:

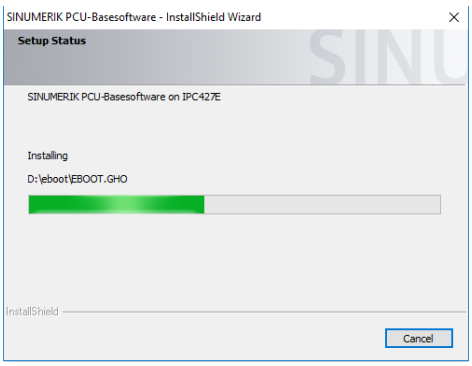


Double click the “SETUP” file to start the installation.

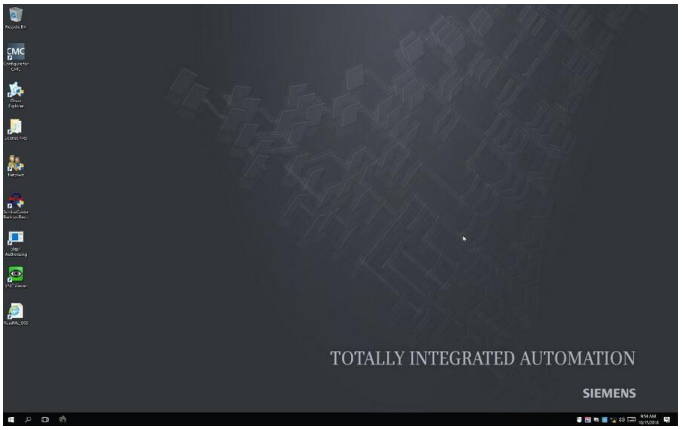


Followed by “NEXT>”





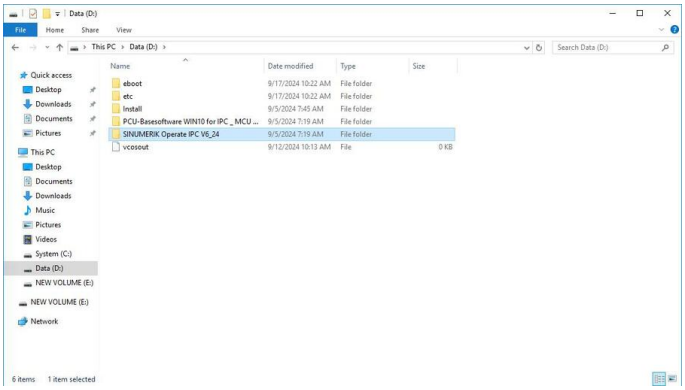
The installation of the Base software will automatically reboot the IPC.



On the desk top ALL the SINUMERIK tools will be present.

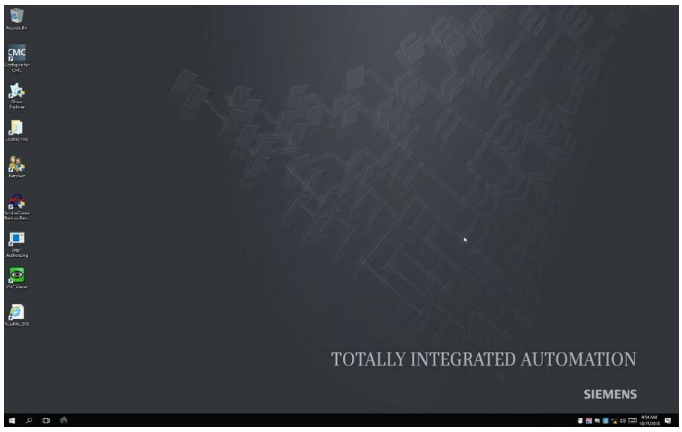
Using windows “Explorer” click on the D:/ drive, where the will be some SINUMERIK folders and files also present:

eboot
etc
Install
vcosout.txt

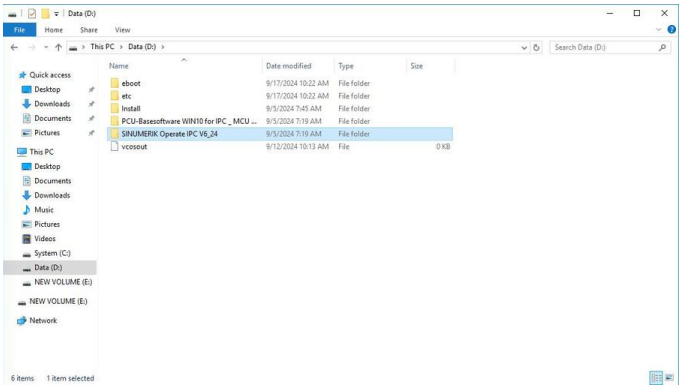


Notes:

Install SINUMERIK Operate IPC V6.24

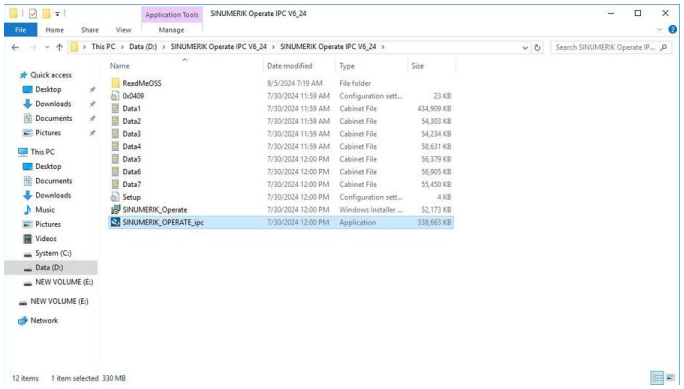


Using windows “Explorer” click on the D:/ drive.

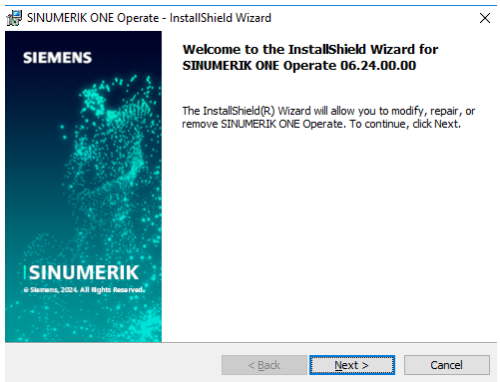


The SINUMERIK software can now be installed:

SINUMERIK Operate IPC V6_24



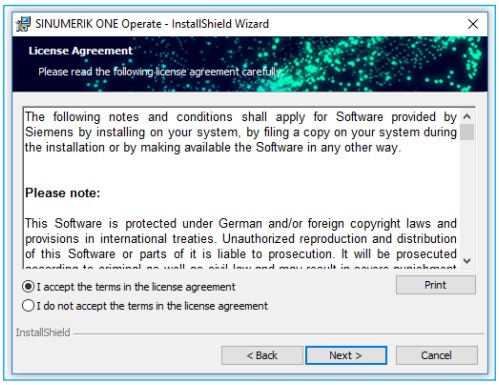
Double click the “SETUP” file to start the installation.

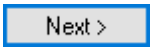


Select “I accept the terms in the licence agreement”

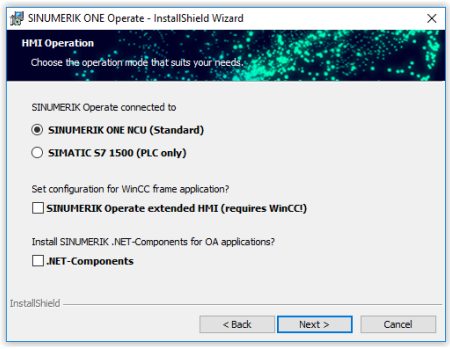
Followed by “NEXT>”



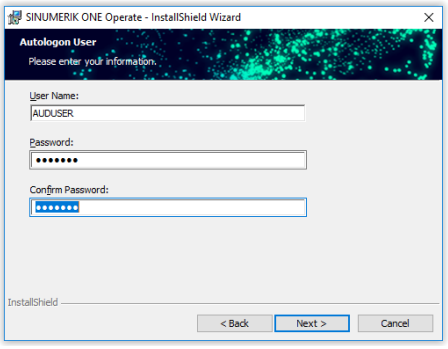




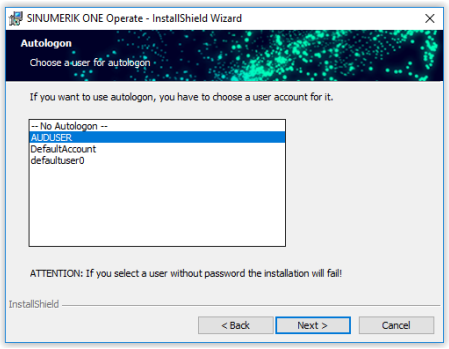
Notes:



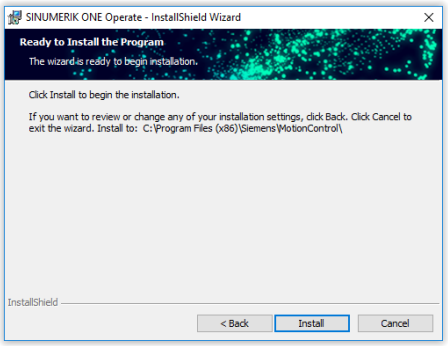
Make sure the “NCU (Standard) is selected.



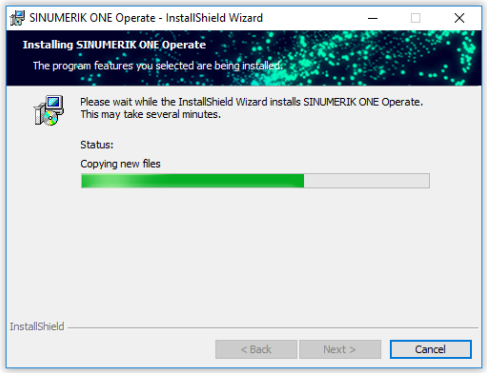
Set the password “SUNRISE” and confirm the password “SUNRISE”

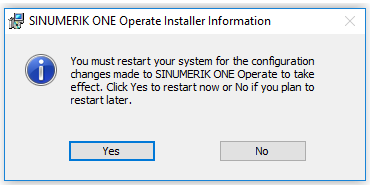


In the “Autologon” window, select the AUDUSER account as setup when installing the OS WIN10



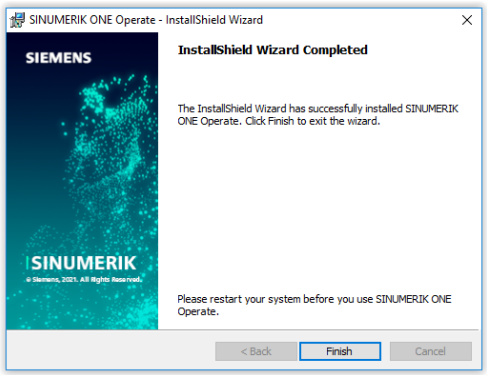
Followed by “Install”

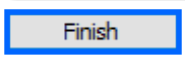




Select “No”

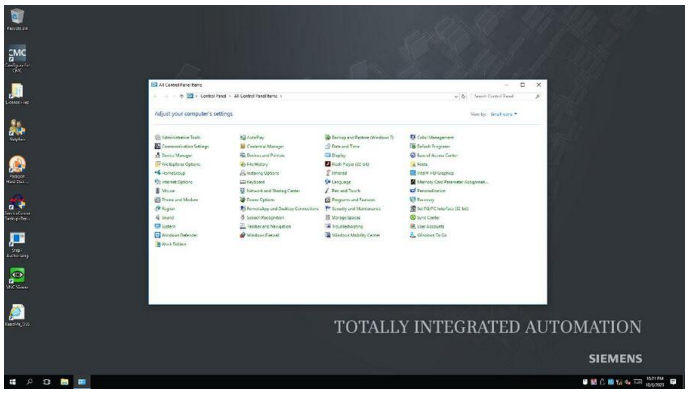




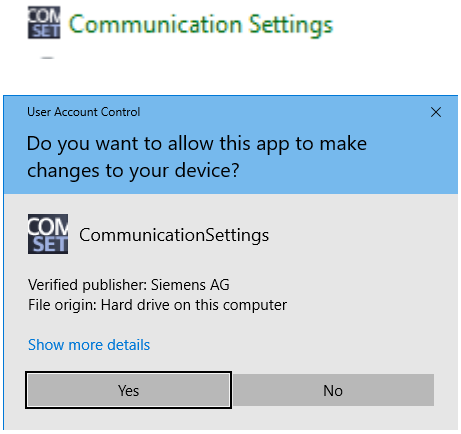


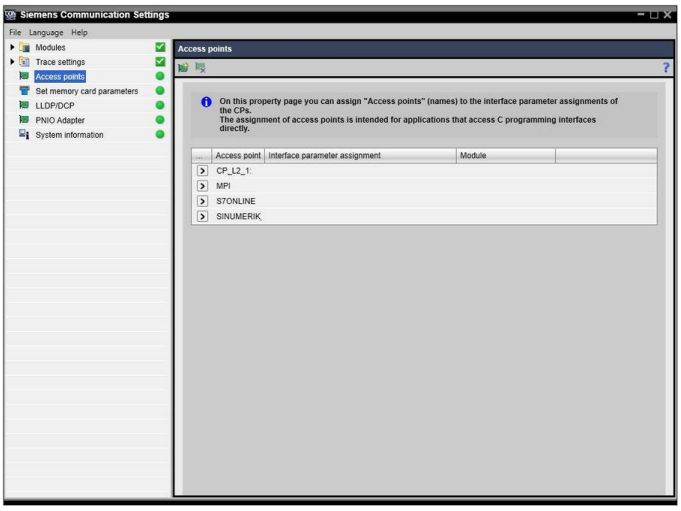
Notes:

Before restarting the IPC, go into the windows Control Panel

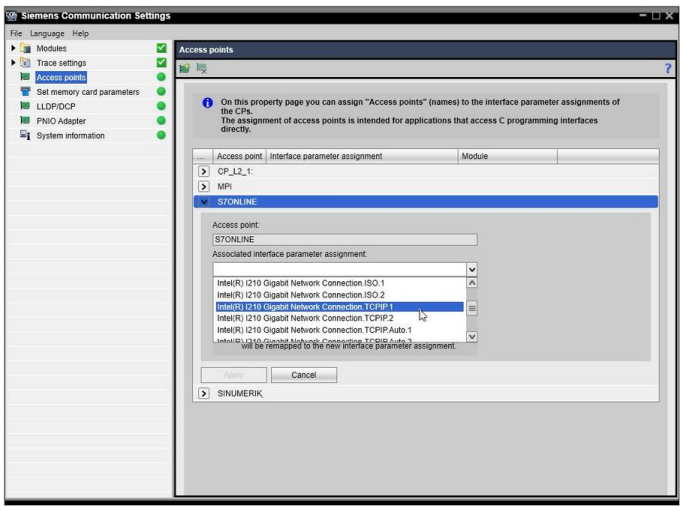


Start Siemens “Communications setting” as shown below





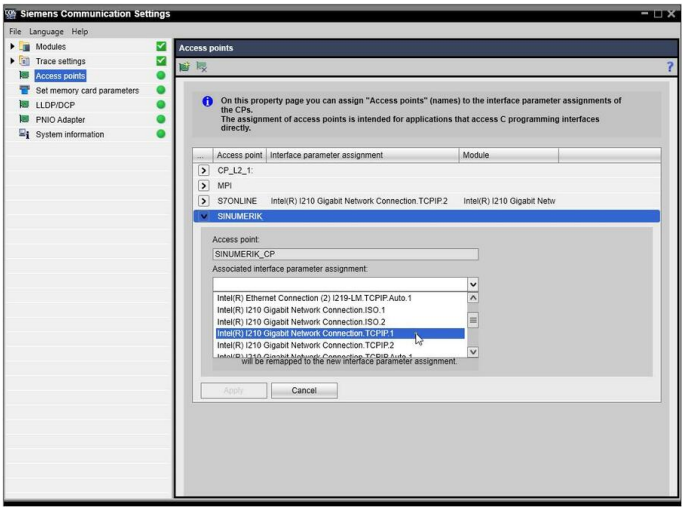
Select “Access points”



Select “S7ONLINE”
Select “Associated interface parameter assignment”
And from the pulldown menu, select “Inter(R)I210 Gigabit Network Connection TCPIP.1”.

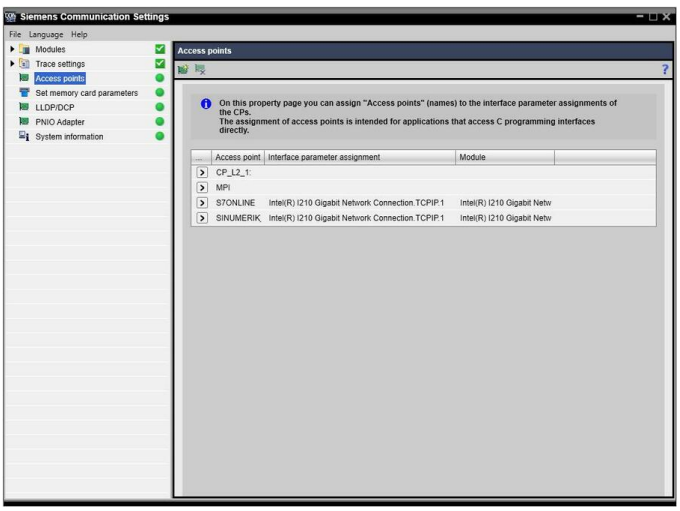


Notes:



Select “SINUMERIK_CP”
Select “Associated interface parameter assignment”
And from the pulldown menu, select “Inter(R)I210 Gigabit Network Connection TCPIP.1”.

Apply



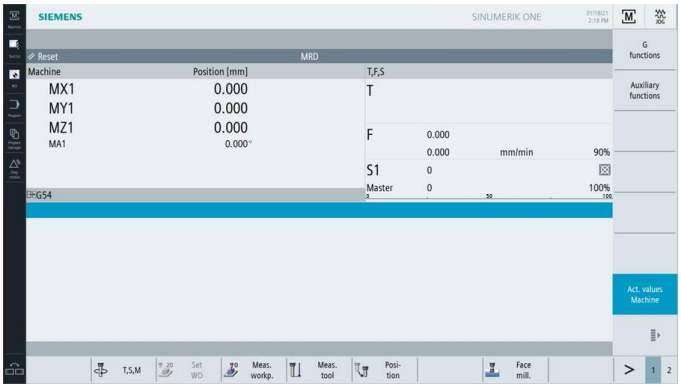
In the **Access points,** set the **S7ONLINE** and **SINUMERIK** to the IPC Ethernet ports, as shown above.

NOTE:

If this is not set alarm150202 will be posted.

Power Off the IPC, and remove the displayPort cable from the IPC box, and power On.

On power Up of the control, SINUMERIK Operate will powerup from the Imbedded HMI on the SD card. Enter the TCU menu.
Courser down to “Select service session” followed by “OK”
When the scan has finished, courser down to
“Show HMI on DESKTOP-xxxxxxxxxxx (192.168.214.241): HMI running”
followed by “OK”



“SINUMERIK ONE v6.24” is now running from the IPC MicroBox

Set the password.

Set the time and date.

A203: END

Notes:

A205

SINUMERIK ONE

Commissioning Virtual to Real

Module Description:

The SINUMERIK ONE controller, together with the “SINUMERIK Operate” operator interface is a modular CNC controller, designed for a wide range of different machine types and technologies.

In order to achieve the full machine functionality many functions have to be brought together and be commissioned.

The sequences which we follow in this module will ensure that the machine is commissioned in the correct sequence and that no faults are generated by incorrect settings appearing at the wrong time.
It will also ensure that the machine is commissioned in an acceptable time period with the highest degree of stability and quality.

Module objective:

Upon completion of this module you will be able to configure a milling machine with 3 linear axes, 1 rotary and spindle according to the defined commissioning procedure.

This procedure is necessary to ensure compatibility and quality of the installation.

Content:

Overall commissioning procedure Virtual to Real for SINUMERIK ONE

Preparing the controller

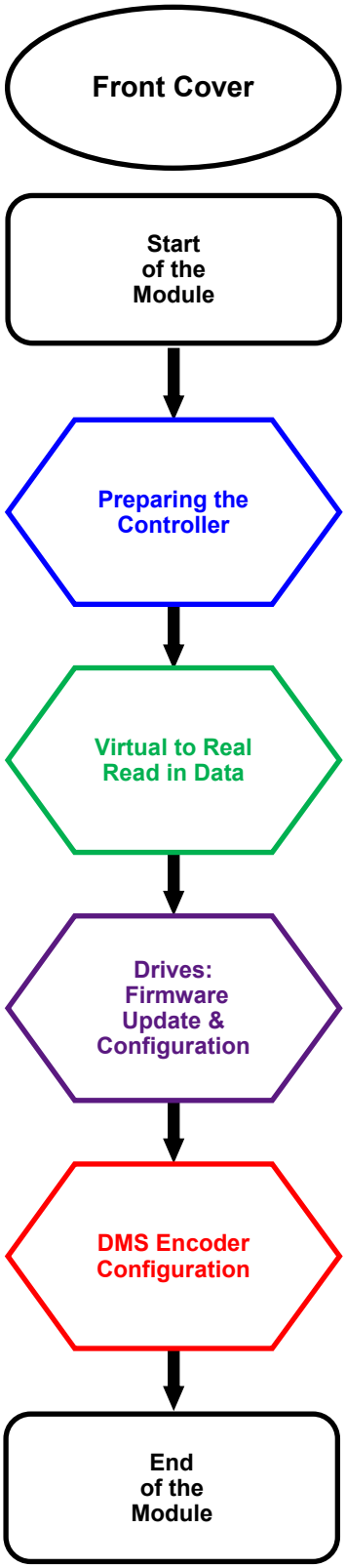
Read in Data Virtual to Real

Drive Configuration

DMS Encoder Configuration

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Preparing the Controller
Virtual to Real Read in Data
Drives: Firmware Update & Configuration
DMS Encoder Configuration
End of the Module

Preparing the controller

To ensure that the controller is ready for the first commissioning the following should be carried out:

- Card is new and has not previously been used.
- Electrical installation finished
- Mechanical installation finished

NC & PLC Standard Data:

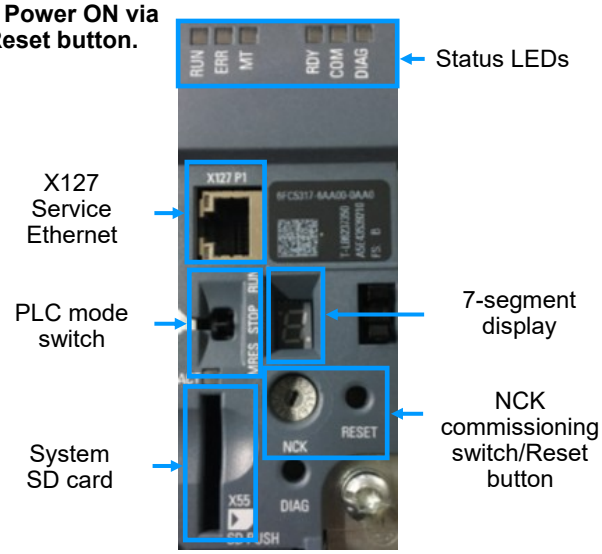
To load NC Standard data:

NCK = 1 + PO

To load PLC Standard data

PLC = STOP + PO

PO = Power ON via the Reset button.

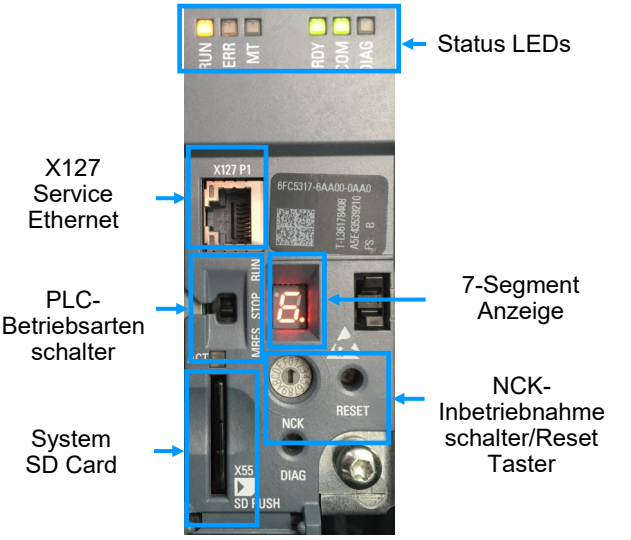


Until the 7-segment display shows the digit “5”, switch PLC from position STOP --> MRES and hold until the 7-segment display changes from digit “5” to digit “1”, within 3 seconds set the toggle switch again to MRES and release -> “**RUN**“ LED indicates the overall reset by flashing rapidly, wait for “**RUN**“ LED showing constant yellow, then switch PLC -> RUN.

Returning NCK switch to position 0.

The following LED status should be reached.

- RUN-LED constant yellow
- RDY-LED flashes green
- COM-LED flashes green
- 7-Segment display on “6“



Notes:

HMI Language - Password - Time and date

At this point in the commissioning procedure it is necessary to perform the following:

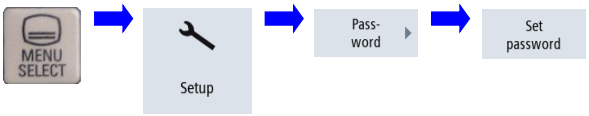
- Set Language
- Set Password
- Correct Time and Date settings

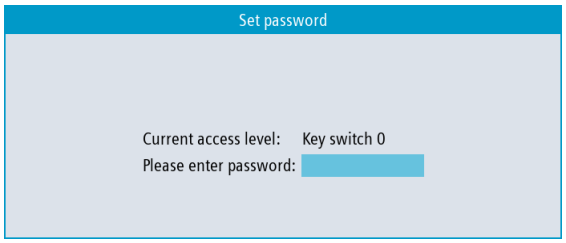
Set Language





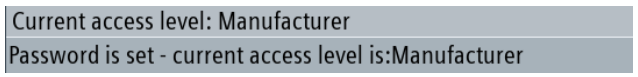
Set Password





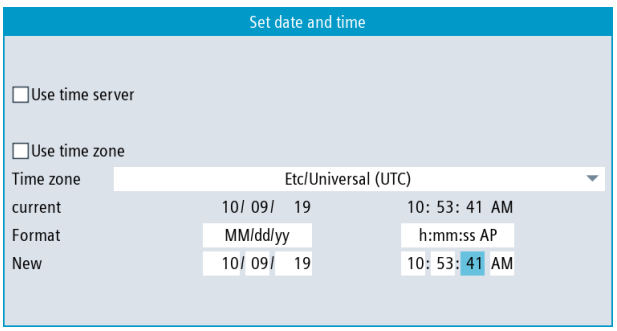
The password - “SUNRISE” is entered , followed by “OK” soft-key.





Correct Time and Date settings





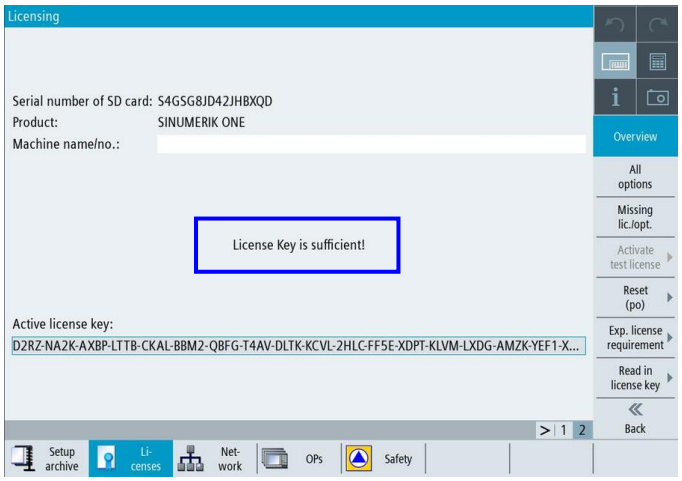


Notes:

Options according to Licenses

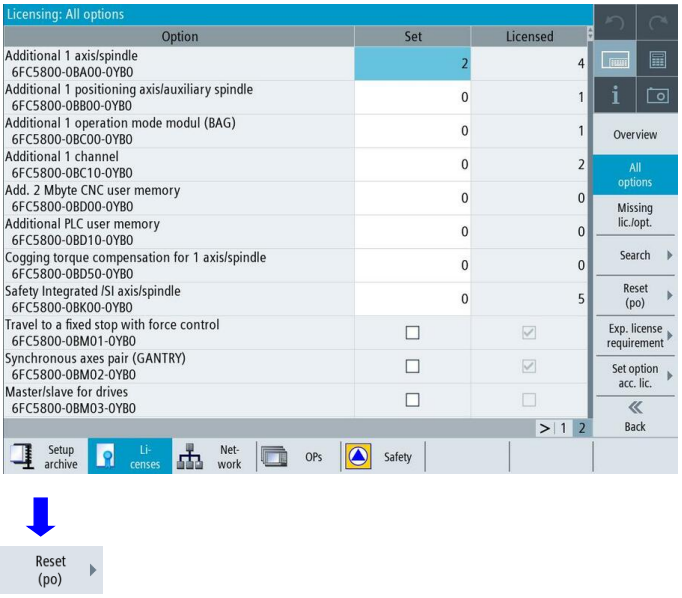
If the System SD card has been purchased with Licensing, the current license can be viewed in the Start-up area under Licenses.

“License key is sufficient“ is displayed, indicating that no options are set without the necessary license.

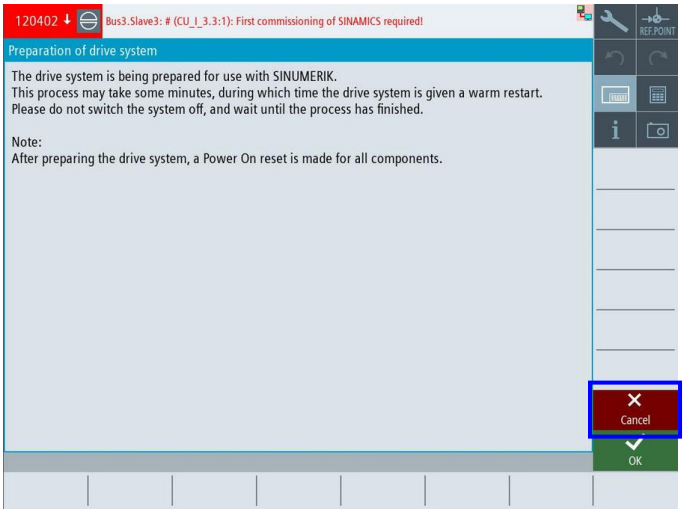


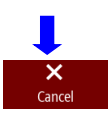
In the next step the options can be set according to the machine configuration.

In order to activate an option an NCK reset is necessary (Reset PO).



After performing an NCK reset, the controller will start-up with the following screen, this screen indicates that an initial start-up of the drive system is necessary.
The start-up of the drive system is carried out later in the start-up process, therefore continue the process with the “cancel“ Soft-key.

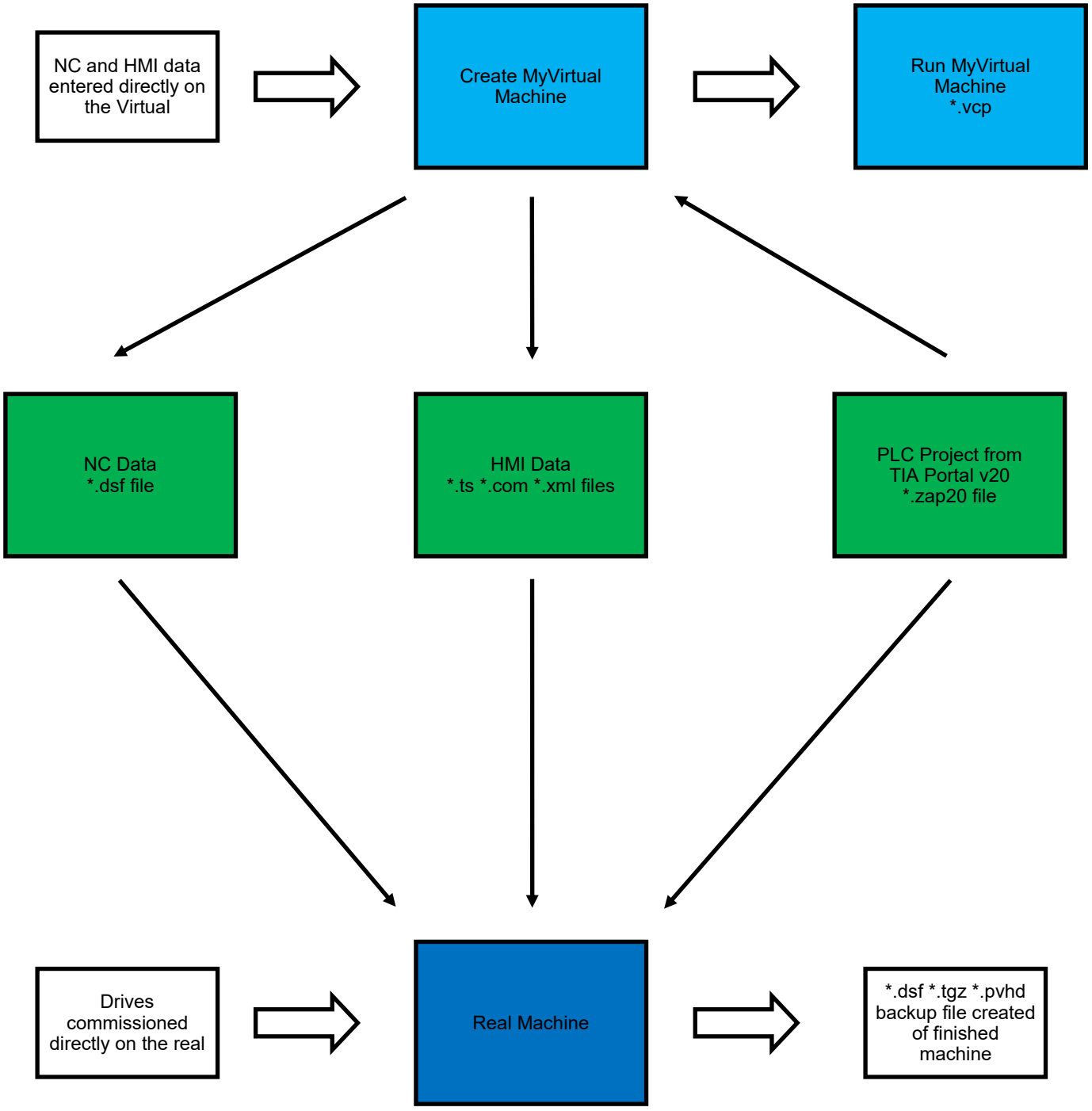




Notes:

Virtual to Real Read in Data

Concept of data movement



Notes:

Drive firmware update and configuration

A system whose drives have not been configured will boot by default to the following screen.

When the “OK” soft-key is selected the system will check the equipment connected to the Drive-CLiQ bus.

The firmware will be loaded to any hardware not at the latest version.

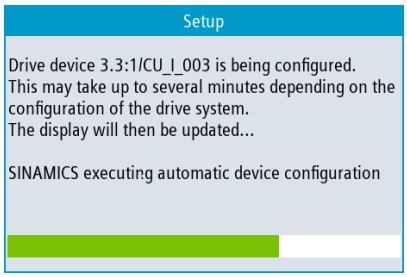
The configuration of the connected drive hardware will be saved to the controller as part of the drive configuration procedure.

Note:

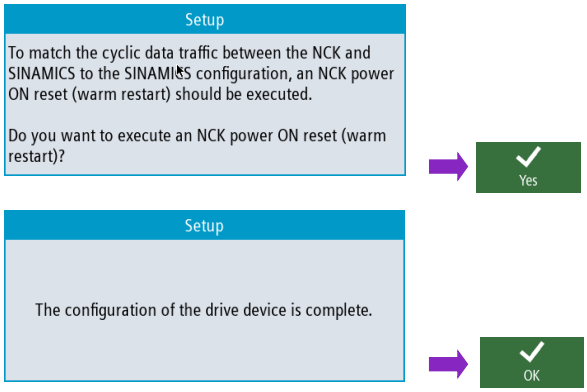
A system whose drives already have been configured must be set to factory settings to perform the configuration.



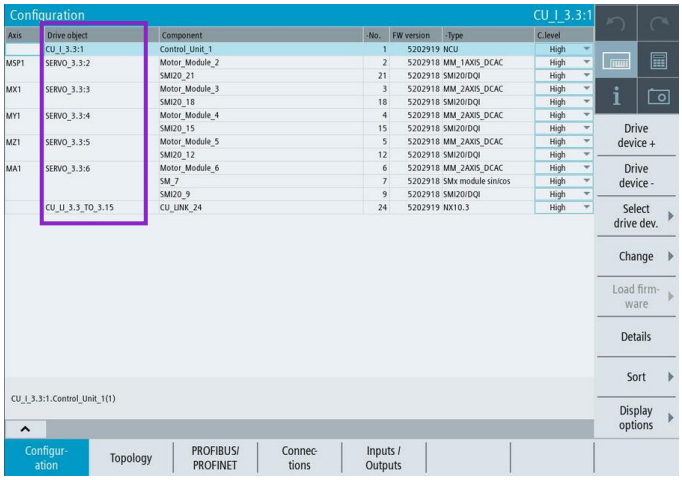
A series of information boxes will be displayed:



An NCK reset is required to complete the procedure:



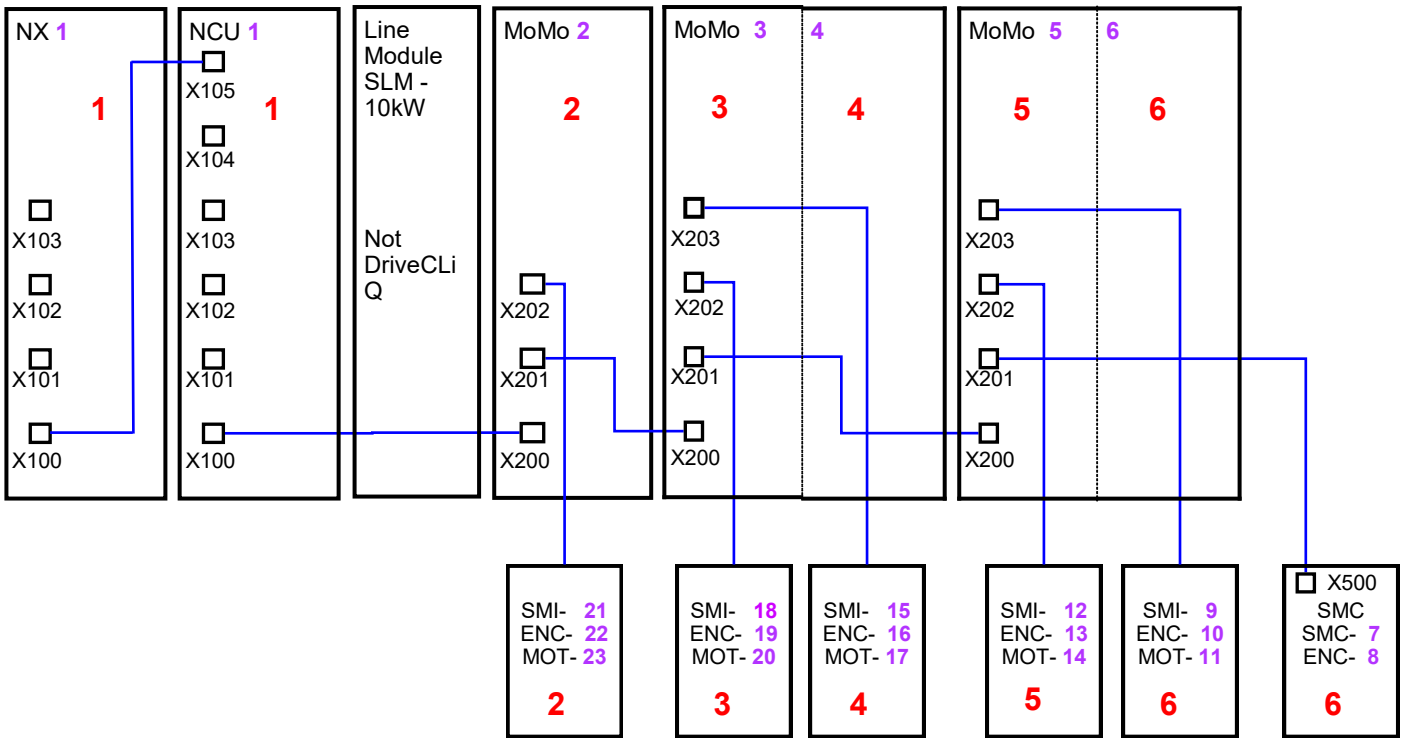
The drive components that are connected to the DriveCLiQ bus are now displayed on the controller.



Note: The axes are already assigned at this stage.

Notes:

Topology of Training Simulator S120



DMS Encoder Configuration

In certain situations it is not possible for the topology to assign components according to the machine configuration.
In this case the assignment has to be changed manually.

The following example is necessary because the Spindle Direct Measuring System (DMS) is connected to another axis (A axis).

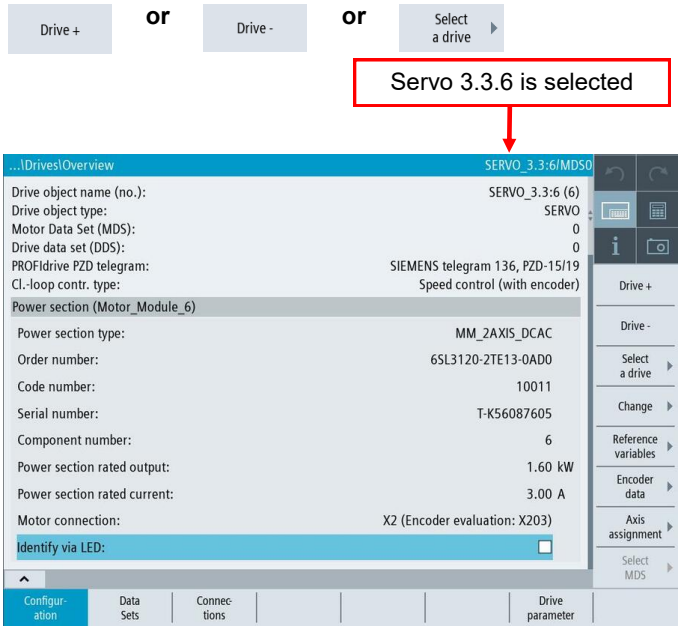
The system needs to be configured so that it associates this encoder with the spindle (Physical location of the encoder).

Deselecting the DMS encoder from Servo 3.3.6 (A axis):

To continue, select the following soft-key sequence:

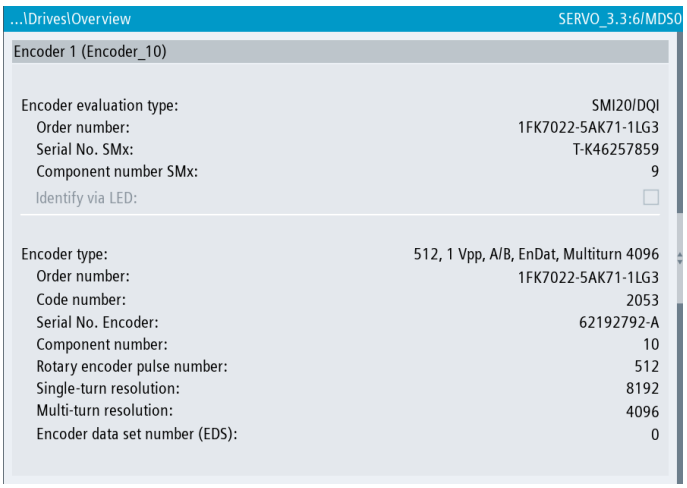


Use the “Drive +“ or “Drive -“ or “Select a drive” soft-keys to locate the drive which has been incorrectly assigned the DMS (Servo 3.3.6).



Notes:

By paging down to the bottom of the screen it can be seen that this drive does in fact have a 2nd encoder assigned to it.
The encoder type is unknown at this stage and will be added shortly.



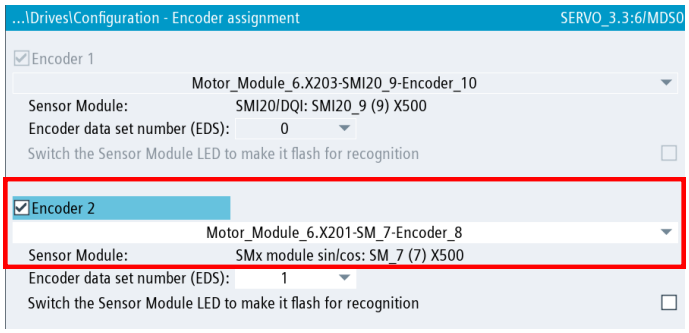
The encoder has to be first deselected from **Servo 3.3.6** (A axis) to release it up for assignment to the **Servo 3.3.2** (spindle).

Select the “Change” soft-key.

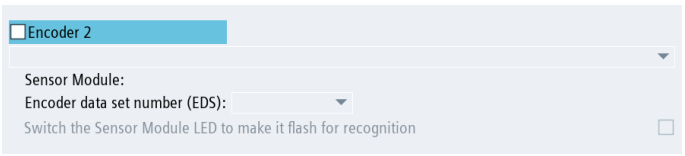


Press the “Next step” soft-key until the **Encoder assignment** screen is displayed, the page will show two encoders assigned to the axis.





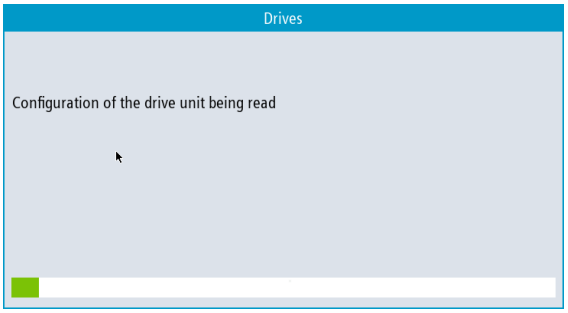
Deselect Encoder 2:



Press the “Next step” soft-key.

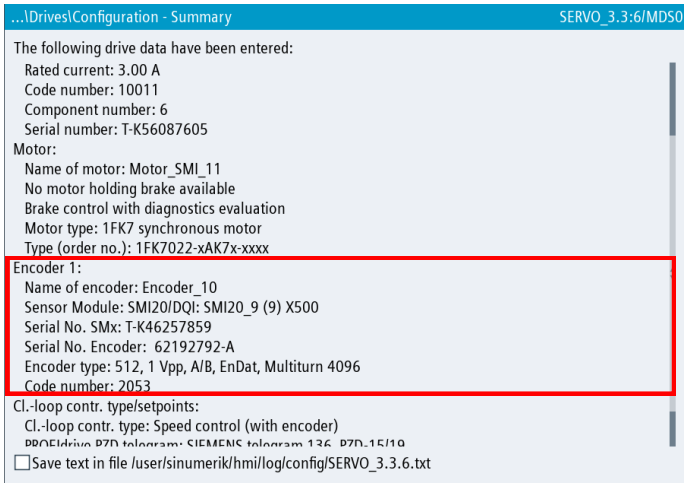


Drive Configuration will be read:

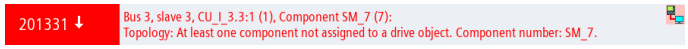


Press the “Next step” soft-key until the **Summary** screen is displayed.
This now shows Servo 3.3.6 with only one encoder.





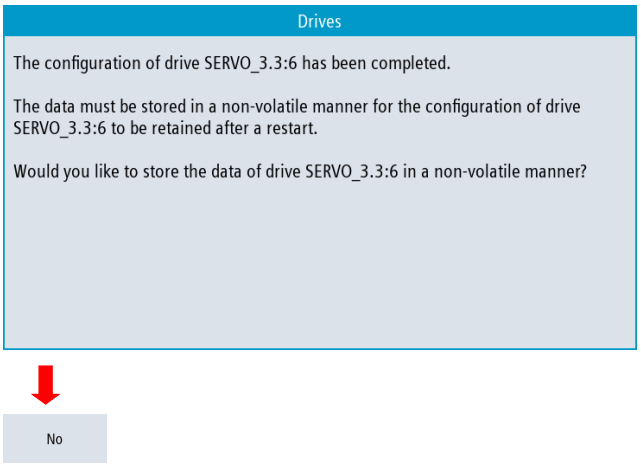
The system detects the removal of the encoder and issues an alarm which is normal at this stage.





Notes:

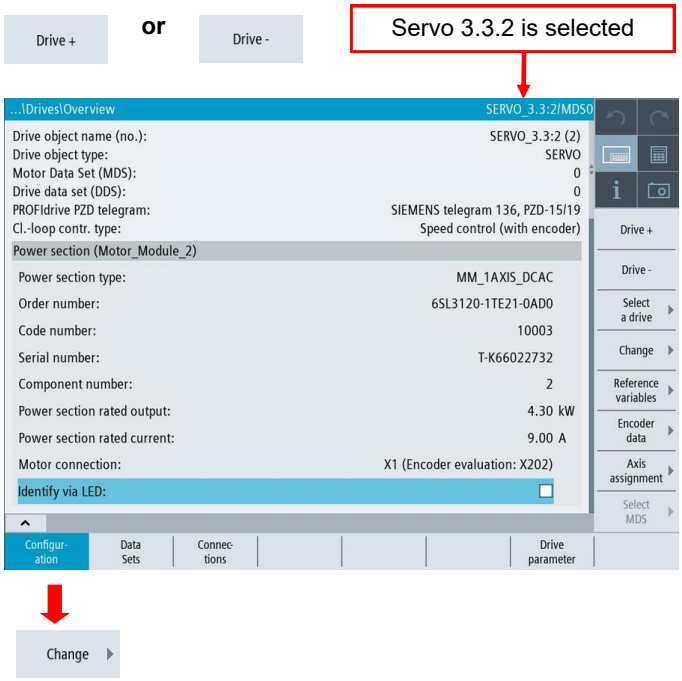
A dialogue box shows that the configuration has changed and that the date must be saved.
This is not necessary at this stage because the encoder has to be assigned to the spindle.



The second encoder must be assigned to the correct axis, in this case this is drive 2 (spindle).

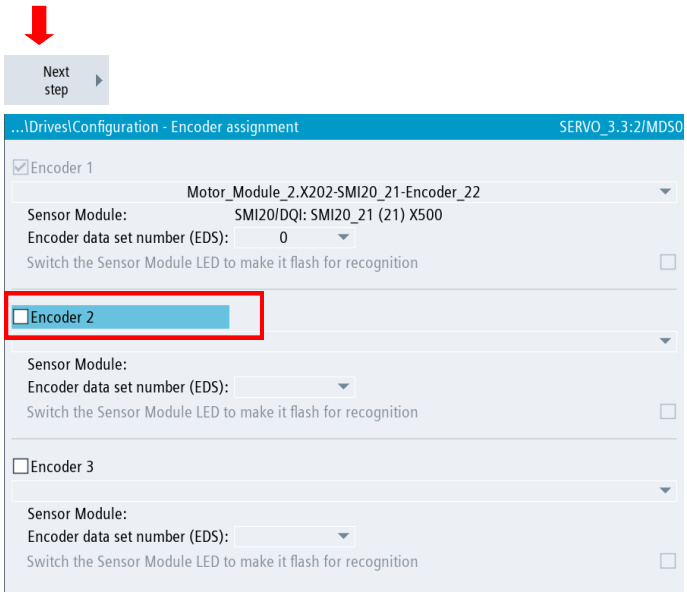
Assigning the DMS encoder to the Spindle:

Use the “Drive +“ or “Drive -“ or “Select a drive” soft-keys to locate **Servo 3.3.2**.



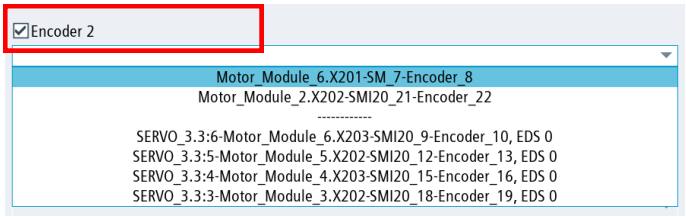
Press the “Next step” soft-key until the **Encoder assignment** screen is displayed.

The page will show one encoder assigned to the servo.

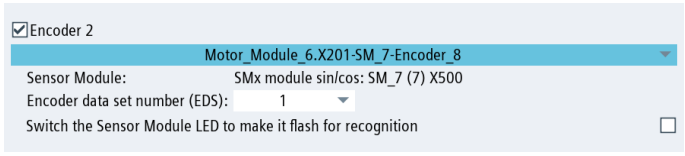


Select the **Encoder 2** checkbox.
From the list select the DMS encoder.
The top two encoders on the list are the ones available to the spindle. The 1st one being the DMS **(Motor_Module_6.X201-SM_7-Encoder_8)**, the 2nd being the one already assigned as the motor encoder.

The remaining items on the list are motor encoders which are already assigned.



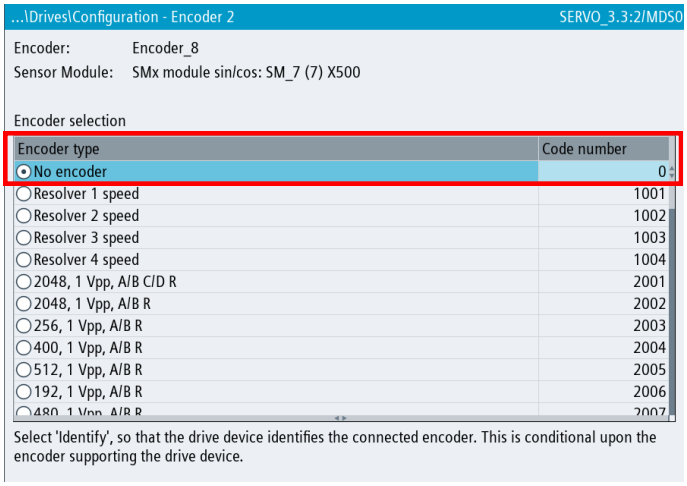
Below: Encoder 2 **(Motor_Module_6.X201-SM_7-Encoder_8)** selected and assigned to the DMS.



Notes:

Press the “Next step” soft-key until the **Encoder 2 configuration** screen appears.



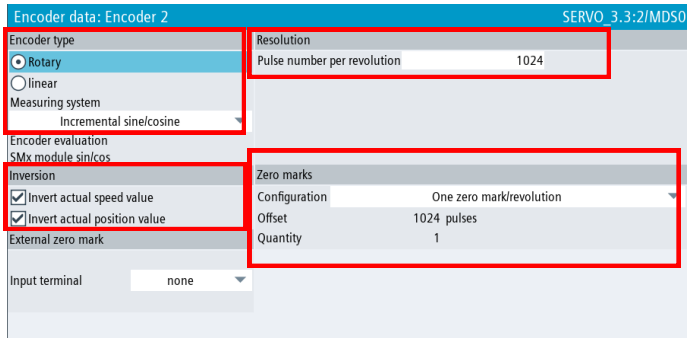


The details of the encoder need to be manually entered, select the “Enter data” soft-key.



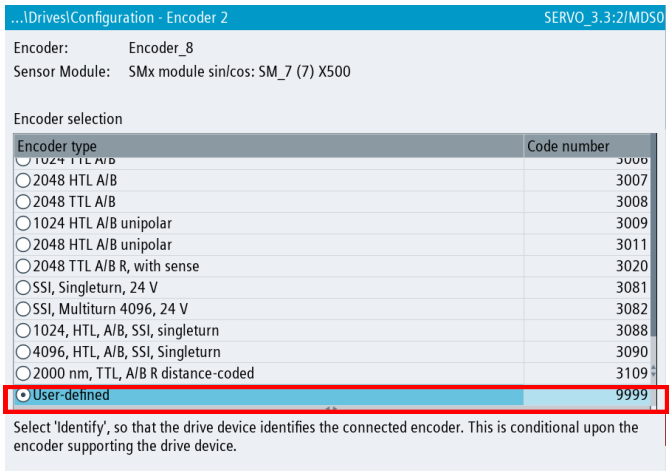
In this case the encoder is of the type: **Rotary incremental Sine/Cosine type** with **1024 pulses** and **one zero mark per revolution.**
The **position actual value** needs to be inverted due to the mounting method of the DMS encoder.

The screenshot below has had the encoder type and information entered:





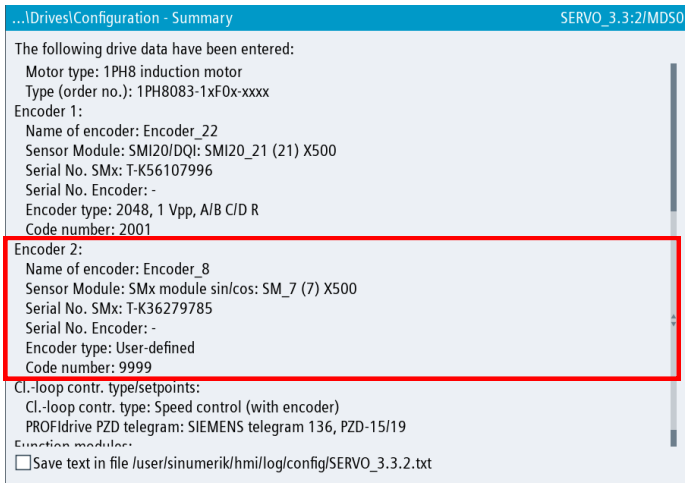
The encoder now appears in the list as a **User-defined** type:



Press the “Next step” soft-key until the **Summary** screen is displayed.



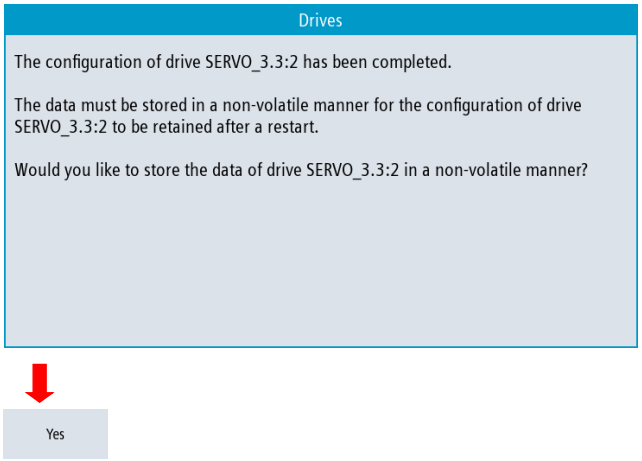
The bottom of the summary screen shows Encoder 2 details:



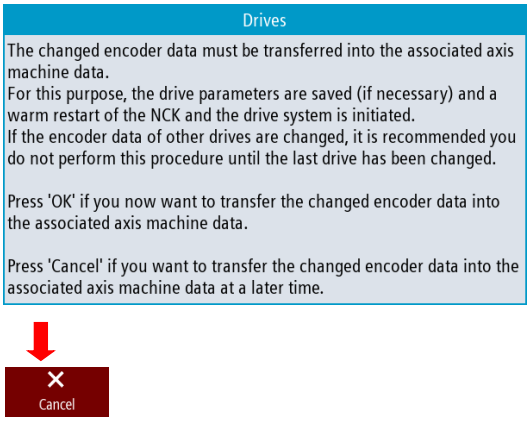


Notes:

The DMS assignment is now complete and needs to be saved, press the “Yes” soft-key.



The encoder data are already assigned at this stage, press the “Cancel” soft-key.



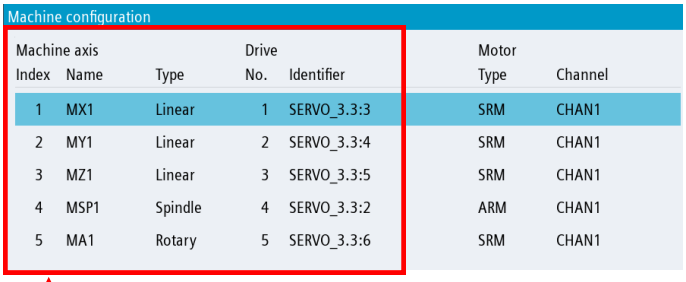
The new settings will now be saved.

The second encoder now is assigned to the correct axis, in this case this is drive 2 (spindle).

Axis assignment confirmation:

At this point the NC axes and the Sinamics Drive objects are configured and logically assigned to each other.

The axis assignments can be seen on the Machine configuration screen:



Axis and drive assignment.

A205: END

Notes:

Notes:

A209

SINUMERIK ONE

Servo Optimisation

Module Description:

This module explains the step-by-step procedure for tuning the speed and position control loops within the SINUMERIK ONE / SINAMICS S120.

The procedure is based on the onboard functions of the SINUMERIK ONE system.

Module Objective:

In order to ensure the required compatibility and consistency level between the machines configured with the SINUMERIK ONE control, a defined tuning procedure should be observed. This procedure is defined in this module.

Notice: This procedure causes the axis to move. All the required safety measures must be taken into account. Before starting, you must read through the entire description and understand it.

Content:

Introduction

Automatic Servo Tuning

Acceleration

Axis jerk

Circularity test

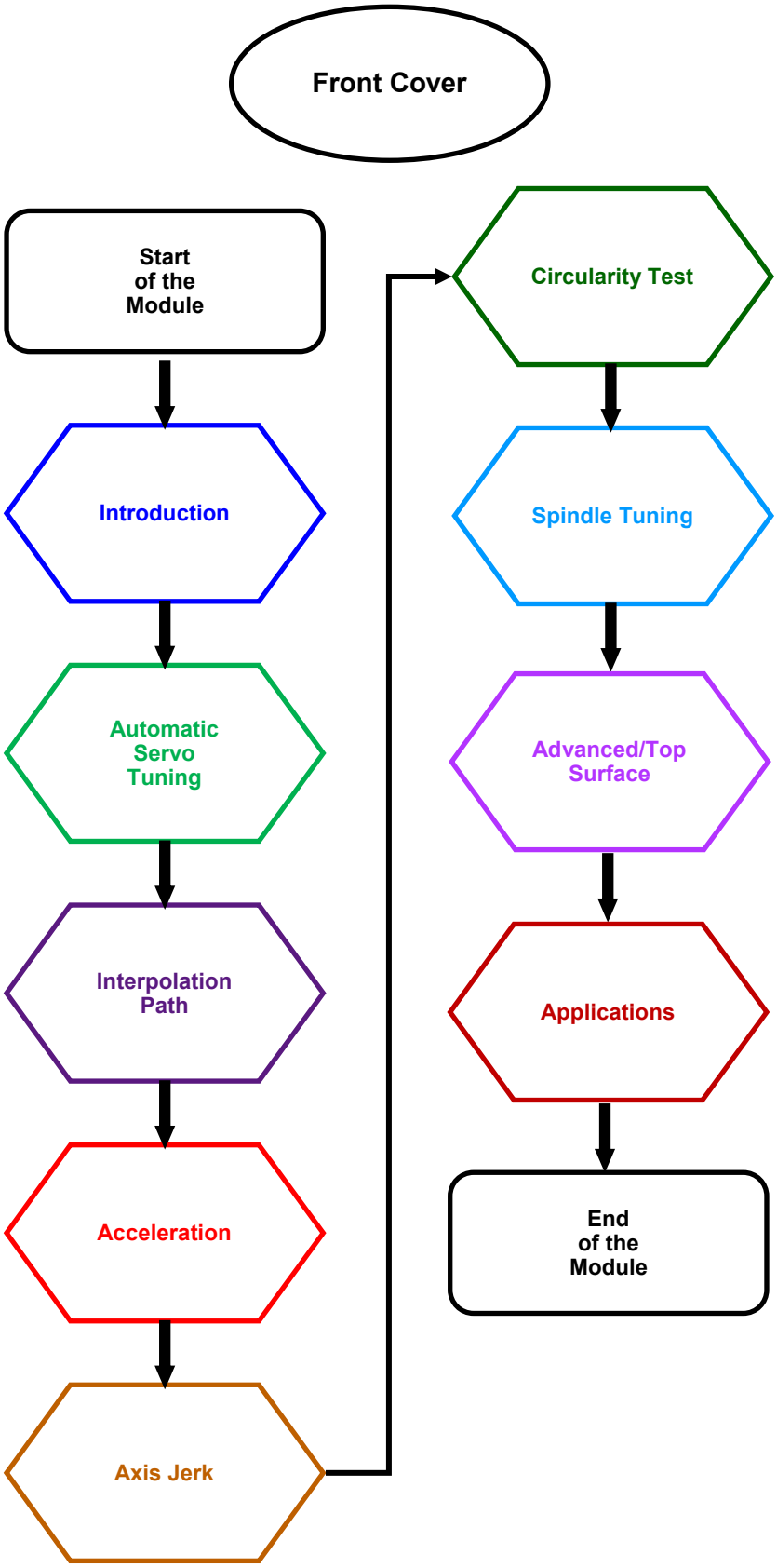
Spindle tuning

Advanced Surface/Top Surface

Applications

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Introduction
Automatic Servo Tuning
Interpolation Path
Acceleration
Axis Jerk
Circularity Test
Spindle Tuning
Advanced/Top Surface
Applications
End of the Module

Introduction

General notes
Since drive tuning is a very complex topic during the commissioning procedure, it is recommended that you first read the document to obtain an overview. The individual tuning steps can then be performed in the sequence described here.
The recommendations are summarized again for the most common applications at the end of the module.

Note that the strategies described here apply to the majority of machines. For special-purpose machines or special requirements, it may be necessary to make expert settings. However, as this topic exceeds the scope of this module, which is only intended to explain the operation of AST under "standard conditions", it is not considered in more detail. Please refer to the Mechatronics Servo Tuning with AST for a more in depth procedure.

Activating the changed machine data
In this procedure of optimization, certain changes to the machine data must be activated. How this change is activated is shown in the last column of the respective parameter value. The following list shows these specifications and their meaning.

po	The relevant machine data requires an "NCK reset" to take effect.
re	The relevant machine data requires an activation through the "Reset" button.
cf	The relevant machine data requires an activation through the "Set MD active" softkey.
im	The relevant machine data is active immediately.

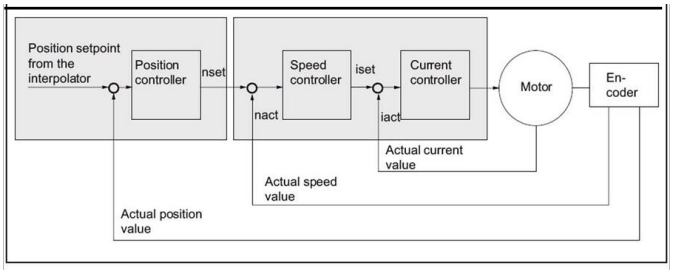
Saving the drive data
Each time the SINAMICS S120 is switched on or a "Drive reset" performed, the configuration data from a file saved on the SINUMERIK control is loaded to the appropriate drive. If drive parameters are changed, the drive data must be saved so that the changes are retained after the next power off/on or after a "Drive reset".

Note: An "NCK reset" has no effect on the drive data.

To save the drive data:
In the Drive parameters area, press the "Save/reset" softkey, then the "Save" softkey and finally the "Drive system" softkey. Messages in the lower left-hand area of the screen show the status of the saving operation. When the saving operation is finished, the message "All drive objects saved" is displayed.

Control block diagram
Sinumerik ONE/ S120 control structure: The Sinumerik ONE CNC and the S120 drive control structure contain three cascaded, closed control loops (current, speed and position). The individual control loops must be tuned in succession: first the current controller, then the speed controller and finally the position controller. The dynamic machine limits for velocity, acceleration and jerk are set in the interpolator. The speed control loop is always closed via the motor encoder, whereas the position control loop can be closed via the motor encoder or the direct measuring system.

SINUMERIK ONE **PROFIBUS-DP DRIVE**



Setup of basic parameters

In order to proceed with the optimization successfully some axis dependent parameters have to be set at the beginning. They are:

ballscrew pitch h	MD31030
gear factor axis i = MD31060 / MD31050	MD31050 (n_{load}) MD31060 (n_{motor})
encoder inversion	p410
max. acceleration	MD32300

However it is difficult to determine the maximum acceleration at this point in time because inertia, friction and holding torque are often unknown. Likewise the maximum permitted jerk is irrelevant for the execution of Auto Servo Tune, we will come back later to this topic.

Notes:

Automatic Servo Tuning

The Automatic Servo Tuning function is used to tune the current, speed and position control loops, including torque feedforward control. This can also be performed by users without extensive mechatronic knowledge. However, it is also possible to make expert settings.

Objective: Fine setting of a single axis or spindle using HMI SINUMERIK Operate.

- Performance of the frequency response measurements
- Calculation of the speed control loop
- Setting of the current setpoint filters
- Calculation of the position control loop
- Verification of the actual response of the planned control loops
- Feedforward control mode: equivalent current control time
- Interpolation paths

Notes
If available, the second measuring circuit should be enabled.

The (axis) **MD32640 dynamic stiffness control** is always set to 1 by the Automatic Servo Tuning function; this enables position control with a faster cycle time.
When using the dynamic stiffness control, **MD32110 ENC_FEEDBACK_POL** must be set to 1. If the polarity has to be reversed, this must be performed in the SINAMICS drive via **p410**.

p410[0] & [1] Reverse actual encoder value:
Bit 0: Reverse actual velocity value
Bit 1: Reverse actual position value

Example: To reverse the actual velocity and actual position values of the second encoder, set p410[1] = 3.

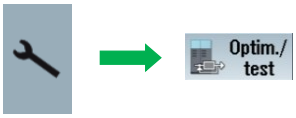
If the automatic tuning is performed for a main spindle, a PLC servo enable signal may not be present. The machine manufacturer should provide a method to enable this PLC interface signal for the servo tuning, e.g. a special key combination or by setting a special marker in the PLC status.

It is strongly recommended that you use the torque feedforward control for the tuning. This ensures the greatest possible contour accuracy.
By keeping the following error sufficiently small, the servo gain factor has almost no effect on the path interpolation. Therefore, different servo gain factors are now permitted in the axes for the path interpolation.

It is recommended that the position, speed and torque setpoints are smoothed by using a "Jerk filter" (AX_JERK_TYPE = 2)

General procedure
If a machine or an axis is tuned for the first time, it is recommended that this tuning is performed with a **robust strategy** without further manual adaptations. This is to ensure that the axis can be traversed correctly and the controllers can be set correctly, even if they have not been tuned . The axis can now be run in, aligned, etc.
In the second step, the tuning is performed again, but now with realistic loading of the machine (e.g. with installed chuck and possibly a workpiece) with the desired strategy. Without any load, the machine may be set well for no load, but is unstable later with the chuck and a workpiece.

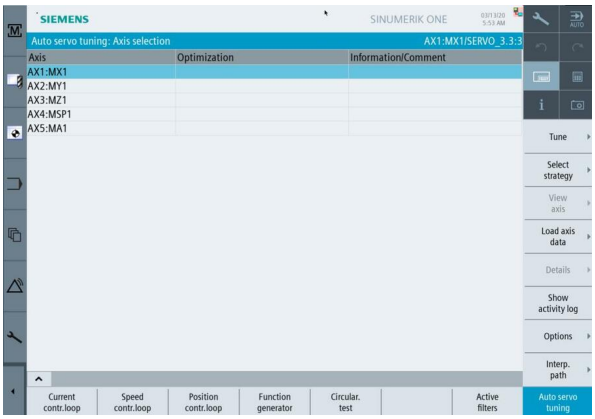
Call of the tuning interface



The Auto servo tuning screen is the standard screen. If this is not the case, press the "Auto servo tuning" softkey.



The screen shows whether an axis has been tuned or not. The time and date of the tuning is shown for a tuned axis.



Notes:

Adapting the settings



The default settings are for tuning at the simple press of a button with minimum user interaction. The "Enable strategy selection during tuning" checkbox is only for experts.



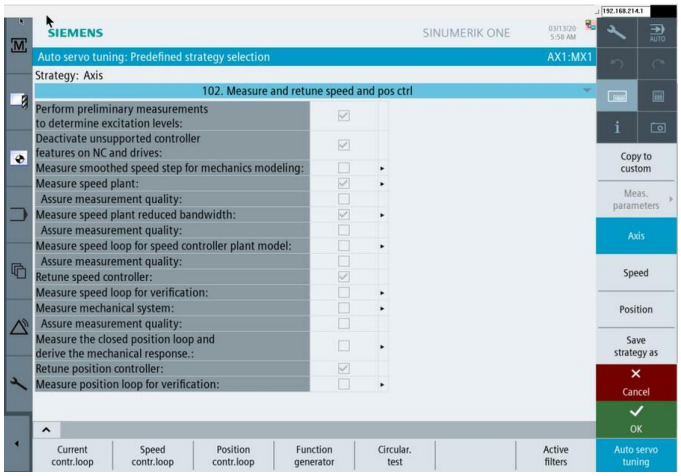


Tuning strategy

The tuning process should now be adapted to the requirements of the respective axis .

First select the axis to be tuned, and then the strategy can be adapted.





Note: The set strategy only applies for the selected axis!

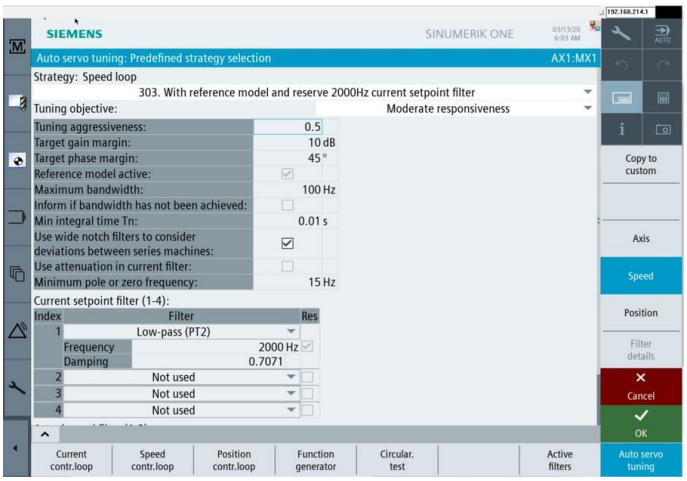
The strategy settings are divided into three areas: axis (general strategy settings), speed (speed controller) and position (position controller).

Depending on the requirements, different strategies can be selected in the axis settings (tune position controller and/or speed controller, with/without verification, etc.). We are using the strategy **"102. Measure and retune speed and pos ctrl"** (the designation can vary slightly depending on the version of the software; however the number of the strategy remains the same). The speed and the position controller are measured and tuned. Verification of the closed-loop will not be performed due to the easier operation.

Speed controller settings



Different strategies can also be selected for the speed controller. The strategy **"303. With reference model and reserve 2000 Hz current setpoint filter"** has proven itself here.
It is also possible to select a tuning objective. This affects the closed-loop control. In this way, it is possible to adapt the open-loop control to the requirements of the machine.



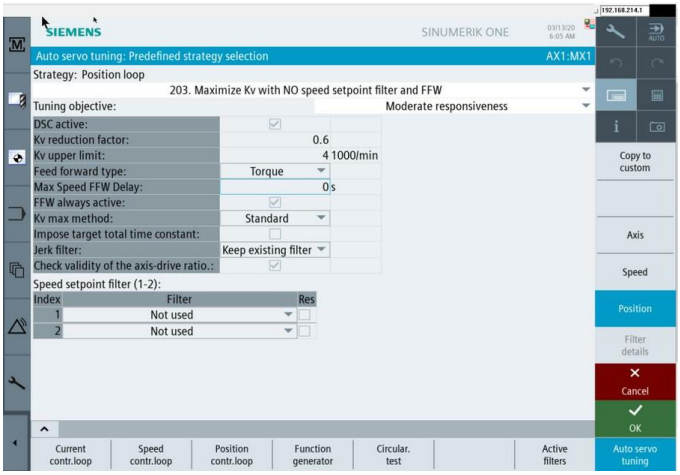
Selection of the tuning objective

We recommend that the **"Moderate responsiveness"** objective is used for standard applications. It represents a compromise between dynamic control response and robust settings. The **"Conservative / robust responsiveness"** option should be used when high tolerances are to be expected in the components used and later in the production, or when the masses, e.g. for a rotary table, vary greatly. These settings increase the controller stability but have a negative effect on the dynamic response and the reaction to disturbance.

Notes:

The last objective **"Maximum responsiveness"** is particularly suited for machines with high positioning accuracy or higher machining speeds. However, the robustness of the control usually decreases, and the axis becomes unstable when changes are made in the mechanical system or loading.

It may be that the preset objectives are not sufficient for the desired axis behavior. The strategies can be adapted here based on the specifications (press the "Adapt strategy" softkey and confirm the prompt with "Yes"). However, these settings should only be made with adequate background knowledge. They are therefore not discussed in more detail in this module.
Note: The description of the tuning objectives applies in the same way to the position controller. The set tuning objective for the speed controller and the position controller is always identical. If the objectives are to be set differently, this must be performed by adapting the strategy (see above).



As already shown with the speed controller, there are different strategies to adapt the position controller. The strategy **"203. Maximize Kv with NO speed setpoint filter and FFW"** has proved to return respectable results for most applications.
It is strongly recommended that the feedforward control (FFW) be activated and the feedforward type be set to **"Torque"**. This greatly improves the positioning accuracy. This is already preset in the above strategy.

Position controller settings



Setting parameters
In order to be able to understand the settings better, the parameters used for the tuning are explained briefly here:

Speed controller

- Aggressiveness: Determines the settings of the controller gain (Kp) and the integral time (Tn) depending on the specified stability limits
  0: maximum stability
  1: maximum aggressiveness
- Minimum integral time: Prevents AST setting an integral time that is too low. If the determined value is lower than the minimum integral time, then this is used instead of the determined value. This provides more stability, particularly when mass changes are expected (e.g. for the main spindle of a turning machine).
- Amplitude/phase reserve: Stability criteria for control loops. The higher the more stable.
- Maximum bandwidth: Limitation of the dynamic controller response.

Position controller

- Kv factor reduction: The maximum Kv factor (= position controller gain) calculated by the AST is reduced by the specified value. The maximum Kv factor is the value at which the position control loop just does not overshoot after a sudden excitation.
- Kv upper limit: Absolute limitation of the Kv value. The gain is limited here in order to achieve a well damped responsiveness to positional faults and to be robust against changes in the mechanical system.
- Feedforward type: The feedforward control is used to reduce the contour deviation on finished components. The setpoint is reached significantly quicker with the feedforward control. Two types of feedforward control can be selected:
  - Speed feedforward control and
  - Torque feedforward control

  whereby the latter is recommended because the contour deviations are always very small here. The only restriction: the smallest moment of inertia (e.g. empty table) should always be used for the tuning.
  For directly driven axes only: note the moment of inertia at the first tuning with robust tuning objective and set it for the second tuning (with real loads).
- Jerk filter: Recommended for mold-making applications. It is important to produce a surface that is as smooth as possible. Quick setpoint changes can be reduced by the filter and the introduction of oscillations avoided.
  The filter time constant (AX_JERK_TIME) must be set manually by the user.

Notes:

*Note: The drive must support **telegram 136** in order to be able to use the torque feedforward control. It must also be activated in each drive in:*

- *MD13060[x] = 136*
- *P922 = [136] SIEMENS telegram 136, PZD-15/19*

Executing the Automatic Servo Tuning function:



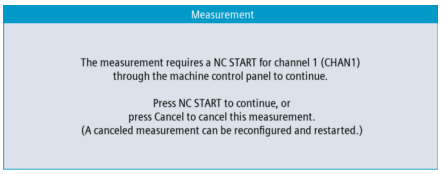


The Automatic Servo Tuning function can now be executed.

Make sure that the axis to be tuned is in a safe position. If this is not the case, you can move the axis to a safe position with the Jog keys on the MCP. Press "OK" to confirm.



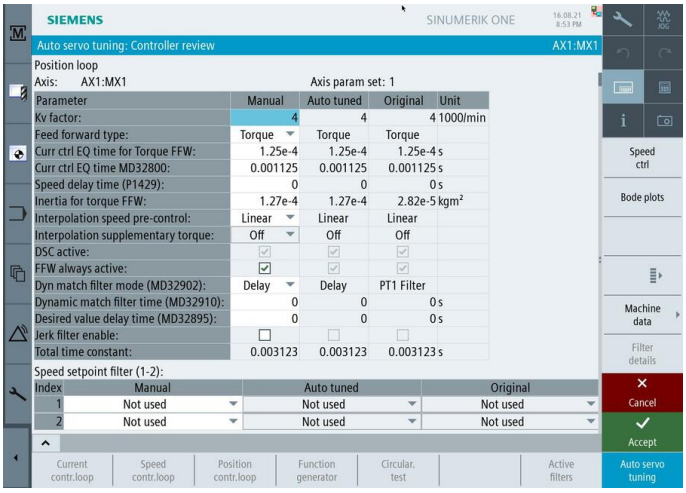
The following dialog box then appears.



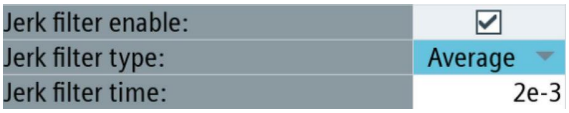


Various screens appear during the automatic tuning procedure. "CYCLE START" must be pressed several times.

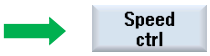
After completing the tuning, the following screen appears in which the new values and the original values can be checked. This example is from the position control.

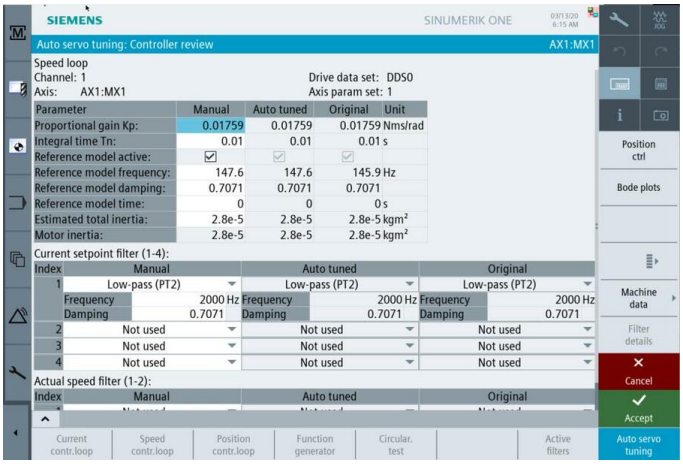


It is recommended that the jerk filter be set active in the axis for mold and die applications. The "Jerk filter enable" checkbox must be activated. "Average" should be set as the jerk filter type. A *Power-On reset* may have to be performed after the tuning (of all axes). If this is the case, a corresponding prompt is displayed.



The values of the speed controller can also be viewed and, if required, adapted.



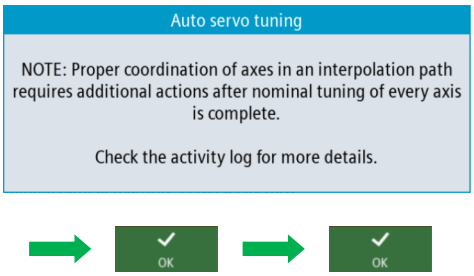


The values can now be accepted and saved:

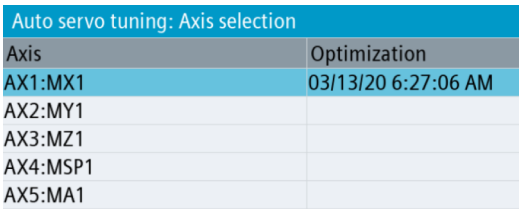


Notes:

The dialog box that now appears indicates that the other axes also have to be tuned for a good tuning result and tuning of the interpolation path later :



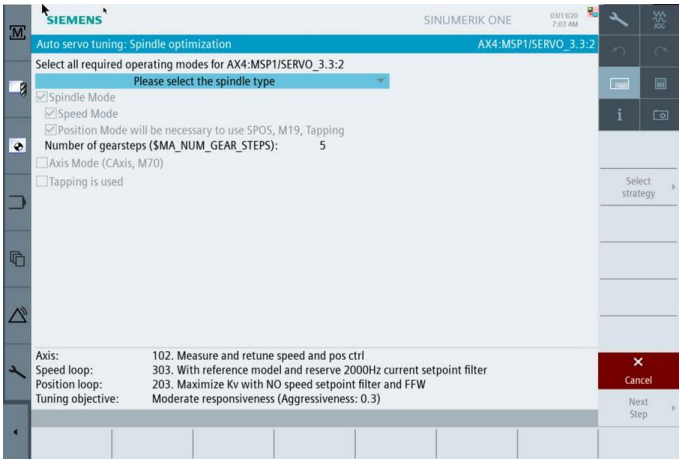
The time and date of the tuning now appear.



The other axes should then be tuned in the same way.

Spindle Tuning strategy

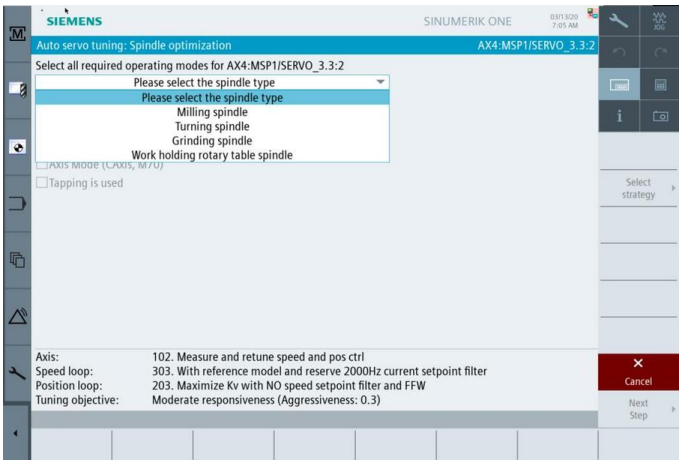
The Tuning strategy for the Spindle is different



The type of Spindle that is to be tuned can be selected.

- Milling
- Turning
- Grinding
- Work holding rotary table spindle

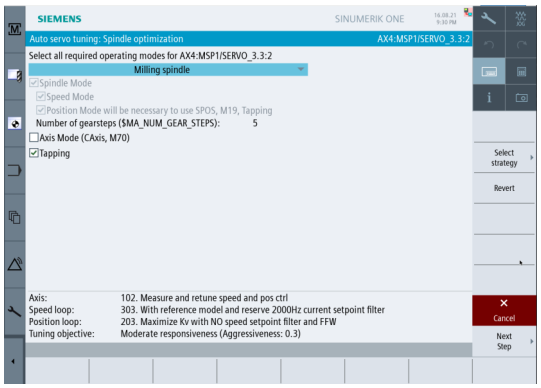
The tuning of a standard Milling or Turning spindle will be less aggressive than a linear axis



Once the Spindle type has been selected further spindle operating modes can be selected.

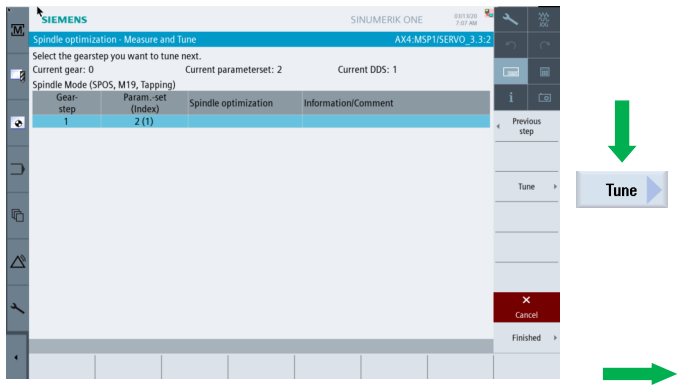
- Position Mode it will be necessary to use SPOS, M19
- Axis mode (C Axis, M70)
- Tapping is used

Selecting the different spindle types/tuning modes affect the aggressiveness of the tuning e.g. Work holding rotary table spindle would normally require a more aggressive speed/ position controller setting than a standard Milling or Turning spindle..

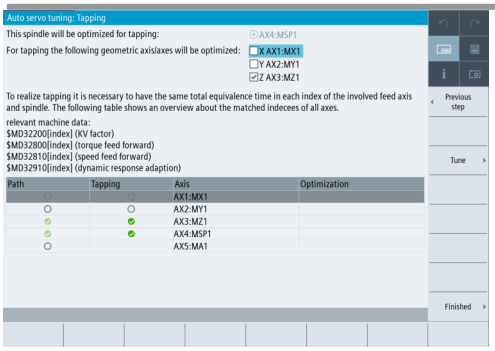




It is also possible to tune different gear ranges

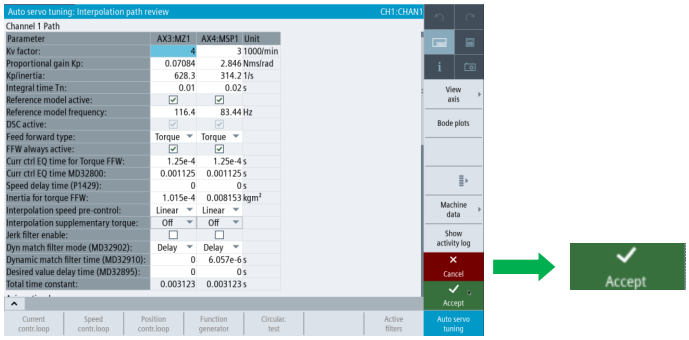


Notes:



After the Spindle tuning has finished it is possible to tune the interpolation path between the spindle and an axis that will be involved in tapping. In this example the Spindle and Z axis are included in the path tuning.





Tuning of the interpolation path

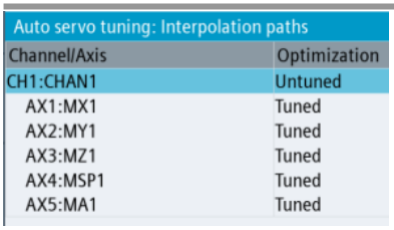
In order to achieve a contour error as small as possible in multi-axis machining, all the relevant axes must have the same following error.

After tuning the individual axes, the axes have different controller settings which are the optimum for the respective axis.

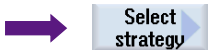
In order to ensure a uniform following error, the settings must be changed to match the axis with the slowest response; this includes the Kv factor and the feedforward control mode. (The parameters selected for the interpolating axes are shown on the next page.)
Select the functions in the tuning menu in order to optimize the path interpolation:



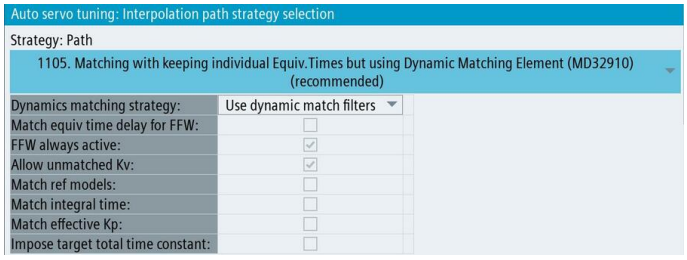
Select the channel that contains the axes to be tuned:



The interpolation path strategy can now be specified.



Strategy "1105. Path optimization with dynamics matching element" is recommended.





Description of the strategies:
1105. Dynamics matching element (recommended)
The equivalent time constants of the single axis tuning are retained.
Parameter DYN_MATCH_TIME is used to adapt the following error.

With speed feedforward control, a reduction of the Kp factor is possible, but not necessary in most cases. The noise suppression worsens when the Kp factor is reduced. The following strategies can be selected:

1102. Moderate adaptation
Effective Kp factors are adapted.
Only relevant for speed feedforward control.
Kp values can be reduced by up to 50% of the original value.
Kp factor of the main spindle is not reduced in order to correspond to a lower effective Kp factor of a feed axis.

1103. Without reduction
Effective Kp factors are not adapted.
With torque feedforward control, an adaptation of the speed controller gain (Kp) is not necessary.

1104. Custom adaptation
Users can adapt the interpolation path strategy according to their requirements in the lower selection field.

General settings for path interpolation
Note: The following specified parameters and adaptations only serve for better understanding. Manual adaptation is normally not necessary.

Dynamic stiffness control**:** Is activated or deactivated in all axes. If activated, it enables a higher position control gain (Kv factor).

Notes:

Feedforward control: If all axes have been tuned with torque feedforward control, this setting is also retained for the path interpolation. If the feedforward control mode is mixed (3 or 4), FFW_MODE=3 is set for all axes.
If an axis has been tuned without feedforward control, FFW_MODE=0 is set for all axes.

Speed setpoint filter: If setpoint filter 1 and DSC and feedforward control are used, this speed setpoint filter has no effect on the following error (and therefore also no effect on the path interpolation). The filter is recommended to stabilize the position controller (increase of the Kv factor).

Jerk filter: Except for the "Dynamics matching element" strategy, the settings for the jerk filter must be the same in all axes.

DYN_MATCH_TIME is used to align different following errors between the axes.

Speed feedforward control
Tn is adapted in all axes.
Reference model (speed control) is adapted.
Kv factor is recalculated and the lowest value is set for all axes.

Torque feedforward control (recommended)

Tn is not adapted.
Reference model (speed control) is not adapted.
Kv factor is not adapted
→ The optimum axis values are retained.

Performing the tuning

Note: No measurements are required for this step.

The actual tuning can now be performed:

→ Edit & tune

The axes that are to interpolate together can be selected.

In the example, these are the MX1, MY1, MZ1, MSP1 and MA1 axes:

Auto servo tuning: Interpolation paths

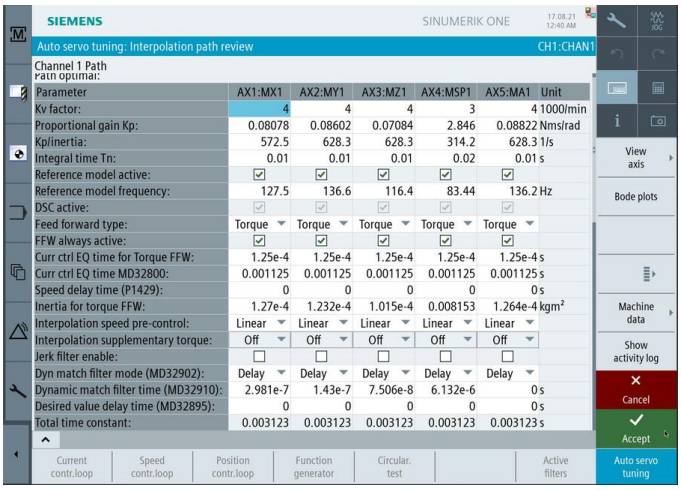
Select axes for matching

Axis	Optimization
☑ AX1:MX1	03/13/20 6:27:06 AM
☑ AX2:MY1	03/13/20 6:32:09 AM
☑ AX3:MZ1	03/13/20 6:34:38 AM
☑ AX4:MSP1	03/13/20 7:12:37 AM
☑ AX5:MA1	03/13/20 7:02:45 AM

Start the tuning:

→ ✓ Tune

After some calculations, the result of the tuning is displayed.
Press the "Accept" softkey to activate the settings.



→ ✓ Accept → ✓ OK

Auto servo tuning: Interpolation paths

Channel/Axis	Optimization
CH1:CHAN1	03/13/20 7:22:44 AM
✓ AX1:MX1	03/13/20 7:22:44 AM
✓ AX2:MY1	03/13/20 7:22:44 AM
✓ AX3:MZ1	03/13/20 7:22:44 AM
✓ AX4:MSP1	03/13/20 7:22:44 AM
✓ AX5:MA1	03/13/20 7:22:44 AM

The automatic servo tuning is now completed.

Further checks and settings are required for a complete tuning of the drives. They are described in the following.

→

Notes:

Maximum acceleration

The acceleration limit of the axis is important to protect the motor, the converter, and the mechanical system of the drive train from overload. This limitation informs the NC during path calculation how much acceleration capacity the axis has. This enables the NC to calculate a path for all situations that can be adhered to by all axes involved without resulting in distortion due to torque limitation of an axis.

In order to determine the maximum permitted acceleration for an axis, the following procedure is recommended:

Note: Ensure that a low value of Axis Jerk MD32431 is not limiting acceleration

1.) Determining the axis' inertia

↓

2.) Determination of the torque maximum limit

↓

3.) Determination of friction and holding torques

↓

4.) Determining of the inertia with maximum loading

↓

5.) Calculation of the maximum acceleration

↓

6.) Check of the determined values

Basic calculations

The physical correlation will be explained briefly because of the need to do some calculations (please note that all calculations here are performed with respect to a rotary drive with a gearbox and a lead screw).

The motor turns with the speed n_M and delivers the torque M_M. That is transferred to the load side by the gear box with the gear factor $i = n_L / n_M = M_M / M_L$. A lead screw with a pitch of h transforms this rotary movement, which is applied to through the slide to the workpiece: $F'_L = M_L * 2\pi/h$ resp. $v'_L = n_L * h/2\pi$.

In order to convert the impact of masses, use is made of the mass moment of inertia. The equations are quite similar to the ones mentioned above; only the screw pitch and the gear factor are squared:
$m'_L = J_M / (h/2\pi * 1/i)^2$.

The mass inertia of an axis will be determined during Automatic Servo Tune and written in MD32650. If the task is to get the inertia of a loaded axis, the additional mass must be converted from the load to the motor side and added to this value.

Last but not least the acceleration is determined by rearranging the formula F = m * a:

$$\mathbf{a = F'_L / m'_L}$$

1. Determining the axis' inertia

The inertia of an axis is automatically determined by AST. 0.9 times this value will be saved to the machine date **MD32650**.

2. Determination of the maximum torque limit

First of all the current limit of the drive (i.e. minimum out of current limit of motor and inverter) has to be identified. Multiplied with the constant k_T (p316) you get a maximum electric torque. Depending on the planned torque reserve (in case friction increases by the time of usage) one only calculates with about 80% of this value.
Now the torque limit of the mechanics has to be calculated:

- Gear box: output sided limitations have to be converted to the motor side: $M_M = F_L \times h/(2\pi) \times 1/i$;
  in the case of calculating with torque the factor *h/(2π) must be left out*; an input sided limitation of the gear box does not have to be converted
- Forces on bearing and joining elements may serve as a limitation and therefore have to be taken into consideration

The maximum torque limit is the smaller value of electrically available and mechanically permitted torque.
By conversion of the result onto the load side the maximum permitted acceleration for both, drive and mechanics, can be determined. This value should be put into MD32300 and activated with **cf** before continuing with the further steps.

3. Determination of friction and holding torque

As friction forces have to be overcome and in case of a suspended axis a constant torque (a so called holding torque) has to be put against gravity acceleration, the permitted maximum acceleration has to be again reduced. Both torques can easily be identified by a trace. For this purpose the axis has to be jogged slowly with constant speed and the consumed torque-producing current i_q has to be recorded:

The following program can be used for this purpose (in case of a suspended axis it must be to be moved upwards!):



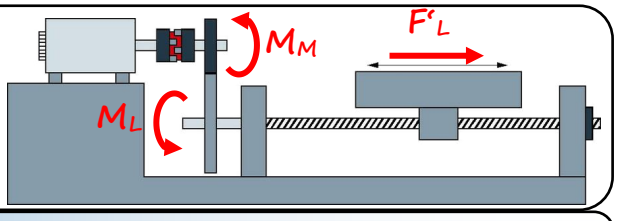

```
SOFT¶
$AN_SLTRACE=0; Reset Start Trace-Trigger¶
G0 X0¶
$AN_SLTRACE=1; Start Trace-Trigger¶
G01 X20 F200; slow constant feed¶
M30¶
```

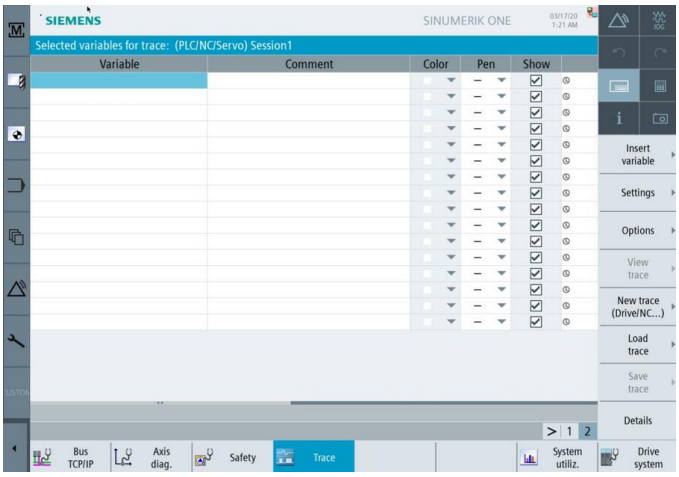
Now open the trace environment:



Notes:

In all calculations in this chapter, quantities that refer to the motor side are marked with the index M, e.g. M_M; rotary quantities which refer to the load side with the index L, e.g. M_L and translational variables which refer to the load side are marked as primed quantities with the index L, e.g. F'_L.

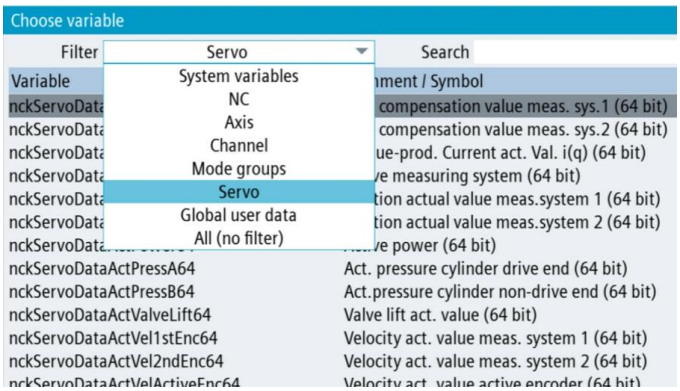




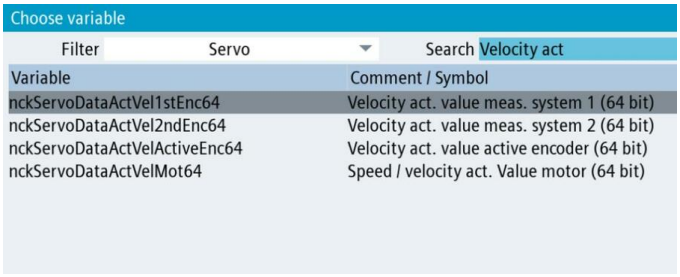
After that add the following variables:

- **Nck/ServoData/nckServoDataActCurr64**
  (torque –producing current actual value)
- **Nck/ServoData/nckServoDataCmdPos2ndEnc64**
  (position setpoint)
- **Nck/ServoData/nckServoDataActVel1stEnc64**
  (velocity actual value).

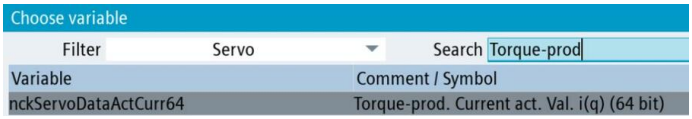




In order to find the signals you can make use of the search filter function.

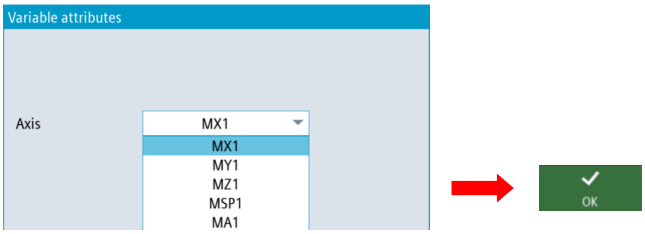


The first signal we need is the actual value of the torque-producing current:





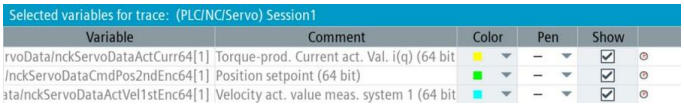
The axis must be chosen, e.g. X-axis:



The selected variable now appears in the list:



Do as well with the two other variables.



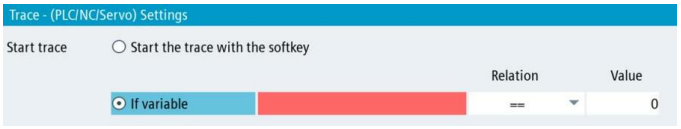
When all variables have been added, you must still perform some settings:

Trace duration and starting conditions:

The trace can be started manually or automatically. Furthermore the time recorded can be set:



In the following example the trace will be started by a command in the part program:
The trace start variable can be chosen easily by using the “Quick list“ softkey:

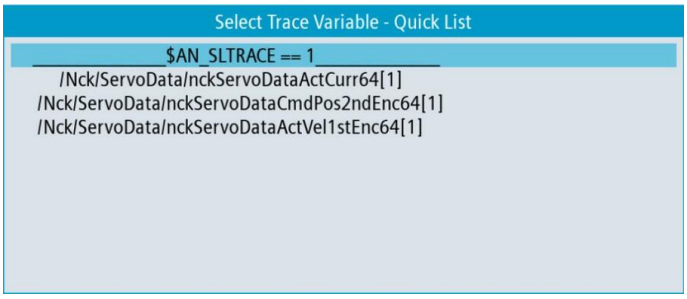


Notes:

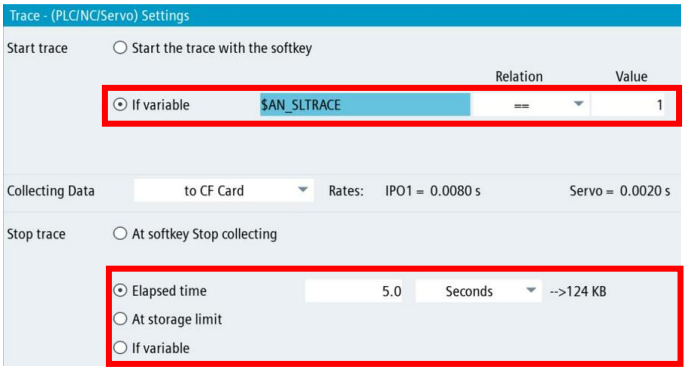


Chose the option **$AN_SLTRACE==1**.
Moreover it is possible if one of the selected signals (using “Quick List“) or any other system variable (using “Choose variable“) reaches the specified condition.

In this case $AN_SLTRACE == 1 is set:





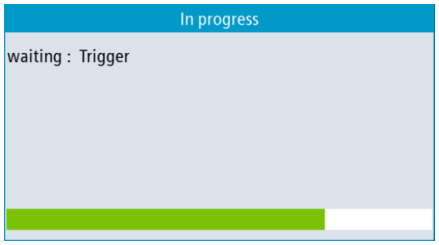




Start/initialize trace:

The trace can be started/initialized by using the “Start trace” softkey.
That depends on your selection in the “Settings“ screen.
In our example the trace is started by a command in the part program.
Hit the “Start trace“ softkey, to initialize the trace:



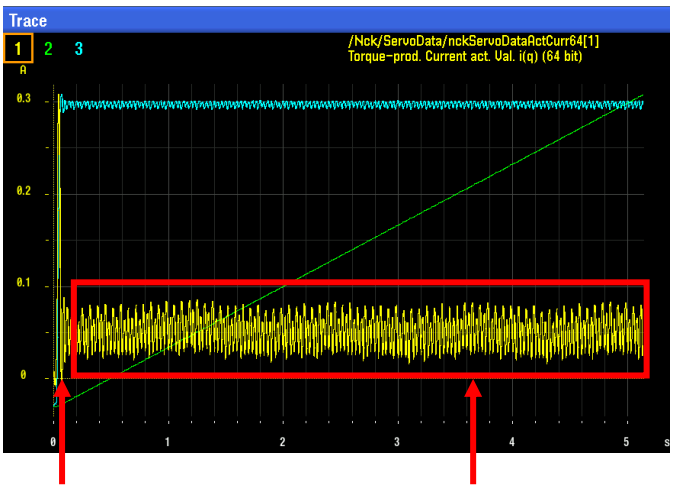


The trace is now initialized and will be started as soon as the trigger is set by the part program.

Now start the trace:



The three traces will be displayed together; the scaling can be adjusted: Click “Zoom“ and choose “X-time axis“ and “Y-value axis“ (both softkeys are now highlighted in blue). Now choose “Fit overlaid“.
Both range and the momentary current actual value will be displayed:



Current peaks when starting out of static friction

Current consumption during constant movement

The peak at the beginning shows the current during the start out of static friction. This value is not relevant to these considerations.
With the help of the cursor the maximum current of the area with constant velocity can be determined.
The determined current can now be, again by means of the torque constant k_T, converted into a torque value $M_{M,FH}$.

Notes:

4. Determining the inertia with maximum loading
Until now only unloaded axes were taken into consideration. If however a work piece is clamped, the motor needs more current to achieve the commanded acceleration. If we set up the formerly determined values, the motor would quickly reach current limitation when trying to provide the corresponding torque. So, the maximum torque has to be adapted to the **maximum loading**.

It is important to figure out realistic values of the loading and possibly set up additional, loading dependent, datasets because the maximum acceleration (depending on the mass of the additional loading) can be very small. Because of this the desired acceleration for this machine can possibly not be reached any more. In the extreme case ($m_{add.loading} = V_{processingarea} \times \rho_{Material}$) resulting masses will realistically never be reached on a real machine. This heavy masses would lead to a very slow acceleration and by this to longer machining times.

5. Calculation of the maximum acceleration
Finally, the determined values have to be combined and converted to the load side (the maximum axis acceleration is saved with regard to the linear movement in the controller).

The maximum torque on the motor side is calculated as $\mathbf{M_{M,max,tot} = M_{M,max} - M_{M,FH}}$ and can be converted to a maximum force $\mathbf{F'_{L,max}}$ on the load side with help of the given formula. The maximum total inertia is determined as $\mathbf{J_{tot} = J_{empty} + J_{maxLoading}}$ and converted to a total mass $\mathbf{m'_{L,tot}}$ on the load side.

The maximum acceleration can be determined by the interrelationship F = m × a.

This value should be put into MD32300 and confirmed with **cf**.

Note: In order to keep the controller from limiting the current further, the parameter p500 in the drive parameters has to be set to “[101] Feed drive“ for feed drives!

6. Checking the determined values
To make sure that the set maximum acceleration is not too high, it can be verified.

It is crucial that the axis is loaded with a realistic workpiece!

For this purpose a new part program has to be generated. This program moves the axis within a given distance so that the programmed rapid traverse velocity can be reached, e.g. 300 mm.




```
SOFT¶
$AN_SLTRACE=0; Reset Start Trace-Trigger¶
LAB:¶
G0 X0¶
$AN_SLTRACE=1; Start Trace-Trigger¶
X200¶
GOTOB LAB¶
M30¶
```

Then open the trace environment:



Add the following variables to the trace as described previously:

- **Nck/ServoData/nckServoDataActVel1stEnc64** (speed actual value)
- **Nck/ServoData/nckServoDataDrvLoad64** (Load (m_cmd / m_cmd,limit))
- **Nck/ServoData/nckServoDataActCurr64** (torque producing current act. value)

Now set the variable **$AN_SLTRACE==1** as starting condition for the trace and set the recording duration to 5 sec.

Start the trace:



and adapt the display:

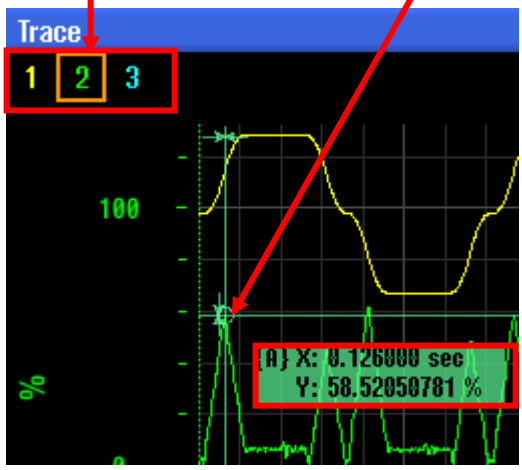


Now the maximum load can be measured in the diagram with the help of the cursors. This value should not exceed 80%. If the value is higher the calculations have to be checked.

Notes:

Trace selection

The cursor shows that the reached maximum load was at about 58.52%.



Example

Relevant machine data

$I_{max,\ MoMo}$	p67	Max. current of the power module
$I_{max,\ motor}$	p323	Max. motor current
k_T	p316	Torque constant
i	(MD31060 / MD31050)	Gear factor
h	MD31030	Spindle pitch
J_{empty}	MD32650	Mass moment of inertia of the axis determined by AST
-	p500	Technological application
a_{max}	MD32300	Maximum acceleration

The example has been performed on the Z axis of a milling machine.

Step 1: Determination of the axis inertia

J_{MD} = 0.0054 kg m^2 (MD32650)
J_{empty} = 1.11 × J_{MD} = 0.0060 kg m^2
(Notice: AST must have been performed for this axis before!)

Step 2: Determination of the torque upper limit

$I_{max,MoMo}$ = 36 Aeff (r67)
$I_{max,motor}$ = 52 Aeff (p323)
k_T = 1.05 Nm / Aeff (p316)
h = 10 mm (MD31030)
i = 1 (MD31060 / MD31050)

Power module current limit ($I_{max,MoMo}$) ≤ motor current limit ($I_{max,motor}$), that means, continue to use $I_{max,MoMo}$ for your calculation. Consequently, the maximum electrical torque M_E (80% of the available torque) results:

$M_{el,max}$ = 0.8 × k_T · $I_{max,MoMo}$ = 0.8 × 1.05 Nm/A × 36 A =
= 30.24 Nm

In this example, the mechanical limitation is defined via the gearbox. The machine specification stipulates that the maximum input torque $M_{me,max}$ may not exceed 29.1 Nm.

$M_{me,\ max}$ < $M_{el,max}$: The mechanics is limiting. $M_{M,max}$ = $M_{me,max}$

Now determine the maximum permissible linear force:

$F'_{L,\ max}$ = $M_{M,max}$ × (2π / h) × i = 29.1 Nm × (2π / 0.01 m) =
= 18275 N = 18.3 kN

Then determine the mass on the load side:
$m'_{L,empty}$ = J_{empty} / (h/2π × 1/i)^2 = 0.0060 kg m^2 / 2.60 × 10^{-6} m^2 =
= 2366 kg

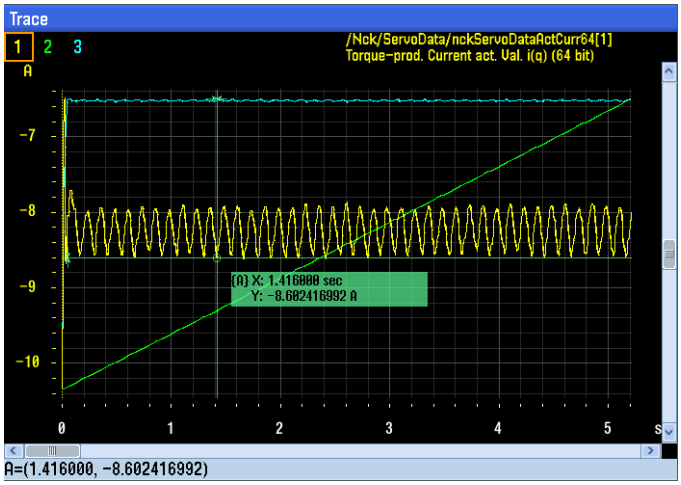
This yields the maximum permissible acceleration:
a_{max} = $F'_{L,max}$ / $m'_{L,empty}$ = 18.3 kN / 2366 kg = 7.72 m/s^2

Enter this value in MD32300 (to ensure that the maximum permissible limit values for mechanics and electronics are not exceeded when performing further tests) and confirm with cf.

Step 3: Determination of the friction and holding torque
As described in the theoretical part, perform the trace (which is displayed here with the Z axis instead of the X axis). The trace is displayed on the next page.

You can now read off the maximum current from the diagram during constant traversal and convert it into a torque value:
$I_{M,RH}$ = |- 8.6 A| = 8.6 A
$M_{M,RH}$ = k_T × $I_{M,RH}$ = 9.0 Nm

Notes:



Step 4: Determination of the maximum load

As, on the milling machine, the Z axis only comprises the tool, we assume a maximum additional load of approx. 30 kg:

$m'_{L,load}$ = 30 kg

Together with the mass of the empty axis determined before, this yields:

$m'_{L,total} = m'_{L,empty} + m'_{L,load}$ = 2366 kg + 30 kg = 2396 kg

Step 5: Calculation of the maximum acceleration

Now, you must include the torques determined in your calculation:

$M_{M,max,total} = M_{M,max} - M_{M,RH}$ = 29.1 Nm - 9 Nm ≈ 20.0 Nm

Calculation of the resulting torque on the load side:

$F'_{L,max,total} = M_{M,max,total} \times (2\pi / h) \times i$ = 20.0 Nm × (2π / 0.01 m) =
= 12.44 kN

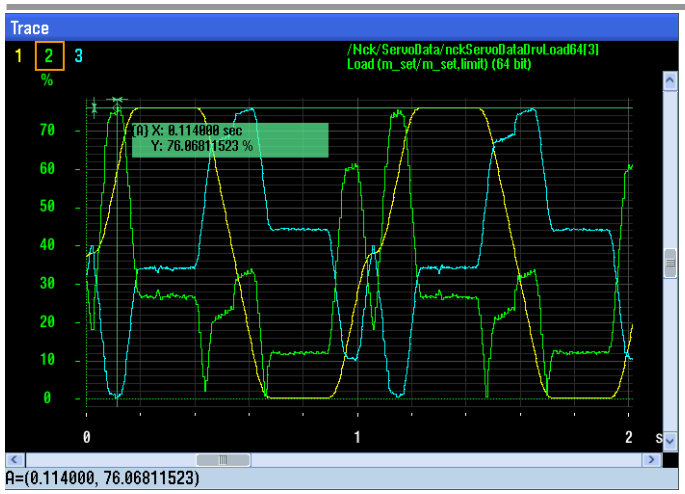
Together with the load mass, this yields the following acceleration:

$a_{max,total} = F'_{L,max,total} / m'_{L,total}$ = 12.44 kN / 2396 kg = 5.2 m/s²

This value is now entered in MD32300 and confirmed.

Step 6: Check of the values determined
Finally, check the settings. The trace is created, as declared in the theoretical part. The result (see next page) shows that the utilization is 76 % and thus close to the requested value of 80 %.

The optimization process is thus completed.



Step 7: Transfer values to other Axial machine data Axis-specific dynamic response settings - Index[0-5]

Example:

MD32300 $MA_MAX_AX_ACCEL[0] = 4 (DYNNORM)
MD32300 $MA_MAX_AX_ACCEL[1] = 4 (DYNPOS)
MD32300 $MA_MAX_AX_ACCEL[2] = 4 (DYNROUGH)
MD32300 $MA_MAX_AX_ACCEL[3] = 2 (DYNSEMIFINISH)
MD32300 $MA_MAX_AX_ACCEL[4] = 1 (DYNFINISH)
MD32300 $MA_MAX_AX_ACCEL[5] = 1 (DYNPREC)

Lower acceleration is specified for semi/finishing groups.

Fault cases

If the utilization is permanently above the requested percentage value, you must reduce the acceleration.

As the friction value of the individual components changes in the course of time, it may become necessary to determine the friction torques again. This is shown by a higher utilization than requested, or, in an extreme case, by a trigger of the contour monitoring. In this case, you must perform the calculation again, starting with step 3.

A larger workpiece mass than the one assumed during the test might have been clamped. In this case, you must repeat the calculation as from step 4.

Notes:

Axis jerk

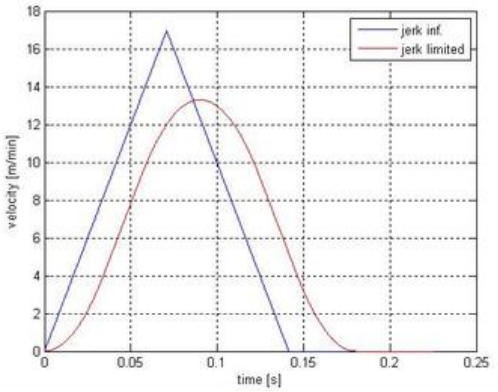
After completing the automatic servo tuning, the axis acceleration should be set to meet the requirements of the machine manufacturer (a check is made later during the torque utilization check).

Large acceleration changes frequently occur with dynamic axis movements. If these acceleration changes (generally called jerk, unit: m/s³) are very large, oscillations can occur on the machine axis and thus result in an inexact workpiece contour. In order to reduce this effect, the control can reduce the jerk. This lengthens the time during which the acceleration changes.
In the chart on the right, the active jerk is indicated by the rounded corners of the velocity curve.

In the part program, the jerk limitation can be activated with the "SOFT" command and deactivated with "BRISK". If neither of these commands is executed, the control uses the default value. This is stored in the channel MD20150[20], whereby 1 normally activates the jerk and 2 (delivery state) deactivates it.

The following section explains how the jerk limitation is configured in the control.

Example: Jerk effects on the velocity profile



without jerk-limited acceleration

with jerk-limited acceleration

Preliminary consideration

For an optimum setting of the jerk, the axis is traversed using a part program and the positioning behavior monitored with the trace function. If the oscillations are too large, the jerk limitation must be adjusted. This procedure is repeated until the curves have obtained the desired shape.

If the machine axis is using a direct measuring system then the actual position value can be determined easily. However, if the position control loop is closed via a motor encoder then it is much more difficult to display the position values (e.g. via an external measuring system). In this case, the measured position curves serve as the minimum criterion: overshoot on the motor encoder must be avoided.

Usually, jerk limitation values between 20 and 100 m/s^3 are set (corresponding values for rotary axes).

There is no need to reach the programmed feedrate. When considering oscillations, short traversed distances (e.g. 3 to 10 mm) often have a greater effect on the mechanical excitation than longer traversing distances and should therefore be considered as worst case. A dwell time (e.g. 0.5 s) should be maintained when the position setpoint is reached.
The first programmed feedrate should be set to half the maximum axis feedrate. The response behavior for various override levels (recommended: 10%, 20%, 50%) up to the programmed velocity (i.e. 100%) can now be considered. Finally, the trace should be traversed with maximum axis feedrate.

Part program for the jerk check
Depending on the configuration of the machine used, the following NC program can be used in Automatic or MDA-operation. The program contains commands for the activation/ deactivation of the trace trigger:

```
FFWON¶
SOFT¶
$AN_SLTRACE=0; Reset Start trace trigger¶
LAB:¶
G01 Y-110 F10000¶
G04 F0.5¶
$AN_SLTRACE=1; Start trace trigger¶
Y-115¶
G04 F0.5¶
GOTOB LAB¶
M30¶
```

Machine data:

To check the mechanical behavior without the influence of the position controller, the servo gain factor (position control gain) and the equivalent time constant of the feedforward control should be set to 0.

Position tolerances should also be expanded in order to avoid errors during the test.

MD 32200 POSCTRL_GAIN = 0
MD 36012 STOP_LIMIT_FACTOR = 100
MD 36400 CONTOUR_TOL = 20

Notes:

Selecting the trace signals:
The characteristic curves of the position setpoints and actual position values are required to set the jerk (see previous section for the precise procedure):

- Position setpoint (64-bit)
  nckServoDataCmdPos2ndEnc64
- Actual position value (64-bit)
  nckServoDataActPos2ndEnc64



After inserting the variables, open the settings:



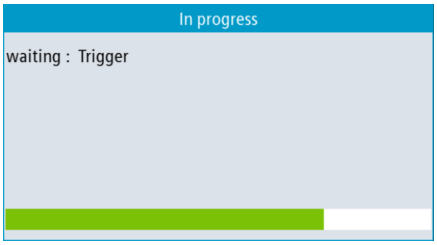
Start trace: Variable $AN_SLTRACE == 1
Stop trace: When the time has expired, 5.000 seconds



Starting/initializing the trace:

The trace is started/initialized via the "Start trace" softkey.
This depends on the selection in the "Settings" window.
In our example, the trace is started via the command of a part program.
Press the "Start trace" softkey to initialize the trace:

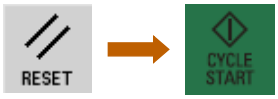




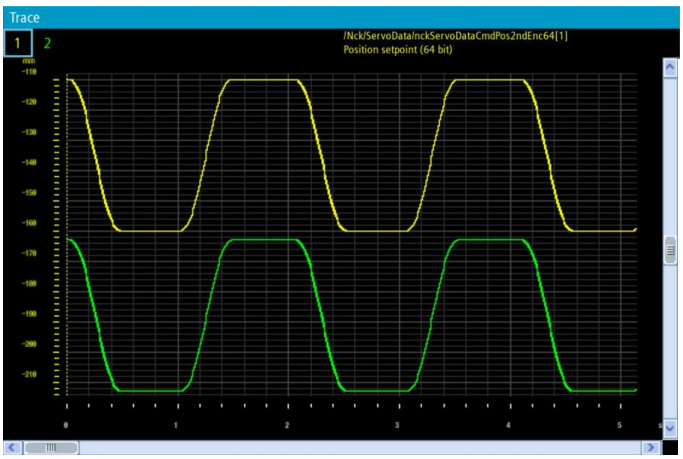
The trace has now been initialized and is started as soon as the trigger has been set via the part program.

Viewing the trace results:

Start the previously entered program:



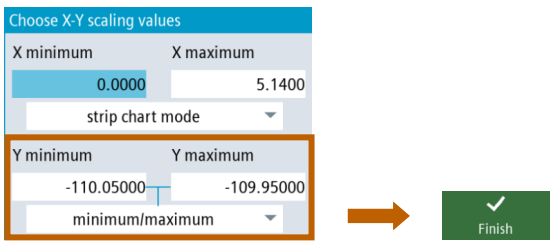
The axis moves and the selected signals are monitored for five seconds. The result is then shown.



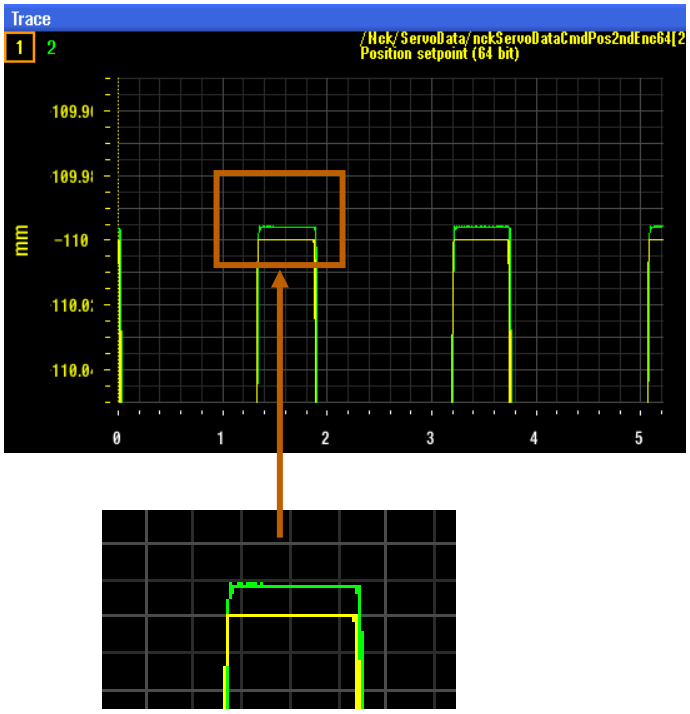
Setting the trace scale:
The part of the trace in which the axis reaches the set position must be viewed. The trace scale must be adapted for this reason:



The following example shows the setting of the scale in order to view the axis positioning at -110 mm.
The entered values result in a 100 micrometer window around the target position of -110 mm.
A different scale may be required depending on the machine/ axis used.
The X minimum / X maximum values can also be adapted when required.



Notes:



Set position (yellow) and actual position (green) do not match because the position controller is not active.

Tuning the jerk setting:
The jerk must be set so that no oscillation is excited at the actual position value. This is particularly the case when the maximum jerk value has been set too high.
A jerk value that is set too low can also have negative effects; the positioning of an axis takes a relatively long time in this case.

Without the position controller, the axis is positioned without overshoot by the control (overshoot must therefore come from the mechanical system). However, a positioning error results because the axis is only traversed with the feedrate signals.

An overshoot when the position is reached is caused by the mechanical system and can be minimized by using the jerk limitation.

The following examples show the effects of different maximum jerk values on the positioning of an axis.

However, the following machine data must be checked (note the old values first):

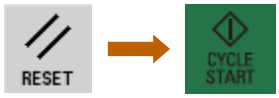
MD 32800 EQUIV_CURRCTRL_TIME = as tuned
MD 32200 POSCTRL_GAIN = 0
MD 32610 VELO_FFW_WEIGHT = 1.0
MD 32620 FFW_MODE = 3 or 4

Example - jerk too high:

Axis machine data 32431 is used to set the jerk.

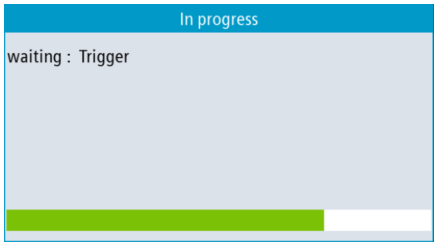
Example - jerk too high:
MD 32431 MAX_AX_JERK = 600

Restart the part program:



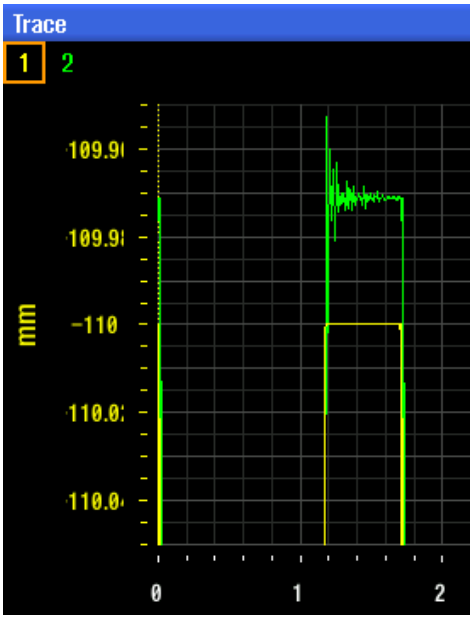
Start the trace:





This chart shows the rise in jerk until overshoot occurs due to the mechanical system.

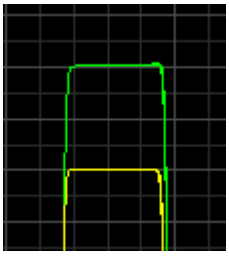
Correct the oscillation of the actual position value by reducing the MAX_AX_JERK.



Notes:

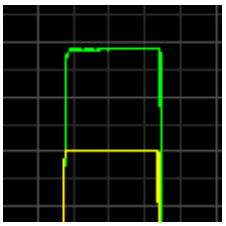
Example - jerk too low:
MD 32431 MAX_AX_JERK = 2

The positioning contour is rounded when the jerk is too low.



Example - jerk tuned:
MD 32431 MAX_AX_JERK = 65

When the jerk is tuned, there is little or no overshoot and the positioning contour is sharp.



Positioning with feedforward control and active position contoller:

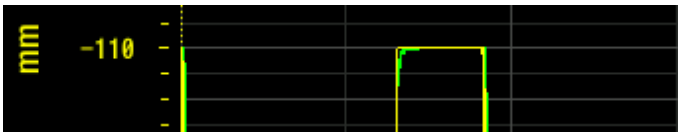
If the jerk has been tuned, the position controller and the feedforward control must be activated. Set the machine data as required:

MD 32200 POSCTRL_GAIN = as noted previously
MD 32610 VELO_FFW_WEIGHT = 1.0
MD 32620 FFW_MODE = as noted previously
MD 32800 EQUIV_CURRCTRL_TIME = as noted previously
MD 32431 MAX_AX_JERK = 65

MD 20150[23] = 2 The feedforward control can then be enabled as default setting.

MD 36012 STOP_LIMIT_FACTOR = original value
MD 36400 CONTOUR_TOL = original value

Example:
Good positioning without overshoot when the feedforward control and jerk have been tuned correctly:



Transfer values to other Axial machine data Axis-specific dynamic response settings - Index[0-5]

Example:

MD32420 $MA_JOG_AND_POS_JERK_ENABLE = 1
MD32430 $MA_JOG_AND_POS_MAX_JERK = 65
MD32431 $MA_MAX_AX_JERK[0] = 65
MD32431 $MA_MAX_AX_JERK[1] = 65
MD32431 $MA_MAX_AX_JERK[2] = 100
MD32431 $MA_MAX_AX_JERK[3] = 65
MD32431 $MA_MAX_AX_JERK[4] = 25
MD32431 $MA_MAX_AX_JERK[5] = 10
MD32432 $MA_PATH_TRANS_JERK_LIM[0] = 65
MD32432 $MA_PATH_TRANS_JERK_LIM[1] = 65
MD32432 $MA_PATH_TRANS_JERK_LIM[2] = 100
MD32432 $MA_PATH_TRANS_JERK_LIM[3] = 65
MD32432 $MA_PATH_TRANS_JERK_LIM[4] = 25
MD32432 $MA_PATH_TRANS_JERK_LIM[5] = 10

Axis MD 32430 = Jog and Position Maximum Jerk is enabled permanently via MD 32420 = 1. MD 32430 is active only for non-interpolating "positioning axis", "reference point", “manual pulse generator” and "Jog" modes. MD 32430 can be set to suit the particular axis required response in the above modes.

Notes:

Circularity test

The circularity test of the controller is used to check the interpolation of the axes working together. This function measures a circle, either with the direct measuring system or the motor measuring system. An alignment of the machine mechanical system is not taken into account in the result. This enables the commissioning engineer to clearly separate the problems of the controller tuning from the mechanical problems. It is recommended to verify the machine geometry by using an inspection square (granite master square).

The following **position error compensations should be switched off** when this procedure is being performed:

- Backlash on reversal compensation MD32450
- Spindle compensation MD32700
- CEC compensation MD32710
- Temperature compensation MD32750
- Friction compensation MD32500.

Note: Backlash on reversal MD32450 must be adapted via an external device, such as circularity test or dial gauge.

The best contour results are only achieved when the circularity test returns the correct actual size and shape as well as a minimum p/p difference for all axis combinations (X-Y, X-Z, Y-Z).

Technically, the worst case scenario of circle radius and path velocity must result in a radial acceleration of which the machine is capable.

A distinction is made between the following scenarios:

Standard circularity tests of machine manufacturers normally use a radius of 100 mm or 150 mm for feedrates that are specified by the machine manufacturer. The machine manufacturer defines the specifications or an acceptable result.

High-speed machining operations generally place high demands on precision at high speed. Therefore, for high-speed milling machines, the test circle radii should be in the range from 10 to 25 mm and the feedrates in the range from 5 to 10 m/min. The results for high-speed milling machines are generally acceptable when the p/p difference ≤ 0.010 mm and the actual size of the circle is the same as the programmed radius.

Paremeterizing the circularity test
Open the circularity test user interface:





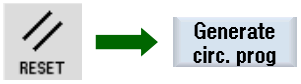
The parameters can now be set:

Select the axes that are to be traversed in the circularity test and the respective measuring system at **"Measurement"**.

The required circle settings can be made at **"Parameter"**.

The display options must be selected in the **"Representation"** category. They can be adapted later in the chart.

Generating the circle program
An NC program can be generated from the circularity test.
After setting the parameters:



The generated program is then displayed and must be confirmed:

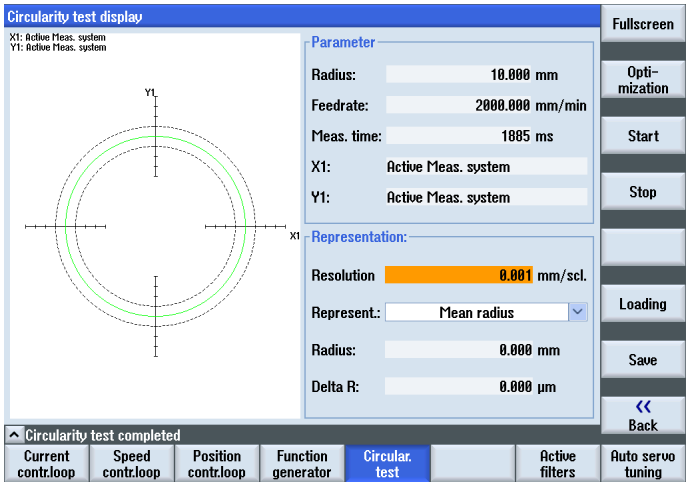


Important: Make sure that the axes can be traversed according to the program and that the paths are free.

Starting the circularity test
The graphic window can now be opened:



Notes:



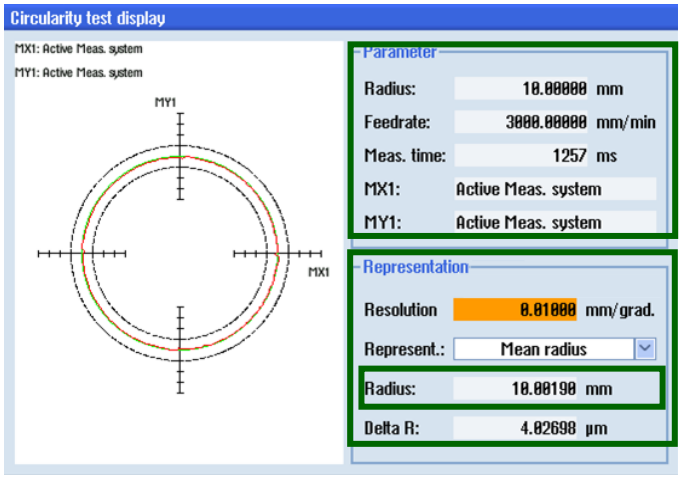
The NC program can now be executed. Make sure that the control is in AUTO mode:



The circularty test can then be started.



After the machine axes have described a circle, the result is displayed:



After tuning with AST, the circle shown here has a mean radius that it **0.0019 mm** too large.

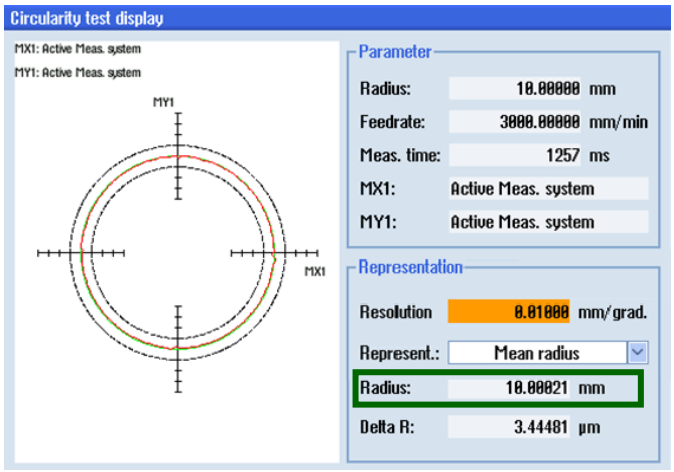
Generally, the actual radius with tuned feedforward control is mostly too large. This can be corrected through an additional time constant (**MD32910 DYN_ MATCH_TIME**). This may have to be activated (MD32900).

The machine data shown below is from a real machine that has been used for the following circularity test examples.

Date	Description	X	Y
32200	POSCTRL_GAIN	8.50	8.50
P1460	SPEEDCTRL_GAIN1	3.01	3.89
P1462	SPEEDCTRL_INTEGRATOR_TIME_1	6.18	6.18
P1433	SPEEDCTRL_REF_MODEL_FREQ	106.3	106.3
32610	VELO_FFW_WEIGHT	1.0	1.0
32620	FFW_MODE	3 or 4	3 or 4
32810	EQUIV_SPEEDCTRL_TIME	0.0022	0.0022
32900	**DYN_MATCH_ENABLE**	**1**	**1**
32902	**DYN_MATCH_MODE**	**1**	**1**
32910	**DYN_MATCH_TIME**	**0.0009**	**0.0009**

The already existing value has been adapted by AST.
To correct the circle radii, the time constant **DYN_MATCH_TIME** is increased in <u>both axes</u> by 0.0002 s to 0.0011 s, and activated by **cf** (whereby the channel must be in **Reset**).

This results in the following figure:



The circle is now only **0.00021 mm** too large.

<u>Note</u>:
It is recommended to make the adjustment only in special cases via the AX_JERK_TIME (MD 32410) and instead, as described, enter the correction value in DYN_MATCH_TIME.

Notes:

Different time constants
If the time constants of two axes are different, this effects the following error and therefore the circularity. The circle becomes and ellipse or a figure eight. The following:

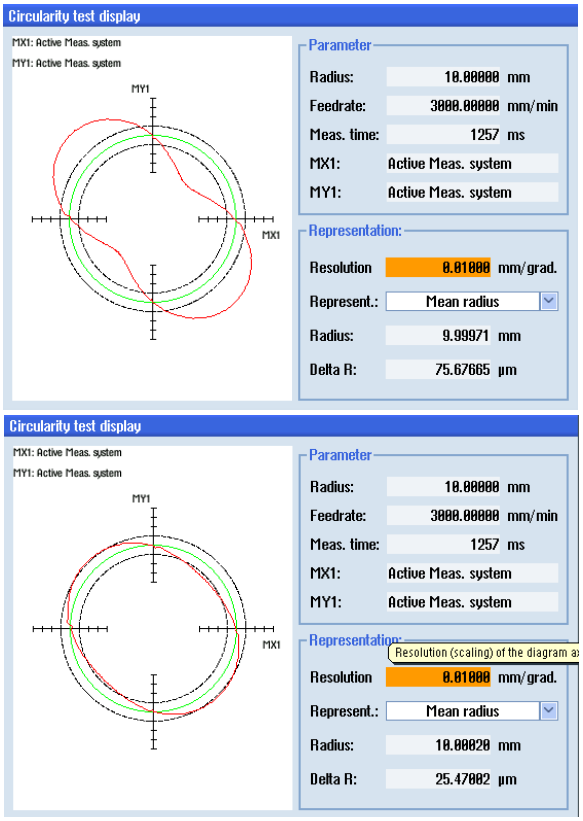
- AX_JERK_TIME (MD32410),
- DYN_MATCH_TIME (MD32910) and
- DESVAL_DELAY_TIME (MD32985)

mainly effect the differing circularity.

As already explained in the previous section, the adaptation should normally be made with the DYN_MATCH_TIME.

The time constant in one or both axes can now be adapted. In this way, it is possible to reduce the difference in the shape.

Example:
In the following example, the X1 axis is too "slow" and cannot follow the movement of the Y axis. The time constant of the X axis must therefore be reduced or that of the Y axis increased. An improved circularity is shown after the adaptation.



Notes:

Additional spindle tuning

Compare the preset MD with the MD shown below as well as all the relevant parameters for spindle/axis interpolation. The example applies to a 8000 rpm spindle motor without mechanical gear stages.

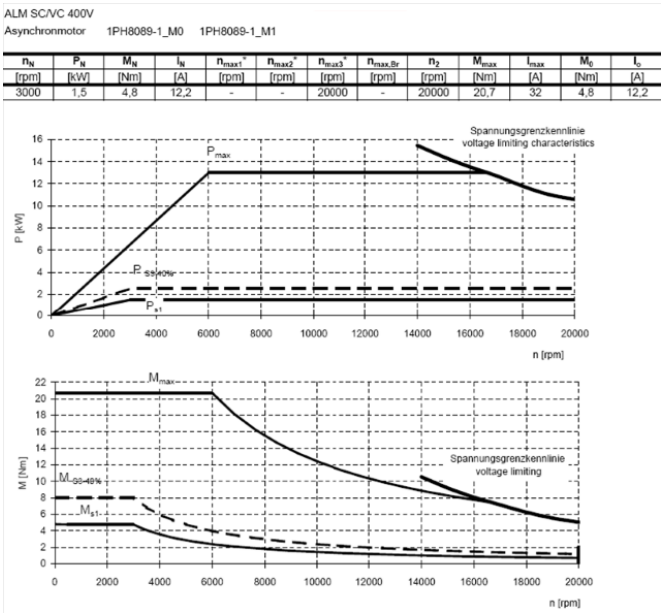
The following parameters can be preset with the values proposed in the table. The values vary depending on the application.

Machine Data	Machine Data Name	Value	Explanation
MD 30300	IS_ROT_AX	1	Rotary axis/spindle
MD 30310	ROT_IS_MODULO	1	Modulo conversion for rotary axis/spindle
MD 30320	DISPLAY_IS_MODULO	1	Modulo 360 degree display for rotary axis/spindle
MD 32000	MAX_AX_VELO[0]	8000	Maximum axis velocity
MD 32010	JOG_VELO_RAPID	60	Traverse rate in rapid jog
MD 32020	JOG_VELO	10	Traverse rate in jog
MD 32040	JOG_REV_VELO_RAPID	60	Rotary traverse rate in rapid jog
MD 32050	JOG_REV_VELO	10	Rotary traverse rate in jog
MD 32200	POSCTRL_GAIN[0]	x	As tuned
MD 32200	POSCTRL_GAIN[1]	x	As tuned
MD 32620	FFW_MODE	3 or 4	3 = Velocity FFW, Torque FFW = 4
MD 32640	STIFFNESS_CONTROL_ENABLE	1	Dynamic Stiffness Control
MD 32800	EQUIV_CURRCTRL_TIME[0]	x	As tuned
MD 32800	EQUIV_CURRCTRL_TIME[1]	x	As tuned
MD 32810	EQUIV_SPEEDCTRL_TIME[0]	x	As tuned
MD 32810	EQUIV_SPEEDCTRL_TIME[1]	x	As tuned
MD 34000	REFP_CAM_IS_ACTIVE	0	If no BERO switch
MD 34020	REFP_VELO_SEARCH_CAM	30	Reference point approach velocity
MD 34040	REFP_VELO_SEARCH_MARKER[0]	30	Creep velocity
MD 34040	REFP_VELO_SEARCH_MARKER[1]	30	
MD 34060	REFP_MAX_MARKER_DIST[0]	370	Maximum distance to reference mark
MD 34060	REFP_MAX_MARKER_DIST[1]	370	
MD 33000	FIPO_TYPE	3	Type of interpolator
MD 35000	SPIND_ASSIGN_TO_MACHAX	1	Assignment of spindle to machine axis
MD 35100	SPIN_VELO_LIMIT	8000	Maximum spindle speed
MD 35110	GEAR_STEP_MAX_VELO[0]	8000	Maximum spindle speed for gear change
MD 35110	GEAR_STEP_MAX_VELO[1]	8000	
MD 35130	GEAR_STEP_MAX_VELO_LIMIT[0]	8000	Maximum spindle speed for gear stage
MD 35130	GEAR_STEP_MAX_VELO_LIMIT[1]	8000	
MD 35200	GEAR_STEP_SPEEDCTRL_ACCEL[0]	X	Acceleration in speed control mode; to be tuned
MD 35200	GEAR_STEP_SPEEDCTRL_ACCEL[1]	X	
MD 35210	GEAR_STEP_POSCTRL_ACCEL[0]	X	Acceleration in position control mode; to be tuned
MD 35210	GEAR_STEP_POSCTRL_ACCEL[1]	X	
MD 35500	SPIND_ON_SPEED_AT_IPO_START	2	Feedrate enable for spindle in the set range
MD 35550	DRILL_VELO_LIMIT[0]	4000	Maximum spindle speed for tapping
MD 35550	DRILL_VELO_LIMIT[1]	4000	
MD 36000	STOP_LIMIT_COARSE	0.4	In position coarse limit
MD 36010	STOP_LIMIT_FINE	0.1	In position fine limit
MD 36030	STANDSTILL_POS_TOL	5	Standstill tolerance
MD 36040	STANDSTILL_DELAY_TIME	1	Delay for standstill monitoring
MD 36050	CLAMP_POS_TOL	1	Clamping tolerance
MD 36060	STANDSTILL_VELO_TOL	2	Tolerance for “Axis/spindle” stopped
MD 36200	AX_VELO_LIMIT[0]	8800	Threshold for velocity monitoring
MD 36200	AX_VELO_LIMIT[1]	8800	
MD 36300	ENC_FREQ_LIMIT[0]	1000000	Encoder limit frequency
MD 36300	ENC_FREQ_LIMIT[1]	1000000	
MD 36400	CONTOUR_TOL	30	Tolerance band for contour monitoring
P 1433	SPEED_CONTROLLER_REFERENCE_MODEL_NATURAL_FREQUENCY	x	As tuned

Notes:

The speed controller acceleration should now be checked.
The acceleration should be determined first without the effect of the NC.
Set MD: **35200 [0] $MA_GEAR_STEP_SPEEDCTRL_ACCEL = 9999** (t=V/a t=9000/60s/9999 t=15 ms therefore no effect)
35200 [1] $MA_GEAR_STEP_SPEEDCTRL_ACCEL = 9999

Now consider the motor data sheet. For example, 1PH8089-1VM02-0MG1 induction motor:

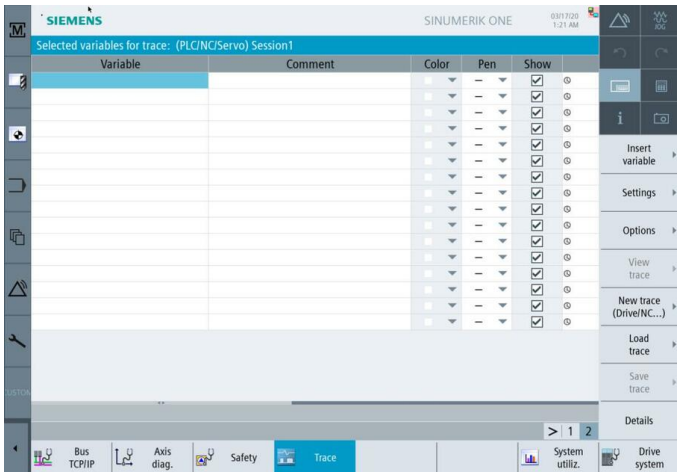


Acceleration/deceleration with the original motor data, corresponds to the S1 curve.

P640	Current limit (P305 rated motor current x 1.5)	17.7 A
P1520	Torque limit upper/motoring	4.8 Nm
P1521	Torque limit lower/regenerative	-4.8 Nm
P1530	Power limit motoring	1.5 KW
P1531	Power limit regenerating	-1.5 KW

Switch to the trace screen:





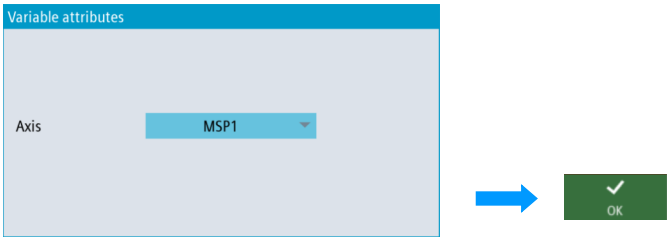
The required variables can be selected:

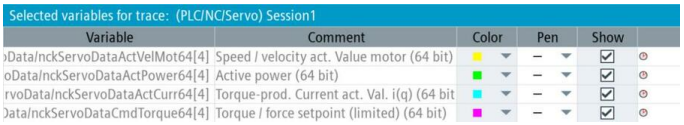


Required variables:

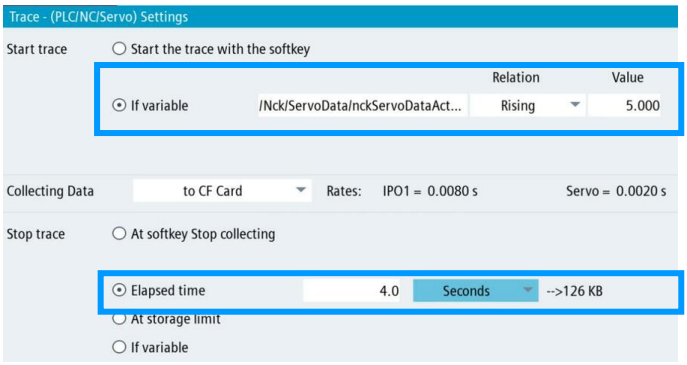
Speed/velocity actual value motor
Active power
Torque-producing current actual value i(q)
Torque/force setpoint (limited)

Make sure that the spindle has been selected when you add the variables:





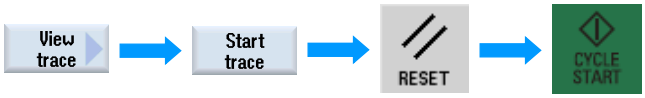
Notes:



Enter the following program in MDA:

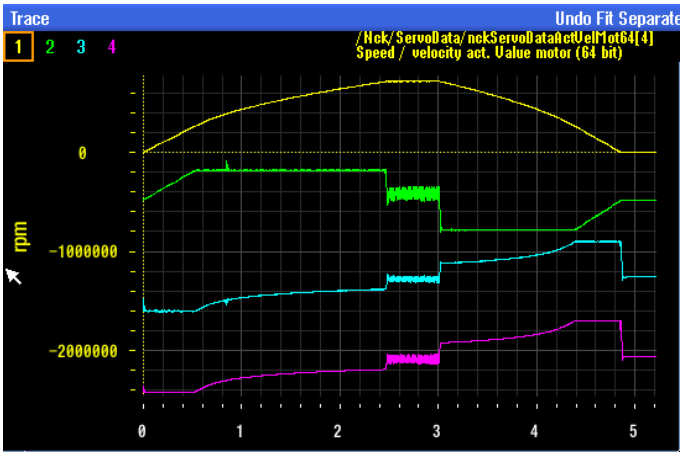
```
MDA
S8000 M03¶
G04 F5¶
M30¶
```

Initialize the trace and execute the program:



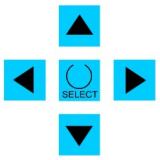
After completing the trace function, press the "Fit all" softkey:





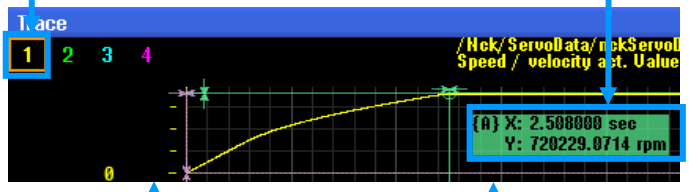
The acceleration time can be determined with the cursor:





Trace selection:
actual speed

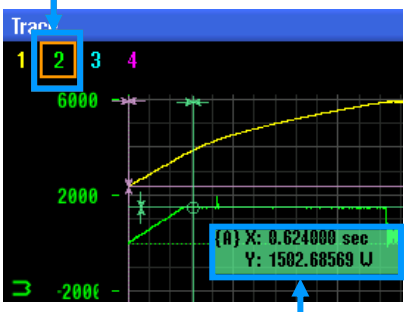
Time required for the acceler-
ation up to the
set speed = 2.5 seconds



Cursor positioned at the start of the
trace and at the point at which the set
speed is reached.

The power that is used/available for the acceleration of the motor can be seen on trace 2:

Trace selection:
actual power

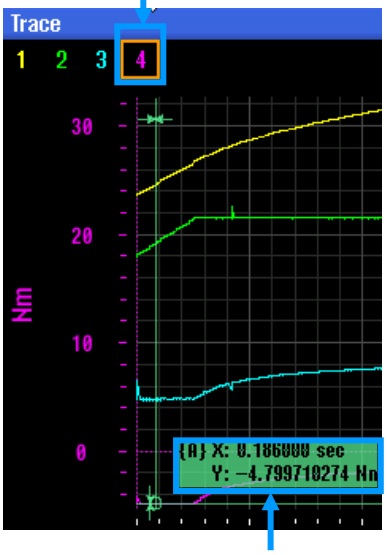


Maximum available power
= 1.5 kW = S1 curve.

Notes:

The torque setpoint can be seen on trace 4:

Trace selection:
torque setpoint



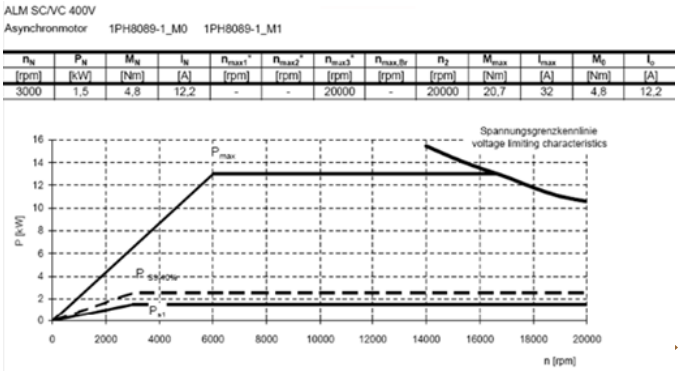
Torque setpoint
= 4.8 Nm = S1 curve.

Based on the motor data sheet, the acceleration/deceleration can be changed up to the maximum values. However, this depends on the respective application.

Enter suitable values for the application. In our example, a short acceleration/deceleration time was required.

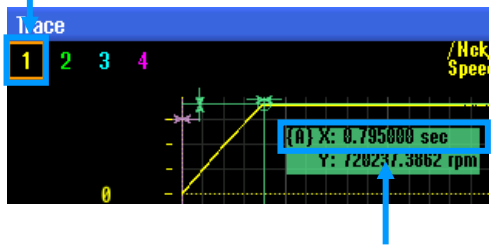
The maximum values from the motor data sheet are entered in the appropriate drive data:

P640	Current limit	32 A
P1520	Upper torque limit / motoring	20.7 Nm
P1521	Lower torque limit / regenerative	-20.7 Nm
P1530	Power limit motoring	13 KW
P1531	Power limit regenerating	-13 KW



Execute the spindle and trace function.
The time required for the acceleration up to the set speed can now be checked.

Trace selection:
actual speed



Acceleration = 0.795 seconds

Set the MD:
35200[0] & [1] $MA_GEAR_STEP_SPEEDCTRL_ACCEL, so that it corresponds to the requirements of the machine manufacturer for the spindle acceleration.

Spindle position controller

A test can now be performed to set the acceleration and jerk of the position control. This test is performed through tapping without a compensating chuck.

Check MD **35550[0] & [1] DRILL_VELO_LIMIT**; this is the maximum tapping velocity.

Check SD **55484 $SCS_DRILL_TAPPING_SET_MC[0] = 1**; it keeps the spindle in positioning mode during the tapping.

The following program, which TAPS 10 holes in the same position, can be used.

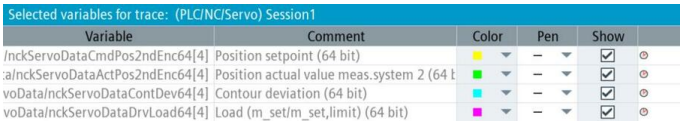
```
SUPA DO G0 G90 Z-200¶
FFWON¶
M19¶
SOFT¶
LAB:¶
G331 Z-214 S4000 K1¶
G332 Z-200 S4000 K1¶
REPEAT LAB P=9¶
M30¶
```

Notes:

The following variables can be selected in the trace window:
Position setpoint (64-bit)
Position actual value measuring system (64-bit)
Contour deviation (64-bit)
Load (m_set/m_set,limit) (64-bit)

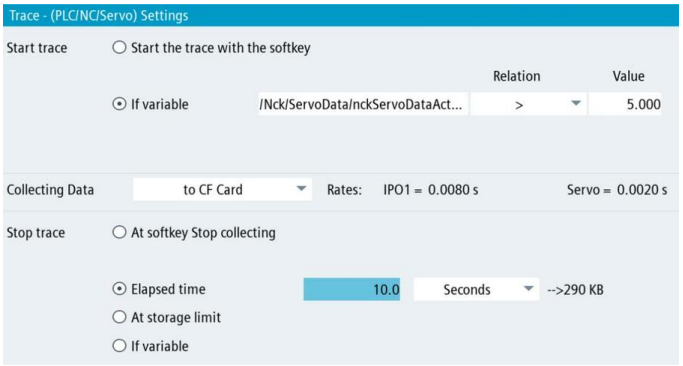
Note: Make sure that the correct signal is selected when monitoring measuring systems, e.g. measuring system 1 or 2.





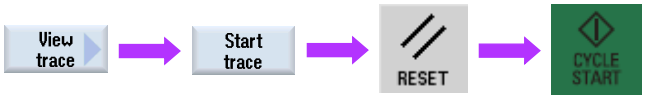
Select the trace trigger, etc. via the "Settings" softkey.
The example shows a trace that is triggered as soon as the actual position exceeds 5 degrees. The trace is recorded for 10 seconds.







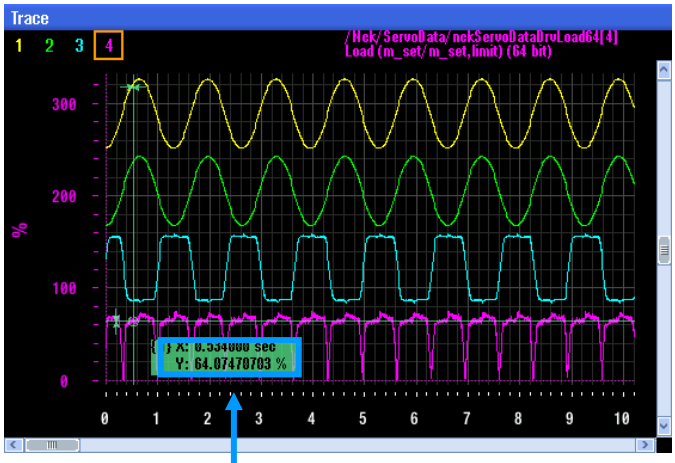
Start the trace and press "CYCLE START" on the machine control panel.



In the example below, the utilization reaches 64% and the position control acceleration can be increased.
However, during tapping the spindle can operate above the S1 curve so that you must ensure that the utilization does not reach 100% as otherwise "Contour monitoring" alarms are output.

Tune **35210 $MA_GEAR_STEP_POSCTRL_ACCEL** and **32431 $MA_MAX_AX_JERK** to obtain optimum results.

Also check the adaptable control range **p1464, p1465** of the spindle.

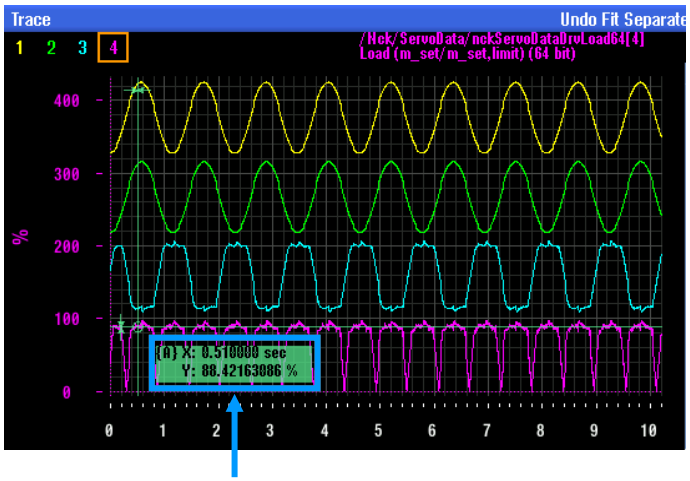


Utilzation/Load = 64%

Notes:

The following example shows the effect of an increase in the position control acceleration from 120 to 160; the utilization now increases from 64% to 88%.

If the application is a pure tapping machine, a test can be performed during which 100 holes are tapped and then the servo trace repeated to ensure that the utilization never reaches 100%.



Utilization = 88%

Spindle maximum speed limit settings by operating mode:

The chart below is a general guide for which axis machine data might be having a limiting effect on maximum spindle speed based on the spindle operating mode. Not all possible machine data and scenarios are covered.

Spindle operating mode	MD32000	MD35550	MD35300	MD35130	SD43220	MD35100 Alarm 22100	MD36200 Alarm 25030
Speed control mode: M3, M$, M5				X	X	X	X
Position control mode: M70 or SPCON	X					X	X
Position control mode: SPOS or M19			X		X	X	X
Position control mode: Rigid Tapping Cycle840, G331/G332		X			X	X	X
Speed control mode: Tapping with floating holder Cycle840				X	X	X	X
Speed control mode: Thread cutting (G33, G34, G35)				X	X	X	X
Position control mode: Synchronous spindle				X	X	X	X

Notes:

Advanced Surface/Top Surface

Expanded functionality of G-Code Group 59

A new command has been introduced in Technology G-Code Group (group 59) - DYNPREC

The dynamic response of the path axes are determined by the commands from G-Code Group 59. Therefore by switching the dynamic response settings, roughing a workpiece can be optimized in terms of time and smoothing a workpiece can be optimized in terms of the surface.

With the Technology G-Code Group a defined parameter state exists for each setting. Because of this no machine data is ever overwritten and therefore at any time the parameters are in a clearly defined state.

The technology G-code group 59 will be called via NC command or the CYCLE832

DYNNORM: normal dynamic response
DYNPOS: dynamic response positioning mode, tapping
DYNROUGH: dynamic response roughing
DYNSEMIFIN: dynamic response semi-finishing
DYNFINISH: dynamic response finishing
DYNPREC: dynamic response superfinishing / precision

The following machine data's are effective when the NC command is changed in G-code group 59 or in the cycle CYCLE832

Axial machine data Axis-specific dynamic response settings - Index[0-5]

MD32300 MAX_AX_ACCEL[<n>] Axis acceleration
MD32431 MAX_AX_JERK[<n>] Max. axial jerk for path motion
MD32432 PATH_TRANS_JERK_LIM[<n>] Max. axial jerk at the block transition in continuous-path mode
MD32310 MAX_ACCEL_OVL_FACTOR[<n>] Overload factor for axial velocity jumps
MD32433 SOFT_ACCEL_FACTOR[<n>] Scaling of acceleration limitation for SOFT
MD32434 G00_ACCEL_FACTOR[<n>] Scaling of the acceleration limitation with G00
MD32435 G00_JERK_FACTOR[<n>] Scaling of the jerk limitation with G00

Path-related dynamic response settings - Index[0-5]

MD20600 MAX_PATH_JERK[<n>] Path-related maximum jerk Channel-specific machine data
MD20602 CURV_EFFECT_ON_PATH_ACCEL[<n>] Influence of path curvature on the path acceleration
MD20603 CURV_EFFECT_ON_PATH_JERK[<n>] Influence of path curvature on the path jerk

Channel machine data – Channel related machine data for free form machining - Index[0-5]

MD 20443 $MC_LOOKAH_FFORM [<n>] Activate extended LookAhead
MD 20485 $MC_COMPRESS_SMOOTH_FACTOR[<n>] Smoothing by compressor
MD 20486 $MC_COMPRESS_SPLINE_DEGREE[<n>] Compressor spline degree
MD 20487 $MC_COMPRESS_SMOOTH_FACTOR_2[<n>] Smoothing by compressor for rotary axis
MD 20605 $MC_PREPDYN_SMOOTHING_FACTOR[<n>] Factor for curve smoothing
MD 20606 $MC_PREPDYN_SMOOTHING_ON[<n>] Activation of curve smoothing

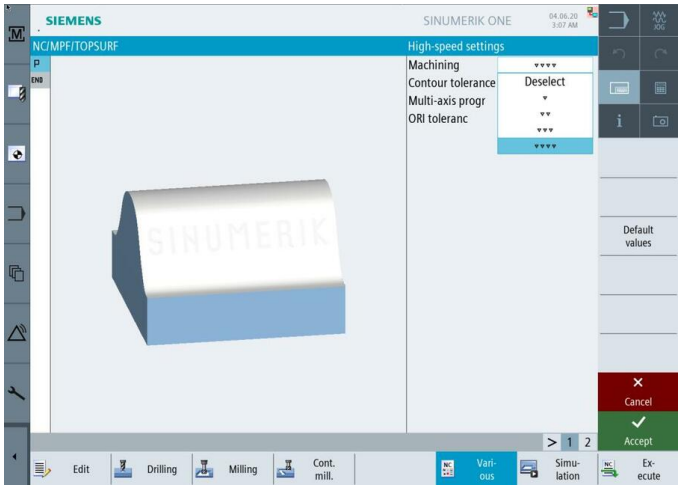
Range of values for index n: Index <n> Meaning

0 - Value for DYNNORM
1 - Value for DYNPOS
2 - Value for DYNROUGH
3 - Value for DYNSEMIFIN
4 - Value for DYNFINISH
5 - Value for DYNPREC

Note:
The recommended values for the machine data above can be checked with a checking program. See the later section in this module.

The machining strategies are shown in the **CYCLE832** below for:

DYNROUGH	▿
DYNSEMIFIN	▿▿
DYNFINISH	▿▿▿
DYNPREC	▿▿▿▿



Notes:

The **CYCLE832** supports the tooling of free-form surfaces in 3- or 5-axis programs. A good compromise can be found between machining time and surface quality. The softer the machine runs, the better the surface gets – i.e. a low jerk value has to be chosen (e.g. between 20-50 m/s^3). However, the machining time takes longer in this case and so in order to save time, you can set the jerk and acceleration higher for roughing by selecting DYNROUGH. However, a relevant value must be found for index 2.

As a rule, it will be determined experimentally which jerk during roughing is not too hard despite the time saved and when the surface and accuracy of the workpiece are nevertheless acceptable during roughing.

CYCLE832 can select four different types of machining: roughing, semi-finishing, finishing and superfinishing. For each of these modes there are, **in addition to the normal axis data (DYNNORM)** additional parameter settings. These additional parameters for jerk and acceleration are in **MD32431** and **MD32300**.

The dynamic G-code DYNPOS is used mainly for positioning purposes, such as tapping.

Machining type	dynamic G-Code (group 59)	field index in axis-MD
Standard (normal machining)	DYNNORM	0
Positioning	DYNPOS	1
Roughing	DYNROUGH	2
Semi-finishing	DYNSEMIFIN	3
Finishing	DYNFINISH	4
Superfinishing/precision	DYNPREC	5

A low value of 1 to 2 m/s² is recommended for the acceleration during finishing/superfinishing in order to avoid many acceleration and braking processes during milling. The acceleration during roughing can be set to the maximum value already determined by calculation.

The previously determined jerk value can be set for finishing. For roughing operations, it is recommended to put in a value that is four times as high as the one for finishing.

Additionally the **jerk of the path** (MD20600[0…5]) should be set to 10000 because the jerk values are set for every individual axis.

In order to be able to determine the appropriate values all optimization steps mentioned before should be performed first of all. To get approximate start values for an optimization, the set-up rules are summarized in the table below:

	Roughing[2]	Semi-finishing [3]	Finishing [4] [5]
MD32300 (acceleration)	a_{max}		≤ 2
MD32431 (jerk)	finishing x 4	finishing x 2	maximum jerk determined in jerk optimization

Now the settings can be refined. It is advisable to create an NC program for a component with which these settings can be checked. The optimization is complete when the component properties (especially the surface quality) turn out as desired. If the surface does not meet the specifications, the maximum jerk and acceleration of the respective dynamic G group must be reduced further.

Notes:

Advanced/Top Surface MD Checking program

Overview

Check programs are now available for checking the Advanced/Top Surface machine data settings.
It is essential that the programs match the respective software version installed on the machine. The result is written to a MDC_RESULT.MPF file in the MPF.DIR folder.

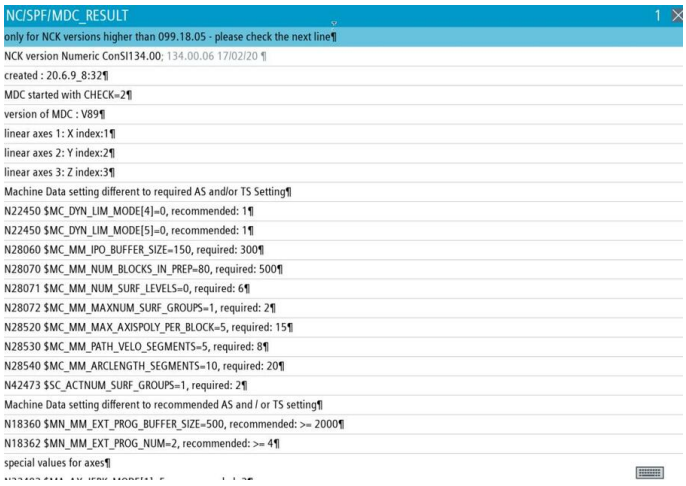
Mode of operation

Primarily this subprogram shall help to set the machine data for AdvancedSurface and TopSurface as recommended.
The subprogram checks the machine data and writes an information about machine data with values differnt to the recommended values.
Additionally the subprogram checks if the relevant G-Codes relevant for Advanced Surface have the required or recommended value.
Usually these values shall be set by CYCLE832.SPF and CUST_832.SPF. To check these values the subprogram has to be executed after CYCLE832.
The subprogram creates the file MDC_RESULT.MPF in the same directory.
Please check N11420 $MN_PROTOCOL_LEN is >= 10 (kB).
Please define the type of check: 0: no check, 1: only for AdvancedSurface, 2: for AdvancedSurface and TopSurface in line N101
And please define the type of machinetool - necessary for machinetools with only three axis for interpolation in line N102
TRUE: machinetool has only three axes, FALSE: machinetool has more than 3 axes
At the end of the subprogram a message is displayed about the state of AdvancedSurface or TopSurface and the active tolerance value.
To determine the rotary axes the transformation has to be written to MD24000 and following MDs, even if the transformation information is used from $NT_...

Test programs for Advanced/Top Surface

https://support.industry.siemens.com/cs/ww/en/view/109738423

An example of MDC_RESULT.MPF/SPF with changes required from Advanced Surface to Top Surface



Example of axial-related dynamic response settings

MD32300 $MA_MAX_AX_ACCEL[0] = 4
MD32300 $MA_MAX_AX_ACCEL[1] = 4
MD32300 $MA_MAX_AX_ACCEL[2] = 4
MD32300 $MA_MAX_AX_ACCEL[3] = 2
MD32300 $MA_MAX_AX_ACCEL[4] = 1
MD32300 $MA_MAX_AX_ACCEL[5] = 1
MD32420 $MA_JOG_AND_POS_JERK_ENABLE = 1
MD32430 $MA_JOG_AND_POS_MAX_JERK = 50
MD32431 $MA_MAX_AX_JERK[0] =100
MD32431 $MA_MAX_AX_JERK[1] = 50
MD32431 $MA_MAX_AX_JERK[2] = 50
MD32431 $MA_MAX_AX_JERK[3] = 20
MD32431 $MA_MAX_AX_JERK[4] = 20
MD32431 $MA_MAX_AX_JERK[5] = 10
MD32432 $MA_PATH_TRANS_JERK_LIM[0] =100
MD32432 $MA_PATH_TRANS_JERK_LIM[1] = 50
MD32432 $MA_PATH_TRANS_JERK_LIM[2] = 50
MD32432 $MA_PATH_TRANS_JERK_LIM[3] = 20
MD32432 $MA_PATH_TRANS_JERK_LIM[4] = 20
MD32432 $MA_PATH_TRANS_JERK_LIM[5] = 10
MD32310 $MA_MAX_ACCEL_OVL_FACTOR[0-5] = 1.2
MD32433 $MA_SOFT_ACCEL_FACTOR[0-5] = 1
MD32434 $MA_G00_ACCEL_FACTOR = 1
MD32435 $MA_G00_JERK_FACTOR = 1

Example of path/channel-related dynamic response settings

MD20600 $MC_MAX_PATH_JERK[0-5] = 10000
MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL[0] = 0
MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL[1] = 0
MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL[2] = 0.65
MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL[3] = 0.6
MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL[4] = 0.5
MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL[5] = 0.4
MD20603 $MC_CURV_EFFECT_ON_PATH_JERK[0-5] = 0
MD20443 $MC_LOOKAH_FFORM[0-1] = 0
MD20443 $MC_LOOKAH_FFORM[2-5] = 1
MD20485 $MC_COMPRESS_SMOOTH_FACTOR[0-1] = 0
MD20485 $MC_COMPRESS_SMOOTH_FACTOR[2-5] = 00001
MD20486 $MC_COMPRESS_SPLINE_DEGREE[0-1] = 3
MD20486 $MC_COMPRESS_SPLINE_DEGREE[2-5] = 5
MD20487 $MC_COMPRESS_SMOOTH_FACTOR_2[0-1] = 0
MD20487 $MC_COMPRESS_SMOOTH_FACTOR_2[2-5] = 0.5
MD20605 $MC_PREPDYN_SMOOTHING_FACTOR[0-5] = 1
MD20606 $MC_PREPDYN_SMOOTHING_ON[0-1] = 0
MD20606 $MC_PREPDYN_SMOOTHING_ON[2-5] = 1

Notes:

AST Applications

Servo motor with gearbox / ball screw:
Selecting the tuning objective:
For axes on which very different loads are expected, "robust" or "moderate" is recommended as tuning objective.

One encoder:	The limitation of the position control gain is active. The max. Kv can be set and is specified, for example, through the selection of the tuning objective. Because of the measurement on the motor encoder, a higher controller gain would probably be possible.
Two encoders:	The limitation of the position control gain is active. The max. Kv can be set and is specified, for example, through the selection of the tuning objective. The mechanical frequency response is also measured. If low-frequency resonances are measured here, this can lead to a further reduction of the position control gain so that the position on the load side does not overshoot.

If the maximum possible position control gain is to be calculated, the "max. Kv" limit must be increased to a higher value (e.g. 20 m/mm/min).

Indirectly driven turning spindle:
The "Robust responsiveness" tuning objective is recommended.
If a higher dynamic response is required in the speed controller, select "Moderate responsiveness" increase the minimum integral time to at least 20 ms in the speed controller strategy. **If different gear stages are used, they must be selected before each tuning and then tuned with AST.**

NOTE: Different tuning strategies can now be selected for spindles during tuning

Milling spindle:
The robust or moderate tuning objective is recommended.
The tuning refers to the establishment of a stable speed controller and a stable position controller. The feedforward control is also set.
If different gear stages are used, they must be selected before each tuning and then tuned with AST.

The ramp-up times are not tuned.

If third-party motors are used, they must be set up with the aid of the respective data sheets or SINAMICS MotID.
The current controller adaptation, etc. is not determined through AST.

Torque motor, linear motor and directly driven turning spindle:
Selecting the tuning objective:
"Moderate" is recommended for most applications.
"Quick" is recommended for high demands on accuracy and mold making.
The "Programmable AST for retune" option is recommended for very different clamping situations and load moments of inertia.
The axis can be retuned automatically (after a significant change in the load moment of inertia).
This must be supported by the machine manufacturer.

Because of the unknown mounting situation and greatly varying loads, it may be necessary to manually influence the measuring parameters for the speed-controlled system.
You can do this after a measurement, or— if the required measuring parameters are already known — selected at "Strategy" before starting the tuning.

NOTE: Different tuning strategies can now be selected for spindles during tuning

Notes:

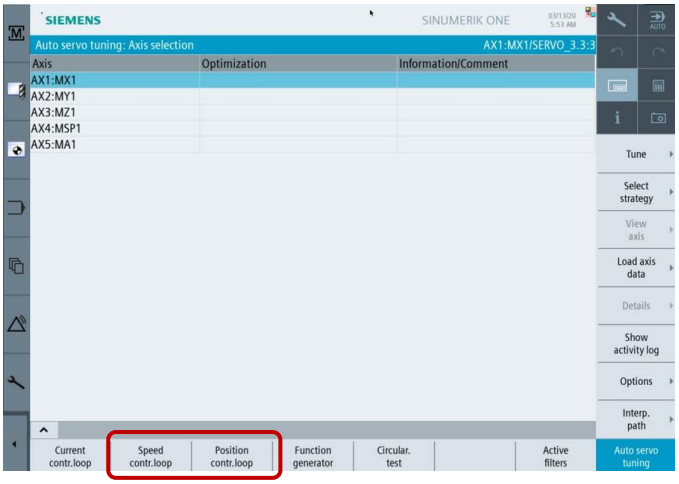
Gantry:
The gantry grouping must be set up and synchronized in the SINUMERIK. AST recognizes which axes belong to the gantry grouping and tunes all the associated axes automatically. Whereby, the gantry grouping is traversed with all axes and each axis measured individually. The tuning results are checked for dynamic equivalence. All gantry axes are then set with the same speed controller and position controller dynamic response. The current setpoint filters can be different (sutiable for the respective axis).

Selecting the tuning objective:
"Moderate" is recommended for most applications.
"Quick" is recommended for high demands on accuracy and mold making.
The "AST for automatic retuning" option is recommended for very different clamping situations and load moments of inertia. The axis can be retuned automatically (after a significant change in the load moment of inertia). This must be supported by the machine manufacturer.

Gantry with torque or linear motors:
Because of the unknown mounting situation and greatly varying loads, it may be necessary to manually influence the measuring parameters for the speed-controlled system.

Requirement: In order to be able to ensure that all gantry axes have the same dynamic response, the axes must have the same structure (same motor types, same encoders, same gear ratio).

Master-slave:
Must be tuned manually
If a clamping arrangement is required (torque difference), the clamping torque must first be selected and the torque compensatory controller set.
The speed-controlled system, the speed controller reference frequency response and the position controller reference frequency response can then be measured manually.
During this process, the master-slave grouping is always moved uniformly with all motors.



Notes:

A209: END

Notes:

A211

SINUMERIK ONE

Adaptive Friction Compensation

Module Description:

The "Adaptive Friction Compensation" option of the SINUMERIK ONE is described in this module.

This option enables acceleration-dependent friction effects at the reversal points of an axis to be partly or totally eliminated.

The friction compensation is part of the Setup user interface.

Module Description:

In order to be able to use the friction compensation with its new functionality, it is important to understand the technical background and to know the individual input steps. This is explained in the module.

Content:

Technical background

Requirements

Implementation in the control

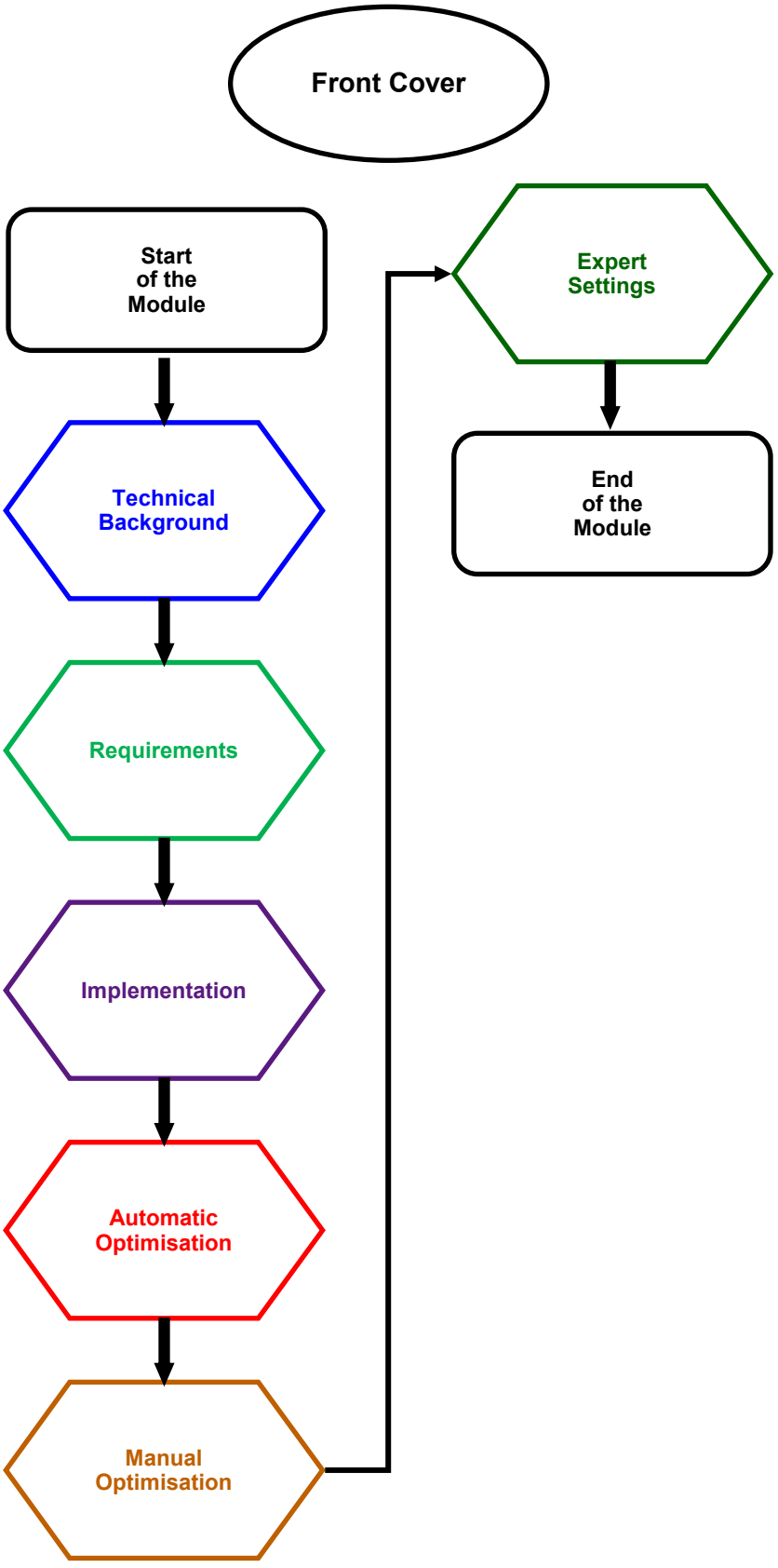
Automatic optimization

Manual optimization

Expert settings

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

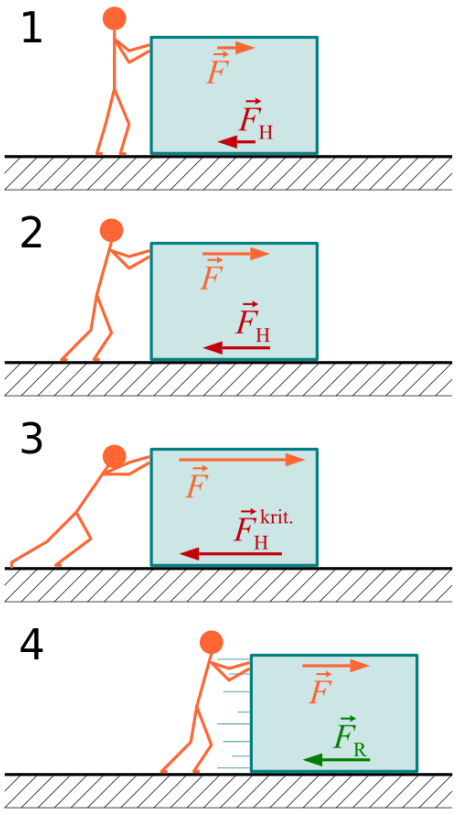

Front Cover
Start of the Module
Technical Background
Requirements
Implementation
Automatic Optimisation
Manual Optimisation
Expert Settings
End of the Module

Technical Background

When starting an axis from standstill, stiction forces occur (F_H) that are opposite to the desired axis movement and keep the axis stationary (see figures 1-3 on the right). Only when a critical friction force is overcome is the body released from standstill and starts to move (figure 4). The stiction forces are caused mainly by the guide mechanism and bearings of the axis. The effect can often be reduced through sufficient lubrication. However, if suitable workpiece quality cannot be achieved at the reversal points, the Adaptive Friction Compensation can be used.

A value can be added to the feedrate to compensate for the friction forces. Since the friction forces depend on the acceleration, the SINUMERIK control can specify this addition, the velocity injection pulse, for up to nine different acceleration values as well as the active time and the decay time. The values are interpolated linearly between these so-called interpolation points.

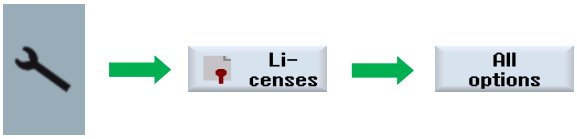
Extensive mechatronic know-how is not required to program the interpolation points. The user is assisted by an easy-to-use user interface.
Of course, it is also possible to make expert settings directly in the machine data.



"Stiction" from Debenben - Own publiction. licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Stiction.svg#mediaviewer/File:Stiction.svg

Requirements

The license **6FC5800-0BS06-0YB0** must be purchased in order to be able to use the Adaptive Friction Compensation function. The associated **Friction compensation with adaptive characteristics** option must be activated in the License menu:



Before the function can be used, some **machine data**, which is to be influenced by the Adaptive Friction Compensation **must be set in all axes**:

- **$MA_FRICT_COMP_MODE** 32490[0] **= 3**
- **$MA_FRICT_COMP_ENABLE** 32500 **= 1**

The calculation accuracy

- **MD10200 $MA_INT_INCR_PER_MM** and
- **MD10210 $MA_INT_INCR_PER_DEG**

must also be set to at least **10000** for an automatic optimization of the friction compensation. This step can be skipped if optimization is only to be performed manually.

The changes to the MD must be activated via a **po**.

The required softkey does not appear until the license has been activated and the machine data has been set for at least one axis.



Notes:

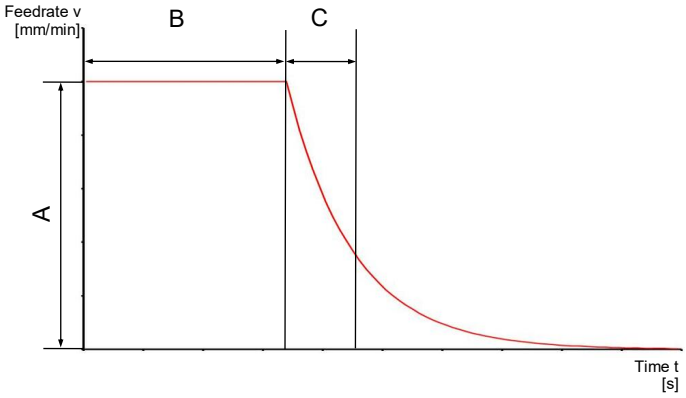
Implementation in the control

The characteristic (see figure on the right) of the acceleration values is stored in the control as a data list. The values are in the axis machine data of **MD32490 — MD32588**.

However, apart from the activation, a direct change of the machine data is **not required** for the setup via the setup wizard.

The values acceleration, amplitude (of the velocity injection pulse), active time and decay time are relevant for the use of the compensation.

The **amplitude (A)** describes the height of the velocity injection pulse at the reversal point.
The **active time (B)** is the time which the amplitude is added to the feedrate.
The amplitude is reduced during the **decay time (C)** in the form of a decaying exponential function.



The feedrate values for the interpolation points can be preset individually. The control calculates the accelerations from feedrate and travelled circle radius with $\mathbf{a=v^2/r}$ and saves the respective values as interpolation points in the machine data.

The required values for amplitude, active time and decay time between the specified interpolation points are then interpolated linearly during operation.

The diagram above shows an example of a characteristic curve for the active time, decay time and amplitude.

Notes:

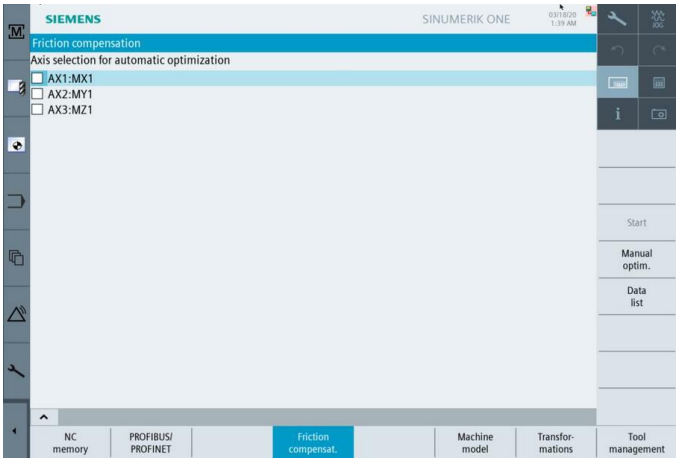
Automatic Optimisation

The automatic optimization of the Adaptive Friction Compensation produces good results even without special knowledge or much experience. The manual mode should therefore only be used in special cases.

Open the user interface of the Adaptive Friction Compensation:



All axes for which the friction compensation is available are shown on the start screen.
Already optimized axes are identified by the keyword "Optimized" and the time stamp of the optimization. These fields are empty before the optimization.



Preparation

To optimize an axis, it is important to specify **possible feedrates** for a circular path and a **target tolerance band**. This information is necessary because a machine cannot arbitrarily be "well" optimized and a compromise has to be found between speed and accuracy.

The values can now be entered in the control. To do this, open the data list. Alternatively, you can accept the preset values.



Friction compensation		CH1:CHAN1
55820[0]	$SCS_FRICT_OPT_RADIUS	10 mm
55822[0]	$SCS_FRICT_OPT_FEED	5500 mm/min
55822[1]	$SCS_FRICT_OPT_FEED	200 mm/min
55822[2]	$SCS_FRICT_OPT_FEED	400 mm/min
55822[3]	$SCS_FRICT_OPT_FEED	800 mm/min
55822[4]	$SCS_FRICT_OPT_FEED	1200 mm/min
55822[5]	$SCS_FRICT_OPT_FEED	3000 mm/min
55822[6]	$SCS_FRICT_OPT_FEED	2000 mm/min
55822[7]	$SCS_FRICT_OPT_FEED	4000 mm/min
55822[8]	$SCS_FRICT_OPT_FEED	5000 mm/min
55821[0]	$SCS_FRICT_OPT_RADIUS_ROT	90°
55823[0]	$SCS_FRICT_OPT_FEED_ROT	90 rpm

The following parameters can be set:

Radius: *$SCS_FRICT_OPT_RADIUS*

Feed: *$SCS_FRICT_OPT_FEED[i]*
- i = 1 … 9 (interpolation points)
- Not all possible interpolation points have to be assigned feedrate values; however, an entry of 0 cannot be followed by entries not equal to 0

The values of the rotary axes are in the respective variable with the extension *_ROT*.

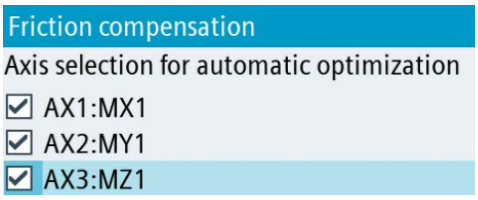
Complete the entries and back.



Note: During the optimization, the feedrates are processed in a sequence determined according to an algorithm. This may not correspond to the sequence of the values in the data list.

Axis selection

The axes on which the Adaptive Friction Compensation is to be used can now be selected. It is recommended that you optimize all of these axes. Each axis is traversed and optimized separately.
Note: The channel must be in the AUTO mode and all selected axes must be enabled.



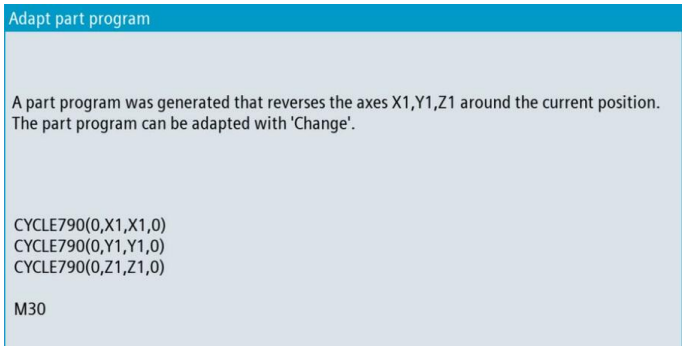
After the required axes have been selected, press "Start".



Notes:

Adapting the program

The following window which contains the NC program appears:

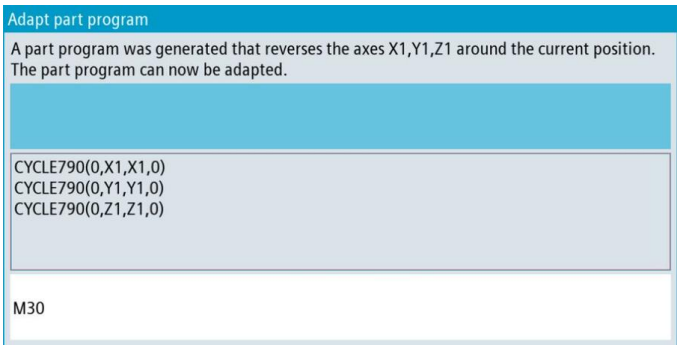


CYCLE790 ensures that the axes traverse circularly.

Changing the program

It may be necessary to insert changes in the program, e.g. cancellation of a clamping for rotary axes.

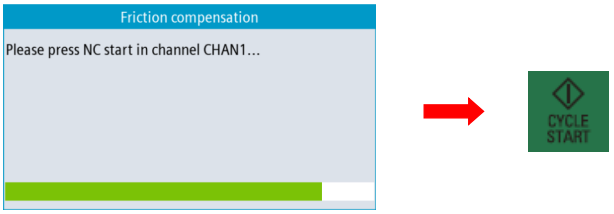




Note: To insert a line break, press [Shift]+[Enter]. [Enter] alone switches to the next input field.



Starting the optimization

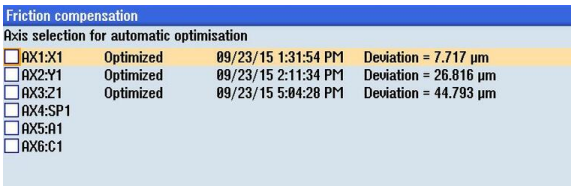


The measuring cycle now starts. The measurement progress is displayed next to the axis being optimized.

The current results and the determined parameters can be displayed. To do this, click in the graphic (see the figure on the right).

The cycle automatically runs through all selected axes.

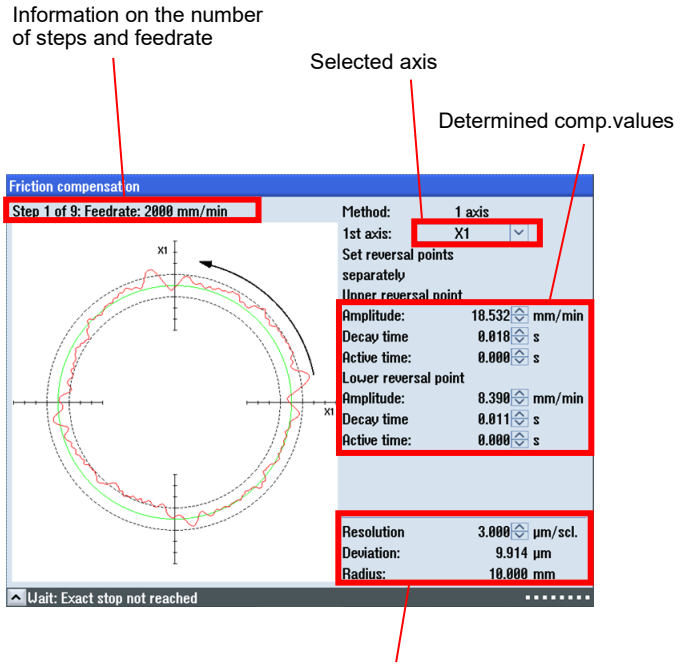
After completion of the optimization, the result is displayed next to the respective axis:



If the optimization does not provide any satisfactory results, a manual optimization may be necessary subsequently.

The automatically determined compensation values are then used as the starting point.

The interpolation points can also be changed subsequently and the axis optimized again automatically. The algorithm takes the active compensation values into account, which reduces the optimization duration.

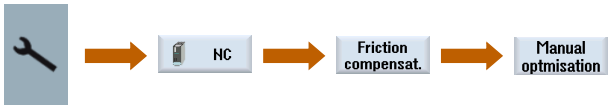


Notes:

Manual Optimisation

The manual mode is available for special applications or if the results of the automatic optimization are not adequate.
The optimization is performed on the basis of the currently active values (i.e. the optimization results if an automatic or a manual optimization has been performed previously).

Open the user interface of the manual optimization:





- Use Axis + / Axis - to select the axis to be adapted

Method
Note: Only the axis that has been selected with the above buttons is adapted.
The measured values of the selected axis are displayed on the vertical axis in the diagram.

1 axis:
- Selected machine axis is traversed via a sine polynomial
- Spindles, rotary axes or axes in general can also be adapted in this way without coupling
- Diagram horizontal axis: Measured values of the machine axis rotated through 90°

2 axes:
- Traversing of two geometry axes in a circle
- This function is only relevant for special cases and is not considered further here

Extended settings
The parameters to be set are identical to those of the automatic optimization and are described in the relevant section. Only the direction of rotation is also required (*$SCS_FRICT_OPT_DIR_MINUS). However, this is only active for two-axis traversing.*

Strategy
Adaptation of the parameters until the measurements are within the required tolerance band (**trial-and-error** method).

Before the adaptation: Determine the **target tolerances** and specify the **possible feedrates** for a circular path.

There is a plausibility check before the start of the measurement. The following parameters are checked:
- Observance of the maximum permissible feedrates for the machine
- No 0 for the feedrates of the interpolation points (unless it is the last specified value)

Recommendation for the setting of the feedrates:
1. Average feedrate
2. High feedrate
3. Low feedrate

Now measure and adapt these three interpolation points.

If required, interpolation points can be inserted one after the other and the compensation parameters set if the measured values are not in an acceptable tolerance band. If the result is satisfactory from the start, the relevant intermediate point can be removed again.

Starting the measurement and compensation values
The measurement is started in the same way as in automatic mode:

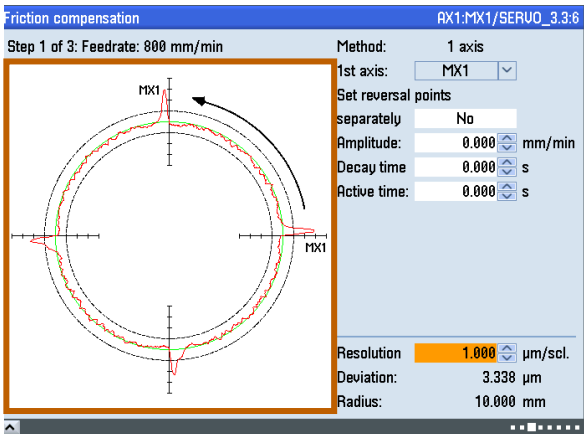
- Operating mode = AUTO and enables active
- "Start" softkey
- If required, adaptation of the program
- Confirmation of the program
- NC start

The NC now performs the circle measuring program cyclically and displays the result in the circle diagram. Changes can now be made to the compensation values. The current measurement is terminated, the new parameters activated and the measurement started again automatically.

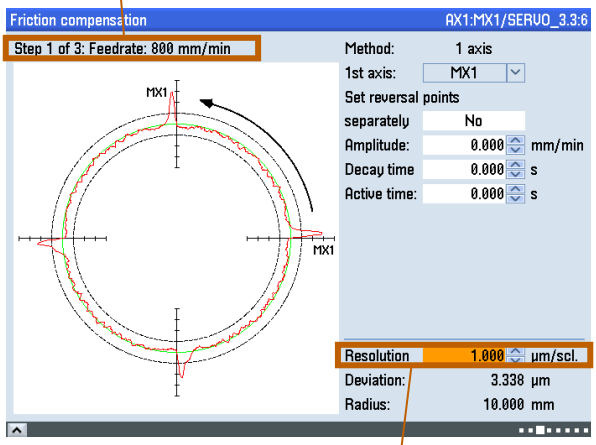
As an example, the X axis of a real machine is measured.

The circularity diagram shows the following result:

Notes:

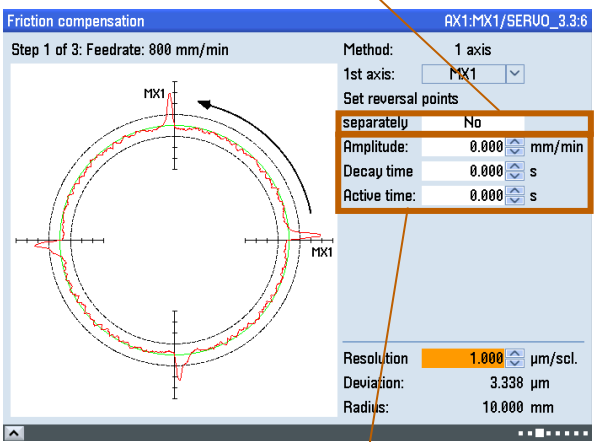


Information on the number of steps and feedrate



Resolution setting. If no value is entered, this parameter is adapted after every measurement. If a value is entered manually, this value is retained in all the subsequent steps.

Separate parameters for the upper and lower reversal points. Particularly with vertical axes, the force of gravity generates an acceleration in the opposite direction to the motion at the lower reversal point. Different values must then be set for each reversal point.

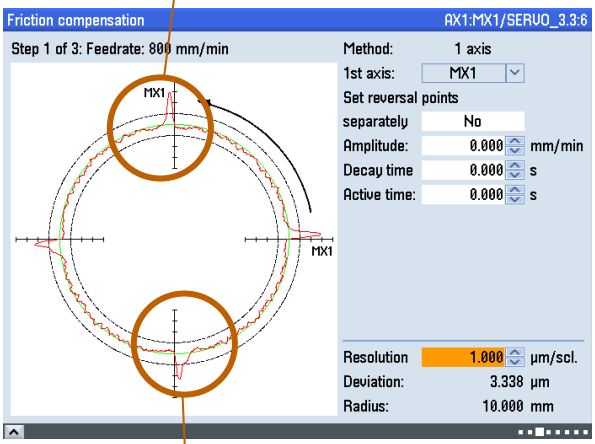


Setting of amplitude, decay time and active time

Performing a measurement with one axis

- One axis (X1)
- Resolution set to 1 µm/scl

Circularity outside the required tolerance (1 µm)

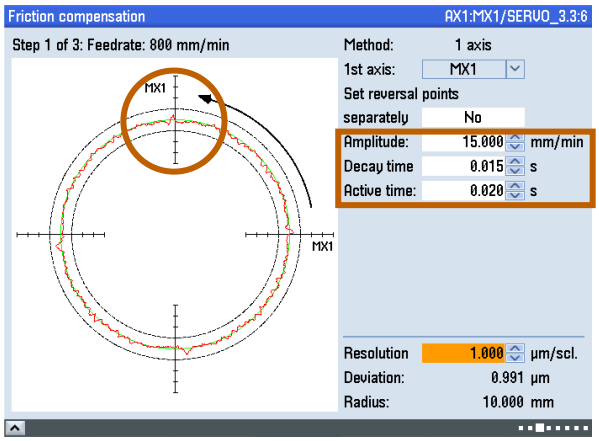


Equidirectional deviation
→ Reversal points do not have to be adapted separately

Notes:

The deviation can already be reduced by adapting the amplitude and the decay time.

The measured values are now within the tolerance band and the measurement can be continued with the next step.





Repeat this procedure for all subsequent interpolation points.

Note: The arrow and page up/down keys can be used to make fine settings.

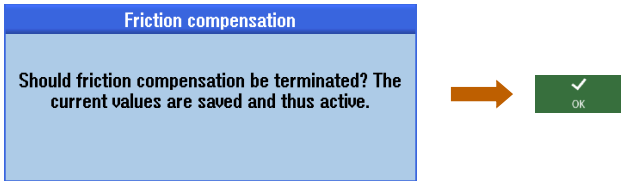
Select the text field:



	▲	▼	PAGE UP	PAGE DOWN
Amplitude	+0.1	-0.1	+1.0	-1.0
Times	+0.001	-0.001	+0.01	-0.01

Accepting changes
After adapting the last step:





Important note
Determined and edited parameters are **only kept temporarily** during the measurement and not saved and activated in the machine data until **after** the measurement is completed.

This is performed through confirmation **after the last step**.

Parameters have only been completely saved when the Start button is grayed-out.

If there is a power failure during this stage, all determined values are lost. It is therefore recommended that only a few steps are adapted and then the changes saved before continuing with the next step.

Expert settings

The control has further parameters for special application:

- Delay time before addition of the velocity injection pulse
- Additional trigger delay
- Smoother rise of the curve
- Torque injection pulse (+ delay time and smoother rise of the curve)
- Effect of the compensation on the actual value and not on the setpoint (32490[0] = 4)

The setting of these values requires greater knowledge of the mechanical system and the control, and should only be performed by experts. These parameters are also not supported by the Settings user interface.

A211: END

Notes:

213

SINUMERIK

Electronic Logbook

Module Description:

The electronic logbook function of the SINUMERIK controller is used to enter data such as manufacturer (OEM), dealer and user information, 1st and 2nd commissioning dates as well as records of service visits.
This data is then saved to a log file, and the machine must also be register, via the identSNAPSHOT
The function is not only necessary for warranty issues but also for service personnel (records of new and previous service visits) and also for quality control measures.

Module objective:

Upon completion of this module you will be able to perform the following tasks:

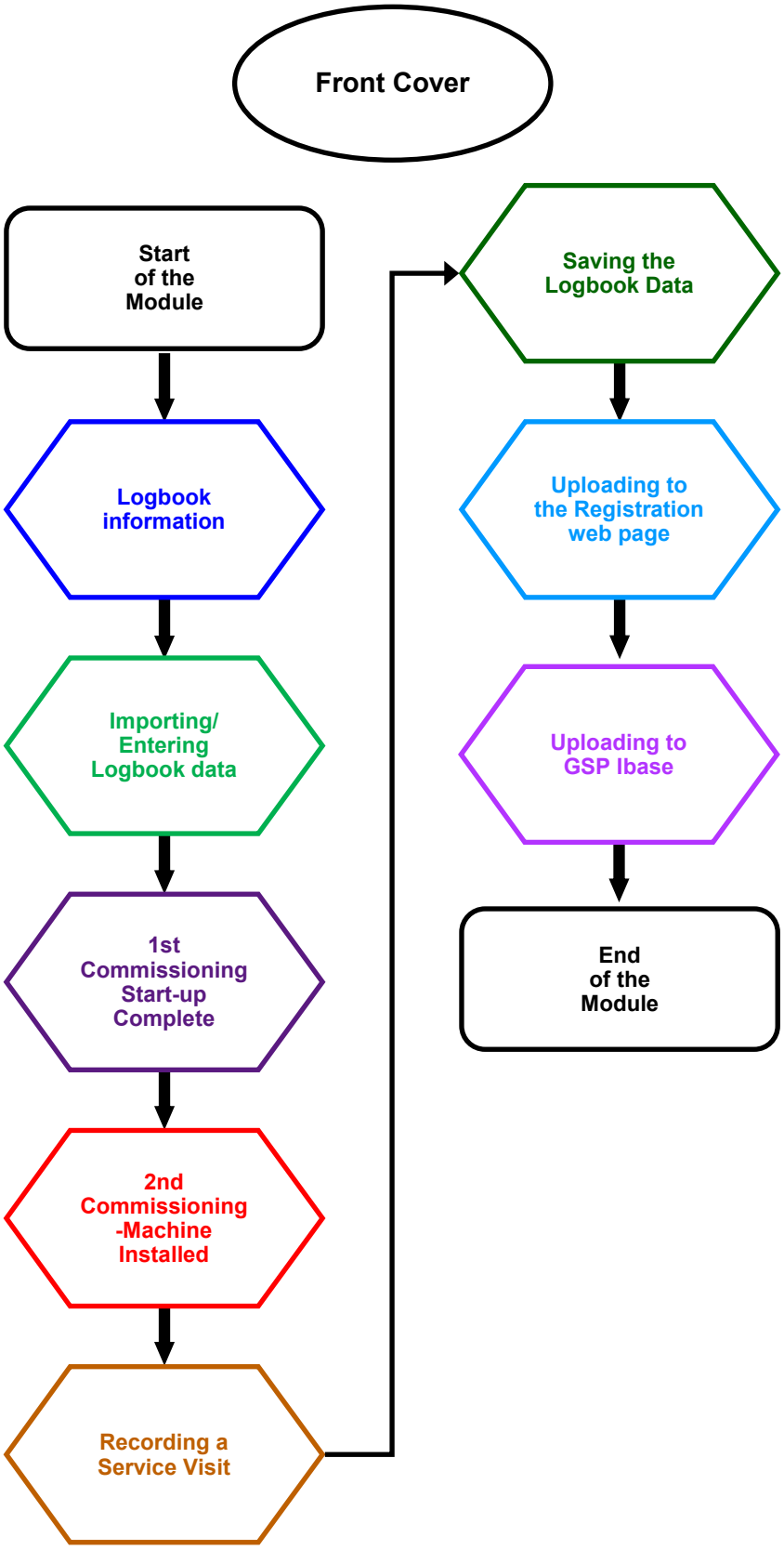
- Import/Enter Logbook data
- Make logbook entries
- Save and load the log file and registration of the machine via the identSNAPSHOT software.

Content:

Sinumerik

Logbook information
Importing/Entering Logbook data
1st Commissioning-Start-up Complete
2nd Commissioning-Machine Installed
Recording a Service Visit
Saving the Logbook data - identSNAPSHOT
Uploading to the Registration web page
Uploading to GSP iBase—828D ONLY

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Logbook information
Importing/ Entering Logbook data
1st Commissioning Start-up Complete
2nd Commissioning -Machine Installed
Recording a Service Visit
Saving the Logbook Data
Uploading to the Registration web page
Uploading to GSP Ibase
End of the Module

Logbook Information

In the past, the validity of warranty claims was difficult to manage and was the cause of delayed service response times. System/hardware faults were logged in paper logbooks which were then usually left on the machine and often became damaged or lost.

By using GSP iBase and the Electronic Logbook system, the warranty will automatically start, a record of the machine details and breakdowns will also be available on the controller.

The time and date of each entry is recorded; it is important that the time and date settings of the controller are correct. Incorrect time and date settings can lead to early expiry of the warranty.

It is necessary to make entries at certain stages during the life of the machine:

- Manufacturer information at OEM
- Dealer information at delivery and installation of machine
- End user information at delivery and installation of machine
- 1st Commissioning - End of start-up at Manufacturer
- 2nd Commissioning Machine installed - End of start-up at end user
- All service visits

identSNAPSHOT

When the information has been entered it needs to be saved, this is via a soft-key on the Logbook screen. This procedure will create a log file.

This is known as the “identSNAPSHOT” function and is a snapshot of the system including version data, cycle data, hardware components etc

Each time a logbook entry is made, the log file should be saved and uploaded to the GSP iBase database via the identSNAPSHOT software.

The name of the file is made from a combination of the machine name and system SD card serial number.

Importing/Entering the Logbook Data

The OEM and Dealer information can be imported to the controller in three ways:

- Import the XML file via the “Import” soft-key. (Preferred method)
- Copy/paste the XML file in the “System data” area.
- Type the information directly on the controller. (When the files are not available.)

The preferred method is to obtain the correct xml files from your Siemens contact partner. These are pre-populated files containing the relevant OEM/machine information etc. Using this method guarantees the consistency between the data stored in GSP iBase and the data stored on the machine.

The following files can be imported:

File	Function
oem.xml	Manufacturer information
oemsubs.xml	Manufacturer subsidiary information
dealer.xml	Dealer information
user.xml	End user information
ma_types.xml	List of available machine types

The files are imported using the “Import” soft-key.
If the files are not available they can be created directly on the controller or offline on a computer.

Machine Identity File (identity.xml)

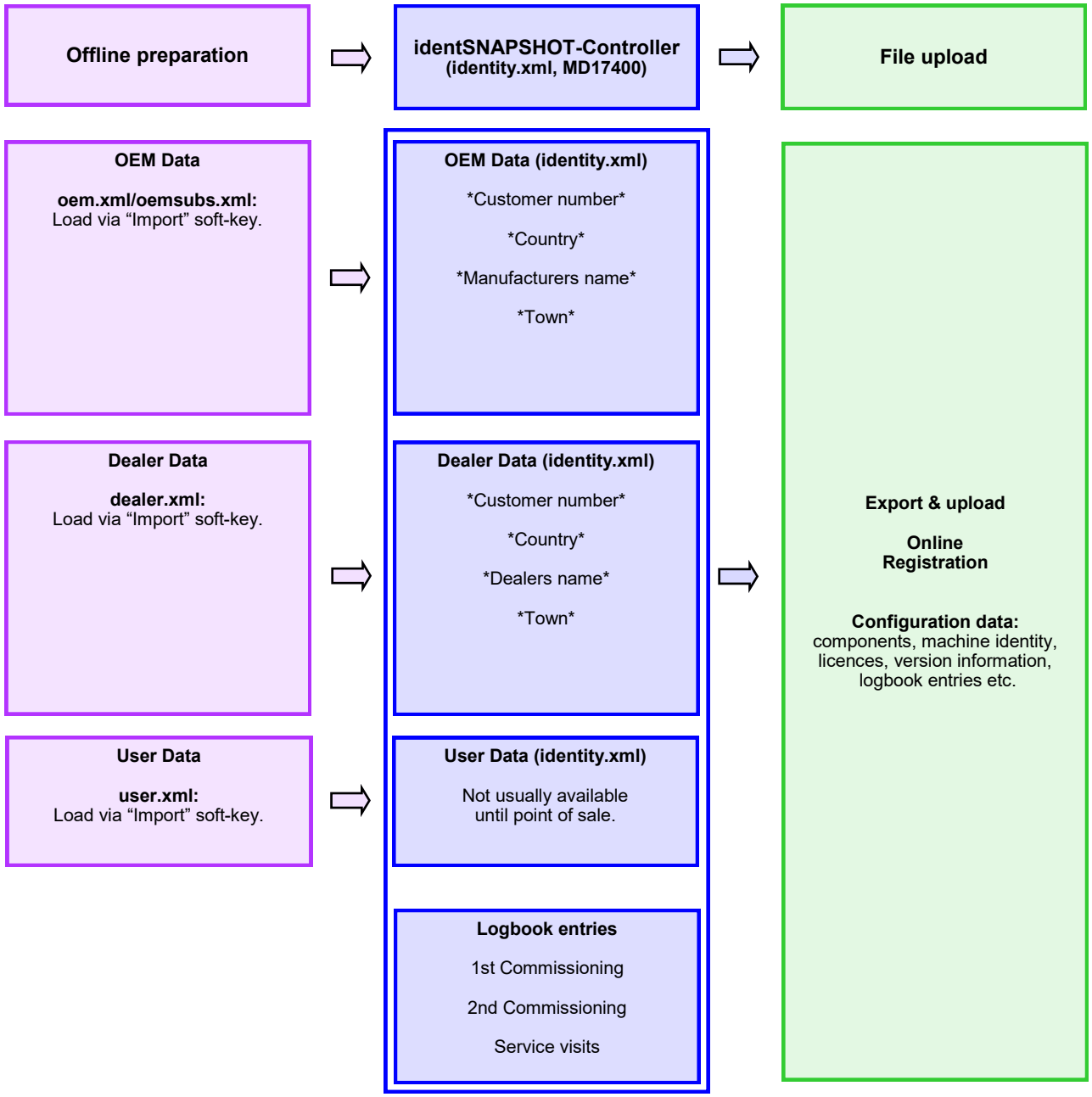
When the information has been imported the “identity.xml” file is automatically created which contains all the information from the imported xml files.

If data is already present in the logbook area and needs to be overwritten, then the identity.xml should first be deleted from the system.

The machine name/number is stored in General Machine Data 17400. This is individual to the machine and needs to be entered here.

Notes:

Concept for Electronic Logbook Data Import and Upload:



If the controller has existing machine identity data, it is necessary to delete the “identity.xml” file.
The location of the file varies between systems operating with a IPC or NCU.

The location of the “identity.xml” file:

IPC: System hard disk\C:\ProgramData\Siemens\MotionControl\user\sinumerik\hmi\data\version\identity.xml

NCU: SINUMERIK Data\user\sinumerik\hmi\data\version\identity.xml

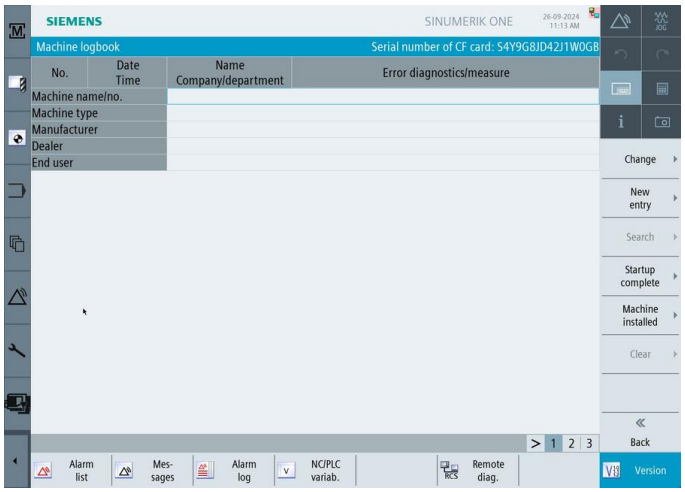
Notes:

Data Import function

The Electronic Logbook is located in the Diagnostics area of the controller:



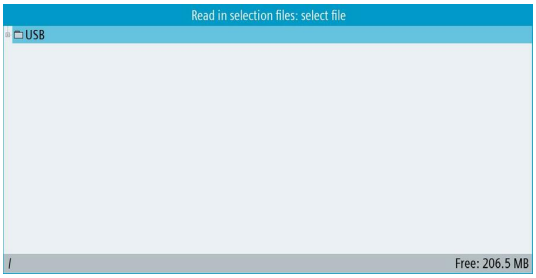
Default Manufacturer screen with no data entered:



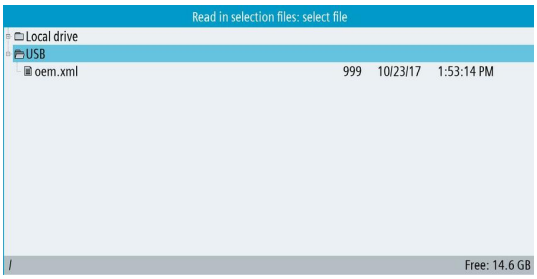
Ensure that the files to be imported are available on the controller, network PC or USB memory device.



Select the location of the file to be imported, e.g. USB.

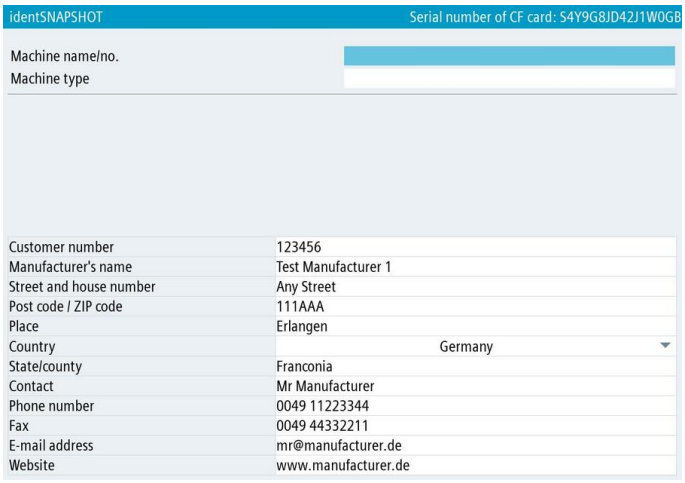


Select the file to be imported, e.g. oem.xml



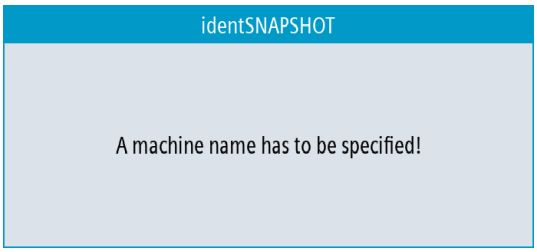


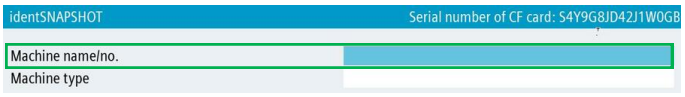
The information will now appear on the screen. In this example the Manufacturer file (oem.xml) was loaded.
To see data for Dealer or End User select the relevant softkey.



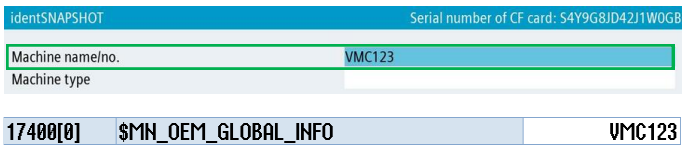


A machine name/number is compulsory.





Enter the machine name and/or number, e.g. VMC123.
The value entered here is transferred to General MD 17400.





Notes:

Machine type

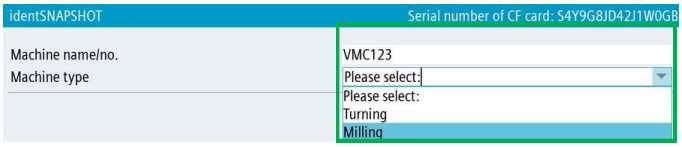
The machine type can be manually typed at the same time as the name/number is entered. Alternatively a list of machine types can be viewed if the “ma_types.xml” has been imported.

Example of an “ma_types.xml” file that has been edited to include three machine types.

USB/ma_types.xml

```
<?xml version="1.0" encoding="utf-8"?>¶
<Machinetypes>¶
  <Type>Turning</Type>¶
  <Type>Milling</Type>¶
</Machinetypes>¶
```

When the file is imported, the machine types will appear in the drop down list.



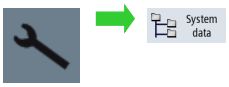




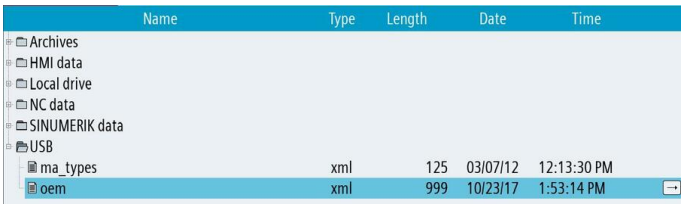
Once the choice of machine type is made it cannot be undone unless the “identity.xml” file is deleted.

Copy and Paste function

It is also possible to copy the files from USB/networked PC via the “system data” area using the copy and paste functions.



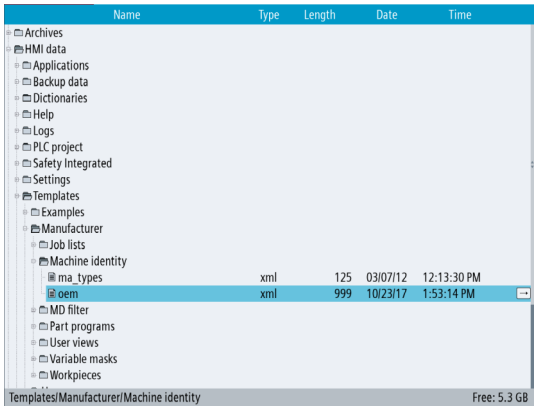
Cursor to the source containing the file e.g. USB device.





Locate the destination folder (Machine identity).





The controller now needs to be powered off/on.

Notes:

Subsidiary files

Subsidiary files are used to define the details of manufacturer regional offices.

Example: A machine manufacturer has its head office in Germany, and also has regional offices in other countries, e.g. United Kingdom.
In this case, “oem.xml” would contain the details of the head office; “oemsubs.xml” would contain the details of the regional office.
When an “oemsubs.xml” file is imported to the controller, the “Manufacturer” soft-key is used to toggle between the OEM and OEM subsidiary data.

Example: Import the “oem.xml” and “oemsubs.xml” files as described previously.



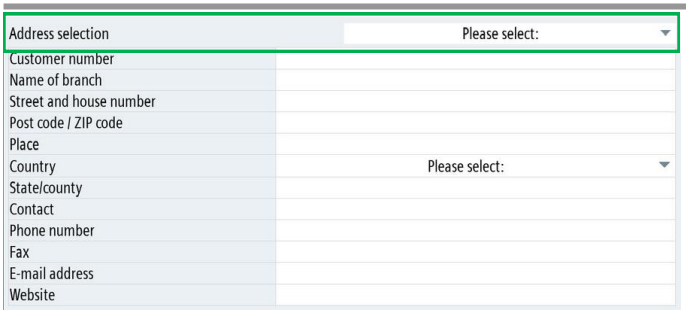
The default screen displays the manufacturer data.



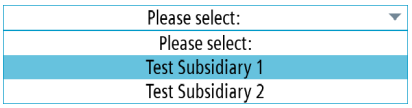
Select the “Manufacturer” soft-key to view the “oem subsidiary” data



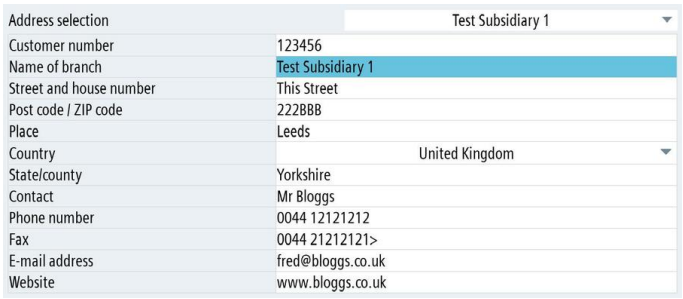
No data will be displayed until an “Address selection” has been made. This is the name of the branch, i.e. the subsidiary.



Select the branch/subsidiary name;
in this case “Test Subsidiary 1”



The subsidiary details will be displayed.

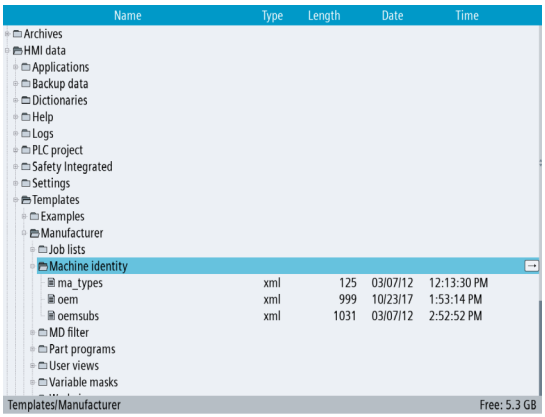


To return to the manufacturer details, select the “Reg. office of manufact.” soft-key.
This the original manufacturer soft-key with different text. The text of the soft-key changes depending on the current selected screen.



Templates for xml Files

Templates for the various xml files are available in the “HMI data\Templates\Examples\Machine identity” folder.



Notes:

The templates can be used if the files cannot be obtained from the Siemens partner. They should be copied and edited to leave the original templates intact.
Example template for dealer.xml.




```
<?xml version="1.0" encoding="utf-8"?>¶
<Addresses>¶
 <Dealer>¶
  <Customer-Id></Customer-Id>¶
  <Name></Name>¶
  <Street></Street>¶
  <ZIP-Code></ZIP-Code>¶
  <Location></Location>¶
  <Country></Country>¶
  <State></State>¶
  <Contact></Contact>¶
  <Phone></Phone>¶
  <Fax></Fax>¶
  <E-mail></E-mail>¶
  <URL></URL>¶
 </Dealer>¶
</Addresses>¶
```

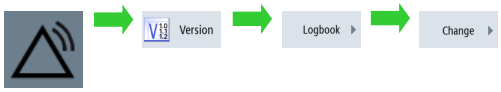
The file contains various xml commands but no dealer data.

Example: A “dealer.xml” file with dealer information. The file can be imported using the method described earlier.

```
<?xml version="1.0" encoding="utf-8"?>¶
<Addresses>¶
 <Dealer>¶
  <Customer-Id>123456</Customer-Id>¶
  <Name>Test Dealer 1</Name>¶
  <Street>This Street</Street>¶
  <ZIP-Code>333CCC</ZIP-Code>¶
  <Location>Dublin</Location>¶
  <Country id="IE">Ireland</Country>¶
  <State>Dublin</State>¶
  <Contact>Mr Murphey</Contact>¶
  <Phone>00353 98989898</Phone>¶
  <Fax>00353 89898989</Fax>¶
  <E-mail>mr@murphey.ie</E-mail>¶
  <URL>www.murphey.ie</URL>¶
</Dealer>¶
```

Manually Entering Machine Identity Data

If the data has to be manually entered due to lack of an xml file, the data can be typed directly into the controller.

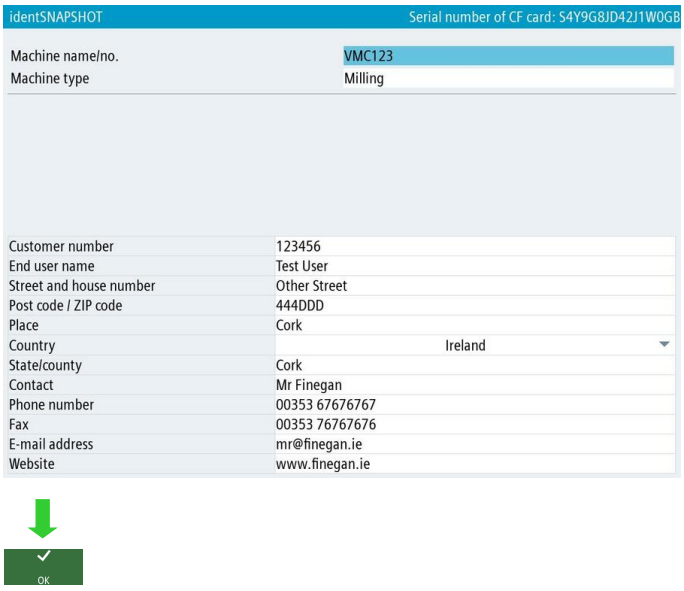


Select which data is to be entered; Manufacturer, Dealer, End user.

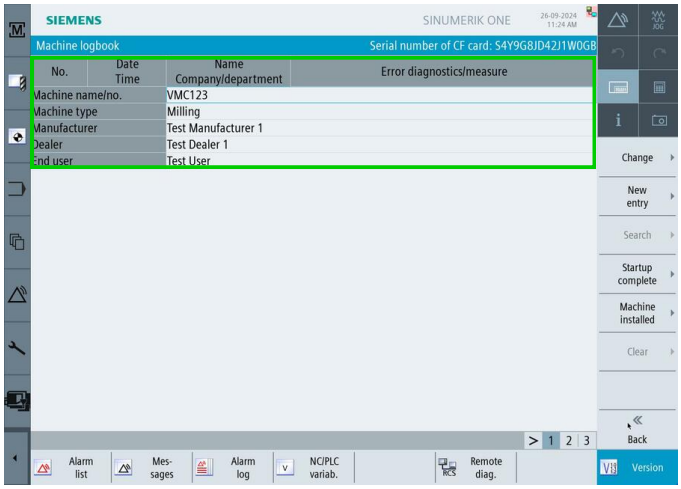
Manu-facturer Or Dealer Or End user

Example: Manually entered “End user” data.





A summary of the information can be seen.

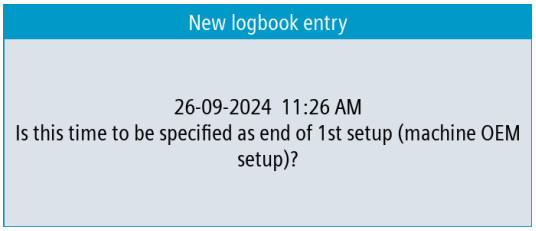


Notes:

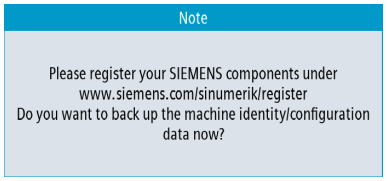
1st Commissioning-Start-up Complete

When the machine has been commissioned at the manufacturer the “Start-up Complete” soft-key is selected. This automatically logs the current time and date.





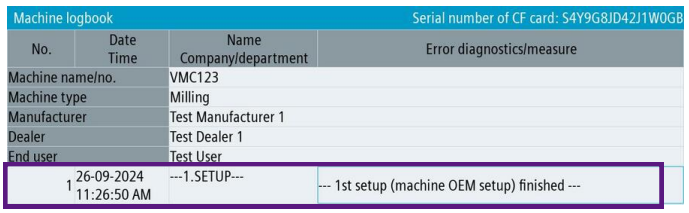






To confirm that 1st Commissioning is complete.

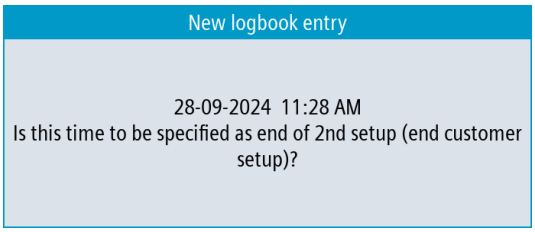
The information will appear on the main logbook page.



2nd Commissioning-Machine Installed

When the machine has been installed at the end-user (2nd commissioning) the “Machine Installed” soft-key is pressed This automatically logs the current time and date.



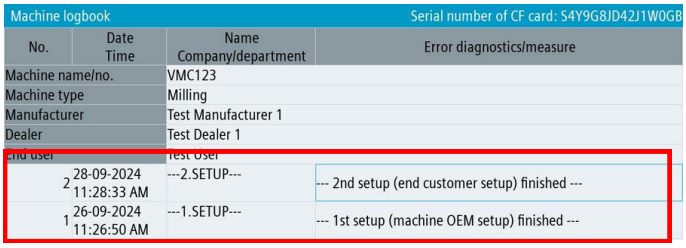






To confirm that 2nd Commissioning is complete.

The information is now added to the main logbook page.



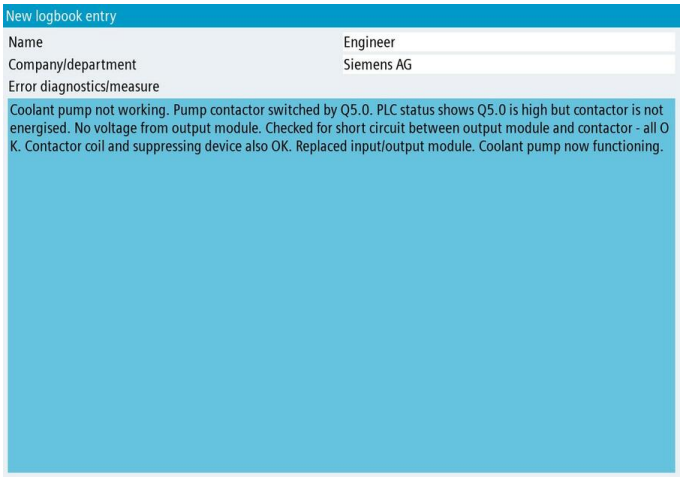
Recording a Service Visit

Whenever an engineer visits the machine to carry out maintenance etc., a record of the visit and information about the work carried out should be entered into the Logbook.

To do this select the “New Entry” soft key.



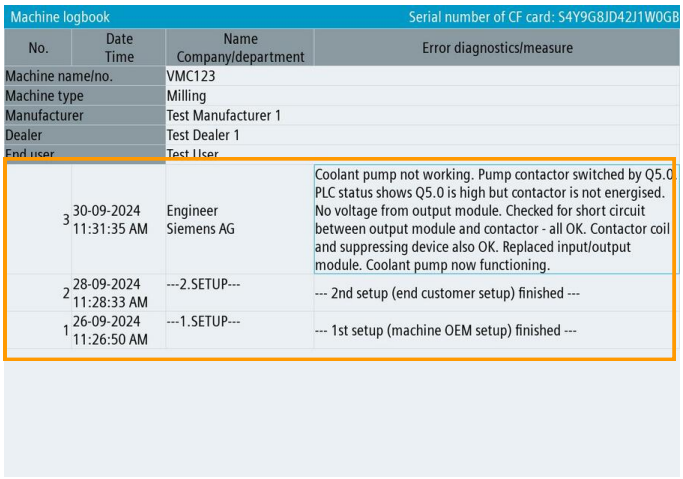
The name and company/department of the engineer should be entered followed by a detailed description of the fault and work carried out.





Notes:

The details of the service visit are now shown on the Logbook page.



Saving the Logbook data

The following procedure shows how to save the newly entered data to a file.
The file contains all information about the system (Controller, drives, software and hardware etc) as well as the information entered on the logbook pages.

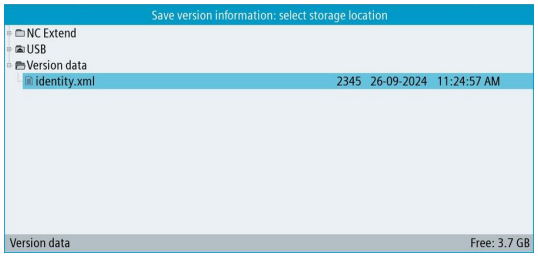
The save procedure should be carried out after each data entry in the logbook.

To save the data to a file first locate and select the “Save” soft-key.

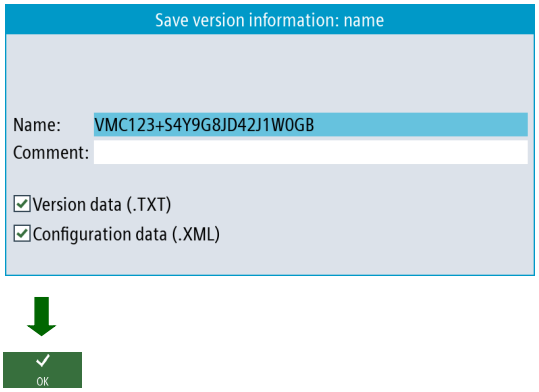


The file should be saved to the default “Version data” folder as a standard procedure to ensure that a copy is locally available .

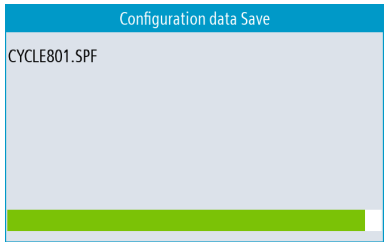
The file is to be uploaded to the online registration website via an external PC, therefore it should also be saved to an external USB device.



The file is given a default name using details from the machine information page. This can be changed. A comment can also be added.
Two files are created by default, Version data and Configuration data.
The “Version data (.TXT)” file is a text file and contains details regarding the versions of the controller, drives etc.
The “Configuration data (XML)” file is an xml file and contains more information about the whole system including the logbook entries. It is this file which needs to be uploaded to the database via the internet.



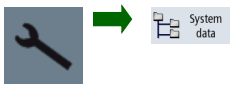
The information will now be saved.

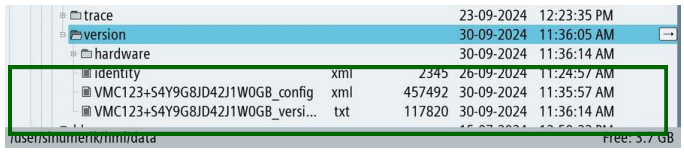


In this example, the files are saved in the “version” folder.

IPC: **System hard disk\C:\ProgramData\Siemens\ MotionControl\user\sinumerik\ hmi\data\version**

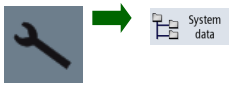
NCU: **SINUMERIK data\user\sinumerik\hmi\ data\version**

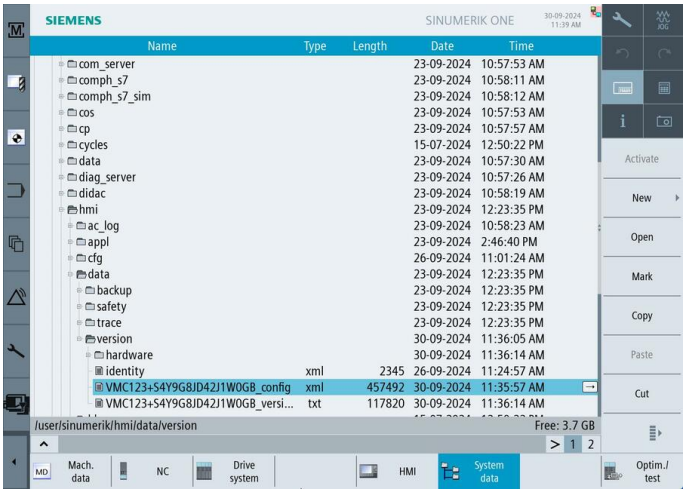




Notes:

The file can be opened for viewing in the “System data” area:





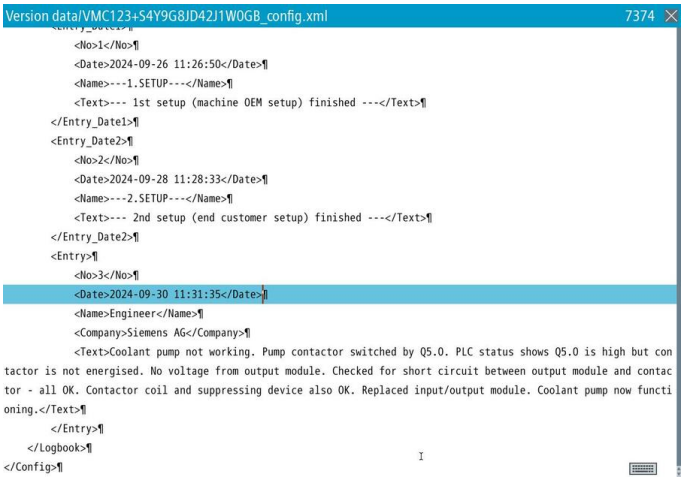


Open

The information that was entered via the Logbook screen is now stored in the xml file.



Service visits are found at the end of the file:



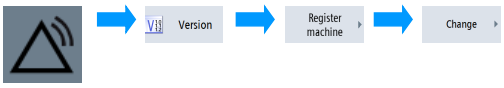
The file is closed with the following soft-key selection.



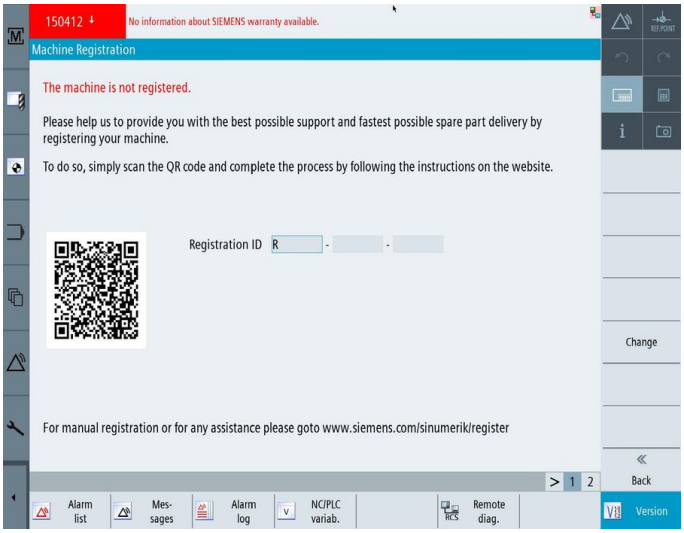
Uploading to the Registration web page

The machine will post an alarm 150412 “No information about SIEMENS warranty available”. To remove this alarm the machine must be registered, this can be done using the link below:
www.siemens.com/sinumerik/register

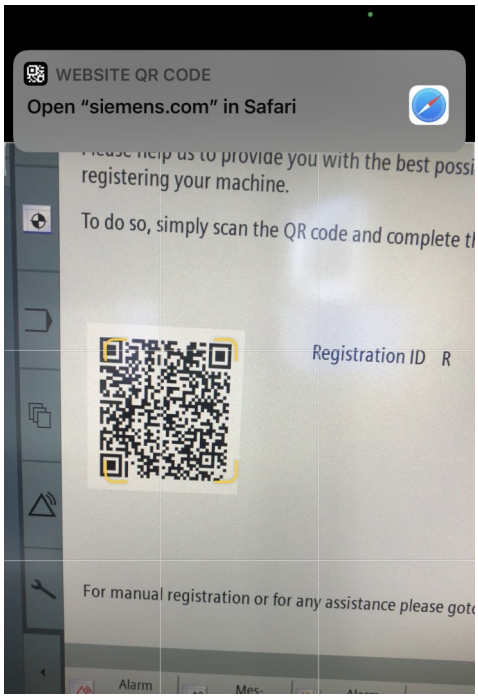
But the preferred method is to use the QR code that is present on the machine.

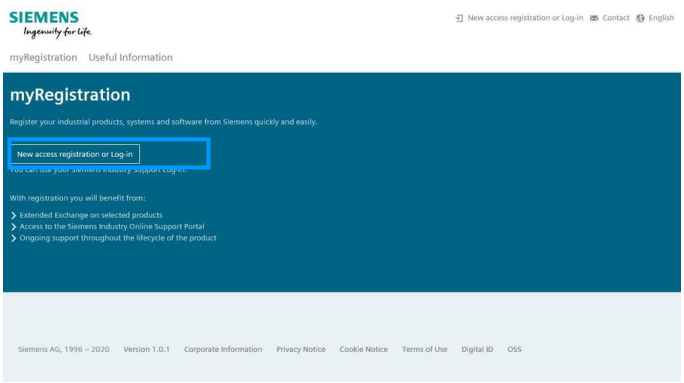


Notes:

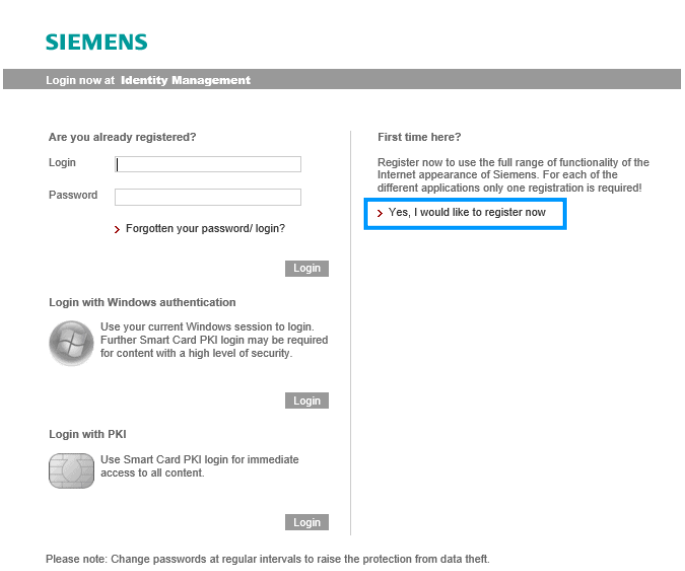


Use a device to scan the QR code, which take information from the machine and display a link to the “registration web page.





For first time users, registration is required.

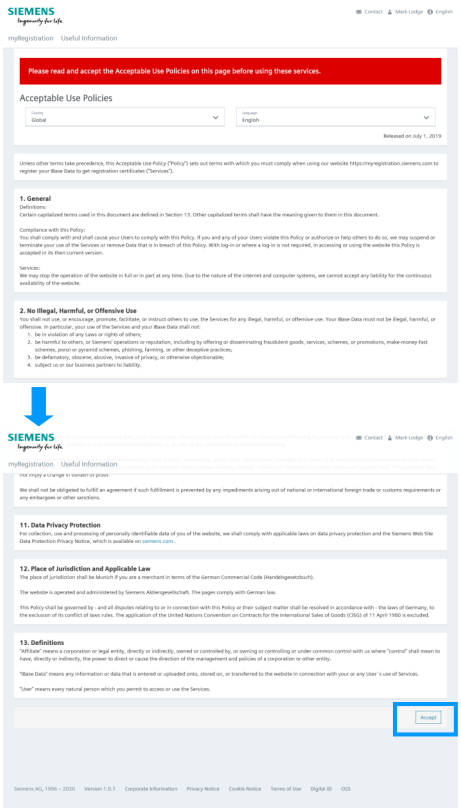


Once you have registered you can login using your “Login and Password” details.

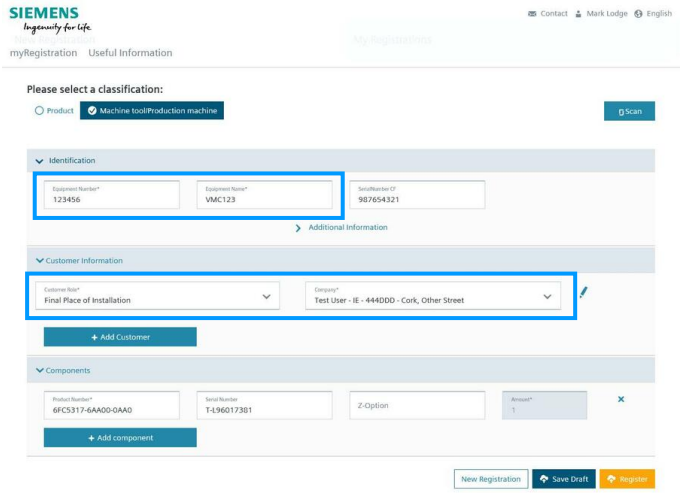
If you are employed by Siemens, your PKI card can be used.

Notes:

Accept the “Acceptable Use Policies”



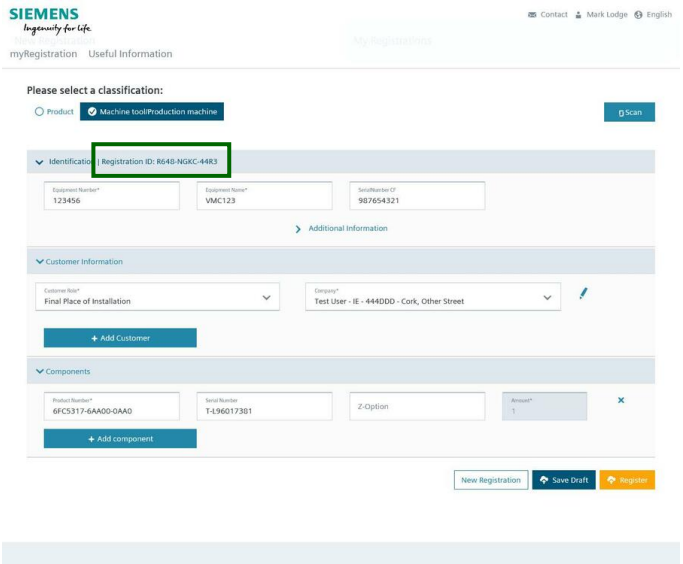
Details from the QR code are then entered into this page, these can be viewed under the “Component” menu.
But first you have to must enter details that have an “*”, for Equipment Number*, Equipment Name*, Customer Role*, and Company*.



Once all entries are populated the machine can be “Registered”.

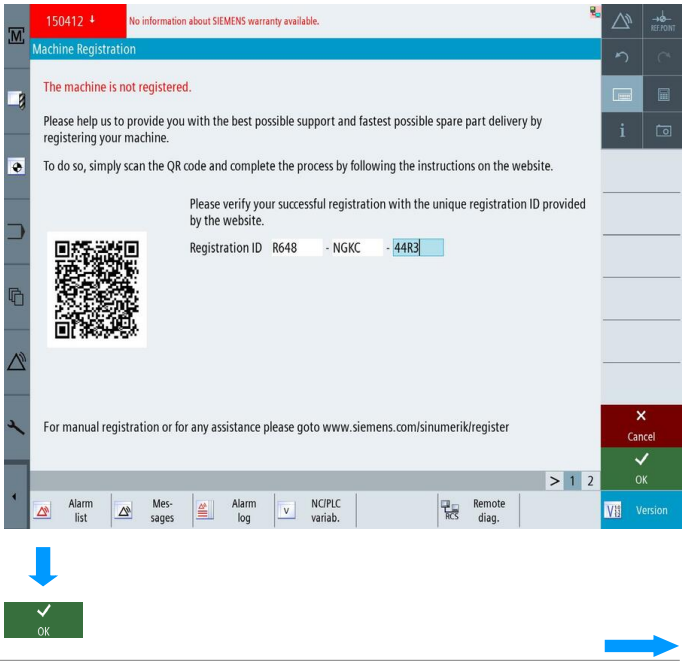


This well then create a “Registration ID”

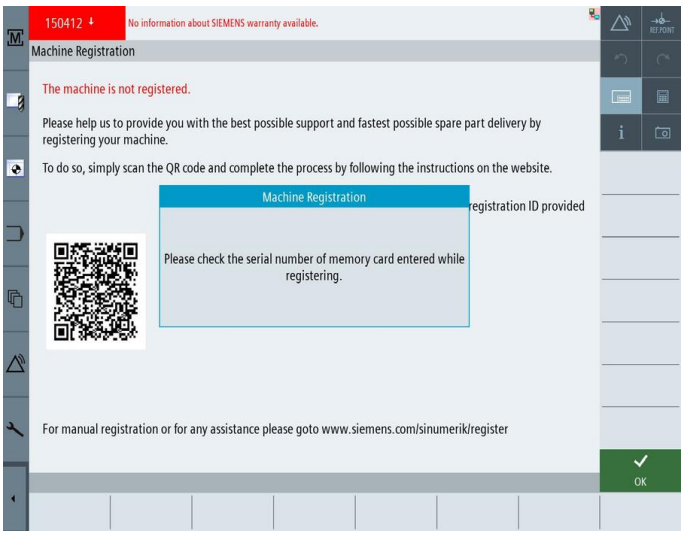


This “Registration ID” can then be entered into the control.

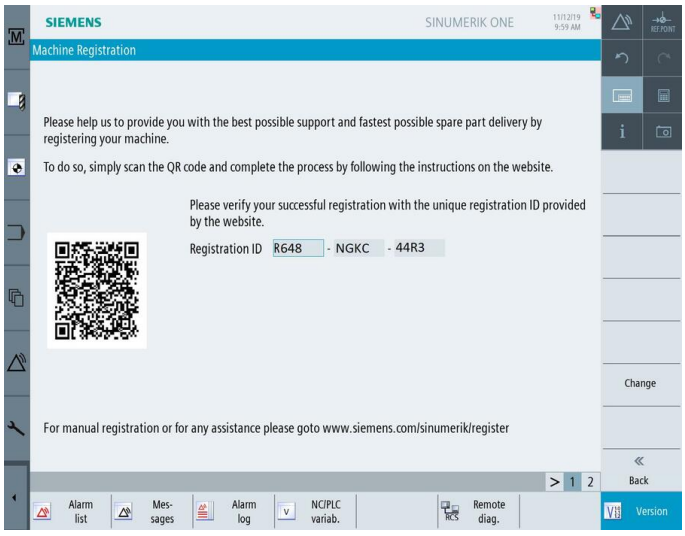




Notes:







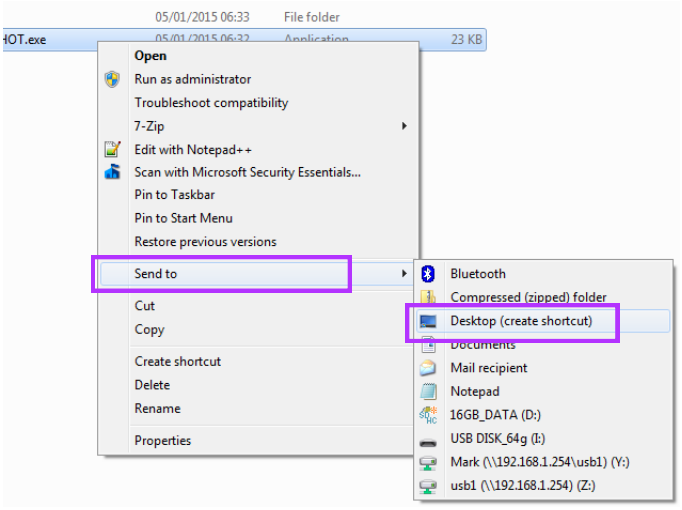
The alarm 150412 “No information about SIEMENS warranty available” has now been cancelled, and the machine is registered.

Uploading to GSP iBase - 828D ONLY

Use the copy/paste function to transfer the xml file to a USB stick in preparation for uploading to the Online Registration Website.

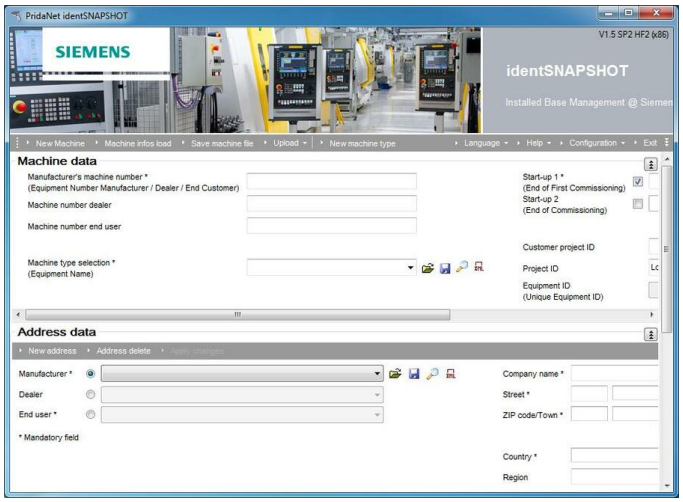
New Log Files
A new log file needs to be created when anything new is added to the logbook.
The file must then be uploaded to the GSP iBase. This is done using identSNAPSHOT software, which can be downloaded via:
www.siemens.com/sinumerik/register

Once identSNAPSHOT has been downloaded, and the zip file has been extracted, a shortcut can be created of the "Start_IdentSNAPSHOT.exe" onto the desktop of the computer being used.

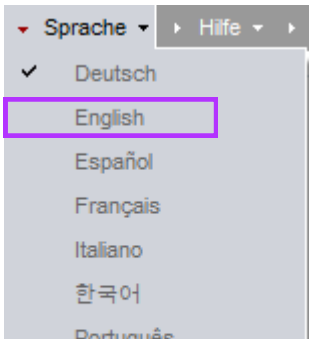


Double click the desktop short cut to start the "identSNAPSHOT" software.

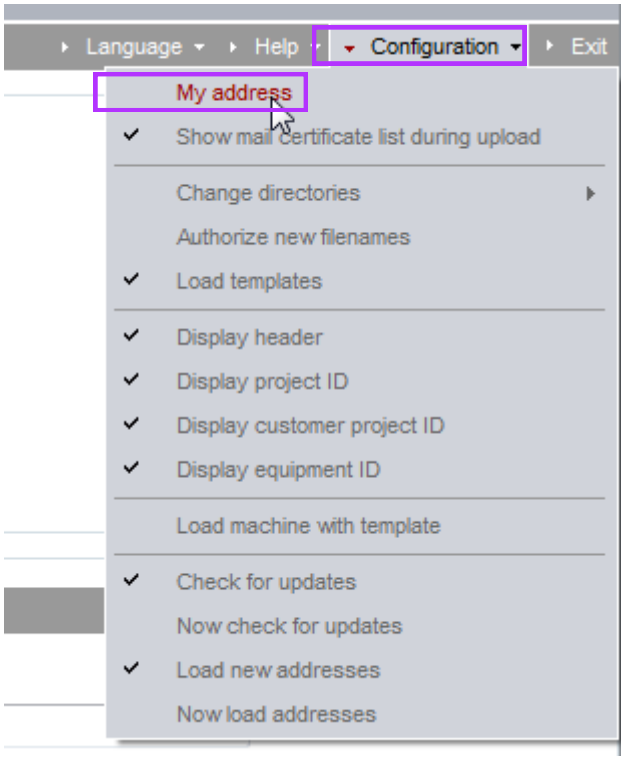
Notes:

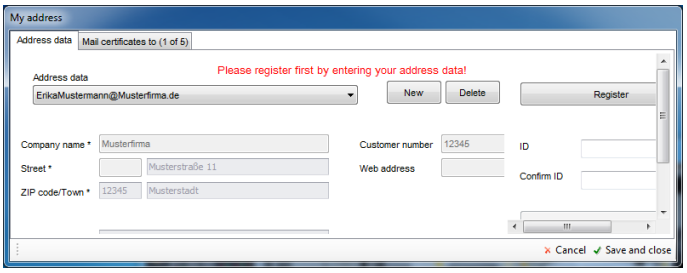


A language selection is available from the menu bar.

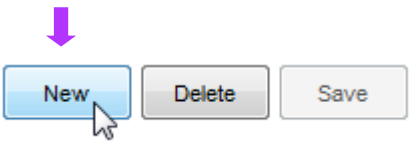


You must first register in order to upload data into the GSP iBase **S**ervice **P**latform **I**ndustry database using identSNAPSHOT. You only have to register once before the first upload. All of the required data are saved locally on your PC.
This is achieved by the pull down menu "Configuration" on the right hand side at the top of identSNAPSHOT, then selecting "My address".





Select "New"

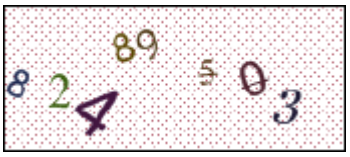


Enter your details that are heighted in RED* followed by.



Notes:

As response, you receive an e-mail from the identSNAPSHOT registration with your user ID in the form of a graphic.



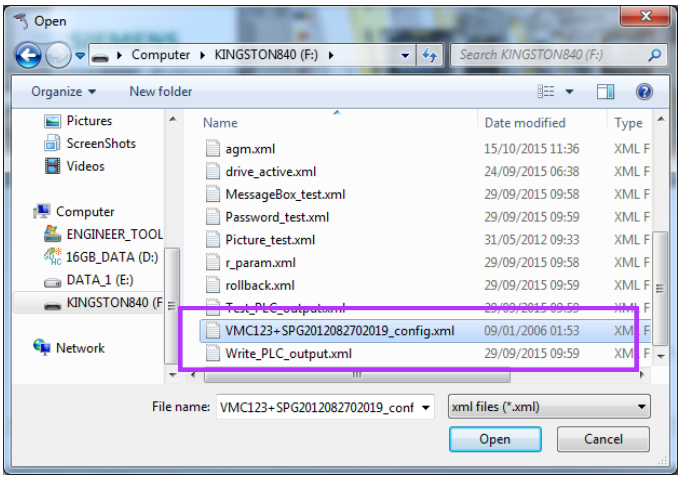
The numerical sequence 82489503 is shown here as example

These digits must be entered in this sequence into the input field "ID" and "Confirm ID". The registration is completed with "OK". If the registration was successful, the e-mail address has a gray background. The user ID is then permanently assigned to this e-mail address, and the e-mail address can no longer be changed.

Save and close

To upload the "Logbook" XML file to GSP iBase, highlight "Machine infos load" from the menu bar.

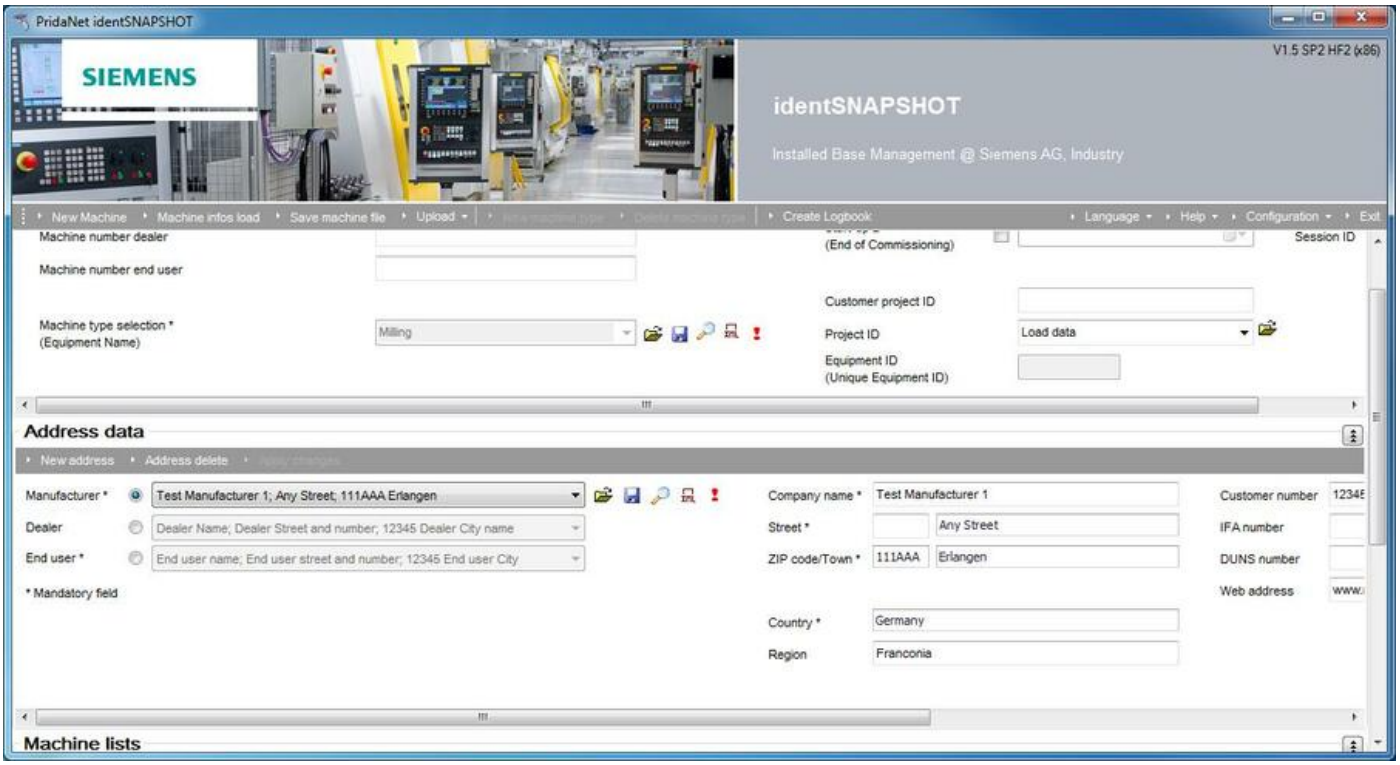




Use the "Open" window to locate the XML file on the memory stick

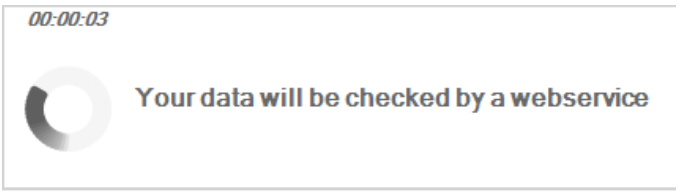
Open

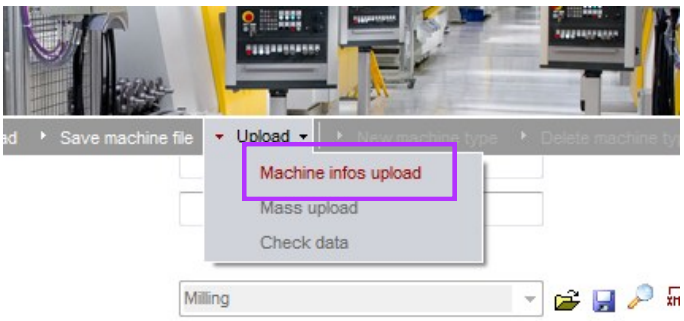
Notes:



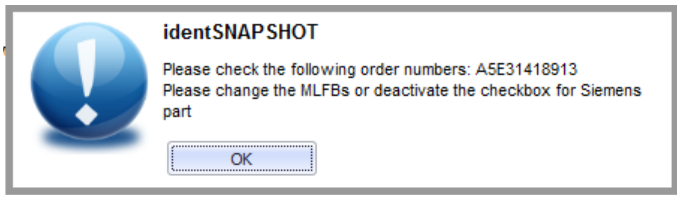
This will populate all relevant fields within identSNAPSHOT, from the data of the "Logbook".

To upload the XML file to GSP iBase use the "Upload" --> "Machine infos upload" from the menu bar.





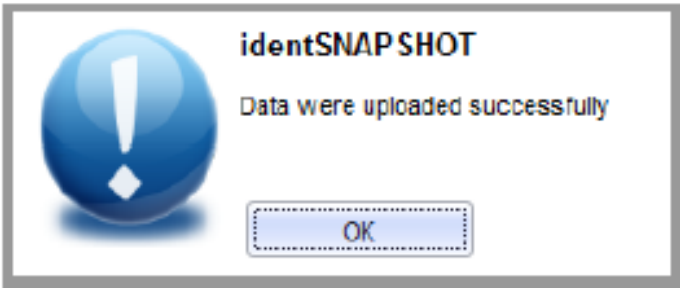
If there is any problems, a window will appear advising you of data to be checked



This will start the upload process.

Notes:

When the "Logbook" file is uploaded successfully, a window will appear.



213: END

Notes:

Notes:

A137

SINUMERIK ONE

Data Management

Module Description:

It is necessary to have procedures in place for saving and restoring the customers data stored in the CNC.

The following cases have to be taken into consideration:

- Series machine production
- Exchange of the IPC/SSD
- Exchange of the NCU/PLC
- Exchange of the system SD card

The back-up method used can differ between machines, depending upon the hardware configuration in use.

This module describes the methods for saving data so that it can be restored in the event of any of the above eventualities.

Module objective:

Upon completion of this module you will be able to perform the following tasks:

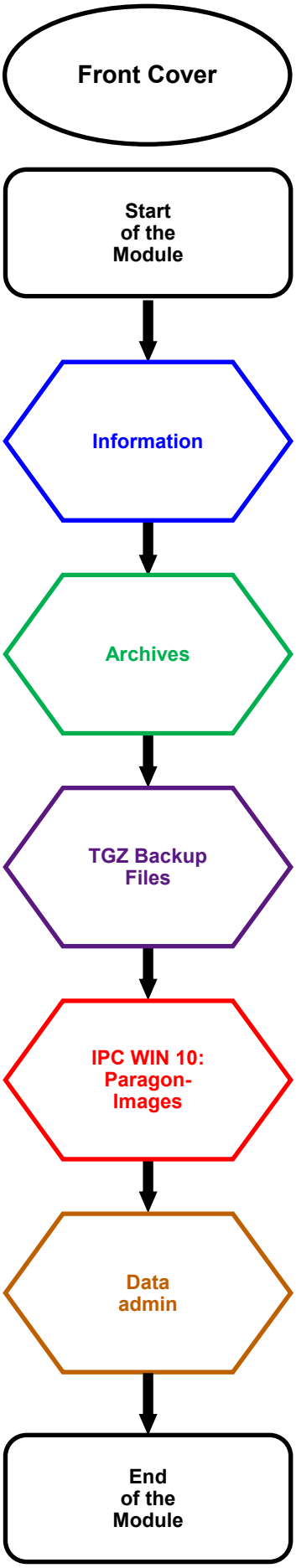
- Create and restore archives
- Create and restore a system image
- Create and restore a Paragon image
- Manage MD, SD, drive data and compensation data on the file level

Content:

Sinumerik

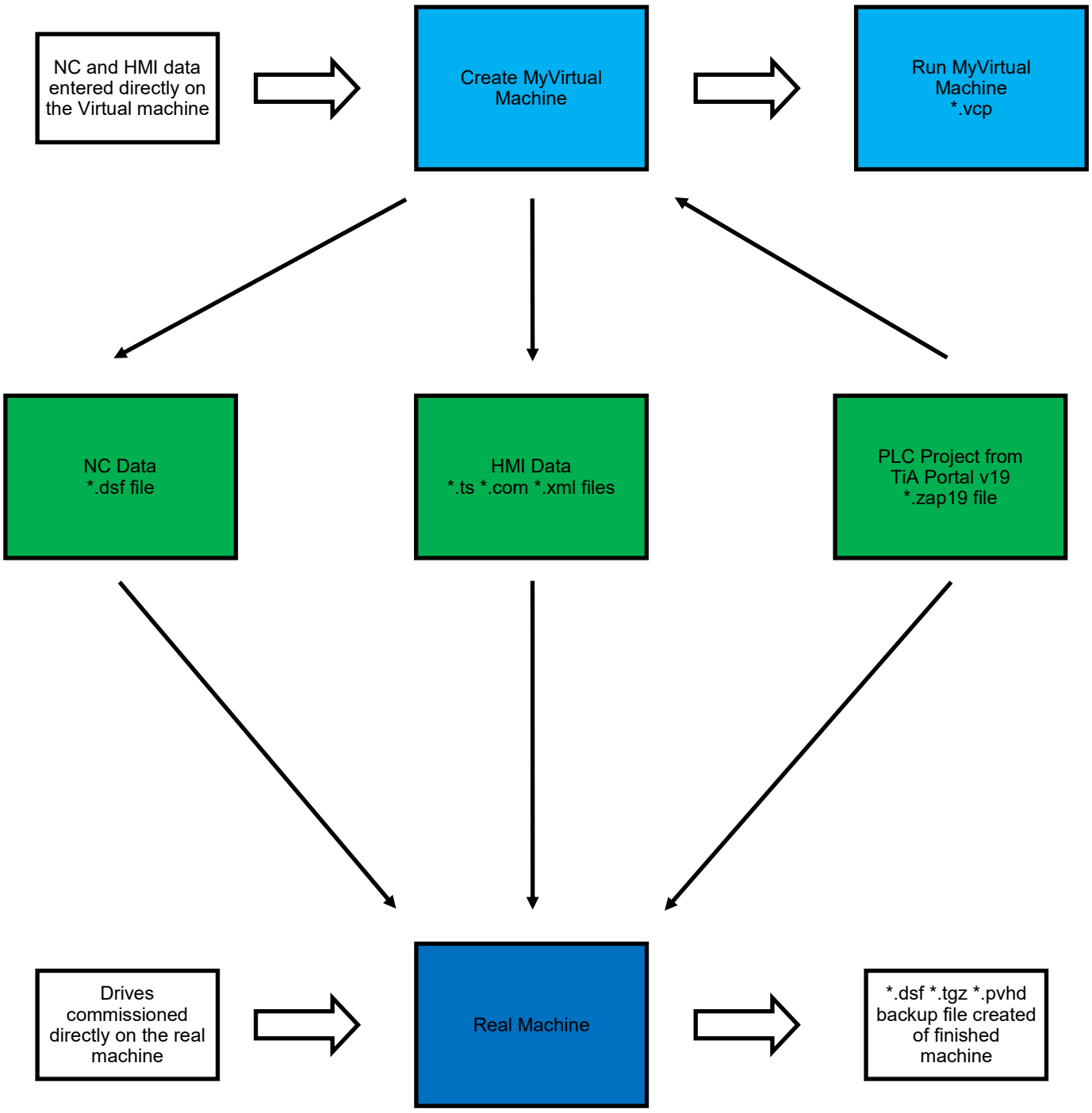
Information
Archives: TCU & IPC
TGZ Backup Files
IPC WIN 10 Paragon-Images
Data admin

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Information
Archives
TGZ Backup Files
IPC WIN 10: Paragon-Images
Data admin
End of the Module

Information

Concept of data movement



Notes:

Archives

Series Setup archives can be used to save NC, PLC, Drive and HMI data.
Archives are created for a variety of reasons:

- Series machine production
- Exchange of the TCU
- Exchange of the IPC
- Exchange of the NCU/PLC
- Exchange of the system SD card

To create a Series Setup Archive locate the relevant area by using the following key sequence.



The screen displayed has five choices:

All data of this machine (backup) - Archives of the five control components (NC, PLC, Drives, HMI, system settings) can be created with this option

Data for a series setup - Archives of the four control components (NC, PLC, Drives, HMI) can be created with this option

Factory settings of this machine (original) - Used to create an archive that will return the machine to its delivery state.

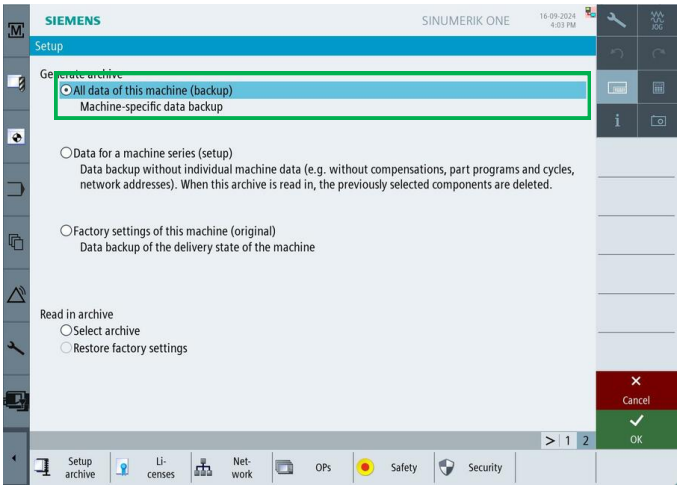
Select archive - Archives are read-in with this option.

Restore factory settings - For reading in the original status archive

Creating an Archive:

The method for creating archives is the same for both IPC & NCU. However, when the data is stored locally it is to different destinations.
In the case of the NCU the archives are stored on the system SD card.
With the IPC the data is stored on the SSD.

Ensure that the option “All data of this machine (backup)” is selected and select “OK”.





Control Components:

The controller data is split into four components

NC data:- The NC data consists of:
All machine and Setting data (except for drive machine data)
Machine data
Setting data
Option data
Global user data (GUD)
Tool and magazine data
Protection zone data
R paramters
Zero offsets
Workpieces, part programs and subprograms
Manufacturer cycles
User cycles
Definitions and macros
Collision avoidance data
Compensation data (QEC, CEC, EEC)
Axis optimization data (AST)
Compile cycles

PLC data: This will create an archive suited to restoring the PLC user program.

Drive data: This will create an archive containing data for the Sinamics drives.

Notes:

HMI data: This saves data that is stored on the HMI.
The list comprises of:-
Cycle storage
Texts
Templates
Applications
Configurations
Version data
Logs
User views
Dictionaries

System settings: This saves data that is stored on the HMI.
The list comprises of:-
Drive configurations
TCU settings
Network settings (for embedded only)
mmc.ini

Local drive: This saves data that is stored on the local drive (NCextend).

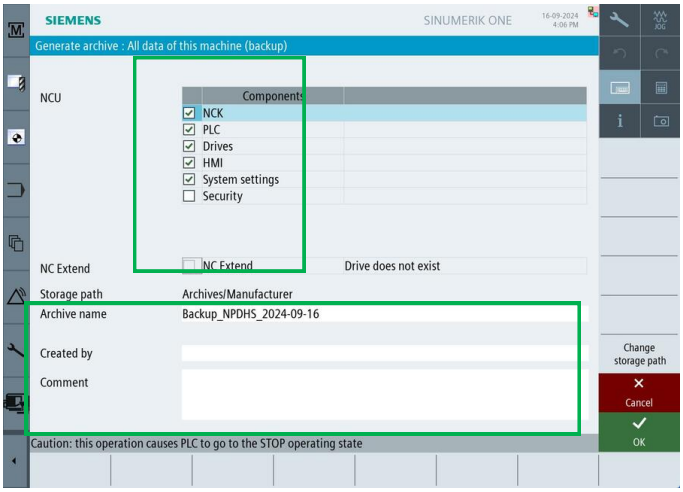
The five control components are available for selection. They can be selected in any combination to create archives containing different data.

Compensation data is machine specific data such as lead screw error compensation. This would not be included in an archive used for series machine production.

Creating an NC archive

The following example will create an NC archive including compensation data.

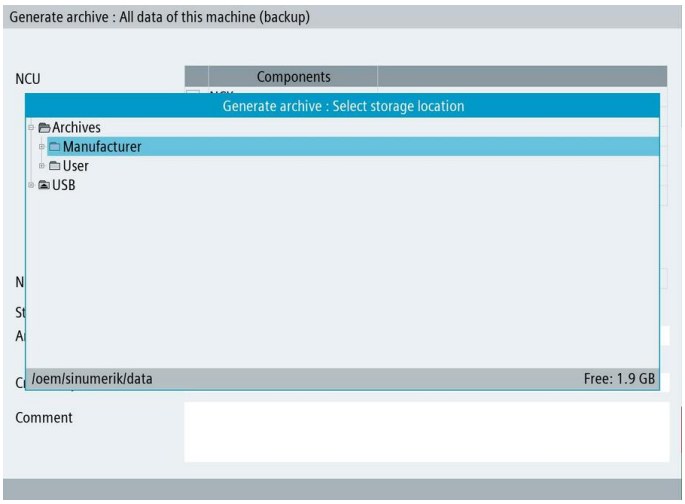
Use the cursor and select keys to select the necessary check boxes.
The comment area at the bottom of the screen offers the chance to add information about the archive.





The destination of the archive can be changed, this can be locally on the system SD card/SSD or externally to a USB device or networked PC.
It is suggested that an archive be kept locally on the controller. Copies can be made externally for use in the event of a data restore.





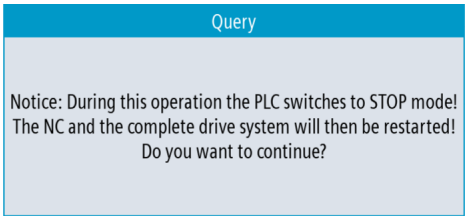
Notes:

A name is automatically entered for the archive, but this can be changed.

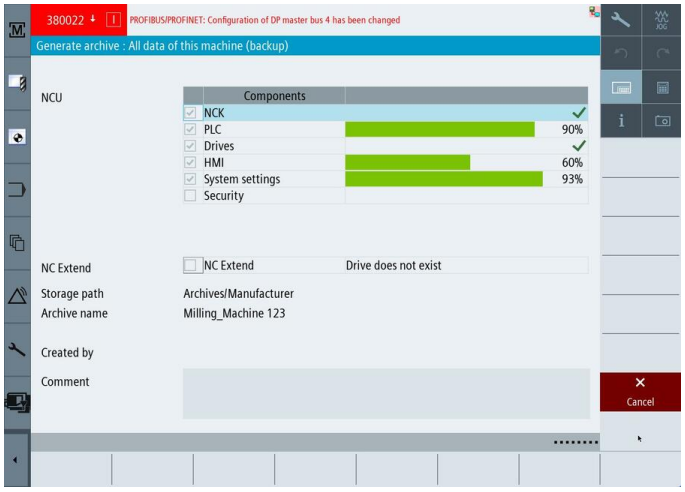




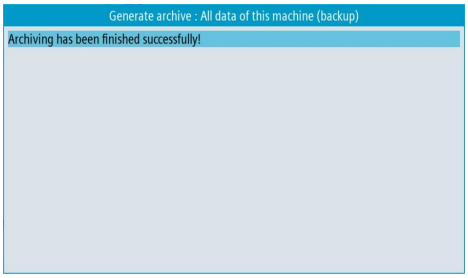
The archive will now be created and stored in the destination specified earlier.







An information box indicates that the archive was successfully created.





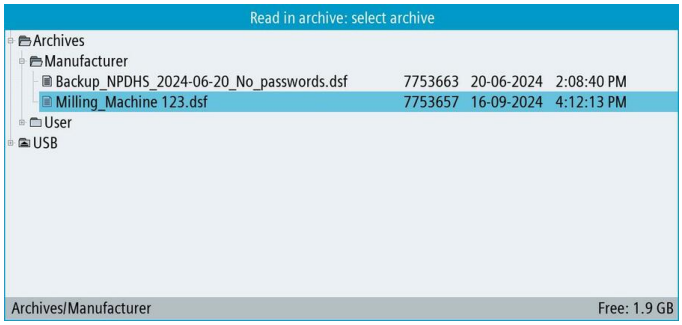
Restoring a Series Setup Archive:

To restore an archive the option ”Read in setup archive” needs to be selected





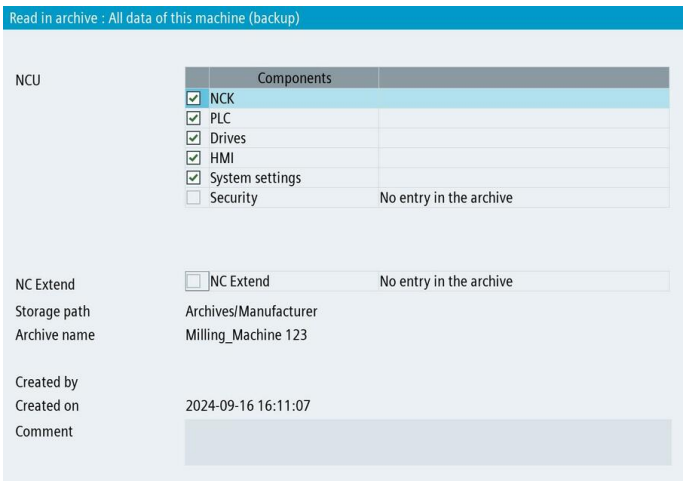
Select the source from which the archive is to read in from.





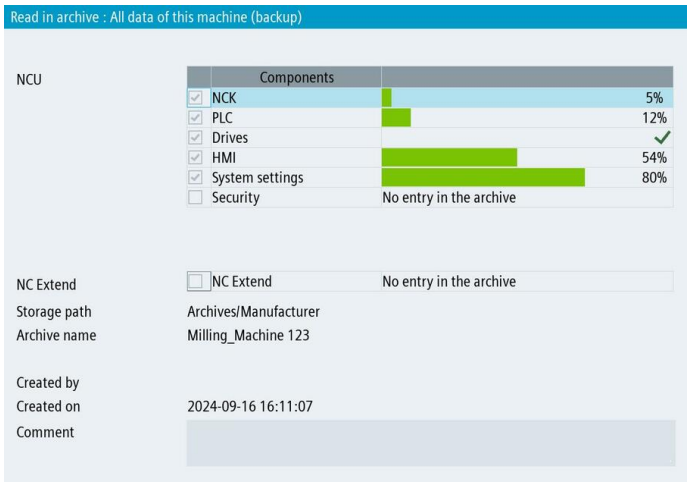
Notes:

Information about the selected archive is displayed. This includes any comments entered by the creator.
The software version of the system that the archive was created is also displayed.

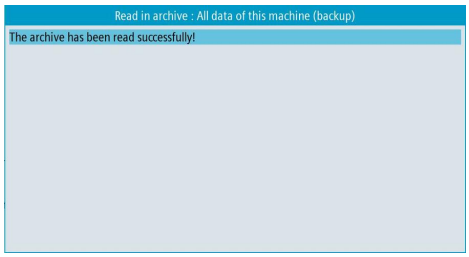


OK

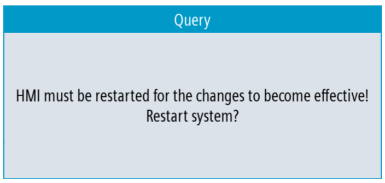
The archive will be read in and a series of status boxes are displayed.



A final status box will be displayed which confirms that the archive was read in successfully.



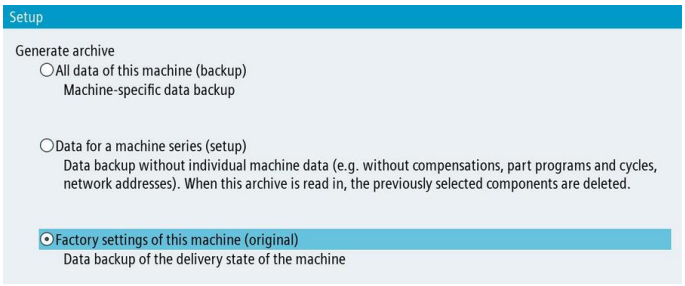
OK



OK

Creating Factory settings of this machine (original) Archives

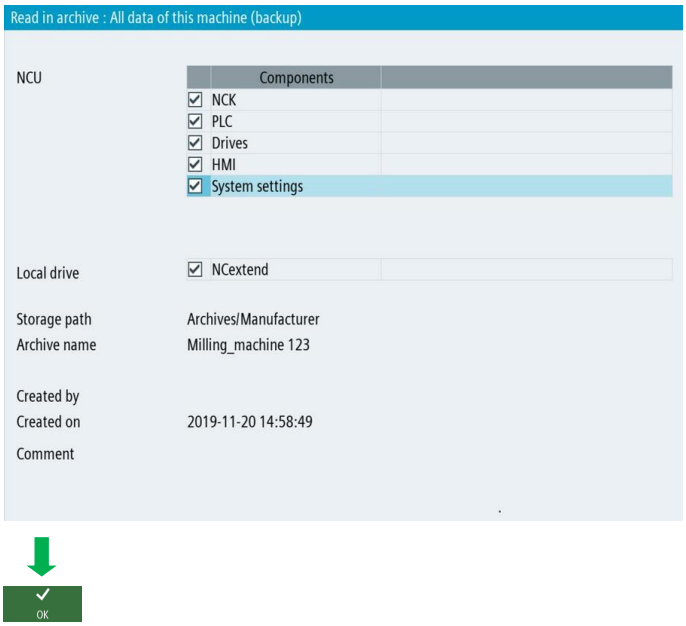
Creating Factory settings of this machine (original) Archives are used to restore the machine to its delivery state. (Part programs, tool data, work offsets etc will not be present)

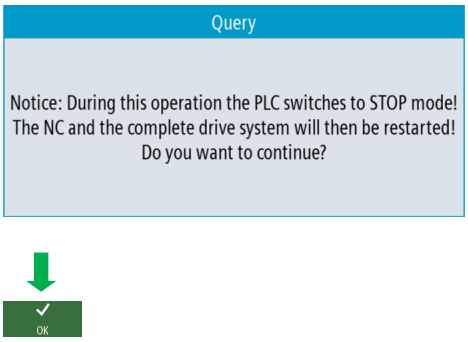


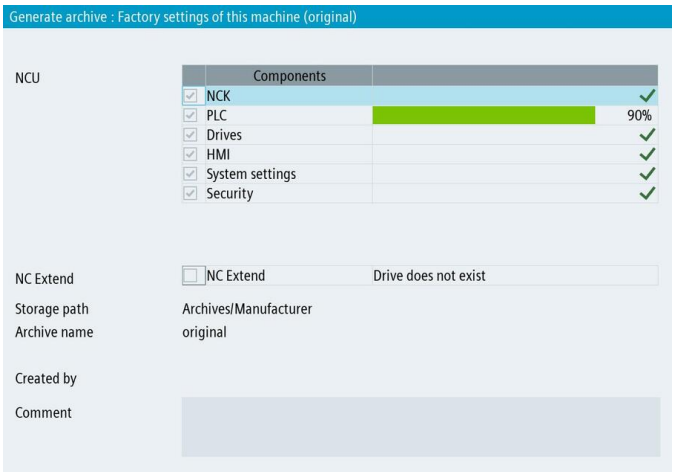
OK

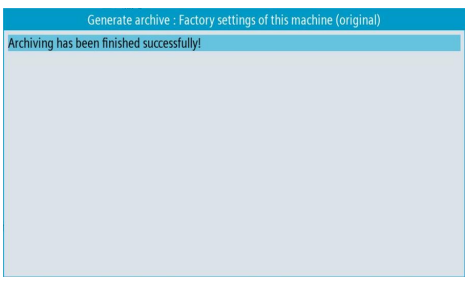
The default selections are shown below. It has to be decided by the machine builder which components are needed to return the machine to its original delivery state.

Notes:











Reading in Original Status Archives

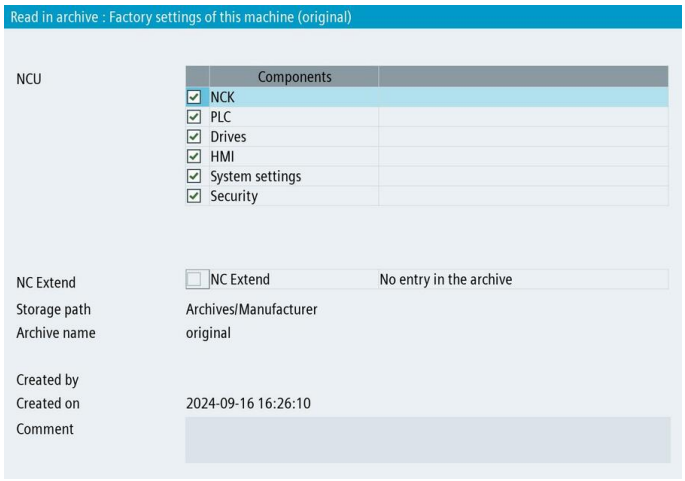
To restore an original status archive, the option ”Read in original status archive” needs to be selected.



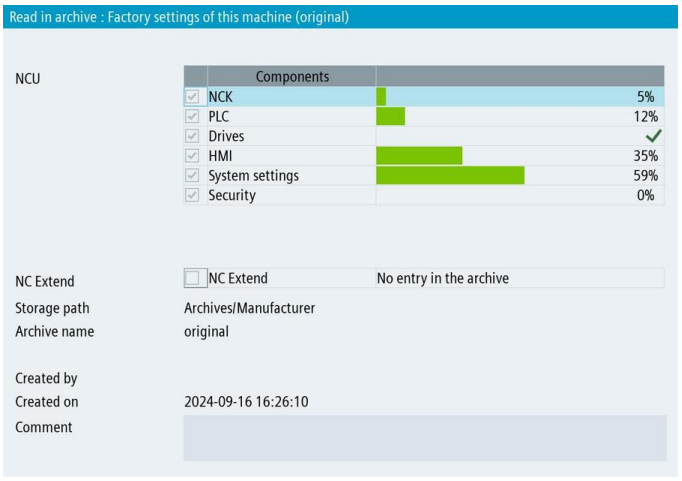


Notes:

As only one Original Archive is allowed there isn't a list to select from. Information about the archive contents is displayed.

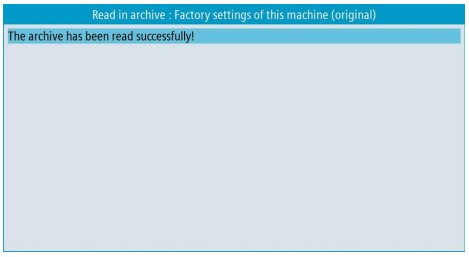


Status boxes will be displayed informing of the progress of the read-in procedure

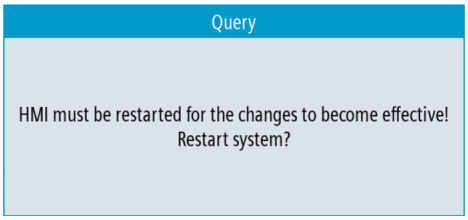


OK

When the archive has been read in a status box is displayed to confirm this.



OK



OK

Notes:

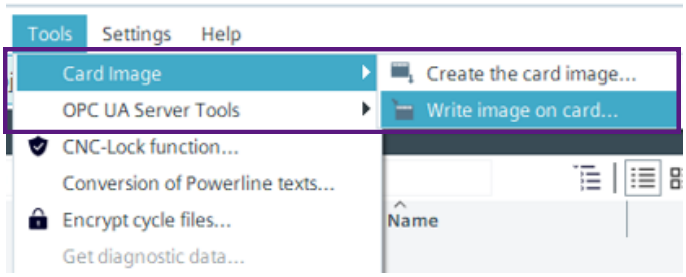
TGZ backup files

Another method of creating a backup of the System SD card is using the Service System.
A USB service stick is required. This is a bootable stick which contains Service System software.
When plugged into interface X125 the system will boot from the USB device into the Service System menu.
From this menu it is possible to create/restore a backup file.

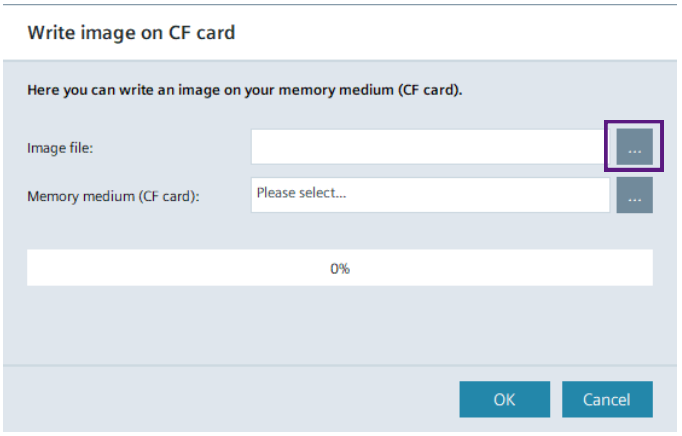
The backup file will be of the **TGZ** type.
TGZ files are ideal for creating a backup for use with series machine production. This is because the licence details contained on the destination SD card will be left intact.

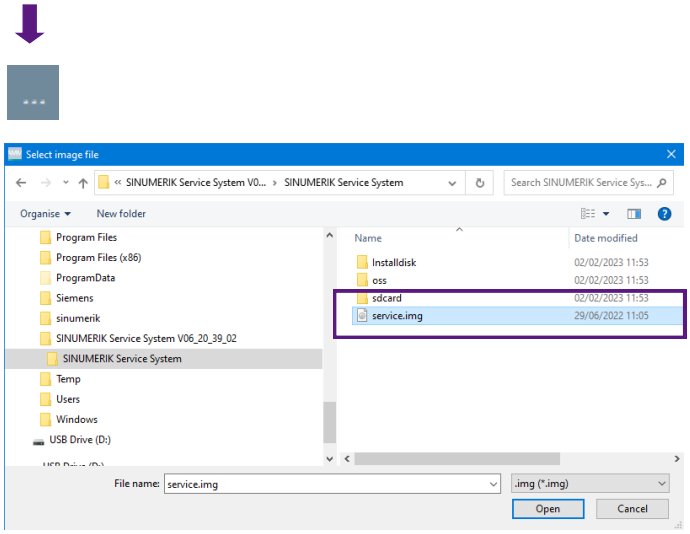
Creating a USB service stick.
Access MyMachine is used for creating a USB service stick.
The software is delivered in “zip” format and should be extracted to a suitable place on a computer.
When extracted, the file “service.img” can be found in the following path:
SINUMERIK Service System NCU V06_60_00_00/
SINUMERIK Service System/service.img

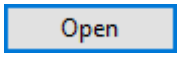
Run the Access MyMachine and select “Writing image to CF card” found on the “Tools” menu.



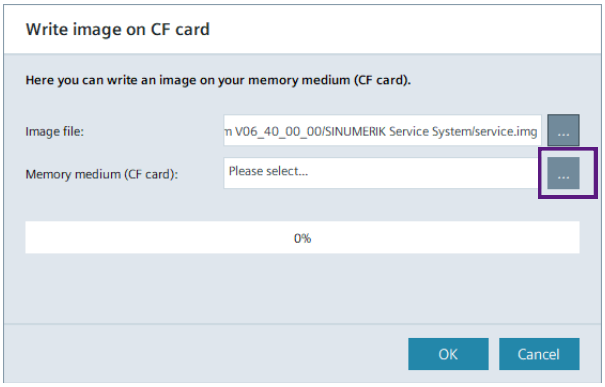
Locate the “service.img” file and select “Open”



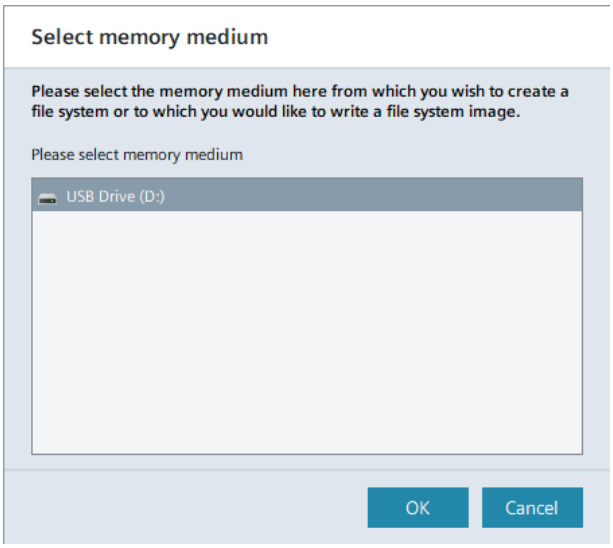




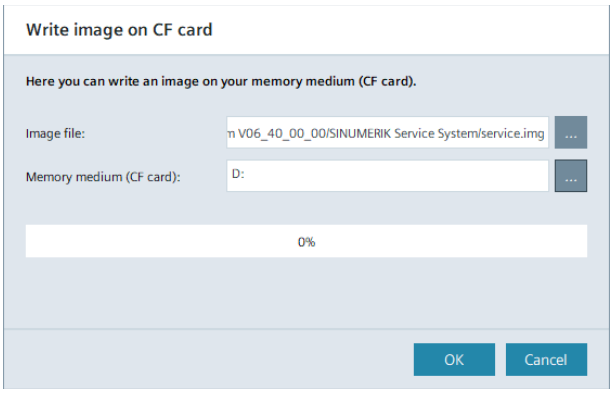
Select the drive letter of the USB stick.



Notes:

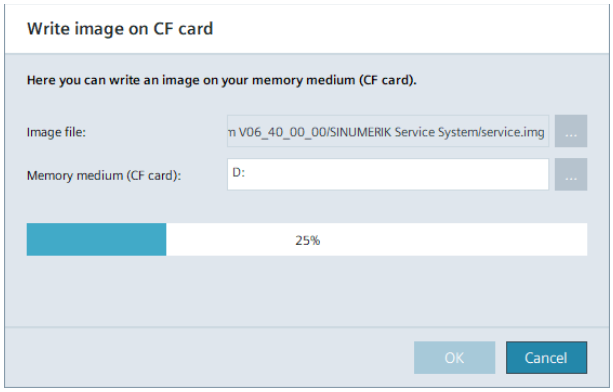


OK

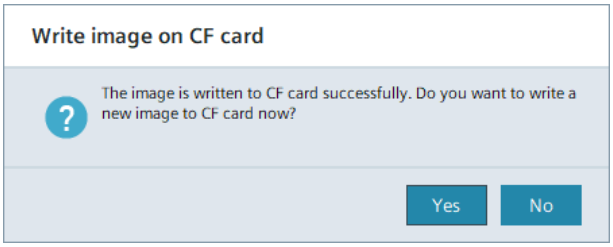


OK

The writing process will begin.



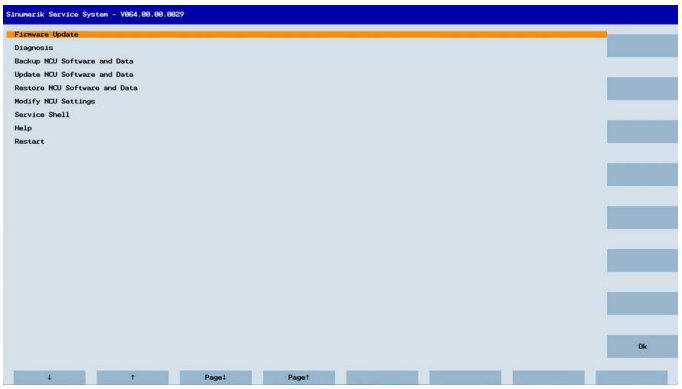
The process can be repeated if required.



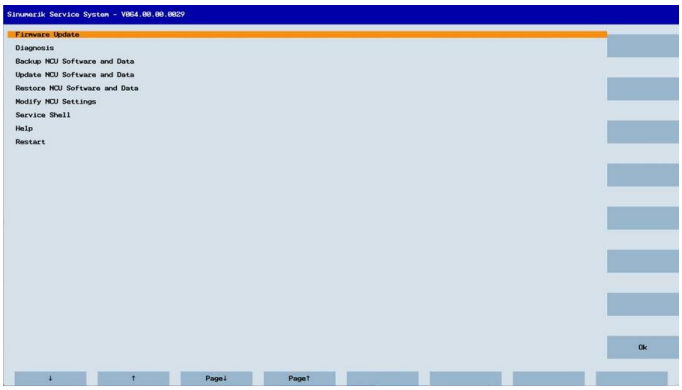
No

Starting the Service System.

Switch off the controller and insert the bootable USB stick into X125 of the NCU, and place the NCK rotary switch in any unused rotary position eg. F.
Power ON the controller.
The following screen will be displayed:



Notes:

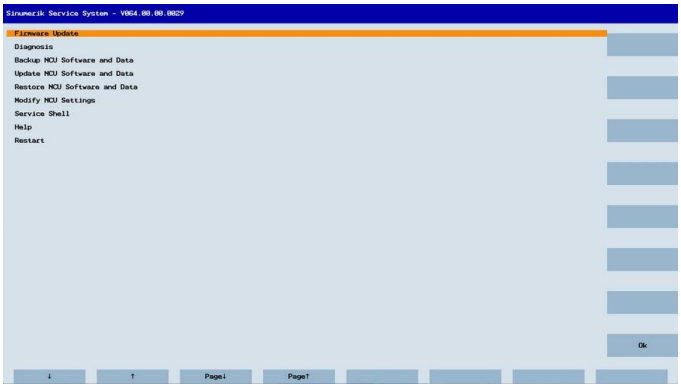


By powering up the boot stick in the NCU, the correct partitions are created .
Power OFF the NCU box.

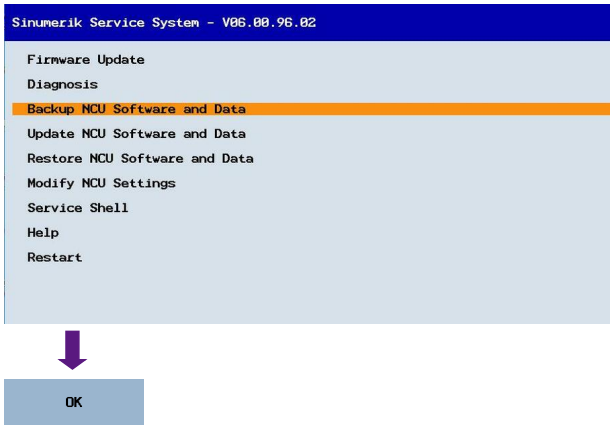
Creating a SD card backup (TGZ file)

Starting the Emergency Boot System.

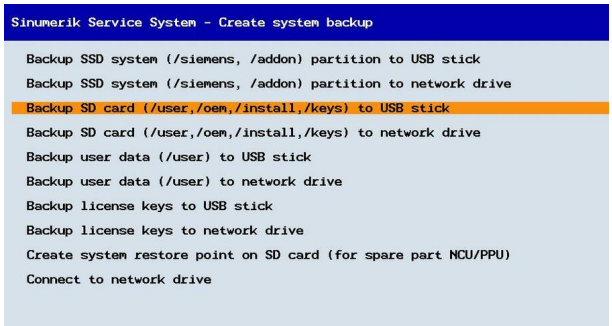
Switch off the controller and insert the bootable USB stick into X125 of the NCU, and place the NC rotary switch in any unused rotary position eg. F.
The following screen will be displayed:

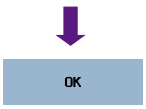


Cursor to the “Backup NCU Software and Data” menu item and select “Ok”

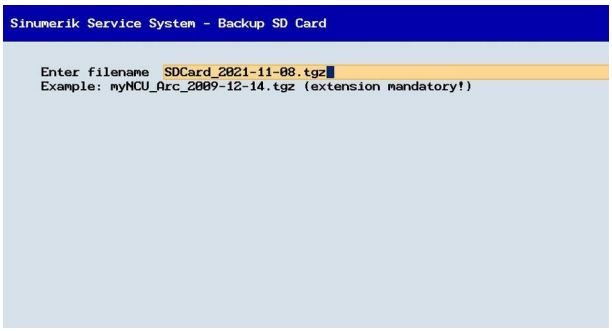


A number of backup choices are available, select the menu item “Backup Complete SD card to USB memory stick”





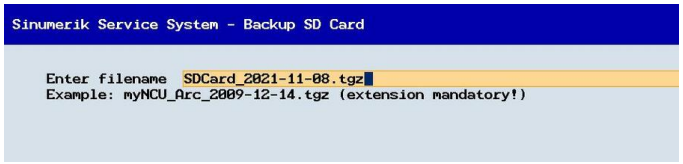
A suitable filename should be entered. This should contain information that helps identify the file to other users. The TGZ extension is mandatory.

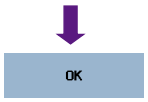


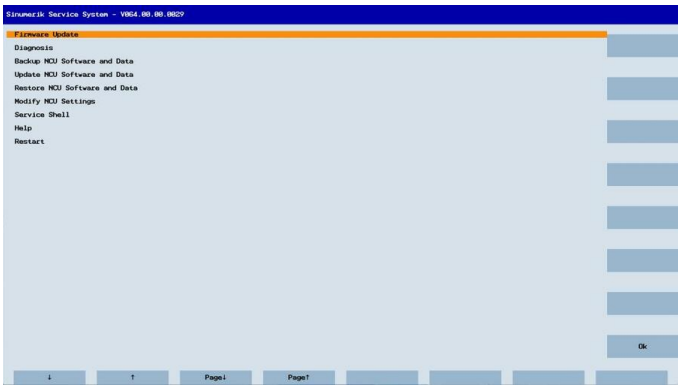


Notes:

The backup procedure will begin.



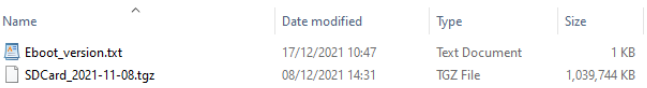




On completion, press any soft-key.



Check that the file exists on the USB stick, the file should be stored in a safe place, and be accessible to service personnel.



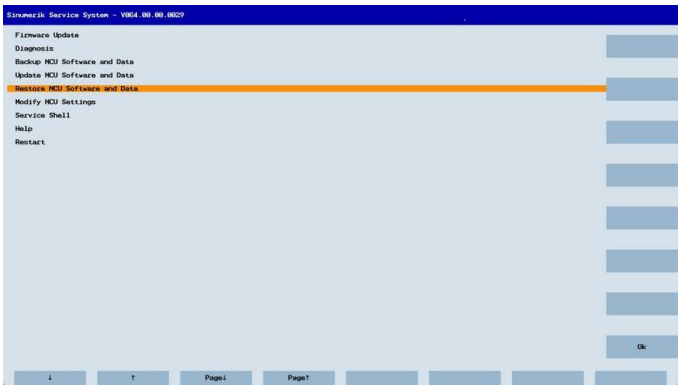
The original screen will be displayed.
The USB stick should be removed and the system rebooted.

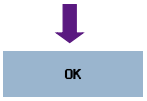
Restoring a SD card backup (TGZ)

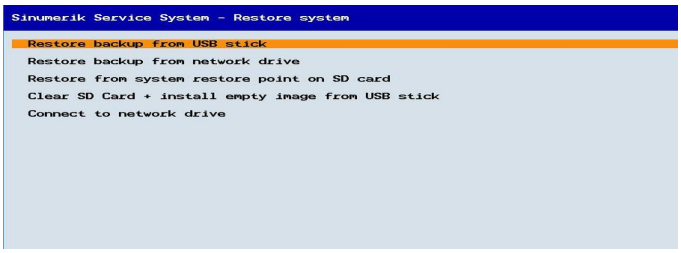
The restoring of the TGZ file is also achieved from the Emergency Boot System menu.

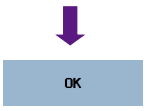
The backup TGZ file needs to be present on the USB Service stick.

Insert the USB service stick
Select “Restore NCU Software and Data”



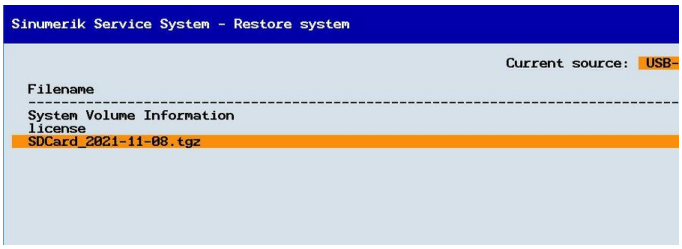


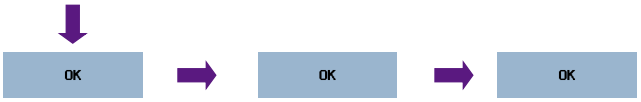






Notes:





The status of the procedure is displayed, allow this to continue.



On completion, press any of the soft-keys.





Switch off the controller and remove the bootable USB stick from X125 of the NCU, and place the NC rotary switch into position 0, switch on the NCU.

Notes:

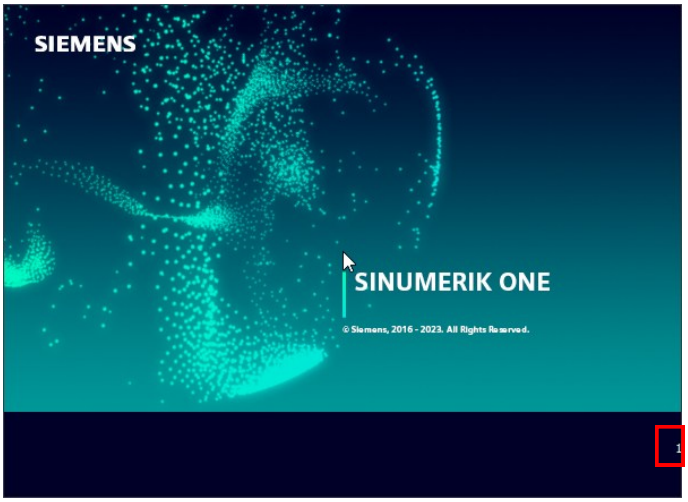
IPC WIN 10: Paragon Images

The IPC operates with a Solid State Disk (SSD) which contains the operating system (Windows®), Sinumerik basic software, SINUMERIK ONE software, archives, user files etc.
The (SSD) is divided into two partitions
C: SYSTEM (incl. EMERGENCY)
D: USER (incl. TMP)

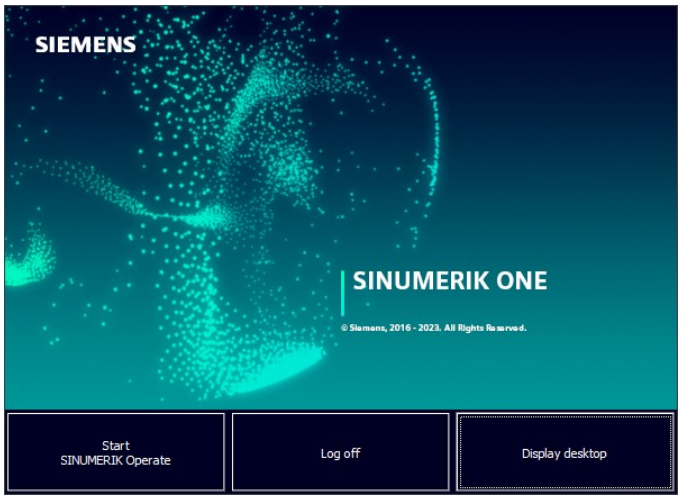
The method used to back-up the SSD drive is Paragon Hard Disk Suit®.
Paragon Hard Disk Suit is present on all replacement SSD drives to facilitate creating and restoring a Paragon image.
The image can be saved externally to a USB device, provided that there is sufficient space or via network.
It is also possible to create local Paragon images.
Prior to creating a Paragon image it is recommended that the archiving procedure is followed and that the files are saved to the IPC.

Partitions
C: SYSTEM (incl. EMERGENCY)

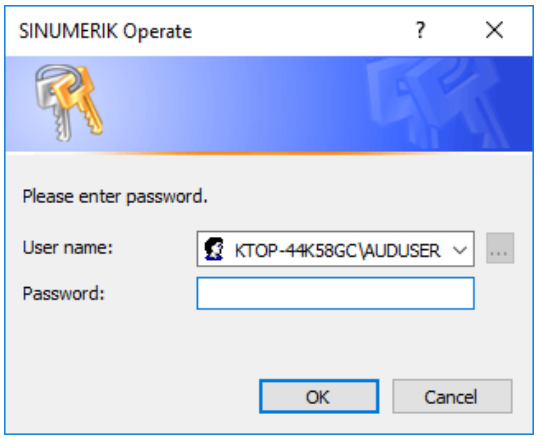
D: USER (incl. TMP)



Press **"3"** as the IPC boots up and the screen shows a count down from 5 to 0.

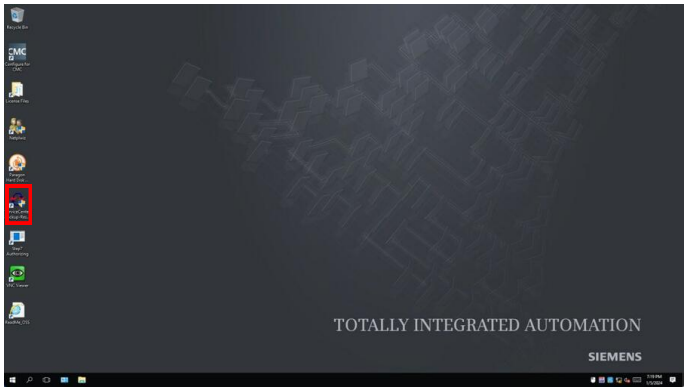


Select "Display descktop"



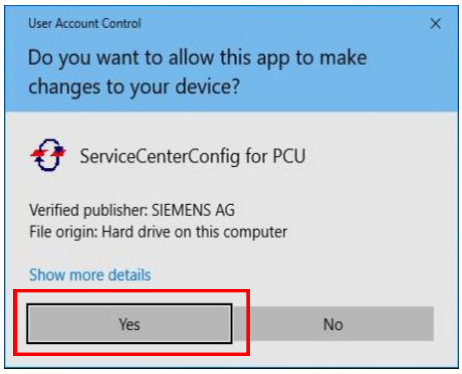
Enter the password.

Notes:

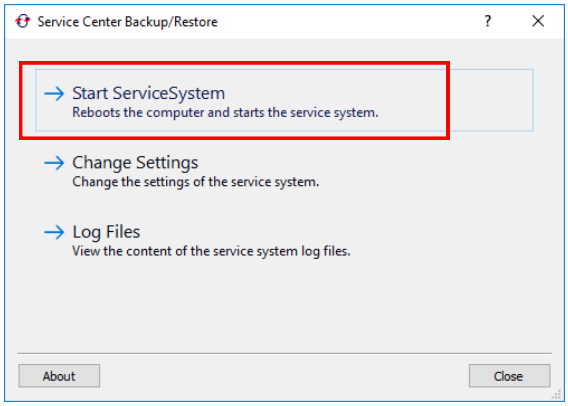


Double click, the “ServiceCenter Backup - Restore”



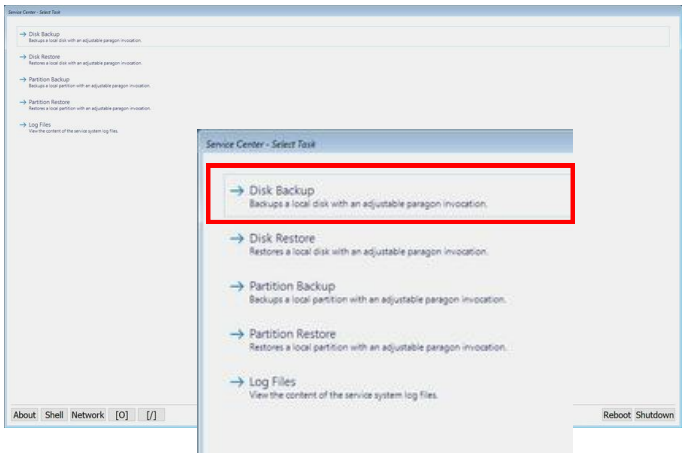








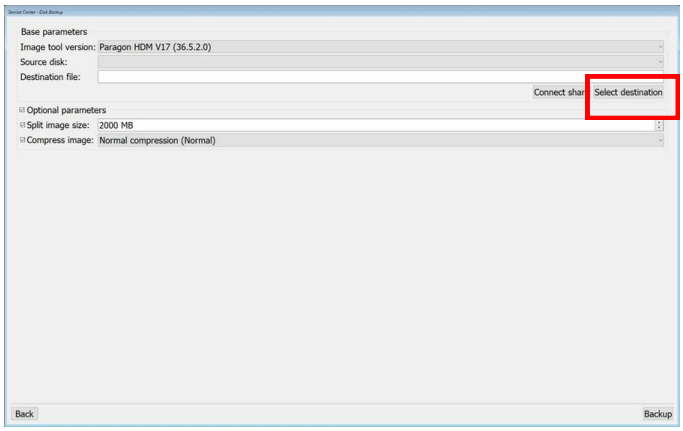
The ServiceCenter will now be started. This runs from Ram disk and takes a short while to load.
First insert the USB (At least 16GB) device into the USB interface on the IPC.



Select disk backup with mouse or INPUT

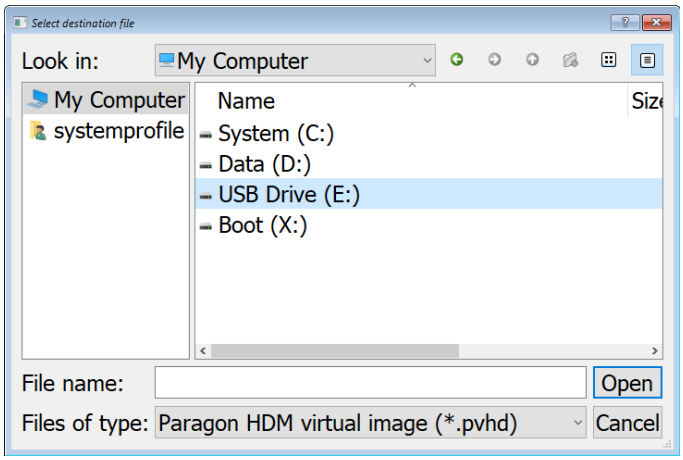
Notes:

Select “Select destination”.



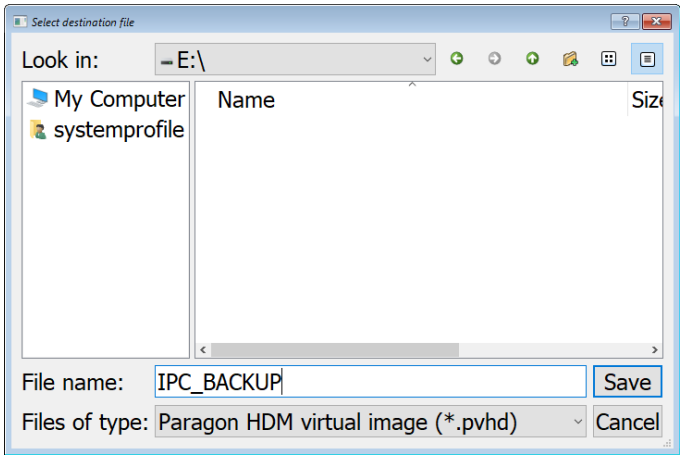
Select destination

Select drive E: (USB memory stick).



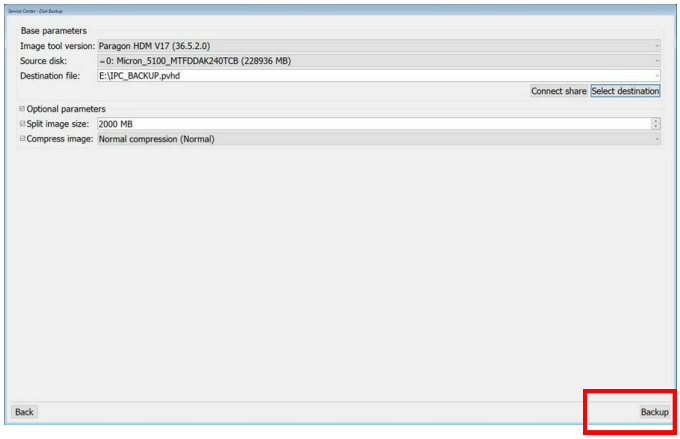
Open

Enter a suitable name for the paragon image. The **.pvhd** extension is added automatically.



Save

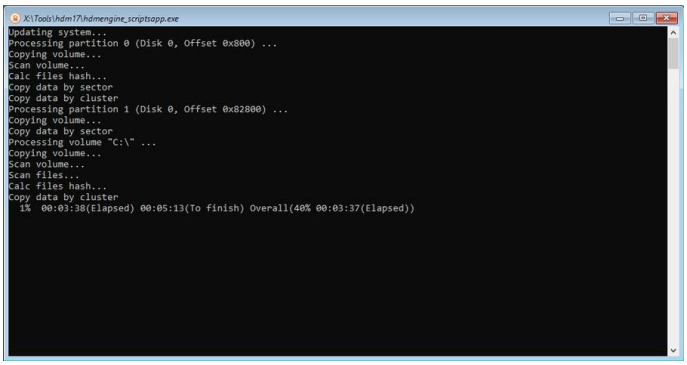
Select Backup to initiate the backup procedure.

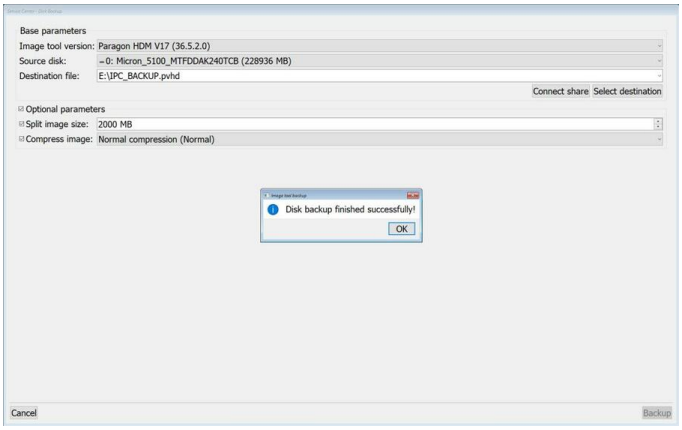


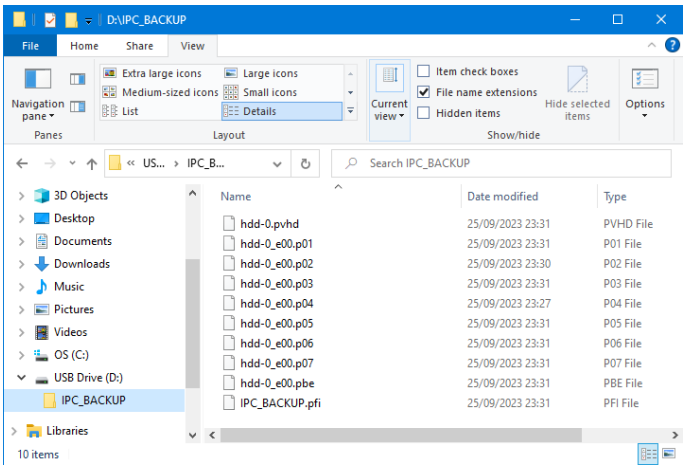
Backup

Notes:

The Paragon software will be started and the backup procedure will begin. Useful information can be viewed via Progress Indicator/Statistics/Details areas.







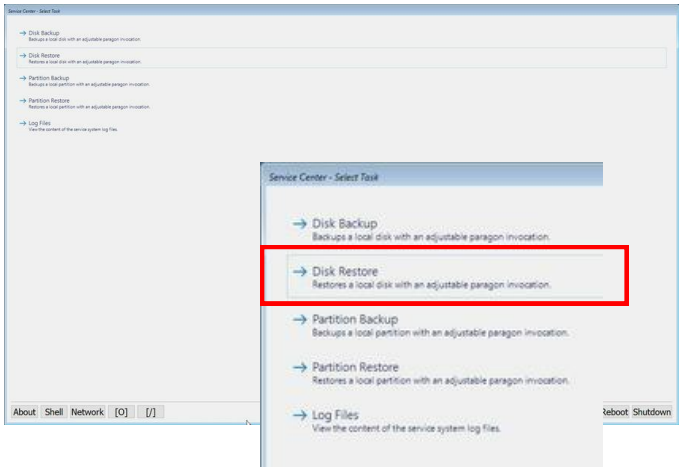
Windows explorer can be used to view the Paragon image file and associated partitions. Example - IPC_BACKUP, hdd-0_e00.p01 etc.

The image/partition size is determined by the Split image size parameter which is set to 2000MB by default.

Notes:

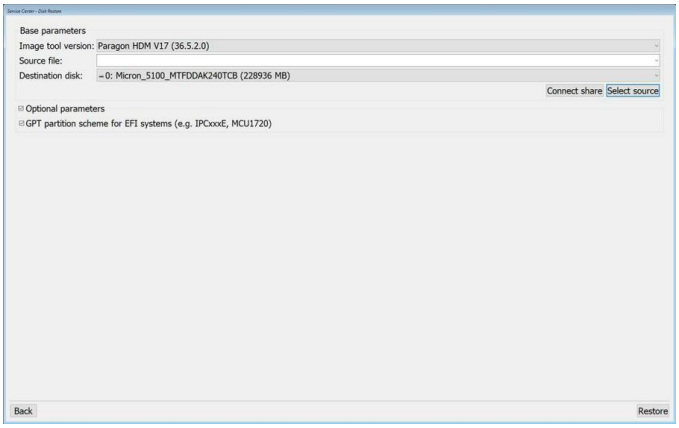
Restoring Paragon Images

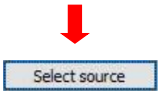
Images are also restored using the Service Menu.



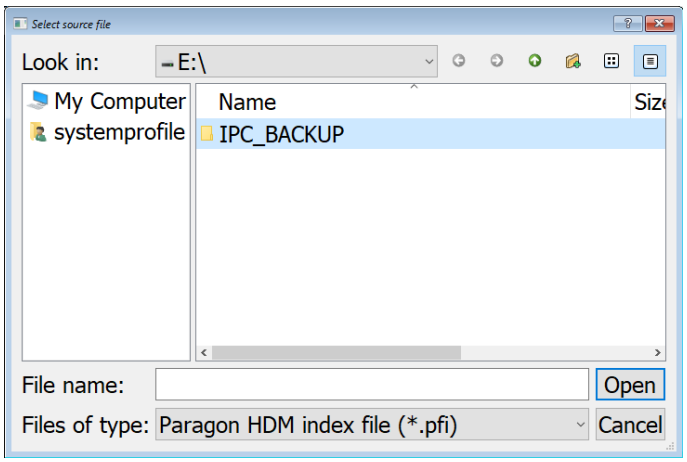
Select “Disk Restore” with mouse or INPUT

With “Select source“, select a paragon file.



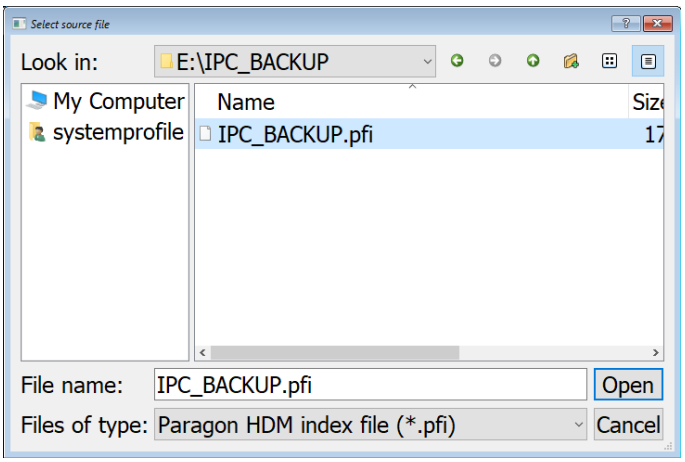


Select drive E: (USB memory stick):





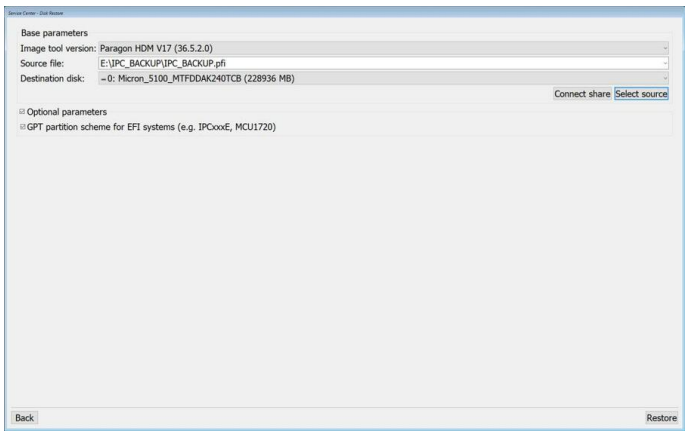
Select the Paragon file (*.pfi):





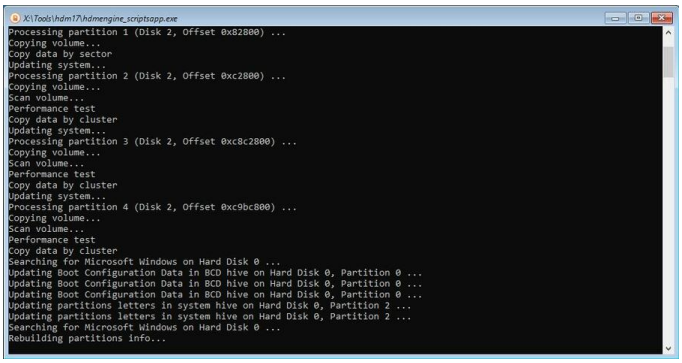
Notes:

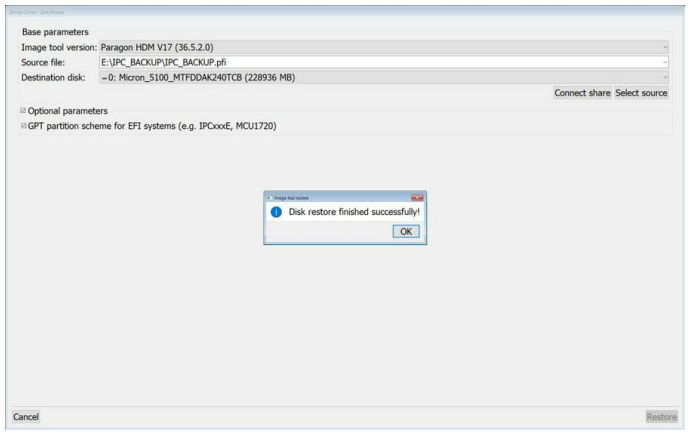
Select Restore to initiate the restore procedure.



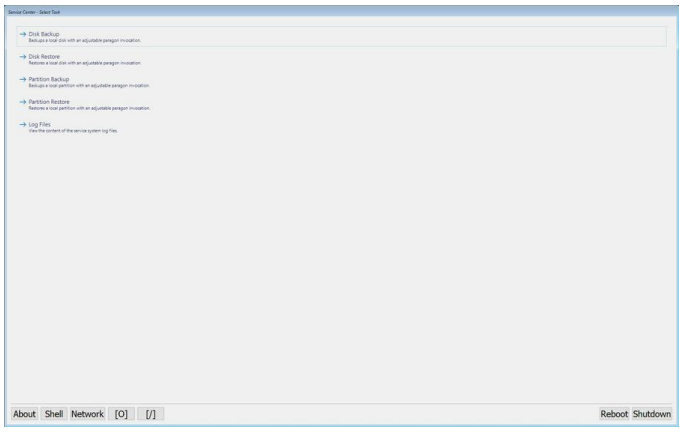
Restore

The Paragon software will be started and the restore procedure will begin. Useful information can be viewed via Progress Indicator/Statistics/Details areas.





OK



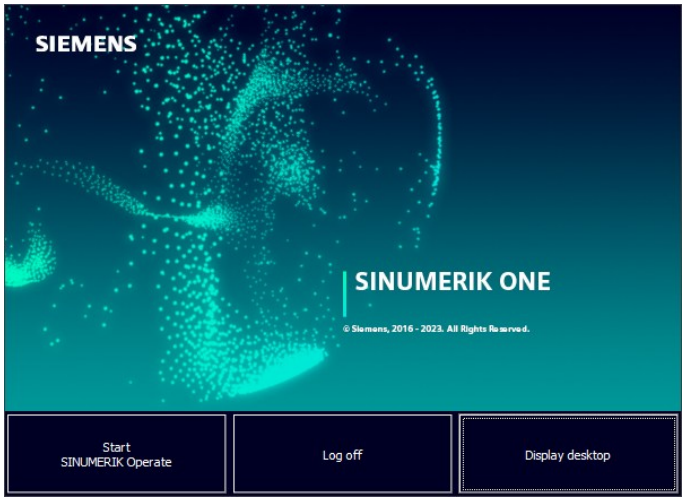
Reboot

Notes:

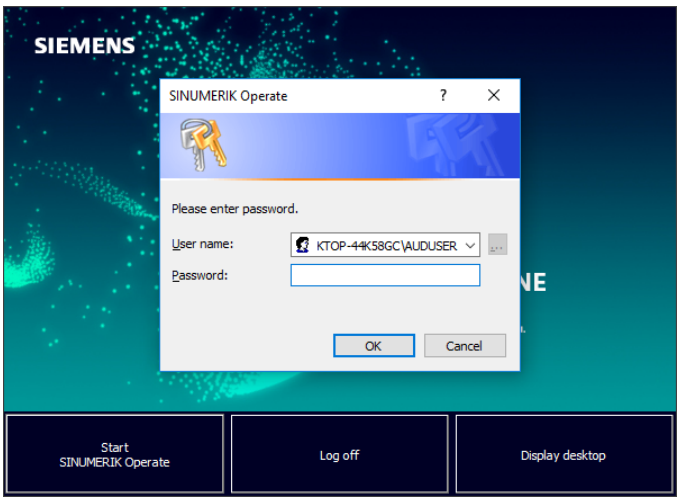
Creating a USB service stick for IPC WIN 10 Paragon

When windows OS gets corrupt and will not boot up, an EBOOT USB stick can be used to load a paragon image of the SSD, and restore this image without the need of re-installing all the required software to get the IPC running.

Press **"3"** as the IPC boots up when the screen shows a count down from 5 to 0.

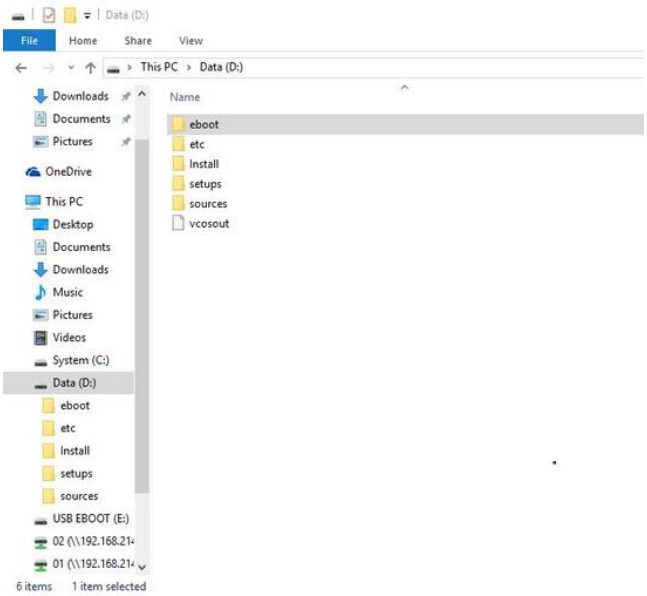


Select "Display descktop"

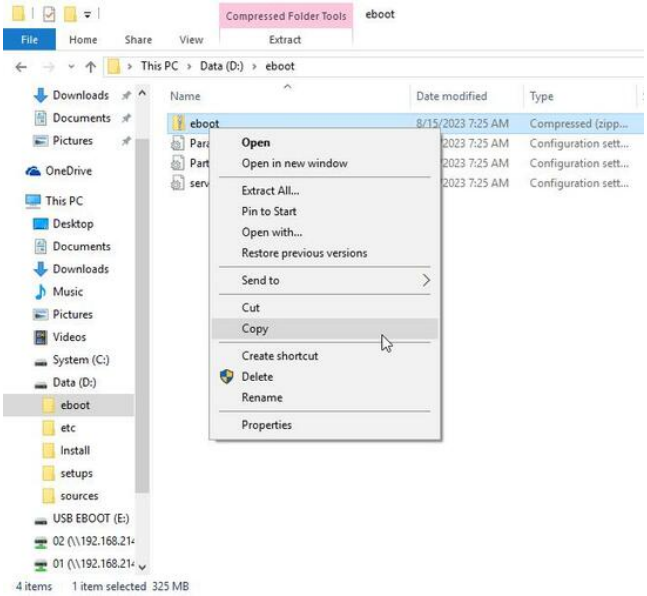


Enter the password.

Using Windows explorer.
Navigate to the D:/ drive and open "eboot" directory.

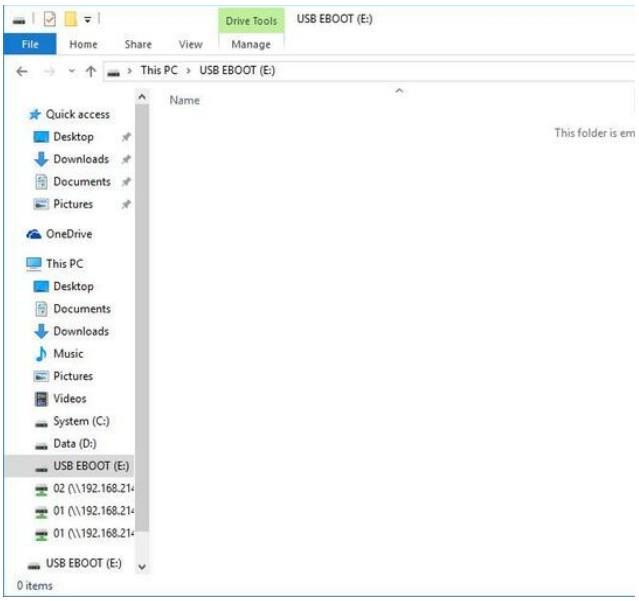


Copy the ZIP file.

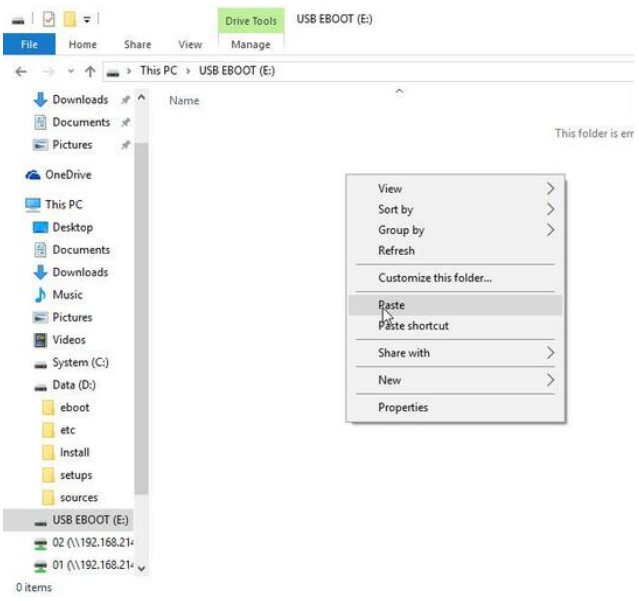


Notes:

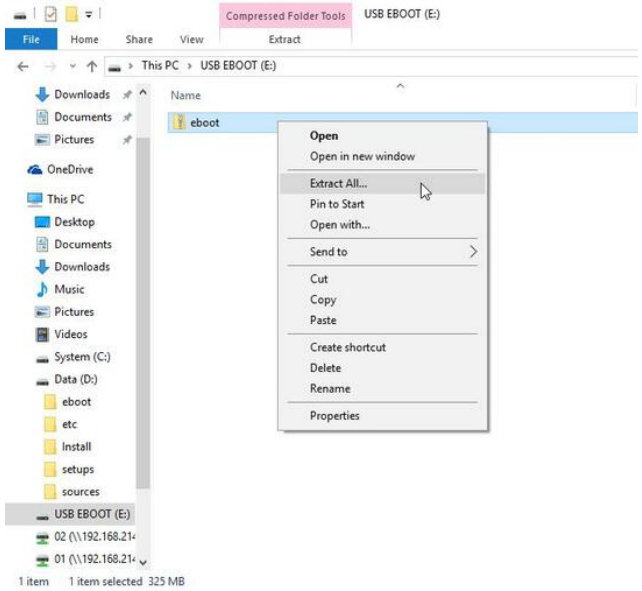
Place an empty USB memory stick into the USB port on the IPC.



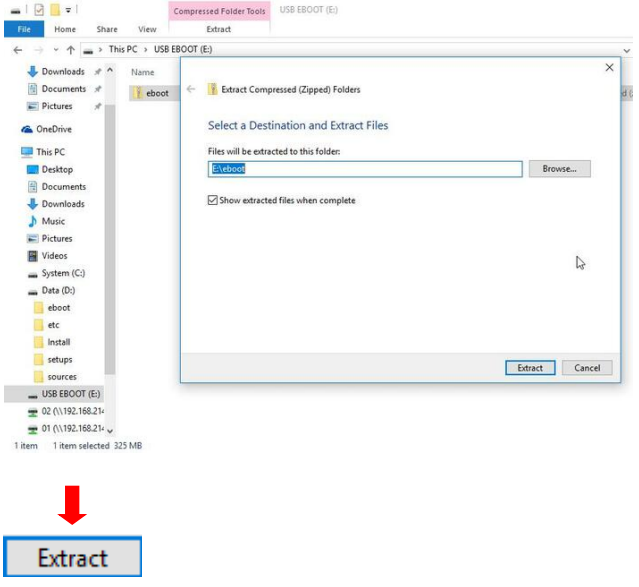
Paste ZIP file into the USB memory stick.



"Extract All"

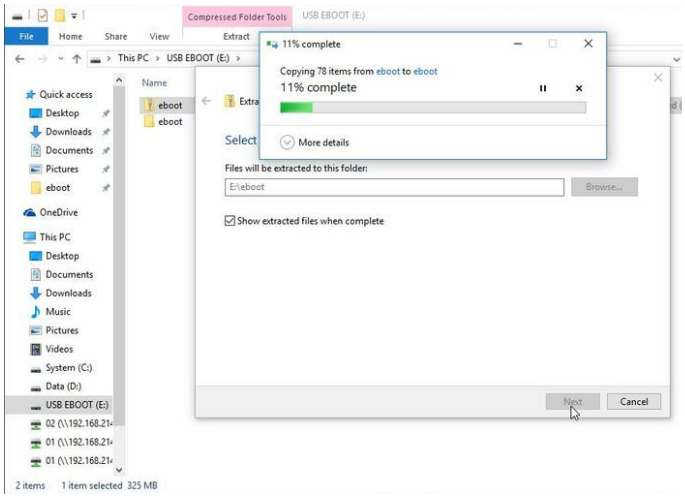


Confirm with "Extract".

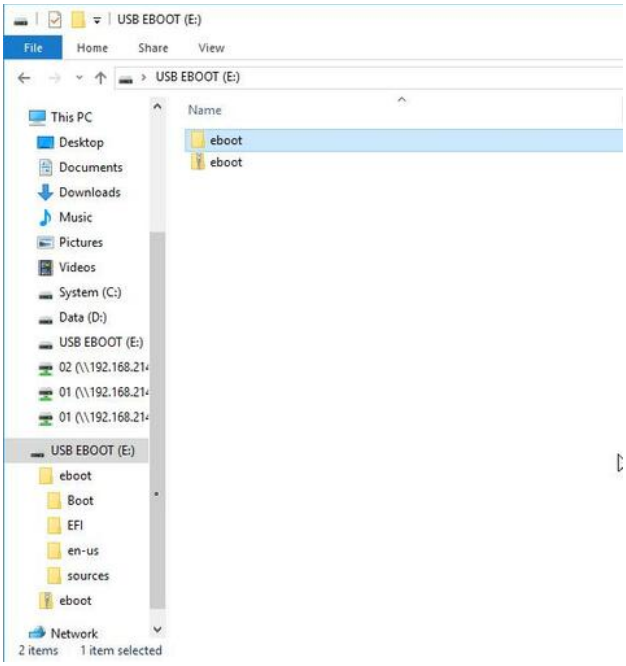


Notes:

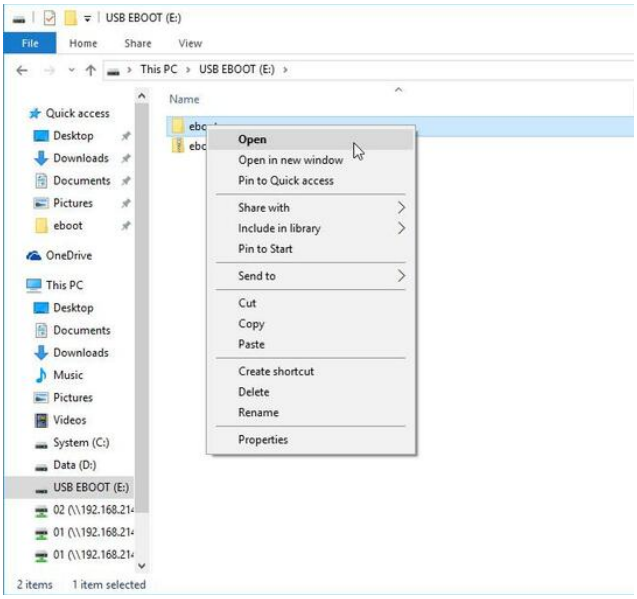
Allow the zip file to unpack.



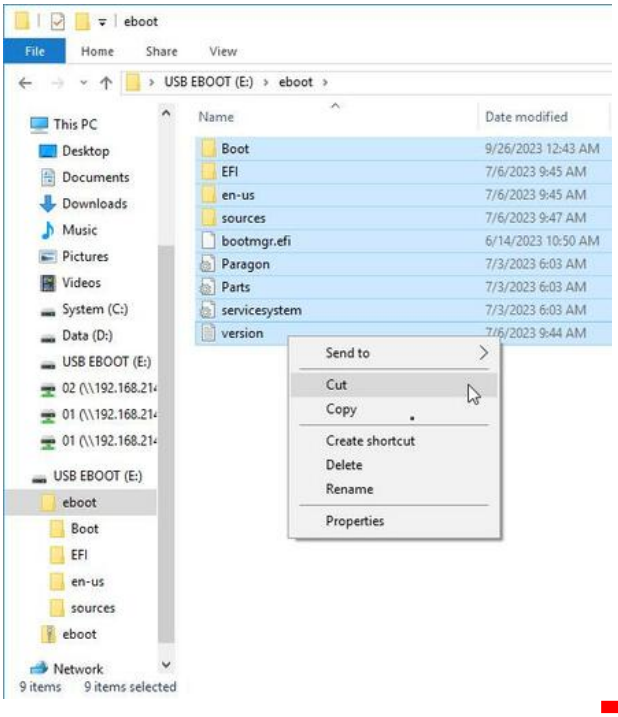
When this is completed there will be a new directory call “eboot”.



Open the new directory call “eboot”.

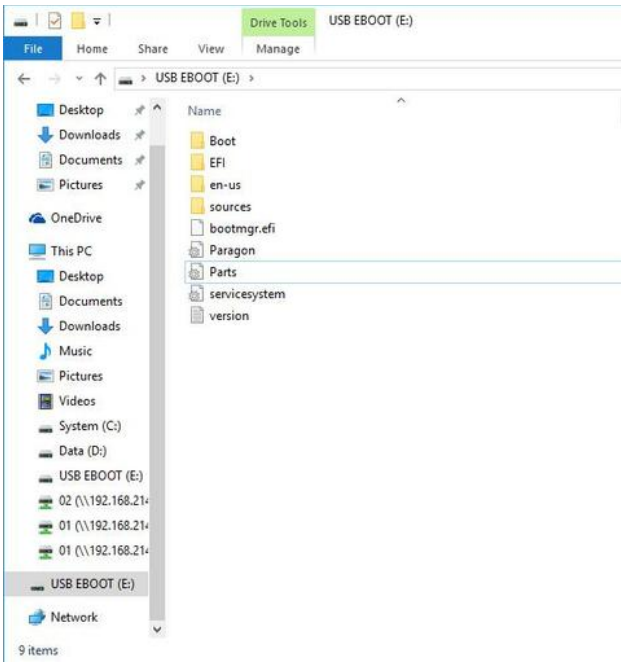


Highlight the contents of “eboot” and “Cut”.

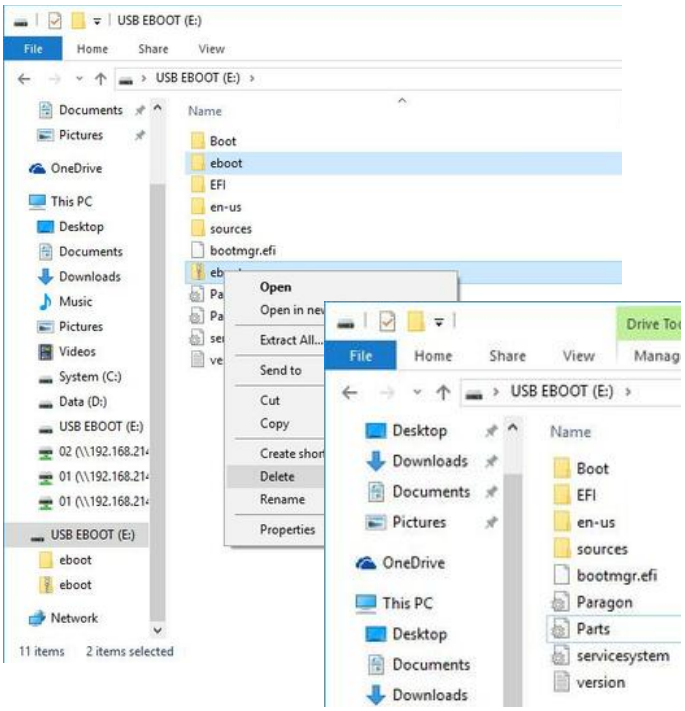


Notes:

And “paste” into the root of the USB memory stick.



Highlight eboot directory and eboot.zip and delete them.

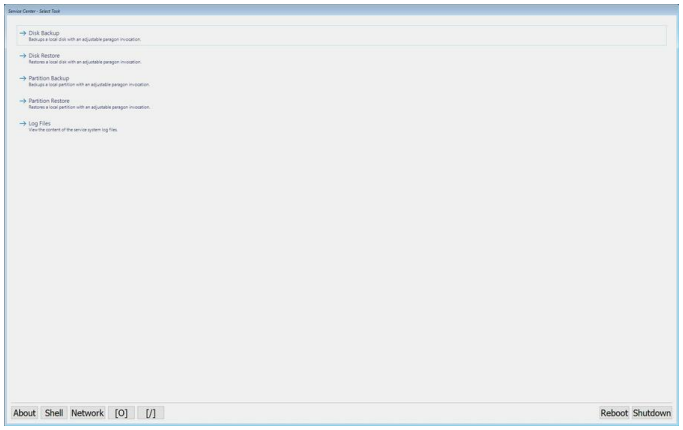


To test/use, place USB EBOOT stick into the IPC USB socket.

Connect an external monitor, keyboard, and mouse, to the IPC.

On powerup select “Esc” when the “Siemens” logo appears. Use the boot manager to boot from the USB boot stick.

The “ServiceCenter” will now be running from the USB EBOOT stick, the same menu’s will be available, to backup and restore paragon images.



Note:

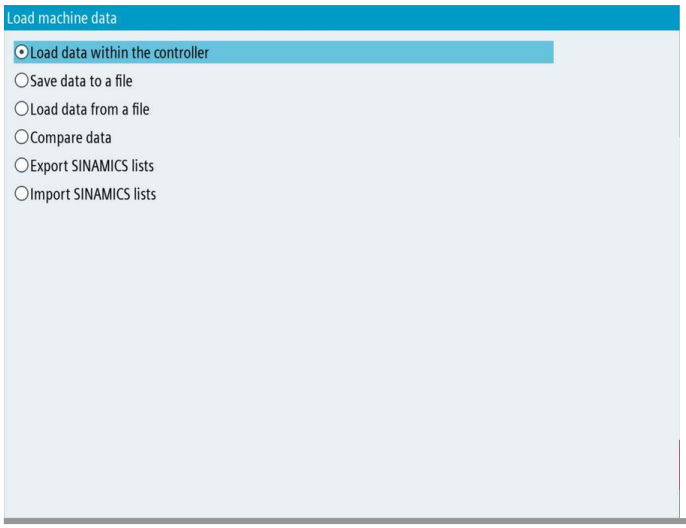
An EBOOT USB stick can only be created on a running IPC system.

Notes:

Data admin

The "**Data admin**" function simplifies the commissioning and permits the transfer, backup, loading and comparing of machine, setting, compensation and drive data at the file level. The files are saved in the .tea ASCII format.
The "Data admin" softkey is visible as of access level "Service" (= 2).
To find the "Machine data loaded" area:





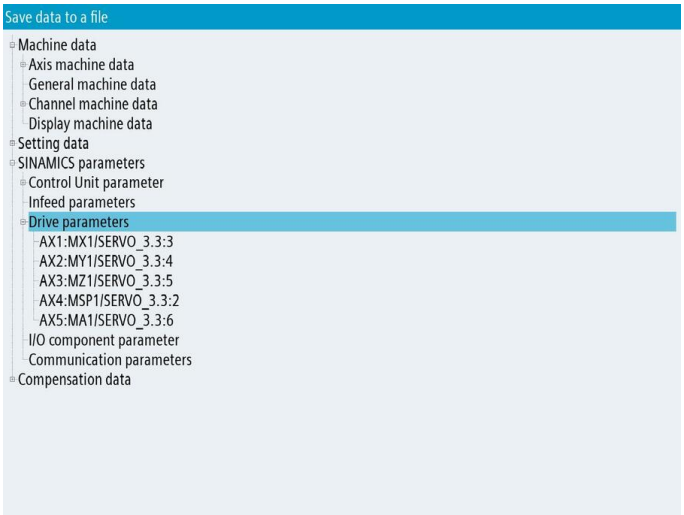
Tasks that can be carried out:

- **Load data within the controller**
  Only those controller objects are listed that are currently available in the controller.
- **Save data to a file**
  Additional files with the extensions .acx and .log are created for the SINAMICS parameters:
  - Files with the extension .acx contain the parameter descriptions for the selected drive object.
  - The log file may also contain the warnings and errors that occurred during the conversion from .acx to .tea.
- **Load data from a file**
- **Compare data**
  Several files (.tea) and/or controller objects (axis, channel, servo, infeed, etc.) can be compared at the same time. An online comparison of constantly changing data is not possible because the values are saved temporarily at the time of the comparison. Up to eight elements can be taken into the list of the data to be compared.
- **Export SINAMICS lists**
- **Import SINAMICS lists**

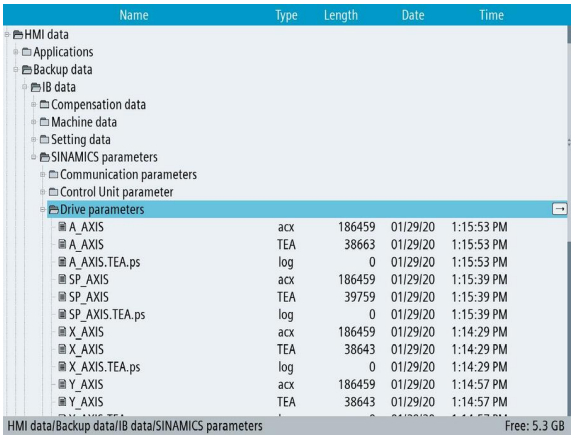


Save all DOs (drive parameters) individually in a file!

The drive data **"DOs"** saved in this way can then be used for new topologies!



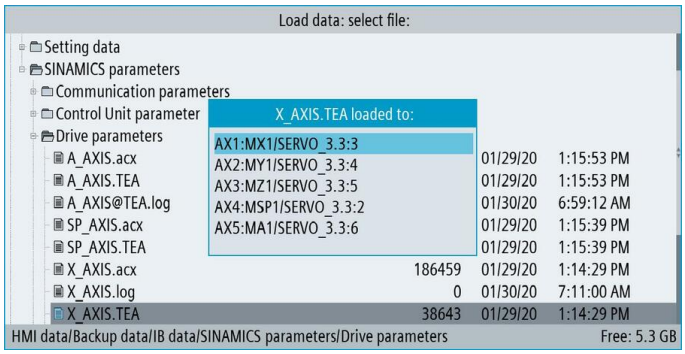
The saved data is in the following directory.



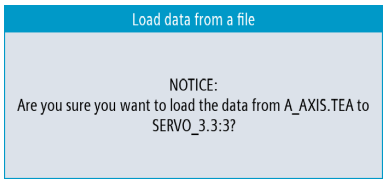
Use drive data modularly:

Optimized drive data "DOs" can be loaded to a new topology. Prerequisite is that the drive objects comprising "power unit, motor, encoder interface and encoder" are identical!
With the **"Load data from a file"** function, select a drive file with the extension **.TEA** (source) and the servo (target). Confirm the selection with the **"OK"** softkey.

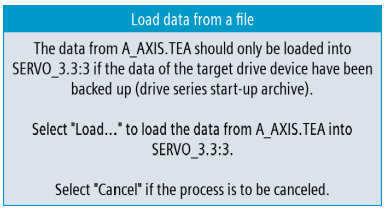
Notes:



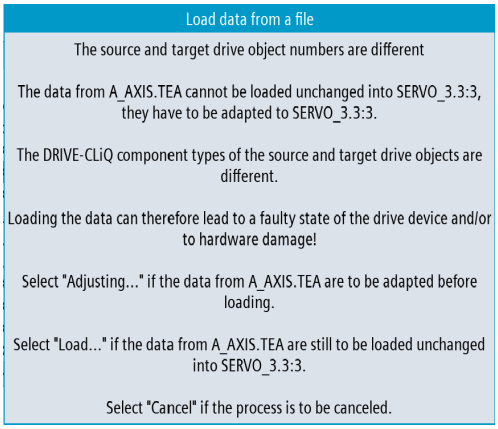




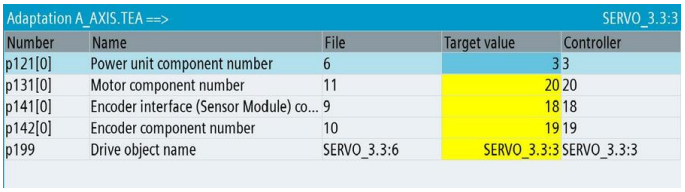




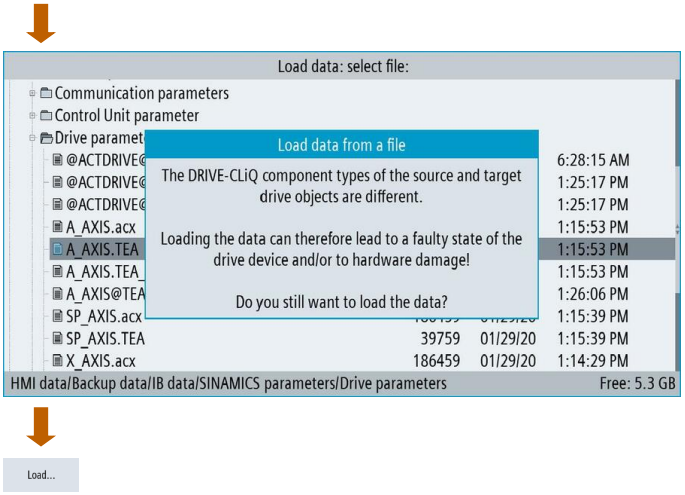






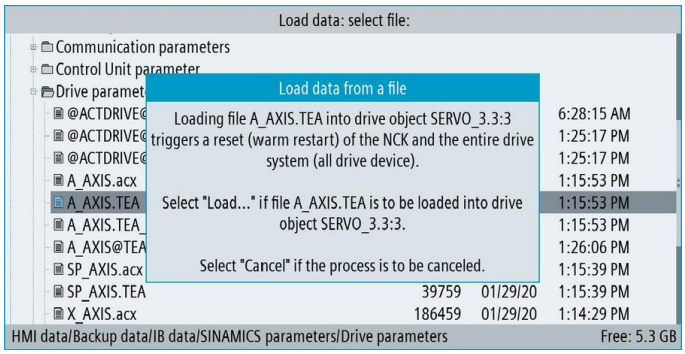




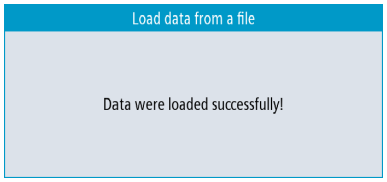


The source DOs must be loaded via the **"Adjusting..."** soft-key function!
Caution: Never direct **Load** "Actual topology will be over-written"!

Notes:





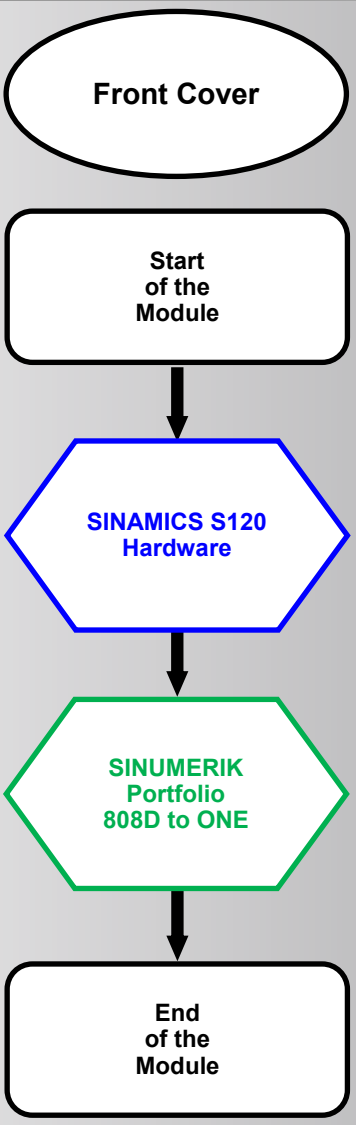




A137: END

Notes:

Sinamics Hardware Information



Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

Sinumerik ONE: Sinamics Components

SINUMERIK and SINAMICS automation system components.

Line Modules
Smart Line Modules
Active Line Modules
Basic Line Modules



Line-side components
Line chokes
line filter
Active Interface Modules



Motor Modules
Single Motor Modules
Double Motor Modules
Booksize Compact



Chassis Modules
AIM / ALM / MM



SINAMICS S120
S120M
HLA



SINAMICS S120 Combi



DC Link Components
Braking resistors
Braking Module
Capacitor Module
Control Supply Module



Control Units
NX10.3 / NX15.3
CU320-2
CU310-2



Supplementary Components
DMC20 / DME20
TM31
TM41
TM15
TM120
VSM10

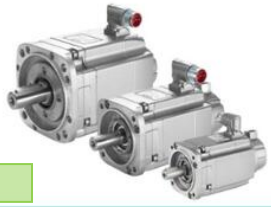


Sensor Modules
SME20 /25/120/125
SMC10 /20/30/40



Sinumerik ONE: Simotics Motors

SIMOTICS S servo motors
SIMOTICS S-1FT7
SIMOTICS S-1FK7
SIMOTICS S-1FK2 (2 cable connections)
SIMOTICS S-1FG1



0.18 – 1,650 Nm

SIMOTICS M main motors
SIMOTICS M-1PH8
SIMOTICS M-1FE1
SIMOTICS M-1FE2
SIMOTICS M-1PH2
SIMOTICS M-1PH3



13.0 – 12,435 Nm

SIMOTICS L linear motors
SIMOTICS L-1FN3



Up to 20,700 Nm

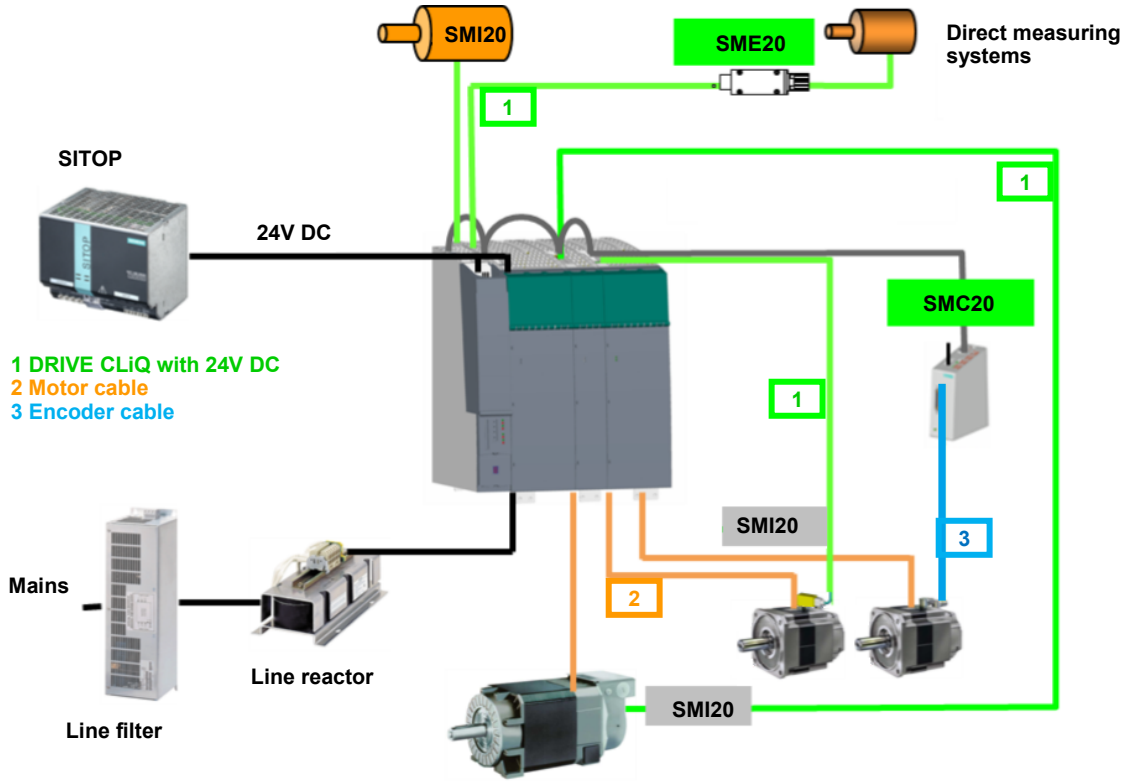
SIMOTICS T torque motors
SIMOTICS T-1FW6
SIMOTICS T-1FW6 HS



34.4 – 10,900 Nm

Notes:

SINAMICS: Component Connection Diagram



SINAMICS: Cabling Information

Quicker and easier Start-up with SMI
Automatic detection of motors and encoders
Storage of technical and logistics data
Storage of the equivalent circuit diagram and encoder data in the motor

Integration of direct position measurement systems
Use of measuring systems with DRIVE-CLiQ interface
Use of SMC/SME modules for signal conversion to DRIVE-CLiQ

Third-Party Encoders / Motors
Integration via SMC/SME modules
Parameter assignment via start-up screens

Motion Connect 500 mainly for fix wiring.

Motion Connect 500 plus for low mechanical load
.

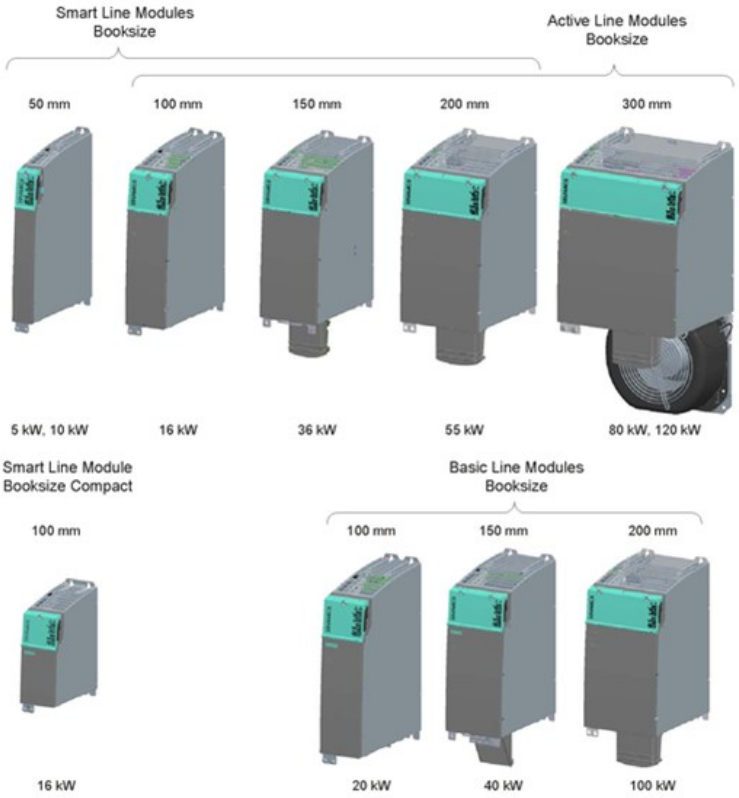
Motion Connect 700 for highly dynamic machines (e.g. Linear drives).

Motion Connect 800 or the high mechanical needs in drag chains.

Degree of Protection: IP20 (Cabinet) up to IP68 for external components

Notes:

S120 Line Modules & DC link components



Smart Line Modules (SLM)

- Unregulated infeed.
- Regenerative feedback (line-commutated feed back via IGBTs)
- Always connected to the DRIVE-CLiQ (Except for 5 & 10kW variants) for control from the Control unit (CU)

Active Line Modules (ALM)

- Self-commutated infeed.
- Regenerative feedback unit with bidirectional IGBTs.
- Reduced line harmonics with high drive dynamics. Ideal for use with weak mains systems.
- Improved utilization of motors via higher regulated DC link voltage.
- Always connected to the DRIVE-CLiQ for control from the Control unit (CU)

Basic Line Modules

- Unregulated DC Link Voltage
- No regenerative feedback
- With all BLM: DC-link rails to both sides of line module.
- Always connected to the DRIVE-CLiQ for control from the Control unit (CU).

Active Interface Modules for 16 kW, 36 kW, 55 kW und 80 W/120 kW



Active Interface Modules (AIM)

Active Interface Modules are connected to the 3 phase drive supply and are designed to work with the Active Line Modules.

They contain the following components:

- Line reactor
- Low-frequency/switching frequency filters
- Line filters to EN 61800-3, category C3

Line Reactor for Line Modules

A line reactor is always required.

- On Active Line Modules, it is essential for the operation of the step-up converter.
- On Smart Line Modules it is mandatory to reduce the line harmonics and to protect against high overshoot currents.

Networks All Line Modules are designed for connection to grounded, star (TN, TT) and non-grounded, symmetrical IT supply systems.

BLM The 100kW module hasn‘t got the integrated braking resistor module control.

Notes:

General Infeed Properties

- Supply voltage 3AC 380-480V, 50/60Hz
- IP20 protection
- Cooling: Internal and external air cooling
- Ambient temperature: 0-40 °C without derating, to 55°C with derating
- Height: 380mm
- Depth: 270mm

Line Module Power Rating and Width Information:

Smart line modules: Supply power (kW) / width (mm)
- 5/50 (No Drive-CLiQ ports)
- 10/50 (No Drive-CLiQ ports)
- 16/100
- 36/150
- 50/200

Active line modules: Supply power (kW) / width (mm)
- 16/100
- 36/150
- 55/200
- 80/300
- 120/300

ALM Control Modes

The active line module can be operated in one of two different control modes, depending on the parameterised line supply voltage (p210):

Active Mode DC link voltage is controlled with respect to a settable reference value. The Line Module acts as a Step-up converter with regulated DC link voltage and sinusoidal supply voltage.

Smart mode The DC link voltage is not regulated. It results from the rectified supply voltage.

ALM Control Modes

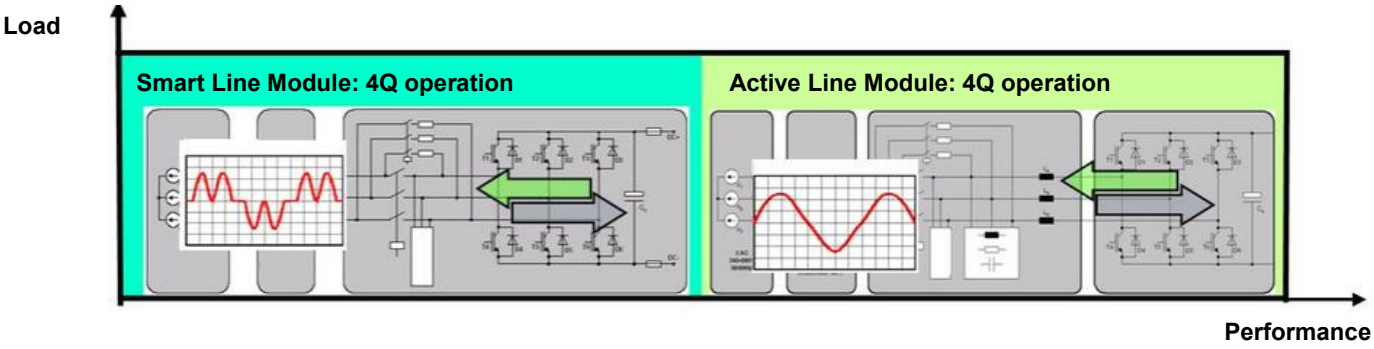
The control mode - active or smart - depends on the supply voltage to the ALM.
Active mode is operational in the supply range 380V to 415V and the DC link is stabilised.
When the supply voltage is above 415V smart mode is active.
The DC link is not stabilised and is the result of 1.35 x the supply voltage.

Mode	DC link voltage	Feedback
Active mode	Regulated	Active
Smart mode	Rectified supply voltage	Active

Supply Voltage (drive parameter p210)	380-240	401-415	416-440	460	480
Control mode (infeed parameter p3400)	Active Mode		Smart Mode (1.35 x supply voltage)		
Vdc_sol (infeed parameter p3510)	600	625	562-594	621	648

Notes:

SINAMICS Power Components: Line Modules



Properties	Smart Line Module	Active Line Module
Operating mode	Unregulated	Regulated Sinusoidal mains infeed
Electronics power	IGBT	IGBT
Mains supply fluctuations	Yes	No
Regenerative feedback	Yes	Yes
Harmonics	High	Low
Power factor compensation	NO	Yes

SINAMICS: DC Link Components.

Braking Module and braking resistors

Two braking resistors are available:
0.3 kW (Booksize) continuous braking power (25 kW peak power)
1.5 kW continuous braking power (100 kW peak power)

Conversion of the DC link energy into thermal energy via the external braking resistors.
Braking modules and braking resistors are not usually required for normal operation due to the Smart and Active Line Modules being capable of regenerative feedback.
Braking components must only be provided if a machine must be stopped in a controlled way at power failure (feedback operation no longer possible).



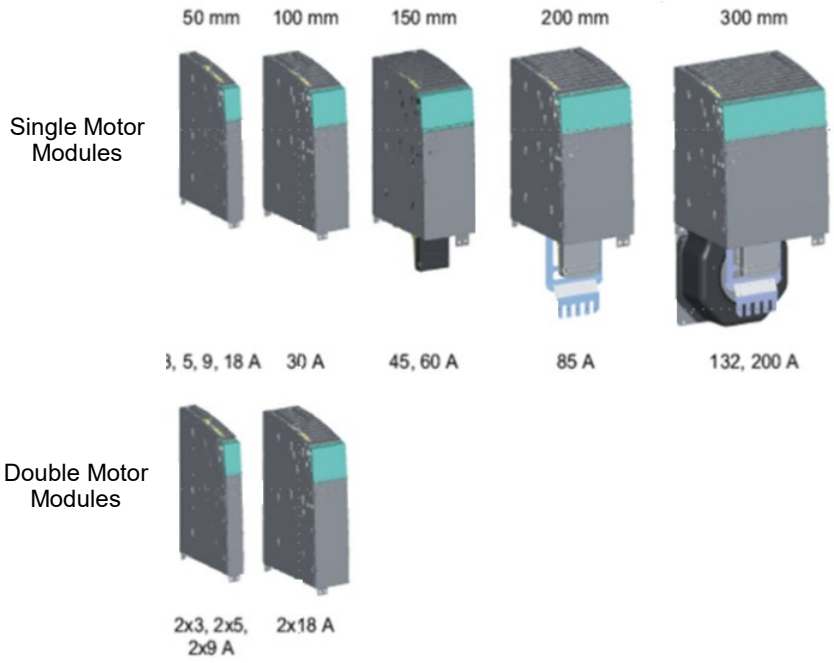
Capacitor modules
Is used for increasing the DC link capacity in order to deal with short power failures

Control Supply Modules
Supplies DC 24V power via the mains or via the DC link.
This way it is possible, for example, to perform emergency retraction motions in a controlled way



Notes:

S120 Booksize Motor Modules



Booksize Format Information

- Compact design achieved by high power density, particularly with the double motor modules.
- Integrated DC link and 24V Dc bus bars.
- Diagnostic LEDs
- Short circuit and earth fault protection.
- Integrated safety functions - Safe Brake Control (SBC) and Safe Torque Off (STO)
- Integrated motor brake connection (up to and including 200A modules) and braking by armature short circuiting.

Features - Booksize Format

- With the double motor modules two autonomous motor modules are integrated in one housing (space savings up to 50%)
- Connection via the DC link and DC 24V bus bars integrated in the system.
- Safety Integrated functions such as **Safe Operational Stop (SOS)** and **Safe Brake Control (SBC).**
- Overload capacity of the motor modules similar to SIMODRIVE 611.
- The specified depth of 270mm only applies to devices with air cooling. Models with forced Convection cooling and cold plate cooling have a reduced depth.
- All components in the SINAMICS S system are **cULus** approved. The cULus mark is a test mark of Underwriters Laboratories (UL) and indicates compliance with **UL** and **CSA** (Canadian Standard Association) standards.

Properties of the Motor Modules

- Supply voltage DC 510-720V
- IP20 protection
- Cooling: Internal / external air cooling or cold plate cooling
- Ambient temperature: 0-40 °C without derating, up to 55°C with derating
- Height: 380mm
- Depth: 270mm

Single Motor Modules

Rated output current: from 3A to 200A
Maximum output current: from 6A to 282A
Width: from 50mm to 300mm

Double Motor Modules

Rated output current: from 3A to 18A
Maximum output current: from 6A to 36A
Width: from 50mm to 100mm

Notes:

Sensor Modules

General Information

Sensor modules can be used if motors without a DRIVE-CLiQ interface are used (non Siemens motors, for example) or, if additional to the motor encoder, further encoders are required e.g. linear scales.

SMC (Sensor Module - Cabinet Mounted) SMC10/SMC20/SMC30/SMC40

The SMC is designed to be mounted inside electrical enclosures.
SMCs evaluate the encoder data which is then available to the drive system via the DRIVECLiQ drive bus.
The SMC supplies the power to the encoder; the SMC, however, must be provided with a 24 VDC power source.

SMC10/20



Product name	SMC10 Sensor Module Cabinet-Mounted
Current requirement, max. At 24 V DC, not taking encoder into account	0.2 A
• Conductor cross-section, max.	2.5 mm^2
• Fuse protection, max.	20 A
Power loss, max.	10 W
Encoders which can be evaluated	• 2-pole resolver • Multipole resolver
• Excitation voltage, rms	4.1 V
• Excitation frequency	5 ... 10 kHz depending on the current controller clock cycle of the Motor Module or Power Module
• Transformation ratio	0.5
• Encoder frequency, max.	2 kHz (120000 rpm) depending on the number of resolver pole pairs and current controller clock cycle of the Motor Module or Power Module
• Signal subdivision (interpolation), max.	16384 times (14 bit)
• Cable length to encoder, max.	130 m (427 ft)

Product name	SMC20 Sensor Module Cabinet-Mounted
Current requirement, max. At 24 V DC, not taking encoder into account	0.2 A
• Conductor cross-section, max.	2.5 mm^2
• Fuse protection, max.	20 A
Power loss, max.	10 W
Encoders which can be evaluated	• Incremental encoder sin/cos 1 V_{pp} • Absolute encoder EnDat • SSI encoder[1] with incremental signals sin/cos 1 V_{pp} (firmware version 2.4 and later)
• Encoder supply	5 V DC/0.35 A
• Encoder frequency incremental signals, max.	500 kHz
• Signal subdivision (interpolation), max.	16384 times (14 bit)
• SSI baud rate	100 kBaud
• Cable length to encoder, max.	100 m (328 ft)

SMC30



Product name	SMC30 Sensor Module Cabinet-Mounted
Current requirement, max. At 24 V DC, not taking encoder into account	0.2 A
• Conductor cross-section, max.	2.5 mm^2
• Fuse protection, max.	20 A
Power loss, max.	10 W
Encoders which can be evaluated	• Incremental encoder TTL/HTL • SSI encoder with TTL/HTL incremental signals • SSI encoder without incremental signals
• Encoder supply	24 V DC/0.35 A or 5 V DC/0.35 A
• Encoder frequency, max.	300 kHz
• SSI baud rate	100 ... 250 kBaud
• Limiting frequency	300 kHz
• Resolution absolute position SSI	30 bit
• Cable length, max.	
- TTL encoder	100 m (328 ft) (only bipolar signals permitted)[1]
- HTL encoder	100 m (328 ft) for unipolar signals 300 m (984 ft) for bipolar signals[1]
- SSI encoder	100 m (328 ft)

1) Signal cables must be in twisted pairs and shielded.

The SMC10/20/30 have the following interfaces:

- 1 DRIVE-CLiQ interface
- 1 encoder connection, including motor temperature detection (KTY84-130 or PTC) via Sub -D connector or terminals
- 1 connection for the electronics power supply via the 24 V DC power supply connector
- 1 PE/protective conductor connection

Notes:

SMC40

The **SMC40** is used to convert the signals from absolute encoders with EnDat to DRIVECLiQ.



Article No.	6SL3055-0AA00-5DA0
Product designation	SMC40 Sensor Module Cabinet-Mounted
Power requirement at 24 V DC, max. without considering encoder	0.1 A
• Conductor cross-section, max.	2.5 mm^2
• Fuse protection, max.	20 A
Power loss, max.	4 W
Encoders which can be evaluated	Absolute encoder EnDat 2.2
PE connection	M4 screw
Width	30 mm (1.18 in)
Height	150 mm (5.91 in)
Depth	111 mm (4.37 in)
Net weight	0.45 kg (0.99 lb)
Certificate of suitability	cULus

The **SMC40** have the following interfaces:

- 2 DRIVE-CLiQ interfaces
- 2 encoder system interfaces
- 1 connection for the 24 V DC electronics power supply.
- 1 PE connection

The maximum encoder cable length between SMC40 modules and encoder systems is 100 m. The specified supply voltage of the encoder must be observed.

The maximum cable length for DRIVE-CLiQ cables is 30 m.

Notes:

SME (Sensor Module - Externally Mounted) SME20/SME25/SME120/SME125

The SME is designed to be mounted externally to electrical enclosures.

The SME supplies the power to the encoder; the SME is provided with a 24 VDC power source from the drive system via the DRIVECLiQ cable.

SME20/25



Encoder Types

The following encoder signals can be evaluated:

- Incremental encoder sin/cos 1 Vpp without rotor position track C and D tracks)
- Absolute encoder EnDat 2.1
- SSI absolute encoder1) with incremental signals sin/cos 1 Vpp (firmware V2.4 and later)

It is possible to connect a motor with a 17-pole circular encoder connector to the 12-pole circular connector of the SME20 using adapter cable 6FX8002-2CA88-…

- KTY/PT1000/PTC temperature sensors can be used for motor temperature evaluation (only possible with SME20).
- The Sensor Module is only suitable for motors without absolute track signals (C and D tracks), e.g.: - Synchronous motors with pole position identification (1FN, 1FW)
  - Asynchronous motors (1PH)

SME20/SME25 Interfaces:

- 1 encoder connector via circular connector
- 1 DRIVE-CLiQ interface with integrated 24 V DC electronics power supply from the Control Unit or Motor Module
- 1 PE connection

Article No.		6SL3055-0AA00-5EA3	6SL3055-0AA00-5HA3
Product designation		SME20 Sensor Module External	SME25 Sensor Module External
Encoder		• Incremental encoder sin/cos 1 V_{pp} with 5 V voltage supply 0.35 A	• Absolute encoder EnDat with 5 V voltage supply 0.35 A • Absolute encoders SSI with incremental signals sin/cos 1 V_{pp} with 5 V voltage supply 0.35 A
Signal subdivision (interpolation)		≤ 16384 times (14 bit)	≤ 16384 times (14 bit)
Max. encoder frequency that can be evaluated	kHz	≤ 500	≤ 500
SSI/EnDat 2.1 baud rate	kHz	–	100
Measuring system interface		12-pin M23 circular connector	17-pin M23 circular connector
Output		IP67 DRIVE-CLiQ connector	IP67 DRIVE-CLiQ connector
Power requirement, max. At 24 V DC, without considering encoder	A	0.11	0.11
• Conductor cross-section		According to connector contacts	According to connector contacts
• Fuse protection		Via DRIVE-CLiQ power supply source	Via DRIVE-CLiQ power supply source
Power loss, max.	W	4	4
PE connection		M4 screw/1.8 Nm	M4 screw/1.8 Nm
Cable length, max.			
• To measuring system[1)]	m (ft)	3 (9.84)	3 (9.84)
• To automatic speed control	m (ft)	100 (328)	100 (328)
Degree of protection		IP67	IP67
Dimensions			
• Width	mm (in)	58 (2.28)	58 (2.28)
• Height	mm (in)	44 (1.73)	44 (1.73)
• Depth	mm (in)	112 (4.41)	112 (4.41)
Net weight	kg (lb)	0.31 (0.68)	0.31 (0.68)
Certificate of suitability		cULus	cULus

Notes:

SME120/125



SME120/SME125 have the following interfaces:

- 1 encoder connector via circular connector
- 1 temperature sensor connection via circular connector
- 1 Hall-effect sensor connection via circular connector (SME120 only)
- 1 DRIVE-CLiQ interface with integrated 24 V DC electronics power supply from the Control Unit or Motor Module
- 1 PE connection

SME120/125 Information

- Suitable for use in linear and torque motor applications. They can be installed close to the motors and encoders on the machine.
- SMEs evaluate the encoder signals and motor temperature sensors specifically and convert the information for DRIVE-CLiQ.
- The motor temperature signals are safely electrically separated.
- A Hall-effect sensor box can be connected for the SME120 to determine the commutation position of a linear motor.
- The SME120 and SME125 can be operated on Control Units with firmware release V2.4 and later.

Encoder Types (depending on the type of Sensor Module):

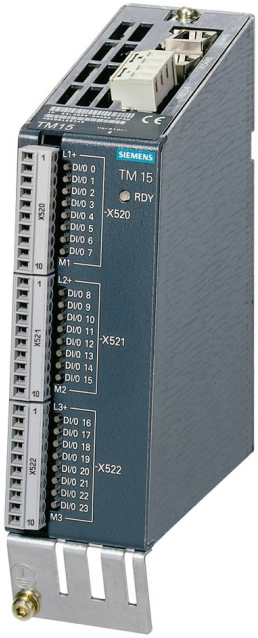
- Incremental encoder sin/cos 1 Vpp
- Absolute encoder EnDat 2.1
- SSI absolute encoder1) with sin/cos 1 Vpp incremental signals, but without reference signal
- The motor temperature can also be detected using KTY84-130, PT1000 or PTC thermistors.

Article No.		6SL3055-0AA00-5JA3	6SL3055-0AA00-5KA3
Product designation		SME120 Sensor Module External	SME125 Sensor Module External
Encoder		• Incremental encoder sin/cos 1 V_{pp} with 5 V voltage supply	• Absolute encoder EnDat with 5 V voltage supply • Absolute encoders SSI with incremental signals sin/cos 1 V_{pp} with 5 V voltage supply
Signal subdivision (interpolation)		≤ 16384 times (14 bit)	≤ 16384 times (14 bit)
Max. encoder frequency that can be evaluated	kHz	≤ 500	≤ 500
SSI/EnDat 2.1 baud rate	kHz	–	100
Measuring system interface		12-pin M23 circular connector	17-pin M23 circular connector
Temperature sensor input		6-pin M17 circular connector	6-pin M17 circular connector
Hall-effect sensor input		9-pin M23 circular connector	–
Output		IP67 DRIVE-CLiQ connector	IP67 DRIVE-CLiQ connector
Power requirement, max. at 24 V DC, without considering encoder	A	0.16	0.16
• Load rating of encoder power supply, for measuring system (at 5 V DC) and, where applicable, including Hall-effect sensor box	A	0.35	0.35
• Conductor cross-section		According to connector contacts	According to connector contacts
• Fuse protection		Via DRIVE-CLiQ power supply source	Via DRIVE-CLiQ power supply source
Power loss, max.	W	4.5	4.5
PE connection		M4 screw/1.8 Nm	M4 screw/1.8 Nm
Cable length, max.			
• To measuring system[1] / temperature sensor	m (ft)	3 (9.84)	3 (9.84)
• To automatic speed control	m (ft)	100 (328)	100 (328)
Degree of protection		IP67	IP67
Dimensions			
• Width	mm (in)	117.6 (4.63)	117.6 (4.63)
• Height	mm (in)	44 (1.73)	44 (1.73)
• Depth	mm (in)	127 (5)	127 (5)
Net weight	kg (lb)	0.7 (1.54)	0.7 (1.54)
Certificate of suitability		cULus	cULus

Notes:

Terminal Modules (TM)

Terminal modules offer a different types of functionality, depending on the module version in use.
The functionality includes evaluation of digital and analogue I/O, temperature sensor monitoring, relay outputs, encoder zero mark monitoring, encoder simulation.
Terminal modules are always used on the DRIVECLiQ drive bus.



TM 15

The number of available digital inputs and outputs within a drive system can be expanded with the TM15 Terminal Module.
The TM15 Terminal Module communicates with a Control Unit via DRIVE-CLiQ.

The following interfaces are located on the TM15 Terminal Module:

- 24 bidirectional digital inputs/outputs (isolation in 3 groups with 8 channels each)
- 24 green status LEDs for indicating the logical signal status of the relevant terminal
- 2 DRIVE-CLiQ sockets
- 1 connection for the 24 V DC electronics power supply
- 1 PE connection

Article No.	6SL3055-0AA00-3FA0
Product designation	TM15 Terminal Module
Power requirement, max. at 24 V DC without load	0.15 A
• Conductor cross-section, max.	2.5 mm^2
• Fuse protection, max.	20 A
Number of DRIVE-CLiQ sockets	2
I/O devices	
• Digital inputs/outputs	Can be parameterized channel-by-channel as DI or DO
• Number of digital inputs/outputs	24
• Electrical isolation	Yes, in groups of 8
• Connection method	Plug-in screw-type terminals
• Conductor cross-section, max.	1.5 mm^2
Digital inputs	
• Voltage	-30 ... +30 V
• Low level (an open digital input is interpreted as "low")	-30 ... +5 V
• High level	15 ... 30 V
• Current consumption at 24 V DC	5 ... 11 mA
• Delay times of digital inputs, typ. [1]	
- L → H	50 µs
- H → L	100 µs

Digital outputs (continuously short-circuit proof)	
• Voltage	24 V DC
• Load current per digital output, max.	0.5 A
• Delay times (resistive load) [1]	
- L → H, typ.	50 µs
- L → H, max.	100 µs
- H → L, typ.	150 µs
- H → L, max.	225 µs
• Total current of outputs (per group), max.	
- To 60 °C (140 °F)	2 A
- To 50 °C (122 °F)	3 A
- To 40 °C (104 °F)	4 A
Power loss, max.	3 W
PE connection	M4 screw
Dimensions	
• Width	50 mm (1.97 in)
• Height	150 mm (5.91 in)
• Depth	111 mm (4.37 in)
Weight, approx.	1 kg (2.20 lb)
Certificate of suitability	cULus

Notes:

TM 31

The TM31 can be used to increase the number of available digital/analogue inputs and outputs within a drive system.
The TM31 module also features relay outputs with changeover contact and a temperature sensor input.



The following interfaces are located on the TM31 Terminal Module:

- 8 digital inputs
- 4 bidirectional digital inputs/outputs
- 2 relay outputs with changeover contact
- 2 analogue inputs
- 2 analogue outputs
- 1 temperature sensor input (KTY84 -130/PT1000 or PTC)
- 2 DRIVE-CLiQ connections
- 1 connection for the 24 V DC electronics power supply
- 1 PE connection

Article No.	6SL3055-0AA00-3AA1
Product designation	TM31 Terminal Module
Current requirement, max. At 24 V DC without taking account of the digital outputs and DRIVE-CLiQ supply	0.5 A
• Conductor cross-section, max.	2.5 mm^2
• Fuse protection, max.	20 A
Digital inputs In accordance with IEC 61131-2 Type 1	
• Voltage	-3 ... +30 V
• Low level (an open digital input is interpreted as "low")	-3 ... +5 V
• High level	15 ... 30 V
• Current consumption at 24 V DC, typ.	9 mA
• Delay times of digital inputs, 1), approx.	
- L → H	50 µs
- H → L	100 µs
• Conductor cross-section, max.	1.5 mm^2
Digital outputs (continuously short-circuit proof)	
• Voltage	24 V DC
• Load current per digital output, max.	100 mA
• Total current of digital outputs, max.	400 mA
• Delay times of digital outputs 1)	
- Typ.	150 µs with 0.5 A resistive load
- Max.	500 µs
• Conductor cross-section, max.	1.5 mm^2
Analog inputs a switch is used to toggle between voltage and current input	
• As voltage input	
- Voltage range	-10 ... +10 V
- Internal resistance R_i	100 kΩ
- Resolution 2)	11 bits + sign
• As current input	
- Current ranges	4 ... 20 mA, -20 ... +20 mA, 0 ... 20 mA
- Internal resistance R_i	250 Ω
- Resolution 2)	10 bits + sign
• Conductor cross-section, max.	1.5 mm^2
Analog outputs (continuously short-circuit proof)	
• Voltage range	-10 ... +10 V
• Max. load current	-3 ... +3 mA
• Current ranges	4 ... 20 mA, -20 ... +20 mA, 0 ... 20 mA
• Load resistance, max.	500 Ω in the range -20 ... +20 mA
• Resolution	11 bits + sign
• Conductor cross-section, max.	1.5 mm^2
Relay outputs CO contacts	
• Max. load current	8 A
• Operational voltage, max.	250 V AC, 30 V DC
• Switching capacity, max.	
- At 250 V AC	2000 VA (cos φ = 1) 750 VA (cos φ = 0.4)
- At 30 V DC	240 W (resistive load)
• Required minimum current	100 mA
• Conductor cross-section, max.	2.5 mm^2
Power loss, max.	10 W
PE connection	M4 screw
Dimensions	
• Width	50 mm (1.97 in)
• Height	150 mm (5.91 in)
• Depth	111 mm (4.37 in)
Weight, approx.	0.49 kg (1.08 lb)
Certificate of suitability	cULus

Notes:

TM 41

The TM41 can be used to increase the number of available digital/analogue inputs and outputs within a drive system.
The module also features an encoder interface which delivers TTL signals, as incremental encoder simulation, to the **C**ontrol **U**nit.



The following interfaces are located on the TM41 Terminal Module:

- 4 bidirectional digital inputs/outputs
- 4 digital inputs (with electrical isolation)
- 1 analog input
- 1 interface for emulation of TTL incremental encoder (RS422)
- 1 LED for signaling zero mark detection for encoder interface
- 2 DRIVE-CLiQ sockets
- 1 connection for the 24 V DC supply of the digital outputs
- 1 connection for the electronics power supply via the 24 V DC supply connector
- 1 PE connection

Article No.	6SL3055-0AA00-3PA1
Product designation	TM41 Terminal Module
Power requirement (X524 at DC 24 V) without DRIVE-CLiQ supply or digital outputs (X514)	0.5 A
• Conductor cross-section, max.	2.5 mm^2
• Fuse protection, max.	20 A
I/O devices	
• Digital inputs/outputs	Individually parameterizable as DI or DO
• Number of digital inputs/outputs	4
• Number of digital inputs/outputs (with isolation)	4
• Connection method	Plug-in screw-type terminals
• Conductor cross-section, max.	1.5 mm^2
Digital inputs	
• Voltage	
- Without electrical isolation	-3 ... +30 V
- With electrical isolation	-30 ... +30 V
• Low level (an open digital input is interpreted as "low")	
- Without electrical isolation	-3 ... +5 V
- With electrical isolation	-30 ... +5 V
• High level	15 ... 30 V
• Current consumption at 24 V DC, typ.	<9 mA
• Delay times of digital inputs, max. [1]	
- L → H	3 ms
- H → L	3 ms

Digital outputs (continuously short-circuit-proof)	
• Voltage	24 V DC
• Load current per digital output, max.	0.5 A
• Delay times (resistive load) [1]	
- L → H, typ.	50 µs
- L → H, max.	100 µs
- H → L, typ.	75 µs
- H → L, max.	150 µs
Analog input (difference)	
• Voltage range	-10 ... +10 V
• Internal resistance	≥100 kΩ
• Resolution [2]	12 bits + sign
Pulse encoder emulation	
• Level	TTL (RS422), A+, A-, B+, B-, zero track N+, N-
• Limit frequency $f_{max.}$	512 kHz
• Ratio Encoder pulses: Encoder emulation	Any number of ratio/ reduction ratio of pulses
Power loss, max.	10 W
PE connection	M4 screw
Dimensions	
• Width	30 mm (1.18 in)
• Height	151 mm (5.94 in)
• Depth	110 mm (4.33 in)
Net weight	0.32 kg (0.71 lb)
Certificate of suitability	cULus

Notes:



TM120

The Terminal Module is a DRIVE-CLiQ component for safe electrically isolated temperature evaluation.

It can be used for 1FN, 1FW6, and third-party motors in which the temperature sensors cannot be installed with safe electrical separation.

When a TM120 is being used, temperature evaluation and encoder evaluation functions are isolated from one another.

The TM120 can detect the motor temperature via 4 channels with different temperature sensors. Encoder evaluation functions are performed via Sensor Modules (e.g. SMCxx, SMExx).

This means that, when connected to a Sensor Module SMCxx, the TM120 represents a control cabinet alternative to the SME120/SME125.

The following interfaces are located on the TM120 Terminal Module:

- 4 temperature sensor inputs (KTY84-130/PT1000 or PTC)
- 2 DRIVE-CLiQ sockets
- 1 connection for the 24 V DC electronics power supply
- 1 PE connection

Article No.	6SL3055-0AA00-3KA0
Product designation	TM120 Terminal Module
Power requirement, max. at 24 V DC	0.5 A
• Conductor cross-section, max.	2.5 mm^2
• Fuse protection, max.	20 A
Temperature sensor inputs The inputs can be individually parameterized to evaluate sensors of type KTY84-130/PT1000 or PTC, or temperature switches	
• Conductor cross-section	0.2 ... 6 mm^2
• Constant current per sensor, approx.	2 mA
• Safe electrical separation up to line voltage, max.	480 V AC
Power loss, typically	2.4 W
PE connection	M4 screw
Dimensions	
• Width	30 mm (1.18 in)
• Height	150 mm (5.91 in)
• Depth	111 mm (4.37 in)
Net weight	0.41 kg (0.90 lb)

Notes:

DRIVE CLIQ Hubs

DMC20 and DME20 Functionality:

- To expand upon the number of available DRIVECLiQ interfaces, for example, when there are multiple additional drive components such as second measuring circuits etc.
- The hubs can be connected in a Star configuration to
  - facilitate the easy removal/parking of optional drive objects and components without disturbing the DIVECLiQ connections to remaining axes.
  - reduce cabling requirements between the electrical cabinet and machine mounted drive components.

Hubs in series: it is possible to connect a maximum of two hubs in series.



DMC20 - Cabinet Mounted

The following interfaces are located on the DMC20 DRIVE-CLiQ Hub Module:

- 6 DRIVE-CLiQ sockets for connecting 5 additional DRIVE-CLiQ devices
- 1 connection for the 24 V DC electronics power supply

Article No.	6SL3055-0AA00-6AA1
Product designation	DMC20 DRIVE-CLiQ Hub Module
Power requirement at 24 V DC, max. without DRIVE-CLiQ supply	0.15 A
• Conductor cross-section, max.	2.5 mm^2
Degree of protection	IP20
Width	50 mm (1.97 in)
Height	151 mm (5.94 in)
Depth	110 mm (4.33 in)
Net weight	0.36 kg (0.79 lb)
Certificate of suitability	cULus

DME20 - Externally Mounted



The following interfaces are located on the DME20 DRIVECLiQ Hub Module:

- 6 DRIVE-CLiQ sockets for connecting 5 DRIVE-CLiQ devices
- 1 connection for the 24 V DC electronics power supply
  (circular connector - 4 × 0.75 mm², pins 1+2 internally bridged, pins 3+4 internally bridged)

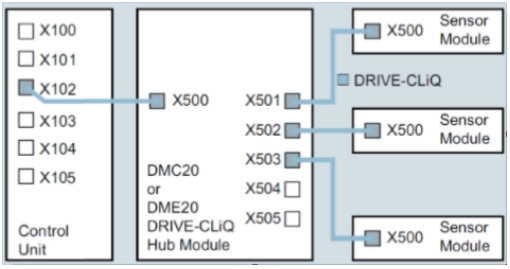
Blanking plugs are available for unused DRIVECLiQ sockets.

Article No.	6SL3055-0AA00-6AB0
Product designation	DME20 DRIVE-CLiQ Hub Module
Power requirement at 24 V DC, max. without DRIVE-CLiQ supply	0.15 A
• Conductor cross-section, max.	4 × 0.75 mm^2
Degree of protection	IP67
Width	99 mm (3.9 in)
Height	149 mm (5.87 in)
Depth	55.7 mm (2.19 in) without connector
Net weight	0.8 kg (1.76 lb)
Certificate of suitability	cULus

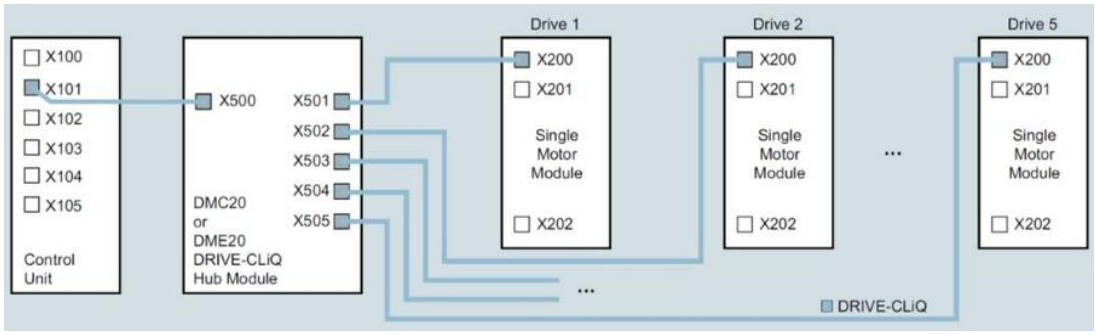
Notes:

DRVECLiQ Hub Connection Examples:

Expanding upon the number of available DRIVECLiQ interfaces, for example, when there are multiple additional drive components such as second measuring circuits etc.



Facilitating the easy removal/parking of optional drive objects and components without disturbing the DIVECLiQ connections to remaining axes and/or reducing the cabling requirements between the electrical cabinet and machine mounted drive components.



Notes:

Sinumerik CNC Portfolio - 808D to ONE

Entry level machines

SINUMERIK 808D





- Panel based Compact CNC
- Turning and Milling
- Up to 6 axes/spindles
- 1 channel
- 7.5” / 8.4” - display
- SIMATIC S7-200 based PLC

Compact machines

SINUMERIK 828D



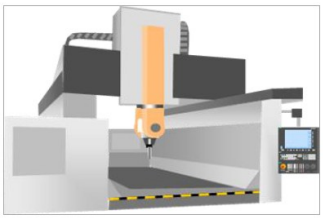


- Panel based Compact CNC
- Turning, Milling and grinding
- Up to 10 axes/spindles
- 2 channels
- 10.4” / 15.5” - multi-touch display
- SIMATIC S7-200 based PLC

Modular machines

SINUMERIK 840D sl



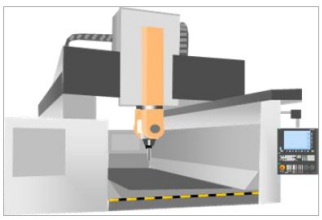


- Drive based modular CNC
- Multi technology
- Up to 31 axes/spindles
- Up to 10 channels
- 15“ to 24“- multi touch display modular
- SIMATIC S7-300 based PLC

Modular machines

SINUMERIK ONE - Real and virtual





- Drive based Compact CNC & panel based Compact CNC
- Multi technology
- Up to 31 axes/spindles
- Up to 10 channels
- 15“ to 24“- multi touch display modular
- SIMATIC S7-1500 based PLC

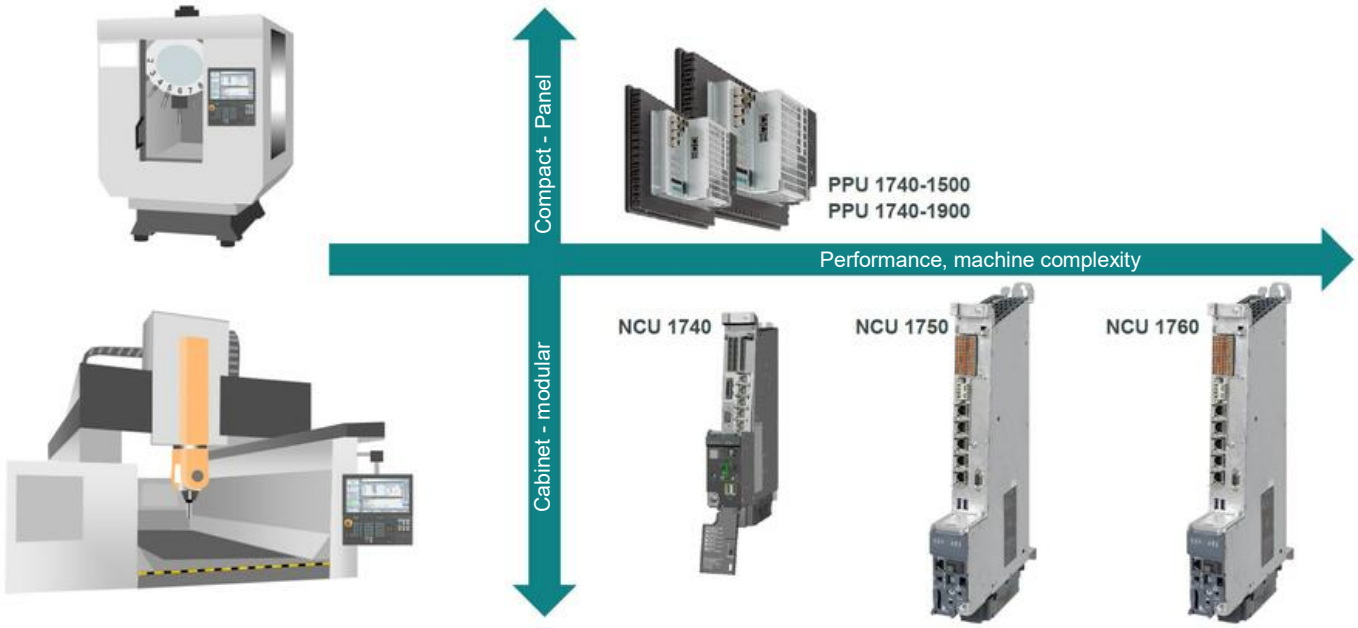
Virtual SINUMERIK:

- Digital twin of product (machine tool)
- Digital twin of production

Notes:

SINUMERIK ONE - Overview

Portfolio Overview:



SINUMERIK ONE – NCUs

Main features in comparison with SINUMERIK 840D sl;

SINUMERIK 840D sl		SINUMERIK ONE	
	NCU710: 8 axes/spindles NCU720/730: 31 axes/spindles	NCU1740*: 12 axes/spindles NCU1750/60: 31 axes/spindles	
	PLC S7-300	PLC S7-1500F	
	1x PROFINET (2 ports NRT/RT/IRT)	1x PROFINET (2 ports, NRT/RT/IRT) 1x PROFINET (1 port, NRT/RT)	
	2x PROFIBUS DP	1x PROFIBUS DP	
	2x USB 2.0	2x USB 3.0	
	PLC rotary switch	PLC toggle switch	
	CF card slot	SD card slot	
	Buffer battery	No buffer battery	

* Note: NCU 1740 - in preparation

Notes:

SINUMERIK ONE PPU1740

Combining the best from Sinumerik 840D sl NCU 710 and SINUMERIK 828D

SINUMERIK 840D sl NCU 710—cabinet based	
	• All high end technologies • Machine network (X120) • Standard PROFINET periphery
SINUMERIK 828D / PPU 290 - panel based	
	• Mult-touch panel • No buffer battery • Onboard periphery: 2 hand wheels 32 digital I/O 1 analogue output

Additional advantages SINUMERIK ONE PPU1740

SINUMERIK ONE PPU 1740 - Panel based	
	PLC S7-1500F
	2nd PROFINET interface
	SD Card slot
	12 axes / spindles + 8 positioning axes

SINUMERIK ONE PPU 1740

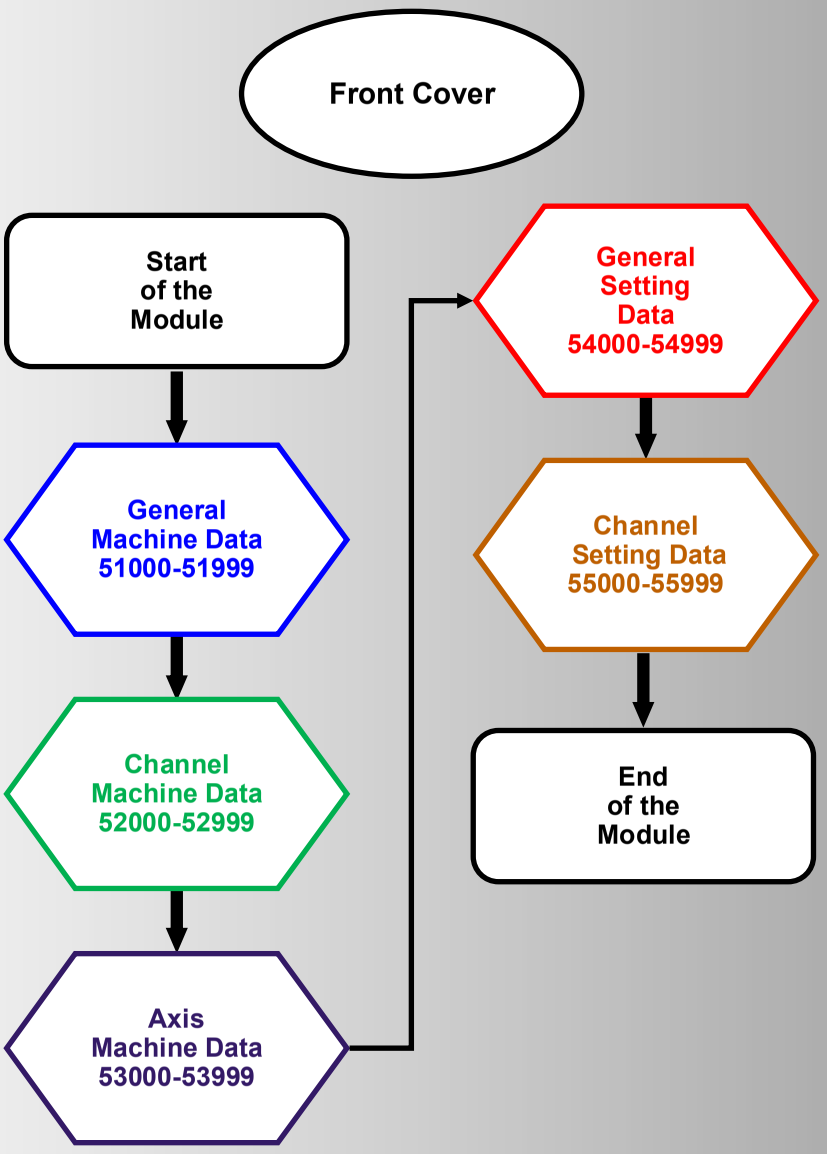
Main features in comparison with SINUMERIK 840D sl NCU710 / 828D:

SINUMERIK 840D sl NCU 710—cabinet based		SINUMERIK ONE PPU 1740	
	8 axes / spindles, + 7 position axes	12 axes/spindles + 8 position axes	
	-	2 x onboard hand-wheels	
	24 digital I/O onboard -	32 onboard digital I/O 1 onboard analogue output (spindle)	
	1 x PROFINET 2x PROFIBUS DP	2 x PROFINET -	
	2x USB 2.0	3 x USB 3.0	
	Fan and battery	Without fan and battery	
SINUMERIK 828D / PPU 290 - panel based			
	PLC S7-200	PLC S7-1500F	
	Proprietary periphery	Standard PROFINET periphery	
	-	System network (X120)	
	M, T, G technologies	All high end technologies	

Appendix A: END

Notes:

Technology Settings



Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

[General Machine Data]

MD	Identifier	Description	Comment
51000	$MNS_DISP_RES_MM	Display resolution in mm	
51001	$MNS_DISP_RES_MM_FEED_PER_REV	Display resolution in mm feedrate/rev	
51002	$MNS_DISP_RES_MM_FEED_PER_TIME	Display resolution in mm feedrate/min	
51003	$MNS_DISP_RES_MM_FEED_PER_TOOTH	Display resolution in mm feedrate/tooth	
51004	$MNS_DISP_RES_MM_CONST_CUT_RATE	Display resolution constant cutting speed m/min	
51010	$MNS_DISP_RES_INCH	Display resolution in inch	
51011	$MNS_DISP_RES_INCH_FEED_P_REV	Display resolution in inch feedrate/rev	
51012	$MNS_DISP_RES_INCH_FEED_P_TIME	Display resolution in inch feedrate/min	
51013	$MNS_DISP_RES_INCH_FEED_P_TOOTH	Display resolution in inch feedrate/tooth	
51014	$MNS_DISP_RES_INCH_CUT_RATE	Display resolution constant cutting speed ft/min	
51018	$MNS_DISP_RES_SCALE	Display resolution scaling	
51019	$MNS_DISP_RES_ROT_WO	Display resolution of rotation in work offsets	
51020	$MNS_DISP_RES_ANGLE	Display resolution of angle	
51021	$MNS_DISP_RES_SPINDLE	Decimal places in speed entry field	
51022	$MNS_DISP_RES_ROT_AX_FEED	Display resolution of rotary axis feedrate	
51023	$MNS_ACT_VALUE_SPIND_MODE	Only display spindles in actual values window This affects the display of the spindles in the axis actual values window. 0 = Spindles are always displayed 1 = Spindles are displayed only if in axis mode	
51024	$MNS_BLOCK_SEARCH_MODE_MASK_JS	Bit mask for available search modes (ShopMill/ShopTurn single-channel) Bit 0: Block search with calculation without approach Bit 1: Block search with calculation with approach Bit 2: Reserved Bit 3: Skip EXTCALL programs Bit 4: Reserved Bit 5: Block search with test run with approach Bit 6: Block search with test run without approach	
51025	$MNS_FRAMES_ACT_IMMEDIATELY	Active data (frames) are activated immediately after change	
51026	$MNS_AXES_SHOW_GEO_FIRST	When the machine data value is 1, the geometry axes of the channel are displayed first.	
51027	$MNS_ONLY_MKS_DIST_TO_GO	Distance-to-go display in work window	

[General Machine Data]

MD	Identifier	Description	Comment
51028	$MNS_BLOCK_SEARCH_MODE_MASK	Bit mask for available block search modes Bit 0: Block search with calculation without approach Bit 1: Block search with calculation with approach Bit 3: Skip EXTCALL programs Bit 4: Block search without calculation Bit 5: Block search with test run with approach Bit 6: Block search with test run without approach	
51029	$MNS_MAX_SKP_LEVEL	The machine data defines how many skip levels are made available for operation.	
51030	$MNS_SPIND_MAX_POWER	Maximum value of the permissible spindle power rating in percent; the display bar in the machine image is shown in green within the range between 0 and the value stored in SPIND_MAX_POWER.	
51031	$MNS_SPIND_POWER_RANGE	Scale end value for spindle power rating in percent; value must be equal to or greater than SPIND_MAX_POWER. The display bar in the machine image is shown in red in the range between the values of SPIND_MAX_POWER and SPIND_POWER_RANGE.	
51034	$MNS_TEACH_MODE	Teach mode to be activated Bit 0: Default teach-in Taught-in block is transferred to the program using the Accept softkey. Bit 1: Acceptance of teach block can be blocked by the PLC. - For ONE/840D sl : PLC bit DB19.DBX13.0 Disable teach transfer - For 828D : PLC bit DB1700.DBX1000.0 Disable teach transfer Bit 2: Block selection only explicitly	
51035	$MNS_WRITE_FRAMES_FINE_LIMIT	Input limit for all work offsets fine	
51036	$MNS_ENABLE_COORDINATE_REL	Display REL coordinate system 0 = no relative coordinate system selectable 1 = REL coordinate system can be selected as an alternative of the WCS/SZS coordinate system	
51037	$MNS_ENABLE_COORDINATE_ACS	Activate settable coordinate system 0 = WCS coordinate system is displayed. 1 = SZS coordinate system is displayed. (SZS is WCS reduced by the offset components defined in MD24030)	

[General Machine Data]

MD	Identifier	Description	Comment
51038	$MNS_SET_ACT_VALUE	Set actual value selection 0 = Set actual value is not offered. 1 = If a user frame (settable work offset e.g. G54) is active, it will be used. In G500 Set actual values is not offered (system frame is no longer used).	
51039	$MNS_PROGRAM_CONTROL_MODE_MASK	Options for machine - program influence: Bit 0: Program test function available	
51040	$MNS_SWITCH_TO_MACHINE_MASK	Automatic operating area switchover to machine Bit 0: When selecting a program in the program manager, the operating area does not automatically switch over to Machine. Bit 1: When switching the type of operation via the MCP, the operating area is not automatically switched over to Machine. Bit 2: Do not switch automatically in Programs during program selection. Bit 3: No auto. Start of block search on selection / execution under Programs.	
51041	$MNS_ENABLE_PROGLIST_USER	Activates the PLC program list of the USER area. The programs entered there can be selected by the PLC for processing.	
51042	$MNS_ENABLE_PROGLIST_INDIVIDUAL	Activates the PLC program list of the INDIVIDUAL area. The programs entered here can be selected by the PLC for processing.	Only 828D
51043	$MNS_ENABLE_PROGLIST_MANUFACT	Activates the PLC program list of the MANUFACTURER area. The programs entered here can be selected by the PLC for processing.	
51044	$MNS_ACCESS_SHOW_SBL2	Display protection level SBL2	
51045	$MNS_ACCESS_TEACH_IN	Protection level TEACH IN	
51046	$MNS_ACCESS_CLEAR_RPA	Protection level delete R variables	
51047	$MNS_ACCESS_READ_GUD_LUD	Read user variable protection level	
51048	$MNS_ACCESS_WRITE_GUD_LUD	Write protection level of user variables	
51049	$MNS_ACCESS_WRITE_PRG_COND	Write program control protection level	
51050	$MNS_ACCESS_WRITE_PROGRAM	Write part program protection level	
51051	$MNS_ACCESS_WRITE_RPA	Protection level write R variables	
51052	$MNS_ACCESS_WRITE_SEA	Protection level write setting data	
51053	$MNS_ACCESS_WRITE_BASEFRAME	Write basic work offset (basic frame) protection level	
51054	$MNS_ACCESS_WRITE_CYCFRAME	Write cycle frame protection level	
51055	$MNS_ACCESS_WRITE_EXTFRAME	Write external work offset protection level	

General Machine Data

MD	Identifier	Description	Comment
51056	$MNS_ACCESS_WRITE_PARTFRAME	Write table reference protection level	
51057	$MNS_ACCESS_WRITE_SETFRAME	Write basic reference protection level	
51058	$MNS_ACCESS_WRITE_TOOLFRAME	Write basic tool reference protection level	
51059	$MNS_ACCESS_WRITE_TRAFRAME	Write transformation frame protec. level	
51060	$MNS_ACCESS_WRITE_USERFRAME	Write settable work offset (G54 ... G599) protection level	
51061	$MNS_ACCESS_WRITE_WPFRAME	Write workpiece reference protection level	
51062	$MNS_ACCESS_WRITE_FINE	Write protection level for fine offset of all work offsets	
51063	$MNS_ACCESS_SET_ACT_VALUE	Set actual value protection level	
51064	$MNS_ACCESS_WRITE_PROGLIST	Write protection level of program list in USER area	
51065	$MNS_NUM_DISPLAYED_CHANNELS	Number of channels displayed simultaneously	
51066	$MNS_ORDER_DISPLAYED_CHANNELS	Channel numbers of the channels displayed Contains the numbers of the channels to be displayed under machine in the multi-channel view, in the desired order and separated by commas, semicolons or spaces.	
51067	$MNS_ENABLE_HANDWHEEL_WINDOW	Show handwheel window. If the machine data is set to 0, the window for handwheel assignment is hidden.	
51068	$MNS_SPIND_DRIVELOAD_FROM_PLC1	Machine axis index of spindle 1 utilization display from PLC	
51069	$MNS_SPIND_DRIVELOAD_FROM_PLC2	Machine axis index of spindle 2 utilization display from PLC	
51070	$MNS_ACCESS_CAL_TOOL_PROBE	Protection level for calibration of the tool probe (ShopTurn)	
51071	$MNS_ACCESS_ACTIVATE_CTRL_E	Protection level of CTRL-Energy	
51072	$MNS_ACCESS_EDIT_CTRL_E	Protection level of CTRL-Energy for changing profiles	
51073	$MNS_ACCESS_SET_SOFTKEY_ACCESS	Protection level of ”Adapt softkeys”	
51074	$MNS_ACCESS_WRITE_WPC_COUNTER	Write protection level of workpiece counter	
51075	$MNS_ACCESS_WRITE_MDI	Write protection level of MDI program	
51090	$MNS_HMI_HANDWHEEL_BUTTON	Number of the HMI handwheel for the handwheel button	

General Machine Data

MD	Identifier	Description	Comment
51160	$MNS_ACCESS_WRITE_CA_MACH_JOG	Write protection level collision monitoring machine jog	
51161	$MNS_ACCESS_WRITE_CA_MACH_AUTO	Write protection level collision monitoring machine automatic	
51162	$MNS_ACCESS_WRITE_CA_TOOL	Write protection level collision monitoring tool	
51198	$MNS_ACCESS_READ_TM_ALL_PARAM	Protection level TM details - read all parameters	
51199	$MNS_ACCESS_WRITE_TM_GRIND	Protection level of tool management for writing grinding data	
51200	$MNS_ACCESS_WRITE_TM_GEO	Write tool offset geometry data protection level	
51201	$MNS_ACCESS_WRITE_TM_WEAR	Write tool offset wear data protection level	
51202	$MNS_ACCESS_WRITE_TM_WEAR_DELTA	Protection level for restricted writing of tool wear values, s. SD 54213: TM_WRITE_DELTA_LIMIT	
51203	$MNS_ACCESS_WRITE_TM_SC	Write tool offset sum offset protection level	
51204	$MNS_ACCESS_WRITE_TM_EC	Write tool offset use offsets protection level	
51205	$MNS_ACCESS_WRITE_TM_SUPVIS	Write tool offset monitoring data protection level One authorization applies to all limit values: quantity, service life, wear and the monitoring type.	
51206	$MNS_ACCESS_WRITE_TM_ASSDNO	Write tool offset unique D number protection level	
51207	$MNS_ACCESS_WRITE_TM_WGROUP	Write tool offset wear groups (magazine location / magazine) protection level	
51208	$MNS_ACCESS_WRITE_TM_ADAPT	Write tool offset tool adapter geometry data protection level	
51209	$MNS_ACCESS_WRITE_TM_NAME	Write tool offset tool name and duplo data protection level	
51210	$MNS_ACCESS_WRITE_TM_TYPE	Write tool offset tool type protection level	
51211	$MNS_ACCESS_READ_TM	Read tool offset data protection level	
51212	$MNS_TM_WRITE_WEAR_ABS_LIMIT	Maximum tool wear value	
51213	$MNS_TM_WRITE_WEAR_DELTA_LIMIT	Maximum difference value restricted tool wear input	
51214	$MNS_TM_WRITE_LIMIT_MASK	Validity of the restricted tool wear input Bit 0: Use for cutting edge data, wear Bit 1: Use for SC data, sum offsets Bit 2: Use for EC data, use offsets	

Appendix B

[General Machine Data]

MD	Identifier	Description	Comment
51215	$MNS_ACCESS_WRITE_TM_ALL_PARAM	Protection level TM details - write all parameters	
51216	$MNS_ACCESS_TM_TOOL_CREATE	Protection level TM create tool	
51217	$MNS_ACCESS_TM_TOOL_DELETE	Protection level TM delete tool	
51218	$MNS_ACCESS_TM_TOOL_LOAD	Protection level TM load tool	
51219	$MNS_ACCESS_TM_TOOL_UNLOAD	Protection level TM unload tool	
51220	$MNS_ACCESS_TM_TOOL_MOVE	Protection level TM relocate tool	
51221	$MNS_ACCESS_TM_TOOL_REACTIVATE	Protection level TM reactivate tool	
51222	$MNS_ACCESS_TM_TOOL_MEASURE	Protection level TM measure tool	
51223	$MNS_ACCESS_TM_TOOLEDGE_CREATE	Protection level TM create tool cutting edge	
51224	$MNS_ACCESS_TM_TOOLEDGE_DELETE	Protection level TM delete tool cutting edge	
51225	$MNS_ACCESS_TM_MAGAZINE_POS	Protection level TM position magazine	
51226	$MNS_FUNCTION_MASK_SIM	Function mask Simulation Bit 0: No automatic start on simulation selection Bit 1: Deactivate simulation Bit 4: Ignore frame components in X and Y (compatibility) Bit 5: Turning tool can rotate in the tool spindle Bit 6: Enable handwheel as simulation override Bit 7: Interpretation of handwheel values as absolute values Bit 10: Hide zero point symbol Bit 22: Machine model for simulation from kinematic chain (also without collision avoidance) Bit 27: Stop simultaneous recording on termination	
51228	$MNS_FUNCTION_MASK_TECH	Function mask Cross-technology Bit 0: G code programming without multi-channel data Bit 1: Enable print function of editor Bit 2: With external program selection, test program	
51230	$MNS_ENABLE_LADDER_DB_ADDRESSES	Activation of DB addresses in PLC Ladder Viewer	Only 828D
51231	$MNS_ENABLE_LADDER_EDITOR	Activation of PLC ladder add-on tool for INT100/101	Only 828D
51232	$MNS_ENABLE_LADDER_EDITOR_ADV	Activation of PLC ladder editor for the complete PLC project	Only 828D
51233	$MNS_ENABLE_GSM_MODEM	Activation of GSM Modem for Easy Message	Only 828D
51235	$MNS_ACCESS_RESET_SERV_PLANNER	Protection level for acknowledgement of maintenance tasks	Only 828D

[General Machine Data]

MD	Identifier	Description	Comment
51240	$MNS_MACHINE_REGISTRATION	Machine registration Bit 0: Enable machine registration via QR code Bit 1: Machine has been registered	
51300	$MNS_MAXNUM_WAITM_USER	Number of wait marks available to the end user	
51600	$MNS_MEA_CAL_WP_NUM	Number of calibration data fields for workpiece probes	
51601	$MNS_MEA_CAL_EDGE_NUM	Number of geometry data fields of gauging block for workpiece probes	
51602	$MNS_MEA_CAL_TP_NUM	Number of calibration data fields for tool probes	
51603	$MNS_MEA_CAL_TPW_NUM	Number of calibration data fields for tool probes	
51618	$MNS_MEA_CM_ROT_AX_POS_TOL	Tolerance of rotary axis position—´measuring with orientable toolholder	
51740	$MNS_MEA_FUNCTION_MASK	Function mask for measuring cycles, workpiece and tool measuring Bit 0: Workpiece, monitoring of the calibration status Bit 1: Workpiece, length ref. of probe in infeed axis at the ball circumference Bit 2: Workpiece, correction in the tool, taking into account toolholders Bit 4: Workpiece, probe fixed in the machine (not in a spindle) Bit 5: Workpiece, correction in WO course, if no WO fine is set up Bit 6: Measuring blocks in measuring cycles with feedrate override 100% Bit 7: Workpiece, accept active WO in non-active WO, JOG mode only, set edge Bit 14: Workpiece, simultaneous measuring function active Bit 15: Workpiece, traverse measuring block with path behaviour G60 exact stop Bit 16: Tool, turning technology, taking into account toolholders	
51742	$MNS_MEA_FUNCTION_MASK	Meas. cycles: Limitation of protection levels when writing system variables	
51750	$MNS_J_MEA_M_DIST	This parameter defines the measuring path in front of and behind the measuring setpoint.	
51751	$MNS_J_MEA_M_DIST_MANUELL	Measuring path, manual measurement ((for "Measure in JOG")) [mm] (in front of and behind the measuring setpoint)	
51752	$MNS_J_MEA_M_DIST_TOOL_LENGTH	Measuring path for tool length measuring.	
51753	$MNS_J_MEA_M_DIST_TOOL_RADIUS	Measuring path for tool radius measuring.	
51757	$MNS_J_MEA_COLL_MONIT_FEED	Feedrate in the working plane with active collision detection.	
51758	$MNS_J_MEA_COLL_MONIT_POS_FEED	Feedrate of the infeed axis with active collision detection, for "Measure in JOG".	

General Machine Data

MD	Identifier	Description	Comment
51770	$MNS_J_MEA_CAL_RING_DIAM	Calibration ring diameter for Measure in JOG	
51772	$MNS_J_MEA_CAL_HEIGHT_FEEDAX	Calibration height in the infeed axis for probe length calibration	
51780	$MNS_J_MEA_T_PROBE_DIAM_RAD	Diameter of the tool probe for radius measurement for Measure in JOG	
51781	$MNS_MEA_T_PROBE_THICKNESS	Tool probe thickness	
51784	$MNS_J_MEA_T_PROBE_APPR_AX_DIR	Approach direction in the plane on the tool probe for Measure in JOG	
51786	$MNS_J_MEA_T_PROBE_MEASURE_DIST	Measuring path for tool measurement with stationary spindle for Measure in JOG	
51840	$MNS_GRIND_FUNCTION_MASK	Function mask for grinding cycles Bit 0: Reference point for recording dresser/grinding wheel is a work offset Bit 1: Reserved	
51900	$MNS_CAS_ACCESS_LEVEL	Collision Avoidance System access level for UI functionality	
51901	$MNS_CAS_CONFIG_MASK	Collision Avoidance System function mask Bit 0: Bit0 Bit 1: Bit1	
51902	$MNS_CAS_CONFIG_MASK_EXT	Collision Avoidance System extended function mask Bit 0: Bit0 Bit 1: Bit1	
51903	$MNS_CAS_CONFIG_MASK_INT	Collision Avoidance System configuration - integers	

Channel Machine Data

MD	Identifier	Description	Comment
52000	$MCS_DISP_COORDINATE_SYSTEM	Coordinate system position With this MD you adapt the operator panel of the coordinate system to the machine's coordinate system. Depending on the selected position, all help screens, the sequence graphic, the simulation and the input fields with the circular direction specified will change automatically. Also note MD 52210 $MCS_FUNCTION_MASK_DISP, Bit 1.	
52001	$MCS_DISP_COORDINATE_SYSTEM_2	Coordinate system for turning on milling machines	
52004	$MCS_PROG_COORDINATE_SYS_CHAN	Coordinate system during programming as described in the specific channel	
52005	$MCS_DISP_PLANE_MILL	Plane selection Milling 0: plane selection on the operator panel 17: always G17 18: always G18 19: always G19	
52006	$MCS_DISP_PLANE_TURN	Plane selection Turning 0: plane selection on the operator panel 17: always G17 18: always G18 19: always G19	
52010	$MCS_DISP_NUM_AXIS_BIG_FONT	Number of actual values with large font	
52011	$MCS_ADJUST_NUM_AXIS_BIG_FONT	Adapt the number of actual values with large font if the number of geometry axes changes, e.g. due to transformations like TRANSMIT or TRACYL. 0 = Only MD 52010 DISP_NUM_AXIS_BIG_FONT is valid. The number is assigned as a fixed value. 1 = Only the geometry axes are displayed in large font. MD 52010 DISP_NUM_AXIS_BIG_FONT is ignored. 2 = The number of geometry axes plus the content of MD 52010 DISP_NUM_AXIS_BIG_FONT are displayed in large font.	
52020	$MCS_ORIAXES_EULER_ANGLE_NAME	Axis identifier of the orientation axes: Euler angle or geometry axis name	
52032	$MCS_STAT_DISPLAY_BASE	Number basis for display of articulated joint STAT	
52033	$MCS_TU_DISPLAY_BASE	Number basis for display of rotary axis position TU	

Channel Machine Data

MD	Identifier	Description	Comment
52200	$MCS_TECHNOLOGY	Technology 0: no specific configuration 1: turning 2: milling	
52202	$MCS_TECHNOLOGY_ADDON	Additional technologies Bit 0: Grinding tools (types 40x) Bit 1: Grinding tools (types 41x)	
52201	$MCS_TECHNOLOGY_EXTENSION	Extended technology 0: no specific configuration 1: turning 2: milling	
52206	$MCS_AXIS_USAGE	Meaning of the axes in the channel 0 = no special meaning 1 = tool spindle (driven tool) 2 = auxiliary spindle (driven tool) 3 = main spindle (turning) 4 = C axis of the main spindle (turning) 5 = counter spindle (turning) 6 = C axis of the counter spindle (turning) 7 = linear axis of the counter spindle (turning) 8 = tailstock (turning) 9 = steady rest (turning) 10 = B axis (turning) 11 = reserved 12 = B axis of the counter spindle (turning) 13 = Traverse travel X of the counter spindle (turning)	
52207	$MCS_AXIS_USAGE_ATTRIB	Axis attributes Bit 0: Rotates around the 1st geometry axis (applies to rotary axes) Bit 1: Rotates around the 2nd geometry axis (applies to rotary axes) Bit 2: Rotates around the 3rd geometry axis (applies to rotary axes) Bit 3: Positive direction of rotation is counter-clockwise (applies to rotary axes) Bit 4: Displayed direction of rotation for M3 is counter-clockwise (applies to spindles) Bit 5: Direction of rotation M3 corresponds to rotary axis minus (applies to spindles) This bit must be set in the same way as a PLC bit. - For ONE/840D sl : PLC bit DBnn.DBX17.6, nn = 31 + machine axis index - For 828D : PLC bit DB38xx.DBX2001.6, xx is machine axis index Bit 6: Show rotary axis as offset target for measuring Bit 7: Offer rotation axis in position pattern Bit 8: Offer rotation axis to accept blank (at milling machines) Bit 9: Spindle is not SPOS-capable Bit 10: Rotary axis rotates around 1st geometry axis (for position pattern only) Bit 11: Rotary axis rotates around 2nd geometry axis (for position pattern only) Bit 12: Rotary axis rotates around 3rd geometry axis (for position pattern only) Bit 13: Position of the axis for blank selection from active WO (for rotary axes) Bit 14: Enable spindle for teach-in in axis mode	

Channel Machine Data

MD	Identifier	Description	Comment
52210	$MCS_FUNCTION_MASK_DISP	Function mask Display Bit 0: Measuring system for programs always in the base system Bit 1: Front view for turning in school coordinate system Bit 2: Hide “T,S,M” soft key in JOG area Bit 3: Generate automatic end-of-program in MDI (with the “Delete blocks” soft key) Bit 4: Show follow-on tool in T,F,S window Bit 5: Hide softkey “Actual machine values“ Bit 6: Hide tool radius/diameter in T,F,S window Bit 7: Hide tool length in T,F,S window Bit 8: Hide tool icon in T,F,S window Bit 9: Display selection ”Simple input” Bit 10: Do not offer channel in job lists Bit 11: For Work activate the display of the logical spindles Bit 12: Hide mold making view for G code Bit 13: Hide channel axes which are currently located in another channel Bit 14: Display overlay of DRF under machine Bit 15: Display overlay of tool ($AA_TOFF) under machine Bit 16: Select JOG axis via HMI Bit 17: Select increment via HMI Bit 18: Offer select “With set stop override to0%” with configured stop Bit 19: Softkey bars in the Program area user-defined	
52211	$MCS_FUNCTION_MASK_DISP _ZOA	Function screen for displaying overview of work offsets Bit 0: Show machine position Bit 1: reserved Bit 2: Display DRF offset Bit 3: Display $AA_OFF position offset Bit 4: Display $P_PARTFRAME Bit 5: Display $P_SETFRAME Bit 6: Display $P_EXTSFRAME Bit 7: Display $P_ISO1FRAME Bit 8: Display $P_ISO2FRAME Bit 9: Display $P_ISO3FRAME Bit10: Display $P_ACTBFRAME Bit 11: Display $P_IFRAME Bit 12: Display $P_TOOLFRAME Bit 13: Display $P_WPFRAME Bit 14: Display $P_TRAFRAME Bit 15: Display $P_PFRAME Bit 16: Display $P_ISO4FRAME Bit 17: Display $P_CYCFRAME Bit 18: Display sum of work offsets Bit 19: Display offset of active tool Bit 20: Display Work position Bit 21: Display BCS position Bit 22: Display $P_GFRAME Bit 23: Display TOFF Bit 24: Display overlay $AA_TOFF	

Channel Machine Data

MD	Identifier	Description	Comment
52212	$MCS_FUNCTION_MASK_TECH	Function mask Cross-technology Bit 0: Enable swivel plane/align tool with tool carrier Bit 1: No optimized travel along software limit switches (ShopMill/ShopTurn) Bit 2: Special approach logic for step drill (ShopTurn) Bit 3: Call block search cycle for ShopMill/ShopTurn Bit 4: Approach logic through cycle (ShopTurn) Bit 5: Call block search cycle for SERUPRO Bit 6: Work offset ZV cannot be entered as an absolute value (ShopTurn) Bit 7: Detect the expiry of tool lives (ShopMill/ShopTurn) Bit 8: Manual machine (ShopMill/ShopTurn) Bit 9: Selection/deselection of work offset via softkey Bit 10: Enable swivel plane/align tool using kinematic transformation Bit 11: Switch off layer check for drilling and milling tools (ShopTurn) Bit 12: reserved Bit 13: Approach tool change point Y in Machine (ShopTurn) Bit 14: With approach logic only check tool position with cutting edge (ShopTurn) Bit 15: Do not set $MC TOOL_CARRIER_RESET_VALUE after block search Bit 16: Do not automatically execute swivel plane after block search Bit 17: Do not automatically execute align turning tool after block search Bit 18: Enable Determine total load : empty axis + workpiece Bit 19: Do not execute automatic TRAFO activation after block search (SEATRAON) Bit 20: Hide interpolation turning in this channel Bit 21: Hide Y turning in this channel	
52214	$MCS_FUNCTION_MASK_MILL	Function mask Milling Bit 0: Enable cylinder surface transformation (ShopMill) Bit 1: Offer fixed table for blank clamping (at milling machines) Bit 2: Special handling of slide mill/saw for polyhedron Bit 3: Enable inside/rear machining (ShopTurn) Bit 4: Enable spindle clamping (C axis) (ShopTurn) Bit 5: Enable spindle control of tool spindle via surface Bit 6: Enable spindle control of turning spindle via surface Bit 7: Offer fixed table for blank clamping (on boring mills) Bit 8: Enable damping braking (C axis) (ShopTurn)	

Channel Machine Data

MD	Identifier	Description	Comment
52216	$MCS_FUNCTION_MASK_DRILL	Function mask Drilling Bit 0: CYCLE84 Unhide technology fields Bit 1: CYCLE840 Unhide technology fields Bit 2: Thread tapping also without encoder (ShopMill/ShopTurn) Bit 3: Constant cutting speed in rel. to diameter of centering (ShopMill/ShopTurn)	
52218	$MCS_FUNCTION_MASK_TURN	Function mask Turning Bit 0: Enable zoom under manual for tool measurement Bit 1: Enable parts gripper for cut-off Bit 2: Enable tailstock Bit 3: Enable spindle control of main spindle via surface Bit 4: Enable spindle control of counterspindle via surface Bit 5: Enable spindle control of tool spindle via surface Bit 6: Enable balance cutting for dual-channel stock removal Bit 7: Retraction during stock removal along contour with G1 Bit 8: Input spindle chuck data in the program Bit 9: Additional entry of tailstock data in the program Bit 10: Enable convex thread Bit 11: Reserved Bit 12: Deactivate thread synchronization Bit 13: Stock rem. contour with CYCLE95 (828D programGUIDE without Advanced Technology) Bit 14: Turning on milling machines: cont. transition between pole position and beta plus Bit 15: Enable counterspindle TM,inc (ShopTurn) Bit 16: Y turning with WCS rotation by Z +90° Bit 17: Y turning: WCS rotation selectable in the input screen Bit 18: Enable turning cycle masks on milling machines without turning technology	
52224	$MCS_FUNCTION_MASK_RECHNOLOGY_0	Technology function mask 0 Bit 0: Hide display of tools/tool functions Bit 1: Hide display of spindles/spindle functions Bit 2: Hide “Synchronous actions” softkey Bit 3: Hide “Retract” softkey Bit 4: Hide “Contour” softkey Bit 5: Hide “Diverse” softkey Bit 6: Hide “Simulation” softkey Bit 7: Hide “Simultaneous recording” softkey Bit 8: Hide “Mold making view” softkey	
52229	$MCS_ENABLE_QUICK_M_CODES	Enable fast M functions Bit 0: Coolant OFF Bit 1: Coolant 1 ON Bit 2: Coolant 2 ON Bit 3: Coolant 1 and 2 ON	
52230	$MCS_M_CODE_ALL_COOLANTS_OFF	M code for all coolants OFF	
52231	$MCS_M_CODE_COOLANT_1_ON	M code for coolant 1 ON	
52232	$MCS_M_CODE_COOLANT_2_ON	M code for coolant 2 ON	
52233	$MCS_M_CODE_COOLANT_1_AND_2_ON	M code for coolant 1 + 2 ON	

MD	Identifier	Description	Comment
52240	$MCS_NAME_TOOL_CHANGE_PROG	Tool change program for G code steps	
52241	$MCS_TOOL_CHANGE_POS_Y	Tool change position Y in the Machine (ShopTurn)	
52242	$MCS_TURN_TOOL_FIXING	Tool adapter for turning tools	
52244	$MCS_SUB_SPINDLE_PARK_POS_Y	Parking position of the Y axis with counter spindle	
52248	$MCS_REV_2_BORDER_TOOL_LENGTH	Limit value of tool length X for 2nd revolver: Limit value = 0: There is only one revolver Tool length < limit value: Tool belongs to 1st revolver/multifix Tool length >= limit value: Tool belongs to 2nd revolver/multifix	
52250	$MCS_M_CODE_CHUCK_OPEN	M code for Open chuck with non-rotating spindle. Example: "M34" or "M1=34" Elements: [0]: Main spindle [1]: Counterspindle	
52251	$MCS_M_CODE_CHUCK_CLOSE_OPEN_ROT	M code for Open chuck with rotating spindle. Example: "M35" or "M1=35" Elements: [0]: Main spindle [1]: Counterspindle	
52252	$MCS_M_CODE_CHUCK_CLOSE	M code for Close chuck Example: "M36" or "M1=36" Elements: [0]: Main spindle [1]: Counterspindle	
52253	$MCS_M_CODE_TAILSTOCK_FORWARD	M code for quill forward Example: "M55" or "M1=55" Elements: [0]: Quill opposite main spindle [1]: Quill opposite counterspindle	
52254	$MCS_M_CODE_TAILSTOCK_BACKWARD	M code for quill backward Example: "M54" or "M1=54" Elements: [0]: Quill opposite main spindle [1]: Quill opposite counterspindle	
52255	$MCS_Y_TURN_SWIVEL_ANG	Swivel angle B axis for Y-turning tools	
52260	$MCS_MACHINE_JOG_INTERRUPT_PRIO	Priory for start ASUP under machine JOG	

Channel Machine Data

MD	Identifier	Description	Comment
52270	$MCS_TM_FUNCTION_MASK	Function mask Tool management Bit 0: Do not permit Create tool on magazine location Bit 1: Load/unload disable when machine is not in reset Bit 2: Load/unload disable for emergency stop Bit 3: Load/unload tool or relocate tool to/from buffer disabled Bit 4: Load directly in spindle Bit 5: Use grinding configuration file Bit 6: Permit relocating to/from buffer despite disable (Bit3) Bit 7: Create tool using the T number Bit 8: Hide Relocate tool Bit 9: Hide Position magazine Bit 10: Reactivate tool using Position magazine Bit 11: Reactivate tool in all monitoring modes Bit 12: Fade out Reactivate tool	
52271	$MCS_TM_MAG_PLACE_DISTANCE	Distance between individual magazine locations	
52272	$MCS_TM_TOOL_LOAD_DEFAULT_MAG	Default magazine for tool loading 0 = No default magazine	
52273	$MCS_TM_TOOL_MOVE_DEFAULT_MAG	Default magazine for tool relocation 0 = No default magazine	
52274	$MCS_TM_TOOL_LOAD_STATION	Number of load station 0 = All configured stations are taken into account	
52281	$MCS_TOOL_MCODE_FUNC_ON	M code for tool-specific function ON	
52282	$MCS_TOOL_MCODE_FUNC_OFF	M code for tool-specific function OFF	
52290	$MCS_SIM_DISPLAY_CONFIG	Location of status display of the channel in the simulation (OP019 only) Bit 0: Corner left/top Bit 1: Corner right/top Bit 2: Corner left/bott Bit 3: Corner right/bott	
52441	$MCS_LIMIT_RES_PATH_SPEED	Limiting the resulting speed of the path motion	
52740	$MCS_MEA_FUNCTION_MASK	Function mask for measuring cycles, workpiece and tool measuring Bit 0: Workpiece probe connected at probe input 2 Bit 1: Workpiece, turning technology, measuring in the third geometry axis (Y) Bit 2: Calc. of correction angle with/without active orientation transformation Bit 3: Workpiece measuring, turning+milling in the third geometry axis (Y) in JOG Bit 16: Tool probe connected at probe input 2 Bit 17: Tool measurement turning, recording the cutting edge position	

Appendix B

Channel Machine Data

MD	Identifier	Description	Comment
52750	$MCS_J_MEA_FIXPOINT	Z value for measuring fixed point	
52751	$MCS_J_MEA_MAGN_GLAS_POS	Zoom position for tool measurement [0] = Position in 1st axis [1] = Position in 2nd axis	
52780	$MCS_J_MEA_T_PROBE_APPR_MODE	Approach mode tool measurement in JOG	
52800	$MCS_ISO_M_ENABLE_POLAR_COORD	Polar coordinates 0: OFF 1: ON	
52802	$MCS_ISO_ENABLE_INTERRUPTS	Interrupt process 0: OFF 1: ON	
52804	$MCS_ISO_ENABLE_DRYRUN	Machining skipped during tapping G74/G84 at DRY RUN 0: OFF 1: ON	
52806	$MCS_ISO_SCALING_SYSTEM	Basic system 0: not defined 1: METRIC 2: INCH	
52808	$MCS_ISO_SIMULTAN_AXES_START	Simultaneous approach to the boring position on all programmed axes 0: OFF 1: ON	
52810	$MCS_ISO_T_DEEPHOLE_DRILL_MODE	Select the type of deep hole drilling 0: deep hole drilling with chip breaking 1: deep hole drilling with stock removal	
52818	$MCS_ISO_M_FUNCTION_MASK	Function mask for ISO milling mode Bit 0: Reserved	
52819	$MCS_ISO_T_FUNCTION_MASK	Function mask for ISO turning mode Bit 0: Reserved	
52840	$MCS_GRIND_FUNCTION_MASK	Function mask for grinding Bit 7: Retraction while profiling with G1	
52842	$MCS_GRIND_DIAMETER_LENGTH	Number of diameter length	
52843	$MCS_GRIND_WIDTH_LENGTH	Number of grinding wheel with	

Axis Machine Data

MD	Identifier	Description	Comment
53030	$MAS_AXIS_MAX_POWER	Maximum value of axis power rating display	
53031	$MAS_AXIS_POWER_RANGE	Display range of axis power rating display	
53035	$MAS_SHOW_SPINDLE_UTIL_TIME	Display spindle service live	
53220	$MAS_AXIS_MCS_POSITION	Position of axis in the Machine	
53230	$MAS_SIM_START_POSITION	Axis position at start of simulation	
53240	$MAS_SPINDLE_PARAMETER	Spindle chuck data [0]: Chuck dimension [1]: Stop dimension [2]: Jaw dimension	
53241	$MAS_SPINDLE_CHUCK_TYPE	Spindle jaw type 0 = Clamp from outside 1 = Clamp from inside	
53242	$MAS_TAILSTOCK_PARAMETER	Tailstock data [0]: Tailstock diameter [1]: Tailstock length	
53250	$MAS_CLAMPING_TOLERANCE	Permissible tolerance when an axis jams	
53350	$MAS_ILC_BASE_VALUE	Basic interia of the axis	

General Setting Data

MD	Identifier	Description	Comment
54215	$SNS_TM_FUNCTION_MASK_SET	Function mask Tool management Bit 0: Diameter display for rotating tools Bit 1: Default direction of rotation for all turning tools is M4 Bit 2: Create tool without name suggestion Bit 3: Input disable for tool name and tool type in loaded tools Bit 4: Input disable for loaded tools, if the channel is not in Reset Bit 5: Calculate tool wear inputs additively Bit 6: Numerical input of the tool ident. Bit 7: Fade out tool monitoring parameters Bit 8: Diameter display for transverse axis - Geometry Bit 9: Diameter display for transverse axis - Wear Bit 10: Enable loading/relocating of tools on buffer locations Bit 11: Creating of new tools in gripper locations is disabled Bit 12: Do not unload measuring tools if ”Unload all” function is executed Bit 13: Do not delete tool wear when inputting the geometry value Bit 14: Load/relocate tool. Search without pre-assignment of the magazine	

Appendix B

General Setting Data

MD	Identifier	Description	Comment
54480	$SNS_AST_MMC_HANDLER_NAME	Assignment of an HMI-Operate instance for AST commands	
54481	$SNS_AST_MMC_DEFAULT_IS_PCU	Selection HMI-Operate internal (NCU) / external (PCU) for AST	
54600	$SNS_MEA_WP_BALL_DIAM	Effective diameter of the probe sphere for the workpiece probe	
54601	$SNS_MEA_WP_TRIG_MINUS_DIR_AX1	Trigger point of the 1st measuring axis in negative direction	
54602	$SNS_MEA_WP_TRIG_PLUS_DIR_AX1	Trigger point of the 1st measuring axis in positive direction	
54603	$SNS_MEA_WP_TRIG_MINUS_DIR_AX2	Trigger point of the 2nd measuring axis in negative direction	
54604	$SNS_MEA_WP_TRIG_PLUS_DIR_AX2	Trigger point of the 2ns measuring axis in positive direction	
54605	$SNS_MEA_WP_TRIG_MINUS_DIR_AX3	Trigger point of the 3rd measuring axis in negative direction	
54606	$SNS_MEA_WP_TRIG_PLUS_DIR_AX3	Trigger point of the 3d measuring axis in positive direction	
54607	$SNS_MEA_WP_POS_DEV_AX1	Position deviation of the probe sphere in the 1st measuring axis	
54608	$SNS_MEA_WP_POS_DEV_AX2	Position deviation of the probe sphere in the 2nd measuring axis	
54609	$SNS_MEA_WP_STATUS_RT	Calibration status axis positions, reserved for internal use!	
54610	$SNS_MEA_WP_STATUS_GEN	Calibration status in general, reserved for internal use!	
54611	$SNS_MEA_WP_FEED	Measuring feed for calibration	
54615	$SNS_MEA_CAL_EDGE_BASE_AX1	Calibration groove base of the 1st measuring axis	
54617	$SNS_MEA_CAL_EDGE_PLUS_DIR_AX1	Calibration groove edge in positive direction of the 1st measuring axis	
54618	$SNS_MEA_CAL_EDGE_MINUS_DIR_AX1	Calibration groove edge in negative direction of the 1st measuring axis	
54619	$SNS_MEA_CAL_EDGE_BASE_AX2	Calibration groove base of the 2nd measuring axis	

General Setting Data

MD	Identifier	Description	Comment
54620	$SNS_MEA_CAL_EDGE_UPPER_AX2	Calibration groove upper edge of the 2nd measuring axis	
54621	$SNS_MEA_CAL_EDGE_PLUS_DIR_AX2	Calibration groove edge in positive direction of the 2nd measuring axis	
54622	$SNS_MEA_CAL_EDGE_MINUS_DIR_AX2	Calibration groove edge in negative direction of the 2nd measuring axis	
54625	$SNS_MEA_TP_TRIG_MINUS_DIR_AX1	Trigger point of the 1st measuring axis in negative direction	
54626	$SNS_MEA_TP_TRIG_PLUS_DIR_AX1	Trigger point of the 1st measuring axis in positive direction	
54627	$SNS_MEA_TP_TRIG_MINUS_DIR_AX2	Trigger point of the 2nd measuring axis in negative direction	
54628	$SNS_MEA_TP_TRIG_PLUS_DIR_AX2	Trigger point of the 2nd measuring axis in positive direction	
54629	$SNS_MEA_TP_TRIG_MINUS_DIR_AX3	Trigger point of the 3rd measuring axis in negative direction	
54630	$SNS_MEA_TP_TRIG_PLUS_DIR_AX3	Trigger point of the 3rd measuring axis in positive direction	
54631	$SNS_MEA_TP_EDGE_DISK_SIZE	Tool probe edge length/wheel diameter	
54632	$SNS_MEA_TP_AX_DIR_AUTO_CAL	Enabling axes and traversing directions for "Automatic calibration" of milling tool probes. The default setting refers in X and Y to the plus and minus direction respectively, in Z only to the minus direction. The parameter is divided into three components the functions of which are to be assigned to calibration data records 1, 2 or 3! The calibration data records are firmly assigned to tool measuring in the working planes G17 (1), G18 (2) and G19 (3)! Meaning of the parameter components Decimal position: Ones: 1st geometry axis (X) Tens: 2nd geometry axis (Y) Hundreds: 3rd geometry axis (Z) Value: =0: Axis not enabled =1: Only minus direction possible =2: Only plus direction possible =3: Both directions possible	

Appendix B

General Setting Data

MD	Identifier	Description	Comment
54633	$SNS_MEA_TP_TYPE	Tool probe type cube / wheel 0: Cube 101: Wheel in XY, working plane G17 201: Wheel in ZX, working plane G18 301: Wheel in YZ, working plane G19	
54634	$SNS_MEA_TP_CAL_MEASURE_DEPTH	Distance between the upper tool probe edge and the lower milling tool edge	
54635	$SNS_MEA_TP_STATUS_GEN	Calibration status in general, reserved for internal use!	
54636	$SNS_MEA_TP_FEED	Measuring feed for tool probe calibration in MCS	
54640	$SNS_MEA_TPW_TRIG_MINUS_DIR_AX1	Trigger point of the 1st measuring axis in negative direction	
54641	$SNS_MEA_TPW_TRIG_PLUS_DIR_AX1	Trigger point of the 1st measuring axis in positive direction	
54642	$SNS_MEA_TPW_TRIG_MINUS_DIR_AX2	Trigger point of the 2nd measuring axis in negative direction	
54643	$SNS_MEA_TPW_TRIG_PLUS_DIR_AX2	Trigger point of the 2nd measuring axis in positive direction	
54644	$SNS_MEA_TPW_TRIG_MINUS_DIR_AX3	Trigger point of the 3rd measuring axis in negative direction	
54645	$SNS_MEA_TPW_TRIG_PLUS_DIR_AX3	Trigger point of the 3rd measuring axis in positive direction	
54646	$SNS_MEA_TPW_EDGE_DISK_SIZE	Tool probe edge length/wheel diameter	

General Setting Data

MD	Identifier	Description	Comment
54647	$SNS_MEA_TPW_AX_DIR_AUTO_CAL	Enabling axes and traversing directions for "Automatic calibration" of milling tool probes. The default setting refers in X and Y to the plus and minus direction respectively, in Z only to the minus direction. The parameter is divided into three components the functions of which are to be assigned to calibration data records 1, 2 or 3! The calibration data records are firmly assigned to tool measuring in the working planes G17 (1), G18 (2) and G19 (3)! Meaning of the parameter components Decimal position: Ones: 1st geometry axis (X) Tens: 2nd geometry axis (Y) Hundreds: 3rd geometry axis (Z) Value: =0: Axis not enabled =1: Only minus direction possible =2: Only plus direction possible =3: Both directions possible	
54648	$SNS_MEA_TPW_TYPE	Tool probe type cube / wheel 0: Cube 101: Wheel in XY, working plane G17 201: Wheel in ZX, working plane G18 301: Wheel in YZ, working plane G19	
54649	$SNS_MEA_TPW_CAL_MEASURE_DEPTH	Distance between the upper tool probe edge and the lower milling tool edge	
54650	$SNS_MEA_TPW_STATUS_GEN	Calibration status in general, reserved for internal use!	
54651	$SNS_MEA_TPW_FEED	Measuring feed for tool probe calibration in WCS	
54652	$SNS_MEA_INPUT_TOOL_PROBE_SUB	CNC measurement input for tool probe with reference to the counterspindle =0: Toolsetter no.=n+1, with reference to the main spindle CNC measurement input corresponds to the value of $MCS_MEA_FUNCTION_MASK, bit16 =10: Toolsetter no.=n+1, with reference to the counterspindle units digit =0, corresponds to CNC measurement input 1 =11: Toolsetter no.=n+1, with reference to the counterspindle units digit =1, corresponds to CNC measurement input 2	

Appendix B

General Setting Data

MD	Identifier	Description	Comment
54670	$SNS_MEA_CM_MAX_PERI_SPEED	Max. permissible peripheral speed of the tool to be measured	
54671	$SNS_MEA_CM_MAX_REVOLUTIONS	Maximum tool speed for tool measuring	
54672	$SNS_MEA_CM_MAX_FEEDRATE	Maximum feed for contact of the tool with the probe	
54673	$SNS_MEA_CM_MIN_FEEDRATE	Minimum feed for 1st contact of the tool with the probe	
54674	$SNS_MEA_CM_SPIND_ROT_DIR	Direction of spindle rotation for tool measuring	
54675	$SNS_MEA_CM_FEEDFACTOR_1	Feedrate factor 1, for tool measuring with rotating spindle =0: Single probing with the feedrate calculated by the cycle (but at least with the value of $SNS_MEA_CM_MIN_FEEDRATE) >=1: First probing with calculated feedrate (but at least with the value of $SNS_MEA_CM_MIN_FEEDRATE). Monitoring parameter for tool measuring with rotating spindle.	
54676	$SNS_MEA_CM_FEEDFACTOR_2	Feedrate factor 2, for tool measuring with rotating spindle =0:Second probing with the feedrate calculated by the cycle (only effective with MEA_CM_FEEDFACTOR_1 > 0) >=1: Second probing with calculated feedrate, feedrate factor 2 Third probing with calculated feedrate (tool speed is influenced by SD54749 $SNS_MEA_FUNCTION_MASK_TOOL, Bit 12) Notice: - Feedrate factor 2 should be smaller than feedrate factor 1! - If the value of feedrate factor 2 is 0, a third probing will not be not performed! Monitoring parameter for tool measuring with rotating spindle.	
54677	$SNS_MEA_CM_MEASURING_ACCURACY	Required measuring accuracy for tool measuring	
54689	$SNS_MEA_T_PROBE_MANUFACTURER	Tool probe type (manufacturer) These indications are required for tool measuring with rotating spindle. =0: No indication =1: TT130 (Heidenhain) =2: TS27R (Renishaw)	

General Setting Data

MD	Identifier	Description	Comment
54691	$SNS_MEA_T_PROBE_OFFSET	Measurement result offset for tool measuring with rotating spindle. =0: No offset =1: Cycle-internal offset (only effective with SD54689 $SNS_MEA_T_PROBE_MANUFACTURER<>0) =2: Offset through user-defined offset table	
54692	$SNS_MEA_T_CIRCULAR_ARC_DIST	Arc distance for single cutting edge measuring from the radius	
54693	$SNS_MEA_T_MAX_STEPS	Max. number of contacts for single cutting edge measuring from the radius	
54695	$SNS_MEA_RESULT_OFFSET_TAB_RAD1	Offset table (measure tool radius with rotating spindle)	
54696	$SNS_MEA_RESULT_OFFSET_TAB_RAD2	Offset table 1st peripheral speed (radius)	
54697	$SNS_MEA_RESULT_OFFSET_TAB_RAD3	Offset table 2nd peripheral speed (radius)	
54698	$SNS_MEA_RESULT_OFFSET_TAB_RAD4	Offset table 3rd peripheral speed (radius)	
54699	$SNS_MEA_RESULT_OFFSET_TAB_RAD5	Offset table 4th peripheral speed (radius)	
54700	$SNS_MEA_RESULT_OFFSET_TAB_RAD6	Offset table 5th peripheral speed (radius)	
54705	$SNS_MEA_RESULT_OFFSET_TAB_LEN1	Offset table (measure tool length with rotating spindle)	
54706	$SNS_MEA_RESULT_OFFSET_TAB_LEN2	Offset table 1st peripheral speed (length)	
54707	$SNS_MEA_RESULT_OFFSET_TAB_LEN3	Offset table 2nd peripheral speed (length)	
54708	$SNS_MEA_RESULT_OFFSET_TAB_LEN4	Offset table 3rd peripheral speed (length)	
54709	$SNS_MEA_RESULT_OFFSET_TAB_LEN5	Offset table 4th peripheral speed (length)	
54710	$SNS_MEA_RESULT_OFFSET_TAB_LEN6	Offset table 5th peripheral speed (length)	

Appendix B

General Setting Data

MD	Identifier	Description	Comment
54740	$SNS_MEA_FUNCTION_MASK	Function mask for measuring cycles, workpiece and tool measuring Bit 0: Workpiece, measurement repeated if TSA, TDIF are exceeded Bit 1: Workpiece, repeat measurement with M0 if TSA, TDIF are exceeded Bit 2: Workpiece, stop at M0 if TUL, TLL are exceeded Bit 3: Workpiece, accept calibrated probe ball radius in tool data Bit 4: Workpiece , plane between calibration and measurement can be different Bit 5: Workpiece , use calibration data set from tool Bit 6: Workpiece , corner meas. polar, do not consider start angle for meas. path Bit 16: Tool, repeat measurement if TDIF, TSA are exceeded Bit 17: Tool, repeat measurement with stop at M0 if TDIF, TSA are exceeded Bit 19: Tool milling, reduction of the spindle speed at the last probing	
54750	$SNS_MEA_ALARM_MASK	Expert mode for cycle alarms Bit 2: Workpiece, deactivate measuring cycle alarms, traversing distance reduction Bit 3: Workpiece, measuring cycle alarm JOG mode when calibration status is reset Bit 4: Workpiece, expanded error reaction Bit 5: Workpiece, expanded error reaction with handling alarm output Bit 16: Tool, extended error reaction Bit 17: Tool, extended error reaction with the action alarm output	
54760	$SNS_MEA_FUNCTION_MASK_PIECE	Setting for input screen, measuring cycles in Automatic, workpiece measuring Bit 1: Selection of display softkey 3D measurement Bit 3: Select probe calibration data field, enable Bit 4: Select Calibrate Input measuring feed Bit 6: Enable selection of WO compensation in basic reference Bit 7: Enable selection of WO compensation in channel-specific basic WO Bit 8: Enable selection of WO compensation in global basic WO Bit 9: Enable selection of WO compensation in work offset Bit 10: Enable selection of WO compensation in course and fine Bit 11: Enable selection of tool offset wear and geometry Bit 12: Enable selection of tool offset not inverted and inverted Bit 13: Enable selection of tool offset radius, length L1, L2, L3 Bit 14: Enable selection of tool offset work offset Bit 15: Enable selection of tool offset dimensional difference check Bit 16: Selection of workpiece measurement with spindle reversal Bit 17: Selection align workpiece probe in switching direction Bit 18: Select Number of measurements Bit 19: Select Offset with averaging Bit 20: Select Empirical values Bit 21: Select Additive setup offset Bit 22: Select Calibrate to unknown or known center point Bit 24: Select Calibrate with or without positional deviation Bit 25: Selection zero offset when measuring the angularity of the spindle Bit 26: Tool offset selection not enabled Bit 27: Select tolerance for linear vectors with kinematic measurement not enabled Bit 28: Select adapt tool length, enable Bit 29: Select MEC tables, enable	

General Setting Data

MD	Identifier	Description	Comment
54762	$SNS_MEA_FUNCTION_MASK_TOOL	Setting for input screen, measuring cycles in Automatic, tool measuring Bit 3: Select probe calibration data field, enable Bit 4: Select Calibrate Input measuring feed Bit 5: Select Input feed and spindle speeds for contacting Bit 7: Select Measure in MCS and WCS Bit 8: Select Measure absolute and incremental Bit 9: Select Tool offset Geometry and wear Bit 10: Select measuring function ”Check teeth individually” Bit 11: Selection spindle reversal when calibrating on the plane Bit 12: Select Number of measurements Bit 13: Select Empirical values Bit 14: Select offset correction with selection of direction	
54764	$SNS_MEA_FUNCTION_MASK_TURN	Setting for input mask, Measuring cycles in Automatic, Workpiece turning Bit 0: Measure Diameter inside/outside with reversal Bit 1: Measure Diameter inside/outside “Travel under turning center“?	
54780	$SNS_J_MEA_FUNCTION_MASK_PIECE	Setting for input screen, Workpiece measurement in JOG Bit 2: Activate calibration for electronic workpiece probe Bit 3: Select probe calibration data field, enable Bit 5: Select WO as measuring basis Bit 6: Select WO compensation in basic reference, enable Bit 7: Select WO compensation in channel basic frame, enable Bit 8: Select WO compensation in global basic frame, enable Bit 9: Select WO compensation in settable frame, enable Bit 16: Selection of workpiece measurement with spindle reversal Bit 17: Selection align workpiece probe in switching direction Bit 24: Select Calibrate with or without positional deactivation Bit 28: Select adapt tool length, enable	
54782	$SNS_J_MEA_FUNCTION_MASK_TOOL	Setting for input screen, Tool measuring in JOG Bit 2: Enable automatic tool measurement Bit 3: Select tool probe calibration data field, enable Bit 10: Select measuring function ”Check teeth individually” Bit 11: Selection spindle reversal when calibrating on the plane	
54784	$SNS_J_MEA_FUNCTION_MASK_TURN	Setting for input screen, Tool measuring in JOG, workpiece turning Bit 0 Measure Diameter inside/outside with reversal Bit 1: Measure Diameter inside/outside Travel “under turning center”	

Appendix B

Channel Setting Data

MD	Identifier	Description	Comment
55200	$SCS_MAX_INP_FEED_PER_REV	Feedrate input upper limit for mm/rev	
55201	$SCS_MAX_INP_FEED_PER_TIME	Feedrate input upper limit for mm/min	
55202	$SCS_MAX_INP_FEED_PER_TOOTH	Feedrate input upper limit for mm/tooth	
55212	$SCS_FUNCTION_MASK_TECH_SET	Function mask Cross-technology Bit 0: Tool preselection active Bit 1: Calculate thread depth from metric thread lead Bit 2: Refer to the table for thread diameter and depth Bit 3: Delete programs generated by contour cycles	
55214	$SCS_FUNCTION_MASK_MILL_SET	Function mask Milling Bit 0: Default setting - milling cycles with synchronous operation Bit 2: Depth calculation in milling cycles without parameter SC = 0: Depth calculation in milling cycles is carried out between reference plane + safety clearance and depth = 1: Depth calculation in milling cycles is carried out without inclusion of the safety clearance Note: Bit 2 is active with the following milling cycles: CYCLE61, CYCLE71, CYCLE76, CYCLE77, CYCLE79, CYCLE899, LONGHOLE, SLOT1, SLOT2, POCKET3, POCKET4 Bit 3: Deactivation of alarm 61012 different scaling in the planet	
55216	$SCS_FUNCTION_MASK_DRILL_SET	Function mask Drilling Bit 0: Reverse direction of spindle rotation during tapping Bit 1: Consider rotation of the tool plane (CYCLE86) Bit 2: Consider swivelled table kinematics (CYCLE86) Bit 3: Monitoring spindle MD 31050/31060 (CYCLE84) Bit 4: Monitoring spindle MD 31050/31060 (CYCLE840) Bit 5: reserved Bit 6: Adapt spindle position to the tool orientation (CYCLE86) Bit 7: Adapt direction of retraction path in the plane to active mirroring (CYCLE86)	
55218	$SCS_FUNCTION_MASK_TURN_SET	Function mask Turning Bit 0: New thread table during thread cutting Bit 1: Reserved (CYCLE93) Bit 2: Reserved (CYCLE93) Bit 3: Chamfer in CYCLE930 as chamfer length (CHF) Bit 4: Retraction from the inside machining (CYCLE951) Bit 5: Disable new smoothing behaviour in thread cutting cycles	
55220	$SCS_FUNCTION_MASK_MILL_TOL_SET	Function mask High Speed Settings CYCLE832 Bit 0: Unhide technology fields Bit 1: Orientation tolerance (OTOL) as a tolerance factor, enter it directly Bit 2: Mould-making function can be selected in High Speed Settings mask Bit 3: High Speed Settings mask with Advanced Surface option Bit 4: Display smoothing selection Bit 5: Suppression alarm61147 Bit 6: Selection standard even without option	

Channel Setting Data

MD	Identifier	Description	Comment
55221	$SCS_FUNCTION_MASK_SWIVEL_SET	Function mask Swivel CYCLE800 Bit 0: Unhide input field "No swivel" Bit 1: Selection text for retraction, “fixed point 1” or “fixed point 2” Bit 2: Show swivel data block "Deselection" Bit 3: Display active swivel plane in JOG Bit 4: Positioning in basic position (pole position) of the kinematics Bit 5: Align tool, absolute (TCOABS) Bit 6: Do not offer swivel mode “directly” under swivel in JOG Bit 7: Swivel plane: Direction selection in initial setting of kinematics Bit 8: Take into account work offset of rotary axes, swivel data set in Work Bit 9: Swivel data set TC permanently assigned to the tool Bit 10: Swivel plane: Display input field “positioning direction” Bit 11: Swivel plane: Positioning of both solutions AB - display kinematics Bit 12: Reset behaviour taking into account the rotary axis position Bit 13: Extended swivel function in pole position	
55230	$SCS_CIRCLE_RAPID_FEED	Positional feed on circular path	
55232	$SCS_SUB_SPINDLE_REL_POS	Z retraction position for the counter spindle	
55260	$SCS_MAJOG_SAFETY_CLEARANCE	Safety clearance for machine JOG	
55261	$SCS_MAJOG_RELEASE_PLANE	Retraction plane for machine JOG	
55300	$SCS_EASY_SAFETY_CLEARANCE	Simple input: Safety clearance	
55301	$SCS_EASY_DWELL_TIME	Simple input: dwell time	
55305	$SCS_EASY_DRILL_DEEP_FD1	Simple input: percentage 1st federate deep - hole drilling	
55306	$SCS_EASY_DRILL_DEEP_DF	Simple input: percentage infeed deep - hole drilling	
55307	$SCS_EASY_DRILL_DEEP_V1	Simple input: min. depth infeed deep - hole drilling	
55308	$SCS_EASY_DRILL_DEEP_V2	Simple input: absolute retraction value deep - hole drilling	
55309	$SCS_EASY_THREAD_RETURN_DIST	Simple input: return distance thread tapping	
55400	$SCS_MILL_ENGRAVE_POINT_RAD	Engraving cycle CYCLE60: Circ. path radius for generating the char. ”point”	

Appendix B

Channel Setting Data

MD	Identifier	Description	Comment
55410	$SCS_MILL_SWIVEL_ALARM_MASK	Hide and unhide cycle alarms Bit 0: Display alarm 62186: Active work offset G54 ff. and base (basic reference) contain rotations Bit 1: Display alarm 62187: Active base and basic reference (G500) contain rotations Bit 2: Display alarm 61148: Swivel plane not possible with active turning tool Bit 3: Display alarm 61426: Kinematics measurement with CYCLE9960/996 Bit 4 Display alarm 61019: Swivel plane, the parameters _DIR and _ST are checked	
55420	$SCS_MILL_SWIVEL_RESET_RETRACT	Initial setting swivel: retract = 0: no change = 1: no = 2: Z = 3: Z XY = 4: Tool direction max. = 5: Tool direction inc.	
55421	$SCS_MILL_SWIVEL_RESET_TRACK	Initial setting swivel: tool correction = 0: no change = 1: do not correct = 2: correct	
55422	$SCS_MILL_SWIVEL_RESET_MODE	Initial setting swivel: swivel mode = 0: axis by axis = 1: direct	
55423	$SCS_MILL_SWIVEL_RESET_SEQ_AXIS	Initial setting swivel: axis sequence = 0: X Y Z = 1: X Z Y = 2: Y X Z = 3: Y Z X = 4: Z X Y = 5: Z Y X	
55441	$SCS_MILL_TOL_FACTOR_ROUGH	Rotary axes tolerance factor for roughing CYCLE832 of G group 59	
55442	$SCS_MILL_TOL_FACTOR_SEMIFIN	Rotary axes tolerance factor for pre-finishing CYCLE832 of G group 59	
55443	$SCS_MILL_TOL_FACTOR_FINISH	Rotary axes tolerance factor for finishing CYCLE832 of G group 59	
55444	$SCS_MILL_TOL_FACTOR_PRECISION	Rotary axes tolerance factor for smooth finishing CYCLE832 of G group 59	
55446	$SCS_MILL_TOL_VALUE_ROUGH	Tolerance value for roughing CYCLE832 (High Speed Settings)	
55447	$SCS_MILL_TOL_VALUE_SEMIFIN	Tolerance value for pre-finishing CYCLE832 (High Speed Settings)	
55448	$SCS_MILL_TOL_VALUE_FINISH	Tolerance value for finishing CYCLE832 (High Speed Settings)	
55449	$SCS_MILL_TOL_VALUE_PRECISION	Tolerance value for smooth finishing CYCLE832 (High Speed Settings)	

Channel Setting Data

MD	Identifier	Description	Comment
55451	$SCS_MILL_ORI_TOL_ROUGH	Orientation tolerance for roughing CYCLE832 (High Speed Settings)	
55452	$SCS_MILL_ORI_TOL_SEMIFIN	Orientation tolerance for pre-finishing CYCLE832 (High Speed Settings)	
55453	$SCS_MILL_ORI_TOL_FINISH	Orientation tolerance for finishing CYCLE832 (High Speed Settings)	
55454	$SCS_MILL_ORI_TOL_PRECISION	Orientation tolerance for smooth finishing CYCLE832 (High Speed Settings)	
55456	$SCS_STD_TOL_FACTOR	Rotary axes tolerance factor for standard machining	
55457	$SCS_STD_TOL_ROUGH	Tolerance value contour tolerance for roughing standard	
55458	$SCS_STD_TOL_FINISH	Tolerance value contour tolerance for finishing standard	
55460	$SCS_MILL_CONT_INITIAL_RAD_FIN	Contour pocket milling: approach circle radius finishing	
55481	$SCS_DRILL_TAPPING_SET_GG12	Settings for tapping G group 12 cycle CYCLE84 and CYCLE840: G group 12: Block change behaviour at exact stop (G60) 0 = Exact stop behaviour, same as before the cycle call 1 = G601 2 = G602 3 = G603 Index[0]: for tool spindle Index[1]: for main spindle - for drilling in centre	
55482	$SCS_DRILL_TAPPING_SET_GG21	Settings for tapping G group 21 cycle CYCLE84: G group 21: Acceleration profile (SOFT, BRISK, ..) 0 = Acceleration behaviour, same as before the cycle call 1 = SOFT 2 = BRISK 3 = DRIVE Index[0]: for tool spindle Index[1]: for main spindle - for drilling in centre	
55483	$SCS_DRILL_TAPPING_SET_GG24	Settings for tapping G group 24 cycle CYCLE84 and CYCLE840: G group 24: Feedforward control (FFWON, FFWOF) 0 = Feedforward control, same as before the cycle call 1 = FFWON 2 = FFWOF Index[0]: for tool spindle Index[1]: for main spindle - for drilling in centre	
55484	$SCS_DRILL_TAPPING_SET_MC	Setting for tapping cycle CYCLE84 spindle operation at MCALL 0= Reactivate spindle operation at MCALL 1= Maintain position-controlled spindle operation at MCALL Index[0]: for tool spindle Index[1]: for main spindle - for drilling in centre	

Appendix B

Channel Setting Data

MD	Identifier	Description	Comment
55489	$SCS_DRILL_MID_MAX_ECCENT	Maximum centre offset for centre drilling	
55490	$SCS_DRILL_SPOT_DIST	Rough drilling depth for drill and thread milling	
55500	$SCS_TURN_FIN_FEED_PERCENT	Roughing feedrate for complete machining in %	
55505	$SCS_TURN_ROUGH_O_RELEASE_DIST	Return distance stock removal for external machining -1: The distance is specified internally.	
55506	$SCS_TURN_ROUGH_I_RELEASE_DIST	Return distance stock removal for internal machining -1: The distance is specified internally.	
55510	$SCS_TURN_GROOVE_DWELL_TIME	Tool clearance time for grooving at the base (neg. value=revolutions) - Negative value in spindle revolutions - Positive value in seconds	
55540	$SCS_TURN_PART_OFF_CTRL_DIST	Path for cut-off check	
55541	$SCS_TURN_PART_OFF_CTRL_FEED	Feedrate for cut-off check	
55542	$SCS_TURN_PART_OFF_CTRL_FORCE	Force for cut-off check, in %	
55543	$SCS_TURN_PART_OFF_RETRACTION	Retraction path prior to cut-off with counter spindle	
55550	$SCS_TURN_FIXED_STOP_DIST	Counterspindle: path for travel to fixed stop (see setting data 55551 $SCS_TURN_FIXED_STOP_FEED).	
55551	$SCS_TURN_FIXED_STOP_FEED	Counterspindle: feedrate for travel to fixed stop In setting data 55550 $SCS_TURN_FIXED_STOP_DIST you specify the distance after which the tool travels in this feedrate.	
55552	$SCS_TURN_FIXED_STOP_FORCE	Counterspindle: force for travel to fixed stop, in %	
55553	$SCS_TURN_FIXED_STOP_RETRACTION	Counterspindle: retraction path prior to chucking after fixed stop	
55580	$SCS_TURN_CONT_RELEASE_ANGLE	Counter turning: retraction angle	
55581	$SCS_TURN_CONT_RELEASE_DIST	Counter turning: retraction value	
55582	$SCS_TURN_CONT_TRACE_ANGLE	Counter turning: minimum angle for rounding along contour	

Channel Setting Data

MD	Identifier	Description	Comment
55583	$SCS_TURN_CONT_VARIABLE_DEPTH	Counter turning: percentage for variable cutting depth	
55584	$SCS_TURN_CONT_BLANK_OFFSET	Counter turning: blank allowance	
55585	$SCS_TURN_CONT_INTERRUPT_TIME	Counter turning: feed interrupt time (neg. value=revolutions) - Negative value in spindle revolutions - Positive value in seconds This setting data is effective only if setting data 55586 is 55586 $SCS_TURN_CONT_INTER_RETRACTION = 0.	
55586	$SCS_TURN_CONT_INTER_RETRACTION	Counter turning: retraction path after feed interrupt >0: Retraction path after feed interrupt (setting data 55585 $SCS_TURN_CONT_INTERRUPT_TIME is ineffective!) =0: No retraction path	
55587	$SCS_TURN_CONT_MIN_REST_MAT_AX1	Counter turning: minimum difference dimension residual machining axis 1 Example: If this MD is set to 50% and if the finishing allowance is 0.5 mm, the residual material which is thinner than 0.25 mm is not removed in a separate machining step, but during finishing.	
55588	$SCS_TURN_CONT_MIN_REST_MAT_AX2	Counter turning: minimum difference dimension residual machining axis 2 Example: If this MD is set to 50% and if the finishing allowance is 0.5 mm, the residual material which is thinner than 0.25 mm is not removed in a separate machining step, but during finishing.	
55595	$SCS_TURN_CONT_TOOL_BEND_RETR	Counter plunge turning: retraction path due to tool bending In this setting data you specify the lateral distance to the last cut, by which each next cut will be shortened.	
55596	$SCS_TURN_CONT_TURN_RETRACTION	Counter plunge turning: retraction path prior to turning	

Appendix B

Channel Setting Data

MD	Identifier	Description	Comment
55613	$SCS_MEA_RESULT_DISPLAY	Selection of measurement result display =0: No measurement result display =1: The measurement result is displayed until cycle end =2: The measurement result is displayed until the start/call of the next cycle =3: With the measurement result display, the cycle is stopped by an internal M0, with NC start, the measuring cycle is continued and the measurement result display is deselected =4: The measurement result display is only shown for cycle alarms 61303, 61304, 61305, 61306 Same behavior as with setting 3!	
55614	$SCS_MEA_RESULT_MRD	Program control of the display of the measurement result display MRD =0: Hide measurement result display =1: Show measurement result display	
55618	$SCS_MEA_SIM_ENABLE	Setting measuring cycles under a simulated environment = 0: Measuring cycles are not executed (measuring cycle is skipped internally) = 1: Measuring cycles are executed; real axes are required! During calibration no values are entered in the probe data fields, no measurement result is displayed, the measuring cycle is not logged, the travel is performed without collision detection.	
55619	$SCS_MEA_SIM_MEASURE_DIFF	Value for simulated error of measurement Provided that SD55618 $SCS_MEA_SIM_ENABLE=1 is used and that the measuring cycles are executed in a simulated environment of HMI Advanced or ShopMill / ShopTurn, a measurement difference can be entered in this parameter. The value of the measurement difference must be smaller than the measuring path in parameter _FA! Otherwise cycle alarm 61301 "Probe does not switch" is output during active simulation.	
55622	$SCS_MEA_EMPIRIC_VALUE_NUM	Number of empirical values	
55623	$SCS_MEA_EMPIRIC_VALUE	Empirical value memory Using parameter $SCS_MEA_EMPIRIC_VALUE_NUM the number of memory elements can be defined! Currently, however, these 20 memory elements can-not be changed! In the empirical value memory, empirical values can be stored which are cleared with the currently calculated difference between the setpoint and the actual value. Using parameter _EVNUM the empirical value element to be cleared is addressed!	
55624	$SCS_MEA_AVERAGE_VALUE_NUM	Number of mean values	

Channel Setting Data

MD	Identifier	Description	Comment
55625	$SCS_MEA_AVERAGE_VALUE	Mean value memory Using parameter $SCS_MEA_AVERAGE_VALUE_NUM the number of memory elements can be defined! Currently, however, these 20 memory elements can-not be changed! In the mean value memory, the mean values calculated in connection with functionality "Automatic tool offset with mean value creation" are stored. Using parameter _EVNUM the mean value element to be used is addressed!	
55628	$SCS_MEA_TP_FEED_MEASURE	Feed for calibrating a tool probe	
55630	$SCS_MEA_FEED_MEASURE	Feed for calibrating a workpiece probe	
55631	$SCS_MEA_FEED_MEASURE_DEG	Measurement feedrate of workpiece probe during positioning of a rotary axis	
55632	$SCS_MEA_FEED_RAPID IN_PERCENT	Rapid traverse velocity in per cent for intermediate positioning	
55634	$SCS_MEA_FEED_PLANE_VALUE	Traverse velocity for intermediate positioning in the plane	
55636	$SCS_MEA_FEED_FEEDAX_VALUE	Positioning velocity in the infeed axis	
55637	$SCS_MEA_FEED_POS_DEG	Feedrate when positioning a rotary axis between individual measurements	
55638	$SCS_MEA_FEED_FAST_MEASURE	Rapid measuring feed The use of "Rapid measuring feed" depends of SD55740 $SCS_MEA_FUNCTION_MASK Bit4!	
55640	$SCS_MEA_FEED_CIRCLE	Circular feed for measuring circle segment	
55642	$SCS_MEA_EDGE_SAVE_ANG	Additional safe angle for measuring corner Set $SCS_MEA_EDGE_SAVE_ANG=10 for compatibility programs.	
55644	$SCS_MEA_KIN_DM_TOL	Measure kinematics: Tolerance of diameter of calibration ball	
55645	$SCS_MEA_KIN_MODE	Free component of the linear vector	
55646	$SCS_MEA_KIN_VALUE	Measure kinematics completely: value of the linear vector	

Appendix B

Channel Setting Data

MD	Identifier	Description	Comment
55647	$SCS_MEA_KIN_MIN_ANG_TRIANGLE	Minimum interior angle of the measurement triangle	
55648	$SCS_MEA_KIN_MIN_ANG_POS	Minimum angle when positioning a rotary axis	
55649	$SCS_MEA_KIN_BALL_VEC	Perform complete kinematics measurement: Vector of the ball mounting	
55650	$SCS_MEA_KIN_PREPOS_RA	Measure kinematics: Angle for prepositioning the rotary axis	
55700	$SCS_MEA_SIMULTAN_LIMIT	Simultaneous measuring, overrun travel	
55701	$SCS_MEA_SIMULTAN_POS_DEV_MAX	Simultaneous measurement, maximum position deviation between the probes	
55730	$SCS_MEA_PROTOCOL_USER_EXT	File extension for application protocol	
55740	$SCS_MEA_FUNCTION_MASK	Function mask for measuring cycles, workpiece and tool measuring Bit 0: Workpiece, collision monitoring for intermediate positioning Bit 1: Workpiece, coupling spindle with coordinate rotation around infeed axis Bit 2: Workpiece, for coupling of spindle pos., reverse positioning direction Bit 3: Workpiece, do not repeat if the probe does not switch Bit 4: Workpiece, 1st probing for each measurement with fast measuring feedrate Bit 5: Workpiece, fast retraction after measurement Bit 6: Workpiece, deactivate probe during spindle positioning Bit 7: Kinematics, normalization based on the input values of rotary axis vectors Bit 8: Kinematics, measurement with active swivel (TCARR) or active TRAORI Bit 9: Kinematics, user-specific scaling of rotary axis vectors Bit 10: Complete kinematics measurement with reference Bit 11: Kinematics, number of measuring points for measuring the calibration ball Bit 12: Kinematics, change cancel alarms for tolerance monitoring to reset alarms Bit 13: Kinematics, scaling/set fixed value on closing vector Bit 14: Kinematics, VCS-Rotary compensation table is not deactivated Bit 15: Workpiece, calib. Radius with start point at venter of ring, only in JOG Bit 16: Tool, collision monitoring for intermediate positioning Bit 17: Tool, do not repeat if the probe does not switch Bit 19: Tool, fast retraction after measuring Bit 20: Reserved Bit 21: Kinematics, calculation pivot point circle equation (least squares error) Bit 22: Kinematics, offset calculation average for head axes Bit 23: Kinematics, generation file for encoder compensation rotary axis Bit 24: Kinematics, no confirmation of change by user Bit 28: Continuous standard protocol for measurement in JOG (append) Bit 29: Table format for standard protocol for measurement in JOG	
55774	$SCS_J_MEA_PROTOCOL_FILE	Name and path of protocol file for measurement in JOG	

Channel Setting Data

MD	Identifier	Description	Comment
55800	$SCS_ISO_M_DRILLING_AXIS_IS_Z	Selection of the drilling axis 0: Drilling axis is vertical to the active plane 1: Drilling axis is always "Z", independently of the active plane	
55802	$SCS_ISO_M_DRILLING_TYPE	Tapping type 0: Tapping without compensating chuck 1: Tapping with compensating chuck 2: Deep hole tapping with chip breakage 3: Deep hole tapping with stock removal	
55804	$SCS_ISO_M_RETRACTION_FACTOR	Factor for retraction speed (0...200%)	
55806	$SCS_ISO_M_RETRACTION_DIR	Retraction direction for precision drilling and reverse counter-sinking G76/G87 0: G17(-X) G18(-Z) G19(-Y) 1: G17(+X) G18(+Z) G19(+Y) 2: G17(-X) G18(-Z) G19(-Y) 3: G17(+Y) G18(+X) G19(+Z) 4: G17(-Y) G18(-X) G19(-Z)	
55807	$SCS_ISO_M_TAPPING_SET_MC	Setting tapping G84: Spindle mode/position control	
55808	$SCS_ISO_T_RETRACTION_FACTOR	Factor (1-200%) for retraction speed at tapping G84/G88	
55810	$SCS_ISO_T_DWELL_TIME_UNIT	Dwell time evaluation for deep hole drilling G83/G87 0: Seconds 1: Revolutions	
55818	$SCS_ISO_M_FUNCTION_MASK	Function mask for ISO milling mode Bit 0: Reserved	
55819	$SCS_ISO_T_FUNCTION_MASK	Function mask for ISO turning mode Bit 0: Without complete residual corner cut Bit 1: Stock removal roughing without tool nose radius compensation	

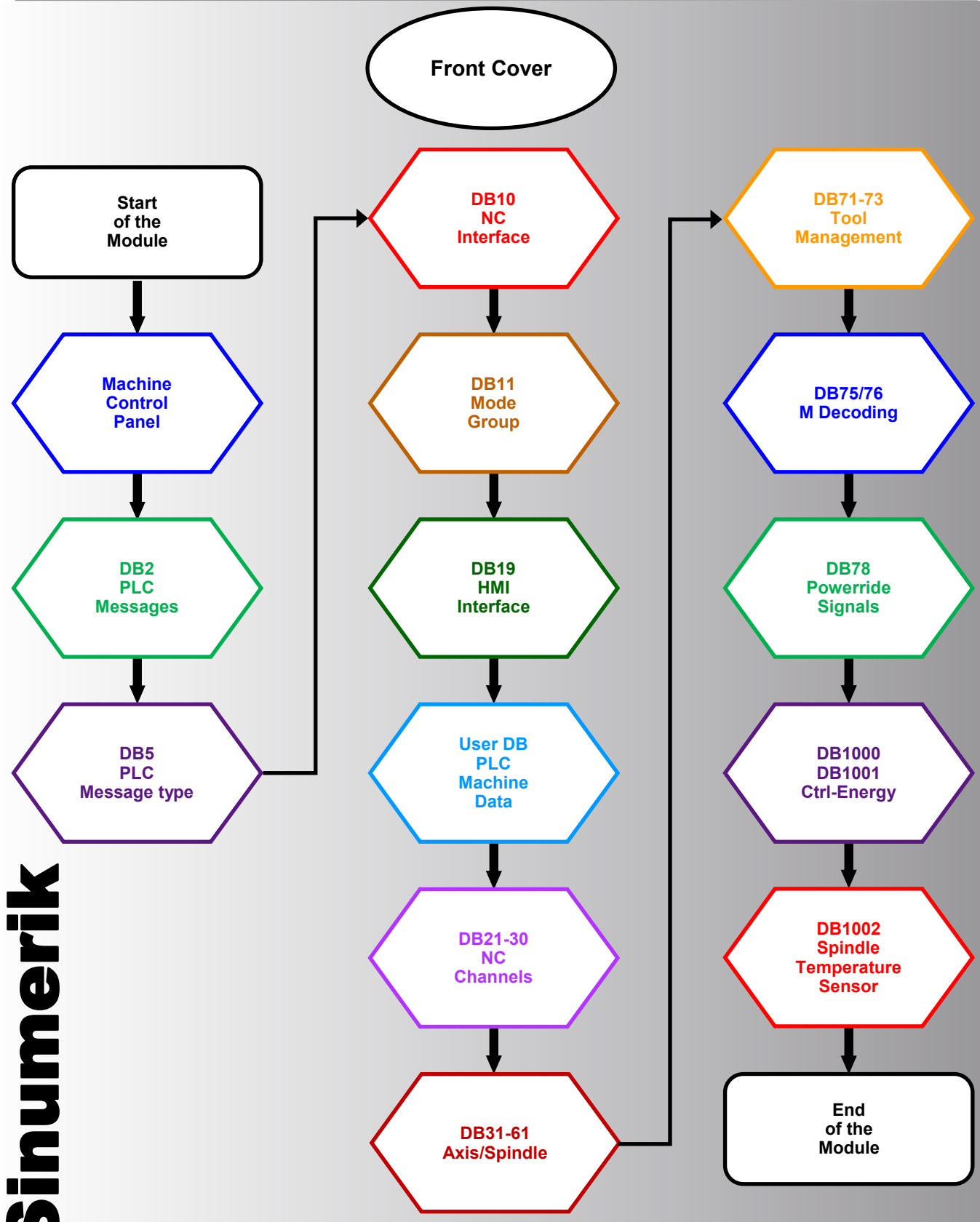
Appendix B

Channel Setting Data

MD	Identifier	Description	Comment
55820	$SCS_FRICT_OPT_RADIUS	Circle radius for optimizing friction compensation	
55821	$SCS_FRICT_OPT_RADIUS_ROT	Circle radius for optimizing friction compensation for rotary axes	
55822	$SCS_FRICT_OPT_FEED	Feedrates for optimizing friction compensation	
55823	$SCS_FRICT_OPT_FEED_ROT	Feedrates for optimizing friction compensation for rotary axes	
55824	$SCS_FRICT_OPT_STEP	Step number for optimizing friction compensation	
55826	$SCS_FRICT_OPT_ACT_STEP	Current step number for optimizing friction compensation	
55828	$SCS_FRICT_OPT_DIR_MINUS	Direction of rotation for optimizing friction compensation	
55844	$SCS_GRIND_MEA_KIN_TOL	Kinematic measurement: Tolerance of kinematic vectors	
55880	$SCS_GRIND_CONT_RELEASE_ANGLE	Grinding - profiling: retraction angle	
55881	$SCS_GRIND_CONT_RELEASE_DIST	Grinding - profiling: retraction value	
55884	$SCS_GRIND_CONT_BLANK_OFFSET	Grinding - profiling: blank allowance	

Appendix B: END

PLC Interface



This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

Machine Control Panel

Signals from/to Machine Control Panel

Machine control panel, M version

Signals from MCP (keys) Input image								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Spindle speed override				Operating mode			
	D	C	B	A	JOG	TEACH IN	MDA	AUTO
IB n + 1	Machine function							
	REPOS	REF	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
IB n + 2	Key-switch position 0	Key-switch position 2	Spindle start	*Spindle stop	Feed start	*Feed stop	NC Start	*NC Stop
IB n + 3	Reset	Key-switch position 1	Single block	Feedrate override				
				E	D	C	B	A
IB n + 4	Direction keys			Keyswitch position 3	Direction keys			
	plus R15	minus R13	Rapid traverse R14		X R1	4th axis R4	7th axis R7	R10
IB n + 5	Axis selection							
	Y R2	Z R3	5th axis R5	Traverse command MCS/WCS	R11	9th axis R9	8th axis R8	6th axis R6
IB n + 6	Unassigned customer keys							
	T9	T10	T11	T12	T13	T14	T15	
IB n + 7	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

Notes:

Signals to MCP (LEDs) Output image								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 0	Machine function				Operating mode			
	1000 INC	100 INC	10 INC	1 INC	JOG	TEACH IN	MDA	AUTO
QB n + 1	Feed start	Feed stop	NC Start	NC Stop	Machine function			
					REPOS	REF	Var. INC	10000 INC
QB n + 2	Direction key minus R13	Axis selection				Single block	Spindle start	Spindle stop
		X R1	4th axis R4	7th axis R7	R10			
QB n + 3	Axis selection							Direction key plus R15
	Z R3	5th axis R5	Travel com-mand MCS/WCS R12	R11	9th axis R9	8th axis R8	6th axis R6	
QB n + 4	Unassigned customer keys							Y R2
	T9	T10	T11	T12	T13	T14	T15	
QB n + 5	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

Notes:

Machine control panel, T version

Signals from MCP (keys) Input image								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Spindle speed override				Operating mode			
	D	C	B	A	JOG	TEACH IN	MDA	AUTO
IB n + 1	Machine function							
	REPOS	REF	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
IB n + 2	Keyswitch position 0	Keyswitch position 2	Spindle start	*Spindle stop	Feed start	*Feed stop	NC Start	*NC Stop
IB n + 3	Reset	Keyswitch position 1	Single block	Feed override				
				E	D	C	B	A
IB n + 4	Direction keys			Keyswitch position 3	Direction keys			
	R15	R13	R14		+Y R1	-Z R4	-C R7	R10
IB n + 5	Direction keys							
	+X R2	+C R3	Rapid traverse override R5	Travel command MCS/ WCS R12	R11	-Y R9	-X R8	+Z R6
IB n + 6	Unassigned customer keys							
	T9	T10	T11	T12	T13	T14	T15	
IB n + 7	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

Signals to MCP (LEDs) Output image								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 0	Machine function				Operating mode			
	1000 INC	100 INC	10 INC	1 INC	JOG	TEACH IN	MDA	AUTO
QB n + 1	Feed start	Feed stop	NC Start	NC Stop	Machine function			
					REPOS	REF	Var. INC	10000 INC
QB n + 2	Direction keys					Single block	Spindle start	Spindle stop
	R13	+Y R1	-Z R4	-C R7	R10			
QB n + 3	Direction keys							
	R3	R5	Travel command MCS/WCS	R11	-Y R9	-X R8	+Z R6	R15
QB n + 4	Unassigned customer keys							Direction key +X
	T9	T10	T11	T12	T13	T14	T15	R2
QB n + 5	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

Notes:

Machine control panel, slimline

Signals from slimline MCP (Keys and switches)
Input image

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IB n + 0	Spindle speed override				Operating mode			
	*NC Stop	SP -	SP 100%	SP +	SINGLEB	JOG	MDA	AUTO
IB n + 1	Spindle				Keyswitch	Machine function		
	NC Start	SP right	*SP Stop	SP left	SS 3	REF.	REPOS	Teach in
IB n + 2	Feedrate			Keyswitch	Machine functions			
	Start	*Stop	Var. INC	SS 0	1000 INC	100 INC	10 INC	1 INC
IB n + 3		Keyswitch		Feed override				
	RESET	SS 2	SS 1	E	D	C	B	A
IB n + 4	Direction keys			Optional customer keys				
	plus R15	minus R13	Rapid traverse R14	KT4	KT3	KT2	KT1	KT0
IB n + 5			Axis selection					
	T17	KT5	6	5	4	Z	Y	X
IB n + 6	Unassigned customer keys				MCS/WCS	Unassigned customer keys		
	T9	T10	T11	T12		T14	T15	T16
IB n + 7	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

Signals to slimline MCP (LEDs)
Output image

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
QB n + 0	NC Stop	Spindle speed override			Operating mode			
		SP -	SP 100 %	SP +	SINGLEB	JOG	MDA	AUTO
QB n + 1	NC Start	Spindle				Machine function		
		SP right	SP Stop	SP left	unassigned	REF.	REPOS	Teach in
QB n + 2	Feedrate				Machine functions			
	Start	Stop	Var. INC	unassigned	1000 INC	100 INC	10 INC	1 INC
QB n + 3	unassigned							
	unassigned	unassigned	unassigned	unassigned	unassigned	unassigned	unassigned	unassigned
QB n + 4	Direction keys			Optional customer keys				
	plus R15	minus R13	Rapid traverse R14	KT4	KT3	KT2	KT1	KT0
QB n + 5			Axis selection					
	T17	KT5	6	5	4	Z	Y	X
QB n + 6	Unassigned customer keys					Unassigned customer keys		
	T9	T10	T11	T12	MCS/WCS	T14	T15	T16
QB n + 7	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

Notes:

Signals from/to handheld unit (HT 2)

Signals from handheld unit (keys) Input image								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Reserved							
IB n + 1	Reserved							
IB n + 2	Feed Start	Unassigned key T2	AUTO	NC Stop	Spindle Stop	Feed Stop	Unassigned key T1	JOG
IB n + 3	Unassigned key T3	Handwheel	4. Axis	Z	Y	X	NC Start	Spindle Start
IB n + 4	Direction key -	Rapid traverse	Direction keys +	Unassigned key T4				
IB n + 5	Acknowledgement Digital display	Keyswitch	Rapid traverse/feed override switch E	D	C	B	A	

Signals to handheld unit (LEDs) Output image								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 0	always 1				Unassigned key T4	Unassigned key T3	Unassigned key T2	Unassigned key T1
QB n + 1	New data for selected line						Line selection	
							3 and 4	1 and 2
QB n + 2	Feed Start	Rapid traverse	AUTO	NC Stop	Spindle Stop	Feed Stop	Direction key +	JOG
QB n + 3	Direction key -	Handwheel	4. Axis	Z	Y	X	NC Start	Spindle Start
	Digital display of handheld unit							
QB n + 4	1st character (right) of the selected line							
QB n + 5	2nd character of the selected line							
...								
QB n + 19	16th character (left) of the selected line							

Notes:

Signals from/to handheld unit (HT 8)

Signals from MCP simulation Interface HT 8 → PLC								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Function key block							
	REF	TEACH	AUTO	MDA	JOG	QUIT	RESET	WCS/MCS
IB n + 1	Function key block							
	Cntr Panel Func (CPF)	U4	U3	BigFct	U2	U1	INC	REPOS
IB n + 2		Change over axes	Travel keys (JOG) positive direction					
		Ax7-Ax12 Ax1-Ax6	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
IB n + 3			Travel keys (JOG) negative direction					
			Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
IB n + 4								
	U16	U15	U14	U13	U12	U11	U10	U9
IB n + 5								
		U8	U7	U6	U5	SBL		
IB n + 6	Start key block							
	Reserved	HT 8	SF2	SF1	SF4	SF3	Start	Stop
IB n + 7	Feedrate override							
				E	D	C	B	A

Signals to MCP simulation Interface PLC → HT 8								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 0	Function key block							
	REF	TEACH	AUTO	MDA	JOG	QUIT	RESET	WCS/MCS
QB n + 1	Function key block							
		U4	U3		U2	U1	INC	REPOS
QB n + 2			Travel keys (JOG) positive direction					
		Axes 7n	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
QB n + 3	Travel keys (JOG) negative direction							
	For WCS: No MCS		Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
QB n + 4								
	U9	U10	U11	U12	U13	U14	U15	U16
QB n + 5								
		U8	U7	U6	U5	SBL		
QB n + 6	Start key block							
	Display travel keys		SF2	SF1	SF4	SF3	Start	Stop
QB n + 7								

Notes:

Signals from/to handheld unit (HT 10)

Signals from MCP simulation Interface HT 10 → PLC								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Function key block							
	REF	TEACH	AUTO	MDA	JOG	QUIT	RESET	WCS/MCS
IB n + 1	Function key block							
	Cntr Panel Func (CPF)	U4	U3	BigFct	U2	U1	INC	REPOS
IB n + 2		Change over axes Ax7-Ax12 Ax1-Ax6	Travel keys (JOG) positive direction					
			Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
IB n + 3	Jog keys HT 10		Travel keys (JOG) negative direction					
	Plus	Minus	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
IB n + 4								
	U16	U15	U14	U13	U12	U11	U10	U9
IB n + 5								
		U8	U7	U6	U5	SBL		Rapid traverse
IB n + 6	Start key block							
	HT 10	HT 8	SF2	SF1	SF4	SF3	Start	Stop
IB n + 7	Feedrate override							
				E	D	C	B	A

Signals to MCP simulation Interface PLC → HT 10								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 0	Function key block							
	REF	TEACH	AUTO	MDA	JOG	QUIT	RESET	WCS/MCS
QB n + 1	Function key block							
		U4	U3		U2	U1	INC	REPOS
QB n + 2	JOG+ (HT 10)	Axes 7-n selected	Travel keys (JOG) positive direction					
			Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
QB n + 3	Travel keys (JOG) negative direction							
	For WCS: No MCS	JOG- (HT 10)	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
QB n + 4								
	U9	U10	U11	U12	U13	U14	U15	U16
QB n + 5								
		U8	U7	U6	U5	SBL		
QB n + 6	Start key block							
	Display travel keys		SF2	SF1	SF4	SF3	Start	Stop
QB n + 7								

Notes:

PLC alarms / messages (DB2)

Signals WITHOUT display of a fault/operating message (DBB0 - 309)

DB2	Signals for PLC messages (PLC → HMI)							
Byte (Message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Signals WITHOUT display of a fault/operating message (DBB0 - 309)							
	Channel 1							
0	Feed disable byte 1							
1	Feed disable byte 2							
2	Read-in disable byte 1							
3	Read-in disable byte 2							
4	NC-Start disable byte 1							
5	NC-Start disable byte 1							
6	Feed stop GEOaxis 1, byte 1							
7	Feed stop GEOaxis 1, byte 2							
8	Feed stop GEOaxis 2, byte 1							
9	Feed stop GEOaxis 2, byte 2							
10	Feed stop GEOaxis 3, byte 1							
11	Feed stop GEOaxis 3, byte 2							
12 - 119	**Channel 2 - Channel 10,** see above “Channel 1”							
	Axis/spindle 1							
120	Feed stop / spindle stop, byte 1							
121	Feed stop / spindle stop, byte 2							
122 - 181	**Axis/spindle 2 - 31,** see above “Axis/spindle 1”							
	Additional values for user area 0							
182	Additional value for event number 700000							
184	Additional value for event number 700001							
...								
308	Additional value for event number 700063							

Notes:

Signals WITH display of a fault/operating message (from DBB310)

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Signals WITH display of a fault/operating message (from DBB310)							
	Channel 1							
	Feed disable (alarm no.: 510000-510015)							
310 (EM)	510007	510006	510005	510004	510003	510002	510001	510000
311 (OM)	510015	510014	510013	510012	510011	510010	510009	510008
312 (EM)	Feed and read-in disable byte 1 (alarm no.: 510100-510107)							
313 (EM)	Feed and read-in disable byte 2 (alarm no.: 510108-510115)							
314 (OM)	Feed and read-in disable byte 3 (alarm no.: 510116-510123)							
315 (OM)	Feed and read-in disable byte 4 (alarm no.: 510124-510131)							
316 (EM)	Read-in disable byte 1 (alarm no.: 510200-510207)							
317 (EM)	Read-in disable byte 2 (alarm no.: 510208-510215)							
318 (OM)	Read-in disable byte 3 (alarm no.: 510216-510223)							
319 (OM)	Read-in disable byte 4 (alarm no.: 510224-510231)							
320 (EM)	NC Start disable byte 1 (alarm no.: 510300-510307)							
321 (OM)	NC Start disable byte 2 (alarm no.: 510308-510315)							
322 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 511100-511107)							
323 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 511108-511115)							
324 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 511200-511207)							
325 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 511208-511215)							
326 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 511300-511307)							
327 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 511308-511315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 2							
	Feed disable (alarm no.: 510000-520015)							
328 (EM)	520007	520006	520005	520004	520003	520002	520001	520000
329 (OM)	520015	520014	520013	520012	520011	520010	520009	520008
330 (EM)	Feed and read-in disable byte 1 (alarm no.: 520100-520107)							
331 (EM)	Feed and read-in disable byte 2 (alarm no.: 520108-520115)							
332 (OM)	Feed and read-in disable byte 3 (alarm no.: 520116-520123)							
333 (OM)	Feed and read-in disable byte 4 (alarm no.: 520124-520131)							
334 (EM)	Read-in disable byte 1 (alarm no.: 520200-520207)							
335 (EM)	Read-in disable byte 2 (alarm no.: 520208-520215)							
336 (OM)	Read-in disable byte 3 (alarm no.: 520216-520223)							
337 (OM)	Read-in disable byte 4 (alarm no.: 520224-520231)							
338 (EM)	NC Start disable byte 1 (alarm no.: 520300-520307)							
339 (OM)	NC Start disable byte 2 (alarm no.: 520308-520315)							
340 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 521100-521107)							
341 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 521108-521115)							
342 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 521200-521207)							
343 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 521208-521215)							
344 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 521300-521307)							
345 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 521308-521315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 3							
	Feed disable (alarm no.: 530000-530015)							
346 (EM)	530007	530006	530005	530004	530003	530002	530001	530000
347 (OM)	530015	530014	530013	530012	530011	530010	530009	530008
348 (EM)	Feed and read-in disable byte 1 (alarm no.: 530100-530107)							
349 (EM)	Feed and read-in disable byte 2 (alarm no.: 530108-530115)							
350 (OM)	Feed and read-in disable byte 3 (alarm no.: 530116-530123)							
351 (OM)	Feed and read-in disable byte 4 (alarm no.: 530124-530131)							
352 (EM)	Read-in disable byte 1 (alarm no.: 530200-530207)							
353 (EM)	Read-in disable byte 2 (alarm no.: 530208-530215)							
354 (OM)	Read-in disable byte 3 (alarm no.: 530216-530223)							
355 (OM)	Read-in disable byte 4 (alarm no.: 530224-530231)							
356 (EM)	NC Start disable byte 1 (alarm no.: 530300-530307)							
357 (OM)	NC Start disable byte 2 (alarm no.: 530308-530315)							
358 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 531100-531107)							
359 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 531108-531115)							
360 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 531200-531207)							
361 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 531208-531215)							
362 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 531300-531307)							
363 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 531308-531315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 4							
	Feed disable (alarm no.: 540000-540015)							
364 (EM)	540007	540006	540005	540004	540003	540002	540001	540000
365 (OM)	540015	540014	540013	540012	540011	540010	540009	540008
366 (EM)	Feed and read-in disable byte 1 (alarm no.: 540100-540107)							
367 (EM)	Feed and read-in disable byte 2 (alarm no.: 540108-540115)							
368 (OM)	Feed and read-in disable byte 3 (alarm no.: 540116-540123)							
369 (OM)	Feed and read-in disable byte 4 (alarm no.: 540124-540131)							
370 (EM)	Read-in disable byte 1 (alarm no.: 540200-540207)							
371 (EM)	Read-in disable byte 2 (alarm no.: 540208-540215)							
372 (OM)	Read-in disable byte 3 (alarm no.: 540216-540223)							
373 (OM)	Read-in disable byte 4 (alarm no.: 540224-540231)							
374 (EM)	NC Start disable byte 1 (alarm no.: 540300-540307)							
375 (OM)	NC Start disable byte 2 (alarm no.: 540308-540315)							
376 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 541100-541107)							
377 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 541108-541115)							
378 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 541200-541207)							
379 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 541208-541215)							
380 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 541300-541307)							
381 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 541308-541315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 5							
	Feed disable (alarm no.: 550000-550015)							
382 (EM)	550007	550006	550005	550004	550003	550002	550001	550000
383 (OM)	550015	550014	550013	550012	550011	550010	550009	550008
384 (EM)	Feed and read-in disable byte 1 (alarm no.: 550100-550107)							
385 (EM)	Feed and read-in disable byte 2 (alarm no.: 550108-550115)							
386 (OM)	Feed and read-in disable byte 3 (alarm no.: 550116-550123)							
387 (OM)	Feed and read-in disable byte 4 (alarm no.: 550124-550131)							
388 (EM)	Read-in disable byte 1 (alarm no.: 550200-550207)							
389 (EM)	Read-in disable byte 2 (alarm no.: 550208-550215)							
390 (OM)	Read-in disable byte 3 (alarm no.: 550216-550223)							
391 (OM)	Read-in disable byte 4 (alarm no.: 550224-550231)							
392 (EM)	NC Start disable byte 1 (alarm no.: 550300-550307)							
393 (OM)	NC Start disable byte 2 (alarm no.: 550308-550315)							
394 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 551100-551107)							
395 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 551108-551115)							
396 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 551200-551207)							
397 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 551208-551215)							
398 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 551300-551307)							
399 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 551308-551315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 6							
	Feed disable (alarm no.: 560000-560015)							
400 (EM)	560007	560006	560005	560004	560003	560002	560001	560000
401 (OM)	560015	560014	560013	560012	560011	560010	560009	560008
402 (EM)	Feed and read-in disable byte 1 (alarm no.: 560100-560107)							
403 (EM)	Feed and read-in disable byte 2 (alarm no.: 560108-560115)							
404 (OM)	Feed and read-in disable byte 3 (alarm no.: 560116-560123)							
405 (OM)	Feed and read-in disable byte 4 (alarm no.: 560124-560131)							
406 (EM)	Read-in disable byte 1 (alarm no.: 560200-560207)							
407(EM)	Read-in disable byte 2 (alarm no.: 560208-560215)							
408 (OM)	Read-in disable byte 3 (alarm no.: 560216-560223)							
409 (OM)	Read-in disable byte 4 (alarm no.: 560224-560231)							
410 (EM)	NC Start disable byte 1 (alarm no.: 560300-560307)							
411 (OM)	NC Start disable byte 2 (alarm no.: 560308-560315)							
412 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 561100-561107)							
413 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 561108-561115)							
414 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 561200-561207)							
415 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 561208-561215)							
416 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 561300-561307)							
417 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 561308-561315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 7							
	Feed disable (alarm no.: 570000-570015)							
418 (EM)	570007	570006	570005	570004	570003	570002	570001	570000
419 (OM)	570015	570014	570013	570012	570011	570010	570009	570008
420 (EM)	Feed and read-in disable byte 1 (alarm no.: 570100-570107)							
421 (EM)	Feed and read-in disable byte 2 (alarm no.: 570108-570115)							
422 (OM)	Feed and read-in disable byte 3 (alarm no.: 570116-570123)							
423 (OM)	Feed and read-in disable byte 4 (alarm no.: 570124-570131)							
424 (EM)	Read-in disable byte 1 (alarm no.: 570200-570207)							
425 (EM)	Read-in disable byte 2 (alarm no.: 570208-570215)							
426 (OM)	Read-in disable byte 3 (alarm no.: 570216-570223)							
427 (OM)	Read-in disable byte 4 (alarm no.: 570224-570231)							
428 (EM)	NC Start disable byte 1 (alarm no.: 570300-570307)							
429 (OM)	NC Start disable byte 2 (alarm no.: 570308-570315)							
430 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 571100-571107)							
431 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 571108-571115)							
432 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 571200-571207)							
433 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 571208-571215)							
434 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 571300-571307)							
435 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 571308-571315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 8							
	Feed disable (alarm no.: 580000-580015)							
436 (EM)	580007	580006	580005	580004	580003	580002	580001	580000
437 (OM)	580015	580014	580013	580012	580011	580010	580009	580008
438 (EM)	Feed and read-in disable byte 1 (alarm no.: 580100-580107)							
439 (EM)	Feed and read-in disable byte 2 (alarm no.: 580108-580115)							
440 (OM)	Feed and read-in disable byte 3 (alarm no.: 580116-580123)							
441 (OM)	Feed and read-in disable byte 4 (alarm no.: 580124-580131)							
442 (EM)	Read-in disable byte 1 (alarm no.: 580200-580207)							
443 (EM)	Read-in disable byte 2 (alarm no.: 580208-580215)							
444 (OM)	Read-in disable byte 3 (alarm no.: 580216-580223)							
445 (OM)	Read-in disable byte 4 (alarm no.: 580224-580231)							
446 (EM)	NC Start disable byte 1 (alarm no.: 580300-580307)							
447 (OM)	NC Start disable byte 2 (alarm no.: 580308-580315)							
448 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 581100-581107)							
449 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 581108-581115)							
450 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 581200-581207)							
451 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 581208-581215)							
452 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 581300-581307)							
453 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 581308-581315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 9							
	Feed disable (alarm no.: 590000-590015)							
454 (EM)	590007	590006	590005	590004	590003	590002	590001	590000
455 (OM)	590015	590014	590013	590012	590011	590010	590009	590008
456 (EM)	Feed and read-in disable byte 1 (alarm no.: 590100-590107)							
457 (EM)	Feed and read-in disable byte 2 (alarm no.: 590108-590115)							
458 (OM)	Feed and read-in disable byte 3 (alarm no.: 590116-590123)							
459 (OM)	Feed and read-in disable byte 4 (alarm no.: 590124-590131)							
460 (EM)	Read-in disable byte 1 (alarm no.: 590200-590207)							
461(EM)	Read-in disable byte 2 (alarm no.: 590208-590215)							
462 (OM)	Read-in disable byte 3 (alarm no.: 590216-590223)							
463 (OM)	Read-in disable byte 4 (alarm no.: 590224-590231)							
464 (EM)	NC Start disable byte 1 (alarm no.: 590300-590307)							
465 (OM)	NC Start disable byte 2 (alarm no.: 590308-590315)							
466 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 591100-591107)							
467 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 591108-591115)							
468 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 591200-591207)							
469 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 591208-591215)							
470 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 591300-591307)							
471 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 591308-591315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 10							
	Feed disable (alarm no.: 500000-500015)							
472 (EM)	500007	500006	500005	500004	500003	500002	500001	500000
473 (OM)	500015	500014	500013	500012	500011	500010	500009	500008
474 (EM)	Feed and read-in disable byte 1 (alarm no.: 500100-500107)							
475 (EM)	Feed and read-in disable byte 2 (alarm no.: 500108-500115)							
476 (OM)	Feed and read-in disable byte 3 (alarm no.: 500116-500123)							
477 (OM)	Feed and read-in disable byte 4 (alarm no.: 500124-500131)							
478 (EM)	Read-in disable byte 1 (alarm no.: 500200-500207)							
479(EM)	Read-in disable byte 2 (alarm no.: 500208-500215)							
480 (OM)	Read-in disable byte 3 (alarm no.: 500216-500223)							
481 (OM)	Read-in disable byte 4 (alarm no.: 500224-500231)							
482 (EM)	NC Start disable byte 1 (alarm no.: 500300-500307)							
483 (OM)	NC Start disable byte 2 (alarm no.: 500308-500315)							
484 (EM)	Feed stop GEOaxis 1 byte 1 (alarm no.: 501100-501107)							
485 (OM)	Feed stop GEOaxis 1 byte 2 (alarm no.: 501108-501115)							
486 (EM)	Feed stop GEOaxis 2 byte 1 (alarm no.: 501200-501207)							
487 (OM)	Feed stop GEOaxis 2 byte 2 (alarm no.: 501208-501215)							
488 (EM)	Feed stop GEOaxis 3 byte 1 (alarm no.: 501300-501307)							
489 (OM)	Feed stop GEOaxis 3 byte 2 (alarm no.: 501308-501315)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Axis/spindle							
	Feed stop/spindle stop for axis/spindle 1 (alarm no.: 600100-600115)							
490 (EM)	600107	600106	600105	600104	600103	600102	600101	600100
491 (OM)	600115	600114	600113	600112	600111	600110	600109	600108
492 (EM)	Feed stop/spindle stop for axis/spindle 2 (alarm no.: 600200-600207)							
493 (OM)	Feed stop/spindle stop for axis/spindle 2 (alarm no.: 600208-600215)							
494 (EM)	Feed stop/spindle stop for axis/spindle 3 (alarm no.: 600300-600307)							
495 (OM)	Feed stop/spindle stop for axis/spindle 3 (alarm no.: 600308-600315)							
496 (EM)	Feed stop/spindle stop for axis/spindle 4 (alarm no.: 600400-600407)							
497 (OM)	Feed stop/spindle stop for axis/spindle 4 (alarm no.: 600408-600415)							
498 (EM)	Feed stop/spindle stop for axis/spindle 5 (alarm no.: 600500-600507)							
499 (OM)	Feed stop/spindle stop for axis/spindle 5 (alarm no.: 600508-600515)							
500 (EM)	Feed stop/spindle stop for axis/spindle 6 (alarm no.: 600600-600607)							
501 (OM)	Feed stop/spindle stop for axis/spindle 6 (alarm no.: 600608-600615)							
502 (EM)	Feed stop/spindle stop for axis/spindle 7 (alarm no.: 600700-600707)							
503 (OM)	Feed stop/spindle stop for axis/spindle 7(alarm no.: 600708-600715)							
504 (EM)	Feed stop/spindle stop for axis/spindle 8 (alarm no.: 600800-600807)							
505 (OM)	Feed stop/spindle stop for axis/spindle 8 (alarm no.: 600808-600815)							
506 (EM)	Feed stop/spindle stop for axis/spindle 9 (alarm no.: 600900-600907)							
507 (OM)	Feed stop/spindle stop for axis/spindle 9 (alarm no.: 600908-600915)							
508 (EM)	Feed stop/spindle stop for axis/spindle 10 (alarm no.: 601000-601007)							
509 (OM)	Feed stop/spindle stop for axis/spindle 10 (alarm no.: 601008-601015)							
510 (EM)	Feed stop/spindle stop for axis/spindle 11 (alarm no.: 601100-601107)							
511 (OM)	Feed stop/spindle stop for axis/spindle 11 (alarm no.: 601108-601115)							
512 (EM)	Feed stop/spindle stop for axis/spindle 12 (alarm no.: 601200-601207)							
513 (OM)	Feed stop/spindle stop for axis/spindle 12 (alarm no.: 601208-601215)							
514 (EM)	Feed stop/spindle stop for axis/spindle 13 (alarm no.: 601300-601307)							
515 (OM)	Feed stop/spindle stop for axis/spindle 13 (alarm no.: 601308-601315)							
516 (EM)	Feed stop/spindle stop for axis/spindle 14 (alarm no.: 601400-601407)							
517 (OM)	Feed stop/spindle stop for axis/spindle 14 (alarm no.: 601408-601415)							
518 (EM)	Feed stop/spindle stop for axis/spindle 15 (alarm no.: 601500-601507)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
519 (OM)	Feed stop/spindle stop for axis/spindle 15 (alarm no.: 601508-601515)							
520 (EM)	Feed stop/spindle stop for axis/spindle 16 (alarm no.: 601600-601607)							
521 (OM)	Feed stop/spindle stop for axis/spindle 16 (alarm no.: 601608-601615)							
522 (EM)	Feed stop/spindle stop for axis/spindle 17 (alarm no.: 601700-601707)							
523 (OM)	Feed stop/spindle stop for axis/spindle 17 (alarm no.: 601708-601715)							
524 (EM)	Feed stop/spindle stop for axis/spindle 18 (alarm no.: 601800-601807)							
525 (OM)	Feed stop/spindle stop for axis/spindle 18 (alarm no.: 601808-601815)							
526 (EM)	Feed stop/spindle stop for axis/spindle 19 (alarm no.: 601900-601907)							
527 (OM)	Feed stop/spindle stop for axis/spindle 19 (alarm no.: 601908-601915)							
528 (EM)	Feed stop/spindle stop for axis/spindle 20 (alarm no.: 602000-602007)							
529 (OM)	Feed stop/spindle stop for axis/spindle 20 (alarm no.: 602008-602015)							
530 (EM)	Feed stop/spindle stop for axis/spindle 21 (alarm no.: 602100-602107)							
531 (OM)	Feed stop/spindle stop for axis/spindle 21 (alarm no.: 602108-602115)							
532 (EM)	Feed stop/spindle stop for axis/spindle 22 (alarm no.: 602200-602207)							
533 (OM)	Feed stop/spindle stop for axis/spindle 22 (alarm no.: 602208-602215)							
534 (EM)	Feed stop/spindle stop for axis/spindle 23 (alarm no.: 602300-602307)							
535 (OM)	Feed stop/spindle stop for axis/spindle 23 (alarm no.: 602308-602315)							
536 (EM)	Feed stop/spindle stop for axis/spindle 24 (alarm no.: 602400-602407)							
537 (OM)	Feed stop/spindle stop for axis/spindle 24 (alarm no.: 602408-602415)							
538 (EM)	Feed stop/spindle stop for axis/spindle 25 (alarm no.: 602500-602507)							
539 (OM)	Feed stop/spindle stop for axis/spindle 25 (alarm no.: 602508-602515)							
540 (EM)	Feed stop/spindle stop for axis/spindle 26 (alarm no.: 602600-602607)							
541 (OM)	Feed stop/spindle stop for axis/spindle 26 (alarm no.: 602608-602615)							
542 (EM)	Feed stop/spindle stop for axis/spindle 27 (alarm no.: 602700-602707)							
543 (OM)	Feed stop/spindle stop for axis/spindle 27 (alarm no.: 602708-602715)							
544 (EM)	Feed stop/spindle stop for axis/spindle 28 (alarm no.: 602800-602807)							
545 (OM)	Feed stop/spindle stop for axis/spindle 28 (alarm no.: 602808-602815)							
546 (EM)	Feed stop/spindle stop for axis/spindle 29 (alarm no.: 602900-602907)							
547 (OM)	Feed stop/spindle stop for axis/spindle 29 (alarm no.: 602908-602915)							
548 (EM)	Feed stop/spindle stop for axis/spindle 30 (alarm no.: 603000-603007)							
549 (OM)	Feed stop/spindle stop for axis/spindle 30 (alarm no.: 603008-603015)							
550 (EM)	Feed stop/spindle stop for axis/spindle 31 (alarm no.: 603100-603107)							
551 (OM)	Feed stop/spindle stop for axis/spindle 31 (alarm no.: 603108-603115)							



Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	User areas							
	User area 0 (alarm no.: 700000-700015)							
554 (EM)	700007	700006	700005	700004	700003	700002	700001	700000
555 (EM)	700015	700014	700013	700012	700011	700010	700009	700008
556 (EM)	User area 0: byte 3 (alarm no.: 700016-700023)							
557 (EM)	User area 0: byte 4 (alarm no.: 700024-700031)							
558 (OM)	User area 0: byte 5 (alarm no.: 700032-700039)							
559 (OM)	User area 0: byte 6 (alarm no.: 700040-700047)							
560 (OM)	User area 0: byte 7 (alarm no.: 700048-700055)							
561 (OM)	User area 0: byte 8 (alarm no.: 700056-700063)							
562 - 565 (EM)	User area 1 bytes 1 - 4 (alarm no.: 700100-700131)							
566 - 569 (OM)	User area 1 bytes 5 - 8 (alarm no.: 700132-700163)							
570 - 573 (EM)	User area 2 bytes 1 - 4 (alarm no.: 700200-700231)							
574 - 577 (OM)	User area 2 bytes 5 - 8 (alarm no.: 700232-700263)							
578 - 581 (EM)	User area 3 bytes 1 - 4 (alarm no.: 700300-700331)							
582 - 585 (OM)	User area 3 bytes 5 - 8 (alarm no.: 700332-700363)							
586 - 589 (EM)	User area 4 bytes 1 - 4 (alarm no.: 700400-700431)							
590 - 593 (OM)	User area 4 bytes 5 - 8 (alarm no.: 700432-700463)							
594 - 597 (EM)	User area 5 bytes 1 - 4 (alarm no.: 700500-700531)							
598 - 601 (OM)	User area 5 bytes 5 - 8 (alarm no.: 700532-700563)							
602 - 605 (EM)	User area 6 bytes 1 - 4 (alarm no.: 700600-700631)							
606 - 609 (OM)	User area 6 bytes 5 - 8 (alarm no.: 700632-700663)							
610 - 613 (EM)	User area 7 bytes 1 - 4 (alarm no.: 700700-700731)							
614 - 617 (OM)	User area 7 bytes 5 - 8 (alarm no.: 700732-700763)							
618 - 621 (EM)	User area 8 bytes 1 - 4 (alarm no.: 700800-700831)							
622 - 625 (OM)	User area 8 bytes 5 - 8 (alarm no.: 700832-700863)							
626 - 629 (EM)	User area 9 bytes 1 - 4 (alarm no.: 700900-700931)							
630 - 633 (OM)	User area 9 bytes 5 - 8 (alarm no.: 700932 - 700963)							
634 - 637 (EM)	User area 10 bytes 1 - 4 (alarm no.: 701000-701031)							
638 - 641 (OM)	User area 10 bytes 5 - 8 (alarm no.: 701032-701063)							
642 - 645 (EM)	User area 11 bytes 1 - 4 (alarm no.: 701100-701131)							
646 - 649 (OM)	User area 11 bytes 5 - 8 (alarm no.: 701132-701163)							
650 - 653 (EM)	User area 12 bytes 1 - 4 (alarm no.: 701200-701231)							
654 - 657 (OM)	User area 12 bytes 5 - 8 (alarm no.: 701232-701263)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
658 - 661 (EM)	User area 13 bytes 1 - 4 (alarm no.: 701300-701331)							
662 - 665 (OM)	User area 13 bytes 5 - 8 (alarm no.: 701332-701363)							
666 - 669 (EM)	User area 14 bytes 1 - 4 (alarm no.: 701400-701431)							
670 - 673 (OM)	User area 14 bytes 5 - 8 (alarm no.: 701432-701463)							
674 - 677 (EM)	User area 15 bytes 1 - 4 (alarm no.: 701500-701531)							
678 - 681 (OM)	User area 15 bytes 5 - 8 (alarm no.: 701532-701563)							
682 - 685 (EM)	User area 16 bytes 1 - 4 (alarm no.: 701600-701631)							
686 - 689 (EM)	User area 16 bytes 5 - 8 (alarm no.: 701632-701663)							
690 - 693 (EM)	User area 17 bytes 1 - 4 (alarm no.: 701700-701731)							
694 - 697 (OM)	User area 17 bytes 5 - 8 (alarm no.: 701732-701763)							
698 - 701 (EM)	User area 18 bytes 1 - 4 (alarm no.: 701800-701831)							
702 - 705 (OM)	User area 18 bytes 5 - 8 (alarm no.: 701832-701863)							
706 - 709 (EM)	User area 19 bytes 1 - 4 (alarm no.: 701900-701931)							
710 - 713 (OM)	User area 19 bytes 5 - 8 (alarm no.: 701932-701963)							
714 - 717 (EM)	User area 20 bytes 1 - 4 (alarm no.: 702000-702031)							
718 - 721 (OM)	User area 20 bytes 5 - 8 (alarm no.: 702032-702063)							
722 - 725 (EM)	User area 21 bytes 1 - 4 (alarm no.: 702100-702131)							
726 - 729 (OM)	User area 21 bytes 5 - 8 (alarm no.: 702132-702163)							
730 - 733 (EM)	User area 22 bytes 1 - 4 (alarm no.: 702200-702231)							
734 - 737 (OM)	User area 22 bytes 5 - 8 (alarm no.: 702232-702263)							
738 - 741 (EM)	User area 23 bytes 1 - 4 (alarm no.: 702300-702331)							
742 - 745 (OM)	User area 23 bytes 5 - 8 (alarm no.: 702332-702363)							
746 - 749 (EM)	User area 24 bytes 1 - 4 (alarm no.: 702400-702431)							
750 - 753 (OM)	User area 24 bytes 5 - 8 (alarm no.: 702432-702463)							
754 - 757 (EM)	User area 25 bytes 1 - 4 (alarm no.: 702500-702531)							
758 - 761 (OM)	User area 25 bytes 5 - 8 (alarm no.: 702532-702563)							
762 - 765 (EM)	User area 26 bytes 1 - 4 (alarm no.: 702600-702631)							
766 - 769 (OM)	User area 26 bytes 5 - 8 (alarm no.: 702632-702663)							
770 - 773 (EM)	User area 27 bytes 1 - 4 (alarm no.: 702700-702731)							
774 - 777 (OM)	User area 27 bytes 5 - 8 (alarm no.: 702732-702763)							
778 - 781 (EM)	User area 28 bytes 1 - 4 (alarm no.: 702800-702831)							
782 - 785 (OM)	User area 28 bytes 5 - 8 (alarm no.: 702832-702863)							
786 - 789 (EM)	User area 29 bytes 1 - 4 (alarm no.: 702900-702931)							
790 - 793 (OM)	User area 29 bytes 5 - 8 (alarm no.: 702932-702963)							



Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
794 - 797 (EM)	User area 30 bytes 1 - 4 (alarm no.: 703000-703031)							
798 - 801 (OM)	User area 30 bytes 5 - 8 (alarm no.: 703032-703063)							
802 - 805 (EM)	User area 31 bytes 1 - 4 (alarm no.: 703100-703131)							
806 - 809 (OM)	User area 31 bytes 5 - 8 (alarm no.: 703132-703163)							
810 - 813 (EM)	User area 32 bytes 1 - 4 (alarm no.: 703200-703231)							
814 - 817 (OM)	User area 32 bytes 5 - 8 (alarm no.: 703232-703263)							
818 - 821 (EM)	User area 33 bytes 1 - 4 (alarm no.: 703300-703331)							
822 - 825 (OM)	User area 33 bytes 5 - 8 (alarm no.: 703332-703363)							
826 - 829 (EM)	User area 34 bytes 1 - 4 (alarm no.: 703400-703431)							
830 - 833 (OM)	User area 34 bytes 5 - 8 (alarm no.: 703432-703463)							
834 - 837 (EM)	User area 35 bytes 1 - 4 (alarm no.: 703500-703531)							
838 - 841 (OM)	User area 35 bytes 5 - 8 (alarm no.: 703532-703563)							
842 - 845 (EM)	User area 36 bytes 1 - 4 (alarm no.: 703600-703631)							
846 - 849 (OM)	User area 36 bytes 5 - 8 (alarm no.: 703632-703663)							
850 - 853 (EM)	User area 37 bytes 1 - 4 (alarm no.: 703700-703731)							
854 - 857 (OM)	User area 37 bytes 5 - 8 (alarm no.: 703732-703763)							
858 - 861 (EM)	User area 38 bytes 1 - 4 (alarm no.: 703800-703831)							
862 - 865 (OM)	User area 38 bytes 5 - 8 (alarm no.: 703832-703863)							
866 - 869 (EM)	User area 39 bytes 1 - 4 (alarm no.: 703900-703931)							
870 - 873 (OM)	User area 39 bytes 5 - 8 (alarm no.: 703932-703963)							
874 - 877 (EM)	User area 40 bytes 1 - 4 (alarm no.: 704000-704031)							
878 - 881 (OM)	User area 40 bytes 5 - 8 (alarm no.: 704032-704063)							
882 - 885 (EM)	User area 41 bytes 1 - 4 (alarm no.: 704100-704131)							
886 - 889 (OM)	User area 41 bytes 5 - 8 (alarm no.: 704132-704163)							
890 - 893 (EM)	User area 42 bytes 1 - 4 (alarm no.: 704200-704231)							
894 - 897 (OM)	User area 42 bytes 5 - 8 (alarm no.: 704232-704263)							
898 - 901 (EM)	User area 43 bytes 1 - 4 (alarm no.: 704300-704331)							
902 - 905 (OM)	User area 43 bytes 5 - 8 (alarm no.: 704332-704363)							
906 - 909 (EM)	User area 44 bytes 1 - 4 (alarm no.: 704400-704431)							
910 - 913 (OM)	User area 44 bytes 5 - 8 (alarm no.: 704432-704463)							
914 - 917 (EM)	User area 45 bytes 1 - 4 (alarm no.: 704500-704531)							
918 - 921 (OM)	User area 45 bytes 5 - 8 (alarm no.: 704532-704563)							
922 - 925 (EM)	User area 46 bytes 1 - 4 (alarm no.: 704600-704631)							
926 - 929 (OM)	User area 46 bytes 5 - 8 (alarm no.: 704632-704663)							

Notes:

Byte (Message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
930 - 933 (EM)	User area 47 Byte 1 - 4 (alarm no.: 704700-704731)							
934 - 937 (OM)	User area 47 Byte 5 - 8 (alarm no.: 704732-704763)							
938 - 941 (EM)	User area 48 Byte 1 - 4 (alarm no.: 704800-704831)							
942 - 945 (OM)	User area 48 Byte 5 - 8 (alarm no.: 704832-704863)							
946 - 949 (EM)	User area 49 Byte 1 - 4 (alarm no.: 704900-704931)							
950 - 953 (OM)	User area 49 Byte 5 - 8 (alarm no.: 704932-704963)							
954 - 957 (EM)	User area 50 Byte 1 - 4 (alarm no.: 705000-705031)							
958 - 961 (OM)	User area 50 Byte 5 - 8 (alarm no.: 705032-705063)							
962 - 965 (EM)	User area 51 Byte 1 - 4 (alarm no.: 705100-705131)							
966 - 969 (OM)	User area 51 Byte 5 - 8 (alarm no.: 705132-705163)							
970 - 973 (EM)	User area 52 Byte 1 - 4 (alarm no.: 705200-705231)							
974 - 977 (OM)	User area 52 Byte 5 - 8 (alarm no.: 705232-705263)							
978 - 981 (EM)	User area 53 Byte 1 - 4 (alarm no.: 705300-705331)							
982 - 985 (OM)	User area 53 Byte 5 - 8 (alarm no.: 705332-705363)							
986 - 989 (EM)	User area 54 Byte 1 - 4 (alarm no.: 705400-705431)							
990 - 993 (OM)	User area 54 Byte 5 - 8 (alarm no.: 705432-705463)							
994 - 997 (EM)	User area 55 Byte 1 - 4 (alarm no.: 705500-705531)							
998 - 1001 (OM)	User area 55 Byte 5 - 8 (alarm no.: 705532-705563)							
1002 - 1005 (EM)	User area 56 Byte 1 - 4 (alarm no.: 705600-705631)							
1006 - 1009 (OM)	User area 56 Byte 5 - 8 (alarm no.: 705632-705663)							
1010 - 1013 (EM)	User area 57 Byte 1 - 4 (alarm no.: 705700-705731)							
1014 - 1017 (OM)	User area 57 Byte 5 - 8 (alarm no.: 705732-705763)							
1018 - 1021 (EM)	User area 58 Byte 1 - 4 (alarm no.: 705800-705831)							
1022 - 1025 (OM)	User area 58 Byte 5 - 8 (alarm no.: 705832-705863)							
1026 - 1029 (EM)	User area 59 Byte 1 - 4 (alarm no.: 705900-705931)							
1030 - 1033 (OM)	User area 59 Byte 5 - 8 (alarm no.: 705932-705963)							
1034 - 1037 (EM)	User area 60 Byte 1 - 4 (alarm no.: 706000-706031)							
1038 - 1041 (OM)	User area 60 Byte 5 - 8 (alarm no.: 706032-706063)							
1042 - 1045 (EM)	User area 61 Byte 1 - 4 (alarm no.: 706100-706131)							
1046 - 1049 (OM)	User area 61 Byte 5 - 8 (alarm no.: 706132-706163)							
1050 - 1053 (EM)	User area 62 Byte 1 - 4 (alarm no.: 706200-706231)							
1054 - 1057 (OM)	User area 62 Byte 5 - 8 (alarm no.: 706232-706263)							
1058 - 1061 (EM)	User area 63 Byte 1 - 4 (alarm no.: 706300-706331)							
1062 - 1065 (OM)	User area 63 Byte 5 - 8 (alarm no.: 706332-706363)							



Notes:

PLC Message type (DB5)

Message type for channels

DB5	Signals for message type (PLC → HMI)							
Bit (Message type in DB2 TRUE: EM FALSE: OM)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 1							
0.0	Feed disable (alarm no.: 5C0000-5C0007)							
0.1	Feed disable (alarm no.: 5C0008-5C0015)							
0.2	Feed and read-in disable byte 1 (alarm no.: 5C0100-5C0107)							
0.3	Feed and read-in disable byte 2 (alarm no.: 5C0108-5C0115)							
0.4	Feed and read-in disable byte 3 (alarm no.: 5C0116-5C0123)							
0.5	Feed and read-in disable byte 4 (alarm no.: 5C0124-5C0131)							
0.6	Read-in disable byte 1 (alarm no.: 5C0200-5C0207)							
0.7	Read-in disable byte 2 (alarm no.: 5C0208-5C0215)							
1.0	Read-in disable byte 3 (alarm no.: 5C0216-5C0223)							
1.1	Read-in disable byte 4 (alarm no.: 5C0224-5C0231)							
1.2	NC Start disable byte 1 (alarm no.: 5C0300-5C0307)							
1.3	NC Start disable byte 2 (alarm no.: 5C0308-5C0315)							
1.4	Feed stop GEOaxis 1 byte 1 (alarm no.: 5C1100-5C1107)							
1.5	Feed stop GEOaxis 1 byte 2 (alarm no.: 5C1108-5C1115)							
1.6	Feed stop GEOaxis 2 byte 1 (alarm no.: 5C1200-5C1207)							
1.7	Feed stop GEOaxis 2 byte 2 (alarm no.: 5C1208-5C1215)							
2.0	Feed stop GEOaxis 3 byte 1 (alarm no.: 5C1300-5C1307)							
2.1	Feed stop GEOaxis 3 byte 2 (alarm no.: 5C1308-5C1315)							
	Channel 2							
2.2	Feed disable (alarm no.: 5C0000-5C0007)							
	...							
	Channel 10							
...	...							
22.3	Feed stop GEOaxis 3 byte 2 (alarm no.: 5C1308-5C1315)							

Notes:

Message type for axes and spindles

Bit (Message type in DB2 TRUE: EM FALSE: OM)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Axis/spindle 1							
22.4	Feed stop/spindle stop for axis/spindle 1 (alarm no.: 60AA00-60AA07)							
22.5	Feed stop/spindle stop for axis/spindle 1 (alarm no.: 60AA08-60AA15)							
	Axis/spindle 2							
22.6	Feed stop/spindle stop for axis/spindle 2 (alarm no.: 60AA00-60AA07)							
	...							
	Axis/spindle 31							
	...							
30.1	Feed stop/spindle stop for axis/spindle 31 (alarm no.: 60AA08-603115)							
30.2	Reserved							
30.3	Reserved							

Message type for user areas

Bit (Message type in DB2 TRUE: EM FALSE: OM)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	User 0							
30.4	User area 0: byte 1 (alarm no.: A70UU00-A70UU07)							
30.5	User area 0: byte 2 (alarm no.: A70UU08-A70UU15)							
30.6	User area 0: byte 3 (alarm no.: A70UU16-A70UU23)							
30.7	User area 0: byte 4 (alarm no.: A70UU24-A70UU31)							
31.0	User area 0: byte 5 (alarm no.: A70UU32-A70UU39)							
31.1	User area 0: byte 6 (alarm no.: A70UU40-A70UU47)							
31.2	User area 0: byte 7 (alarm no.: A70UU48-A70UU55)							
31.3	User area 0: byte 8 (alarm no.: A70UU56-A70UU63)							
	User 1							
31.4	User area 1 bytes 1 (alarm no.: A70UU00-A70UU07)							
	….							
	User 63							
	...							
167.7	User area 63 byte 8 (alarm no.: A70UU56-A70UU63)							

Notes:

Signals from/ to NC, PLC, HMI (DB10)

On-board NC input and outputs to NC (DB10)

DB10	Signals to NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Disabling of digital NC inputs							
	Digital inputs without hardware				On-board inputs			
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
DBB1	Setting of digital NC inputs from PLC							
	Digital inputs without hardware				On-board inputs			
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
DBB2 - DBB3	Unassigned							
DBB4	Disabling of digital NC outputs							
	Digital outputs without hardware				On-board outputs			
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB5	Overwrite screenform of digital NC outputs							
	Digital outputs without hardware				On-board outputs			
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB6	Setting value of digital NC outputs from PLC							
	Digital outputs without hardware				On-board outputs			
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB7	Input screenform of digital NC outputs							
	Digital outputs without hardware				On-board outputs			
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB8 - DBB29	Machine axis number table for FC 19, FC 24, FC 25, FC 26 (1st MCP)							
DBB30	Upper limit of machine axis numbers for FC 19, FC 24 (1st MCP) With 0, the max. number of machine axis numbers applies							
DBB32 - DBB53	Machine axis number table for FC 19, FC 24, FC 25, FC 26 (2nd MCP)							
DBB54	Upper limit of machine axis numbers for FC 19, FC 24 (2nd MCP) With 0, the max. number of machine axis numbers applies							

Notes:

General signals to NC (DB10)

DB10	Signals to NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB56	Keyswitch					Acknowl. Emergency Stop	Emergency Stop	
	Position 3	Position 2	Position 1	Position 0				
DBB57					Reserved			INC inputs in mode group area active
DBB58	Collision avoidance: Deactivate protection zone group							
	For operating mode: JOG				For operating mode: AUTO			
	Workpieces	Workholder	Tools	Machine	Workpieces	Workholder	Tools	Tools
DBB59								

On board NC/HMI inputs and outputs (DB10)

DB10	Signals from (NC -> PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB60	Actual value of the digital inputs of the NC				Actual value of the digital ON-BOARD inputs of the NC On-board inputs			
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
DBB61 - DBB63								
DBB64	Setpoint for the digital outputs of the NC without hardware				Setpoint for the digital on-board outputs of the NC			
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB65 - DBB67	unassigned							
DBB68	Handwheel 1 moved							
DBB69	Handwheel 2 moved							
DBB70	Handwheel 3 moved							
DBB71	Modification counter inch/metric system of units							
DBB72 HT 8 → HMI	Status of the actual value display indicated (1st MCP)							
	Travel keys displayed						MCS/ WCS	Valid display
DBB73 HT 8 → HMI	Status of the actual value display indicated (2nd MCP)							
	Travel keys displayed						MCS/ WCS	Valid display

Notes:

DB10	Signals from (NC -> PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB74 - DBB79 HT 8 → HMI	Machine axis numbers of the displayed axes (1st MCP) MCP1AxisFromHMI							
DBB80 - DBB85 HT 8 → HMI	Machine axis numbers of the displayed axes (2nd MCP) MCP2AxisFromHMI							
DBB86	Reserved							
DBB88	Reserved							

Notes:

DB10	Signals from NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB90 Sinumerik Integr. → PLC								
DBB91 PLC → Sinumerik Integr.								
DBB92 GP → PLC		PN-Bus Supress alarm	DP1 Bus Supress alarm	MPI/DP Bus Supress alarm		PN-Bus Slaves OK	DP1 Bus Slaves OK	MPI/DP Bus Slaves OK
DBB93 HMI → PLC	Collision avoidance: Deactivate protection zone group							
	For operating mode: JOG				For operating mode::AUTO			
	Workpiece	Workholder	Tools	Machine	Workpiece	Workholder	Tools	Machine
DBB94				Warning Direct keys 2 Communication error	Warning Direct keys 1 Communication error	Warning HT Communication error	Warning MCP 2 Communication error	Warning MCP 1 Communication error
DBB95	unassigned							
DBB96 HMI → PLC	Language ID of the HMI Software							
DBB97 HMI → PLC					Channel number for handwheel 1			
					D	C	B	A
DBB98 HMI → PLC					Channel number for handwheel 2			
					D	C	B	A
DBB99 HMI → PLC					Channel number for handwheel 3			
					D	C	B	A
DBB100 HMI → PLC	Machine axis	Handwheel selected	Contour handwheel	Axis number for handwheel 1				
				E	D	C	B	A
DBB101 HMI → PLC	Machine axis	Handwheel selected	Contour handwheel	Axis number for handwheel 2				
				E	D	C	B	A
DBB102 HMI → PLC	Machine axis	Handwheel selected	Contour handwheel	Axis number for handwheel 3				
				E	D	C	B	A
DBB103 HMI → PLC	HMI battery alarm	HMI temperature limit	AT box ready	HMI fan monitoring	HMI HD monitoring			Remote diagnosis active

Notes:

General signals from NC (DB10)

DB10	Signals from NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB104 GP → PLC	NC CPU ready	1. OB1 cycle		Op2Key ready	Op1Key ready	HHU ready	MCP 2 ready	MCP 1 ready
DBB105 GP → PLC								TM Command Cancel
DBB106							Emergency stop active	Collision avoidance Protection area monitoring
DBB107	Inch system	NCU-link active					Probe actuated Probe 2	Probe 1
DBB108	NC ready	Drive ready	Drives in cyclic operation		HMI CPU ready (HMI to OPI)	HMI CPU ready (HMI to MPI)	HMI2 CPU ready E_HMI2 ready	
DBB109	NC battery alarm	Air temp. alarm	NCU heat sink temp. alarm	PC operating system error				NC alarm is active
DBB110	Software cam minus 7	6	5	4	3	2	1	0
DBB111	Software cam minus 15	14	13	12	11	10	9	8
DBB112	Software cam minus 23	22	21	20	19	18	17	16
DBB113	Software cam minus 31	30	29	28	27	26	25	24
DBB114	Software cam minus 7	6	5	4	3	2	1	0
DBB115	Software cam minus 15	14	13	12	11	10	9	8
DBB116	Software cam minus 23	22	21	20	19	18	17	16
DBB117	Software cam minus 31	30	29	28	27	26	25	24
DBB118-121 Sinumerik Integrate → PLC	SINUMERIK Integrate data							

Notes:

External digital inputs of the NC (DB10)

DB10	Signals to NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB122	Disable the external digital NC inputs							
	Input 16	Input 15	Input 14	Input 13	Input 12	Input 11	Input 10	Input 9
DBB123	Values from the PLC for the external digital NC inputs							
	Input 16	Input 15	Input 14	Input 13	Input 12	Input 11	Input 10	Input 9
DBB124	Disable the external digital NC inputs							
	Input 24	Input 23	Input 22	Input 21	Input 20	Input 19	Input 18	Input 17
DBB125	Values from the PLC for the external digital NC inputs							
	Input 24	Input 23	Input 22	Input 21	Input 20	Input 19	Input 18	Input 17
DBB126	Disable the external digital NC inputs							
	Input 32	Input 31	Input 30	Input 29	Input 28	Input 27	Input 26	Input 25
DBB127	Values from the PLC for the external digital NC inputs							
	Input 32	Input 31	Input 30	Input 29	Input 28	Input 27	Input 26	Input 25
DBB128	Disable the external digital NC inputs							
	Input 40	Input 39	Input 38	Input 37	Input 36	Input 35	Input 34	Input 33
DBB129	Values from the PLC for the external digital NC inputs							
	Input 40	Input 39	Input 38	Input 37	Input 36	Input 35	Input 34	Input 33

Notes:

External digital outputs of the NC (DB10)

DB10	Signals to NC (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB130	Disable the external digital NC outputs							
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
DBB131	Overwrite screenform for the external digital NC outputs							
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
DBB132	Value from the PLC for the external digital NC outputs							
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
DBB133	Default screenform for the external digital NC outputs							
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
DBB134	Disable the external digital NC outputs							
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
DBB135	Overwrite screenform for the external digital NC outputs							
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
DBB136	Value from the PLC for the external digital NC outputs							
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
DBB137	Default screenform for the external digital NC outputs							
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
DBB138	Disable the external digital NC outputs							
	Output 32	Output 31	Output 30	Output 29	Output 28	Output 27	Output 26	Output 25
DBB139	Overwrite screenform for the external digital NC outputs							
	Output 32	Output 31	Output 30	Output 29	Output 28	Output 27	Output 26	Output 25
DBB140	Value from the PLC for the external digital NC outputs							
	Output 32	Output 31	Output 30	Output 29	Output 28	Output 27	Output 26	Output 25
DBB141	Default screenform for the external digital NC outputs							
	Output 32	Output 31	Output 30	Output 29	Output 28	Output 27	Output 26	Output 25
DBB142	Disable the external digital NC outputs							
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33
DBB143	Overwrite screenform for the external digital NC outputs							
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33
DBB144	Value from the PLC for the external digital NC outputs							
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33
DBB145	Default screenform for the external digital NC outputs							
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33

Notes:

Analog inputs of the NC (external) (DB10)

DB10	Signals to NC (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB146	Disable the analog NC inputs							
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
DBB147	Specified analog value for NC from PLC							
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
DBW148	Setpoint from PLC for analog input 1 of NC							
DBW150	Setpoint from PLC for analog input 2 of NC							
DBW152	Setpoint from PLC for analog input 3 of NC							
DBW154	Setpoint from PLC for analog input 4 of NC							
DBW156	Setpoint from PLC for analog input 5 of NC							
DBW158	Setpoint from PLC for analog input 6 of NC							
DBW160	Setpoint from PLC for analog input 7 of NC							
DBW162	Setpoint from PLC for analog input 8 of NC							
DBB164 DBB165	unassigned							

Analog outputs of the NC (external) (DB10)

DB10	Signals to NC (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB166	Overwrite screenform for the analog NC outputs							
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB167	Default screenform for the analog NC outputs							
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB168	Disable the analog NC outputs							
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB169	Reserved							
DBW170	Setpoint from PLC for analog output 1 of NC							
DBW172	Setpoint from PLC for analog output 2 of NC							
DBW174	Setpoint from PLC for analog output 3 of NC							
DBW176	Setpoint from PLC for analog output 4 of NC							
DBW178	Setpoint from PLC for analog output 5 of NC							
DBW180	Setpoint from PLC for analog output 6 of NC							
DBW182	Setpoint from PLC for analog output 7 of NC							
DBW184	Setpoint from PLC for analog output 8 of NC							

Notes:

External digital input and output signals of the NC (DB10)

DB10	Signals from NC (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB186	Actual value of external digital NC inputs							
	Input 16	Input 15	Input 14	Input 13	Input 12	Input 11	Input 10	Input 9
DBB187	Actual value of external digital NC inputs							
	Input 24	Input 23	Input 22	Input 21	Input 20	Input 19	Input 18	Input 17
DBB188	Actual value of external digital NC inputs							
	Input 32	Input 31	Input 30	Input 29	Input 28	Input 27	Input 26	Input 25
DBB189	Actual value of external digital NC inputs							
	Input 40	Input 39	Input 38	Input 37	Input 36	Input 35	Input 34	Input 33
DBB190	NC setpoint for external digital NC outputs							
	Output 16	Output 15	Output 14	Output 13	Output 12	Output 11	Output 10	Output 9
DBB191	NC setpoint for external digital NC outputs							
	Output 24	Output 23	Output 22	Output 21	Output 20	Output 19	Output 18	Output 17
DBB192	NC setpoint for external digital NC outputs							
	Output 32	Output 31	Output 30	Output 29	Output 28	Output 27	Output 26	Output 25
DBB193	NC setpoint for external digital NC outputs							
	Output 40	Output 39	Output 38	Output 37	Output 36	Output 35	Output 34	Output 33

Analog input and output signals of the NC (DB10)

DB10	Signals from NC (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW194	Actual value for analog input 1 of the NC							
DBW196	Actual value for analog input 2 of the NC							
DBW198	Actual value for analog input 3 of the NC							
DBW200	Actual value for analog input 4 of the NC							
DBW202	Actual value for analog input 5 of the NC							
DBW204	Actual value for analog input 6 of the NC							
DBW206	Actual value for analog input 7 of the NC							
DBW208	Actual value for analog input 8 of the NC							
DBW210	Setpoint for analog output 1 of the NC							
DBW212	Setpoint for analog output 2 of the NC							
DBW214	Setpoint for analog output 3 of the NC							
DBW216	Setpoint for analog output 4 of the NC							
DBW218	Setpoint for analog output 5 of the NC							
DBW220	Setpoint for analog output 6 of the NC							
DBW222	Setpoint for analog output 7 of the NC							
DBW224	Setpoint for analog output 8 of the NC							

Notes:

Collision avoidance: Protection zone active

DB10	Signals from NC (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB226	Collision avoidance: Protection zone active							
	7	6	5	4	3	2	1	0
DBB227	Collision avoidance: Protection zone active							
	15	14	13	12	11	10	9	8
DBB228	Collision avoidance: Protection zone active							
	23	22	21	20	19	18	17	16
DBB229	Collision avoidance: Protection zone active							
	31	30	29	28	27	26	25	24
DBB230	Collision avoidance: Protection zone active							
	39	38	37	36	35	34	33	32
DBB231	Collision avoidance: Protection zone active							
	47	46	45	44	43	42	41	40
DBB232	Collision avoidance: Protection zone active							
	55	54	53	52	51	50	49	48
DBB233	Collision avoidance: Protection zone active							
	63	62	61	60	59	58	57	56

Collision avoidance: Activate protection zone

DB10	Signals to NC (PLC → NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB234	Collision avoidance: Activate protection zone							
	7	6	5	4	3	2	1	0
DBB235	Collision avoidance: Activate protection zone							
	15	14	13	12	11	10	9	8
DBB236	Collision avoidance: Activate protection zone							
	23	22	21	20	19	18	17	16
DBB237	Collision avoidance: Activate protection zone							
	31	30	29	28	27	26	25	24
DBB238	Collision avoidance: Activate protection zone							
	39	38	37	36	35	34	33	32
DBB239	Collision avoidance: Activate protection zone							
	47	46	45	44	43	42	41	40
DBB240	Collision avoidance: Activate protection zone							
	55	54	53	52	51	50	49	48
DBB241	Collision avoidance: Activate protection zone							
	63	62	61	60	59	58	57	56

Notes:

Expansion handwheel signals from NC (DB10)

DB10	Signals from NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB242	Handwheel 4 moved							
DBB243	Handwheel 5 moved							
DBB244	Handwheel 6 moved							
DBB245			Ethernet handwheel is stationary					
			Handwheel 6	Handwheel 5	Handwheel 4	Handwheel 3	Handwheel 2	Handwheel 1
DBB246			Handwheel active					
			Handwheel 6	Handwheel 5	Handwheel 4	Handwheel 3	Handwheel 2	Handwheel 1

Interface, Roboter status (DB10)

DB10	Signals to NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW248	Roboter status byte 0							
DBW249	Roboter status byte 1							
DBW250	Roboter status byte 2							
DBW251	Roboter status byte 3							
DBW252	Roboter status byte 4							
DBW253	Roboter status byte 5							
DBW254	Roboter status byte 6							
DBW255	Roboter status byte 0							

Interface, Roboter control (DB10)

DB10	Signals from NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW256	Roboter control byte 0							
DBW257	Roboter control byte 1							
DBW258	Roboter control byte 2							
DBW259	Roboter control byte 3							
DBW260	Roboter control byte 4							
DBW261	Roboter control byte 5							
DBW262	Roboter control byte 6							
DBW263	Roboter control byte 0							

Notes:

Mode group-specific signals (DB11)

DB11	Signals to mode group 1 (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Mode group reset	BAG-Stopp Mode group stop Axes plus spindle	Mode group stop	Mode change disable		Operating mode JOG	MDA	AUTO
DBB1	Single block Type A	Type B				Machine function REF	REPOS	TEACH IN
DBB2	Machine function	Continuous travel active	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB3	unassigned							

DB11	Signals from mode group 1 (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB4 HMI → PLC						Strobe mode JOG	MDA	AUTO
DBB5 HMI → PLC						Strobe machine function REF	REPOS	TEACH IN
DBB6	All channels in reset state		NC internal JOG active	Mode Group reseted	Mode group ready	Active operating mode JOG	MDA	AUTO
DBB7						Active machine function REF	REPOS	TEACH IN
DBB8	Machine function	Continuous travel active	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

DB11	Signals from mode group 1 (HMI → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB10 HMI → PLC	Selection from HMI	Continuous travel active	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB11	unassigned							

Notes:

DB11	Signals to mode group 2 (PLC→NC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB20	Mode group reset	Mode group stop Axes plus spindle	Mode group stop	Mode change disable		Operating mode		
						JOG	MDA	AUTO
DBB21	Single block					Machine function		
	Type A	Type B				REF	REPOS	TEACH IN
DBB22	Machine function							
		Continuous travel active	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB23	unassigned							

DB11	Signals from mode group 2 (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB24 HMI → PLC						Strobe mode		
						JOG	MDA	AUTO
DBB25 HMI → PLC						Strobe machine function		
						REF	REPOS	TEACH IN
DBB26	All channels in reset state		JOG is active internally in the NC	Mode group reset per-formed	Mode group ready	Active operating mode		
						JOG	MDA	AUTO
DBB27						Active machine function		
						REF	REPOS	TEACH IN
DBB28	Machine functions							
		Continuous travel active	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

DB11	Signals from mode group 2 (HMI→PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						JOG	MDA	AUTO
DBB29	unassigned							
DBB30 HMI → PLC	Machine function							
		Continuous travel active	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC

Notes:

Signals from/to operator panel (DB19)

DB19	Signals to operator panel (PLC → HMI)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Actual value in WCS 0=MCS	Save action log	HMI-Adv Shutdown	Delete recall alarms	Delete cancel alarms	Key disable	Darken screen	Brighten screen
DBB1							Rights for the external viewer	External Viewer
DBB2								
DBB4								
DBB6	Analog spindle 1, utilization as a percentage							
DBB7	Analog spindle 2, utilization as a percentage							
DBB8	Channel number of machine control panel to HMI							
DBB9	Reserved selection					Auto tool measure	OEM2	OEM1
DBB10	Hardkeys (Values 1 ... 255, Default: 0)							
DBB11	Reserved							
DBB12								
DBB13	Part program Selection	Loading	Unloading	Reserved				Disable Teach trans-fer
DBB14	0=act. FS 1=pas. FS	RS-232 act. FS: Index of file to be transferred in the standard list. RS-232 pass. FS: Number of the control file for user file names.						
DBB15	RS-232 act. FS: Index that specifies the axis, channel or tool no. RS-232 pass. FS: Index of the file to be transferred in the user list							
DBB16	0=act. FS 1=pas. FS	Program selection from the PLC: Index of the program list						
DBB17	Program selection from the PLC: Program index in the program list							
DBB18								
DBB19	Reserved (message counter)							

Notes:

DB19	Signals from operator panel (HMI → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB20	MCS/WCS Change-over	Simulation active	Language 2 switched	Recall alarm cleared	Cancel alarm cleared	Cancel key actuated	Screen is dark	
DBB21	Current number of the active operating area							
DBB22	Current channel number							
DBB23						Screen change active	Data transfer active	Keyboard has been used
DBW24	Current screen number							
DBB25								
DBB26	Program selection from the PLC: Status signals							
	Select	Load	Unload		Active	Error	OK	Reserved
DBB27	Program selection from the PLC: Error identification							
DBW28	Mask number for "Supplement user interface"							
DBB30 PLC → HMI	Screen selection from the PLC: Control signals							
							Screen dese-lection	Screen selec-tion
DBB31 HMI → PLC	Screen selection from the PLC: Status signals							
	Inactive			Error, Screen selection not possible	Screen is deselected	Screen active	Screen is se-lected	Screen selec-tion accepted
DBB32 PLC → HMI	Busy func-tion	Strobe func-tion	Function selection no. from the PLC					
DBB33 PLC → HMI	Parameter 1 for function selection no. (function selection from DBB32)							
DBB34 PLC → HMI	Parameter 2 for function selection no. (function selection from DBB32)							
DBB35 PLC → HMI	Parameter 3 for function selection no. (function selection from DBB32)							
DBB36 HMI → PLC	Error code for function selection no. (function selection from DBB32)							
DBB37 HMI → PLC	Parameter 1 for function selection no. (function selection from DBB48)							
DBB38 HMI → PLC	Parameter 2 for function selection no. (function selection from DBB48)							

Notes:

DB19	Signals from operator panel (HMI → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB39 HMI → PLC	Parameter 3 for function selection no. (function selection from DBB48)							
DBB40 - DBB47	Reserved							
DBB48 HMI → PLC	PLC busy function	HMI strobe function	Function selection no. from HMI					
DBB49 PLC → HMI	Error code for function selection no. (function selection from DBB48)							
DBB50 - DBB99	Interface 2nd HMI Assignment as for DBB0 to DBB49							
DBB100	Switchover interface to HMI Knocking interface (HMI announces itself to NCU) ONL_REQUEST Online request from HMI HMI writes its client identification as online request (bit 8-15: bus type, bit 0-7: HMI bus address)							
DBB102	ONL_CONFIRM Acknowledgment from PLC to online request PLC writes HMI client identification as acknowledgment (bus type, HMI bus address; as with DBB100).							
DBB104	PAR_CLIENT_IDENT HMI writes its client identification (bus type, HMI bus address; as with DBB100).							
DBB106	PAR_HMI_TYP Type of HMI as per NETNAMES.INI: Main / subordinate operator panel / server /…							
DBB107	PAR_MSTT_ADR HMI writes address of MCP to be activated; 255, when no MCP activated							
DBB108	PAR_STATUS PLC writes online enable for HMI							
DBB109	PAR_Z_INFO PLC writes additional info about status							
DBB110	M_TO_N_ALIVE Sign of life from PLC to HMI through M to N block							
DBB112	Reserved bus type MCP							
DBB113	ParOpKeyAdr Direct key index knocking interface							
DBB114	ParTcuIndex TCU index knocking interface							
DBB115	ParHt2Index Ht2 index login interface							

Notes:

DB19	Signals from operator panel (HMI → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB116	Direct key address of 1st Online interface							
DBB117	Direct key address of 2nd Online interface							
DBB118	TCU index of 1st Online interface							
DBB119	TCU index of 2nd Online interface							
DBB120	Online interface HMI 1 (user) MMC1_CLIENT_IDENT PLC writes PAR_CLIENT_IDENT to MMCx_CLIENT_IDENT when HMI goes online.							
DBB122	MMC1_TYP PLC writes PAR_MMC_TYP to MMCx_TYP when HMI goes online.							
DBB123	MMC1_MSTT_ADR PLC writes PAR_MSTT_ADR to MMCx_MSTT_ADR when HMI goes online.							
DBB124	MMC1_STATUS Connection status, HMI and PLC alternately write their requests/acknowledgments							
DBB125	MMC1_Z_INFO Additional info connection status (pos./neg. acknowledgment, error messages...)							
DBB126	Reserved	TCU1_ SHIFT_ LOCK	MMC1_ CHANGE_ DENIED	MMC1_ ACTIVE_ CHANGED	MMC1_ ACTIVE_ PERM	MMC1_ ACTIVE_ REQ	MMC1_ MSTT_ SHIFT_ LOCK	MMC1_ SHIFT_ LOCK
DBB127	Reserved Bus type MCP							
DBB128 - DBB129	Reserved Transline (Transline DB number)							
DBB130	Online interface HMI 2 (user) MMC2_CLIENT_IDENT PLC writes PAR_CLIENT_IDENT to MMCx_CLIENT_IDENT when HMI goes online							
DBB132	MMC2_TYP PLC writes PAR_MMC_TYP to MMCx_TYP when HMI goes online							
DBB133	MMC2_MSTT_ADR PLC writes PAR_MSTT_ADR to MMCx_MSTT_ADR when HMI goes online							
DBB134	MMC2_STATUS Connection status, MMC and PLC alternately write their requests/acknowledgments							
DBB135	MMC2_Z_INFO /Z2-B3/ Additional info connection status (pos./neg. acknowledgment, error messages...)							
DBB136	Reserved	TCU2_ SHIFT_ LOCK	MMC2_ CHANGE_ DENIED	MMC2_ ACTIVE_ CHANGED	MMC2_ ACTIVE_ PERM	MMC2_ ACTIVE_ REQ	MMC2_ MSTT_ SHIFT_ LOCK	MMC2_ SHIFT_ LOCK

Notes:

DB19	Signals from operator panel (HMI → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB137	Reserved Bus type MCP							
DBB138 - DBB139	Reserved Transline (Transline DB number)							
DBB140 - DBB197	Assignment of the PLC for transfer parameters These data blocks are reserved for the “Tool Ident Connection” option.							
DBB198 - DBB249	Assignment of the PLC for return values These data blocks are reserved for the “Tool Ident Connection” option.							
DBB250 - DBB255	Function call of the PLC interface These data blocks are reserved for the “Tool Ident Connection” option.							
DBB256 - DBB267	Commands for Paramtm.exe							
DBB268	Traffic light status							
DBD270 - DBD394	Counter[1 ... 32]							
DBB398	Handwheel number for simulation override							
DBW400	Simulation override							
DBW402	Simulation state							

DB19	Signals from sidescreen (HMI → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB404	Customer keys 8	7	6	5	4	3	2	1
DBB405	Customer keys 16	15	14	13	12	11	10	9
DBB406	Customer keys 24	23	22	21	20	19	18	17
DBB407	Customer keys 32	31	30	29	28	27	27	25
DBB408	Customer keys 40	41	42	43	44	45	46	47
DBB409	Customer keys 48	47	46	45	44	43	42	41
DBB410	Customer keys 56	55	54	53	52	51	50	49
DBB411	Customer keys 64	63	62	61	60	59	58	57

Notes:

DB19	Signals from sidescreen (PLC → HMI)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB412	Customer LED							
	8	7	6	5	4	3	2	1
DBB413	Customer LED							
	16	15	14	13	12	11	10	9
DBB414	Customer LED							
	24	23	22	21	20	19	18	17
DBB415	Customer LED							
	32	31	30	29	LED	27	27	25
DBB416	Customer LED							
	40	41	42	43	44	45	46	47
DBB417	Customer LED							
	48	47	46	45	44	43	42	41
DBB418	Customer LED							
	56	55	54	53	52	51	50	49
DBB419	Customer LED							
	64	63	62	61	60	59	58	57
DBB420	Customer key deactivated							
	8	7	6	5	4	3	2	1
DBB421	Customer key deactivated							
	16	15	14	13	12	11	10	9
DBB422	Customer key deactivated							
	24	23	22	21	20	19	18	17
DBB423	Customer key deactivated							
	32	31	30	29	LED	27	27	25
DBB424	Customer key deactivated							
	40	41	42	43	44	45	46	47
DBB425	Customer key deactivated							
	48	47	46	45	44	43	42	41
DBB426	Customer key deactivated							
	56	55	54	53	52	51	50	49
DBB427	Customer key deactivated							
	64	63	62	61	60	59	58	57

Notes:

PLC machine data (User DB)

User DB for INT values, Bit arrays and REAL values

User DB	PLC machine data (PLC → operator)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW	INT values							
DBW								
DBW	INT values							
DBB	Bit arrays							
DBB								
DBB	Bit arrays							
DBD	REAL values							
DBD								
DBD	REAL values							

Notes:

User DB for INT values

User DB	PLC machine data (PLC → operator)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW	INT values							
DBW								
DBW	INT values							

User DB for Bit arrays

User DB	PLC machine data (PLC → operator)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB	Bit arrays							
DBB								
DBB	Bit arrays							

User DB for REAL values

User DB	PLC machine data (PLC → operator)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBD	REAL values							
DBD								
DBD	REAL values							

Notes:

Channel-specific signals (DB21 - DB30)

Signals to/from NC channel

DB21 - DB30	Signals to NC channel (PLC→ NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0		Activate dry run feedrate	Activate M01	Activate single block	Activate DRF			
DBB1	Activate program test	PLC action completed	CLC correction	CLC stop	Time monitoring active	Disable all synchronized actions	Enable protection zones	Activate referencing
DBB2	Skip block activate							
	/7	/6	/5	/4	/3	/2	/1	/0
DBB3	Nibbling and punching							
			Manual stroke 2 initiation	Stroke inoperative	Delayed stroke	Stroke suppression	Manual stroke initiation	Stroke enable
DBB4	Feedrate override							
	H	G	F	E	D	C	B	A
DBB5	Rapid traverse override							
	H	G	F	E	D	C	B	A
DBB6	Feedrate override active	Rapid traverse override active		Program level abort	Delete subroutine no. of passes	Delete distance-to-go	Read-in disable	Feed disable
DBB7	Reset	Activate configured stop	Suppress Start Lock	NC Stop axes plus spindle	NC Stop	NC Stop to block limit	NC Start	NC Start disable
DBB8	Activate machine-related protection area							
	Area 8	Area 7	Area 6	Area 5	Area 4	Area 3	Area 2	Area 1
DBB9	Activate machine-related protection area							
							Area 10	Area 9
DBB10	Activate channel-specific protection area							
	Area 8	Area 7	Area 6	Area 5	Area 4	Area 3	Area 2	Area 1
DBB11	Activate channel-specific protection area							
							Area 10	Area 9

Notes:

Control signals to geometry axes

DB21 - DB30	Signals to NC channel (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB12	**Geometry axis 1**							
	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel		
	pus	minus				3	2	1
DBB13	machine function							
		Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB14	OEM signals							
DBB15								Invert handwheel direction of rotation
DBB16	**Geometry axis 2**							
	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel		
	plus	minus				3	2	1
DBB17	machine function							
		Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB18	OEM signals							
DBB19								Invert handwheel direction of rotation
DBB20	**Geometry axis 3**							
	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel		
	plus	minus				3	2	1
DBB21	machine function							
		Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB22	OEM signals							
DBB23								Invert handwheel direction of rotation

Notes:

Operating signals from HMI/status signals from NC channel

DB21 - DB30	Signals from NC channel (HMI→PLC, PLC→NC, NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB24 HMI →PLC		Dry run feedrate selected	M01 selected	Select NC-related M01	DRF selected	Set to 0% for Stop Override	Configured stop selected	
DBB25 HMI →PLC	Program test selected			REPOS mode change	Feedrate override selected for rapid traverse	REPOS-Mode C	B	A
DBB26 HMI →PLC	Skip block selected /7	/6	/5	/4	/3	/2	/1	/0
DBB27 HMI →PLC							Skip block selected /9	/8
DBB28 PLC→NC	OEM signals: Requirement							
DBB29 PLC→NC	Do not disable tool	Switch off wear monitoring	Switch off workpiece counter	Activate PTP motion	Activate fixed feed 4	3	2	1
DBB30 PLC→NC	No tool change commands	Jog circles	Activate NC-related M01	Neg. direction simulation contour handwheel	Simulation contour handwheel on	Activate contour handwheel 3	2	1
DBB31 PLC→NC	Skip block active /9	/8	Invert contour handwheel direction of rotation	REPOS mode change		REPOS-Mode C	B	A
DBB32 NC→PLC		Last action block active	M00/M01 active	Approach block active	Action block active			Execution from external active
DBB33 NC→PLC	Program test active	Transformation active	M02/M30 active	Block search active	Handwheel override active	Rotational feedrate active	Orientable toolholder active	Referencing active
DBB34 NC→PLC	OEM signals: Feedback							
DBB35 NC→PLC	Channel status Reset	Interrupted	Active	Program status Aborted	Interrupted	Stopped	Waiting	Running
DBB36 NC→PLC	NC alarm with processing stop present	Channel-specific NC alarm present	Channel ready for operation	Interrupt processing active	All axes stationary	All axes requiring reference points are referenced		

Notes:

DB21 - DB30	Signals from NC channel (HMI→PLC, PLC→NC, NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB37 NC→PLC	Stop at block end with SBL is suppressed	Read-in disable is ignored	CLC stopped upper limit	CLC stopped lower limit	CLC active	Contour handwheel active 3	2	1
DBB38 NC→PLC	Nibbling and punching						Acknowledge-ment manual stroke release	Stroke re-lease active
DBB39 NC→PLC	Stop at end of block due to configured stop	Configured stop is acti-vated	Contour handwheel direction of rotation in-verted active	Stop at end of block due to single block			NC alarm with program stop	Protection zone monitor-ing not guar-anteed

Notes:

Status signals of geometry axes

DB21 – DB30	Signals from NC channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB40	**Geometry axis 1**							
	Traverse command		Travel requests			Handwheel active		
	plus	minus	plus	minus		3	2	1
DBB41	active machine function							
		Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB42	OEM signals							
DBB43								Handwheel direction of rotation inverted
DBB44 HMI → PLC								
DBB46	**Geometry axis 2**							
	Traverse command		Travel requests			Handwheel active		
	plus	minus	plus	minus		3	2	1
DBB47	active machine function							
		Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB48	OEM signals							
DBB49								Handwheel direction of rotation inverted
DBB50 HMI → PLC								
DBB52	**Geometry axis 3**							
	Traverse command		Travel requests			Handwheel active		
	plus	minus	plus	minus		3	2	1
DBB53	active machine function							
		Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB54	OEM signals							
DBB55								Handwheel direction of rotation inverted
DBB56 HMI → PLC								

Notes:

Change signals on auxiliary function transfer from NC channel

DB21 – DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB58				M fct. 5 change	M fct. 4 change	M fct. 3 change	M fct. 2 change	M fct. 1 change
DBB59				M fct. 5 not decoded	M fct. 4 not decoded	M fct. 3 not decoded	M fct. 2 not decoded	M fct. 1 not decoded
DBB60		S fct. 3 quick	S fct. 2 quick	S fct. 1 quick		S fct. 3 change	S fct. 2 change	S fct. 1 change
DBB61		T fct 3 quick	T fct. 2 quick	T fct. 1 quick		T fct. 3 change	T fct. 2 change	T fct. 1 change
DBB62		D fct. 3 quick	D fct. 2 quick	D fct. 1 quick		D fct. 3 change	D fct. 2 change	D fct. 1 change
DBB63				DL fct. quick				DL fct. change
DBB64		H fct. 3 quick	H fct. 2 quick	H fct. 1 quick		H fct. 3 change	H fct. 2 change	H fct. 1 change
DBB65			F fct. 6 change	F fct. 5 change	F fct. 4 change	F fct. 3 change	F fct. 2 change	F fct. 1 change
DBB66				M fct. 5 quick	M fct. 4 quick	M fct. 3 quick	M fct. 2 quick	M fct. 1 quick
DBB67			F fct. 6 quick	F fct. 5 quick	F fct. 4 quick	F fct. 3 quick	F fct. 2 quick	F fct. 1 quick

Notes:

Transferred M/S functions

DB21 – DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW68	Extended address M function 1 (binary)							
DBD70	M function 1 (binary)							
DBW74	Extended address M function 2 (binary)							
DBD76	M function 2 (binary)							
DBW80	Extended address M function 3 (binary)							
DBD82	M function 3 (binary)							
DBW86	Extended address M function 4 (binary)							
DBD88	M function 4 (binary)							
DBW92	Extended address M function 5 (binary)							
DBD94	M function 5 (binary) /Z1-H2/							
DBW98	Extended address S function 1 (binary)							
DBD100	S function 1 (REAL format)							
DBW104	Extended address S function 2 (binary)							
DBD106	S function 2 (REAL format)							
DBW110	Extended address S function 3 (binary)							
DBD112	S function 3 (REAL format)							

Notes:

Transferred T/D/DL functions

DB21 - DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW116	Extended address T function 1 (16 bit INT)							
DBW118 DBD118	T function 1 (binary) For 8-decade T nos., T function 1 (32 bit DINT) is used in DBD 118 (see note)							
DBW120	Extended address T function 2 (16 bit INT)							
DBW122	T function 2 (Int)							
DBW124	Extended address T function 3 (16 bit INT)							
DBW126	T function 3 (Int)							
DBB128	Extended address D function 1 (8 bit INT)							
DBB129	D function 1 (binary)							
DBW130 DBB130	For 5-decade D nos., D function 1 (16 bit DINT) is used in DBD 130 (see note) Extended address D function 2 (8 bit INT)							
DBB131	D function 2 (8 bit Int)							
DBB132	Extended address D function 3 (8 bit INT)							
DBB133	D function 3 (8 bit Int)							
DBW134	Extended address DL function (16 bit INT)							
DBD136	DL function (REAL)							

Notes:

Transferred H/F functions

DB21 - DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW140	Extended address H function 1 (binary)							
DBD142	H function 1 (REAL or Dint)							
DBW146	Extended address H function 2 (binary)							
DBD148	H function 2 (REAL or Dint)							
DBW152	Extended address H function 3 (binary)							
DBD154	H function 3 (REAL or Dint)							
DBW158	Extended address F function 1 (binary)							
DBD160	F function 1 (REAL format)							
DBW164	Extended address F function 2 (binary)							
DBD166	F function 2 (REAL format)							
DBW170	Extended address F function 3 (binary)							
DBD172	F function 3 (REAL format)							
DBW176	Extended address F function 4 (binary)							
DBD178	F function 4 (REAL format)							
DBW182	Extended address F function 5 (binary)							
DBD184	F function 5 (REAL format)							
DBW188	Extended address F function 6 (binary)							
DBD190	F function 6 (REAL format)							

Notes:

Decoded M signals (M0–M99)

DB21 – DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB194	Dynamic M functions							
	M07	M06	M05 *	M04 *	M03 *	M02	M01	M00
DBB195	Dynamic M functions							
	M15	M14	M13	M12	M11	M10	M09	M08
DBB196	Dynamic M functions							
	M23	M22	M21	M20	M19	M18	M17	M16
DBB197	Dynamic M functions							
	M31	M30	M29	M28	M27	M26	M25	M24
DBB198	Dynamic M functions							
	M39	M38	M37	M36	M35	M34	M33	M32
DBB199	Dynamic M functions							
	M47	M46	M45	M44	M43	M42	M41	M40
DBB200	Dynamic M functions							
	M55	M54	M53	M52	M51	M50	M49	M48
DBB201	Dynamic M functions							
	M63	M62	M61	M60	M59	M58	M57	M56
DBB202	Dynamic M functions							
	M71	M70 *	M69	M68	M67	M66	M65	M64
DBB203	Dynamic M functions							
	M79	M78	M77	M76	M75	M74	M73	M72
DBB204	Dynamic M functions							
	M87	M86	M85	M84	M83	M82	M81	M80
DBB205	Dynamic M functions							
	M95	M94	M93	M92	M91	M90	M89	M88
DBB206	Dynamic M functions							
					M99	M98	M97	M96
DBB207								

Notes:

Active G functions

DB21 - DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB208	Number of active G function of G function group 1 (binary)							
DBB209	Number of active G function of G function group 2 (binary)							
DBB210	Number of active G function of G function group 3 (binary)							
DBB211	Number of active G function of G function group 4 (binary)							
DBB212	Number of active G function of G function group 5 (binary)							
DBB213	Number of active G function of G function group 6 (binary)							
DBB214	Number of active G function of G function group 7 (binary)							
DBB215	Number of active G function of G function group 8 (binary)							
...	...							
DBB270	Number of active G function of G function group n-1 (binary)							
DBB271	Number of active G function of G function group n (binary)							

Signals for protection areas from NC channel

DB21 - DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB272	Machine-related protection area preactivated							
	Area 8	Area 7	Area 6	Area 5	Area 4	Area 3	Area 2	Area 1
DBB273	Machine-related protection area preactivated							
							Area 10	Area 9
DBB274	Channel-specific protection area preactivated							
	Area 8	Area 7	Area 6	Area 5	Area 4	Area 3	Area 2	Area 1
DBB275	Channel-specific protection area preactivated							
							Area 10	Area 9
DBB276	Machine-related protection area violated							
	Area 8	Area 7	Area 6	Area 5	Area 4	Area 3	Area 2	Area 1
DBB277	Machine-related protection area violated							
							Area 10	Area 9
DBB278	Channel-related protection area violated							
	Area 8	Area 7	Area 6	Area 5	Area 4	Area 3	Area 2	Area 1
DBB279	Channel-related protection area violated							
							Area 10	Area 9

Notes:

Instruction controlled signals to NC channel

DB21 – DB30	Signals to NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB280							Synch. action disable re-quest	Reserved
DBB281							Synch. action disabled	
DBW282 - DBW298	Reserved							
DBB300	Disable synchronized actions							
	No. 8	No. 7	No. 6	No. 5	No. 4	No. 3	No. 2	No. 1
DBB301	Disable synchronized actions							
	No. 16	No. 15	No. 14	No. 13	No. 12	No. 11	No. 10	No. 9
DBB302	Disable synchronized actions							
	No. 24	No. 23	No. 22	No. 21	No. 20	No. 19	No. 18	No. 17
DBB303	Disable synchronized actions							
	No. 32	No. 31	No. 30	No. 29	No. 28	No. 27	No. 26	No. 25
DBB304	Disable synchronized actions							
	No. 40	No. 39	No. 38	No. 37	No. 36	No. 35	No. 34	No. 33
DBB305	Disable synchronized actions							
	No. 48	No. 47	No. 46	No. 45	No. 44	No. 43	No. 42	No. 41
DBB306	Disable synchronized actions							
	No. 56	No. 55	No. 54	No. 53	No. 52	No. 51	No. 50	No. 49
DBB307	Disable synchronized actions							
	No. 64	No. 63	No. 62	No. 61	No. 60	No. 59	No. 58	No. 57

Notes:

DB21 - DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB308	Synchronized actions can be disabled Nr. 8	Nr. 7	Nr. 6	Nr. 5	Nr. 4	Nr. 3	Nr. 2	Nr.1
DBB309	Synchronized actions can be disabled Nr. 16	Nr. 15	Nr. 14	Nr. 13	Nr. 12	Nr. 11	Nr.10	Nr.9
DBB310	Synchronized actions can be disabled Nr. 24	Nr. 23	Nr. 22	Nr. 21	Nr. 20	Nr. 19	Nr.18	Nr.17
DBB311	Synchronized actions can be disabled Nr. 32	Nr. 31	Nr. 30	Nr. 29	Nr. 28	Nr. 27	Nr. 26	Nr.25
DBB312	Synchronized actions can be disabled Nr. 40	Nr. 39	Nr. 38	Nr. 37	Nr. 36	Nr. 35	Nr. 34	Nr. 33
DBB313	Synchronized actions can be disabled Nr. 48	Nr. 47	Nr. 46	Nr. 45	Nr. 44	Nr. 43	Nr. 42	Nr.41
DBB314	Synchronized actions can be disabled Nr. 56	Nr. 55	Nr. 54	Nr. 53	Nr. 52	Nr. 51	Nr. 50	Nr.49
DBB315	Synchronized actions can be disabled Nr. 64	Nr. 63	Nr. 62	Nr. 61	Nr. 60	Nr. 59	Nr. 58	Nr.57
Cyclic Signals								
DBB316	Active G functions							G00 geo.
DBB317	Tool missing	PTP motion active	Travel request drive test				Workpiece setpoint reached	External language mode active
DBB318	Overstore active	Dry-run feedrate active	AssociatedM01 active	Delayed stop	TOFF movement active	TOFF active	Search active	ASUP stopped
DBB319	No tool change command active	Stop-delay-range not activated	Repos delay	Delay FTS	Active REPOS mode C	B	A	Acknowledgement of the REPOS mode

Notes:

Signals to/from orientation axes

DB21 - DB30	Signals to NC channel (PLC→NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB320	**Orientation axis 1**							
	Traversing keys plus	Traversing keys minus	Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel (bit value coding)		
DBB321	machine function							
		Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB322	OEM signals							
DBB323								Invert handwheel direction of rotation
DBB324	**Orientation axis 2**							
	Traversing keys plus	Traversing keys minus	Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel (bit value coding)		
DBB325	machine function							
		Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB326	OEM signals							
DBB327								Invert handwheel direction of rotation
DBB328	**Orientation axis 3**							
	Traversing keys plus	Traversing keys minus	Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel (bit value coding)		
DBB329	machine function							
		Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB330	OEM signals							
DBB331								Invert handwheel direction of rotation

Notes:

DB21 - DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB332	**Orientation axis 1** Travel command plus	minus	Travel request plus	minus		Handwheel active (bit value coding)		
DBB333	Active machine function	Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB334	OEM signals							
DBB335								Handwheel direction of rotation inverted
DBB336	**Orientation axis 2** Travel command plus	minus	Travel request plus	minus		Handwheel active (bit value coding)		
DBB337	Active machine function	Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB338	OEM signals							
DBB339								Handwheel direction of rotation inverted
DBB340	**Orientation axis 3** Travel command plus	minus	Travel request plus	minus		Handwheel active (bit value coding)		
DBB341	Active machine function	Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB342	OEM signals							
DBB343								Handwheel direction of rotation inverted

Notes:

Tool management functions from NC channel

DB21 - DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Modification signals tool management functions								
DBB344					Last replace-ment tool of tool group	Transfer to new replace-ment tool	Tool limit value reached	Tool pre-warning limit reached
DBB345 - DBB347								
	Transferred tool management functions							
DBD348	T number for tool prewarning limit (DINT)							
DBD352	T number for tool limit value (DINT)							
DBD356	T number of new replacement tool (DINT)							
DBD360	T number of last replacement tool (DINT)							

Notes:

DB21 – DB30	Signals from NC channel (NC→PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB364	CH_CYCLES_SIG_IN (Bit 0 - 7)							
DBB365	CH_CYCLES_SIG_IN (Bit 8 - 15)							
DBB366	CH_CYCLES_SIG_OUT (Bit 0 - 7)							
DBB367	CH_CYCLES_SIG_OUT (Bit 8 - 15)							
DBB368	CH_OEM_TECHNO_SIG_IN (DBB368 - DBB371)							
DBB369								
DBB370								
DBB371								
DBB372	CH_OEM_TECHNO_SIG_OUT (DBB372 - DBB375)							
DBB373								
DBB374								
DBB375								
DBB376	ProgEventDisplay							
DBB377		Jog circle active	Retraction data available	JOG retract active			Stop condition	Collision avoidance Stop
DBB378							Silent ASUP is active	ASUP is active
DBB379								
DBB380	Reserved ASUP							
DBB381	Reserved ASUP							
DBB382	Reserved ASUP							
DBB383	Reserved ASUP							

Notes:

DB21 - DB30	Signals to NC channel (PLC → NC, NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB384 PLC→ NC								Enable GOTOS
DBB385	Grinding: Input signals 1 … 8 ($AC_IN_KEY_G[1 … 8]							
	8	7	6	5	4	3	2	1
DBB386	Grinding: Disable input signals 1 … 8							
	8	7	6	5	4	3	2	1
DBB387	Grinding: Status of the grinding functions 1 … 8 ($AC_IN_KEY_G_RUN_IN[1 … 8]							
	8	7	6	5	4	3	2	1
DBW388 NC → PLC	Active transformation number							
DBB390 PLC → NC	Grinding: Enable status of the input signals 1 … 8 ($AC_IN_KEY_G_ISENABLE[1 … 8]							
	8	7	6	5	4	3	2	1
DBB391 NC → PLC	Grinding: Status of the grinding functions 1 … 8 ($AC_IN_KEY_G_RUN_OUT[1 … 8]							
	8	7	6	5	4	3	2	1
DBB392 PLC → NC	Selection of coordinate system for Cartesian manual traversing and handwheel override in the automatic mode in the tool direction (DRF)							
DBB393-395	Reserved							
DBD396 NC → PLC	H function 1 (DINT)							
DBD400 NC → PLC	H function 2 (DINT)							
DBD404 NC → PLC	H function 3 (DINT)							
DBD408 HMI → PLC	Axis selection type: 0: Machine axis, 1: Geo axis							
DBB409 HMI → PLC	Axis number: Machine-/Geo axis							
DBB410 HMI → PLC								Axis selection key
DBB411	Reserved							
DBB412 PLC → HMI	Active axis selection: 0: Machine axis, 1: Geo axis							
DBB413 PLC → HMI	Active axis number: Machine-/Geo axis							

Notes:

Signals from/to axis/spindle (DB31 – DB61)

DB31 – DB61	Signals to axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0 Axis and spindle	Feedrate override H	G	F	E	D	C	B	A
DBB1 Axis and spindle	Override active	Position measuring system 2	Position measuring system 1	Follow-up mode	Axis/spindle disable	Sensor fixed stop	Acknowl. fixed stop reached	Acknowl. fixed stop reached
DBB2 Axis and spindle	Reference point value 4	3	2	1	Clamping in progress	Delete distance-to-go/ spindle reset	Controller enable	Cam activation
DBB3 Axis and spindle	Program test axis/ spindle release	Velocity/ spindle speed limitation	Activate fixed feed 4	3	2	1	Enable travel to fixed stop	Accept external ZO
DBB4 Axis and spindle	Traversing keys plus	minus	Rapid traverse override	Traversing key disable	Feed stop/ spindle stop	Activate handwheel 3	2	1
DBB5 Axis and spindle	Machine function	Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB6	OEM signals							
DBB7	OEM signals							Invert handwheel direction of rotation
DBB8	Request PLC axis/spindle			Channel assignment changed	Allocate NC axis to channel D	C	B	A
	Note: DBX8.4: is automatically reset after assignment.							
DBB9					Parameter set change inhibited	Control parameter set C	B	A
DBB10								REPOS Delay

Notes:

DB31 – DB61	Signals to axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB11								SI: Start brake test
DBB12 Axis	Delay reference point approach			Modulo rotary axes: Activate traversing range limits	2nd software limit switch plus	2nd software limit switch minus	Hardware limit switch plus	Hardware limit switch minus
DBB13 Axis					Jogging to position	JOG fixed-point approach 2	JOG fixed-point approach 1	JOG fixed-point approach 0
DBB14 Axis							Activate program test	Suppress program test
DBB15 Axis								
DBB16 Spindle	Delete S value	No speed monitoring during gear change /Z1-S1/	Resynchronize spindle 2	Resynchronize spindle 1	Gear is changed over	Actual gear stage C	Actual gear stage B	Actual gear stage A
DBB17 Spindle		Invert M3/M4	Resynchronize spindle during positioning 2	Resynchronize spindle during positioning 1				Feedrate override f. spindle valid
DBB18 Spindle	Oscillation direct. of rotation left	Oscillation direct. of rotation right	Oscillation enable	Oscillation controlled by PLC				
DBB19 Spindle	Spindle override H	Spindle override G	Spindle override F	Spindle override E	Spindle override D	Spindle override C	Spindle override B	Spindle override A
DBB20 Drive			Open motor holding brake				Ramp function generator disable	
DBB21 Drive	Pulse enable	Speed controller integrator disable	Motor selection	Motor selection B	Motor selection A	Drive parameter set selection C	Drive parameter set selection B	Drive parameter set selection A
DBB22								

Notes:

DB31 – DB61	Signals to axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB23								
DBB24	Master/ slave switch on		Setpoint switchover Accept drive control	Master/ slave Activate torque equalization controller	MCS coupling Activate collision protection	MCS coupling deactivate or do not permit	Control axis	Stepping motor Rotation monitoring
DBB25								Activate dynamic backlash compensation
DBB26 Grinding				Enable slave axis overlay	Compensatory controller on			
DBB27 Grinding	Stop				Resume			
	HIAxMove	Corr	DEPBCS	DEPMCS	HIAxMove	Corr	DEPBCS	DEPMCS
DBB28 Grinding Oscillation	PLC controls axis	Stop along braking ramp	Stop at next reversal point	Change reversal point	Set reversal point	Continue	Reset	Oscillation reversal from external
DBB29 Couplings			Disable automatic synchronization	Start gantry synchronization				
DBB30 Technology				Spindle start positioning	Select gear stage	Spindle start counter clockwise rotation	Spindle start clockwise rotation	Spindle stop

Notes:

DB31 – DB61	Signals to axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB31 Technology	Clear synchronism	Clear synchronism	Disable synchronization	Resynchronize				
DBB32								
DBB33								
DBB34					Braking along the NCK ramp when SS2E is selected		Setpoint limitation Bit 1	Setpoint limitation Bit 0
DBB35-DBB55								
DBB56 PLC → HMI						Spindle internal clamping	Spindle speed display	Separate feedrate drive coupled as C axis
DBB57								
DBB58	Reserved							
DBB59								

Notes:

DB31 – DB61	Signals from axis/spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB60 Axis and Spindle	Position reached with exact stop: fine	Position reached with exact stop: coarse	References synchronizes: 2	References synchronizes: 1	Encoder limit frequency exceeded: 1	Encoder limit frequency exceeded: 2	NCU_Link Axis active	Spindle /no axis
DBB61 Axis and Spindle	Current controller active	Speed controller active	Position controller active	Axis/spindle stationary ($n < n_{min}$)	Follow-up mode active	Axis ready	Axial alarm	Travel request
DBB62	Axis container rotation active	Force fixed stop limited	Fixed stop reached	Activate travel to fixed stop	Measurement active	Revolutional feedrate active	Handwheel overlay active	Software cams active
DBB63	Stop: HIAxMove active	Stop: Corr active	Stop: DEPBCS active	Stop: DEPMCS active	Axis/ spindle disable active	Axis stop active	PLC-controlled axis	AXRESET DONE
DBB64 Axis and Spindle	Traverse command: plus	Traverse command: minus	Travel request: plus	Travel request: minus		Handwheel active: 3	Handwheel active: 2	Handwheel active: 1
DBB65 Axis and Spindle	Active machine function	Continuous manual travel	Var. INC	10000 INC	1000 INC	100 INC	10 INC	1 INC
DBB66 Axis and Spindle								MCS coupling Collision protection active
DBB67								Handwheel direction of rotation inverted
DBB68	PLC axis/ spindle	Neutral axis/ spindle	Axis replacement possible	New type requested by PLC	NC axis/spindle channel assignment: D	NC axis/spindle channel assignment: C	NC axis/spindle channel assignment: B	NC axis/spindle channel assignment: A
DBB69	NCU number in NCU link network: E	NCU number in NCU link network: D	NCU number in NCU link network: C	NCU number in NCU link network: B	NCU number in NCU link network: A	Control parameter block: C	Control parameter block: B	Control parameter block: A
DBB70			DRV Safety Integrated with SIC/SCC active			REPOS delay acknowledgment	REPOS offset valid	REPOS offset
DBB71	PLC axis permanently assigned		Position restored: Encoder 2	Position restored: Encoder 1				Brake test active
DBB72 HMI → PLC								REPOS delay active
DBB73 HMI → PLC								

Notes:

DB31 – DB61	Signals from axis/spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB74 Axis				Modulo rotary axes: Traversing range limits active				
DBB75 Axis	JOG-position reached	Position JOG active	JOG fixed-point approach reached 2	1	0	JOG fixed-point approach active 2	1	0
DBB76 Axis	Rounding axis in position	Indexing axis in position	Positioning axis /Z2-P2/	Path axis				Scratch pulse
DBB77 Axis								Reduced-speed collision
DBD78 Axis	F function (REAL format) for positioning axis							
DBB82 Spindle					Gear change-over	Setpoint gear stage C	B	A
DBB83 Spindle	Actual rotat. direction CW	Speed monitoring	Spindle in setpoint range	Support area limits violated	Geometry monitoring	Set speed increased	limited	Speed limit exceeded
DBB84 Spindle	Active spindle operating mode: Control mode	Oscillation mode	Positioning mode	Synchronous mode	Tapping without compensating chuck	CLGON active	SUG active (grinding wheel surface speed)	Const. cutting speed active
DBB85 Spindle			Spindle in position					Tool with dynamic limitation
DBW86 Spindle	M function (binary) for spindle (M3, M4, M5, M19, M70 and/or via MD specified)							
DBD88 Spindle	S function (floating-point) for spindle							
DBB92 Drive	Enable drive operation		Motor holding brake opened	Independent drive motion active			Ramp-function generator disable active	
DBB93 Drive	Pulses enabled	n controller integrator inhibited	Drive ready	Active motor B	A	Active drive parameter set 0 … 7 C	B	A
DBB94 Drive	Variable signaling function	nist < nsoll	|nist| < nx	|nist| < nmin	Md < Mdx	Run-up complete	Temperature prewarning Heat sink	Motor
DBB95 Drive	Alarm of alarm class C is active				ESR: Generator operation minimum speed fallen below (p2161)	ESR: Response triggered or generator operation active (r0887.12)	ESR: DC link undervoltage (p1248)	UDC connection < Alarm threshold

Notes:

DB31 – DB61	Signals from axis/spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB96	Master/ slave active		Setpoint switchover drive control active	Compen-satory control-ler active	Master/ Slave Coarse speed difference	Fine speed difference	Axis control active /Z3-TE6/	Stepping motor Rota-tion monitor-ing error
DBB97					MCS coupling Offset change	Monitoring active	Coupling active	Following axis
DBB98 Synchronous spindle	ESR re-sponse initiat-ed	Accel. warn-ing threshold reached	Velocity warn-ing threshold reached	Superim-posed motion		Actual value coupling	Synchronous operation coarse	fine
DBB99 Synchronous spindle		Max. acce-leration reached	Max. velocity reached	Synchro-nization in progress	Axis acce-lerating	Synchronism correction implemented	Following spindle active	Leading spin-dle active
DBB100 Oscillation grinding	Oscillation active	Oscillation motion active	Sparking-out active	Fault during oscillation motion	Oscillation cannot be started	Oscillation reversal from external ac-tive		
DBB101 Gantry	Gantry axis	Gantry lead-ing axis	Gantry group-ing is syn-chronous	Gantry syn-chroni-zation run ready to start	Gantry warn-ing limit ex-ceeded	Gantry cut-off limit exceed-ed		
DBB102		Position measuring system 2 activated	Position measuring system 1 activated		Clamping tolerance exceeded			Dynam. Backlash compensa-tion active
DBB103			Synchronous operation 2 coarse	Synchronous operation 2 fine				Syn-chronism being consid-ered
DBB104 Grinding	Active infeed axis Axis 8	Axis 7	Axis 6	Axis 5	Axis 4	Axis 3	Axis 2	Axis 1
DBB105 Grinding	Active infeed axis Axis 16	Axis 15	Axis 14	Axis 13	Axis 12	Axis 11	Axis 10	Axis 9
DBB106 Grinding	Active infeed axis Axis 24	Axis 23	Axis 22	Axis 21	Axis 20	Axis 19	Axis 18	Axis 17
DBB107 Grinding	Active infeed axis	Axis 31	Axis 30	Axis 29	Axis 28	Axis 27	Axis 26	Axis 25
DBB108-112								

Notes:

DB31 – DB61	Signale von Achse/Spindel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB128 HMI → PLC							Program test: Activate	Program test: Suppress
DBB129								
DBB130	Motor/drive data set: Formatting is valid			Motor/drive data set Formatting: Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB131								
DBB132 Weiss spindle	Sensor configuration	Sensor S6 available	Sensor S5 available (angular position, motor shaft)	Sensor S4 available (piston end position)			Sensor S1 available (clamped state)	Sensors available
DBB133 Weiss spindle	Sensor configuration					Status value is generated, speed limitation p5043 is active		
DBW134 Weiss spindle	Clamping state (sensor S1) status value							
DBW136 Weiss spindle	Clamping state (sensor S1) analog value							
DBB138 Weiss spindle			Sensor S5 angular position, motor shaft	Sensor S4 piston end position				
DBB139 Weiss spindle	Status digital sensors							

Notes:

Tool management interface (DB71 – DB73)

Interface for loading/unloading magazine (DB71)

DB71	Interface for loading/unloading magazine (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (I) active I8	I7	I6	I5	I4	I3	I2	I1
DBB1	I16	I15	I14	I13	I12	I11	I10	I9
DBB2	Standard end acknowledgment I 8	I 7	I 6	I 5	I 4	I 3	I 2	I 1
DBB3	I 16	I 15	I 14	I 13	I 12	I 11	I 10	I 9
DBB n	Reserved	Reserved	Positioning Multitool	Order from NC-Program	Positioning Magazine	Relocate	Unload	Load
DBBn+1	Data in extended area (DB1071)	Reserved						Acknowledgment Status = 3
DBB n + 2	Assigned channel (8 bit INT)							
DBB n + 3	Tool management no. (8 bit INT)							
DBD n + 4	Unassigned parameter 1 (D word)							
DBD n + 8	Unassigned parameter 2 (D word)							
DBD n + 12	Unassigned parameter 3 (D word)							
DBW n + 16	Identification for loading/unloading station (Int), (fixed value 9999)							
DBW n + 18	No. of loading station (INT)							
DBW n + 20	Magazine no. (source) for unloading/reloading/positioning (INT)							
DBW n + 22	Location no. (source) for unloading/reloading/positioning (INT)							

Notes:

DB71	Interface for loading/unloading magazine (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW n + 24	Magazine no. (target) for loading/reloading/positioning (INT)							
DBW n + 26	Location no. (target) for loading/reloading/positioning (INT)							
DBB n + 28	Reserved							Loading/ unloading without mag- azine move- ment
DBB n + 29	Reserved							

Initial addresses of the loading/unloading stations:
Loading/unloading station 1: n= 4 Loading/unloading station 3: n= 64
Loading/unloading station 2: n= 34 Loading/unloading station 4: n= 94

Load interface 1 is responsible for spindle loading and reloading of tools, for relocating tools and for positioning at any location (e.g. buffer).

Notes:

Interface for spindle as change position (DB72)

DB72	Signals from spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (I) active I8	I7	I6	I5	I4	I3	I2	I1
DBB1	I16	I15	I14	I13	I12	I11	I10	I9
DBB2	Standard end acknowledgment I8	I7	I6	I5	I4	I3	I2	I2
DBB3	I16	I15	I14	I13	I12	I11	I10	I9
DBB n	Spindle tool remains in spindle	Replace manual tool	Insert manual tool	OldT in buffer no (n+42)	T0	Prepare change	Perform change (initiate: M06)	Obligatory change
DBB n + 1	Data in extended area (DB1072)	Reserved						Acknowledgment Status = 3
DBB n + 2	Assigned channel (8 bit INT)							
DBB n + 3	Tool management no. (8 bit INT)							
DBD n + 4	Unassigned parameter 1 (D word)							
DBD n + 8	Unassigned parameter 2 (D word)							
DBD n + 12	Unassigned parameter 3 (D word)							
DBW n + 16	Buffer identification (Int), (fixed value 9998) (corresponds to ”Target position for new tool”)							
DBW n + 18	Relative location (target) in the buffer (INT)							
DBW n + 20	Magazine no. (source) for new tool (INT)							
DBW n + 22	Location no. (source) for new tool (INT)							
DBW n + 24	Magazine no. (target) for old tool (INT)							
DBW n + 26	Location no. (target) for old tool (INT)							

Notes:

DB72	Signals from spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW n + 26	Location no. (target) for old tool (INT)							
DBW n + 28	Tool new: location type (INT)							
DBW n + 30	Tool new: size left (INT)							
DBW n + 32	Tool new: size right (INT)							
DBW n + 34	Tool new: size top (INT)							
DBW n + 36	Tool new: size bottom (INT)							
DBB n + 38	Tool status for tool new							
	Manual tool	1:1 Change		Master tool	to be loaded	to be unloaded	Disabled, but ignore	Tool in the buffer
DBB n + 39	Tool status for tool new							
	Tool was in use	Tool fixed location coded	Tool being changed	Prewarning limit reached	Tool meas-ured	Tool disabled	Tool enabled	Active tool
DBW n + 40	Tool new: T no. (INT)							
DBW n + 42	If DBX (n+0.4) = 1, then buffer location of old tool is entered here.							
DBW n + 44	Original magazine of new tool							
DBW n + 46	Original location of new tool							
Initial addresses of the buffers: Spindle 1:n= 4 Spindle 2:n = 52								

Notes:

Interface for turret (DB73)

DB73	Signals from turret (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (I) active I8	I7	I6	I5	I4	I3	I2	I1
DBB1	I16	I15	I14	I13	I12	I11	I10	I9
DBB2	Standard end acknowledgment I8	I7	I6	I5	I4	I3	I2	I1
DBB3	I16	I15	I14	I13	I12	I11	I10	I9
DBB n	Reserved	Unload manual tool	Reserved		T0	Reserved	Perform change (initiation: T no.)	Obligatory change
DBB n + 1	Data in extended area (DB1073)	Reserved						Acknowl-edgment Status = 3
DBB n + 2	Assigned channel (8 bit INT)							
DBB n + 3	Tool management no. (8 bit INT)							
DBD n + 4	Unassigned parameter 1 (D word)							
DBD n + 8	Unassigned parameter 2 (D word)							
DBD n + 12	Unassigned parameter 3 (D word)							
DBW n + 16	Reserved							
DBW n + 18	Reserved							
DBW n + 20	Magazine no. of the turret (INT)							
DBW n + 22	Location no. of new tool (INT)							
DBW n + 24	Magazine no. of the old tool							
DBW n + 26	Location no. for old tool (INT)							
DBW n + 28	Tool new: location type (INT)							
DBW n + 30	Tool new: size left (INT)							
DBW n + 32	Tool new: size right (INT)							
DBW n + 34	Tool new: size top (INT)							
DBW n + 36	Tool new: size bottom (INT)							

Notes:

DB73	Signals from turret (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB n + 38	Tool status for tool new: Manual tool	1:1 Change		Master tool	to be loaded	to be unloaded	Disabled, but ignore	Tool in the buffer
DBB n + 39	Werkzeugstatus für Werkzeug neu: Tool has been in use	Tool fixed location coded	Tool being changed	Prewarning limit reached	Tool meas-ured	Tool disabled	Tool enabled	Active tool
DBW n + 40	Tool new: T no. (INT)							
DBW n + 42	Original location of new tool in this turret magazine							
Initial addresses of the turrets: Turret 1: n = 4 Turret 2: n = 48								

Interface for loading / unloading: Multitool (DB1071)

DB1071	Loading / unloading points (NC→ PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW n+0	Distance code							
DBW n+2	Number of Multitool locations							
DBW n+4	Multitool distance							
DBW n+8	Multitool number							
DBW n+10	Multitool location number							
DBW n+12	Toolholder							
DBW n+14	Reserved							
DBW n+16	Reserved							
DBW n+18	Reserved							

Notes:

Interface for spindle: Multitool (DB1072)

DB1072	Spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW n + 0	Distance code							
DBW n + 2	Number of Multitool locations							
DBW n + 4	Multitool distance							
DBW n + 8	Multitool number (tool new)							
DBW n + 10	Multitool location number (tool new)							
DBW n + 12	Multitool number (tool old)							
DBW n + 14	Multitool location number (tool old)							
DBW n + 16	Tool new: Location type							
DBW n + 18	Tool new: Size left							
DBW n + 20	Tool new: Size right							
DBW n + 22	Tool new: Size top							
DBW n + 24	Tool new: Size bottom							
DBW n + 26	Tool status for new tool							
DBW n + 28	Tool new: Internal T number of the NC							
DBW n + 30	Tool holder							
DBW n + 32	Original magazine of the new tool							
DBW n + 34	Original location of new tool							
DBW n + 36	Reserved							

Notes:

Interface for turret: Multitool (DB1073)

DB1073	Turret (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW n+0	Distance code							
DBW n+2	Number of Multitool locations							
DBW n+4	Multitool distance							
DBW n+8	Multitool number (tool new)							
DBW n+10	Multitool location numberr (tool new)							
DBW n+12	Multitool number (tool old)							
DBW n+14	Multitool location numberr (tool old)							
DBW n+16	Tool new: Location type							
DBW n+18	Tool new: Size left							
DBW n+20	Tool new: Size right							
DBW n+22	Tool new: Size top							
DBW n+24	Tool new: Size bottom							
DBW n+26	Tool status for new tool							
DBW n+28	Tool new: Internal T number of the NC							
DBW n+30	Tool holder							
DBW n+32	Original magazine of the new tool							
DBW n+34	Original location of new tool							
DBW n+36	Reserved							

Notes:

M Decoding (DB75/76)

Decoding list for M Decoding (DB75)

DB75	Decoding list for M Decoding (NC → PLC)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW0	Extended M address group 1 (16 Bit INT)							
DBD2	First M address group 1 (DINT)							
DBD6	Last M address group 1 DINT)							
DBW10	Extended M address group 2 (16 Bit INT)							
DBD12	First M address group 2 (DINT)							
DBD16	Last M address group 2 (DINT)							
DBW20	Extended M address group 3 (16 Bit INT)							
DBD22	First M address group 3 (DINT)							
DBD26	Last M address group 3 (DINT)							
DBW30	Extended M address group 4 (16 Bit INT)							
DBD32	First M address group 4 (DINT)							
DBD36	Last M address group 4 (DINT)							
...	...							
DBW150	Extended M address group 16 (16 Bit INT)							
DBD152	First M address group 16 (DINT)							
DBD156	Last M address group 16 (DINT)							

Notes:

Signal list for M Decoding (DB76)

DB76	Signal list for M Decoding (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW0	M Signals group 1 (M Signal 0-7)							
	M Signals group 1 (M Signal 8-15)							
DBW2	M Signals group 2 (M Signal 0-7)							
	M Signals group 2 (M Signal 8-15)							
DBW4	M Signals group 3 (M Signal 0-7)							
	M Signals group 3 (M Signal 8-15)							
DBW6	M Signals group 4 (M Signal 0-7)							
	M Signals group 4 (M Signal 8-15)							
...	...							
	...							
DBW30	M Signals group 16 (M Signal 0-7)							
	M Signals group 16 (M Signal 8-15)							

Notes:

Powerride Signals (DB78)

Preconfigured behaviour for Powerride (DB78) - Signal overview

DB78	Feed configuration (Signals)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB0	Switch position at 100% override							
DBB1	Switch position at maximum override							
DBD2	Time delay at 100% before continuing override							
DBB6	Switch position 1 - Override factor							
DBB7	Switch position 1 - Colour of scale LEDs							
DBB8	Switch position 1 - Index of vibration pattern							
DBB10	Switch position 2 - Override factor							
DBB11	Switch position 2 - Colour of scale LEDs							
DBB12	Switch position 2 - Index of vibration pattern							
...	...							
DBB146	Switch position 36 - Override factor							
DBB147	Switch position 36 - Colour of scale LEDs							
DBB148	Switch position 36 - Index of vibration pattern							

Notes:

Preconfigured behaviour for Powerride (DB78) - Signal overview

DB78	Spindle configuration (Signals)							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB150	Switch position at 100% override							
DBB151	Switch position at maximum override							
DBD152	Time delay at 100% before continuing override							
DBB156	Switch position 1 - Override factor							
DBB157	Switch position 1 - Colour of scale LEDs							
DBB158	Switch position 1 - Index of vibration pattern							
DBB160	Switch position 2 - Override factor							
DBB161	Switch position 2 - Colour of scale LEDs							
DBB162	Switch position 2 - Index of vibration pattern							
...	...							
DBB296	Switch position 36 - Override factor							
DBB297	Switch position 36 - Colour of scale LEDs							
DBB298	Switch position 36 - Index of vibration pattern							

Notes:

Signals for Ctrl-Energy (DB1000 - 1001)

Energy saving profiles (DB1000)

DB1000	Energy saving profiles (HMI → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
n+0	Control signals						Set time to pre-warning limit	Immediately activate energy saving profile
n+1	Control signals							Immediately activate energy saving profile
n+2	Signals to check energy saving profile						PLC user signal	Master computer signal
n+3	Reserved							
n+4	Control signals						Activation time T1 has expired	energy saving profile active
n+5	Reserved							
n+6	Actual value: Actual value T1							
n+8	Actual value: Actual value T2							
n+10	Effectiveness profile						Disable energy saving profile	Energy saving profile configured
	State conditions							
n+11						Screen change active	Data transfer active	Operator panel
n+12								MCP operated
n+13	NC channel in reset							
	8	7	6	5	4	3	2	1
n+14	NC channel in reset						10	9
n+15							PLC user signal	Master computer signal
n+16	Activation time T1							
n+18	Prewarning time T2							

Profile instances:
Energy profile 1 (n=0): DB1000.DBB0 … DBB19
Energy profile 2 (n=20): DB1000.DBB20 … DBB39
etc
Energy profile 8 (n=140): DB1000.DBB140 … DBB159

Notes:

SENTRON PAC (DB1001)

DB1001	SENTRON PAC							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0000 PLC ⇨HMI			Regenerative energy: Integration using FW	Infeed energy: Integration using FW	Regenerative energy is read by SENTRON PAC	Infeed energy is read by SENTRON PAC	SENTRON PAC represents the machine	Display manual value
0001								PLC FW should perform measurement
0002 HMI ⇨PLC								Measurement in progress
0003								Power dis-play on
0004	Manual value (REAL) to HMI							
0008	Total active power (REAL) to HMI							
0012	Measured drawn active energy in kWh (REAL) to HMI							
0016	Measured supplied active energy in kWh (REAL) to HMI							
0020	Total active power in watts (REAL) from SENTRON							
0024	Drawn active energy at taritf1 (F) in Wh (REAL) from SENTRON							
0028	Supplied active energy at tariff1 (F) in Wh (REAL) from SENTRON							
0032	Drawn active energy in kWh per day (REAL) to HMI							
0036	Supplied active energy in kWh per day (REAL) to HMI							
0040	Drawn active energy in kWh per previous day (REAL) to HMI							
0044	Supplied active energy in kWh per previous day (REAL) to HMI							
0048	Drawn active energy in kWh per month (REAL) to HMI							
0052	Supplied active energy in kWh per month (REAL) to HMI							
0056	Drawn active energy in kWh per previous month (REAL) to HMI							
0060	Supplied active energy in kWh per previous month (REAL) to HMI							
0064	Drawn active energy in kWh per annum (REAL) to HMI							
0068	Supplied active energy in kWh per annum (REAL) to HMI							
0072	Drawn active energy in kWh per previous year (REAL) to HMI							
0076	Supplied active energy in kWh per previous year (REAL) to HMI							
0096	Production Act (PLC ⇨HMI)							
0097						Value invalid DBD28	Value invalid DBD24	Value invalid DBD20
0098	Value invalid DBD384	Value invalid DBD344	Value invalid DBD304	Value invalid DBD264	Value invalid DBD224	Value invalid DBD184	Value invalid DBD144	Value invalid DBD104
0099							Value invalid DBD464	Value invalid DBD424

Notes:

SENTRON PAC, Auxiliary devices (DB1001)

DB1001	SENTRON PAC							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
n+100	Control/Command bits							
	Reset data structure			Input-Mode: (energy or power)	Different values when measuring energy	Energy measurement running	Read energy actual value update	Process device
n+101	Reserved							
n+102	Reserved							
n+104	Active power or active energy of the auxiliary unit [kW] or [kWh]							
n+108	Drawn active energy of the auxiliary unit [kWh]							
n+112	Active energy supplied by the auxiliary unit [kWh]							
n+116	Active energy drawn by the auxiliary unit at the measurement start [kWh]							
n+120	Active energy supplied by the auxiliary unit at the measurement start [kWh]							
n+124	Active energy drawn by the auxiliary unit at the measurement end [kWh]							
n+128	Active energy supplied by the auxiliary unit at the measurement start end [kWh]							
n+132	Reserved							
n+136	Reserved							

Auxiliary units instances:

Auxiliary units 1 (n=0): DB1001.DBB100 … DBB139
Auxiliary units 2 (n=40): DB1001.DBB140 … DBB179
etc
Auxiliary units 10 (n=360): DB1001.DBB460 … DBB499

Notes:

Spindle temperature sensor (DB1002)

DB1002	ISM_TS							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n+0	Sensor installation location							
n+2	Reserved							
n+4	Temperature sensor actual value [C]							
n+8	Temperature sensor warning threshold value [C]							
n+12	Number of alarm limit value violations							
n+14	Last alarm limit value violation: Year							
n+15	Last alarm limit value violation: Month							
n+16	Last alarm limit value violation: Day							
n+17	Last alarm limit value violation: Hour							
n+18	Last alarm limit value violation: Minute							
n+19	Last alarm limit value violation: Seconds							
n+20	Duration of the warning limit value violations							
n+24	Temperature sensor fault threshold value [C]							
n+28	Number of fault limit value violations							
n+30	Last fault limit value violation: Year							
n+31	Last fault limit value violation: Month							
n+32	Last fault limit value violation: Tag							
n+33	Last fault limit value violation: Day							
n+34	Last fault limit value violation: Minute							
n+35	Last fault limit value violation: Seconds							
n+36	Duration of the fault limit value violation							

Spindle/temperature sensor instances:

Spindle 1, temperature sensor 1 (n=0): DB1002.DBB0000 … DBB0039
Spindle 1, temperature sensor 2 (n=40): DB1002.DBB0040 … DBB0079
Spindle 1, temperature sensor 3 (n=80): DB1002.DBB0080 … DBB0119
Spindle 1, temperature sensor 4 (n=120): DB1002.DBB0120 … DBB0159
Spindle 1, temperature sensor 5 (n=160): DB1002.DBB0160 … DBB0199
Spindle 1, temperature sensor 6 (n=200): DB1002.DBB0200 … DBB0239

Spindle 2, temperature sensor 1 (n=240): DB1002.DBB0240 … DBB0279
Spindle 2, temperature sensor 2 (n=280): DB1002.DBB0280 … DBB0319
Spindle 2, temperature sensor 3 (n=320): DB1002.DBB0320 … DBB0359
Spindle 2, temperature sensor 4 (n=360): DB1002.DBB0360 … DBB0399
Spindle 2, temperature sensor 5 (n=400): DB1002.DBB0400 … DBB0439
Spindle 2, temperature sensor 6 (n=440): DB1002.DBB0440 … DBB0479

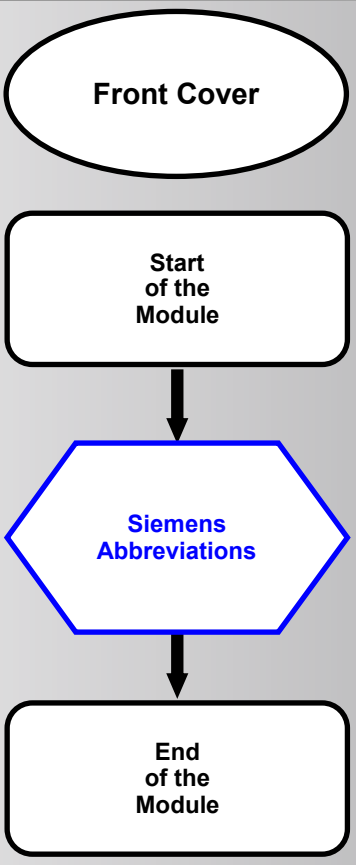
Appendix C: END

Notes:

Appendix D

SINUMERIK ONE

Siemens Abbreviations



Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

Siemens Abbreviations:

A	
A	Ausgang (output)
ADI4	(Analog drive interface for 4 axes)
AC	Adaptive Control
ALM	Active Line Module
ARM	Rotating induction motor
AS	Automation system
ASCII	American Standard Code for Information Interchange: American coding standard for the exchange of information
ASIC	Application-Specific Integrated Circuit: User switching circuit
ASUB	Asynchronous subprogram
AUXFU	Auxiliary function: Auxiliary function
AWL	Statement List PLC language

B	
BAG	Mode group
BCD	Binary Coded Decimals: Decimal numbers encoded in binary code
BERO	Contact-less proximity switch
BI	Binector Input
BICO	Binector Connector
BIN	BINary files: Binary files
BIOS	Basic Input Output System
BCS	Basic Coordinate System
BO	Binector Output

C	
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CC	Compile Cycle: Compile cycles
CEC	Cross Error Compensation
CI	Connector Input
CNC	Computerised Numerical Control: Computer-Supported Numerical Control
CO	Connector Output (Sinamics)
CoL	Certificate of License
COM	Communication
CPA	Compiler Projecting Data: Configuring data of the compiler
CRT	Cathode Ray Tube: picture tube
CSB	Central Service Board: PLC module
CU	Control Unit
CP	Communication Processor
CPU	Central Processing Unit: Central processing unit
CR	Carriage Return
CST	Configured Stop: Configured stop
CTS	Clear To Send: Ready to send signal for serial data interfaces
CUTCOM	Cutter radius Compensation: Tool radius compensation

Notes:

D	
DAC	Digital-to-Analog Converter
DB	Data Block (PLC)
DBB	Data Block Byte (PLC)
DBD	Data Block Double word (PLC)
DBW	Data Block Word (PLC)
DBX	Data block bit (PLC)
DDE	Dynamic Data Exchange
DDS	Drive Data Set: Drive data set
DIN	Deutsche Industrie Norm
DIO	Data Input/Output: Data transfer display
DIR	Directory: Directory
DLL	Dynamic Link Library
DO	Drive Object
DP	Display Port
DPM	Dual Port Memory
DPR	Dual Port RAM
DRAM	Dynamic memory (non-buffered)
DRF	Differential Resolver Function: Differential revolver function (handwheel)
DRIVE-CLiQ	Drive Component Link with IQ
DRY	Dry Run: Dry run feedrate
DSB	Decoding Single Block: Decoding single block
DSC	Dynamic Servo Control / Dynamic Stiffness Control
DW	Data Word
DWORD	Double Word (currently 32 bits)

E	
I	Input
EES	Execution from External Storage
I/O	Input/Output
ENC	Encoder: Actual value encoder
EFP	Compact I/O module (PLC I/O module)
ESD	Electrostatic Sensitive Devices
EMC	ElectroMagnetic Compatibility
EN	European standard
ENC	Encoder: Actual value encoder
EnDat	Encoder interface
EPROM	Erasable Programmable Read Only Memory: Erasable, electrically programmable read-only memory
ePS Network Services	Services for Internet-based remote machine maintenance
EQN	Designation for an absolute encoder with 2048 sine signals per revolution
ES	Engineering System
ESR	Extended Stop and Retract
ETC	ETC key ">"; softkey bar extension in the same menu

F	
FB	Function Block (PLC)
FC	Function Call: Function Block (PLC)
FEPROM	Flash EPROM: Read and write memory
FIFO	First In First Out: Memory that works without address specification and whose data is read in the same order in which they was stored
FIPO	Fine interpolator
FPU	Floating Point Unit: Floating Point Unit
CRC	Cutter Radius Compensation
FST	Feed Stop: Feedrate stop
FBD	Function Block Diagram (PLC programming method)
FW	Firmware

Notes:

G	
GC	Global Control (PROFIBUS: Broadcast telegram)
GDIR	Global part program memory
GEO	Geometry, e.g. geometry axis
GIA	Gear Interpolation dAta: Gear interpolation data
GND	Signal Ground
GP	Basic program (PLC)
GS	Gear Stage
GSD	Device master file for describing a PROFIBUS slave
GSDML	Generic Station Description Markup Language: XML-based description language for creating a GSD file
GUD	Global User Data: Global user data

H	
HDMI	High-Definition Multimedia Interface
HEX	Abbreviation for hexadecimal number
HLA	Hydraulic linear drive
HMI	Human Machine Interface: SINUMERIK user interface
MSD	Main Spindle Drive
HW	Hardware

I	
IBN	Commissioning
ICA	Interpolatory compensation
IM	Interface Module: Interconnection module
IMR	Interface Module Receive: Interface module for receiving data
IMS	Interface Module Send: Interface module for sending data
INC	Increment: Increment
INI	Initializing Data: Initializing data
IPO	Interpolator
ISA	Industry Standard Architecture
ISO	International Standardization Organization

J	
JOG	Jogging: Setup mode

K	
KV	Gain factor of control loop
Kp	Proportional gain
KÜ	Transformation ratio
LAD	Ladder Diagram (PLC programming method)

L	
LAI	Logic Machine Axis Image: Logical machine axes image
LAN	Local Area Network
LCD	Liquid Crystal Display: Liquid crystal display
LED	Light Emitting Diode: Light-emitting diode
LF	Line Feed
LR	Position controller
LSB	Least Significant Bit: Least significant bit
LUD	Local User Data: User data (local)

Notes:

M	
MAC	Media Access Control
MAIN	Main program: Main program (OB1, PLC)
MB	Megabyte
MCI	Motion Control Interface
MCIS	Motion Control Information System
MCP	Machine Control Panel: Machine control panel
MD	Machine Data
MDA	Manual Data Automatic: Manual input
MDS	Motor Data Set: Motor data set
MSGW	Message Word
MCS	Machine Coordinate System
MM	Motor Module
MPF	Main Program File: Main program (NC)
MCP	Machine control panel

N	
NC	Numerical Control: Numerical control with block preparation, traversing range, etc.
NCU	Numerical Control Unit: NC hardware unit
NRK	Name for the operating system of the NC
IS	Interface Signal
NURBS	Non-Uniform Rational B-Spline
WO	Work Offset
NX	Numerical Extension: Axis expansion board

O	
OB	Organization block in the PLC
OEM	Original Equipment Manufacturer
OP	Operator Panel: Operating equipment
OPI	Operator Panel Interface: Interface for connection to the operator panel
OPT	Options: Options
OLP	Optical Link Plug: Fiber optic bus connector
OSI	Open Systems Interconnection: Standard for computer communications

Notes:

P	
PIQ	Process Image Output
PII	Process Image Input
PC	Personal Computer
PCIN	Name of the SW for data exchange with the control
PCMCIA	Personal Computer Memory Card International Association: Plug-in memory card standardization
PCU	PC Unit: PC box (computer unit)
PG	Programming device
PKE	Parameter identification: Part of a PIV
PIV	Parameter identification: Value (parameterizing part of a PPO)
PLC	Programmable Logic Control: Adaptation control
PMS	Position Measuring System
PN	PROFINET
PNO	PROFIBUS user organization
PO	POWER ON
POU	Program Organization Unit
POS	Position/positioning
POSMO A	Positioning Motor Actuator: Positioning motor
POSMO CA	Positioning Motor Compact AC: Complete drive unit with integrated power and control module as well as positioning unit and program memory; AC infeed
POSMO CD	Positioning Motor Compact DC: Like CA but with DC infeed
POSMO SI	Positioning Motor Servo Integrated: Positioning motor, DC infeed
PPO	Parameter Process data Object: Cyclic data telegram for PROFIBUS DP transmission and "Variable speed drives" profile
PPU	Panel Processing Unit (central hardware for a panel-based CNC, e.g SINUMERIK 828D)
PROFIBUS	Process Field Bus: Serial data bus
PRT	Program Test
PSW	Program control word
PTP	Point-To-Point: Point-To-Point
PU	Processing Unit
PUD	Program global User Data: Program-global user variables
PZD	Process data: Process data part of a PPO

Q	
QEC	Quadrant Error Compensation

R	
RAM	Random Access Memory: Read/write memory
REF	REFerence point approach function
REPOS	REPOSition function
RISC	Reduced Instruction Set Computer: Type of processor with small instruction set and ability to process instructions at high speed
ROV	Rapid Override: Input correction
RP	R Parameter, arithmetic parameter, predefined user variable
RPA	R Parameter Active: Memory area in the NC for R parameter numbers
RPY	Roll Pitch Yaw: Rotation type of a coordinate system
RTLI	Rapid Traverse Linear Interpolation: Linear interpolation during rapid traverse motion
RTS	Request To Send: Control signal of serial data interfaces
RTCP	Real Time Control Protocol

Notes:

S	
SA	Synchronized Action
SAM	Safe Acceleration Monitoring (Safety Integrated)
SBC	Safe Brake Control (Safety Integrated)
SBT	Safe Brake Test (Safety Integrated)
SBL	SingleBlock: Single block
SBR	Subroutine: Subprogram (PLC)
SCA	Safe Software Cams (Safety Integrated)
SCC	Safety Command Channel (Safety Integrated)
SD	Setting Data
SDB	System Data Block
SDI	Safe Direction (Safety Integrated)
SEA	Setting Data Active: Identifier (file type) for setting data
SERUPRO	SEarch RUn by PROgram test: Block search, program test
SFB	System Function Block
SFC	System Function Call
SG	Safely Limited Speed (Safety Integrated)
SGA	Safety Related Output (Safety Integrated)
SGE	Safety Related Input (Safety Integrated)
SH	Safe standstill (Safety Integrated)
SIC	Safety Information Channel (Safety Integrated)
SIL	Safety Integrity Level
SIM	Single in Line Module
SK	Softkey
SKP	Skip: Function for skipping a part program block
SLM	Synchronous Linear Motor
SLS	Safely Limited Speed (Safety Integrated)
SLP	Safely Limited Positions (Safety Integrated)
SM	Stepper Motor
SMC	Sensor Module Cabinet Mounted
SME	Sensor Module Externally Mounted
SMI	Sensor Module Integrated
SOS	Safe Operating Stop (Safety Integrated)
SPF	Sub Routine File: Subprogram (NC)
SPL	Safe Programmable Logic Safety Integrated
SS1	Safe Stop 1 (Safety Integrated)
SS2	Safe Stop 2 (Safety Integrated)
SSM	Safe Speed Monitoring (Safety Integrated)
STO	Safe Torque Off (Safety Integrated)
PLC	Programmable Logic Controller
SRAM	Static RAM (non-volatile)
TNRC	Tool Nose Radius Compensation
SRM	Synchronous Rotary Motor
LEC	Leadscrew Error Compensation
SSI	Serial Synchronous Interface: Synchronous serial interface
SSL	Block search
STW	Control word
GWPS	Grinding Wheel Peripheral Speed
SW	Software
SYF	System Files: System files
SYNACT	SYNchronous ACTion: Synchronised Action

Notes:

T	
TB	Terminal Board (SINAMICS)
TCP	Tool Center Point: Tool tip
TCP/IP	Transport Control Protocol / Internet Protocol
TCU	Thin Client Unit
TEA	Testing Data Active: Identifier for machine data
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset: Tool offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TRANSMIT	Transform Milling Into Turning: Coordination transformation for milling operations on a lathe
TTL	Transistor-Transistor Logic (interface type)
TZ	Technology cycle

U	
UFR	User Frame: Work offset
SR	Subprogram
USB	Universal Serial Bus
UPS	Uninterruptible Power Supply

V	
VDI	Internal communication interface between NC and PLC
VDI	Verein Deutscher Ingenieure [Association of German Engineers]
VDE	Verband Deutscher Elektrotechniker [Association of German Electrical Engineers]
VI	Voltage Input
VO	Voltage Output
FDD	Feed Drive

W	
SAR	Smooth Approach and Retraction
WCS	Workpiece Coordinate System
T	Tool
TLC	Tool Length Compensation
WOP	Workshop-Oriented Programming
WPD	Workpiece Directory: Workpiece directory
TRC	Tool Radius Compensation
T	Tool
TO	Tool Offset
TM	Tool Management
TC	Tool change

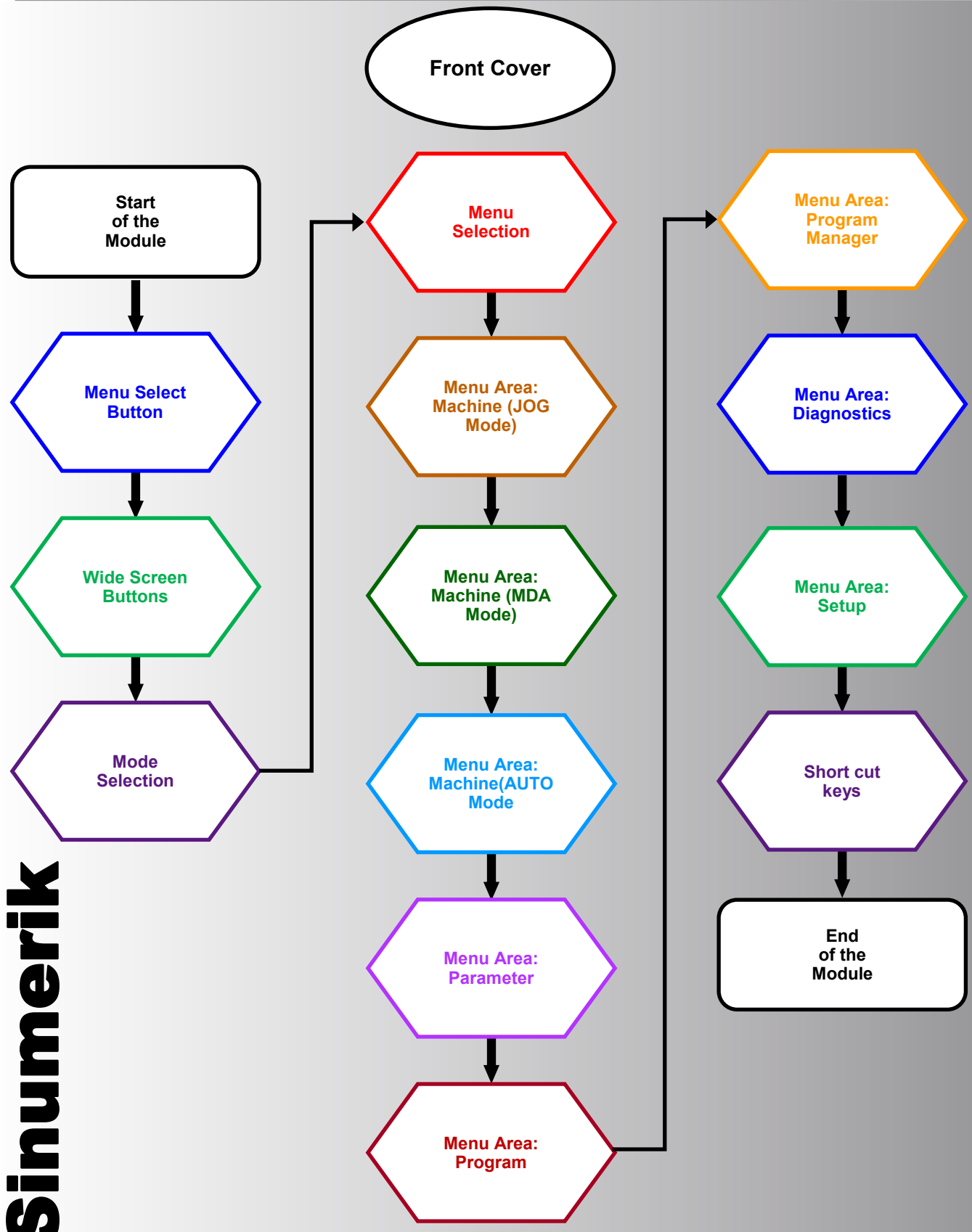
X	
XML	Extensible Markup Language

Z	
WOA	Work Offset Active: Identifier for work offsets
ZSW	Status word (of drive)

A060: END

Notes:

Menu Structure

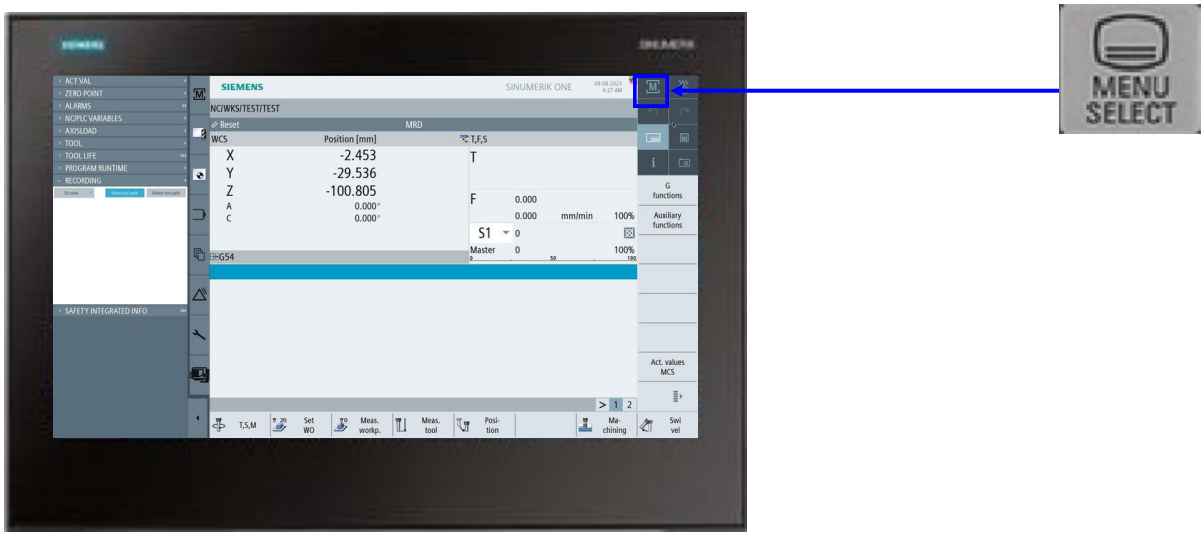


This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

Menu Select Button

The menu system of the SINUMERIK is accessed via the “Menu Select” button.

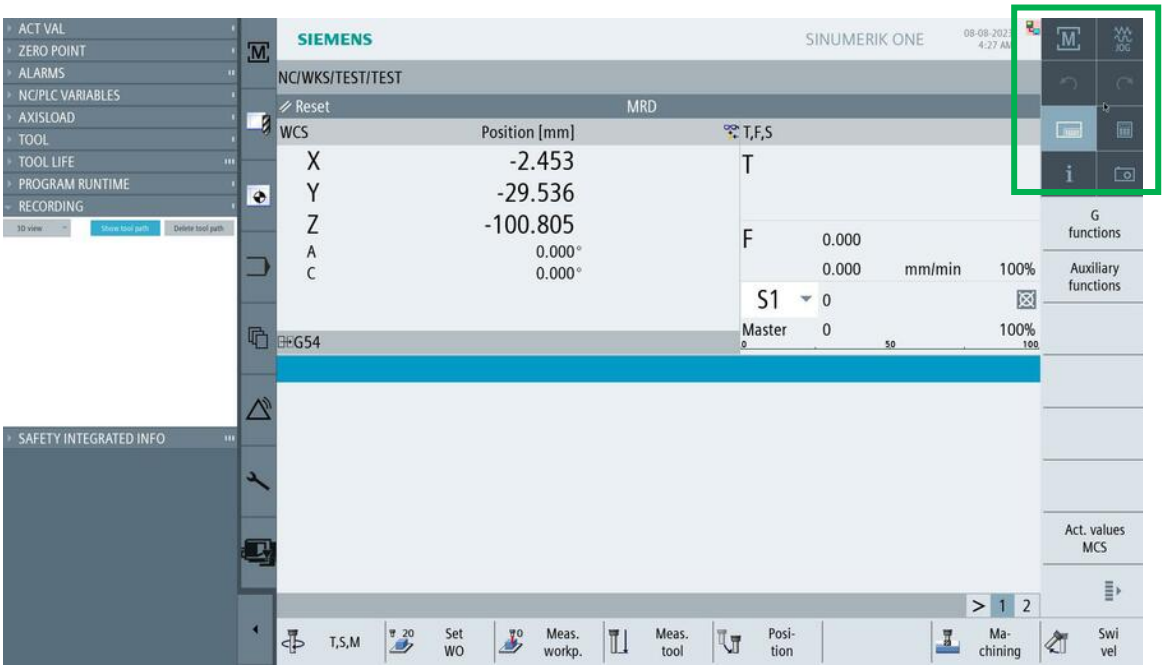
The location of the “Menu Select” button on the Touch Screen is as shown:



Notes:

Wide Screen Buttons

Shortcut softkeys



The 6 vertical soft-keys in the top right hand corner can be used as short cut softkeys to access different functionality of the control.

Shortcut Soft-key Functionality:



Undo if selected.



Redo if selected.



Virtual keyboard displayed if selected.



Virtual calculator displayed if selected.

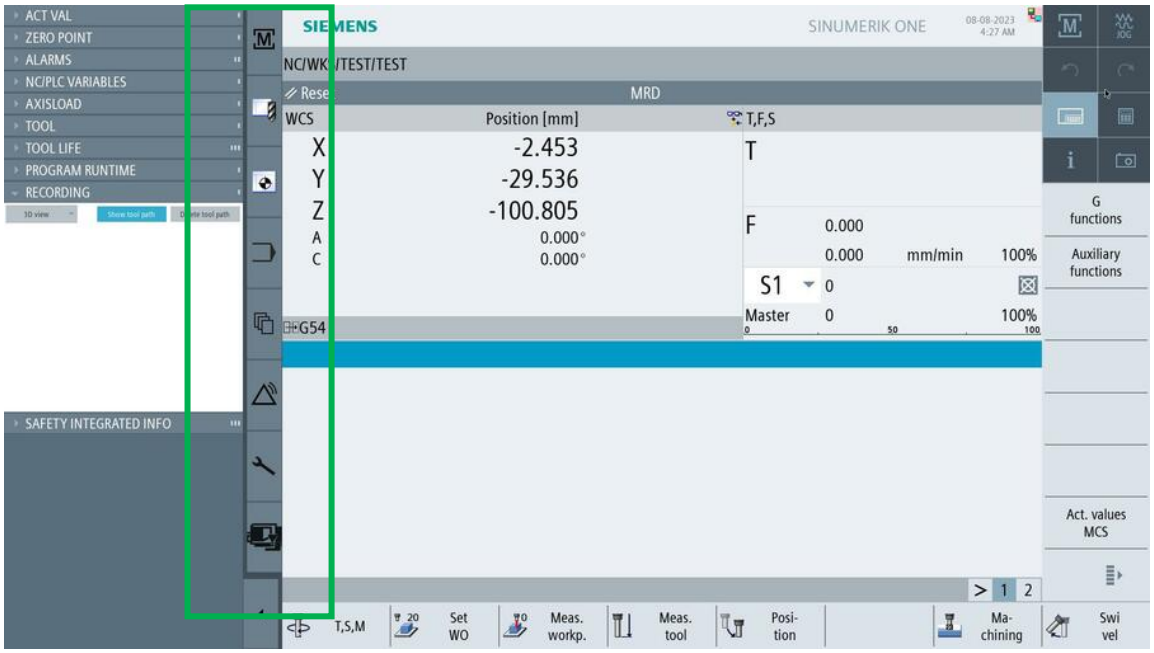


Help screen displayed if selected.



A screen shot is taken if selected.

Notes:



The 7 vertical soft-keys in the left hand of the main window can be used as short cut softkeys to access the areas of the control.

Shortcut Soft-key Functionality:


Displays the **Machine** area menu


Displays the **Tool List** area menu


Displays the **Workpiece offset** area menu


Displays the **Program** area menu


Displays the **Program Manager** area menu

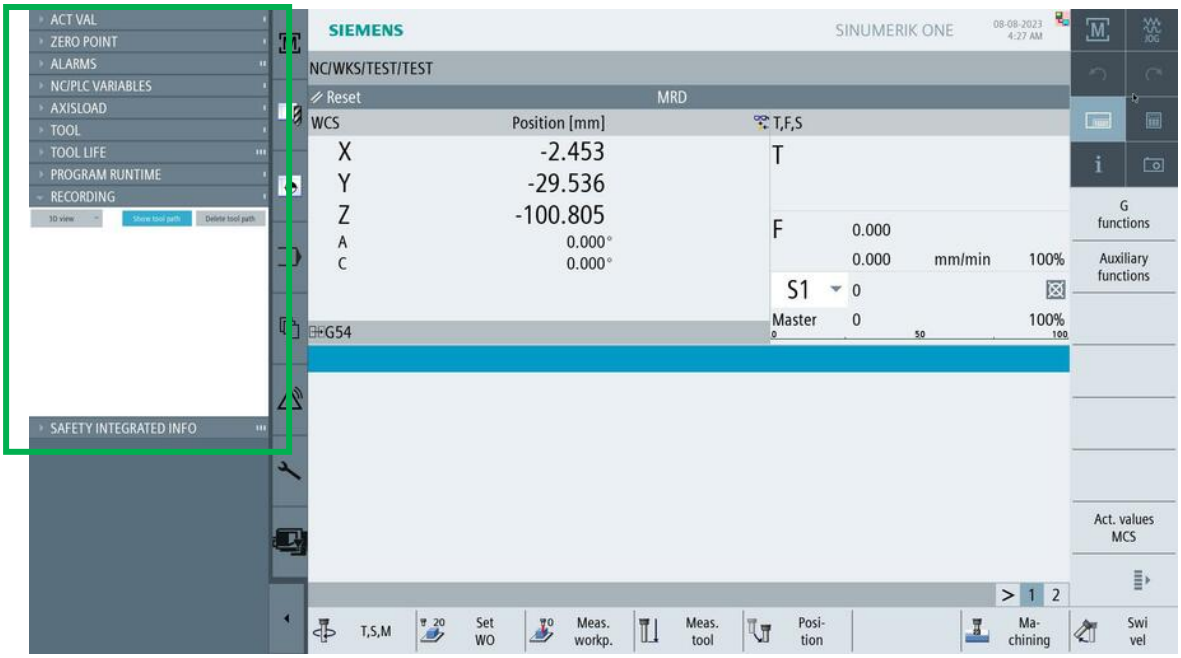

Displays the **alarm List** area menu


Displays the **Setup** area menu


Displays the **Adaptive control and monitoring suite** area menu - (*)

Notes:

(*) ONLY SINUMERIK ONE



The side screen displays 6 areas, workpiece offsets, active alarms, axis loads, active tool, tool management of the active too, and program runtime.

Shortcut Soft-key Functionality:

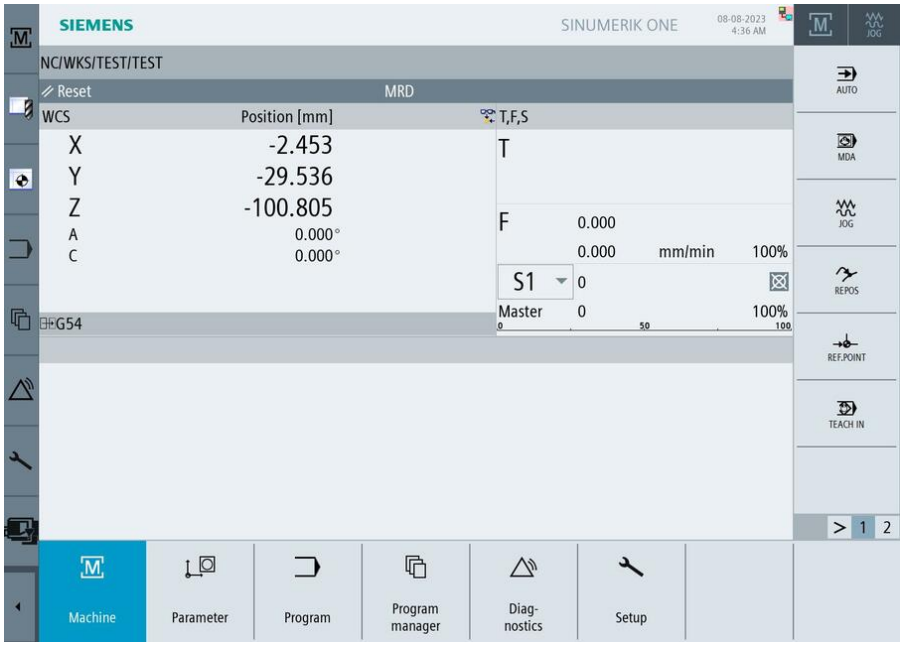
ACT VAL	Displays any active “ACTUAL VALUE” for the displayed co-ordinate system
ZERO POINT	Displays any active “ZERO_POINT” including coarse, fine, rotations etc etc.
ALARMS	Displays any active “ALARMS” only, NO “messages” will be shown.
NC/PLC VARIABLES	Displays any active “NC/PLC ARIABLES”.
AXISLOAD	Displays the active “AXISLOAD” for all axis in the system.
TOOL	Displays the active loaded “TOOL” including “Geometry and Wear”.
TOOL LIFE	Displays the active loaded “TOOL LIFE” along with “Tool management Data”.
PROGRAM RUNTIME	Displays the “PROGRAM RUN TIME” of the active program.
RECORDING	Displays the “RECORDING” of the active program. - (*)
SAFETY INTEGRATED INFO	Displays the “SAFETY INTERGRATED INFO” of the active program. - (*)

Notes:

(*) ONLY SINUMERIK ONE

Mode Selection

Pressing the Menu Select button reveals 6 horizontal menu soft-keys and 6 vertical soft-keys.



The 6 vertical soft-keys can be used to change between the machine modes: This can also be achieved via the Machine Control Panel.

Vertical Soft-key Functionality:

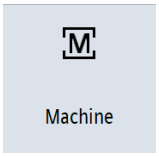
AUTO	Automatic mode (AUTO) is selected.
MDA	Manual Data Automatic mode (MDA) is selected.
JOG	Jog mode (JOG) is selected.
REPOS	Reposition mode (REPOS) is selected.
REF.POINT	Reference Point (REF POINT) mode is selected.
TEACH IN	Teach In (TEACH IN) mode is selected.

Notes:

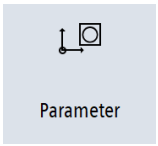
Menu Selection

The 6 horizontal soft-keys are used to select the different menus.
It is also possible to select the menus via the integrated NC keyboard of certain types of OP or the remote NC keyboard versions.

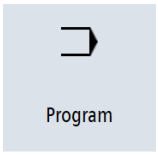
Menu Soft-keys: Brief Description



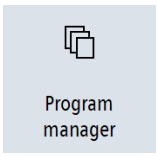
The **Machine** menu displays information such as the axis positions, distances to go, feedrates, spindle speeds, active tool etc
Functionality such as manual tool selection, tool/work-piece measurement, hand-wheel selection can be performed by selecting the relevant soft-key.
The functionality depends upon the selected operating mode.



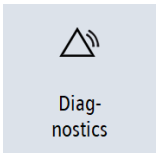
The **Parameter** menu displays information such tool/magazine data, work offsets, user variables etc.
It is possible to carry out functions such as creating tools, editing existing tool/ magazine data, editing work offsets/variables/setting data.



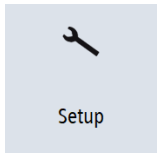
The **Program** menu is used to display a part program, only if a program is already open will its contents be displayed. Editing of the part program is possible here. If a part program is not already open, the Program Manager will be open.



The **Program Manager** menu displays the “Part program/subprogram/workpiece” directories. Programs/subprograms/workpieces can be created or opened for editing. It is also possible to save to/from controller using memory devices (CF/USB) or RS232 interface.



The **Diagnostics** menu is used for diagnosing machine problems, information about current/past alarms can be viewed. The status of drives, axes, networks, NC/PLC variables, utilization etc can be viewed. It is also possible to check the software/hardware versions of the system.
Machine information and service reports are also entered via the diagnostics menu.



The **Setup** menu is used for a wide variety of functions, these Include password entry, machine data modifications, creating archives, creating alarm files, time/date settings, network set-up, licensing/option selection, system data management, servo optimisation etc.

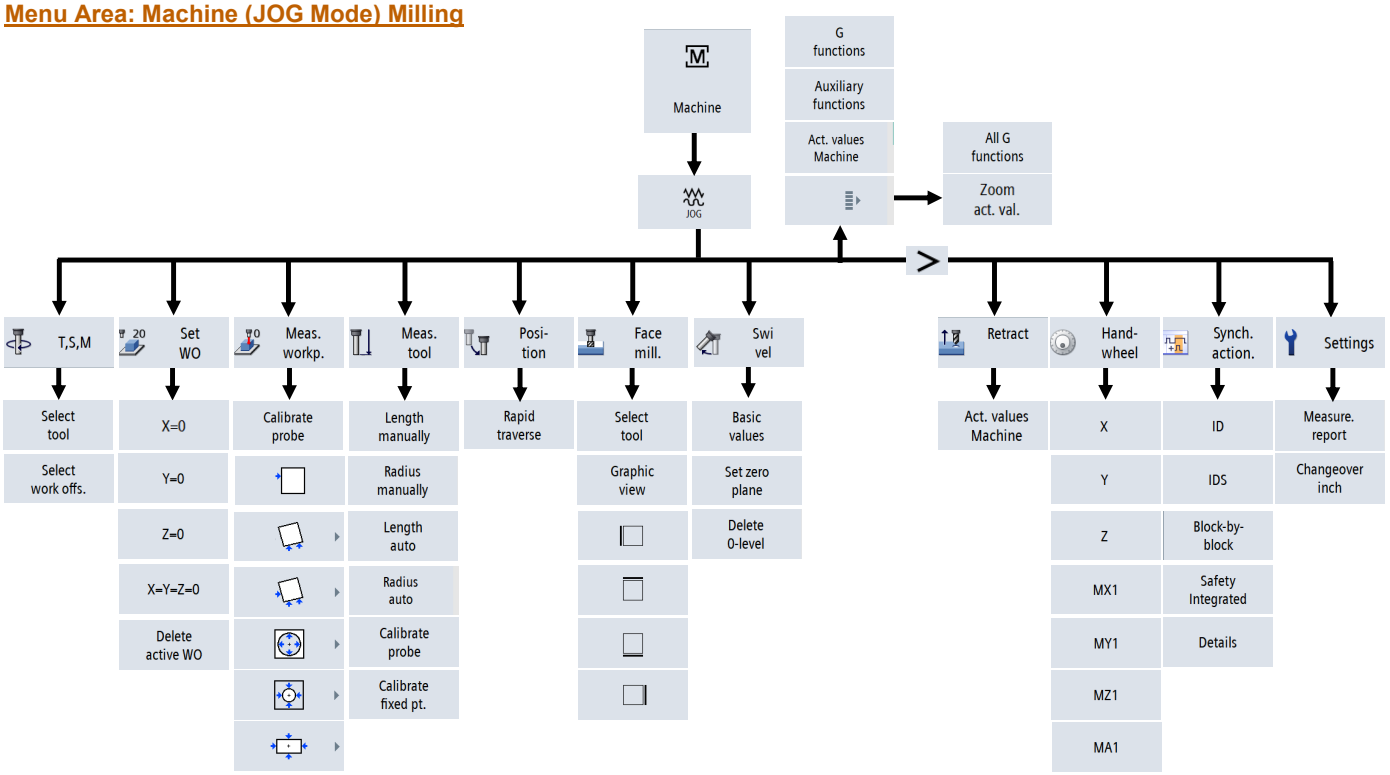




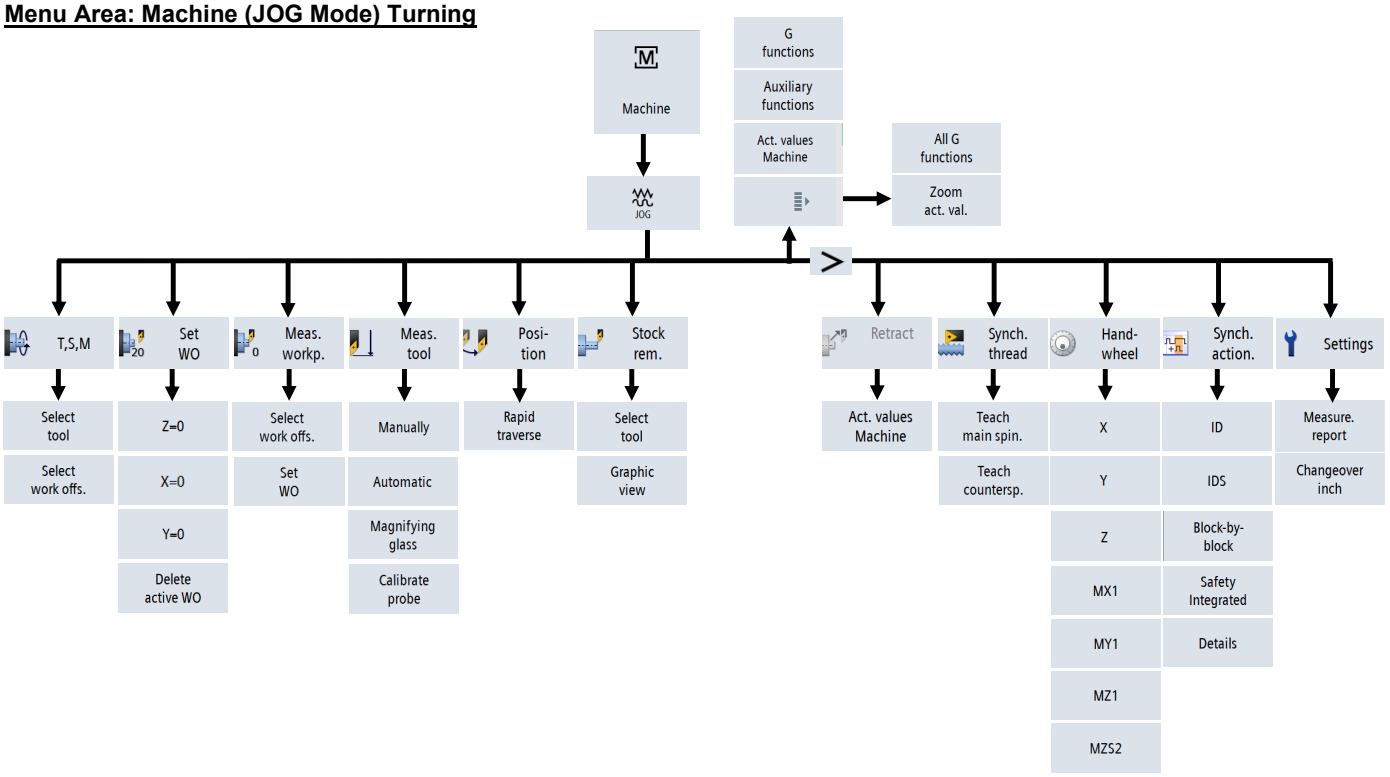
Reboot HMI
A further soft-key is available which is used for rebooting the HMI only. This is found via the Menu extension key.

Notes:

Menu Area: Machine (JOG Mode) Milling

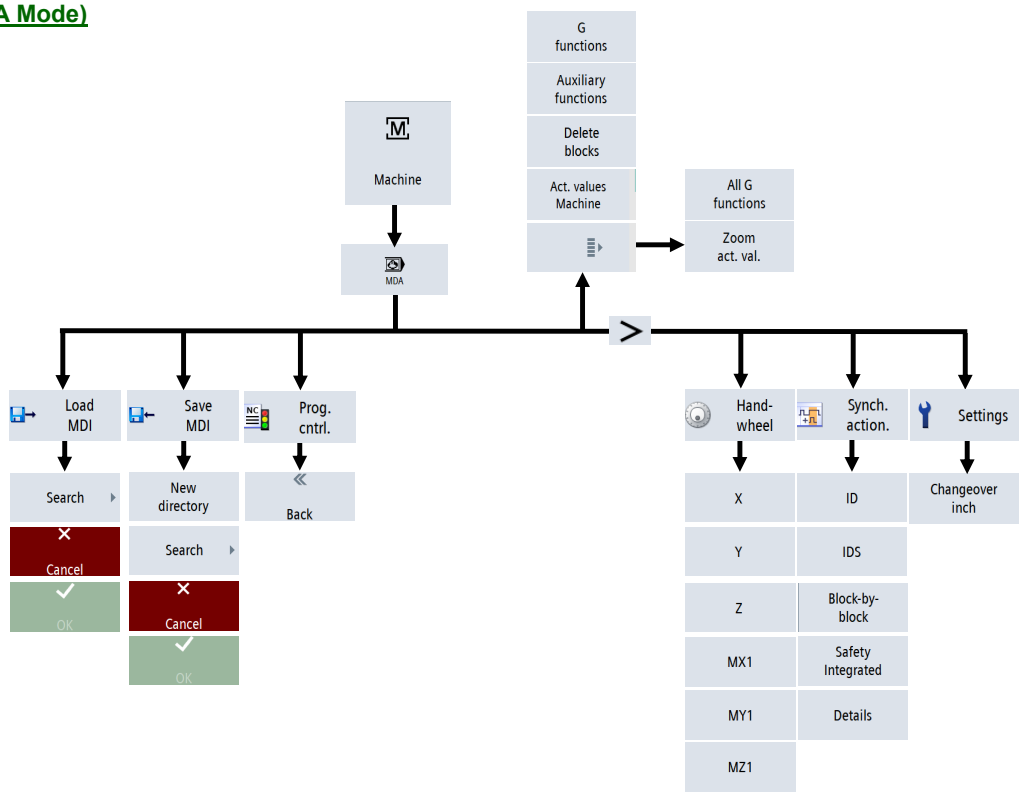


Menu Area: Machine (JOG Mode) Turning

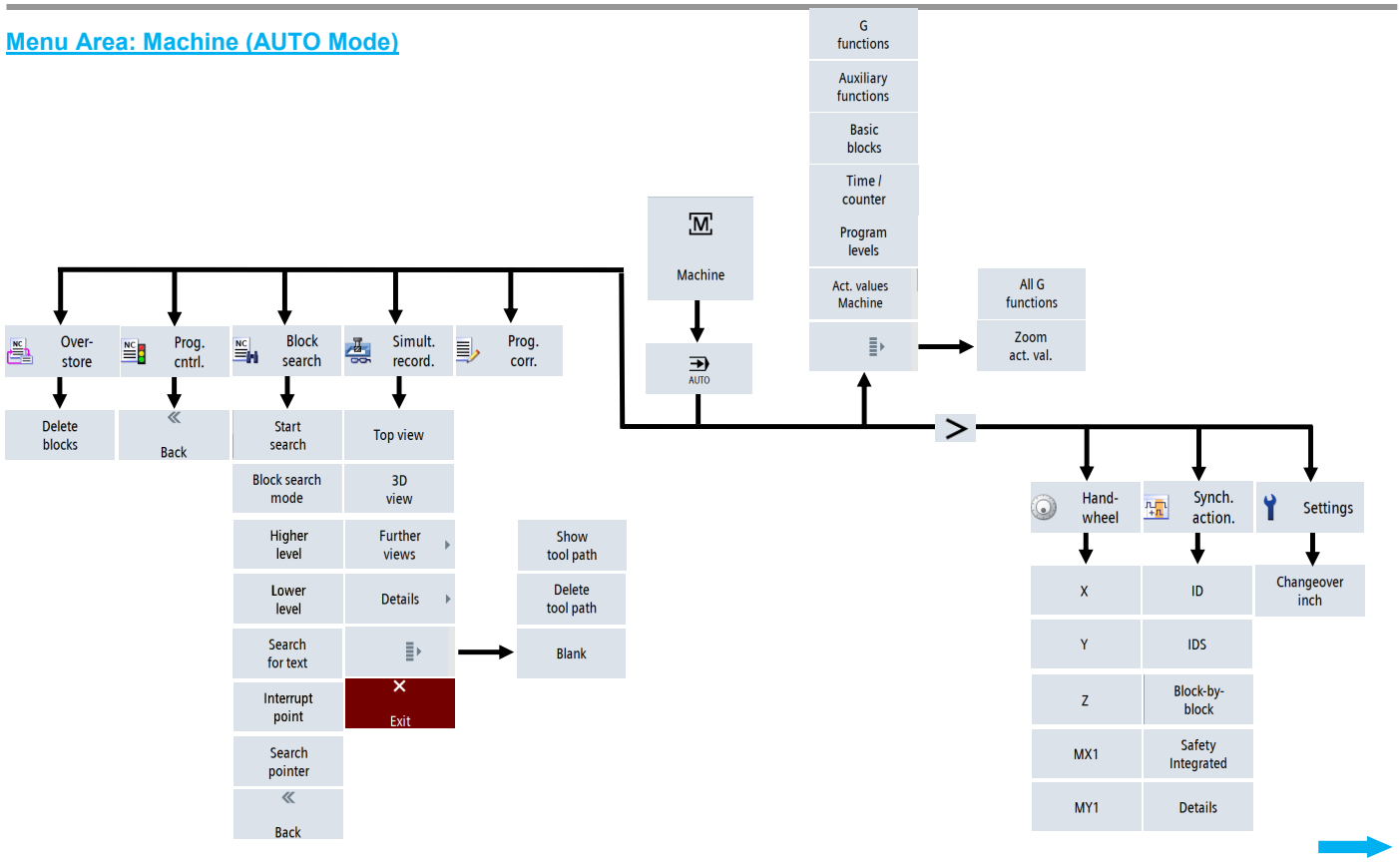


Notes:

Menu Area: Machine (MDA Mode)

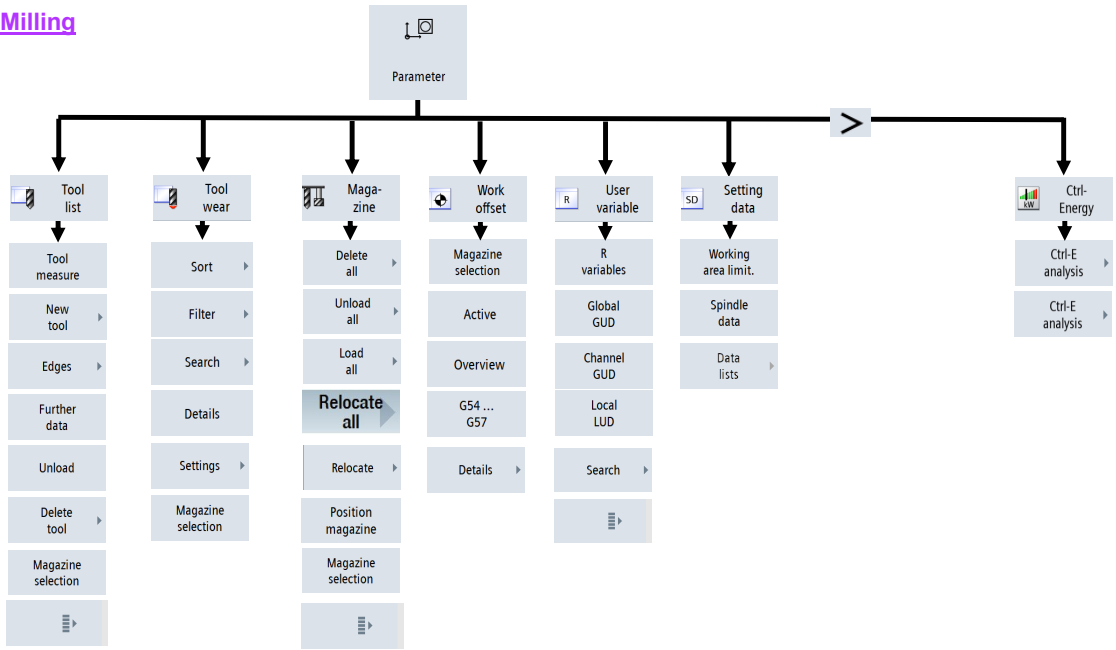


Menu Area: Machine (AUTO Mode)

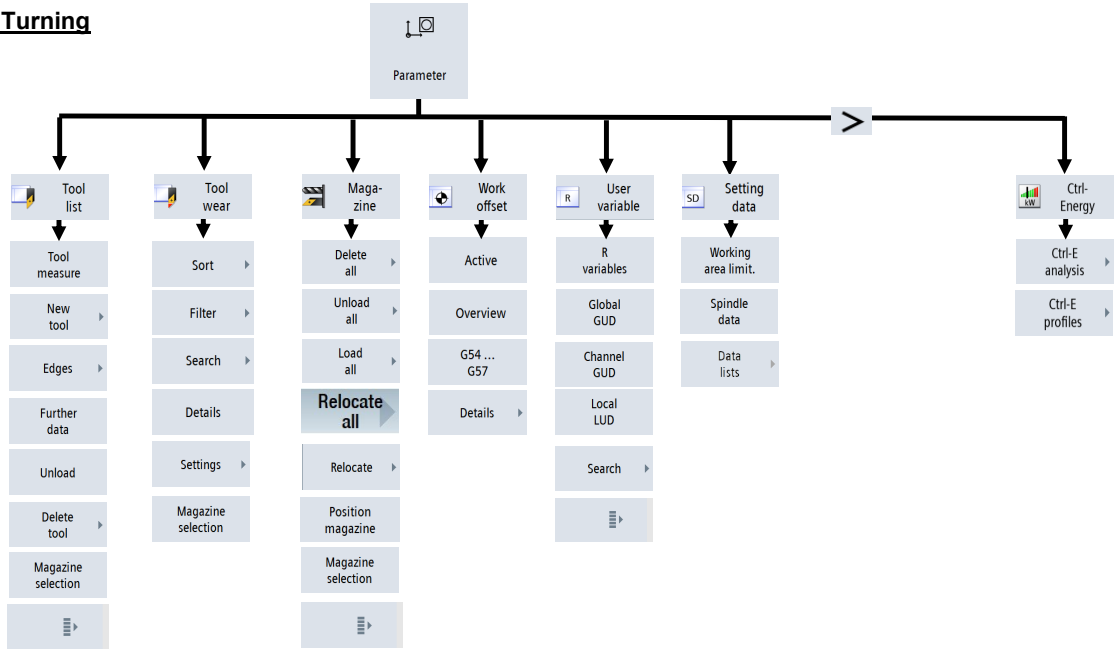


Notes:

Menu Area: Parameter Milling

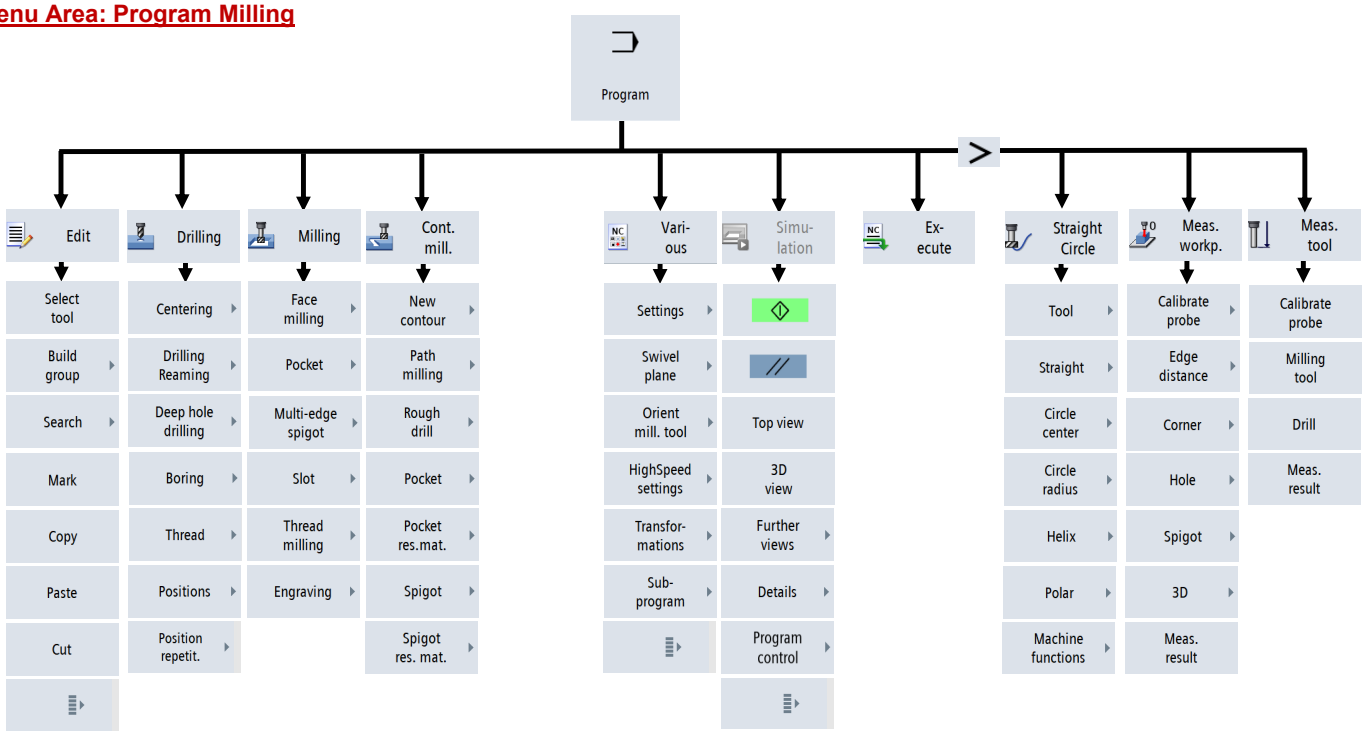


Menu Area: Parameter Turning

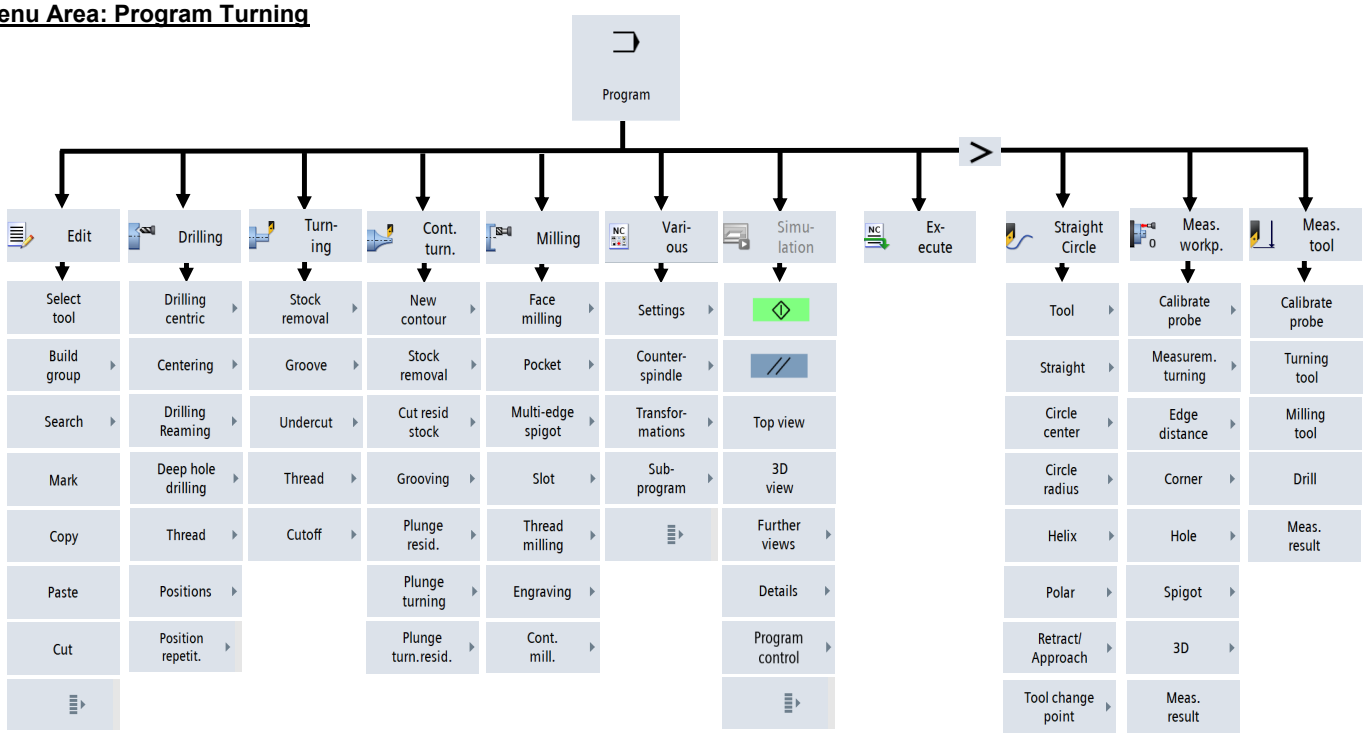


Notes:

Menu Area: Program Milling

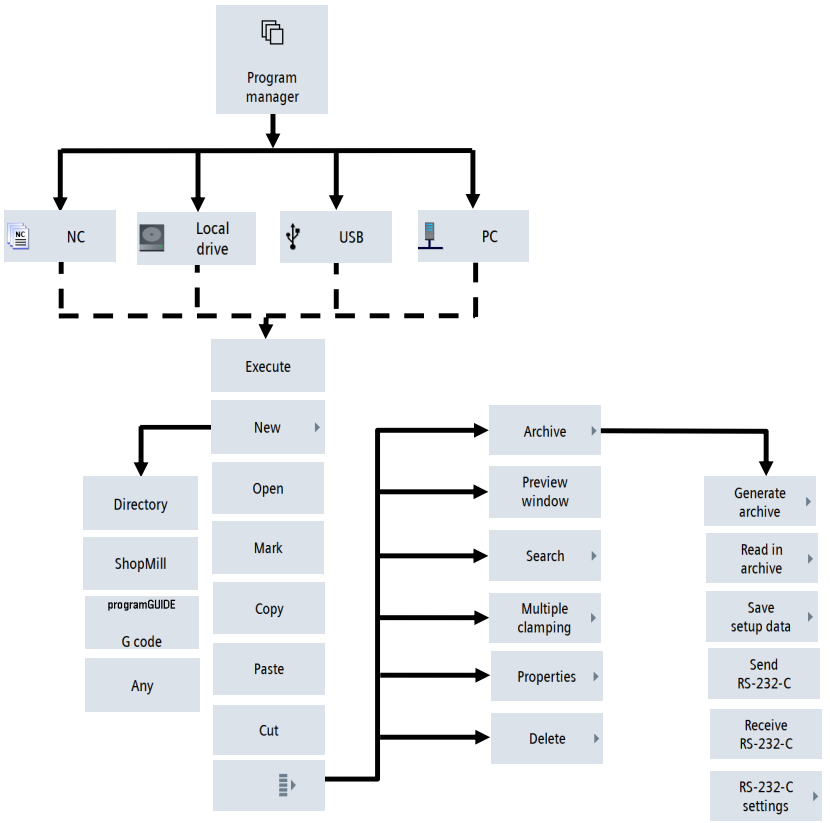


Menu Area: Program Turning

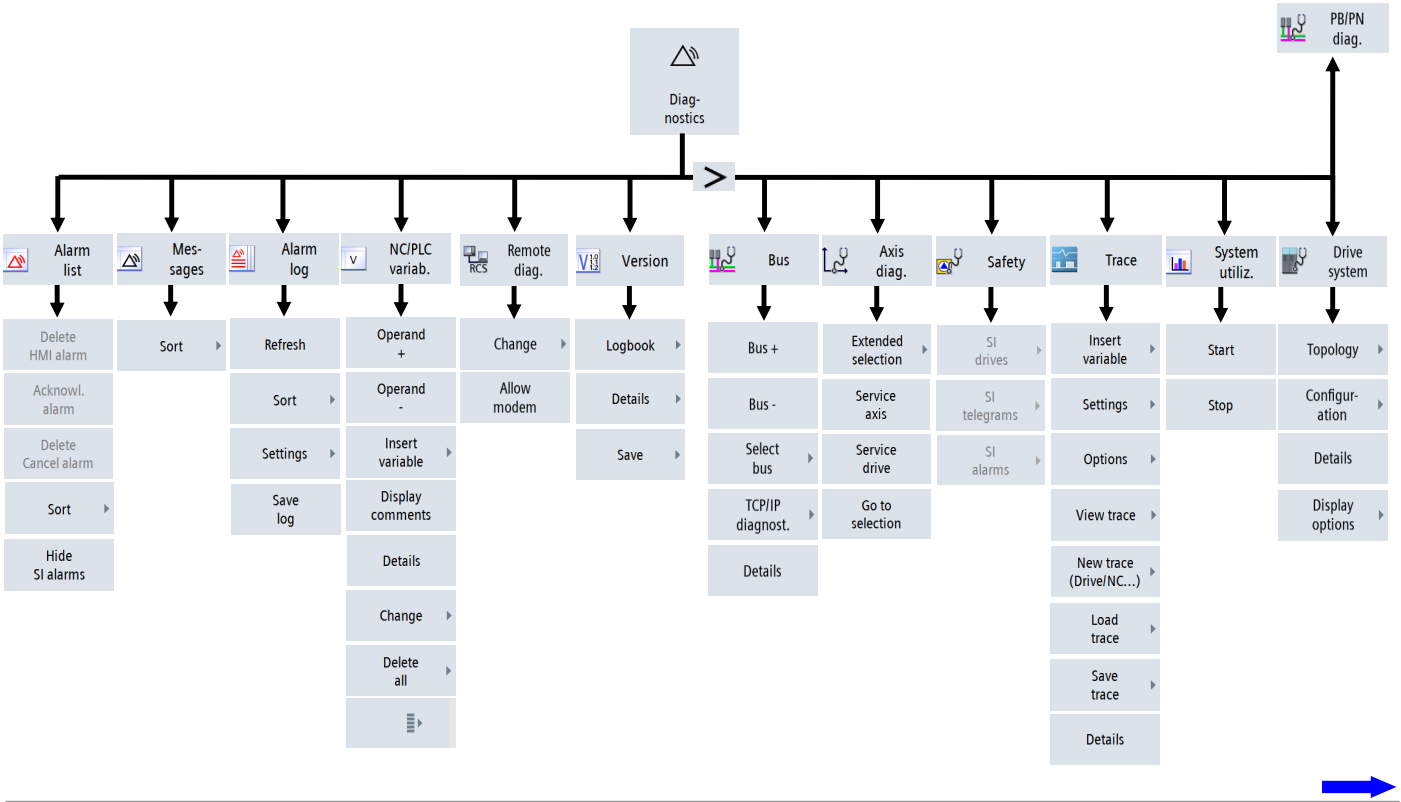


Notes:

Menu Area: Program Manager



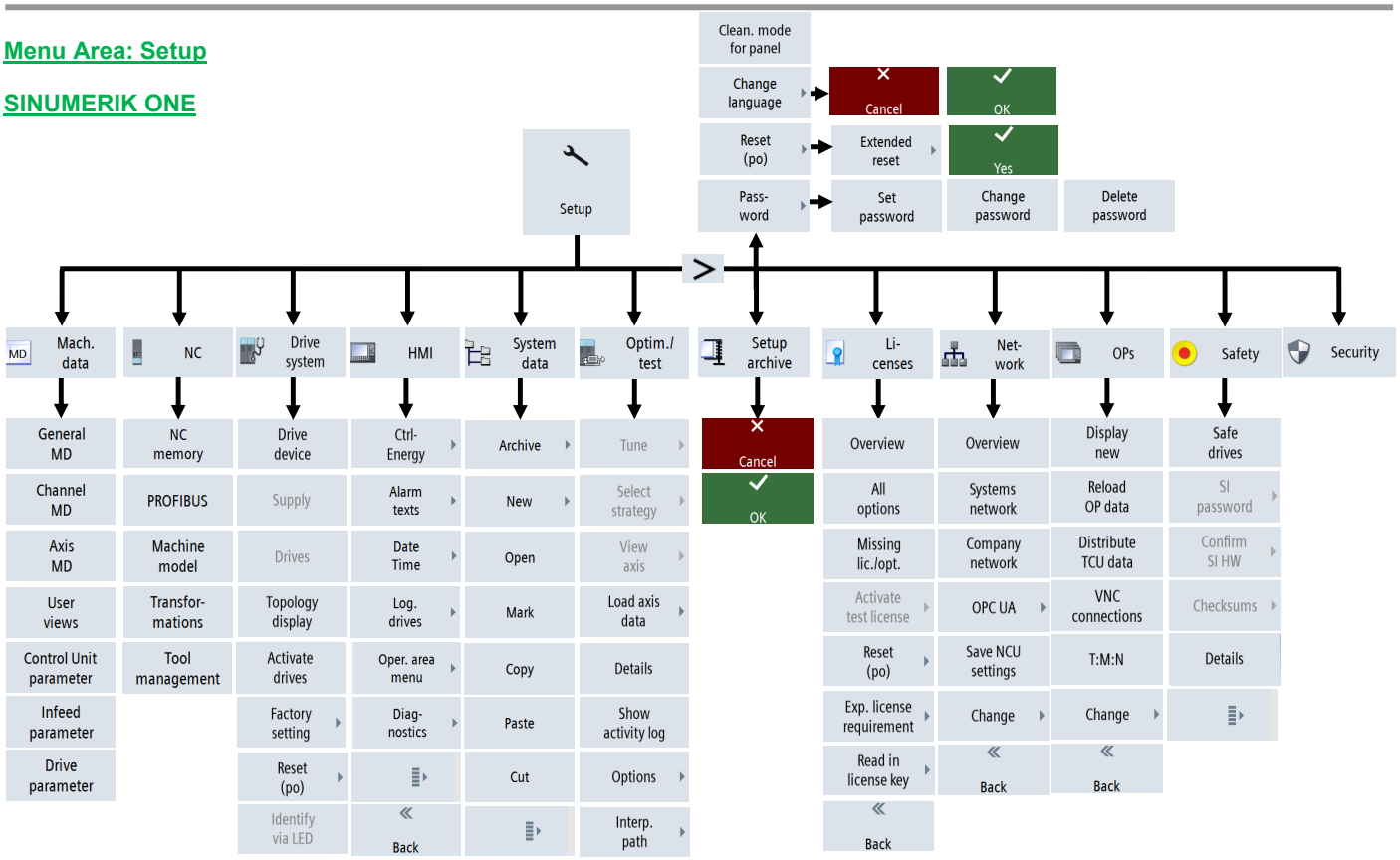
Menu Area: Diagnostics



Notes:

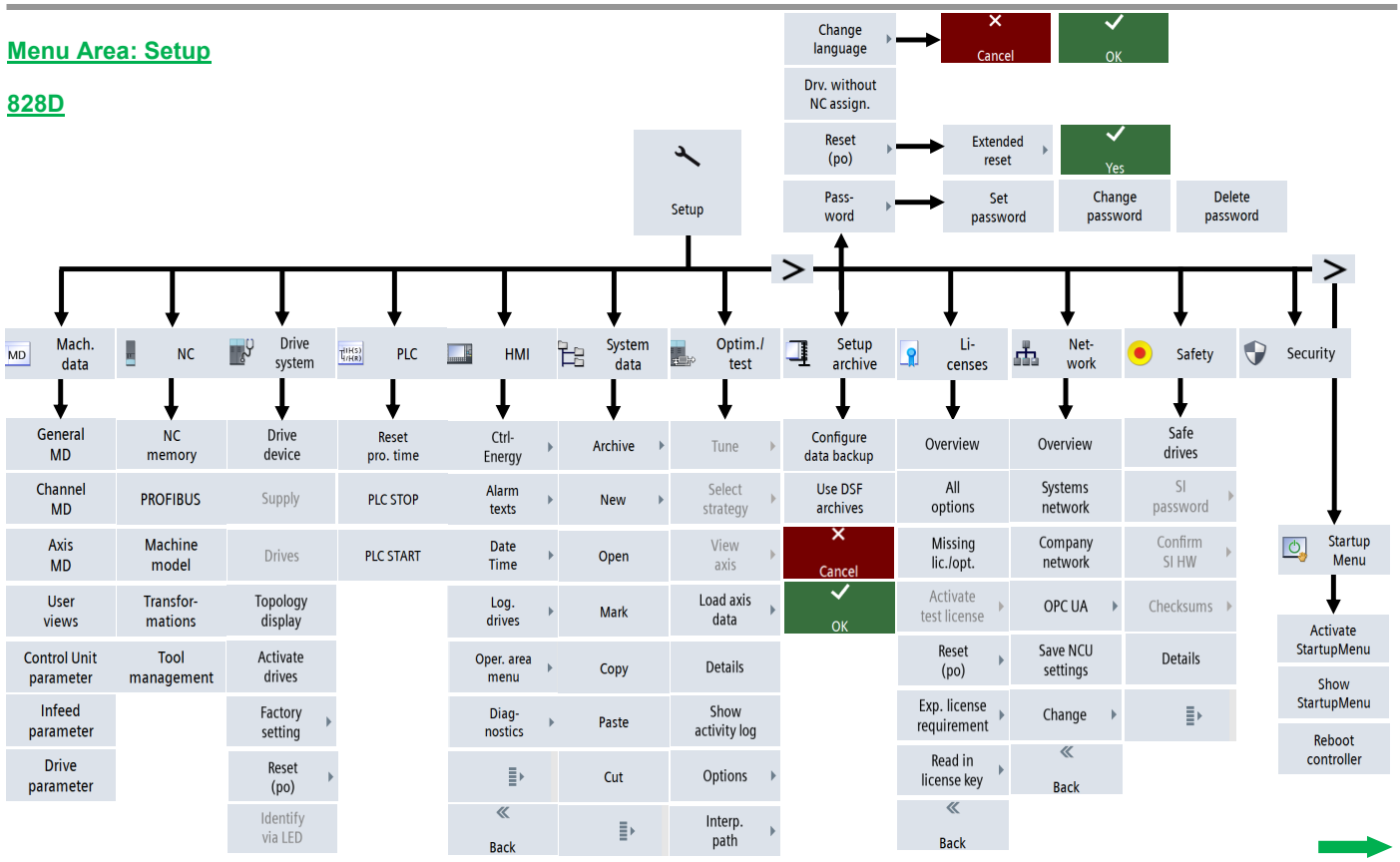
Menu Area: Setup

SINUMERIK ONE



Menu Area: Setup

828D



Notes:

Short cut Key sequences

Keys	Function
PAGE UP + SHIFT	In the program manager and in the program editor from the cursor position, selects directories or program blocks up to the beginning of the window.
PAGE UP + CTRL	Positions the cursor to the topmost line of a window.
PAGE DOWN + SHIFT	In the program manager and in the program editor from the cursor position, selects directories or program blocks up to the end of the window.
PAGE DOWN + CTRL	Positions the cursor to the lowest line of a window.
▶ + CTRL	Editing box - Moves the cursor further to the right by one word. Navigation - Moves the cursor in a table to the next cell to the right.

Keys	Function
◀ + CTRL	Editing box - Moves the cursor further to the left by one word. Navigation - Moves the cursor in a table to the next cell to the left.
▲ + CTRL	Editing box - Moves the cursor in the table to the beginning of the table. Navigation - Moves the cursor to the beginning of a window.
▲ + SHIFT	In the program manager and in the program editor, selects a contiguous selection of directories and program blocks.
▼ + CTRL	Editing box - Moves the cursor in the table to the end of the table. Navigation - Moves the cursor to the end of a window.
▼ + SHIFT	In the program manager and in the program editor, selects a contiguous, selection of directories and program blocks.

Notes:

Keys	Function
END + SHIFT	Moves the cursor to the last entry.
END + CTRL	Moves the cursor to the last entry in the last line.
PAGE DOWN + SHIFT	In the program manager and in the program editor from the cursor position, selects directories or program blocks up to the end of the window.
BACKSPACE + CTRL	Deletes a word selected to the left of the cursor.
CTRL + A	In the actual window, selects all entries (only in the program editor and program manager).

Keys	Function
CTRL + C	Copies the selected content. The text is located in the clipboard.
CTRL + L	Scrolls the actual user interface through all installed languages one after the other.
CTRL + SHIFT + L	Scrolls the actual operator interface through all installed languages in the inverse sequence.
CTRL + P	Generates a screenshot from the actual operator interface and saves it as a file in the HMI data/Logs/Screenshots folder.
CTRL + X	Cuts out the selected text. The text is located in the clipboard.

Notes:

Keys	Function
CTRL + Y	Reactivates changes that were undone (only in the program editor).
CTRL + V	Inserts text from the clipboard: either pastes the text from the clipboard at the actual cursor position or pastes text from the clipboard at the position of the selected text.
CTRL + ALT + D	Backs up the log files on the USB-FlashDrive. If a USB-FlashDrive is not inserted, then the files are backed-up in the manufacturer's area of the System SD-Card.

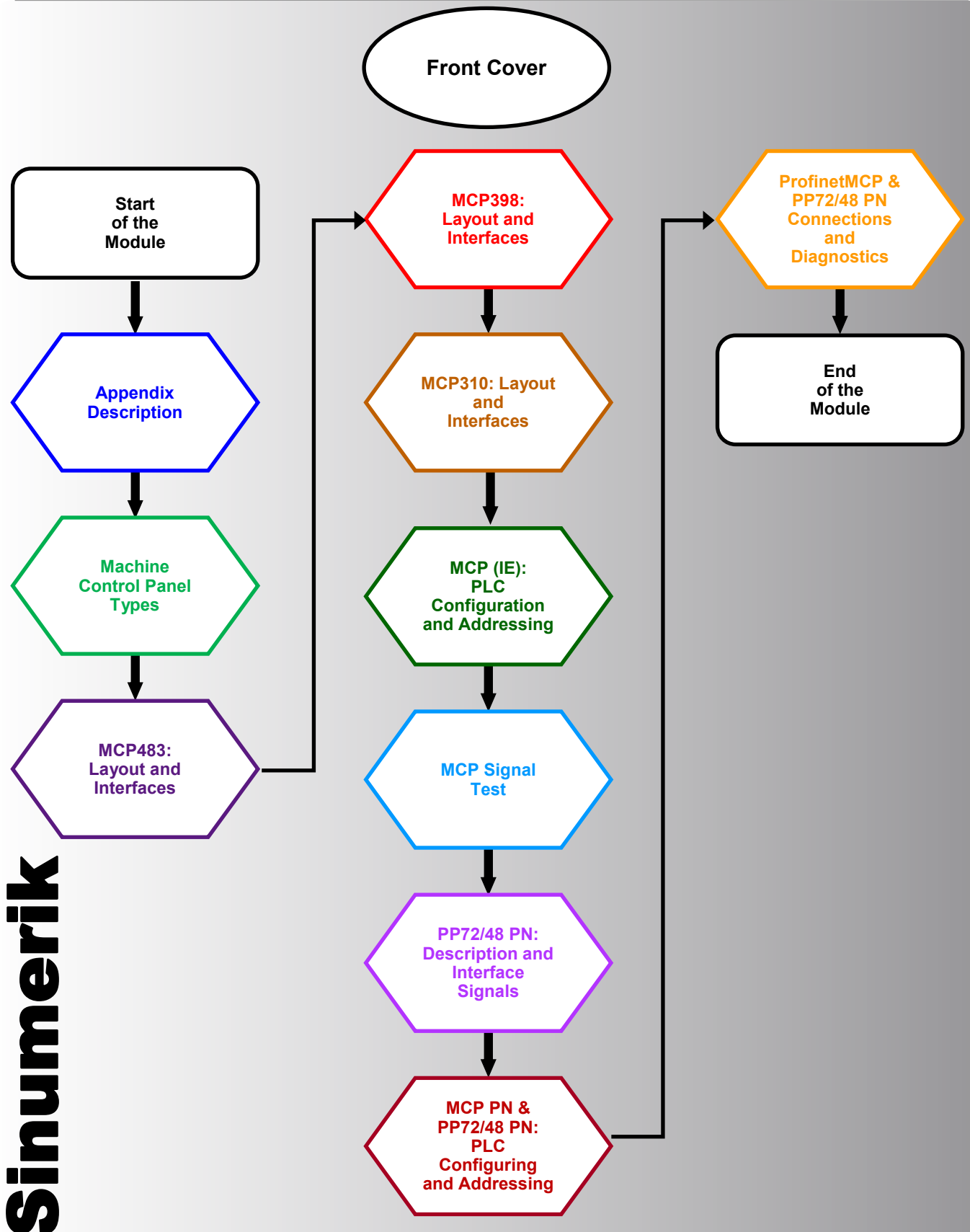
Keys	Function
CTRL + M	Feed over ride to maximum in Simulation
CTRL + ▲	Feed over ride is increased in Simulation
CTRL + ▼	Feed over ride to decreased in Simulation
ALT + S	Opens the Editor to enter Asian characters.
DEL + CTRL	Editing box - Deletes the first word to the of the cursor. Navigation - Deletes all characters

Appendix E: END

Notes:

Appendix F
SINUMERIK ONE

MCP & PP72/48 PN Connections and Diagnostics



This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

Appendix Description

Machine Control Panel (MCP) :
The MCP handles signals such as axis selection, jog commands, cycle start etc.
It is possible to have a maximum of two MCPs per controller.
They can be connected via the system network (interface X120) or the Profinet network (interface X150) of the Sinumerik ONE controller.
Each MCP needs to be uniquely addressed via its addressing switch and the parameterisation of PLC function block, FB1.
FB1 is also used to set start address of the MCP I/O and connection type (System network or Profinet)

PP72/48 PN:
The PP72/48 PN modules are used for the connection of external PLC I/O such as relays, lamps, switches etc.
They are connected to the Profinet network (interface X150) and have an addressing switch which must be set to a unique address. This address must match the one set in the PLC hardware configuration.

This module shows how the MCPs are addressed, how the parameterisation of the function block is made, and the Profinet configuration of PP72/48 I/O modules.

Notes:

Machine Control Panel Types

Machine control panels are available in three formats (MCP483, MCP398 and MCP310). The MCP483 is 483mm wide, the MCP398 is 398mm wide and the MCP310 is 310mm wide.
Mechanical and membrane button types are available for the MCP483 and MCP310, while the MCP398 is only available in with membrane type keys.

The major differences between the three formats are the number of keys/layout of keys etc. The input/output assignment also differs.
The functionality and purpose of the three variants is, however, identical (controlling of axes, mode selection etc).
There are no separate MCPs for turning or milling technologies.

Part numbers:

MCP483C PN (Mechanical type keys) **6FC5303-0AF22-0AA1**
MCP483 PN (Membrane type keys) **6FC5303-0AF22-1AA1**

MCP398C PN (Mechanical type keys) **6FC5303-0AF25-0AA0**

MCP310C PN (Mechanical type keys) **6FC5303-0AF23-0AA1**
MCP310 PN (Membrane type keys) **6FC5303-0AF23-1AA1**

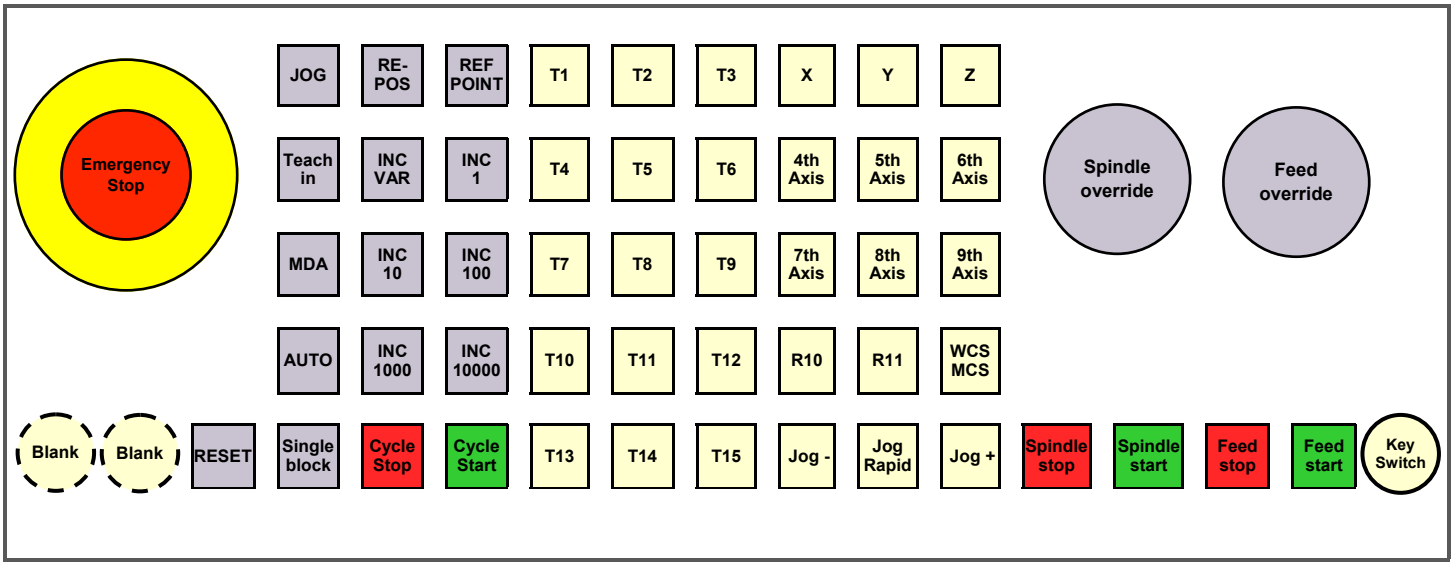
MCP483: Layout and Interfaces

Front View:



Notes:

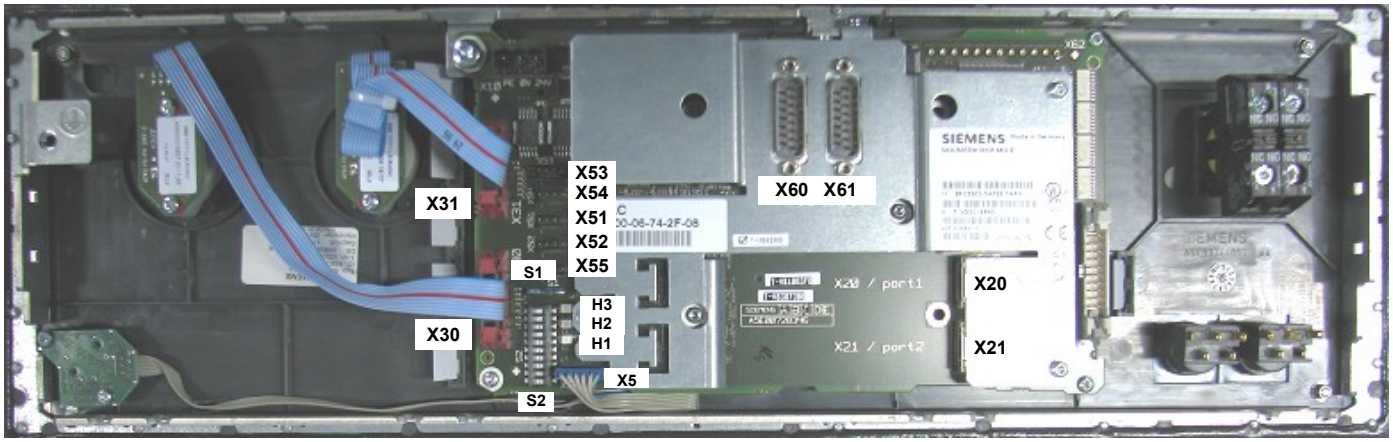
MCP483 Button Identification



Note: Cycle stop, Feed stop & Spindle stop signals are normally closed.

Buttons, T1 to T15 & R10/R11, are free for use by the manufacturer.

MCP483 PN Rear layout



Interface	Description
S1	Hand wheel signal type (see * below)
S2	Addressing switch
X5	Key-switch interface
X10	24V DC supply
X20	Port 1 Ethernet connection in.
X21	Port 2 Ethernet connection out.
X30	Feed rate override interface.
X31	Spindle override interface

Interface	Description
X51	Digital inputs
X52	Digital inputs
X53	Digital outputs
X54	Digital outputs
X55	Digital inputs
X60	Hand wheel
X61	Hand wheel

LED	Description
H1	Power OK (Green)
H2	PROFINET communication ok
H3	PROFINET group fault

* Hand wheel switch S1 - Open (default): TTL. Closed: Differential

Notes:

MCP483 Interface Signals.

The following tables show the interface signals for an MCP483. The letter **n** should be substituted for the ***Pointer*** value, entered when FB1 was parameterised (described later in this module).

MCP483 PN Signals from machine control panel (keys)								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IBn	Spindle override				Operating Mode			
	D	C	B	A	JOG	TEACH IN	MDA	AUTO
IBn+1	REPOS	REF	INC var.	10000 INC	INC 1000	INC 100	INC 10	INC 1
IBn+2	Key-switch position 0	Key-switch position 2	Spindle start	*Spindle stop	Feed start	*Feed stop	NC Start	*NC Stop
IBn+3	Reset	Key-switch position 1	Single block	Feedrate override				
				E	D	C	B	A
IBn+4	Direction keys			Key switch position 3	Axis selection keys			
	Plus R15	Minus R13	Rapid traverse R14		X R1	4th axis R4	7th axis R7	Spare R10
IBn+5	Axis selection keys							
	Y R2	Z R3	5th axis R5	MCS/WCS R12	Spare R11	9th Axis R9	8th axis R8	6th axis R6
IBn+6	Unassigned customer keys							
	T9	T10	T11	T12	T13	T14	T15	T16
IBn+7	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8
IBn+10	KT-IN8 (X55.2)	KT-IN7 (X55.1)	KT-IN6 (X52.3)	KT-IN5 (X52.2)	KT-IN4 (X52.1)	KT-IN3 (X51.3)	KT-IN2 (X51.2)	KT-IN1(X51.1)
IBn+11								KT-IN9 (X55.3)

MCP483 PN Signals to machine control panel (LEDs)								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
QBn	Machine function				Operating mode			
	INC 1000	INC 100	INC 10	INC 1	JOG	TEACH IN	MDA	AUTO
QBn+1	Feed start	*Feed stop	NC Start	*NC Stop	REPOS	REF	INC var.	INC 10000
QBn+2	Direction Minus R13	X R1	4th axis R4	7th axis R7	R10	Single block	Spindle start	*Spindle stop
QBn+3	Z R3	5th axis R5	MCS/WCS	R11	R9	8th axis R8	6th axis R6	Direction Plus R15
QBn+4	Unassigned customer keys							Y
	T9	T10	T11	T12	T13	T14	T15	R2
QBn+5	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8
QBn+6							Reset	R14
QBn+7			KT-OUT6 (X54.3)	KT-OUT5 (X54.2)	KT-OUT4 (X54.1)	KT-OUT3 (X53.3)	KT-OUT2 (X53.2)	KT-OUT1 (X53.1)

Signals marked with * are normally closed signals.

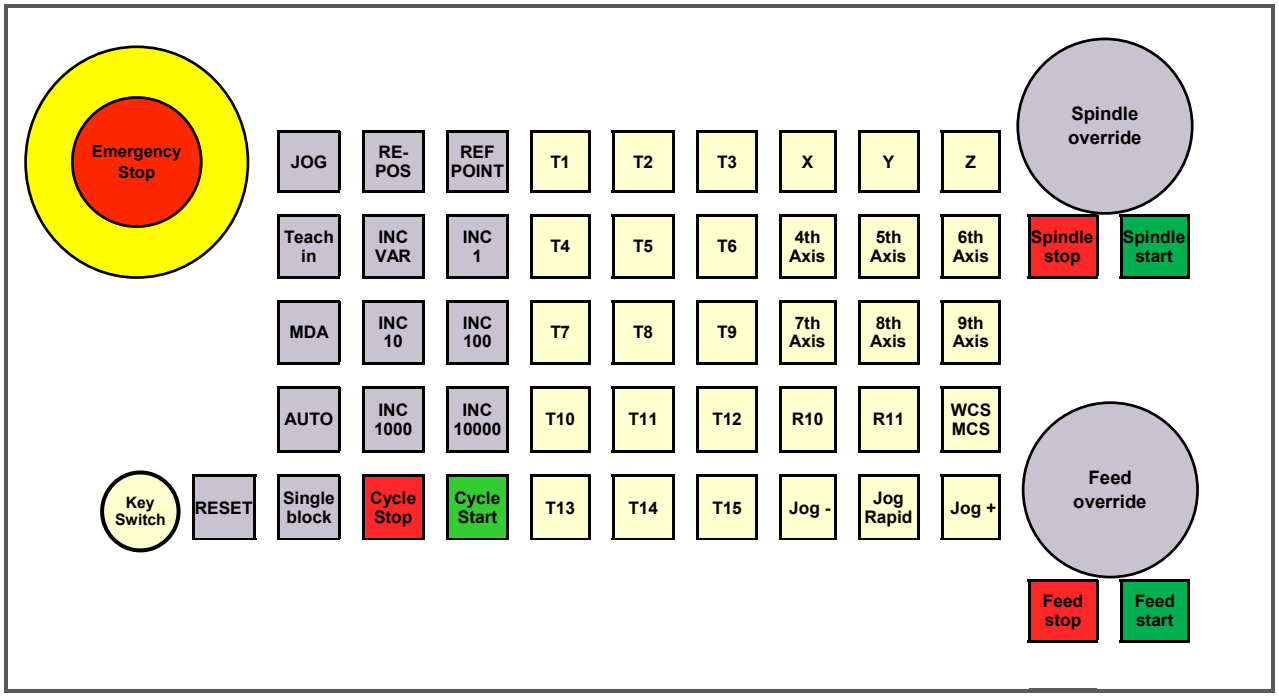
Notes:

MCP398: Layout and Interfaces

6FC5303-0AF25-0AA0 MCP398C PN (Mechanical type keys)



MCP398 Button Identification

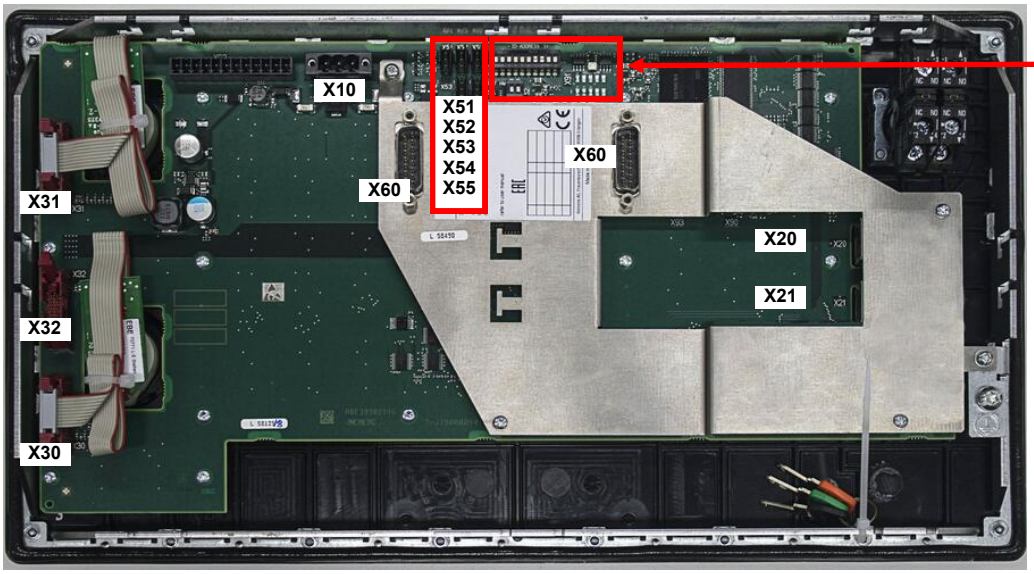


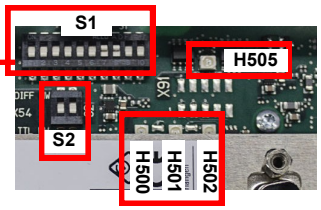
Note: Cycle stop, Feed stop & Spindle stop signals are normally closed.

Buttons, T1 to T15 & R10/R11, are free for use by the manufacturer.

Notes:

MCP398 PN Rear Layout





Above - detailed view of switches and LEDS.

Interface	Description
S1	Addressing switch
S2	Handwheel signal type switch
X10	24V DC supply
X20	Port 1 Ethernet connection in.
X21	Port 2 Ethernet connection out.
X30/X32	Feedrate override interface.
X31	Spindle override interface

Interface	Description
X51	Digital inputs
X52	Digital inputs
X53	Digital outputs
X54	Digital outputs
X55	Digital inputs
X60	Hand-wheel
X61	Hand-wheel

LED	Description
H500	Power OK (Green)
H501	PROFINET communication
H502	PROFINET group fault
H505	Over temperature

S2 - MCP398C handwheel signal settings		
S2	ON	OFF
Position 1	Differential	TTL
Position 2	Reserved	Reserved

Notes:

MCP398 Interface Signals.

The following tables show the interface signals for an MCP398C. The letter **n** should be substituted for the ***Pointer*** value entered when FB1 was parameterised (MCP1In, MCP2In, MCP1Out and MCP2Out).

MCP398C Signals from machine control panel (keys)								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IBn	Spindle override				Operating Mode			
	D	C	B	A	JOG	TEACH IN	MDA	AUTO
IBn+1	REPOS	REF	INC var.	10000 INC	INC 1000	INC 100	INC 10	INC 1
IBn+2	Key-switch position 0	Key-switch position 2	Spindle start	*Spindle stop	Feed start	*Feed stop	NC Start	*NC Stop
IBn+3	Reset	Key-switch position 1	Single block	Feedrate override				
				E	D	C	B	A
IBn+4	Jog keys				Axis selection keys			
	Plus R15	Minus R13	Rapid R14	Key switch position 3	X R1	4th axis R4	7th axis R7	Spare R10
IBn+5	Axis selection keys							
	Y R2	Z R3	5th axis R5	MCS/WCS R12	Spare R11	9th Axis R9	8th axis R8	6th axis R6
IBn+6	Unassigned customer keys							
	T9	T10	T11	T12	T13	T14	T15	T16
IBn+7	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8
IBn+10	KT-IN8 (X55.2)	KT-IN7 (X55.1)	KT-IN6 (X52.3)	KT-IN5 (X52.2)	KT-IN4 (X52.1)	KT-IN3 (X51.3)	KT-IN2 (X51.2)	KT-IN1(X51.1)
IBn+11								KT-IN9 (X55.3)

MCP398C Signals to machine control panel (LEDs)								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
QBn	Machine function				Operating mode			
	INC 1000	INC 100	INC 10	INC 1	JOG	TEACH IN	MDA	AUTO
QBn+1	Feed start	*Feed stop	NC Start	*NC Stop	REPOS	REF	INC var.	INC 10000
QBn+2	Direction Minus R13	X R1	4th axis R4	7th axis R7	R10	Single block	Spindle start	*Spindle stop
QBn+3	Z R3	5th axis R5	MCS/WCS	R11	R9	8th axis R8	6th axis R6	Direction Plus R15
QBn+4	Unassigned customer keys							Y
	T9	T10	T11	T12	T13	T14	T15	R2
QBn+5	Unassigned customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8
QBn+6							Reset	R14
QBn+7			KT-OUT6 (X54.3)	KT-OUT5 (X54.2)	KT-OUT4 (X54.1)	KT-OUT3 (X53.3)	KT-OUT2 (X53.2)	KT-OUT1 (X53.1)

Signals marked with * are normally closed signals.

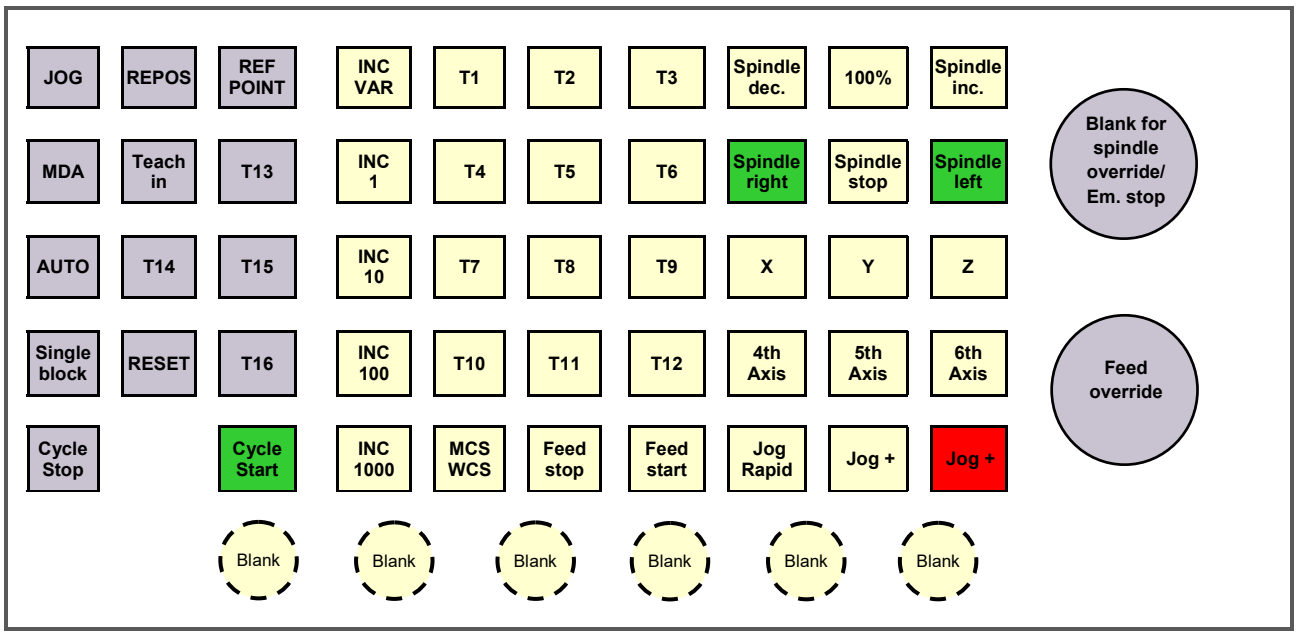
Notes:

MCP310: Layout and Interfaces

6FC5303-0AF23-0AA1 MCP310C PN (Mechanical type keys)
6FC5303-0AF23-1AA1 MCP310 PN (Membrane type keys)



MCP310 Button Identification



Note:- Cycle stop, Feed stop & Spindle stop-signals are inverted i.e. normally closed.
Buttons T1 to T16 are free for use by the manufacturer.
The six blanks can be removed to allow the user to add extra buttons/switches/lamps.

Notes:

MCP310 Interface Signals.

The following tables show the interface signals for an MCP310. The letter “n” should be substituted for the “Pointer” value entered when FB1 was parameterised.

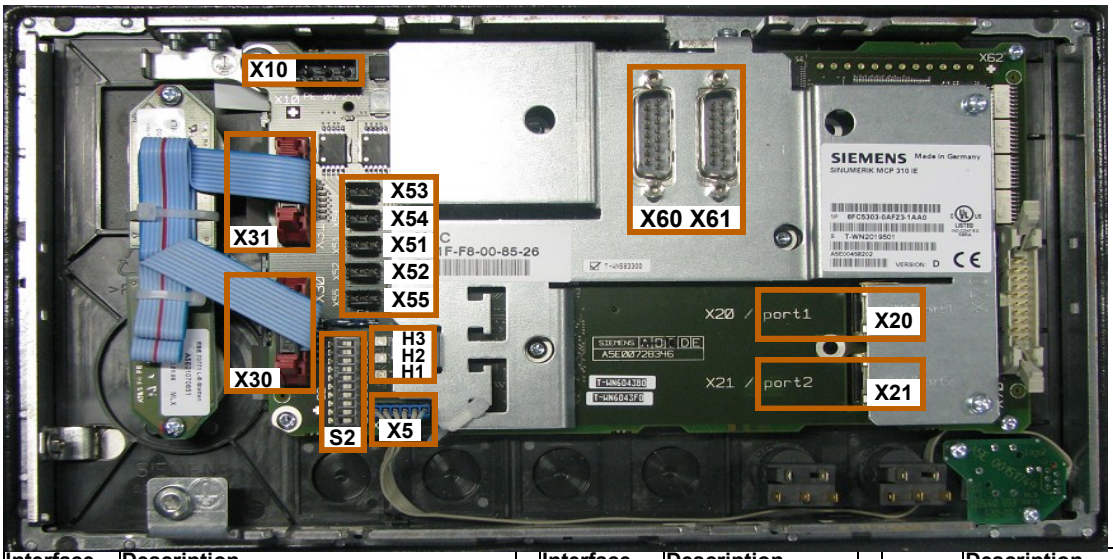
MCP310 PN Signals from machine control panel (keys)								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IBn	*NC Stop	Spindle -	Spindle 100%	Spindle +	Single block	JOG	MDA	AUTO
IBn+1	NC Start	Spindle right	*Spindle stop	Spindle left	Key-switch position 3	REF	REPOS	Teach IN
IBn+2	Feed start	*Feed stop	INC var.	Key-switch position 0	INC 1000	INC 100	INC 10	INC 1
IBn+3	Reset	Key-switch position 2	Key-switch position 1	Feedrate override E (16)	D (8)	C (4)	B (2)	A (1)
IBn+4	Direction keys +	Direction keys -	Rapid traverse	KT-IN5 (X52.2)	KT-IN4 (X52.1)	KT-IN3 (X51.3)	KT-IN2 (X51.2)	KT-IN (X51.1)
IBn+5	T16	KT6(X52.3)	6th axis	Axis selection 5th axis	4th axis	Z	Y	X
IBn+6	T9	T10	T11	T12	WCS MCS	T13	T14	T15
IBn+7	T1	T2	T3	T4	T5	T6	T7	T8
IBn+10	KT-IN8 (X55.2)	KT-IN7 (X55.1)	KT-IN6 (X52.3)	KT-IN5 (X52.2)	KT-IN4 (X52.1)	KT-IN3 (X51.3)	KT-IN2 (X51.2)	KT-IN (X51.1)
IBn+11								KT-IN9 (X55.3)
IBn+13				Spindle override E (16)	D (8)	C (4)	B (2)	A (1)

MCP 310 PN Signals to machine control panel (LEDs)								
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
QBn	*NC Stop	Spindle Dec.	Spindle 100%	Spindle Inc.	Single Block	JOG	MDA	AUTO
QBn+1	NC start	Spindle right	*Spindle Stop	Spindle left	RESET	REF	REPOS	Teach In
QBn+2	Feed Start	*Feed Stop	INC var.		INC 1000	INC 100	INC 10	INC 1
QBn+3								
QBn+4	Minus Direction	Plus Direction	Rapid traverse	KT-OUT5 (X54.2)	KT-OUT4 (X54.1)	KT-OUT3 (X53.3)	KT-OUT2 (X53.2)	KT-OUT1 (X53.1)
QBn+5	T16	KT-OUT6 (X54.3)	6th axis	5th axis	4th axis	Z	Y	X
QBn+6	T9	T10	T11	T12	MCS/WCS	T13	T14	T15
QBn+7	T1	T2	T3	T4	T5	T6	T7	T8

Signals marked with * are normally closed contacts..

Notes:

MCP310 Rear Layout



Interface	Description
S1	Hand wheel signal type (see * below)
S2	Addressing switch
X5	Key-switch interface
X10	+24V DC supply
X20	Port 1 Ethernet connection in.
X21	Port 2 Ethernet connection out.
X30	Feed rate override interface.
X31	Spindle override interface

Interface	Description
X51	Digital inputs
X52	Digital inputs
X53	Digital outputs
X54	Digital outputs
X55	Digital inputs
X60	Hand-wheel
X61	Hand-wheel

	Description
	Power OK (Green)
	PROFINET communication ok
	PROFINET group fault

* Hand wheel switch S1 - Open (default): TTL. Closed: Differential

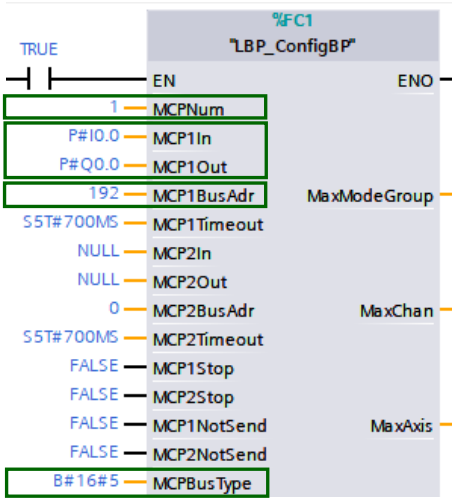
Notes:

PLC Configuration and Addressing

Determining the MCP parameters

Function Block, FB1, is used for configuring the I/O assignment, bus address and bus type of the MCPs.
FB1 is called from within Organisation Block, OB100. There are many additional parameters associated with FB1.

The following is an example of the FB1 parameters for setting for a single Ethernet MCP with an I/O start address of 0, and a Bus address of 192.



Parameter description:

MCPNum Number of MCPs on the system.

MCP1In Pointer - the start address of the MCP inputs.

MCP1Out Pointer - the start address of the MCP outputs.

MCP1BusAdr Addressing of MCP - MCP dip-switch needs to be set to the same address.

MCP1BusType Bus type; (5 = Ethernet, 6 = PROFINET)

System network, interface X120, is a DHCP server with a default IP address of 192.168.214.1 and a subnet mask of 255.255.255.0.

Module, ***A033 - PLC Functions,*** describes the full configuration procedure for the MCPs.

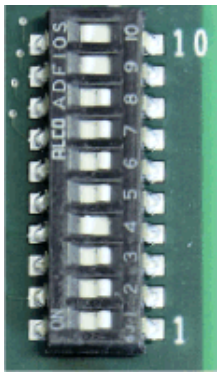
Each MCP connected to the network needs a unique address. This is determined by the setting of S1 (MCP398) or S2 MCP310/MCP483) on the MCP.

Each switch position, from 1 to 8, represents a value which is active when the switch position is set to ***On***.
All switches in the ***On*** position have their values added together to give the address of the MCP.

Switch positions 9 & 10 must be set to ***Off*** for Ethernet operation or ***On*** for PROFINET operation.

Below - switch S1/S2 positions and relevant values.

Switch S1/S2 Position	Value when in ***On*** state
1	1
2	2
3	4
4	8
5	16
6	32
7	**64**
8	**128**
9	**OFF**
10	**OFF**



Switch S1/S2

The following table shows examples of S2 switch positions and the resulting addresses, including the default address of 192.

Switch S1/S2										Meaning/ address
10	9	8	7	6	5	4	3	2	1	
Off	Off	-	-	-	-	-	-	-	-	Ethernet
On	On	-	-	-	-	-	-	-	-	PROFIBUS
-	-	Off	Off	Off	Off	Off	Off	Off	Off	0
-	-	Off	Off	Off	Off	Off	Off	Off	On	1
-	-	Off	Off	Off	Off	Off	Off	On	Off	2
-	-	Off	Off	Off	Off	Off	Off	On	On	3
-	-	:	:	:	:	:	:	:	:	:
-	-	**On**	**On**	Off	Off	Off	Off	Off	Off	**192 (Delivery setting)**
-	-	:	:	:	:	:	:	:	:	:
-	-	On	On	On	On	On	On	On	Off	254
-	-	On	On	On	On	On	On	On	On	255

Notes:

MCP User I/O Pin Assignment

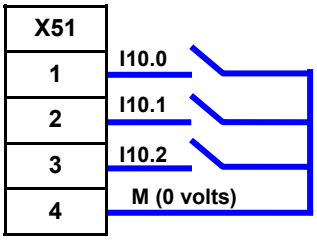
The 9 inputs and 6 outputs available on plugs X51 to X55 are designed for use with switches/buttons and indicator lamps which can be installed in the blanked holes of the MCP. Below shows the I/O assignment of each individual plug.

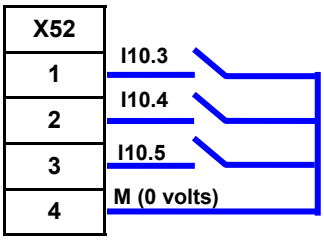
The input terminals in the normally open state measure 5 V DC (Supplied by the MCP)
The inputs work by sinking the 5 V DC signals to 0 V (M).

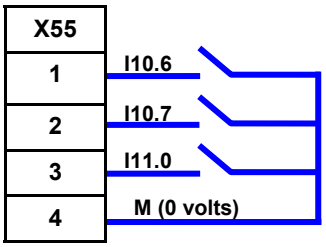
The outputs terminals are supplied with +24 V DC from the MCP.

No external voltages should be applied to the input and output terminals.

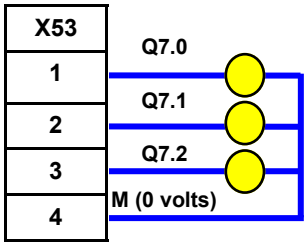
Input Assignment (X51, X52 & X55):
The input addresses shown are assuming FB1 parameter, MCP1In:= P#I 0.0, for an MCP483.

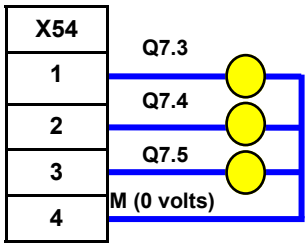






Output Assignment (X53 & X54): The output addresses shown are assuming FB1 parameter, MCP1Out:= P#Q 0.0, for an MCP483.

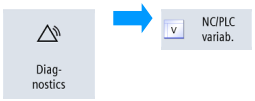




The outputs are capable of supplying 1.2 Watts per output, it is not permissible to connect relays, valves or other inductive devices.

MCP Signal Test

The signals to and from the MCP can be checked using the on-board diagnostic feature of the controller.
To locate the diagnostic screens use the following soft-key selection:



On the screen it is possible to enter the inputs and outputs of the MCP and check the functionality of each button/key.

Example - monitoring the status of MCP398C IB3, I3.6, QB0 & Q0.3 (Assuming FB1 parameters MCP1In:= P#I 0.0 and MCP1Out:= P#Q 0.0)

NC/PLC variables

Variable	Format	Value
IB3	B	01001101
I3.6	B	1
QB0	B	00001000
QB0.3	D	1

MCP398C Signals from machine control panel (keys)								
Byte	Bit 7	**Bit 6**	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IBn+3	Reset	**Key-switch position 1**	Single block	Feed rate override	Feed rate override	Feed rate override	Feed rate override	Feed rate override
MCP398C Signals to machine control panel (LEDs)								
Byte	Bit 7	**Bit 6**	Bit 5	Bit 4	**Bit 3**	Bit 2	Bit 1	Bit 0
QBn	*NC Stop	Spindle Dec.	Spindle 100%	Spindle Inc.	**Single Block**	JOG	MDA	AUTO

Notes:

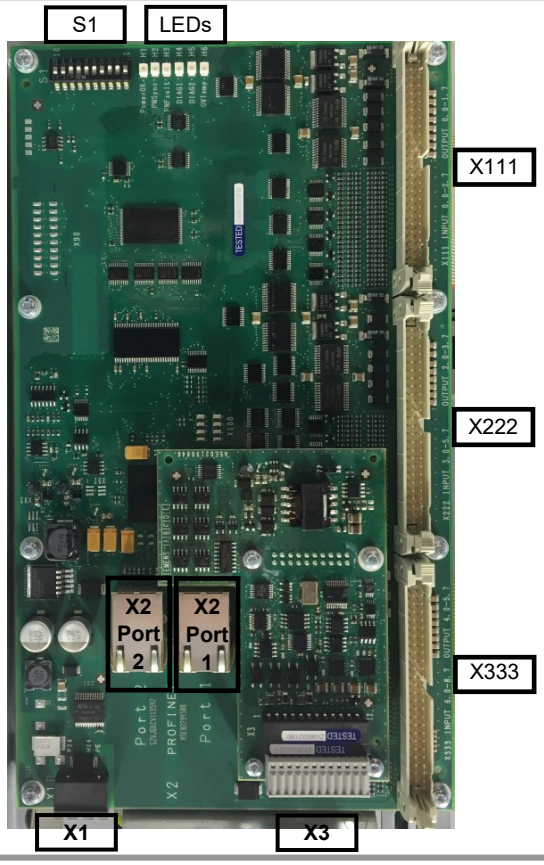
PP72/48 PN Description and Interface Signals

PP72/48 PN modules are a compact component which connect to the Profinet network (Ports X150 P1/P2 of the NCU).

There are two variants, one of which is all digital while the second offers both digital and analogue functionality.

A 24 V DC supply is required.
In total 72 digital inputs and 48 digital outputs are available on each of the two variants.
The analogue module has an additional two analogue inputs and two analogue outputs.

Part number:- 6FC5311-0AA00-0AA0 Digital only
(PP72/48D PN)
6FC5311-0AA00-1AA0 Digital/analogue
(PP72/48 2/2A PN)



PP72/48 PN Interface description:

Interface	Description
S1	Addressing switch
X1	+24V DC supply
X2	PROFINET - ports 1 & 2
X3	Analogue interface - option
X111	3 Input bytes/2 output bytes
X222	3 Input bytes/2 output bytes
X333	3 Input bytes/2 output bytes

PP72/48 PN LED description:

Name	Functionality	Colour	Description
H1	Power	Green	On: Power OK Off: One of the generated logic voltages is low.
H2	PNSync	Green	ON: Bus cycle clock is synchronised. Off: Bus cycle clock is not synchronised Flashing @ 0.5 Hz: Bus clock cycle is synchronised and cycli data exchange is running.
H3	PNFault	Red	Off: No errors present, data exchange with all configured I/O devices is running. On: Serious bus fault due to one the following: • No physical connection to a subnet/switch. • Incorrect transmission rate. • Full duplex transmission is not activated.
H4	DIAG1	Green	Reserved
H5	DIAG2	Green	reserved
H6	OVTemp	Red	Over temperature Indication.

Notes:

Ribbon cable connections.

The following tables show the pins and I/O addresses for ribbon connectors X111, X222, X333.
This is assuming the PP72/48 PN is configured with a start address of 80 (Input bytes 80, 81, & 82 and output bytes 80 & 81) This is determined in the PLC hardware configuration.

Pin assignment X111

Pin	Signal	Pin	Signal	Pin	Signal
1	0V DC	18	I81.7	35	Q80.4
2	+24VDC O/P	19	I82.0	36	Q80.5
3	I80.0	20	I82.1	37	Q80.6
4	I80.1	21	I82.2	38	Q80.7
5	I80.2	22	I82.3	39	Q81.0
6	I80.3	23	I82.4	40	Q81.1
7	I80.4	24	I82.5	41	Q81.2
8	I80.5	25	I82.6	42	Q81.3
9	I80.6	26	I82.7	43	Q81.4
10	I80.7	27	Not used	44	Q81.5
11	I81.0	28	Not used	45	Q81.6
12	I81.1	29	Not used	46	Q81.7
13	I81.2	30	Not used	47	+24V DC I/P
14	I81.3	31	Q80.0	48	+24V DC I/P
15	I81.4	32	Q80.1	49	+24V DC I/P
16	I81.5	33	Q80.2	50	+24V DC I/P
17	I81.6	34	Q80.3		

Pin assignment X222

Pin	Signal	Pin	Signal	Pin	Signal
1	0V DC	18	I84.7	35	Q82.4
2	+24V DC O/P	19	I85.0	36	Q82.5
3	**I83.0***	20	I85.1	37	Q82.6
4	**I83.1***	21	I85.2	38	Q82.7
5	**I83.2***	22	I85.3	39	Q83.0
6	**I83.3***	23	I85.4	40	Q83.1
7	**I83.4***	24	I85.5	41	Q83.2
8	**I83.5***	25	I85.6	42	Q83.3
9	**I83.6***	26	I85.7	43	Q83.4
10	**I83.7***	27	Not used	44	Q83.5
11	I84.0	28	Not used	45	Q83.6
12	I84.1	29	Not used	46	Q83.7
13	I84.2	30	Not used	47	+24V DC I/P
14	I84.3	31	Q82.0	48	+24V DC I/P
15	I84.4	32	Q82.1	49	+24V DC I/P
16	I84.5	33	Q82.2	50	+24V DC I/P
17	I84.6	34	Q82.3		

***Note: The first input byte of X222 has FAST gating hardware**.

Fast Input Information:
X222, byte 1 - maximum delay time - 600µs.
All other inputs, delay time - 0.5 to 3ms.

Pin	Signal	Pin	Signal	Pin	Signal
1	0V DC	18	I87.7	35	Q84.4
2	+24V DC O/P	19	I88.0	36	Q84.5
3	I86.0	20	I88.1	37	Q84.6
4	I86.1	21	I88.2	38	Q84.7
5	I86.2	22	I88.3	39	Q85.0
6	I86.3	23	I88.4	40	Q85.1
7	I86.4	24	I88.5	41	Q85.2
8	I86.5	25	I88.6	42	Q85.3
9	I86.6	26	I88.7	43	Q85.4
10	I86.7	27	Not used	44	Q85.5
11	I87.0	28	Not used	45	Q85.6
12	I87.1	29	Not used	46	Q85.7
13	I87.2	30	Not used	47	+24V DC I/P
14	I87.3	31	Q84.0	48	+24V DC I/P
15	I87.4	32	Q84.1	49	+24V DC I/P
16	I87.5	33	Q84.2	50	+24V DC I/P
17	I87.6	34	Q84.3		

Notes:

MCP PN & PP72/48 PN: PLC Configuring and Addressing

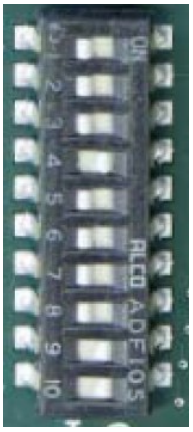
Each item connected to the PROFINET network needs a unique address.
This is determined by the setting of S1 (PP72/48, MCP310, MCP483) S2 on the MCP and S1 on the PP72/48.

Switch S1/S2

The selected IP address will determine the I/O that will be assigned. This is seen in the following tables:
For all PROFINET applications, switch positions 9 and 10 need to be set to on.

Switch S1/S2 Position	Binary Value
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128
9	**Profinet=ON**
10	**Profinet=ON**

Switch S1 - PP72/48
Switch S2 - MCP



PROFINET Default Device Names:

The S1/S2 switches (positions 1 to 8) define the default device name of the PROFINET modules.
There are up to 128 default device names.
If these default device names are used, there is no need for initialization.

PP72/48 = pp72x48pn xxx
The PP72/48 PN module delivery address is 9 (S1 switch pos. 1 & 4 = on) - the default name (pp72x48pn9) must be set in the PLC hardware configuration.
MCP = mcp-pnxxx
The MCP delivery address is 192 (S2 switch pos. 8 & 9 = on)
This is outside the address range for PROFINET devices (0 to 127) .
The switch should be set to an suitable address (see the table)
Example: S2 switch pos. 1 = on will define the MCP with a device name of mcp-pn1 (the PLC hardware configuration should be set to match).

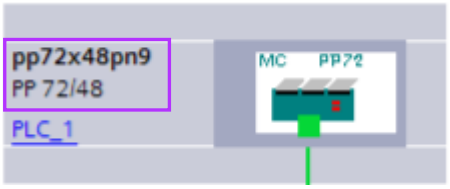
DCP-Mode
Switches S1 to S8 need to be switched to the on position.
In this mode, there is no default device name.
The device name must be set using initialization and remains saved on the I/O module. It is first deleted when the factory setting is restored, e.g. using STEP7.

Default Device Name and S1 Switch Settings for PP72/48 PN PROFINET operation:

S1/S2 Switch setting										Default device name	
1	2	3	4	5	6	7	8	9	10	PP72/48 PN	MCP
Off	Off	Off	Off	Off	Off	Off	Off	**On**	**On**	pp72x48pn	mcp-pn
On	Off	Off	Off	Off	Off	Off	Off	**On**	**On**	pp72x48pn**1**	mcp-pn**1**
Off	**On**	Off	Off	Off	Off	Off	Off	**On**	**On**	pp72x48pn**2**	mcp-pn**2**
On	**On**	Off	Off	Off	Off	Off	Off	**On**	**On**	pp72x48pn**3**	mcp-pn**3**
Off	Off	**On**	Off	Off	Off	Off	Off	**On**	**On**	pp72x48pn**4**	mcp-pn**4**
On	Off	**On**	Off	Off	Off	Off	Off	**On**	**On**	pp72x48pn**5**	mcp-pn**5**
Off	**On**	**On**	Off	Off	Off	Off	Off	**On**	**On**	pp72x48pn**6**	mcp-pn**6**
On	**On**	**On**	Off	Off	Off	Off	Off	**On**	**On**	pp72x48pn**7**	mcp-pn**7**
Off	Off	Off	**On**	Off	Off	Off	Off	**On**	**On**	pp72x48pn**8**	mcp-pn**8**
On	Off	Off	**On**	Off	Off	Off	Off	**On**	**On**	pp72x48pn**9 (default)**	mcp-pn**9**
"	"	"	"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"	"	"	"
On	**On**	**On**	**On**	**On**	**On**	**On**	Off	**On**	**On**	pp72x48pn**127**	mcp-pn**127**

Below - PLC hardware configuration examples

PP72/48 PN set to the default device name of ***pp72x48pn9.***



MCP PN set to the device name of ***mcp-pn1.***



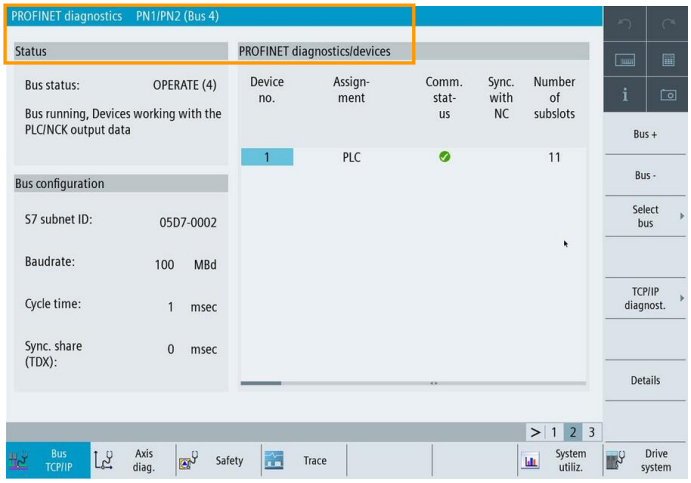
Notes:

PROFINET Network Diagnostics

Integrated PROFIBUS (Servos) information is available on the Diagnostic screens.



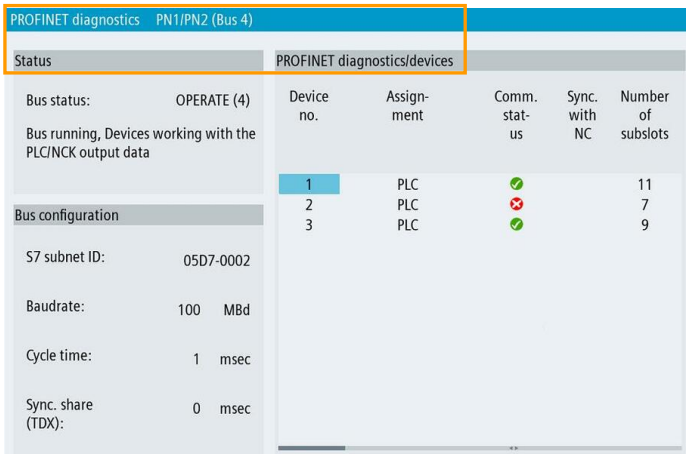
The default screen is the PROFIBUS diagnostic screen.



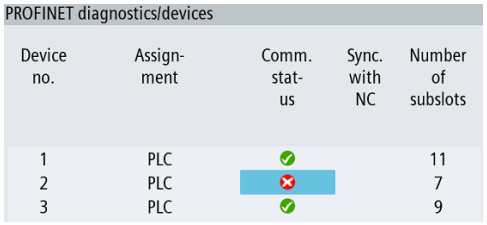
The PROFIBUS diagnostic screen is accessed with the following soft-keys:

Bus + or Bus - or Select bus

Components connected to the PROFINET network - X150 - can be seen:



The ***Comm. status*** indicates whether a component is functioning correctly.

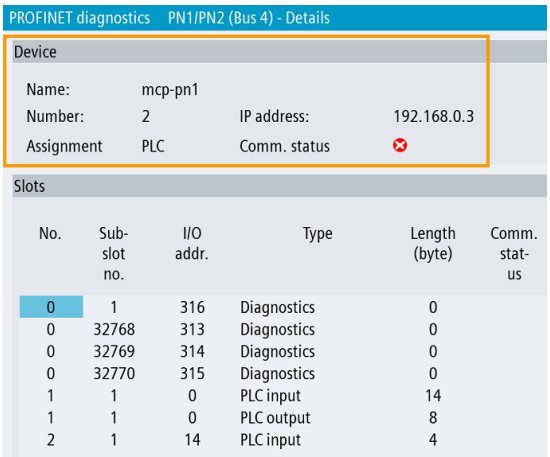


The ***Details*** soft-key shows more details for each individual device.



Device + or Device -

In the following example, the ***mcp-pn1*** device with an IP address of ***192.168.0.3*** has got a problem. In this case the PROFINET cable was not connected.

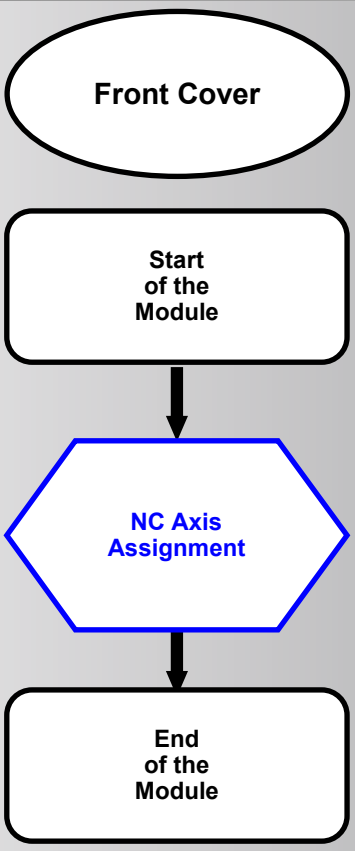


Note: the IP addresses are set by default in the PLC hardware configuration. These can be changed by the machine builder.

Appendix F: END

Notes:

NC - HMI Axis Assignment



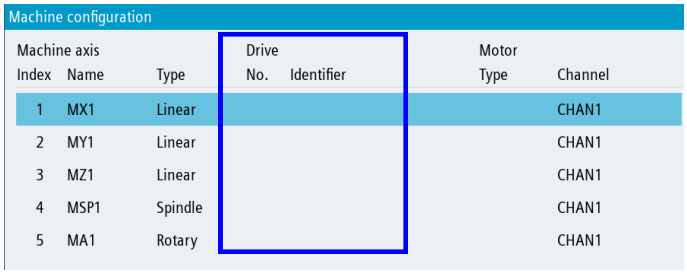
Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

NC Axis Assignment

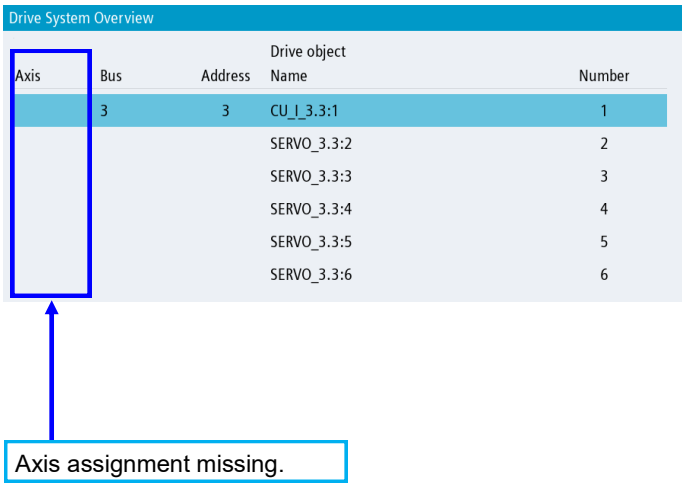
At this point the NC axes and the Sinamics Drive objects are configured but not logically assigned to each other.
The following describes how to make this assignment.

The machine NC configuration shows no drives are assigned:



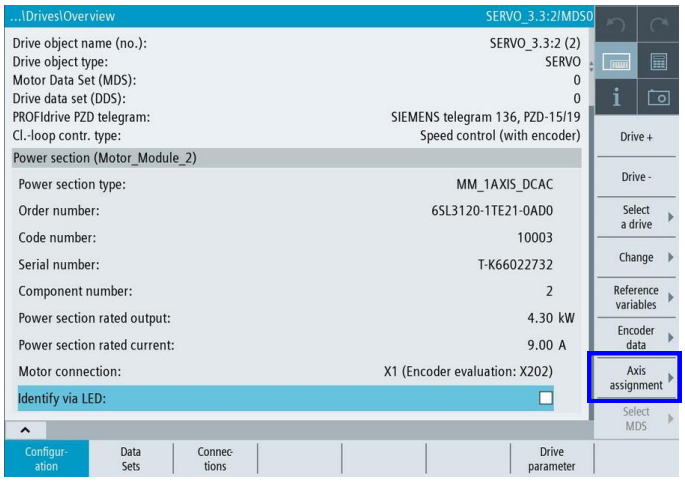
Using the “Drive system” soft-key you can enter the area for assigning the drive objects to the NC Axis.







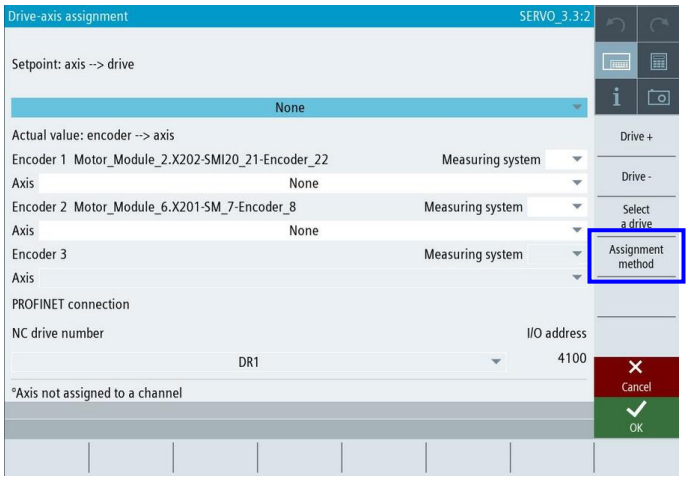
Drive overview screen:

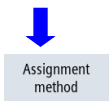


Axis assignment is now achieved with the “Axis assignment” soft-key:

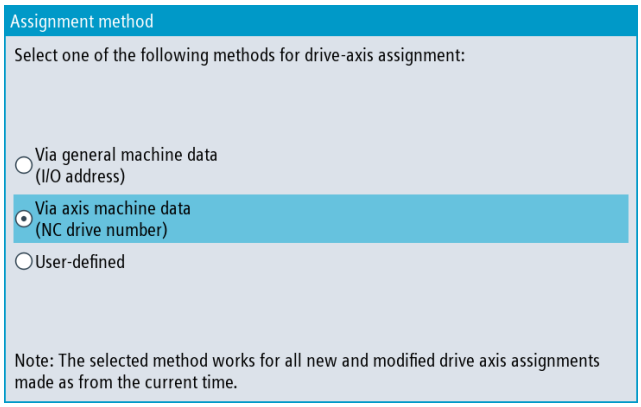


Before you make the assignment of the DRIVE object to the NC Axis you should select the Strategy to be used for the assignment.
Three strategies are available to the machine builder. The strategy should be selected which best suits the build strategy of the builder.





Notes:



It can be seen in the following tables how the assignments are made when selecting either the first or the second strategy from the selection box.
In one case the General Machine data is changed - in the second case the Axis Machine data.
The third strategy is the custom strategy and is intended for use by advanced users only.

Assignment Method 1:
Via General Machine Data - (I/O Address) MD13050[n]

Axis name	Index	**MD - not changed** MD30110 CTRL_OUT_MODULE_NR
MX1	AX1	30110 =1
MY1	AX2	30110 =2
MZ1	AX3	30110 =3
MSP1	AX4	30110 =4
MA1	AX5	30110 =5

MD - changed MD13050 DRIVE_LOGIC_ADDRESS
13050[0] = 4140
13050[1] = 4180
13050[2] = 4220
13050[3] = 4100
13050[4] = 4260

The axis assignment procedure will automatically set various machine data including MD 13050.

Notes:

Assignment Method 2:
Via Axis Machine Data - (Drive number) MD30110

Axis name	Index	**MD changed** MD30110 CTRL_OUT_MODULE_NR	**MD not changed** MD13050 DRIVE_LOGIC_ADDRESS
MX1	AX1	30110 =2	13050[0] = 4100
MY1	AX2	30110 =3	13050[1] = 4140
MZ1	AX3	30110 =4	13050[2] = 4180
MSP1	AX4	30110 =1	13050[3] = 4220
MA1	AX5	30110 =5	13050[4] = 4260

The axis assignment procedure will automatically set various machine data including Axis MD 30110.

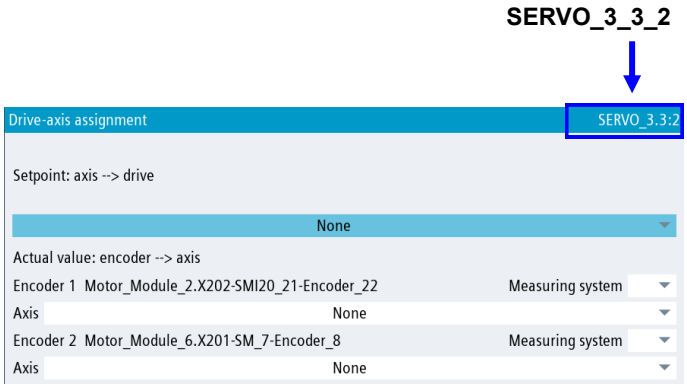
Assignment procedure:

Once the strategy has been decided the axes/servos can be correctly assigned to each other.

Training equipment axis/servo assignments:

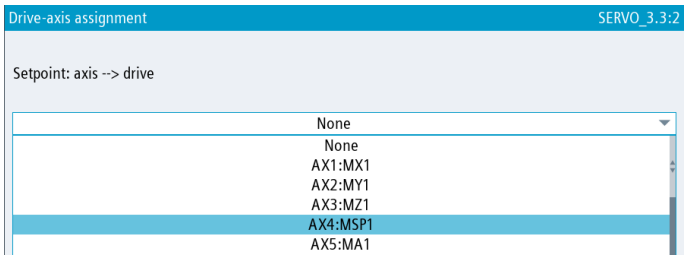
SERVO 3.3.2 MSP1 Spindle
SERVO 3.3.3 MX1 X axis
SERVO 3.3.4 MY1 Y axis
SERVO 3.3.5 MZ1 Z axis
SERVO 3.3.6 MA1 A axis

Check the top right of the screen for the currently selected servo: e.g. **SERVO_3_3_2**

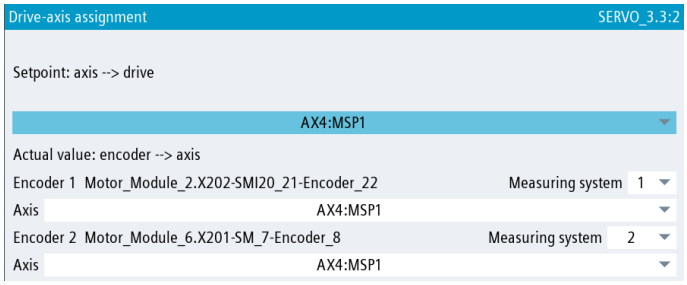


Notes:

From the ***Setpoint: axis - -> drive*** list, select the correct NC axis for the servo.
In this example SERVO 3.3.2 & AX4: MSP1



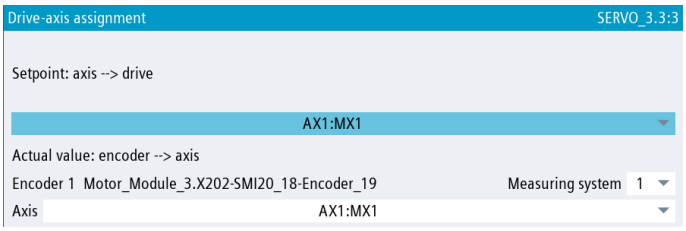
Any encoders associated with the axis will be automatically selected:



Use the ***Drive +*** or ***Drive -*** or ***Select a drive*** soft-keys to locate the next servo to be assigned e.g. SERVO 3.3.3. and assign it to the correct NC axis e.g. MX1.

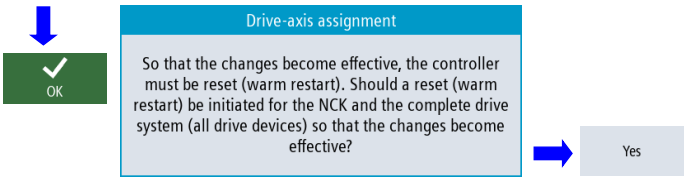


Example: SERVO_3_3_3 assigned to NC axis MX1



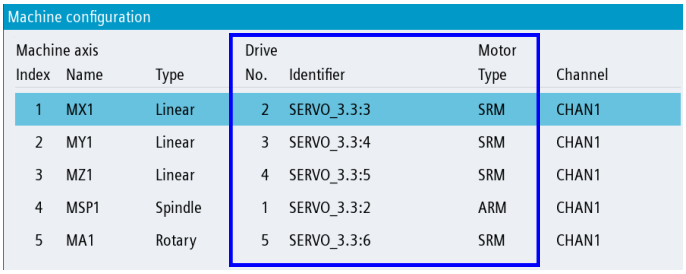
Repeat the process until all Servos have been assigned to the correct NC axis.

Confirm the procedure with the ***OK*** soft-key:



Axis assignment confirmation:

The axis assignments can be seen on the Machine configuration screen:

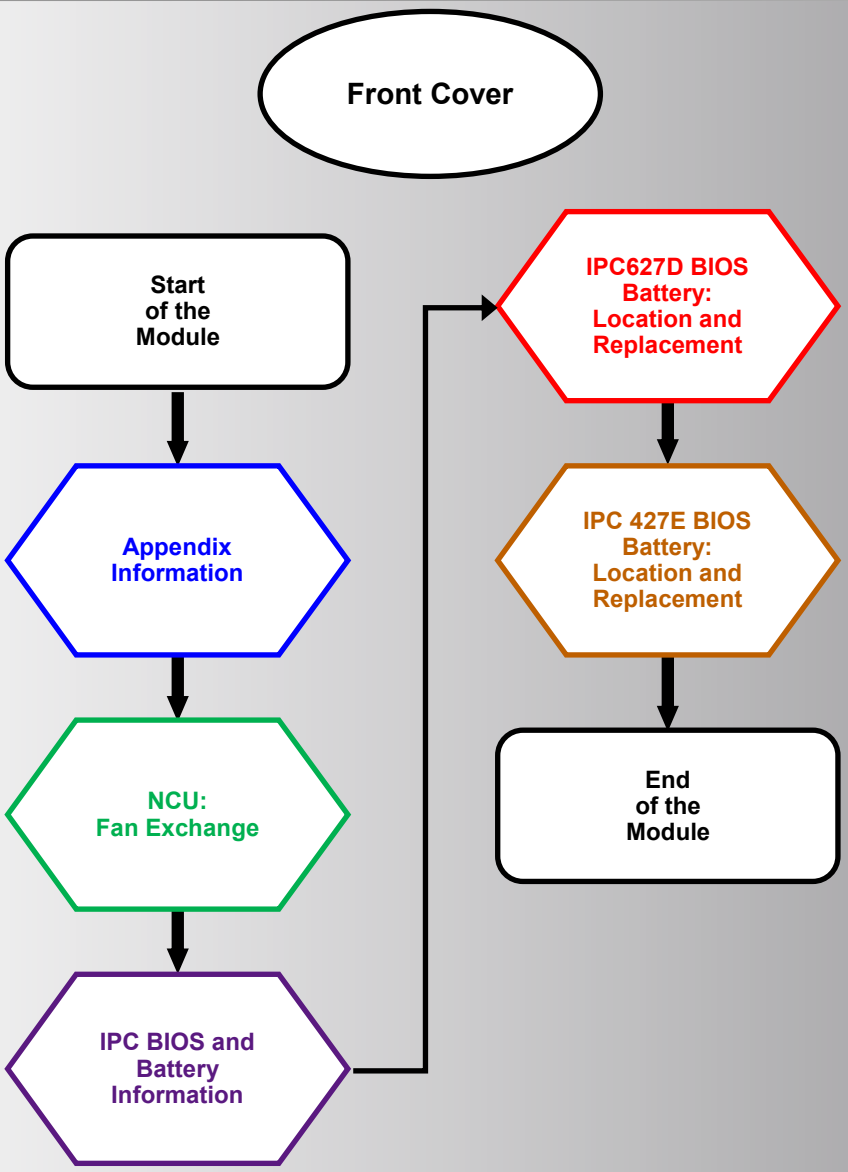


Appendix H: END

Notes:

Notes:

Battery and Fan Exchange



Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

Appendix Information

Certain components are equipped with backup batteries which maintain data stored in the volatile ram area. Some are fitted with fans for the cooling of the electronics.

During the lifespan of the equipment it is possible that the battery will become less efficient, resulting in a drop in voltage or that the cooling fans show signs of wear and tear. In either case the batteries or fans must be replaced.
A data backup should be made before exchanging the battery (this is described in Module A017).

Caution
The Lithium-batteries must be replaced only by identical replacements or by types recommended by the manufacturer. Otherwise component damage may occur.

Notes:

NCU: Fan Exchange

The fans of the Sinumrik ONE NCU are contained in a housing which fits on the underside of the NCU box.



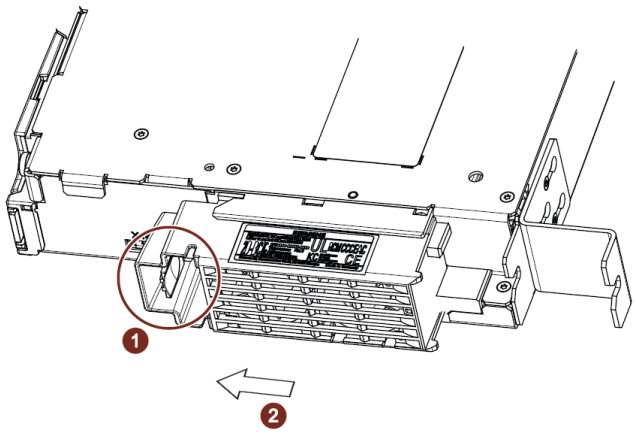
NCU fan location:



Double fan module part number:
6FC5348-0AA30-3AA0

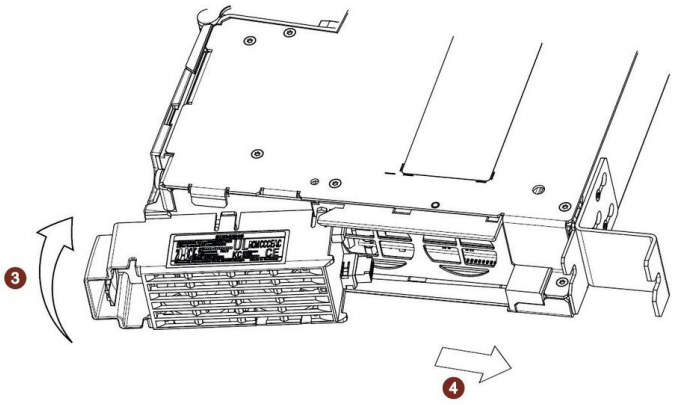
Fan Removal:

The fan module can be removed/replaced in the powered on state. However, the NCU will shut down after 1 minute if the fans are not present.



① Locate press the lug on the front of the fan housing.
② Gently pull the housing forward until it is free of the guideways.

Fan Replacement:



① Present the fan housing at a slight angle and locate the guideways.
② Gently slide the housing forward until it is free of the guideways.

Notes:

IPC BIOS and Battery Information

The battery is used to buffer the BIOS (Basic Input Output Settings) of the IPC627D.

WARNING
Risk of explosion and release of harmful substances!
For this reason, do not burn lithium batteries, do not solder on the cell body, do not open, do not short circuit, do not reverse polarity, do not heat above 100°C, dispose of correctly, and protect against direct sunlight, dampness and dew.
Risk of damage!
The lithium battery may only be replaced with an identical battery or with a type recommended by the manufacturer (Order No.: A5E00331143).
Disposal
Batteries must be disposed of in accordance with local regulations.
Batteries are perishable components. They should be changed every 5 years in order to guarantee the long-term functionality of the device.
If a plug-in PCI card is also supplied from the battery (centralized battery concept), the replacement period is reduced to 4 years.

Information prior to replacement:

- The battery can be exchanged in the powered-on state. This carries some small risks and it is recommended that the power is removed.
- If the battery is replaced within 30 seconds there will be no loss of settings.
- If the exchange takes longer than 30 seconds and the ***User Profile*** BIOS settings are in use, then the BIOS settings will need to be re-entered and the time and date will need to be set.
- If the exchange takes longer than 30 seconds and the ***Standard Profile*** BIOS settings are in use, then only the time and date will need to be set.
- To access the BIOS, press ***Escape,*** on power up, when the Siemens logo is displayed. The settings can be noted, prior to exchanging the battery, if required.

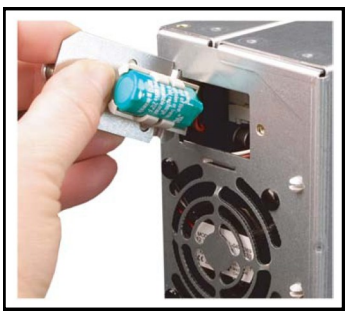
IPC627D BIOS Battery: Location and Replacement



Connector which allows easy exchange of the battery.



The battery is attached to a removable cover.
1) Undo the retaining screw and open the battery cover



2) Unhinge and remove the battery cover.



3) Gently withdraw the cable to reveal the battery connector.

Replacement of the battery is in reverse order.

Notes:

IPC 427E BIOS Battery: Location and Replacement



Battery cover location.

1. Remove the two screws and take off the cover.



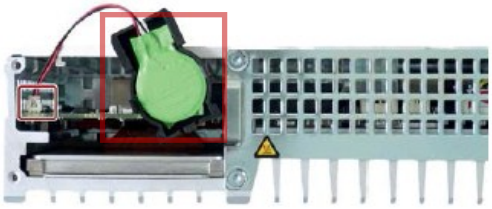
2. Unplug the battery connector.



3. Slide out the battery holder.



4. Remove the battery from the holder.

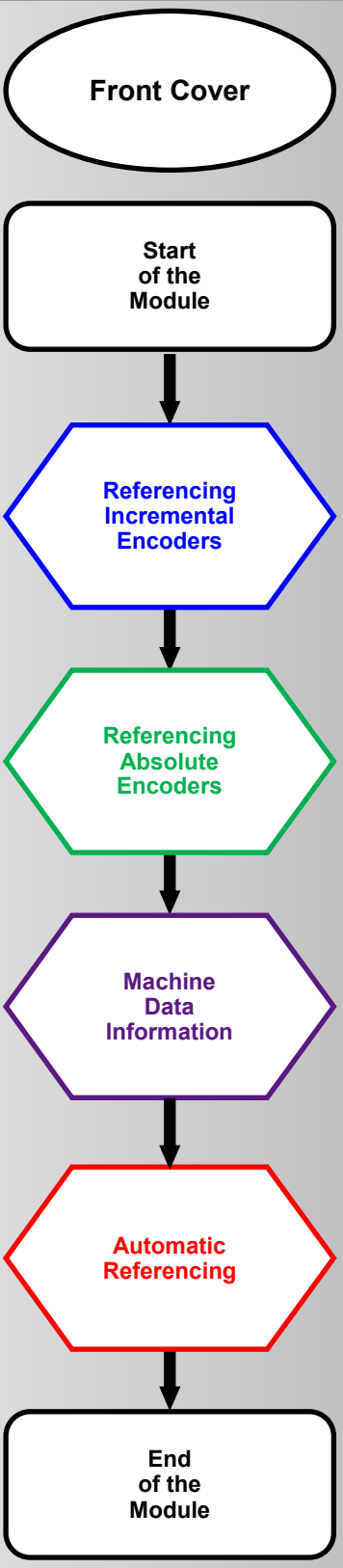


- Replace the battery in reverse order.

Appendix I: END

Notes:

Encoder Adjustment



Sinumerik

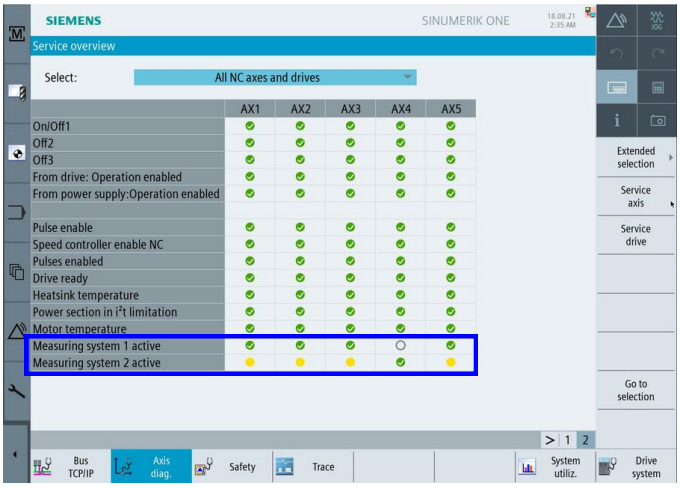
This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

Referencing Incremental Encoders :

General Information

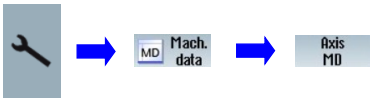
Axes equipped with incremental encoders require referencing each time that the NCK is powered off/on or reset.
This usually entails traversing the axis to a switch which then initiates the referencing procedure. Certain machine data's are available to influence this procedure and adjust it to individual requirements.
Some axes have a second encoder fitted for positioning operations. It needs to be checked which measuring system is active, as that will be the one which requires referencing.
To check which measuring system is active we can look at the “Axis diagnostics” area.



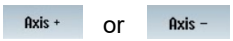


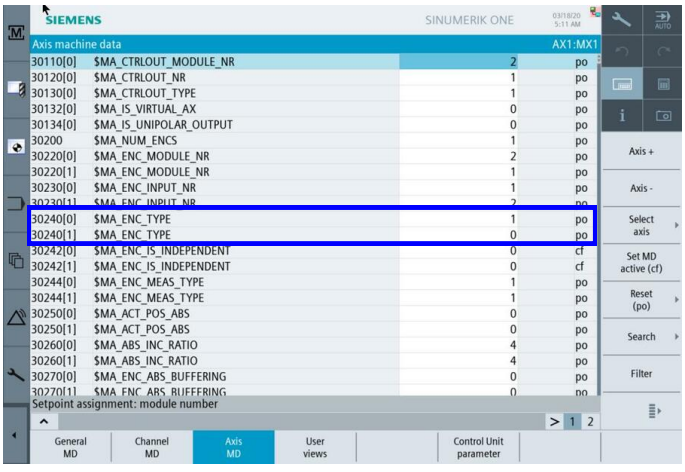
Look for the “Measuring system 1 active” and “Measuring system 2 active”. A green tick symbol indicates which measuring system is active for each axis., typically the motor encoder. A 2nd measuring circuit is typically a linear scale on an axis or a 2nd encoder for a spindle.
Some manufacturers may also switch between measuring systems. If there is uncertainty about this please contact the manufacturer for information.

If it is uncertain whether an axis has an incremental or absolute encoder fitted, this can be checked via axis machine data 30240.
To check the machine data use the following soft key sequence.



Ensure that the correct axis is selected.





Machine data **30240[0]** & **[1]** should be checked.

If the active measuring circuit is 1, machine data **30240[0]** should be used to identify the encoder type.

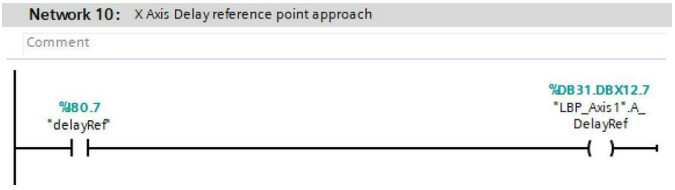
If the active measuring circuit is 2, machine data **30240[1]** should be used to identify the encoder type.

Incremental encoder = 1
Absolute encoder = 4

To reference an axis fitted with an incremental encoder, an interface signal needs to be switched high in the user PLC program. This is usually done with a switch positioned at a suitable place on the axis slide way.
During the reference procedure the axis will traverse towards the switch, on activation of the switch the interface signal will be set.
A sequence of axis moves will then take place which are controlled by the NC and influenced by machine data.
(Further details are shown on the following pages)

Example of an axis reference switch input activating a reference point approach interface bit.

Notes:



Interface signal DB3**x**.DBX12.7 “Delay Ref. Pt approach” needs to be set to high to initiate the referencing procedure. Each axis has its own variable which is identified by changing the **x** for a numerical value which represents the axis, see below-

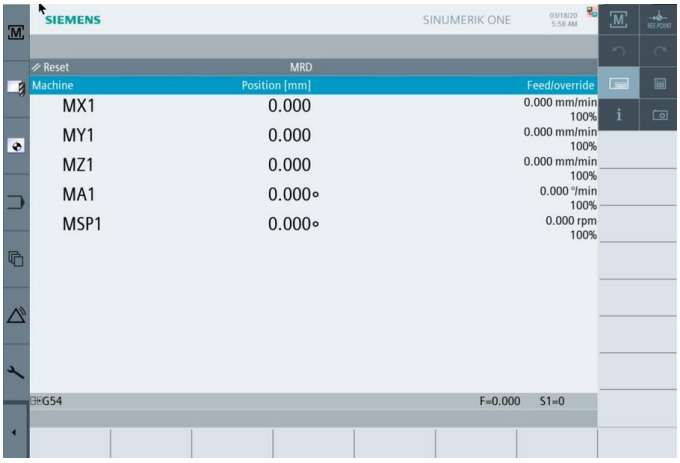
1st axis - DB3**1**.DBX12.7
2nd axis - DB3**2**.DBX12.7
3rd axis - DB3**3**.DBX12.7
etc.

Before referencing can be initiated the controller needs to be placed into Reference Mode by selecting the “REF.POINT” button on the machine control panel. The LED should Illuminate, the “JOG” mode LED remains illuminated.



Confirmation that the correct mode is selected can be seen at the top of the screen.





Each axis can be referenced individually. It is also possible to reference the axes automatically. This is usually set up by the machine builder and will be explained in more detail later in the module.

To reference an individual axis select the correct axis via the machine control panel.

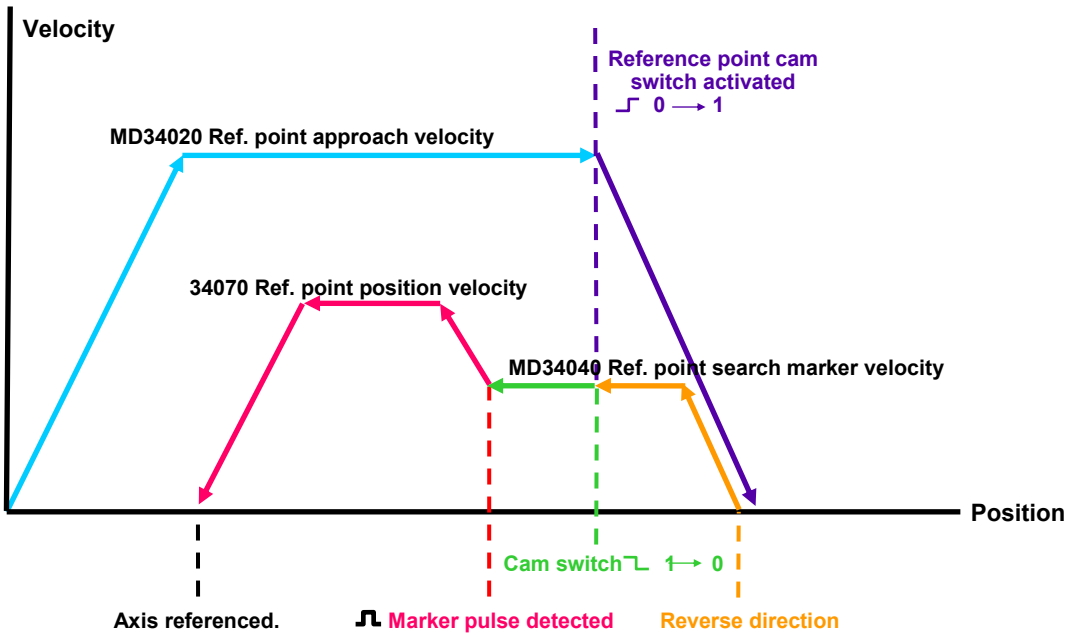


The axis must now be moved in the direction of the reference switch. This can be done manually via the axis jog keys or automatically if automatic referencing is set-up.
The direction of movement is set via machine data.
The axis should move uninterrupted towards the reference switch. Some systems require that the jog direction button is continually depressed throughout the reference cycle, however it is more common to find that the axis will continue to move even when the jog direction button is released.

Notes:

Referencing sequence diagram for incremental encoder

In the event of the values of machine data 34080 & 34090 equalling less than the stopping distance of the axis, the controller will automatically readjust to the correct position.
34040 $MA_REFP_VELO_SEARCH_MARKER = 0
34080 $MA_REFP_MOVE_DIST + 34090 $MA_REFP_MOVE_DIST_CORR = > 0



The control receives a reference axis command, the axis accelerates to the feed-rate in machine data **34020 $MA_REFP_VELO_SEARCH_CAM** then searches for the reference switch input to become high.

The reference switch is operated and the axis decelerates.

The axis reverses direction at the feed-rate set in machine data.
34040 $MA_REFP_VELO_SEARCH_MARKER.

Feed-rate as machine data **34040 $MA_REFP_VELO_SEARCH_MARKER**.
Search for next reference marker

Reference marker is detected. The axis accelerates to the feed-rate set in machine data **34070 $MA_REFP_VELO_POS** to a position which is the sum of machine data **34080 $MA_REFP_MOVE_DIST** and **34090 $MA_REFP_MOVE_DIST_CORR**

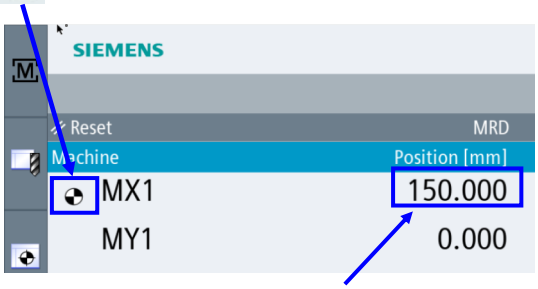
Notes:

When an axis is referenced the controller issues interface signals:

1st axis DB31.DBX60.4 1st measuring circuit is referenced
1st axis DB31.DBX60.5 2nd measuring circuit is referenced
2nd axis DB32.DBX60.4 1st measuring circuit is referenced
2nd axis DB32.DBX60.5 2nd measuring circuit is referenced
etc.

The position display will indicate the axis is referenced and display any values stored in machine data:
34100 $MA_REFP_SET_POS

Symbol indicating the axis is referenced.



Value from Axis MD 34100

Referencing Absolute Encoders :

General information
Absolute encoders need referencing on first commissioning and remain referenced until a problem occurs with the encoder or the NCK. Re loading of an archive will also necessitate the referencing of the encoder.
Procedure:
First it must be ascertained that the encoder is of the absolute type. This can be done by checking the motor data rating plate (In the case of an integrated motor encoder)
Checking with the supplier of the encoder is another method of determining the correct type.

Machine data **30240[0]** & **[1]** can also be checked.

If the active measuring circuit is number 1, then the value of machine data **30240[0]** should be used to identify the encoder type.

If the active measuring circuit is 2, machine data **30240[1]** should be used to identify the encoder type.

The following values determine the encoder type:

Incremental encoder = 1
Absolute encoder = 4

Axis machine data			AX5:MA
30110[0]	$MA_CTRLOUT_MODULE_NR	5	po
30120[0]	$MA_CTRLOUT_NR	1	po
30130[0]	$MA_CTRLOUT_TYPE	1	po
30132[0]	$MA_IS_VIRTUAL_AX	0	po
30134[0]	$MA_IS_UNIPOLAR_OUTPUT	0	po
30200	$MA_NUM_ENCS	1	po
30220[0]	$MA_ENC_MODULE_NR	5	po
30220[1]	$MA_ENC_MODULE_NR	5	po
30230[0]	$MA_ENC_INPUT_NR	1	po
30230[1]	$MA_ENC_INPUT_NR	2	po
30240[0]	$MA_ENC_TYPE	4	po
30240[1]	$MA_ENC_TYPE	0	po
30242[0]	$MA_ENC_IS_INDEPENDENT	0	cf
30242[1]	$MA_ENC_IS_INDEPENDENT	0	cf
30244[0]	$MA_ENC_MEAS_TYPE	1	po
30244[1]	$MA_ENC_MEAS_TYPE	1	po

The procedure requires that the axis be moved to a known mechanical position e.g. 150 degrees. Locate machine data **34100 $MA_REFP_SET_POS** and enter 150.

34093[1]	$MA_REFP_CAM_MARKER_DIST	0 °	po
34100[0]	$MA_REFP_SET_POS	150 °	re
34100[1]	$MA_REFP_SET_POS	0 °	re
34100[2]	$MA_REFP_SET_POS	0 °	re

Check that machine data **34200 ENC_REFP_MODE** is set to **0** This enables the control to take over the value in machine data **34100 $MA_REFP_SET_POS.**
Check that machine data **34210 ENC_REFP_STATE** is set to **1**. This is to enable encoder calibration to take place.

34110	$MA_REFP_CYCLE_NR	1	re
34200[0]	$MA_ENC_REFP_MODE	0	po
34200[1]	$MA_ENC_REFP_MODE	1	po
34210[0]	$MA_ENC_REFP_STATE	1	im
34210[1]	$MA_ENC_REFP_STATE	0	im

Check whether machine data
34010 REFP_CAM_DIR_IS_MINUS is set to **0** OR **1**.

33120	$MA_PATH_TRANS_POS_TOL	0.005 °	cf
34000	$MA_REFP_CAM_IS_ACTIVE	0	re
34010	$MA_REFP_CAM_DIR_IS_MINUS	0	re
34020	$MA_REFP_VELO_SEARCH_CAM	30 rpm	re
34030	$MA_REFP_MAX_CAM_DIST	720 °	re

Notes:

The axis to be referenced should be moved to the known mechanical position using “Jog” mode in the direction determined by the value of machine data:
34010 REFP_CAM_DIR_IS_MINUS.

A value of 0 and the approach will be in the positive direction, a value of 1 will make the approach negative.
The feed rate when approaching this position should be reduced to avoid inaccuracies caused by backlash.

Before referencing can be initiated the controller needs to be placed into Reference Mode by selecting the “REF.POINT” button on the machine control panel. The LED should Illuminate, the “JOG” mode LED remains illuminated.



Confirmation that the correct mode is selected can be seen at the top of the screen.





The A axis of the training equipment is fitted with an absolute encoder and therefore the following procedure can be used to reference it.



Select the A axis and then press the same direction key used to position the axis earlier.
The axis will not physically move but the position display will change to show the axis is referenced symbol and the reference point set position will be shown as the position of the axis (150 mm).
Machine data **34210** will have changed to **2** to indicate a successful procedure.



34200[0]	$MA_ENC_REFP_MODE	0	po
34200[1]	$MA_ENC_REFP_MODE	1	po
34210[0]	$MA_ENC_REFP_STATE	2	im
34210[1]	$MA_ENC_REFP_STATE	0	im
34220[0]	$MA_ENC_ABS_TURNS_MODULO	4096	po

Notes:

Machine Data Information :

If a machine does not have reference switch cam it is possible to reference from the zero mark by changing machine data-
34000 $MA_REFP_CAM_IS_ACTIVE (Default value 1)
A value of 0 will deactivate the cam.
This is used for example on axes that have only one reference mark along the entire traversing range and rotary axes that have only one reference mark per revolution.

The direction that the axis approaches the reference switch can be changed by machine data-
34010 $MA_REFP_CAM_DIR_IS_MINUS (Default value 0)
The default value will result in the reference point approach in a positive direction. To reverse the direction a value of 1 has to be entered.

The initial approach speed for referencing is determined by machine data- **34020 $MA_REFP_VELO_SEARCH_CAM** (Default value 5000.0 mm/min)

If the axis fails to reach the reference cam it is probably due to machine data-**34030 $MA_REFP_MAX_CAM_DIST** (Default value 10000 mm) This is the maximum distance to the reference cam.
The controller will issue alarm:-20000 Channel %1 axis %2 reference cam not reached.

When the reference switch is activated the axis reverses direction in search of the reference marker. This search speed is determined by machine data- **34040 $MA_REFP_VELO_SEARCH_MARKER** (Default value 300.0 mm/min).

The direction for the reference marker search is changed by machine data-
34050 $MA_REFP_SEARCH_MARKER_REVERSE (Default value 0)
The default value causes the axis to search for the reference cam in the opposite direction to the direction specified in machine data **34010 $MA_REFP_CAM_DIR_IS_MINUS**.
Entering a value of 1 in machine data 34050 will reverse the cam search direction.

If the axis fails to reach the reference marker it is probably due to machine data-
34060 $MA_REFP_MAX_MARKER_DIST (Default value 20 mm) This is the maximum distance the axis can travel without finding the marker..
The controller will issue alarm:-20002 Channel %1 axis %2 zero mark missing.

When the marker is detected the axis will either stop or move to a position determined by the values of two machine data (34080 & 34090 see below for further explanation)
The speed at which this position sought is determined by machine data-
34070 $MA_REFP_VELO_POS (Default value 10000.0 mm/min).

34080 $MA_REFP_MOVE_DIST (Default value -2.0 mm).
Once the marker has been detected it is advisable to move the axis away from the marker (Into the machining area). This is achieved by putting a value in machine data 34080.
If the axis was asked to reference again while still on the marker this could result in the axis going toward the end of the slide.

34090 $MA_REFP_MOVE_DIST_CORR (Default value 0.0 mm) This machine data is used to accurately set the position and correct discrepancies caused by an encoder exchange or mechanical work.
The values in machine data 34080 & 34090 are added together to determine the distance the axis moves.

Usually the position display shows zero after an axis has been referenced. This can be changed by machine data
34100 $MA_REFP_SET_POS (Default value 0.0 mm)
The value entered in here will be displayed on the screen as soon as referencing is complete.
This is used when the reference position is not the mechanical zero of the machine. The machine data does not result in the axis physically moving.

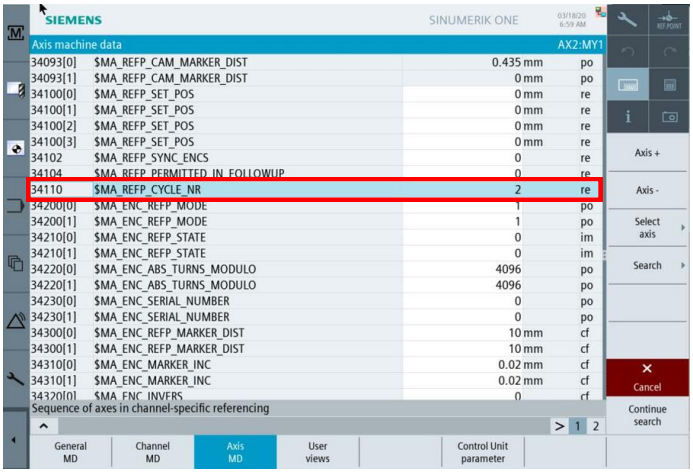
Notes:

Automatic Referencing :

Automatic referencing cycle
To set the automatic referencing function it has to be decided in what order you want the axes to reference. It is common for example that on a milling machine the Z axis needs to be referenced first, this helps eliminate the chance of a collision.
The order is set by axis machine data
$MA_REFP_CYCLE_NR (Default 1)
Possible entries:-
0 = no referencing cycle participation.
1 to 15 = allowable range for setting the sequence.
-1 = no reference required for Cycle start.

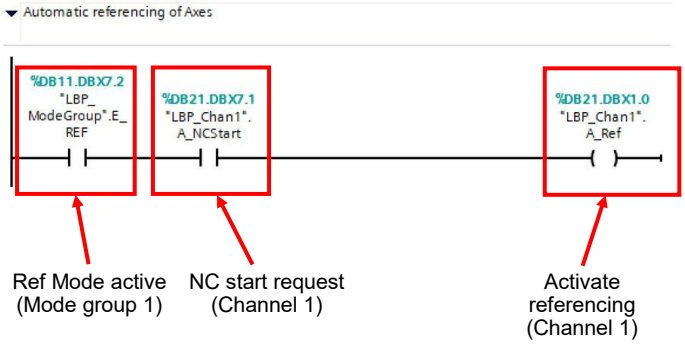
Several axes can be referenced at the same time.

Example: The Y axis is set as 2nd in the referencing sequence.



To initiate the automatic cycle the necessary interface bit (DB21.DBX1.0) has to be set.

Example of how the automatic referencing cycle can be initiated:



Interface signal DB21.DBX33.0 changes to 1 while the referencing cycle is active.

Appendix J: END

Notes:

A201

SINUMERIK ONE

Preparation of system software for NCU on the PPU

Module Description:

Before the commissioning of a machine can take place, the NCU on the PPU needs to be in a running state.
This module discribes the nessecery steps to install the system softwares on this item of hardware.

Module objective:

Upon completion of this module you will be able to perform the following task:

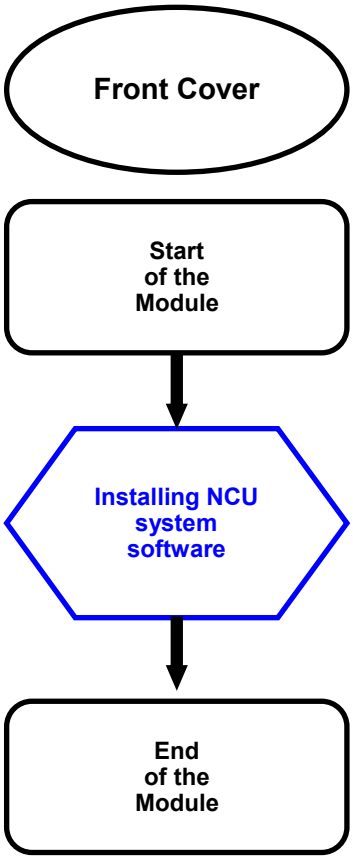
- Install the system softwares on the NCU.

Content:

Installing NCU system software

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

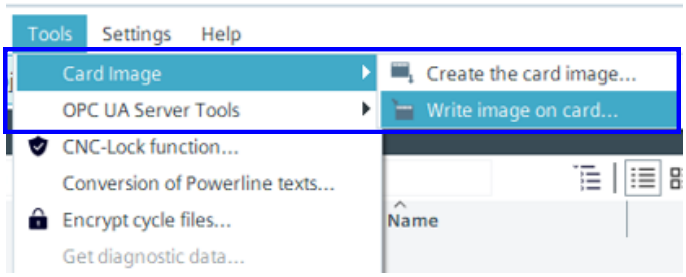

Front Cover
Start of the Module
Installing NCU system software
End of the Module

Installing NCU system software

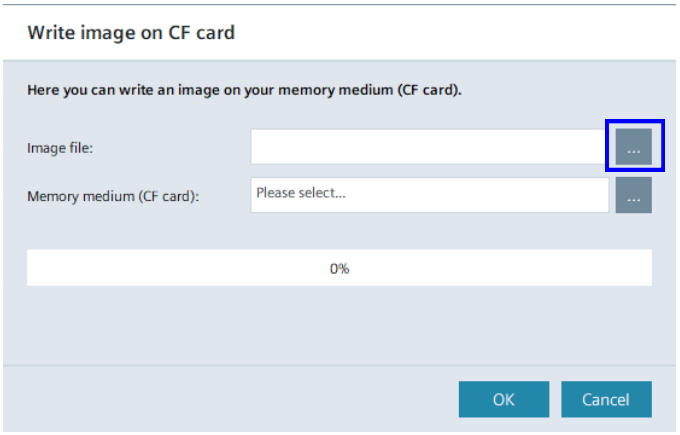
Installing the System software onto SD card is carried out by using the “Service System” software.
A USB service stick is required. This is a bootable stick which contains Service System software.
When plugged into interface X125 the system will boot from the USB device into the “Service” stick system menu.
From this menu it is possible to install the CNC software file.
.

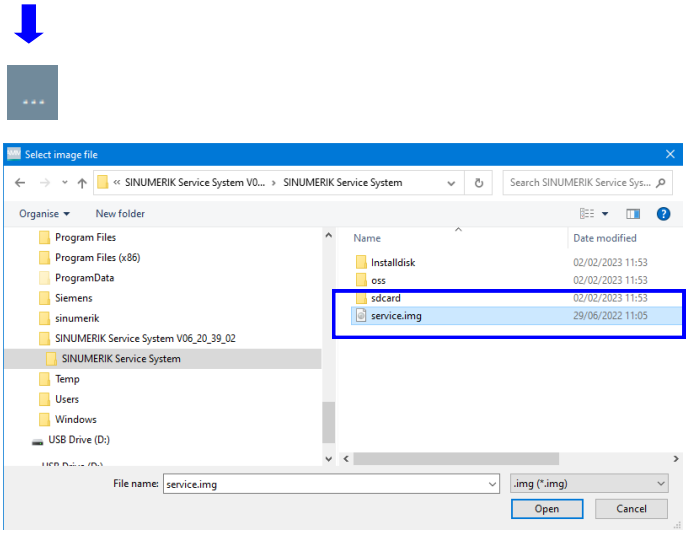
Creating a USB service stick.
Access MyMachine is used for creating a USB service stick.
The software is delivered in “zip” format and should be extracted to a suitable place on a computer.
When extracted, the file “service.img” can be found in the following path:
SINUMERIK Service System NCU V06_60_00_00/
SINUMERIK Service System/service.img

Run the Access MyMachine and select “Writing image to CF card” found on the “Tools” menu.



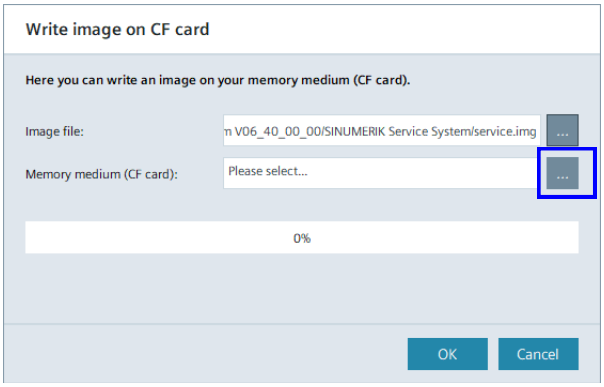
Locate the “service.img” file and select “Open”



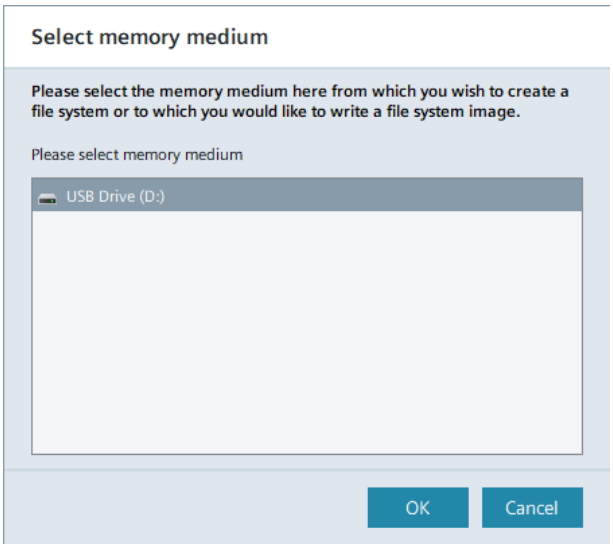




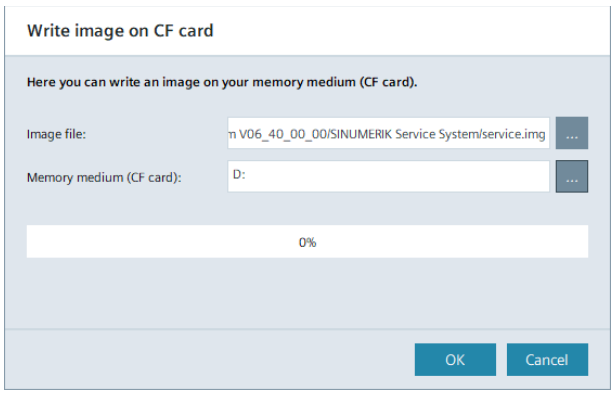
Select the drive letter of the USB stick.



Notes:

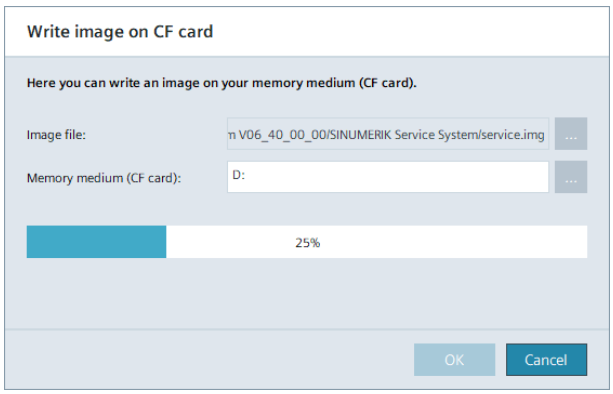


OK

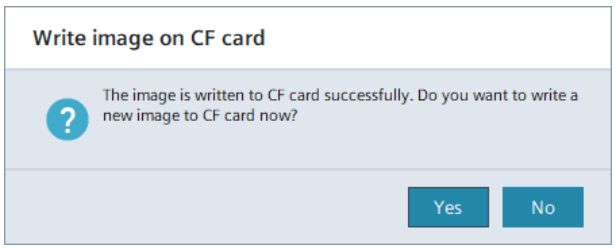


OK

The writing process will begin.



The process can be repeated if required.

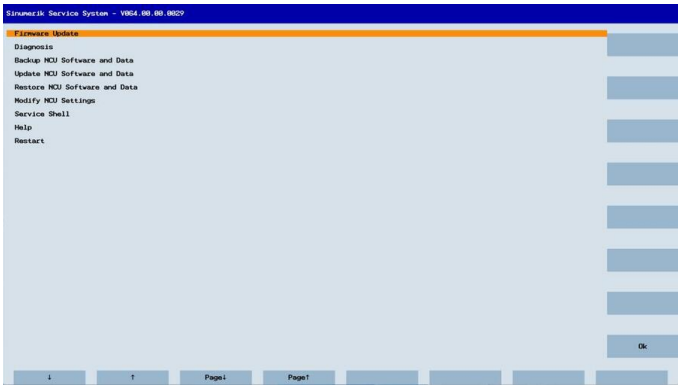


No

Notes:

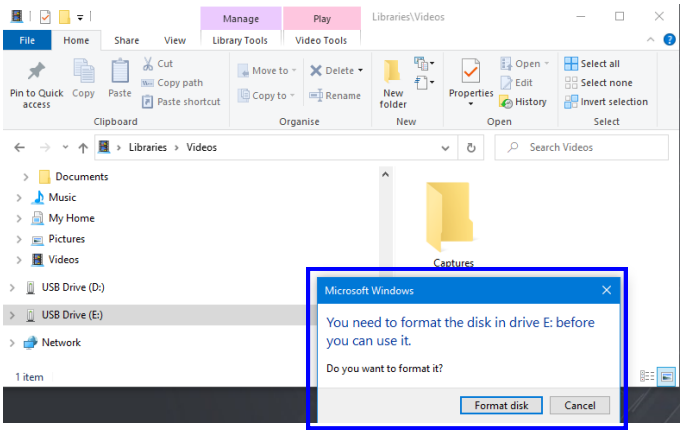
Starting the Service System.

Switch off the controller and insert the bootable USB stick into X125 of the NCU, and place the NCK rotary switch in any unused rotary position eg. F
Power on the controller
The following screen will be displayed:

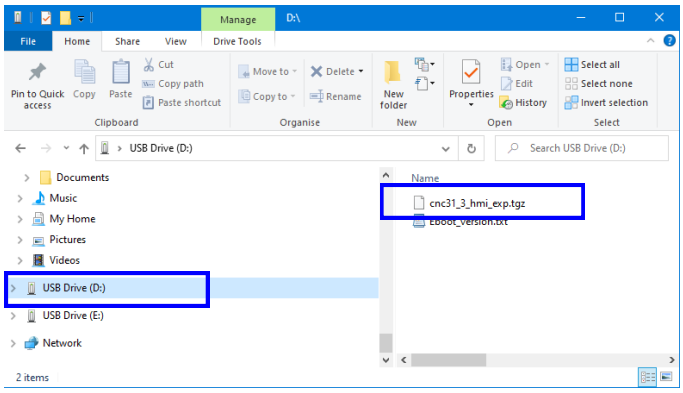


By powering up the boot stick in the NCU, the correct partitions are created .
Power OFF the NCU box, and view the boot stick in Windows Explorer.

The boot stick in Windows Explorer, will show two drives. Drive E:/ is the Linux boot partition and MUST NOT be formatted.



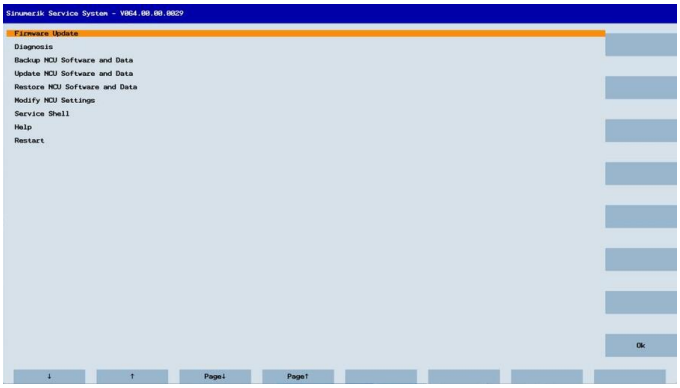
Drive D:/ is the partition where the system software *.tgz file must be transferred to.



Installing the system software (TGZ file) to the SD card of the NCU

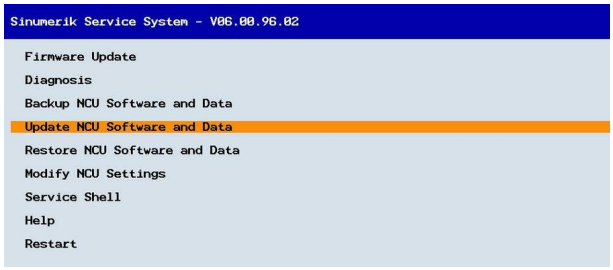
Starting the Service System.

Switch off the controller and insert the bootable USB stick into X125 of the NCU, and place the NCK rotary switch in any unused rotary position eg. F
Power on the controller
The following screen will be displayed:



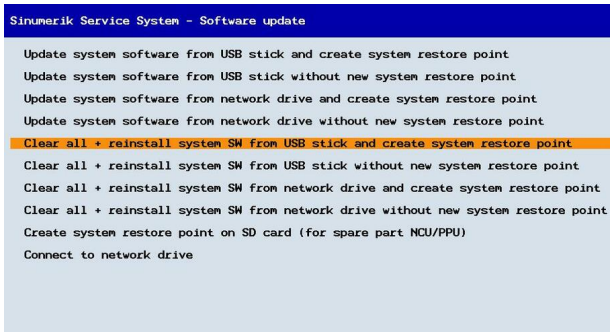
Cursor to the “Update NCU Software and Data” menu item and select “Ok”

Notes:



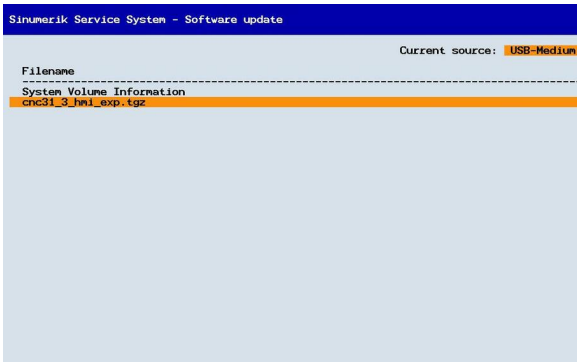
OK

A number of “reinstall” choices are available, select the menu item “Clear all + reinstall system SW from USB stick and create system restore point”



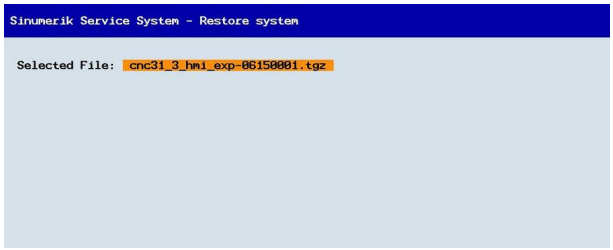
OK

Select the system software file, in this case “cnc31_3_hmi_exp.tgz”.

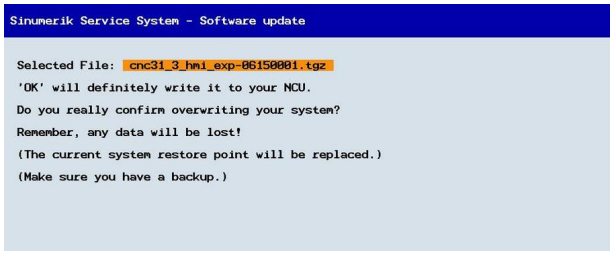


OK

The procedure will begin.

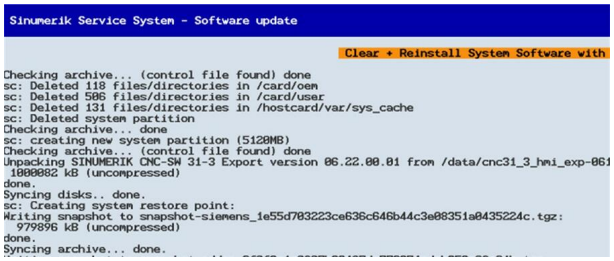


OK

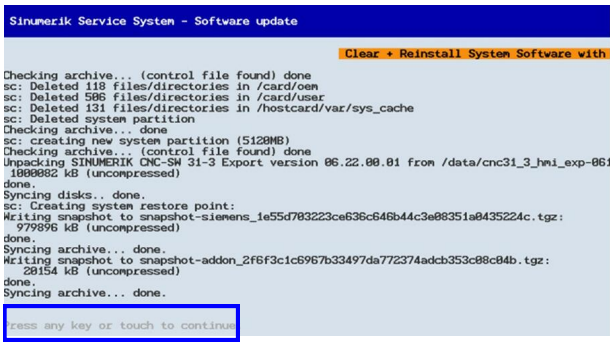


OK

The Installation procedure will begin.

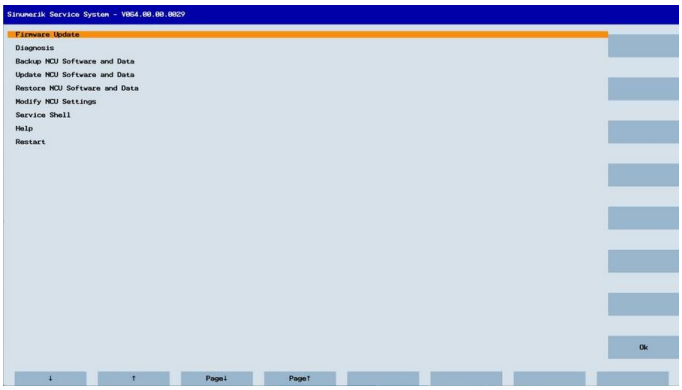


On completion, press any soft-key.

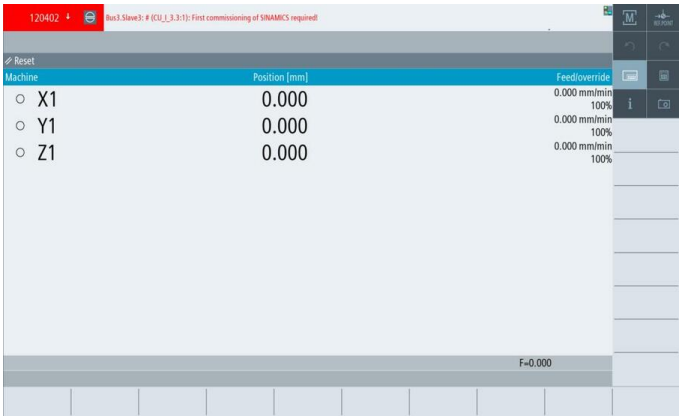


Notes:

The original screen will be displayed.
The USB stick should be removed and the system rebooted.



Switch off the controller and remove the bootable USB stick from X125 of the NCU, and place the NC rotary switch into position 0, switch on the NCU.



The system software running from the NCU,
Before commissioning can take place:

Standard machine data needs to be loaded.
PLC area needs to be cleared.

Set the password.

Set the time and date.

A201: END

Notes:

301

SINUMERIK

Additional Languages

Module Description:

This module explains the installation and switching to the new language.

The 828D software can be used in many different languages.
Nine languages are installed as standard (English, French, German, Italian, Korean, Portuguese, Simplified Chinese, Spanish, and Traditional Chinese).

The Sinumerik ONE software can be used in many different languages.
Six languages are installed as standard (English, French, German, Italian, Simplified Chinese, and Traditional Chinese).

An additional 20/23 languages are available.
This module explains the installation of the additional languages.

Module objective:

Upon completion of this module you will know how to install additional languages.

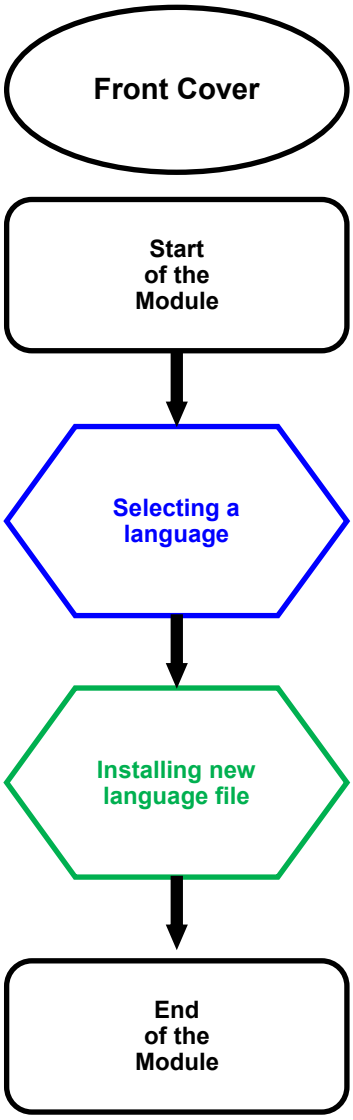
Content:

Selecting a language

Installing new language file

Sinumerik

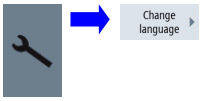
This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

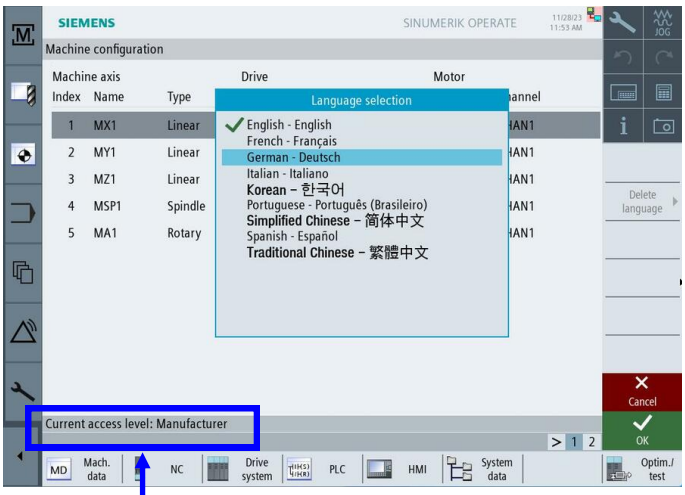

Front Cover
Start of the Module
Selecting a language
Installing new language file
End of the Module

Selecting a language

The SINUMERIK Operate has nine installed languages as standard:

English
French
German
Italian
Korean (828D)
Portuguese (828D)
Simplified Chinese
Spanish (828D)
Traditional Chinese





Current access level: Manufacturer

Access level must be set to 2 (Service) or higher to make changes of the languages.
Cursor over the required language and press the OK soft-key.



Installing new language file

The 20 additional languages are:
Bulgarian
Danish
Finnish
Greek
Hinidi
Indonesian
Japanese
Croatian
Malayan
Dutch
Polish
Romanian
Russian
Swedish
Slovak
Slovenian
Tamil
Turkish
Hungarian
Vietnamese

These additional languages can be used with the 828D SW 4.8 (or later) and SINUMERIK ONE SW 6.11 (or later) .
They can be installed additionally to the nine standard languages already provided. The number of installable additional languages on an NCU is limited only by the available memory space on the CF/SD card of the PPU and IPC.
The additional languages are now available in the archive format and read in via the SINUMERIK Operate HMI.

In this example Turkish language will be used.

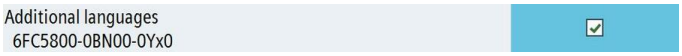
Notes:

The additional languages are available on the PPU hardware, but on an IPC hardware the option "Additional languages" must been set.

Activating the "Additional Languages" option:

The option, **6FC5800-0AN00-0YB0**, needs to be set.





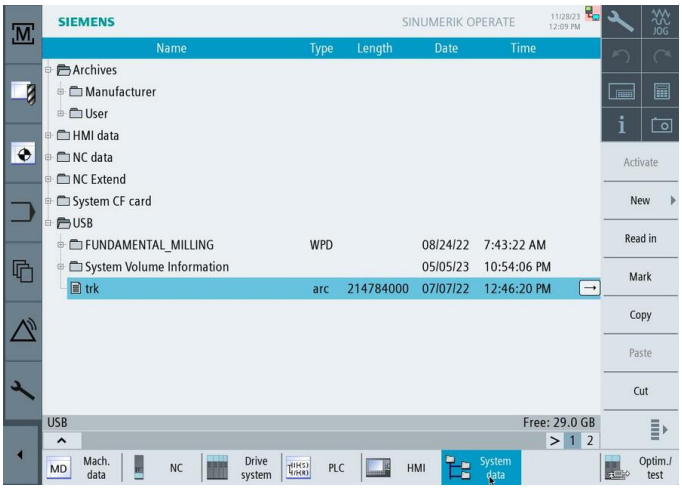
Activate the option with an NCK Reset.



Copy the additional language archive, e.g. trk.arc, onto a USB stick.
On the SINUMERIK Operate, select the setup area and set the service or manufacturer password (protection level 2 or 1).

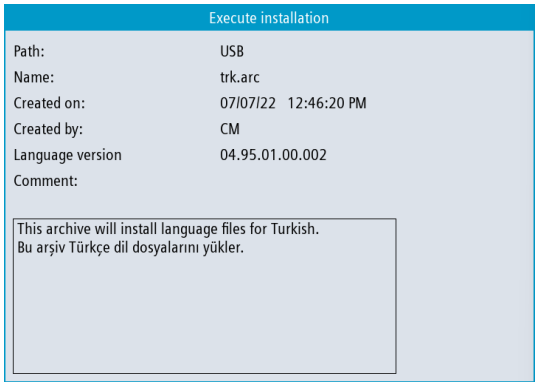
Archives can be read directly from USB.
Cursor to/highlight the rus.arc file and select the Read in soft-key





Cursor over the rus.arc file and press the Read in soft-key.

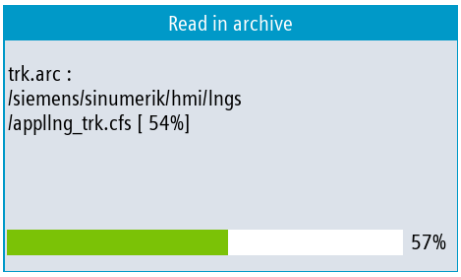




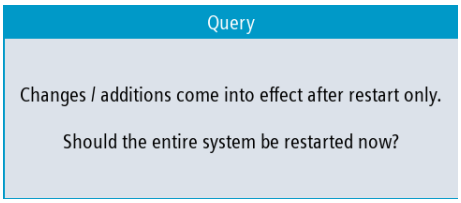
To continue, press the OK soft-key.



Notes:



When the archive has been read in a window will appear.

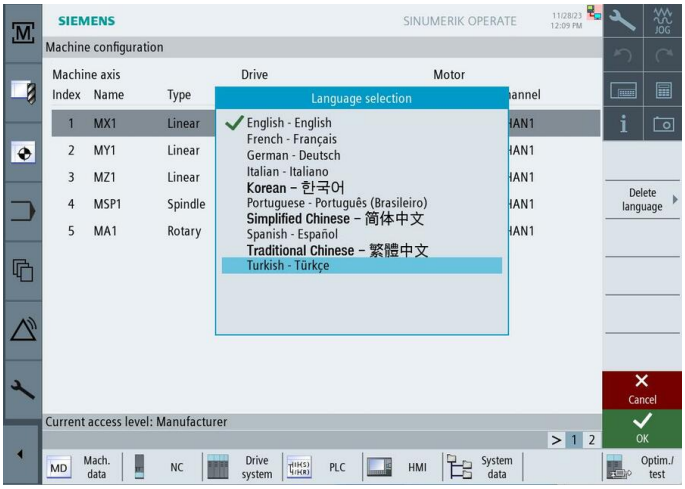


If another language is to be added, press the No soft-key and repeat the procedure.
If only one language is to be installed in this example therefore press the Yes soft-key.

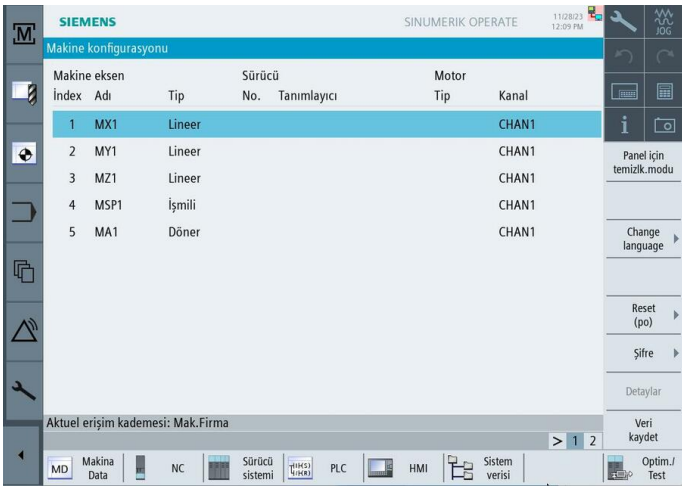


When the control powers up, the “Additional Language" will be available for selection

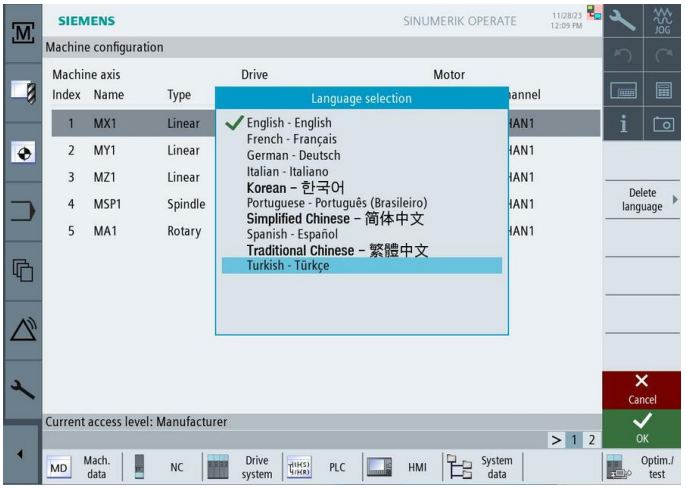






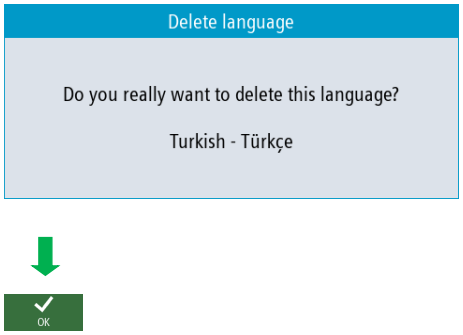


Additional languages can be deleted using the "Delete language" soft-key on the Language selection screen.





Notes:



A complete power off / On of the 828D controller is required after removing the language files.

301: END

Notes:

301: END

Notes:

A303

SINUMERIK ONE

Global User Data and Macros

Module Description:

This module explains the use of Global user data (GUDs) and Macros in the Sinumerik ONE Controller

Global user data (GUDs) are user variables that are defined exclusively by the user using definition files or via definition in programs.

A macro is a sequence of individual statements which have together been assigned a name of their own

Module Objective:

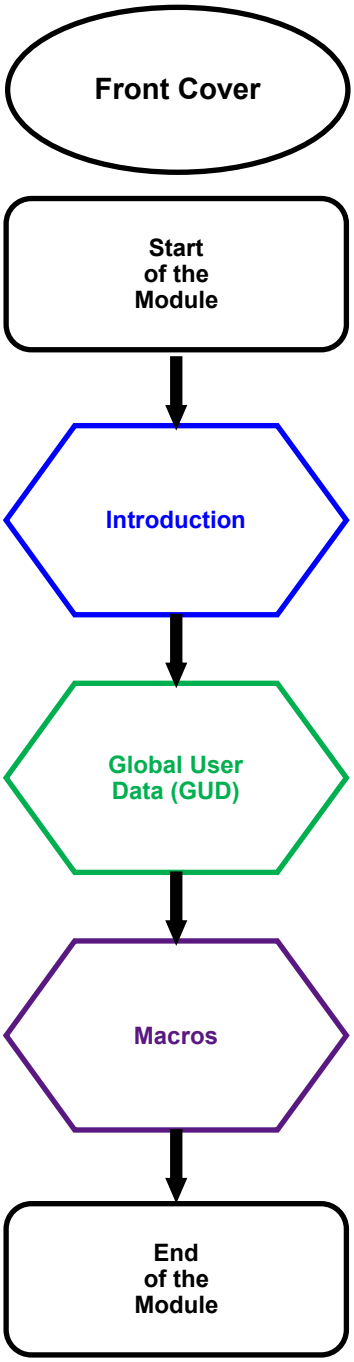
At the end of this module you should be able to create global user data and macros using definition files.

Content:

Introduction
GUDs
Defining/Redefining/Creating/Activating/Displaying/Editing
Reading/Writing from PLC
Macros
Defining/Creating/Activating/Editing
NC MD to call cycles

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Introduction
Global User Data (GUD)
Macros
End of the Module

Introduction

User variables GUDs, LUDs, PUDs

User variables are variables whose meaning is not known to the system and they are not evaluated by the system. The meaning is defined exclusively by the user using definition files or via definition in programs.

Predefined user variables GUDs (Global user data)

Predefined user variables are variables which have already been defined in the system using definition files and whose value simply has to be parameterized by the user. The user can make changes to the properties of these variables.

User-defined variables LUDs, PUDs (Local user data, Program user data)

User-defined variables are variables which are defined exclusively by the user and are not created by the system until runtime. Their number, data type, visibility, and all other properties are defined exclusively by the user and are only valid in the program or subprogram where they were defined.

Macros

A macro is a sequence of individual statements which have together been assigned a name of their own. G, M and H functions or L subprogram names can also be used as macros. When a macro is called during a program run, the statements programmed under the program name are executed one after the other.

Definition files

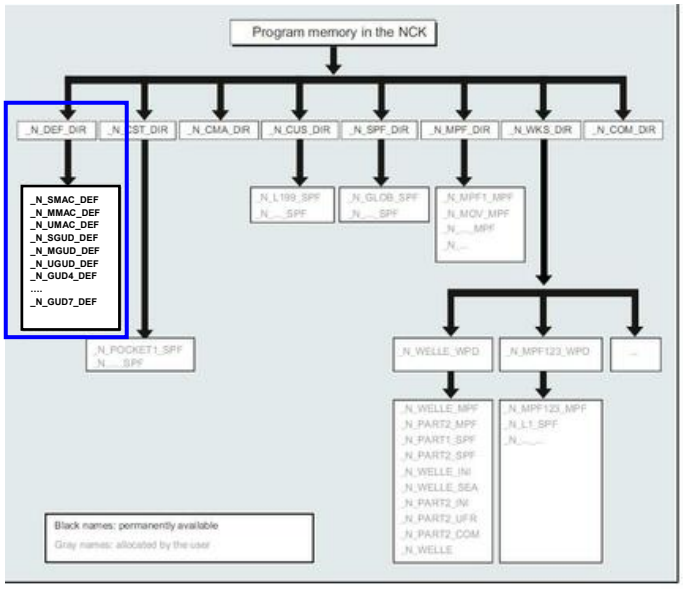
The following definition files are available for defining GUDs and Macros.

MGUD.DEF Definition of Manufacturer GUDs
UGUD.DEF Definition of User GUDs
GUD4.DEF Definition of additional User GUDs
To
GUD7.DEF Definition of additional User GUDs
MMAC.DEF Definition of Manufacturer Macros
UMAC.DEF Definition of User Macros

SGUD.DEF & SMAC.DEF are reserved for Siemens functions

Program memory in the NCK for Definition files

Definition Files are saved in the non-volatile program memory (→ passive file system).



Global user variables (GUD) machine data

In the context of the definition of global user variables (GUDs), the following machine data has to be taken into account:

MD No	Identifier $MN_	Meaning
11140	GUD_AREA_SAVE_TAB	Additional save for GUD blocks
18118	MM_NUM_GUD_MODULES	Number of GUD files in the active file system (SRAM)
18120	MM_NUM_GUD_NAMES_NCK	Number of global GUD names (SRAM)
18130	MM_NUM_GUD_NAMES_CHAN	Number of channel-specific GUD names (SRAM)
18150	MM_GUD_VALUES_MEM	Memory location for global GUD values (SRAM)
18660	MM_NUM_SYNACT_GUD_REAL	Number of configurable GUD of the REAL data type
18661	MM_NUM_SYNACT_GUD_INT	Number of configurable GUD of the INT data type
18662	MM_NUM_SYNACT_GUD_BOOL	Number of configurable GUD of the BOOL data type
18663	MM_NUM_SYNACT_GUD_AXIS	Number of configurable GUD of the AXIS data type
18664	MM_NUM_SYNACT_GUD_CHAR	Number of configurable GUD of the CHAR data type
18665	MM_NUM_SYNACT_GUD_STRING	Number of configurable GUD of the STRING data type

Notes:

User variables GUDs, LUDs, PUDs

Definition of user variables (DEF)

The DEF command defines user-specific variables and assigns values to them. To set them apart from system variables, these are called user-defined variables or user variables (user data).
According to the range of validity (in other words, the range in which the variable is visible) there are the following categories of user variable:

Global user variables (GUD)

Global user variables (GUD) are NC or Channel-global variables which are defined in a data block (MGUD, UGUD, GUD4 to GUD7) and are retained following shutdown and restart. GUD can be accessed in all part programs.

File Name	Meaning
MGUD.DEF	Definitions for global manufacturer data
UGUD.DEF	Definitions for global user data
GUD4.DEF—GUD7.DEF	User-definable data

GUD user variables must be defined before they can be used (read/write). The following rules must be observed in this context:

- GUDs must be defined in a definition file, e.g. MGUD.DEF.
- PUDs and LUDs must be defined in a definition section of the part program.
- The data must be defined in a dedicated block.
- Only one data type may be used for each data definition.
- Several variables of the same data type can be defined for each data definition.

A GUD variable is defined with the following:

- Keyword DEF
- Range of validity NCK or CHAN
- Data type (INT, REAL, BOOL, ….)
- Variable names
- Value assignment (optional)

Examples

DEF NCK INT COUNTER1
DEF CHAN REAL X_POS

Local user variables (LUD)

Local user variables (LUD) are variables defined in a part program which is not the main program at the time of execution. They are created when the part program is called, and deleted at the end of the part program or when the NC is reset. Local user variables can only be accessed within the part program in which they are defined.

A LUD is defined with the following:

- Keyword DEF
- Data type (INT, REAL, BOOL, ….)
- Variable names
- Value assignment (optional)

Program-global user variables (PUD)

Program-global user variables (PUD) are user variables defined in a part program used as the main program. They are created when the part program starts, and
deleted at the end of the part program or when the NC is reset. It is possible to access PUD in the main program and in all subprograms of the main program.

A PUD is defined with the following:

- Keyword DEF
- Data type (INT, REAL, BOOL, ….)
- Variable names
- Value assignment (optional)

Syntax for GUDs, LUDs, PUDs

GUD
DEF <range> <pp_stop> <access_rights> <data class> <type>
<phys_unit> <limit values> <name>[<value_1>, <value_2>, <value_3>]=<init_value>

LUD and PUD
DEF <type> <phys_unit> <limit values> <name>[<value_1>, <value_2>,
<value_3>]=<init_value>

Notes:

Meaning of Syntax

DEF:	Command for defining GUD, PUD, LUD user variables	
<range>:	Range of validity, only relevant for GUD:	
	NCK:	NC-global user variable
	CHAN:	Channel-global user variable
<PP_stop>:	Preprocessing stop, only relevant for GUD (optional)	
	SYNR:	Preprocessing stop when reading
	SYNW:	Preprocessing stop when writing
	SYNRW:	Preprocessing stop when reading/writing
<access rights>:	Protection level for reading/writing GUD via part program or OPI (optional)	
	APRP <protection level>:	Read: Part program
	APWP <protection level>:	Write: Part program
	APRB <protection level>:	Read: OPI
	APWB <protection level>:	Write: OPI
	<protection level>:	Range of values: 0 … 7

<type>:	Data type:	
	INT:	Integer with sign
	REAL:	Real number (LONG REAL to IEEE
	BOOL:	Truth value TRUE(1)/FALSE(0)
	CHAR:	ASCII character
	STRING: [<Maxlength>]:	Character string of a defined length
	AXIS:	Axis/spindle identifier
	FRAME:	Geometric data for a static coordinate transformation
<phys_unit>:	Physical unit (optional)	
	PHU <unit>:	Physical unit
<limit values>	Lower/upper limit value (optional)	
	LLI <limit value>:	Lower limit value (lower limit)
	ULI <limit value>:	Upper limit value (upper limit)
<name>:	Name of variable NOTE: ♦ Maximum 31 characters ♦ The first 2 characters must be a letter and/or an underscore ♦ The $ sign is reserved for system variables and must not be used	
[<value_1, <value_2, <value_3>]:	Specification of array sizes for 1– to max 3-dimensional array variables (optional)	
<init_value>	Initialisation value (optional)	

Examples

Example 1 : Definition of GUD user variables in the data block for machine user (UGUD)

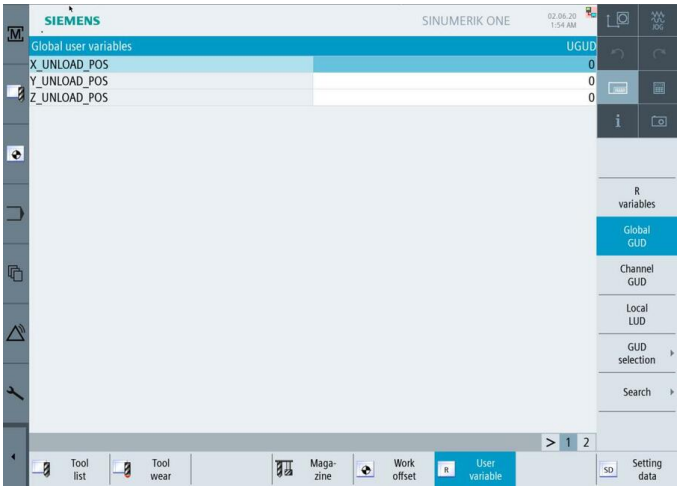
UGUD.DEF Program code

```
DEF NCK REAL X_UNLOAD_POS
DEF NCK REAL Y_UNLOAD_POS
DEF NCK REAL Z_UNLOAD_POS
M30
;Comments
;Definition of three GUD variables: X_UNLOAD_POS, Y_UNLOAD_POS, Z_UNLOAD_POS
;Range of validity: Throughout the NCK
;Data type: REAL
;Physical unit: Not programmed => default value = 0 = no phys. unit
;Access rights: Not programmed => default value = 7 = key-operated switch position 0
;Initialization value: Not programmed => default value = 0.0
```

Use in part/sub program

```
G71 G90 G01 G94 SUPA D0 Z=Z_UNLOAD_POS F20000
SUPA X=X_UNLOAD_POS Y=Y_UNLOAD_POS
```

This is how the definitions appear on the HMI in the “Parameter” operating area.



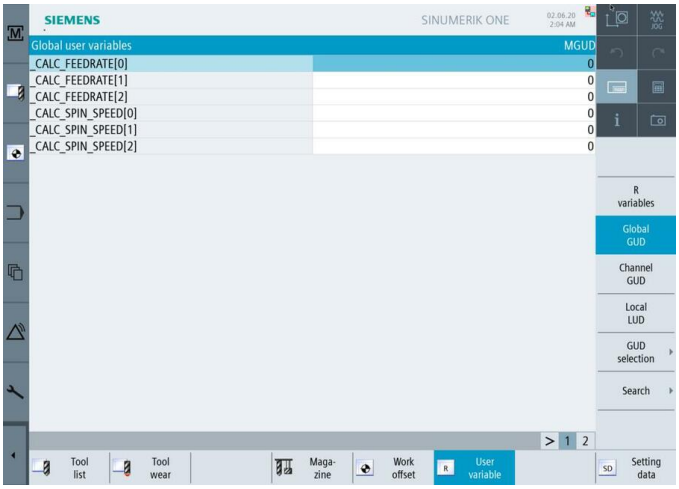
Notes:

Example 2: Definition of NCK GUD user variable types REAL and INT in the data block for machine manufacturer (MGUD), simple Arrays

MGUD.DEF Program code

```
APR 7 APW 7
DEF NCK REAL _CALC_FEEDRATE[3]
DEF NCK INT _CALC_SPIN_SPEED[3]
M30
;Comments
;Definition of six GUD variables:
;Range of validity: Throughout the NCK
;Data type: 3 x REAL, 3 x INT
;Physical unit: Not programmed => default value = 0 = no phys. unit
;Access rights:  = 7 = key-operated switch position 0
;Initialization value: Not programmed => default value = 0.0
```

This is how the simple Arrays appear on the HMI in the “Parameter” operating area.

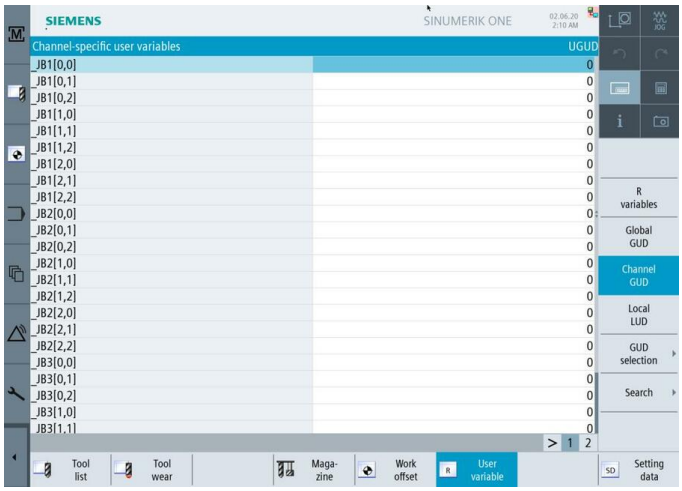


Example 3: Definition of CHAN GUD user variable types REAL in the data block for machine user (UGUD), Arrays

UGUD.DEF Program code

```
APR 7 APW 7
DEF CHAN REAL _JB1[3,3]
DEF CHAN REAL _JB2[3,3]
DEF CHAN REAL _JB3[3,3]
M30
;Comments
;Definition of 27 GUD variables:
;Range of validity: In the NC Channel
;Data type: REAL
;Physical unit: Not programmed => default value = 0 = no phys. unit
;Access rights:  = 7 = key-operated switch position 0
;Initialization value: Not programmed => default value = 0.0
```

This is how the Arrays appear on the HMI in the “Parameter” operating area.



Notes:

Example 4: Definition of NCK GUD user variables type REAL in the data block for machine manufacturer (MGUD), showing use of physical units, limit values, access rights and initialization values

MGUD.DEF Program code

```
DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 TIME_1=12, TIME_2=45
;Comments
;Definition of two GUD variables: TIME_1, TIME_2
;Data type: REAL
PP stop: Not programmed => default value = no PP stop
;Physical unit: 13 = [s]
;Limit values: low = 10.0, high = not programmed => upper definition range limit
;Access rights:
;Part program: Write/read = 3 = end user
;OPI: Write = 0 = Siemens, read = 3 = end user
;Initialization value: TIME_1 = 12.0, TIME_2 = 45.0
```

Example 5: Definition of an NCK GUD user variable type STRING in the data block for machine manufacturer (MGUD), showing use of access rights and initialization value

MGUD.DEF Program code

```
DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNT"
;Comments
;Definition of one GUD variable: GUD5_NAME
;Range of validity: Throughout the NCK
;Data type: STRING, max. 5 characters
PP stop: Not programmed => default value = no PP stop
;Physical unit: Not programmed => default value = 0 = no phys. unit
;Limit values: Not programmed => definition range limits: Low = 0, high = 255
;Access rights:
;Part program: Write/read = 3 = end user
;OPI: Write = 0 = Siemens, read = 3 = end user
;Initialization value: "COUNT"
```

Example 6: Global program and local user variables (PUD/LUD)

Main Program code	Comment
PROC MAIN	; Main program
DEF INT VAR1	;PUD definition
....	
SUB2	;Subprogram SUB2
….	
M30	

SUB2 Program code	Comment
PROC SUB2	;Subprogram SUB2
DEF INT VAR2	;LUD DEFINITION
....	
IF (VAR1==1)	;Read PUD
VAR1=VAR1+1	;Read & write PUD
VAR2=1	;Write LUD
ENDIF	
SUB3	;Subprogram call
….	
M17	

SUB3 Program code	Comment
PROC SUB3	;Subprogram SUB3
….	;
IF (VAR1==1)	;Read PUD
VAR1=VAR1+1	;Read & write PUD
VAR2=1	;Error: LUD from SUB2 not known
ENDIF	
….	
M17	

Notes:

Redefinition of system variables, user variables, and NC language commands (REDEF)

The REDEF command changes the attributes of system variables, user variables and NC language commands. A fundamental condition of redefinition is that it has to post-date the corresponding definition.

Multiple attributes cannot be changed simultaneously during redefinition. A separate REDEF command must be programmed for each attribute to be changed.
If several concurrent attribute changes are programmed, the last change is always active.

Resetting attribute values
The attributes for access rights and initialization time change with REDEF can be reset to their default values by reprogramming REDEF, followed by the name of the variable or the NC language command:

- Access rights: Protection level 7
- Initialization time: No initialization or retention of the current value

Local user variables (PUD/LUD)
Redefinitions are not permitted for local user variables (PUD/LUD).

Examples

Example 1 : Redefinition of the access rights of machine data in the data block for machine manufacturer (MGUD)

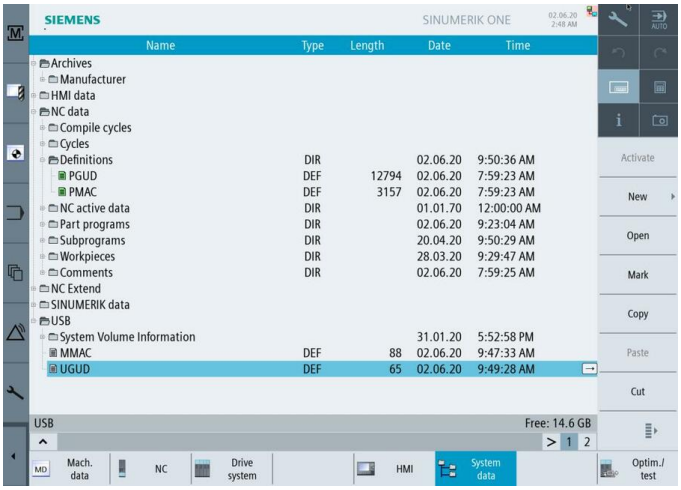
MGUD.DEF Program code

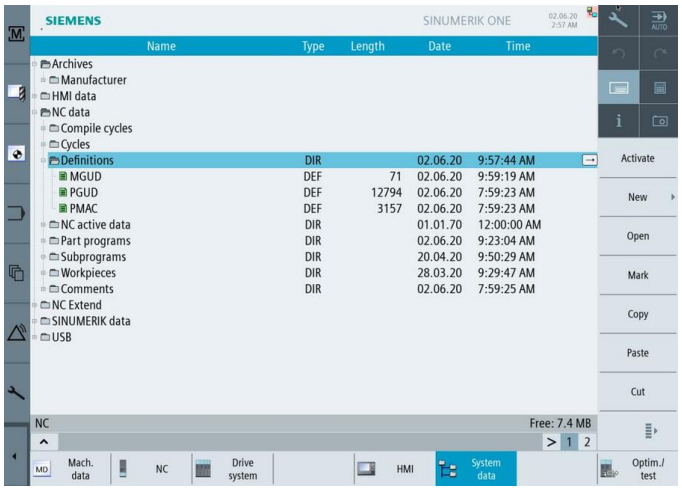
```
REDEF $MN_DRIVE_IS_ACTIVE APR 7 APW 7
REDEF $MC_AXCONF_MACHAX_USED APR 7 APW 7
REDEF $MA_CTRLOUT_TYPE APR 7 APW 7
REDEF $MA_NUM_ENCS APR 7 APW 7
;Comments
;Redefinition of the access rights of machine data for use in OEM/user cycles
```

Creating global user data (GUD)

GUDs definition files can be created offline and then transferred to the control system using copy/paste or it is also possible to create the definition file online

It is also possible to create the GUD file online by pressing the “New” softkey when on the Definitions directory.

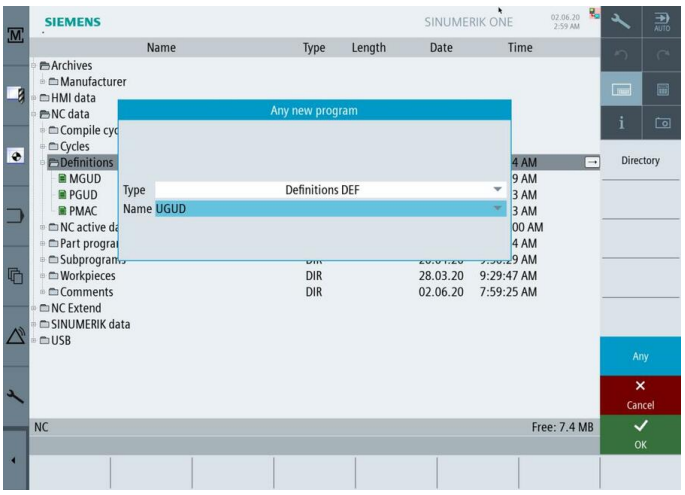




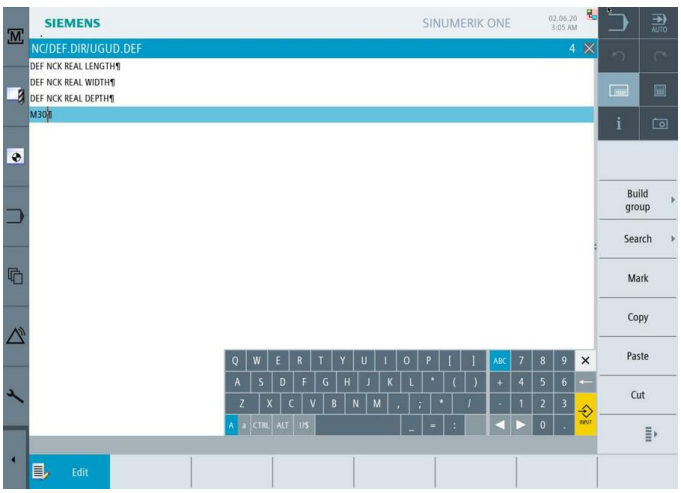
In the System data area of the “Startup” operator area it would be possible for example to copy/paste a file generated offline from USB to the Definitions directory.

Notes:

The “Toggle” key can be used to select the definition file to be generated, i.e. MGUD, UGUD, GUD4 to GUD7.

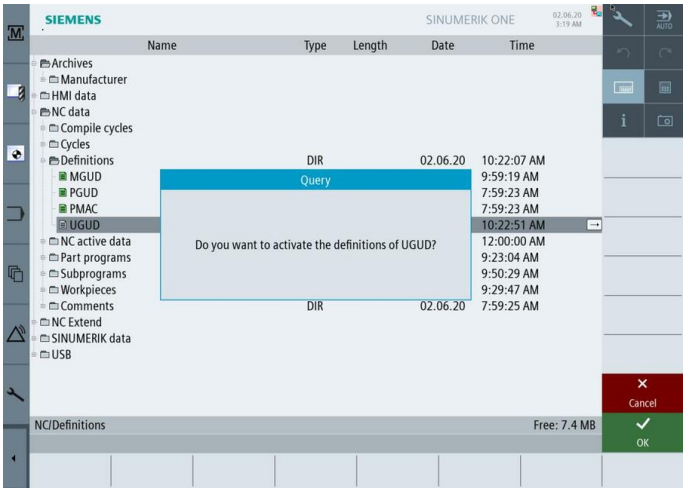


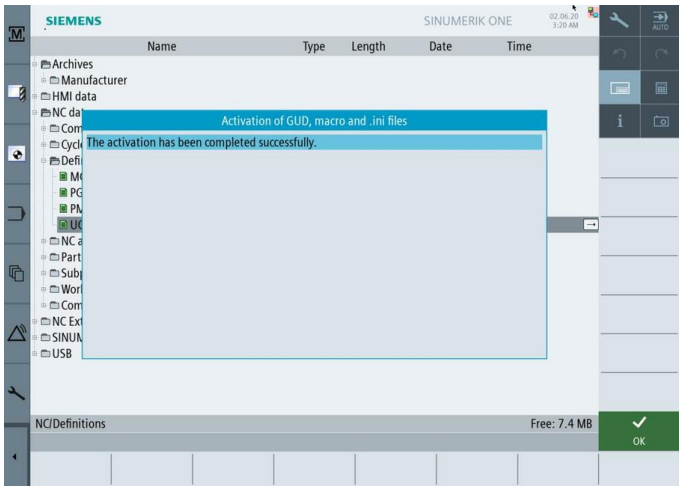
Press “OK” and the program editor will open. Now type in the required GUD variables



Activating global user data

When the GUD file is complete and you close it you will be prompted to activate the definitions





Once the Definitions are successfully activated the screen above will be shown, and the file will be shown in a green colour in the Definitions directory.



Notes:

Editing global user data

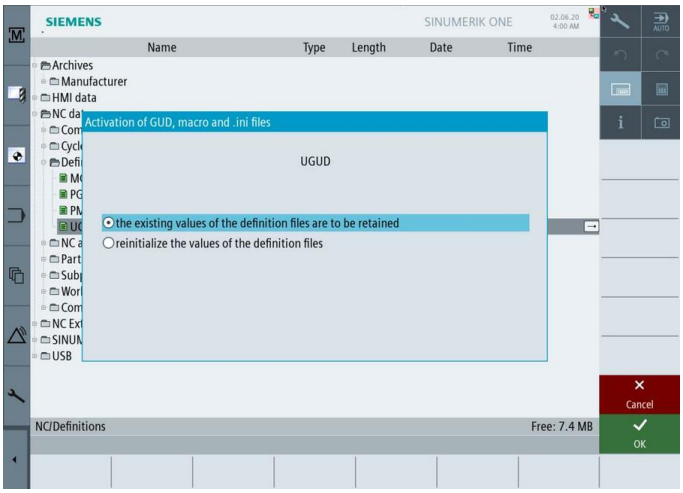
If a definition file is edited you will be prompted to reactivate the GUDs.

1. The prompt shown opposite is displayed.

2. Select whether the current values in the definition files should be retained

- OR -
Select whether the current values in the definition files should be deleted.
This will overwrite the definition files with the initial values.

3. Press the "OK" softkey to continue the process.



Once the definition file is activated it will be shown in a green colour

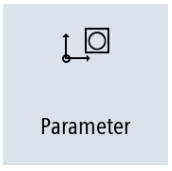


Displaying global user data (GUD)

Global user variables

Global GUDs are NC global user data (Global UserData) that remains available after switching the machine off. GUDs apply in all programs.

Procedure to display Global GUDs



1. Select the “Parameter” operating area



2. Press the “User variable” softkey



3. Press the “Global GUD” softkey



Press the “GUD selection” softkey and the “SGUD” to “GUD7” softkeys if you wish to display SGUD, MGUD, UGUD as well as GUD4 to GUD7 of the global user variables.

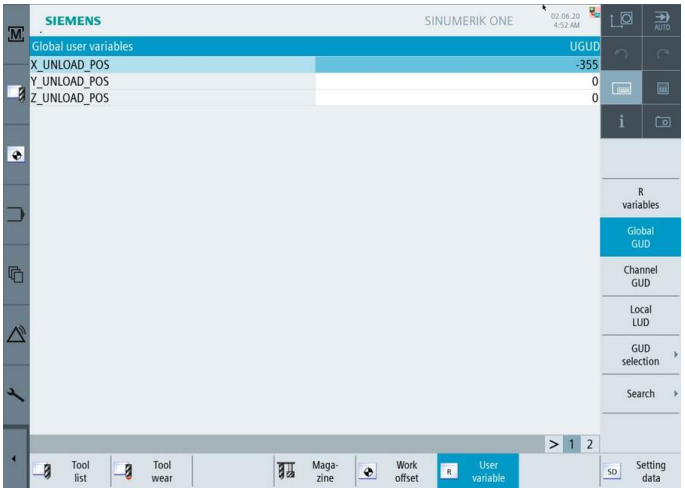




GUD4 to GUD7 selection softkeys will only be available if GUD4 to GUD7 have been configured

Notes:

Display of Global user data



Parameter

1. Select the “Parameter” operating area

User variable

2. Press the “User variable” softkey

Channel GUD

3. Press the “Channel GUD” and “GUD selection” softkeys

GUD selection

SGUD

4. A new vertical softkey bar appears
Press the “SGUD”….”GUD7” softkeys if you want to display the SGUD, MGUD, UGUD as well as GUD4 to GUD7 of the channel-specific user variables

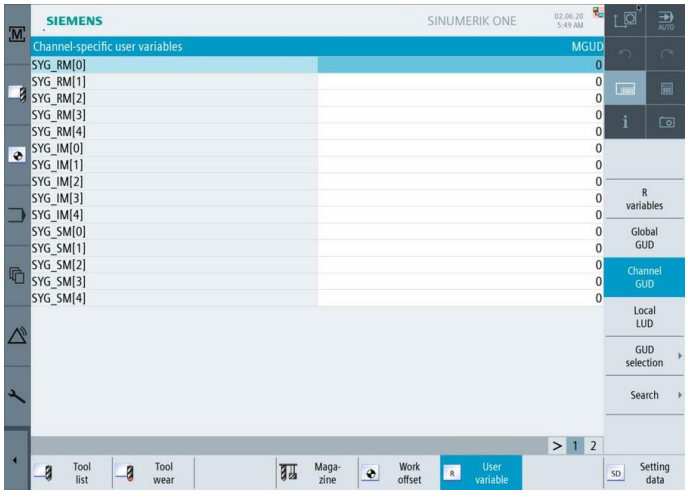
GUD7

Displaying channel GUDs

Channel-specific user variables

Like the Global GUDs, channel-specific user variables are applicable in all programs for each channel.

Display of Channel-specific user variables



Displaying local user data (LUD)

Local user variables

LUDs are only valid in the program or subprogram in which they were defined.
The controller displays the LUDs after the start of program processing. The display is available until the end of program processing.

Procedure:

Parameter

1. Select the “Parameter” operating area

User variable

2. Press the “User variable” softkey

Local LUD

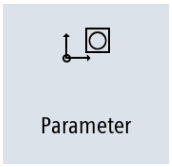
3. Press the “Local LUD” softkey

Notes:

Displaying program user data (PUD)

Program-global user variables

PUDs are global part program variables (Program User Data). PUDs are valid in all main programs and subprograms, where they can also be written and read.



1. Select the "Parameter" operating area.



2. Press the "User variable" softkey.





3. Press the "Program PUD" softkey.

Note:

Visibility of program-global user variables (PUD)

Program-global user variables (PUD) defined in the main program will only be visible in subprograms if the following machine data is set:

MD11120 $MN_LUD_EXTENDED_SCOPE = 1

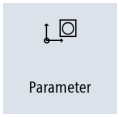
If MD11120 = 0 the program-global user variables defined in the main program will only be visible in the main program.

Only local or program-global user data can be displayed at one time.
Whether the user data are available as LUD or PUD depends on the current control configuration.

Searching for user variables

You can search for “R parameters or user variables.

Procedure via “Parameter”



1. Select the "Parameter" operating area.





2. Press the "R parameters", "Global GUD", "Channel GUD", "Local GUD" or "Program PUD" softkeys to select the list in which you would like to search for user variables.

3. Press the "Search" softkey. The "Search for R Parameters" or "Search for User Variables" window opens.



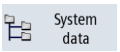
4. Enter the desired search term and press "OK". The cursor is automatically positioned on the R parameter or user variables you are searching for, if they exist.

By editing a DEF/MAC file, you can alter or delete existing definition/macro files or add new ones.

Procedure via “Setup”

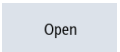


1. Select the "Setup" operating area.



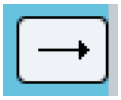
2. Press the "System data" softkey.

3. In the data tree, select the "NC data" folder and then open the "Definitions" folder.
4. Select the file you want to edit.



5. Press the "Open" softkey.

OR -



Press the <Cursor right> key.
The selected file is opened in the editor and can be edited there.

6. Define the desired user variable.

7. Press the "Close" softkey to close the editor.

Notes:

Reading/Writing Global User data

Function Description:

Global User Data can be both read and written to by the automation system (PLC).

Global User Data are often referred to as GUDs . The OEM is able to define his own user variables in this way.
Dependant upon the OEM functionality of the relevant GUD it may be necessary in process to read the current value to the PLC or to write the value from the PLC.

With the function block FB5 you are able to read the value of a GUD.

Because GUDs are generated by the OEM (user), it is not possible to obtain the internal address of the variable with the NC Var selector in order to write the GUD.

By reading the content of the variable with FB5 and with additional parameterization of the FB5 the internal address can be obtained as a result parameter of the FB5, this result / output parameter can be used as a destination parameter with the FB3 allowing an indirect way to write the GUDs.

Prerequisites for the Function:

HMI Requirement:
None

Drive requirement (Sinamics):
None

NCK Requirement:
GUD Data available

PLC Requirement:
Parameterization of FB5 in the PLC

Steps to achieve function:

Parameter description of **FB5**

Signal	Type	Format	Value range	Comment
Req	I	BOOL		Job start with positive signal edge
Addr	I	VARIANT	[DBName].[VarName]	GUD variable name in a variable of data type STRING
Area	I	BYTE	0, 2	Area address: 0: NCK variables 2: Channel variables
Unit	I	BYTE	1, 2, .. 10	NCK area: Unit:=1 Channel area: Channel no.
Index1	I	INT	0, 1, 2, .. <max. field index>	Field index 1 of variable Variable has the value 0 if no field index is used.
Index2	I	INT	0, 1, 2, .. <max. field index>	Field index 2 of variable Variable has the value 0 if no field index is used.

Signal	Type	Format	Value range	Comment
CnvtToken	I	BOOL		Activate generation of a 10 byte variable token
VarToken	I	VARIANT	[DBName].[S7-Name]	Address to a 10-byte token (see example)
Error	Q	BOOL		Negative acknowledgment of job or execution of job impossible
Done	Q	BOOL		Job successfully executed
State	Q	WORD		See error identifiers
RD	I/Q	VARIANT		Target area for read data, e.g. DB area or Marker of the same data type as the variable to be read

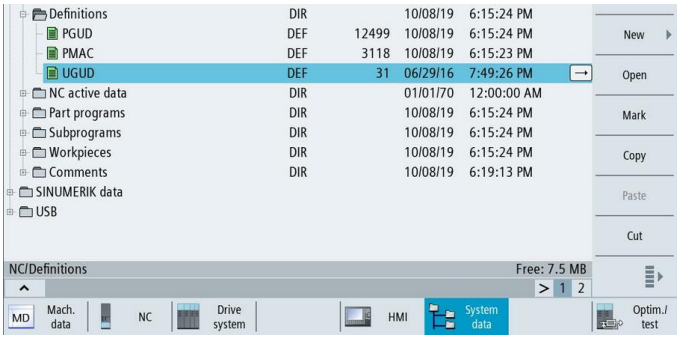
Notes:

FB5
Error codes

State	Description	Note
1	Access error	
2	Error in job	Incorrect compilation of variables in a job
3	Negative acknowledgement, job not executable	Internal error, possible remedy. NC-Reset
4	Data areas or data types do not tally	Check data to be read in RD
4	Insufficient local user memory available	read variable is longer than specified in RD
5	Format conversion error	Adr parameter must be transferred as string
6	FIFO full	Job must be repeated since queue is full
10	Error in variable addressing	Unit contains value 0
11	Address of variable invalid	Address check (or variable name), "Area", "Unit"
15	String more than 32 characters	GUD variable name too long

In order to read a GUD value, the OEM must first generate and activate the GUD file using the HMI function available in the System explorer.

Create and activate GUD file and variables



Syntax in order to generate an NCK GUD array variable with the name __GJ.

```
NC/DEF.DIR/UGUD.DEF
N100 DEF NCK INT __GJ[10]¶
M30¶
```

The name of the GUD name has to be parameterized as a STRING variable to the function block, at parameter; Addr.
If the GUD variable is an array, the array index can be parameterized as well.

According to the description, field index refers to array index.

You can additionally use the parameters, to obtain the address of the GUD for parameterization with FB3.

CnvtToken	Activate generation of a 10 byte variable token
VarToken	[DBName].[VarName] Address to a 10-byte token

Notes:

DB "PLCFunc" with GUD string

PLCFunc

	Name	Data type	Start value
1	Static		
2	L_IN	String[8]	'SUNRISE '
3	GUD_GJ	String[8]	'__GJ'
4	GUD_GJ_TOKEN	Struct	
5	SYNTAX_ID	Byte	16#0
6	Area_Unit	Byte	16#0
7	Column	Word	16#0
8	Line	Word	16#0
9	BlockType	Byte	16#0
10	NumOfLine	Byte	16#0
11	DataType	Byte	16#0
12	Length	Byte	16#0
13	Path	String[32]	'/_N_CMA_DIR/'
14	PName	String[32]	'_N_PLCASUP1_SPF'

Example: Reading and writing GUD parameter.

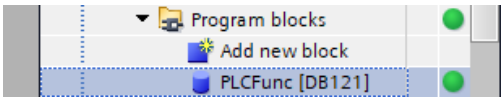
```
NETWORK
TITLE =Read a GUD FB5

      A     I     81.4;
      A     I     84.1; // Rotary switch 1 - position 2
      FP    M    128.0; // Transition Marker
      S     M    128.1;
      A     M    128.3;
      FP    M    128.4; // Transition Marker
      R     M    128.1;
      CALL "LBP_ReadGUD" , "InstReadGud" (
           Req                := M    128.1,
           Addr               := DB121.GUD_GJ,
           Area               := B#16#0,
           Unit               := B#16#1,
           Index1             := 2,//__GJ[2]
           Index2             := 0,
           CnvtToken          := TRUE,
           VarToken           := DB121.GUD_GJ_TOKEN,
           Error              := M    128.2,
           Done               := M    128.3,
           State              := MW   129,
           RD                 := MD   131);
```

```
NETWORK
TITLE =Write a GUD FB3

      A     I     81.5;
      A     I     84.1; // Rotary switch 1 - position 2
      FP    M    143.0; // Transition Marker
      S     M    143.1;
      A     M    143.3;
      FP    M    143.4; // Transition Marker
      R     M    143.1;
      CALL "LBP_WriteVar" , "InstWriteGud" (
           Req                := M    143.1,
           NumVar             := 1,
           Addr1              := DB121.GUD_GJ_TOKEN,
           Line1              := W#16#4,//__GJ[3]
           Error              := M    143.2,
           Done               := M    143.3,
           State              := MW   144,
           SD1                := MD   131);
```

Following the VAR Data Block with Symbol name and DB Number.



Content of VAR DB with STRUCT information.

PLCFunc

	Name	Data type	Start value
1	Static		
2	L_IN	String[8]	'SUNRISE '
3	GUD_GJ	String[8]	'__GJ'
4	GUD_GJ_TOKEN	Struct	
5	SYNTAX_ID	Byte	16#0
6	Area_Unit	Byte	16#0
7	Column	Word	16#0
8	Line	Word	16#0
9	BlockType	Byte	16#0
10	NumOfLine	Byte	16#0
11	DataType	Byte	16#0
12	Length	Byte	16#0
13	Path	String[32]	'/_N_CMA_DIR/'
14	PName	String[32]	'_N_PLCASUP1_SPF'

Notes:

Macros (DEFINE ... AS)

A macro is a sequence of individual statements which have together been assigned a name of their own. G, M and H functions or L subprogram names can also be used as macros. When a macro is called during a program run, the statements programmed under the program name are executed one after the other.

Application

Statement sequences that are repeated are only programmed once as macro in a dedicated macro block (macro file) or once at the beginning of the program. The macro can then be called in any main program or subprogram and executed.

Activation

In order to use the macros of a macro file in the NC program, the macro file must be available and activated in the NC.

File Name	Meaning
MMAC.DEF	Definitions for manufacturer macro
UMAC.DEF	Definitions for user macro

Syntax

Macro definition:
DEFINE <Macro name> AS <statement 1> <statement 2> ...
Call in the NC program:
<macro name>

DEFINE .... AS:	Keyword combination to define a Macro
<macro name>:	Macro name Only identifiers are permissible as macro names The macro is called from the NC program eith the macro name
<statement>:	Programming statement that should be included in the macro

Rules when defining a macro
- Any identifier, G, M, H functions and L program names can be defined in a macro.
- Macros can also be defined in the NC program.
- G function macros can only be defined in the macro module globally for the entire control (modal).
- H and L functions can be programmed with two digits.
- M and G functions can be programmed with three digits.

! Supplementary conditions
Nesting of macros is not possible.

Example 1: Macro definition at the beginning of a program

Program code	Comment
DEFINE LINE AS G1 G94 F300	; Macro definition
.....	
N70 LINE X10 Y20	; Macro call
.....	

Example 2: Macro definitions in a macro file

Program code	Comment
DEFINE M6 AS L6	; A subprogram is called at tool change to handle the necessary data transfer. The actual M function is output in the subprogram (e.g. M106)
DEFINE G81 AS DRILL(81)	; Emulation of the DIN G function
DEFINE G33 AS M333 G333	; During thread cutting, synchronisation is requested with the PLC. The original G function G33 was renamed to G333 by machine data so that the programming is identical for the user

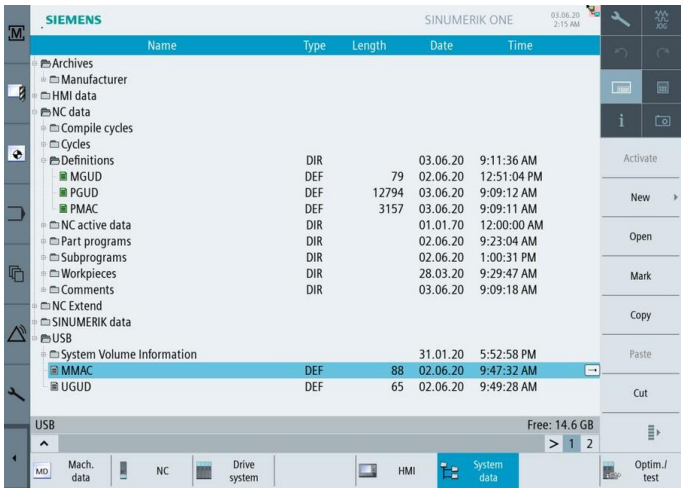
Example 3: The macro file must be transferred into the definitions directory and be activated. Only then can the macros be used in the NC program

Program code	Comment
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; Spindle clockwise, coolant on
DEFINE M14 AS M4 M7	; Spindle counter-clockwise, coolant on
DEFINE M15 AS M5 M9	; Spindle stop, coolant off
DEFINE M6 AS L6	; Call tool change program
DEFINE G80 AS MCALL	; Deselect drilling cycles
M30	

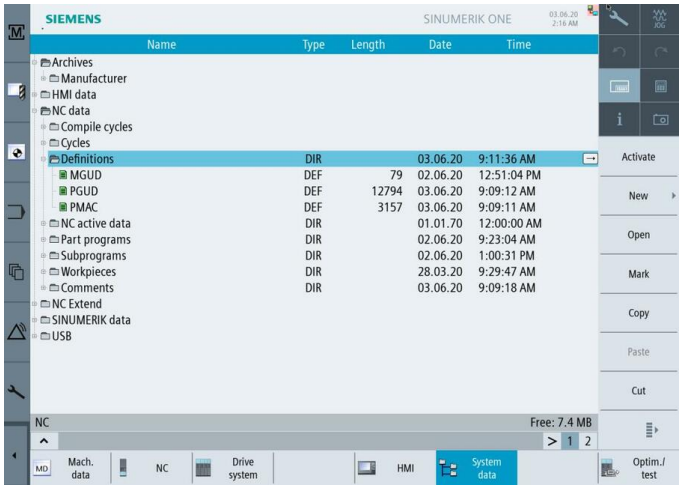
Notes:

Creating Macro files

Macro files can be created offline and then transferred to the control system using copy/paste or it is also possible to create the definition file online

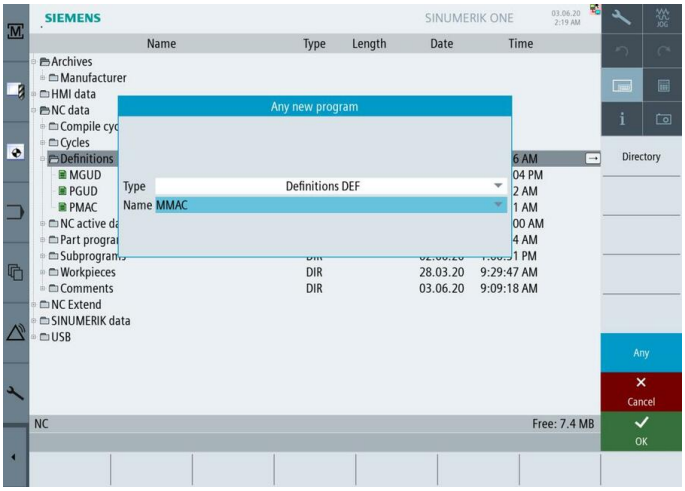


It is also possible to create the Macro file online by pressing the “New” softkey when on the Definitions directory.

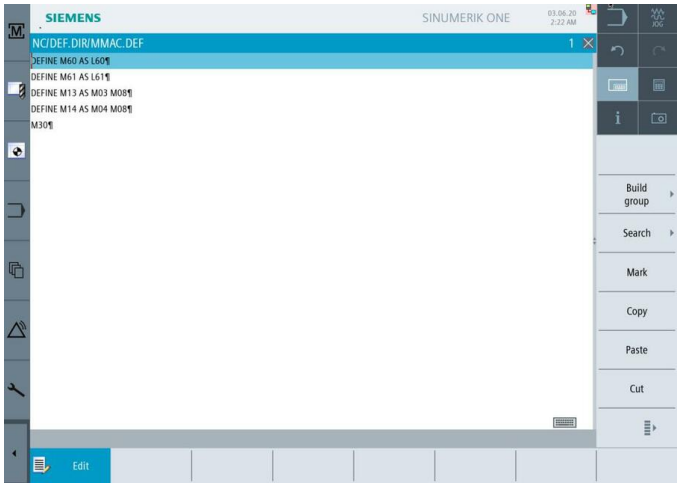


In the System data area of the “Startup” operator area it would be possible for example to copy/paste a file generated offline from USB to Definitions directory.

The “Toggle” key can be used to select the Macro file to be generated, i.e. MMAC, UMAC.



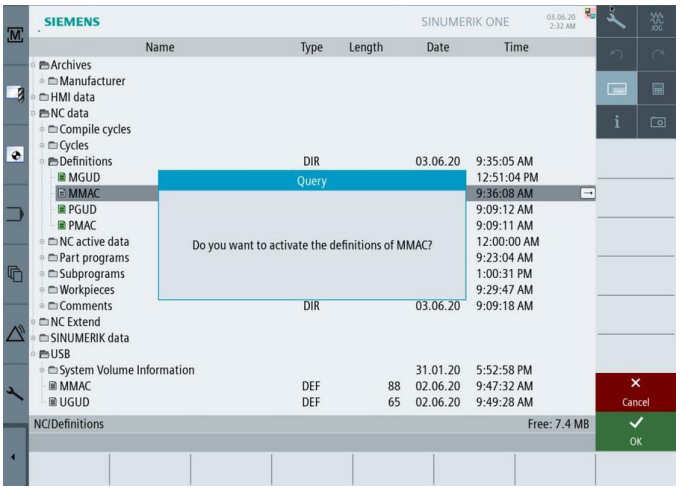
Press “OK” and the program editor will open. Now type in the required macro instructions

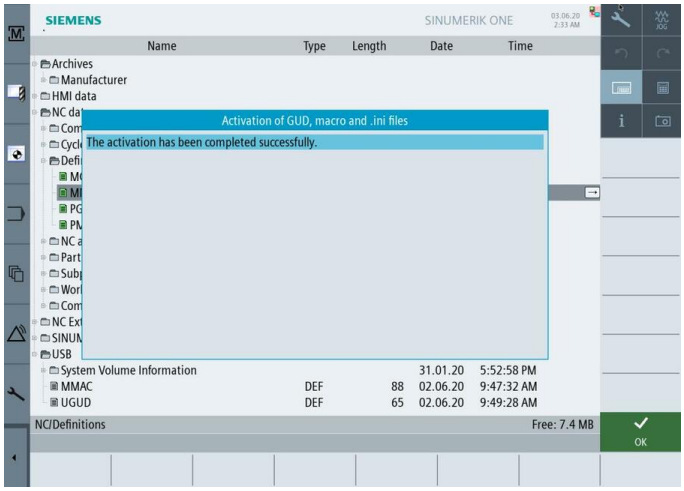


Notes:

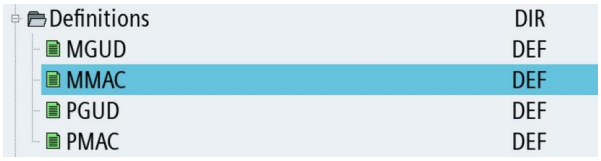
Activating macros

When the Macro file is complete and you close it you will be prompted to activate it.





Once the macro is successfully activated the screen above will be shown, and the file will be shown in a green colour in the Definitions directory.



Editing macros

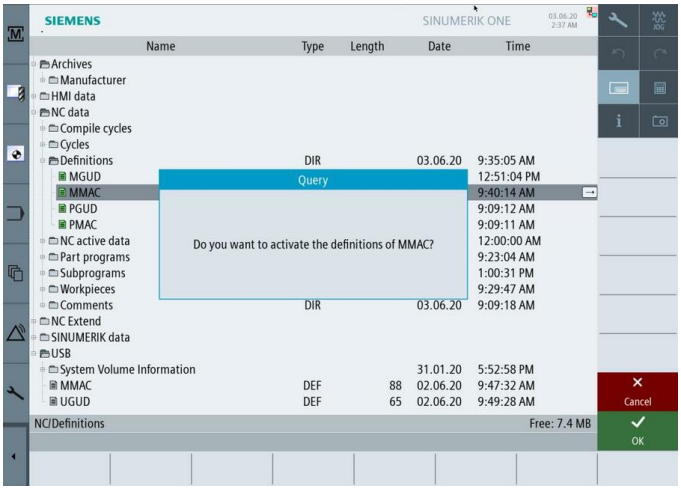
If a Macro file is edited you will be prompted to reactivate it.

1. The prompt shown opposite is displayed.

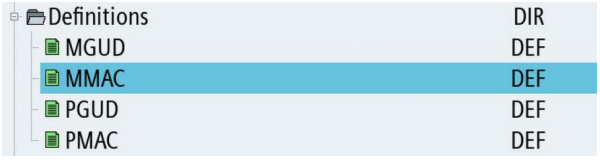
2. Select whether the current values in the definition files should be retained

- OR -
Select whether the current values in the definition files should be deleted.
This will overwrite the definition files with the initial values.

3. Press the "OK" softkey to continue the process.



Once the macro file is activated it will be shown in a green colour



Notes:

Using NC Machine data to call subroutines

It is possible to use NC general MD to define an M code to call a subroutine instead of using a macro definition

MD 10715[0] to [29] $MN_M_NO_FCT_CYCLE and
MD 10716[0] to [29] $MN_M_NO_FCT_CYCLE_NAME

Can be used for this purpose

E.g. DEFINE M6 AS L6
DEFINE M62 AS L62
DEFINE M63 AS L63

Could be realised in the following way using MD

10715[0]	$MN_M_NO_FCT_CYCLE	6	po
10715[1]	$MN_M_NO_FCT_CYCLE	62	po
10715[2]	$MN_M_NO_FCT_CYCLE	63	po

10716[0]	$MN_M_NO_FCT_CYCLE_NAME	L6	po
10716[1]	$MN_M_NO_FCT_CYCLE_NAME	L62	po
10716[2]	$MN_M_NO_FCT_CYCLE_NAME	L63	po

A303: END

Notes:

A305

SINUMERIK ONE

Synchronous Actions

Module Description:

Synchronous actions permit certain actions to be performed in parallel, irrespective of the current state of the program. These cycles can be called conditionally or in every interpolation cycle.

Synchronous actions are used mainly for start-up topics, such as custom cycles, but can also be used effectively in part programs.

Module Objective:

This module describes the use of synchronous actions in NC programs, whereby the possible application scenarios including program examples are shown.

Content:

Introduction

Syntax

Functions

Coordination

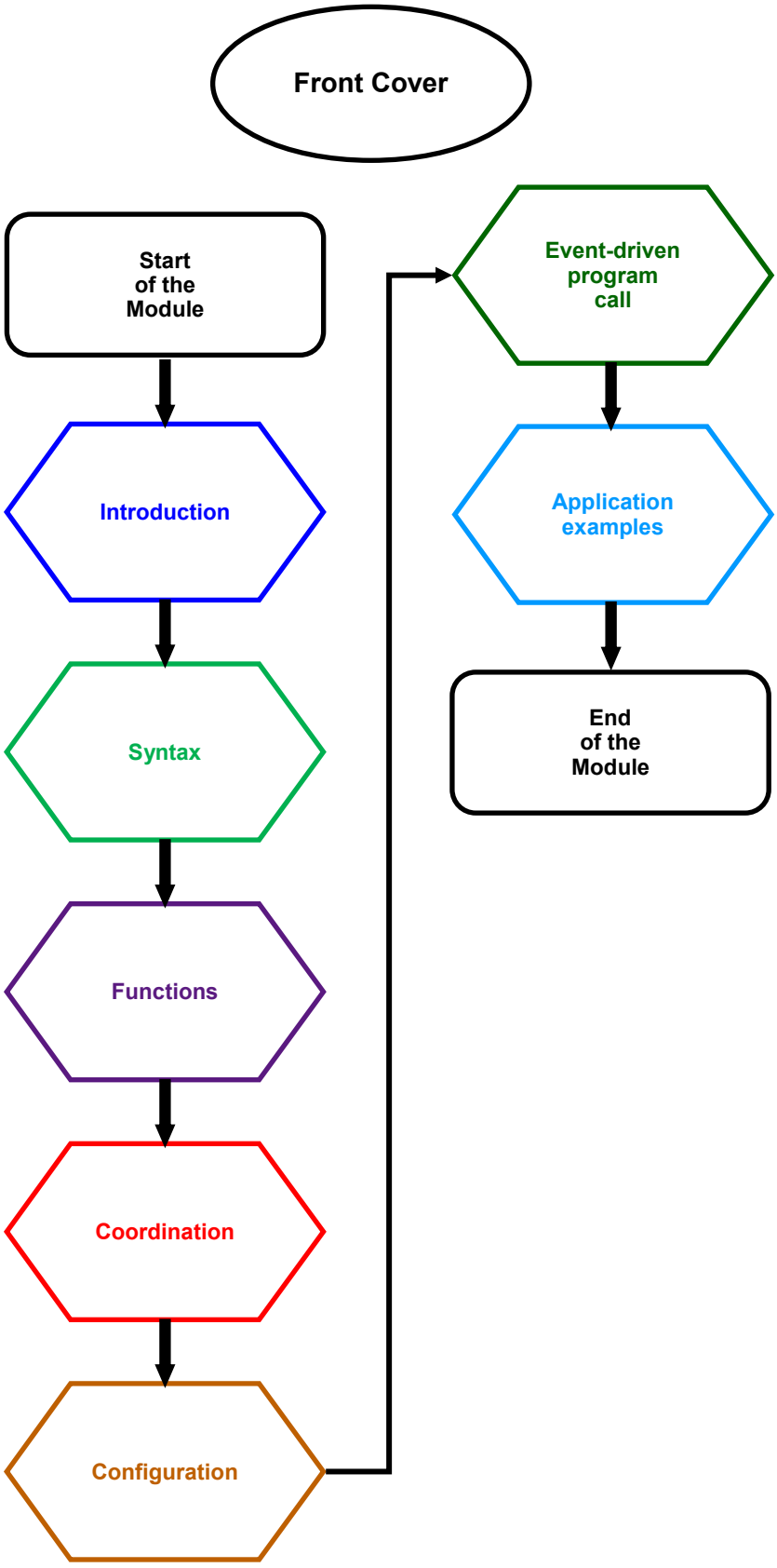
Configuration

Event-driven program call

Application examples

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Introduction
Syntax
Functions
Coordination
Configuration
Event-driven program call
Application examples
End of the Module

Introduction :

Basics
Motion-synchronous actions (shortened to synchronous actions) are statements programmed by the user (or OEM) that are evaluated by the NCK synchronously to the processing of the part program in the interpolation cycle.

If the condition contained in the synchronous action is satisfied, or not, the assigned actions are activated synchronously for further processing.

This module describes the various parts of synchronous actions and their application.

Structure of synchronous actions :

The syntax of synchronous actions is always structured similarly:

	Validity, identification number	Frequency	G code condition	Condition	DO	G code action	Action
All possibilities	ID=[1...999] IDS=[1...999]	*None* WHENEVER FROM WHEN EVERY	G70 G71 G700 G710	...	DO	G70 G71 G700 G710	...
Example	IDS=1	EVERY	G70	$AA_IM[B] > 15	DO	G71	POS[X] = 100

The individual parts of a call are explained in detail in the following.

Notes:

	Validity, identification number	Frequency	G code condition	Condition	DO	G code action	Action
Example	IDS=1	EVERY	G70	$AA_IM[B] > 15	DO	G71	POS[X] = 100

Validity, identification number

There are three ways to specify the range of validity of a synchronous action:

- **No specification**: Act non-modally and only in automatic mode. Non-modal synchronous actions act modally across all (also implicitly generated) preprocessing stop blocks and across implicitly generated intermediate blocks.
- **ID = 1...999:** Act modally in all the following programmed blocks and only in automatic mode. They act until a synchronous action with the same identification number (also IDS) is programmed, or until they are deleted by a CANCEL statement.
- **IDS = 1...999:** Statically effective synchronous actions (option M43) are active in all operating modes.

Synchronous actions are deleted with **CANCEL(i)**.

A newly programmed synchronous action replaces a previous synchronous action with the same number. It is not relevant whether the synchronous action has been programmed with ID or IDS!
Within an interpolation cycle, modal synchronous actions are processed in the order of their ascending ID numbers. Only then are the non-modal synchronous actions processed.

	Validity, identification number	Frequency	G code condition	Condition	DO	G code action	Action
Example	IDS=1	EVERY	G70	$AA_IM[B] > 15	DO	G71	POS[X] = 100

Frequency

None	Action is executed in each interpolation cycle.
WHENEVER	Action is executed cyclically in each interpolation cycle whenever the condition is satisfied.
FROM	When the condition has been satisfied once, the action is executed cyclically in each interpolation cycle whenever the synchronous action is satisfied.
WHEN	When the condition has been satisfied, the action is executed once. The condition is then no longer checked.
EVERY	Action is executed once when the condition has been satisfied. Thereafter, the action is only executed when the state of the condition changes from False to True again.

Notes:

	Validity, identification number	Frequency	G code condition	Condition	DO	G code action	Action
Example	IDS=1	EVERY	G70	$AA_IM[B] > 15	DO	G71	POS[X] = 100

G code condition

The G codes G70, G71, G700 and G710 can be programmed in synchronous actions. In this way, it is possible to have a defined setting for the evaluation of the condition, irrespective of the currently active state of the part program. The decoupling of synchronous actions from the program environment is necessary because synchronous actions can occur at any time and a defined state is required for the evaluation.

The dimension systems can be specified with the permitted G codes. Only one G code per condition is permitted.

G70 dimensions in inch
G71 dimensions in mm
G700 dimensions in inch and inch/min
G710 dimensions in mm and mm/min

	Validity, identification number	Frequency	G code condition	Condition	DO	G code action	Action
Example	IDS=1	EVERY	G70	$AA_IM[B] > 15	DO		POS[X] = 100

Condition

A condition is a logical expression which specifies the requirements that have to be satisfied for a synchronous action to be executed. If a condition has not been specified, the execution of the action is only specified via the frequency.

The following conditions are supported:

- Comparison of a real-time variable with an expression calculated in the preprocessing, *e.g. $AA_IM[X] > 10.5 * SIN (45)*
- Comparison of two real-time variables, *e.g. $AA_IM[X] > $$AA_IM[X1]*
- Comparisons are also possible with the Boolean operators **NOT, AND, OR, XOR, B_NOT, B_AND, B_OR, B_XOR**. Comparisons are possible between variables of the same type and partial expressions, *e.g. ($A_IN[1]==1) OR ($A_IN[3]==0)*

Action codeword DO
Each statement of a synchronous action must contain the action codeword DO, followed by the actual statement.
Action codeword ELSE
The else branch can be used for all query occurances - when, whenever, every, from

Notes:

	Validity, identification number	Frequency	G code condition	Condition	DO	G code action	Action
Example	IDS=1	EVERY	G70	$AA_IM[B] > 15	DO	G71	POS[X] = 100

G code action

As for the condition, one of the following G codes, G70, G71, G700 or G710 can be programmed for the action. In this way, it is possible to have a defined setting for the execution of the action, irrespective of the currently active state of the part program. The decoupling of synchronous actions from the program environment is necessary because synchronous actions can occur at any time and a defined state is required for the execution.

The dimension systems can be specified with the permitted G codes. Only one G code per condition is permitted.

G70 dimensions in inch
G71 dimensions in mm
G700 dimensions in inch and inch/min
G710 dimensions in mm and mm/min

	Validity, identification number	Frequency	G code condition	Condition	DO	G code action	Action
Example	IDS=1	EVERY	G70	$AA_IM[B] > 15	DO	G71	POS[X] = 100

Action

One or more actions or a technology cycle are programmed in each synchronous action. They are executed when the condition (including the specified frequency) has been satisfied. Up to 16 different actions can be programmed for each synchronous action. When the condition has been satisfied, they are executed in the same interpolation cycle.
Several actions are simply separated by blanks.

Technology cycles

A program (name) can also be specified as an action. All actions that can be programmed individually for synchronous actions are permitted in this program. These programs are also called technology cycles in the following. Technology cycles are therefore a number of actions that are executed sequentially in the interpolation cycle.

Notes:

Possible actions

Mxx, Sxx, Hxx	Auxiliary function output on PLC
SETAL(nr)	Set alarm, reaction to error
$A … = ; $V ... = $AA_OFF = $AC_OVR = $AA_OVR = $AC_VC = $AA_AC = $$SN_SW_CAM_... $AC_FCT...	Writing of real-time variables: - Overlaid movement - Velocity override Path velocity Axis velocity Add. feedrate offset Add. axis offset value - Changing of SW cam positions - Overwriting of FCTDEF parameters
RDISABLE STOPREOF DELDTG FTOC SYNFCT ZYKL_T1 (z.B.) ICYCON / ICYCOF	Synchronous action procedures: - Activate read disable - Stop preprocessing stop - Delete distance-to-go - Online tool offset - Polynomial evaluation - Call of technology cycles - Execute each block of tech. cyc. in separate IPO-cycle
MEAWA MEAC	Measuring without delete dist.-to-go Cyclic measuring

$AA_OVR[x] = 0 ACHSE_X (z.B.) POS[u] FA[u] MOV[u]	Control of the positioning axes: - Locking of an axis movement - Call of an axis program - Positioning - Specify axis feedrate (move axis continuously) - Move positioning axis
LOCK (id) UNLOCK (id) RESET (id) CANCEL (id)	Synchronous action / techn. cycle - Lock - Unlock - Reset - Cancel
SPOS M3, M4, M5, S = $AA_OVR[S1] = 0	Spindles: - Positioning - Direction of rotation, stop, speed - Locking of the spindle movement
PRESETON (,)	Set actual value
SETM CLEARM	Channel synchronization - Setting of a wait marker - Clearing of a wait marker

For a list of all possible actions, see SINUMERIK ONE Programming Manual.

Explanation of terms

Variables
All real-time variables are evaluated in the interpolation cycle. These include the main run and servo variables $A… / $V…. For special identification, they can be programmed in synchronous actions with $$ (i.e. $AA_IM[X] or synonymously $$AA_IM[X]). Setting data and machine data must be programmed with $$S… / $$M..., if their value changes during processing.

Preprocessing
The traversing blocks are processed before execution and stored in a preprocessing buffer.

Main run
The blocks of the part program decoded and processed in the preprocessing, are executed here.

Notes:

Functions for synchronous actions :

FCTDEF (PolynomialNo., lower limit, upper limit, a0, a1, a2, a3)

Define the polynomial function

y = a0 + a1 x + a2 x^2 + a3 x^3

Store in the following variable (n = polynomial number):
$AC_FCTLL[n]: Lower limit
$AC_FCTUL[n]: Upper limit
$AC_FCT0[n]: a0 coefficient
$AC_FCT1[n]: a1 coefficient
$AC_FCT2[n]: a2 coefficient
$AC_FCT3[n]: a3 coefficient

SYNFCT (PolynomialNo., RealtimeVariableOutput, RealtimeVariableInput)

The specified polynomial is evaluated with *x=RealtimeVariableInput* and the result written to *RealtimeVariableOutput*.

$AC_MARKER[n]

Marker/counter variables

Channel-specific markers of the INTEGER type. Markers can be read and written.
The number of markers is set via MD28256 (max. 20,000).

The markers are in the dynamic memory and are set to 0 after a Power On, NC reset and End of program.

$AC_TIMER[n]

Timer

A timer allows actions to be started after a defined wait time. The number of available timers is set via MD 28258.

Set timer: A timer is started by assigning a value, e.g. *$AC_TIMER[n]=1.3* (REAL start value in s)

Stop timer: Assignment of value –1

Read timer: The current time value can be read when the timer variables are running or stopped. When the timer is stopped by assigning –1, the current time value is retained and can still be read.

Example: WHEN $A_IN[1]==1 DO $AC_TIMER[1]=0
WHEN $AC_TIMER[1]>=0.5 DO $A_OUTA[3]=$AA_IM[X] $AC_TIMER[1]=-1

Notes:

$AC_PARAM[n]

Synchronous action parameters

The $AC_PARAM[n] variables of the REAL type are used as buffer storage in synchronous actions. They can be read and written. The number of parameters per channel is specified via MD28254.

The parameters are in the dynamic memory and are set to 0 after a Power On, NC reset and End of program.

$R...

With the prefix $ in front of R-parameters, they can also be used as real-time variables. If an R-variable used in the part program is to be used "normally" again in the part program, it must be synchronized using the STOPRE command.

Reading MD, SD that cannot change

Machine data and setting data that cannot change is addressed from the synchronous action as in normal part pro grams. A $ is used as prefix.

Reading and writing MD, SD that can change

Machine data and setting data that can change during processing is addressed from the synchronous action with $$ as prefix. This data can also be written with the appropriate access rights.

FIFO variables

Up to ten FIFO variables are available to store related data sequences: **$AC_FIFO1[n]** to **$AC_FIFO10[n]**. The number of available FIFO variables is specified in MD28260. The number of storable values per FIFO is specified in MD28264. Values in the FIFO have the REAL data type. Read and write access is possible to each element of the FIFO.

FIFO values are stored in the R-parameter range. The values are therefore in the static memory and are also retained after Power On, Reset and End of program. When archiving R parameters, the FIFO values are also saved. MD28262 specifies the number of R-parameters as of which FIFO variables are stored in the R-parameter range.

Special indices

- **n=0:** When writing, a new value is stored in the FIFO. When reading, the oldest element is read and removed from the FIFO
- **n=1:** Access to the oldest stored element
- **n=2:** Access to the newest stored element
- **n=3:** Sum of all FIFO elements
  (MD28266 determines the mode of the summation:
  Bit 0 = 1: Update the sum with each entry
  Bit 0 = 0: No summation)
- **n=4:** Number of elements available in the FIFO
  (reset of the FIFO with, e.g. $AC_FIFO1[4]=0)
- **n=5:** Current write index relative to the start of the FIFO

Notes:

Coordination of synchronous actions :

Control functions

LOCK

Lock technology cycle
An active action is interrupted.

UNLOCK(ID)

Continue the technology cycle from the position where it was interrupted
Interrupted positioning operation is continued.

RESET(ID)

Cancel technology cycle
Active positioning operations are cancelled. When the cycle is restarted, processing is started with the first block.
Depending on the synchronous action type, the actions are executed when the condition occurs again.
Already executed actions of the WHEN type are no longer processed after a RESET.

CANCEL (ID)

Delete synchronous action

PLC

Modal synchronous actions (ID, IDS) can also be locked and unlocked from the PLC program.

Field of influence

The PLC can lock the first 64 modal synchronous actions. The actions that can be locked by the PLC are marked by the NC with 1 in a 64-bit array of the interface DB21...30, DBB308-315.

Lock all synchronous actions

Through the setting of DB21...30, DBB1 Bit 2 by the PLC program, all modal synchronous actions that have already been defined and saved in the NC, are excluded from the activation. By resetting DB 21...30, DBB1 Bit 2, the general lock by the PLC is cancelled.

Specific lock

For the first 64 IDs, one bit each is reserved, DB21...30, DBX300.0 to DBX307.7, in the PLC interface.
If the bits are set, the lock is active. Per default, the actions are unlocked.

Status activation

In order to activate the locking of individual synchronous actions (or their unlocking) by the PLC, bit DB 21...30, DBX280.1 must be set.

Protected synchronous actions

It is possible to protect synchronous actions being cancelled or locked, e.g. by the manufacturer. An area of actions to be protected can be specified in MD11500:

MD11500[0]: First protected synchronous action
MD11500[1]: Last protected synchronous action

Synchronous actions entered there cannot be

- Overwritten
- Deleted or
- Locked

once they have been defined. Protected actions also cannot be locked by the PLC. They are shown in the appropriate DBs as unlockable.

Notes:

Configuration :

Number of synchronous actions

The number of programmable synchronous action blocks depends on the configurable number of synchronous action elements. The number of memory elements of motion-synchronous actions is specified with MD28250.

Example: Three elements are required for the following synchronous action block:

WHENEVER($AA_IM[X]>10.5)OR($A_IN[1]==1)DO

Number of FCTDEF functions

The number of programmable FCTDEF functions (polynomials) per block is configured via MD28252. The default value is 3 for all control types.

Event-driven program call :

With this function, a special user program is started implicitly when certain events occur.

These events can be:

- Part program start (Bit 0)
- Part program end (Bit 1)
- Operator panel reset (Bit 2)
- Control power-up (Bit 3)

For which of these events the program is executed, can be configured in MD 20108. The appropriate bit (see above) must be set.

The executed program must be in the *OEM cycles* directory and be called **PROG_EVENT.SPF**.

The event resulting in the program call can be queried in **$P_PROG_EVENT**:

1 = Part program start
2 = Part program end
3 = Operator panel reset
4 = Control power-up

Notes:

Applicaion examples :

Example 1: SYNFCT

Description: Example of SYNFCT with synchronous actions.
The override is reduced linearly when the actual value falls below the start value and becomes 0 when it reaches the minimum value.

Variables:

R0	Actual value
R1	Start value
R2	Minimum value

Program:

```
%_N_MDA112_MPF
;$PATH=/_N_MPF_DIR
ID=1 DO $R3=($R0-$R2)/($R1-$R2)*100
ID=2 WHENEVER  $R3<0 DO $ac_ovr=0
ID=3 WHENEVER $R3>100 DO $ac_ovr=100
ID=4 WHENEVER ($R3>0)AND($R3<100) DO $ac_ovr=$R3
G1 F100 X1000
X0
M0
M2
```

Example 2:

Description:
The deceleration and jerk for braking are to be increased when a light barrier is triggered.

Variables:

$a_in1	Light barrier: $a_in1
R99	Auxiliary marker (light barrier triggered)
$a_in2	Acknowledgement (continue operation)

Program:

```
%_N_MDA1_MPF
;$PATH=/_N_MPF_DIR
R99=0
SOFT
ID=1 WHENEVER $a_in[1] DO $R99=1
ID=2 WHENEVER $R99>0 DO $ac_pathacc=5 $ac_ovr=0 $ac_pathjerk=1000
ID=3 EVERY $a_in[2] DO $ac_pathacc=0 $ac_pathjerk=0 $r99=0
G0 X0 Y0 Z0
G1 F5000 X1000 Z1000
X0 Z0
```

Notes:

Example 3: Travel to fixed stop

Description:

With a positive edge of input 1, the X1 axis starts to move with F100 (IDS=1). If input 1 now becomes 0, Travel to fixed stop is activated (IDS=2). When the fixed stop is reached, Travel to fixed stop is deselected (IDS=3). If the checkback signal "Axis not at fixed stop" is detected, the X1 axis is set to 0 via preseton.

Variables:

R1 Step marker

Program:

```
IDS=1 EVERY $a_in[1]==1 DO pos[X1]=ic(500) fa[X]=1000 $R1=1
IDS=2 EVERY ($a_in[1]==0) AND ($R1==1) DO fxs[X1]=1 fxst[X1]=10 pos[X1]=ic(200) fa[X1]=200 $R1=2
IDS=3 EVERY $aa_fxs[x1]==1 DO fxs[X1]=0 $R1=3
IDS=4 EVERY (($R1==3) AND ($aa_fxs[X1]==0)) DO preseton(X1,0) $R1=0
JMP1: MSG(<<$aa_fxs[X1]<<"  r: "<<r1)
STOPRE
GOTOB JMP1
```

Example 4: Functionality with ELSE branch

Program

```
ID=1 WHENEVER (ABS($VA_DESVAL_FILTERS_DELTA_POS)<0.01[mm] AND ($AC_G0MODE=1)
DO $AA_DESVAL_FILTERS_SELECT[Axis1] = 1
ELSE $AA_DESVAL_FILTERS_SELECT[Axis1] = 0
```

A305: END

Notes:

A313

SINUMERIK ONE

Memory Configuration

Module Description:

This module explains the configuration of the memory of Sinumerik ONE

Static User Memory

Dynamic User Memory

Module Objective:

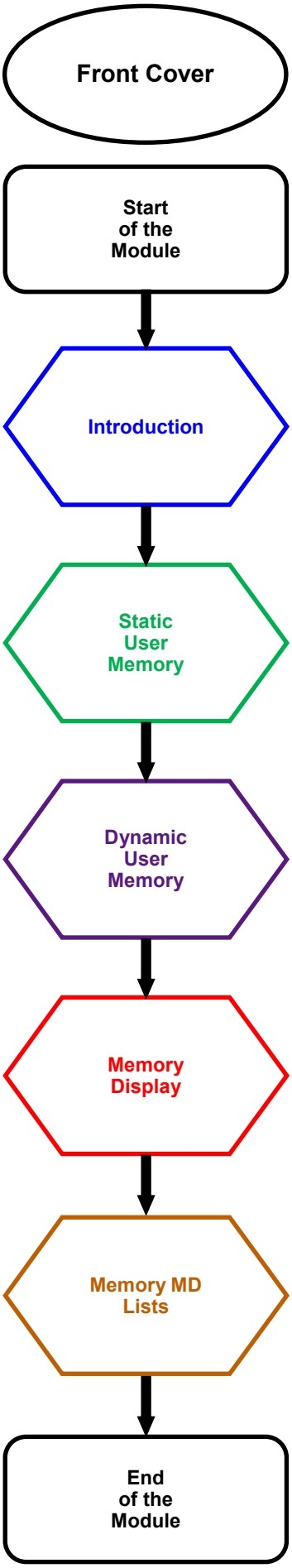
To give a basic understanding of the memory configuration of Sinumerik ONE

Content:

Introduction
Static User Memory
Dynamic User Memory
Memory Display
Memory Machine Data Lists

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Introduction
Static User Memory
Dynamic User Memory
Memory Display
Memory MD Lists
End of the Module

Introduction

Memory types

To store and manage data, the NC requires a static memory and a dynamic memory: The SD card of the NCU contains two memory areas for storing and managing the data of the local persistent and non-persistent data of the NC:

- Static NC memory

In the static NC memory, the program data (part programs, cycles, etc.) and the current system and user data (tool management, global user data, etc.) is saved to persistent memory.

- Dynamic NC memory

In the dynamic NC memory the data is saved to non-persistent memory. The data here concerns information generated by the NC that is only required for a limited time (e.g., macros, local user data, interpolation buffer, etc.).

Memory configuration

The memory areas of the local static and dynamic NC memory are set up when the control system is first started based upon the standard settings of the machine data that configures the memory. These settings are adequate in most cases.

Reconfiguration

If necessary, the memory configuration can be adapted to the user-specific requirements. The necessary reconfiguration of the memory can result in loss of all user data in the static NC memory. Therefore, after a change to memory-configuring machine data, the Alarm 4400: "MD change results in reorganization of the buffered memory (loss of data!)" is displayed. Prior to activating the machine data change or reconfiguration of the memory by an NC reset, a commissioning archive should be created.

The memory configuration will be changed the next time the NC is started. Then the created commissioning archive can be read-in for restoring the user data.

Memory Expansion

The following memory expansions are available as licensed software options:

- Additional CNC user memory in 2MB increments
  (6FC5800-0BD00-0YB0 D001….D009)
  10MB standard, expandable to maximum 28MB
- Additional 100MB user memory on the SD Card of the NCU/PPU (6FC5800-0BP77-0YB0)
- Additional 6 GB HMI/User memory on the SD card of the NCU/PPU (6FC5800-0BP12-0YB0). Note option P77 also required
- Expansion of the PLC user memory by 1.5MB program memory/5 MB data memory (6FC5800-0BD10-0YB0)
  Standard 1.5MB/5MB
  Expandable to PPU1740 3MB/10MB
  NCU1750 6MB/20/MB
  NCU1760 6MB/20MB

Notes:

Static User Memory

Active and Passive file system

The user specific data of the NC are stored in the local static NC memory. The memory area contains the data of the active and the passive file system.

Active file system

The active file system contains system data used to parameterize the NC

- Machine data
- Setting data
- Option data
- Global user data (GUD)
- Tool-offset/magazine data
- Protection zones
- R parameters
- Work offsets/FRAME
- Sag compensations
- Quadrant error compensation
- Leadscrew error compensation

This data represents the current working data of the NC.

Passive file system

The passive file system contains all files loaded onto the NC

- Main programs
- Subprograms
- Global-user-data and macro definition files(.DEF)
- Standard cycles
- User cycles
- Workpieces
- Comments
- Binary files (e.g pictures, PDF documents)

Reconfiguration

The first time the controller starts up, the sizes of the memory areas of the local NC memory are set based on the default settings of the memory-configuring machine data. Generally, these settings are sufficient for the operation of the control. If the memory areas of the local NC memory are adapted because of special user-specific requirements, this results in reconfiguration of the local NC memory. If the active and/or passive file system of the static NC memory is affected, Alarm 4400 is displayed. The user should then back up the relevant data by creating a commissioning archive on an external storage medium.

Alarm 4400 is displayed in the following cases:
● Memory-configuring machine data of the passive file system was changed
● Memory-configuring machine data of the active file system was changed and the function
"Automatic Memory Reconfiguration" (AMR) is not active
● Memory-configuring machine data of the active file system was changed, the AMR function
is active, but the data of the active file system cannot be buffered locally

Automatic memory reconfiguration function (AMR)

Active file system

The AMR function enables the reconfiguration of memory areas of the active file system without having to create and then read back a commissioning archive to prevent the loss of user data.
If the function is active, and memory-configuring machine data that affects the active file system is modified, it is first checked whether all data of the active file system can be buffered locally. If this is the case, the data of the active file system will be first buffered locally on activation of the modified memory configuration by triggering a warm restart. The memory of the active file system is then reconfigured. Finally, the buffered data are read back to the newly configured active file system.
If the data of the active file system cannot be buffered, Alarm 4400 is displayed. Then, before activating the machine data change, a commissioning archive containing the relevant user data should be created.

Passive file system

If machine data that requires a reconfiguration of the passive file system is changed, Alarm 4400 will always be displayed, independent of the AMR function. Then, before activating the machine data change, a commissioning archive containing the relevant user data should be created.
If machine data that only requires a reconfiguration of the active file system is changed, the data of the passive file system is retained.

Activation

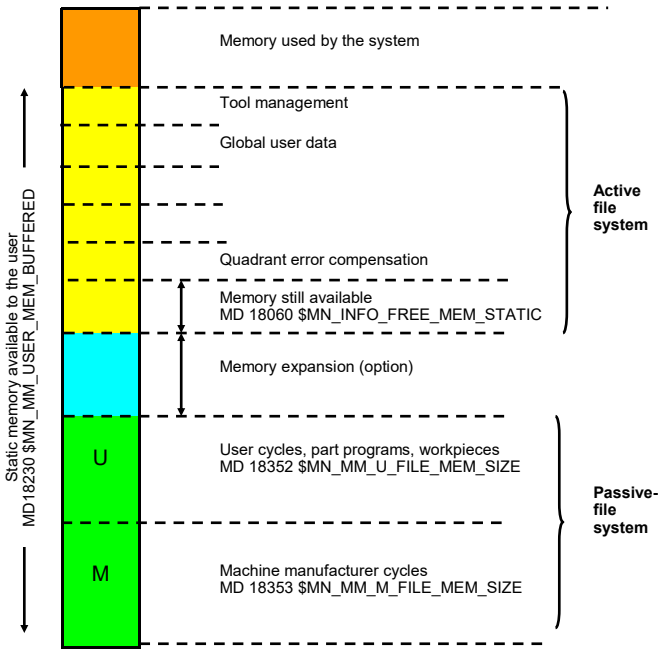
The "Automatic memory reconfiguration" function (AMR) is activated via machine data:
MD17950 $MN_IS_AUTOMATIC_MEM_RECONFIG = TRUE



Notes:

Configuration of the static user memory

Division of the static NC memory
The figure below shows the division of the static NC memory for SINUMERIK ONE:



Static user memory

The static NC memory is used jointly by the system and by the user. The area available to the user is defined as the static user memory. It contains the data from the active and passive file system.

Static-user-memory size

The size of the static user memory is defined in machine data:

MD18230 $MN_MM_USER_MEM_BUFFERED

18230	$MN_MM_USER_MEM_BUFFERED	30653

Memory space for passive file system

The memory space for the passive file system has a defined size and is divided into the following partitions:

Partition	Storage of:
M (Manufacturer = Machine manufacturer)	Files from the _N_CMA_DIR directory (machine manufacturer cycles)
U (User = End customer)	Files from the _N_CUS_DIR directory (User cycles, part programs, workpieces)

Size of the partitions:

You can divide the remaining memory available for the passive file system into the partitions U and M as you see fit. The settings are made with machine data:
MD18352 $MN_MM_U_FILE_MEM_SIZE
(end-user memory for part programs/cycles/files)
MD18353 $MN_MM_M_FILE_MEM_SIZE
(memory size for cycles/files of the machine manufacturer)

The maximum adjustable values depend on:

- The system and thus the memory space available (including an optional memory expansion) from the defined maximum values.

Note:

The settings of the MD18352 and MD18353 depend upon the license Additional CNC user memory in 2MB increments (6FC5800-0BD00-0YB0 D001….D009)

10MB is standard D000, expandable to maximum D009 28MB

Memory space for active file system

The memory space for the active file system is divided into various data areas (tool management, global user data, etc.), which can be defined individually using machine data.
The memory still available is shown in machine data:

MD18060 $MN_INFO_FREE_MEM_STATIC

18060	$MN_INFO_FREE_MEM_STATIC	12583936

You can expand the sizes of the individual memory areas for the active file system with the relevant machine data until the available memory has been used.

NOTE:-

The memory required to expand the memory areas is displayed in the "Startup" area of the user interface. This information enables the system startup engineer to estimate the actual memory requirements for the planned expansion.

Memory expansion (option)

The machine manufacturer can also purchase additional static user memory as an option. You can use the additional memory space as required to expand partitions U and M or to expand the memory area of the active file system.

Notes:

Commissioning Procedure

1. Load standard machine data.

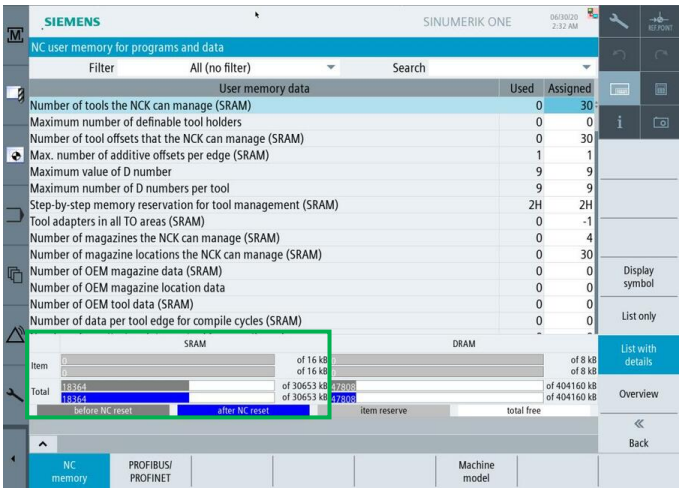
MD18230 $MN_MM_USER_MEM_BUFFERED
Will show the size in KB of the static user memory, in this example 30.6 MB (30653 KB)

18230	$MN_MM_USER_MEM_BUFFERED	30653

MD18060 $MN_INFO_FREE_MEM_STATIC
Will show the static user memory in bytes still available, in this example 12.5 MB (12583936 B)

18060	$MN_INFO_FREE_MEM_STATIC	12583936

It is also possible to see a graphical display of the memory configuration in the "Startup" operator area with “NC\NC Memory\Displays” softkeys, followed by “List with details”.



2. Set the sizes of partitions U and M in machine data if requirement > standard data:

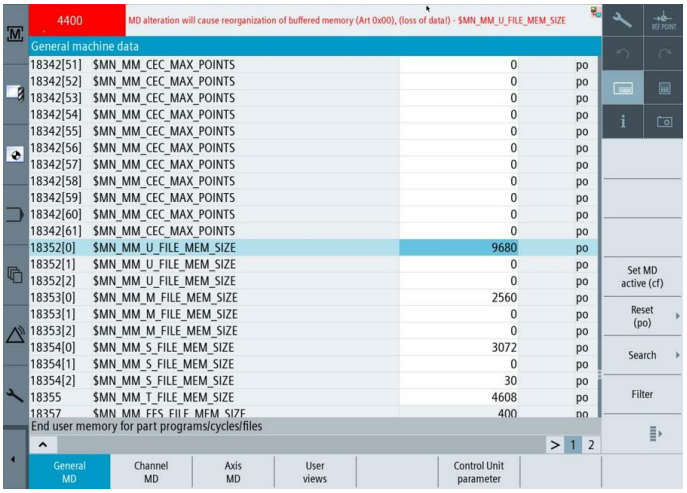
MD18352 $MN_MM_U_FILE_MEM_SIZE
(end-user memory for part programs/cycles/files)
MD18353 $MN_MM_M_FILE_MEM_SIZE
(memory size for cycles/files of the machine manufacturer)

18352[0]	$MN_MM_U_FILE_MEM_SIZE	7680
18352[1]	$MN_MM_U_FILE_MEM_SIZE	0
18352[2]	$MN_MM_U_FILE_MEM_SIZE	0
18353[0]	$MN_MM_M_FILE_MEM_SIZE	2560
18353[1]	$MN_MM_M_FILE_MEM_SIZE	0
18353[2]	$MN_MM_M_FILE_MEM_SIZE	0

This example is showing that 7.68 MB of memory is available for end-user part programs/cycles/files, and 2.56 MB of memory is available for cycles/files of the machine manufacturer.

It is possible to increase these values however it should be taken into consideration that additional memory will also be required later for further commissioning. e.g increasing Tool Offsets, GUDs, etc

So for example if it is decided to increase the memory available for end-user part programs/cycles/files from 7.68 MB to 9.68 MB then MD
MD18352 $MN_MM_U_FILE_MEM_SIZE
Should be increased from 7680 to 9680



A warning is shown, however at this stage because only standard data is loaded this does not matter. After Reset (po) the memory will be reorganised.

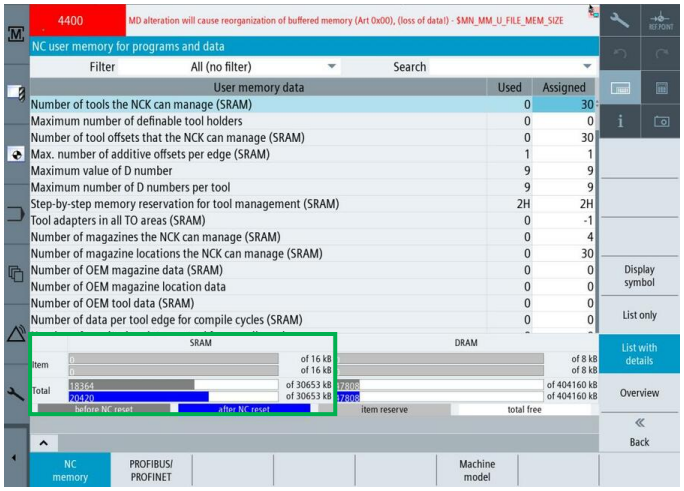
Notes:

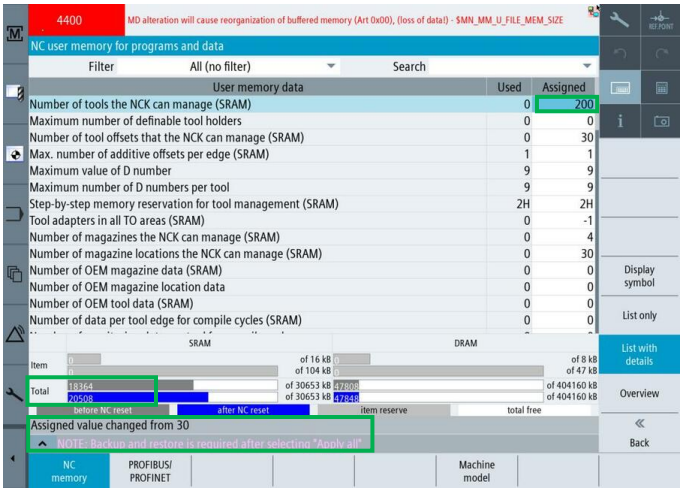
It is possible to see a graphical display of the memory configuration in the "Startup" operator area with “NC\NC Memory\Display” softkeys, followed by “List with details”. Information is shown in the display regarding how much more memory will be used by the new configuration prior to applying the changes. So in the display it can be seen that after NC reset the SRAM memory used will increase by approx 2 MB

3. Activate the number of required channels and axes.

4. You can then adjust the default memory by increasing/ decreasing individual memory areas of the active file system (tool management, global user data, etc.).

So for example if the requirement is to increase the number of tools that the NCK can manage to 200





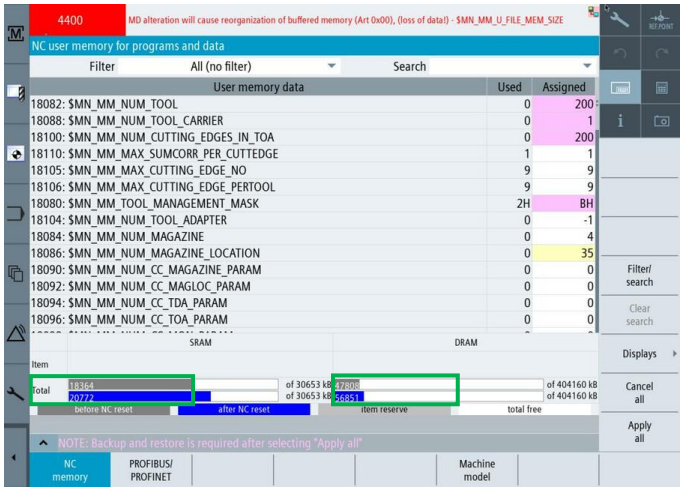
Information is shown in the display regarding how much more memory will be used by the new configuration prior to applying the changes. Also a NOTE: saying that after pressing the “Apply all” softkey then a backup and restore is required to reconfigure the SRAM.

The following table shows an example of memory MD changes from standard that could be considered for a machining centre with “Shopmill”

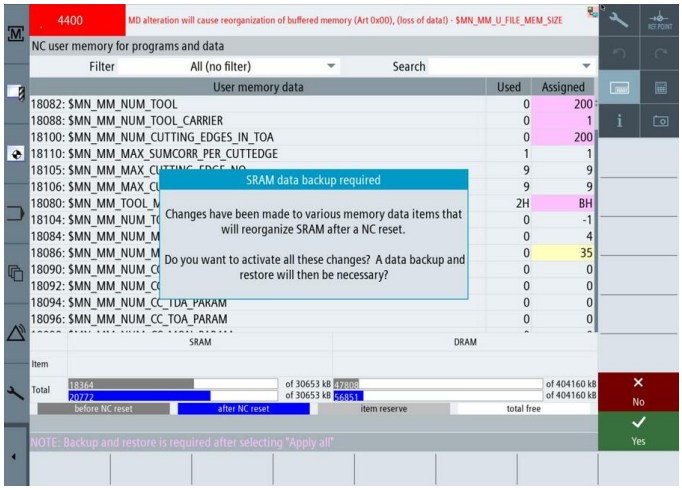
Machine Data	Standard	New
N18080 $MN_MM_TOOL_MANAGEMENT_MASK	0	B
N18082 $MN_MM_NUM_TOOL	30	200
N18086 $MN_MM_NUM_MAGAZINE_LOCATION	30	35
N18088 $MN_MM_NUM_TOOL_CARRIER	0	1
N18100 $MN_MM_NUM_CUTTING_EDGES_IN_TOA	30	200
N18120 $MN_MM_NUM_GUD_NAMES_NCK	60	100
N18360 $MN_MM_EXT_PROG_BUFFER_SIZE	50	500
N18362 $MN_MM_EXT_PROG_NUM	1	2
N28000 $MC_MM_REORG_LOG_FILE_MEM	50	75
N28010 $MC_MM_NUM_REORG_LUD_MODULES	8	20
N28060 $MC_MM_IPO_BUFFER_SIZE	10	150
N28070 $MC_MM_NUM_BLOCKS_IN_PREP	50	80

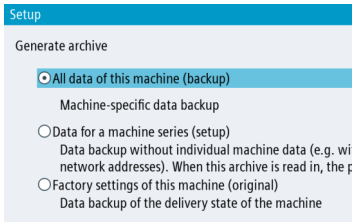
Notes:

5. After applying all of the MD changes, the “Apply all” softkey should be pressed. It is possible to see from the display the effect this will have on the SRAM and DRAM before an NC reset is carried out.



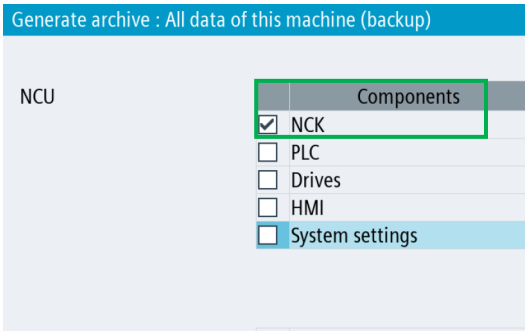
6. After pressing “Apply all” a warning is given that the SRAM will be reorganised, therefore a data backup and restore will be necessary.





After pressing “Yes”, the “Generate archive” screen will be automatically displayed.

A reconfiguration of the static NC memory results in a loss of data in the active and passive file system. Before activating the modified memory configuration, you must first save the data by creating a machine-specific data backup and reading it back into the controller.



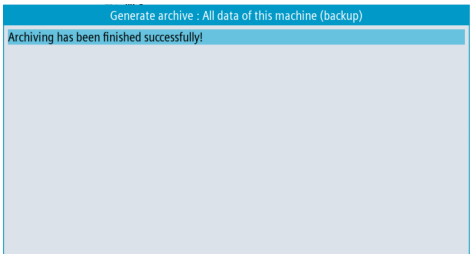
The five control components are available for selection. They can be selected in any combination to create archives containing different data.
At this stage of commissioning we are concerned with the NC memory data therefore the NCK component can be selected.

The destination of the archive has to be chosen, this can be locally on the system SD card/hard disk drive or externally to a USB device or networked PC.
In this example the manufacturer archives directory is selected and the standard file name is used

Storage path	Archives/Manufacturer
Archive name	Backup_NL_2020-06-30

Notes:

After entering the archive name and pressing the “OK” softkey the procedure will start.
An information box indicates that the archive was successfully created.



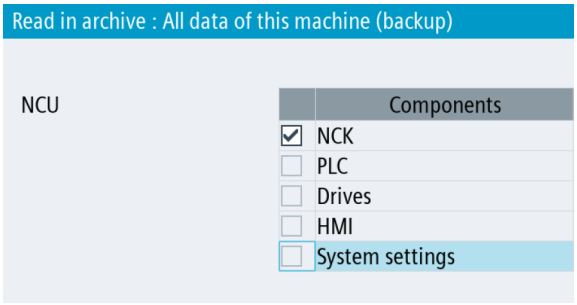
To restore the archive the option ”Read in setup archive”
should be selected
needs to be selected



Select the source from which the archive is to be read in from and press the “OK” softkey

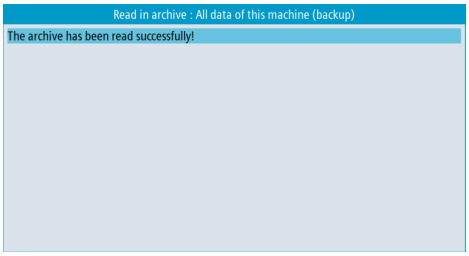


Information about the selected archive is displayed. This includes any comments entered by the creator.
The software version of the system that the archive was created on is also displayed.



After pressing the “OK” softkey the archive will be read in and a series of status boxes are displayed.

A final status box will be displayed which confirms that the archive was read in successfully.



The static memory has now been reorganised to suit the memory requirements of the different areas.

7. After reloading the setup archive the new memory configuration can be seen

MD18230 $MN_MM_USER_MEM_BUFFERED
Will show the size of the static user memory, in this example 30.6 MB (30653 KB) which is unchanged.



MD18060 $MN_INFO_FREE_MEM_STATIC
Will show the static user memory still available, in this example 10.1 MB (10118144 B)

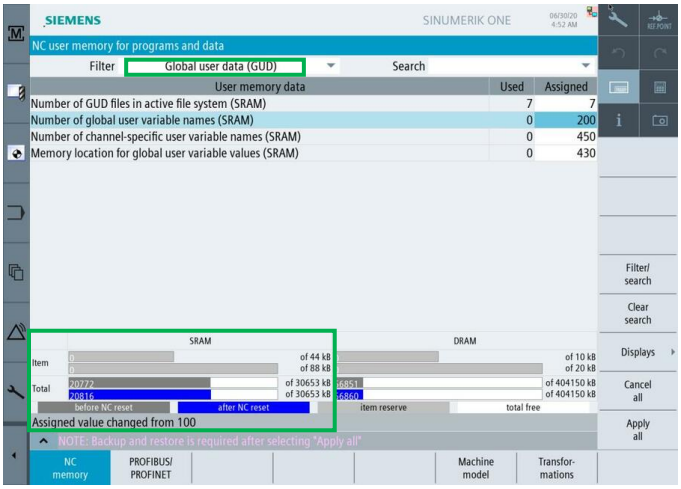


Notes:

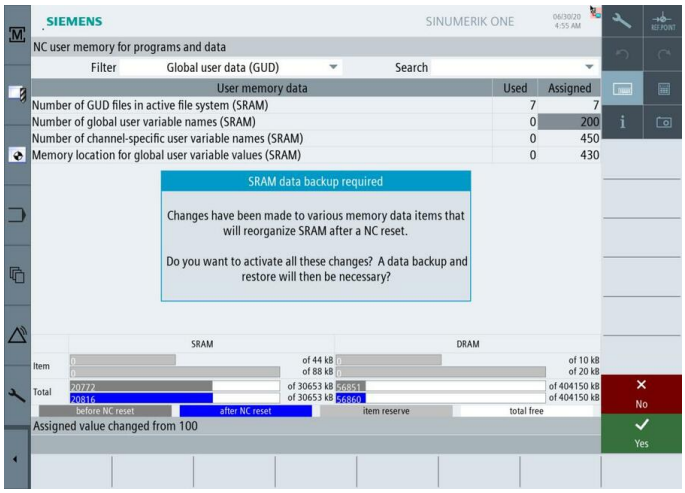
Modifying a single SRAM machine data

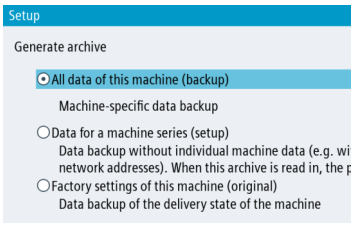
Example: Increasing the number of NCK GUDs from 100 to 200

It is possible to see a graphical display of the memory configuration in the "Startup" operator area with “NC\NC Memory” softkeys, followed by “List with details”. Then the “filter” toggle box can be used to find the MD required



It is possible to see from the display the effect this will have on the SRAM and DRAM before an NC reset is carried out. After pressing “Apply all” a warning is given that the SRAM will be reorganised, therefore a data backup and restore will be necessary.





After pressing “Yes”, the “Create setup archive” screen will be automatically displayed.

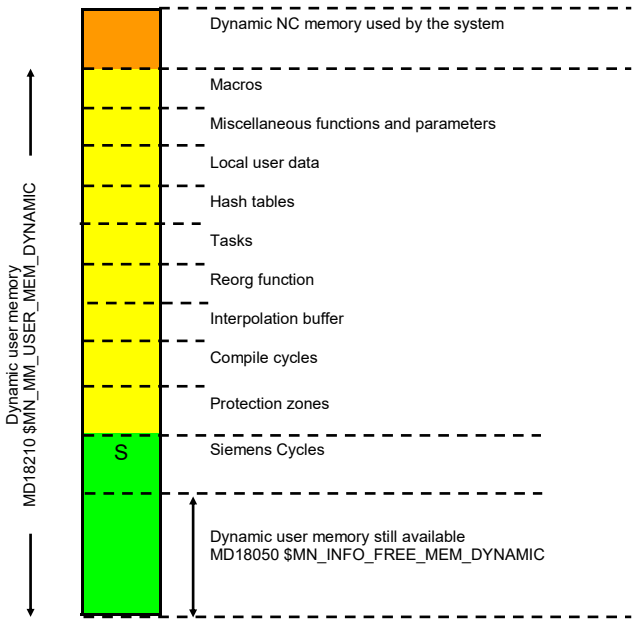
After saving and restoring the setup archive as per the procedure on pages 9 and 10 the extra GUD memory will be available.

Notes:

Dynamic User memory

Configuration of the dynamic user memory

Division of the dynamic NC memory
The figure below shows the division of the dynamic NC memory:



Dynamic user memory

The dynamic NC memory is used jointly by the system and by the user. The area available to the user is defined as the dynamic user memory.

Dynamic-user-memory size

MD18210 $MN_MM_USER_MEM_DYNAMIC

Will show the size in KB of the dynamic user memory, in this example 404 MB (404150 KB).

18210	$MN_MM_USER_MEM_DYNAMIC	404150

Parts of the Dynamic user memory

Passive file system:

Partition S of the passive file system is in the dunamic user memory

Partition	Storage of:
S (Siemens = Control manufacturer)	Files from the _N_CST_DIR directory (Siemens cycles)

The size of the partition S is preset and cannot be modified

User Data Areas

The Dynamic user memory contains the user data areas for saving the non persistent NC data dynamically generated by the NC, e.g. macros, local user data, buffer etc.

Dynamic user memory still available

MD18050 $MN_INFO_FREE_MEM_DYNAMIC

The content of this machine data indicates how much memory is available in bytes to expand the user data areas (local user data, IPO buffer, etc.) for each channel. In this example 355 MB (355625099 B)

18050	$MN_INFO_FREE_MEM_DYNAMIC	355625099

Commissioning

You can adjust the default memory by increasing/decreasing individual memory areas.

Procedure

1. The dynamic user memory still available is displayed in machine data:

MD18050 $MN_INFO_FREE_MEM_DYNAMIC

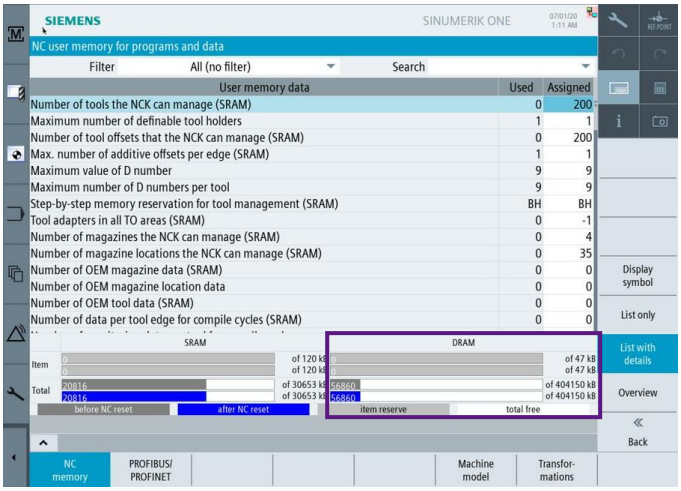
18050	$MN_INFO_FREE_MEM_DYNAMIC	355625099

2. Set the relevant machine data , e.g. N28060 $MC_MM_IPO_BUFFER_SIZE

3. Reset the NCK.

The memory is set up again.

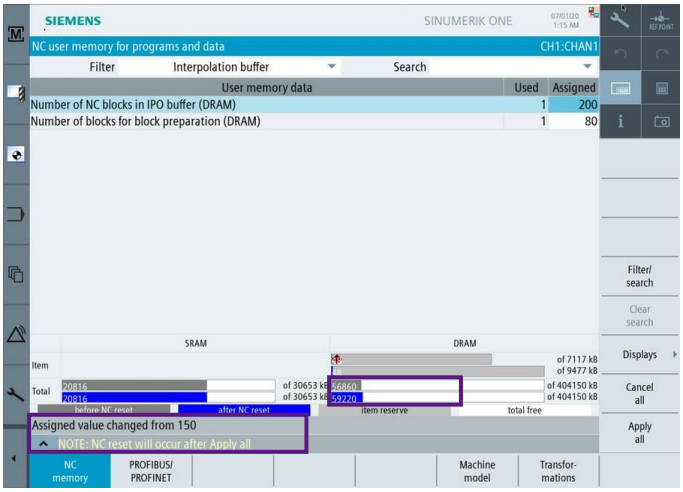
It is also possible to see a graphical display of the memory configuration in the "Startup" operator area with “NC\NC Memory” softkeys, followed by “List with details”.

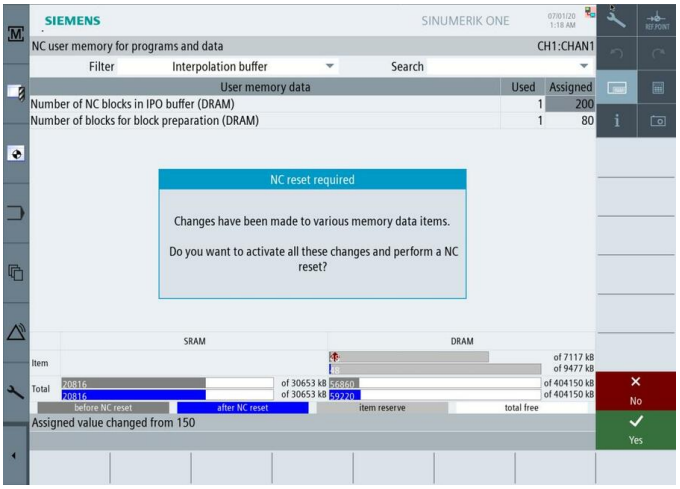


Modifying a single DRAM machine data

Example: Increasing the IPO Buffer from 150 to 200

Notes:



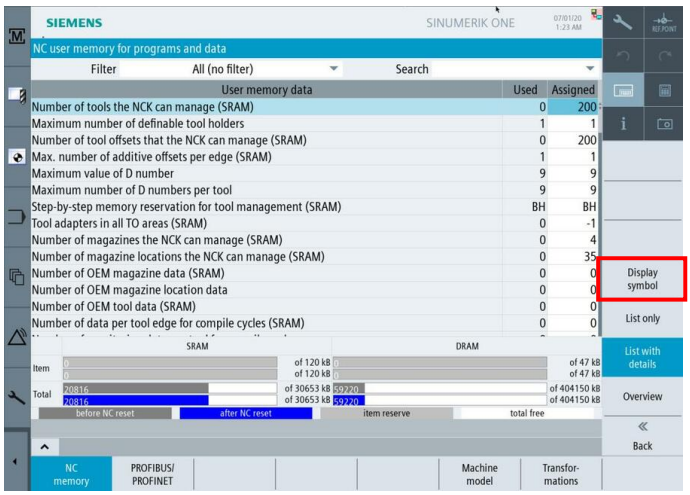


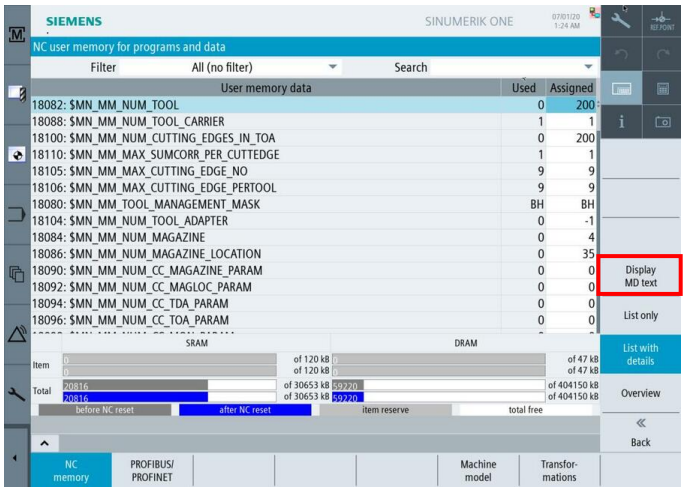
The “filter” toggle box can be used to find the MD required. After applying the MD change, the “Apply all” softkey should be pressed. It is possible to see from the display the effect this will have on the DRAM before an NC reset is carried out. After pressing “Apply all” a warning is given that a NC reset will be necessary to activate the changes.

Notes:

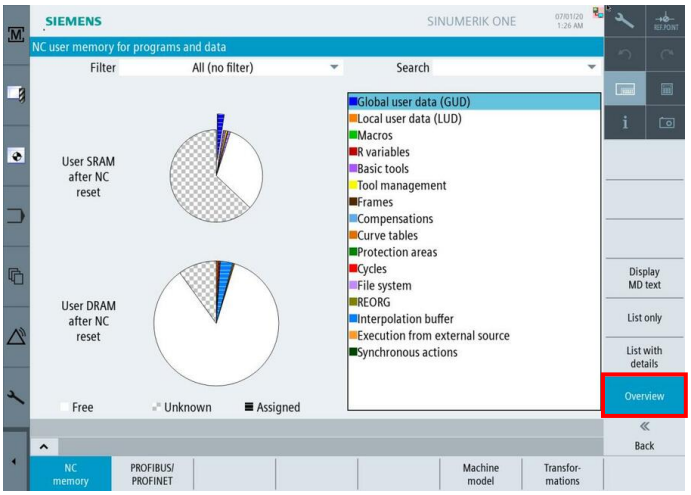
Memory Display

The Static and Dynamic user memory can be displayed in the "Startup" operator area with “NC\NC Memory\Displays” soft-keys, followed by “List only”, “List with details”, “Overview”. Reconfiguration of static and dynamic memory can also be carried out in this area. The advantage of using this area is that it can be seen in advance the effect on the memory prior to making the changes.

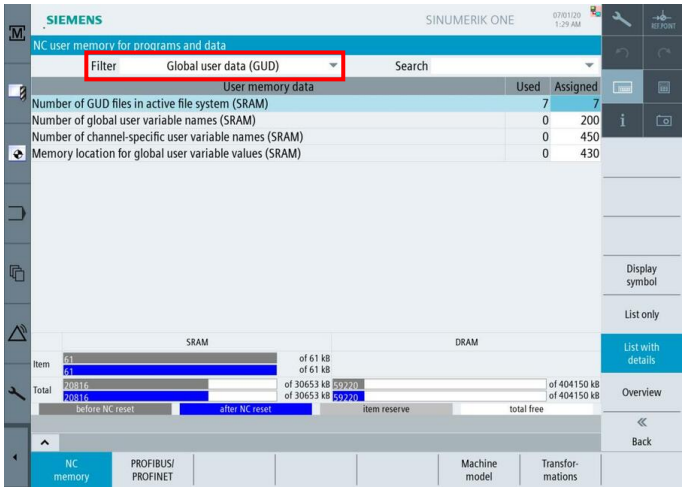




Using the “Display symbol” or “Display MD text” it is possible to switch the display between the MD and the description.



Using the “Overview” softkey it is possible to see a graphical representation of each individual memory area.



It is also possible to use the “Filter” to search for machine data in different areas.

Notes:

Memory General Machine Data

Number	Identifier: $MN_	Description
10134	MM_NUM_MMC_UNITS	Number of simultaneous HMI communication partners
10850	MM_EXTERN_MAXNUM_OEM_GCODES	Maximum number of OEM-G codes
10880	MM_EXTERN_CNC_SYSTEM	Definition of the control system to be adapted
10881	MM_EXTERN_GCODE_SYSTEM ISO_3	Mode: GCodeSystem
18050	INFO_FREE_MEM_DYNAMIC	Free-dynamic-memory display data
18060	INFO_FREE_MEM_STATIC	Free-static-memory display data
18070	INFO_FREE_MEM_DPR	Display data for free memory in dual-port RAM
18072	INFO_FREE_MEM_CC_MD	Displays memory available in the CC-MD memory
18078	MM_MAX_NUM_OF_HIERARCHIES	Maximum number of definable magazine-location-type hierarchies
18079	MM_MAX_HIERARCHY_ENTRIES	Maximum permissible number of entries in a magazinelocation- type hierarchy
18080	MM_TOOL_MANAGEMENT_MASK	Mask for reserving memory for tool management
18082	MM_NUM_TOOL	Number of tools managed by NCK
18084	MM_NUM_MAGAZINE	Number of magazines managed by NCK
18086	MM_NUM_MAGAZINE_LOCATION	Number of magazine locations
18088	MM_NUM_TOOL_CARRIER	Maximum number of definable toolholders
18090	MM_NUM_CC_MAGAZINE_PARAM	Compile-cycle tool management: quantity of magazine data
18092	MM_NUM_CC_MAGLOC_PARAM	Compile-cycle tool management: quantity of magazine-location data
18094	MM_NUM_CC_TDA_PARAM	Compile-cycle tool management: quantity of TDA data
18096	MM_NUM_CC_TOA_PARAM	Compile-cycle tool management: quantity of TOA data
18098	MM_NUM_CC_MON_PARAM	Compile-cycle tool management: quantity of monitor data
18100	MM_NUM_CUTTING_EDGES_IN_TOA	Number of tool offsets in NCK
18102	MM_TYPE_OF_CUTTING_EDGE	Type of D-number programming
18104	MM_NUM_TOOL_ADAPTER	Tool adapter in TO area
18105	MM_MAX_CUTTING_EDGE_NO	Maximum value of D number
18106	MM_MAX_CUTTING_EDGE_PERTOOL	Maximum number of D numbers per tool
18108	MM_NUM_SUMCORR	Additive offsets in the TO area
18110	MM_MAX_SUMCORR_PER_CUTTEDGE	Maximum number of sum offsets per cutting edge
18112	MM_KIND_OF_SUMCORR	Properties of additive offsets in the TO area
18114	MM_ENABLE_TOOL_ORIENT	Assign orientation to cutting edges
18116	MM_NUM_TOOL_ENV	Number of tool environments in TO area
18118	MM_NUM_GUD_MODULES	Number of GUD modules
18120	MM_NUM_GUD_NAMES_NCK	Number of global user variables
18130	MM_NUM_GUD_NAMES_CHAN	Number of channel-specific user variables
18150	MM_GUD_VALUES_MEM	Memory space for global user variables
18160	MM_NUM_USER_MACROS	Number of macros
18170	MM_NUM_MAX_FUNC_NAMES	Number of miscellaneous functions
18180	MM_NUM_MAX_FUNC_PARAM	Number of additional parameters
18190	MM_NUM_PROTECT_AREA_NCK	Number of protection zones in NCK
18200	MM_NUM_CCS_MAGAZINE_PARAM	Number of Siemens OEM magazine data
18201	MM_TYPE_CCS_MAGAZINE_PARAM	Siemens OEM magazine data type
18202	MM_NUM_CCS_MAGLOC_PARAM	Number of Siemens OEM magazine location data
18203	MM_TYPE_CCS_MAGLOC_PARAM	Siemens OEM magazine location data type
18204	MM_NUM_CCS_TDA_PARAM	Number of Siemens OEM tool data
18205	MM_TYPE_CCS_TDA_PARAM	Siemens OEM tool data type
18206	MM_NUM_CCS_TOA_PARAM	Number of Siemens OEM data per cutting edge
18207	MM_TYPE_CCS_TOA_PARAM	Siemens OEM data type per cutting edge
18208	MM_NUM_CCS_MON_PARAM	Number of Siemens OEM monitor data
18209	MM_TYPE_CCS_MON_PARAM	Siemens OEM monitor data type
18210	MM_USER_MEM_DYNAMIC	User memory in DRAM
18220	MM_USER_MEM_DPR	User memory in dual-port RAM
18230	MM_USER_MEM_BUFFERED	User memory in SRAM
18231	MM_USER_MEM_BUFFERED_TYPEOF	Data-buffer technology
18232	MM_ACTFILESYS_LOG_FILE_MEM	System: Log-file size
18238	MM_CC_MD_MEM_SIZE	Machine-data compile cycles in SRAM
18240	MM_LUD_HASH_TABLE_SIZE	Hash-table size for user variables
18242	MM_MAX_SIZE_OF_LUD_VALUE	Maximum LUD-variable array size
18250	MM_CHAN_HASH_TABLE_SIZE	Hash-table size for channel-specific data
18260	MM_NCK_HASH_TABLE_SIZE	Hash-table size for global data
18270	MM_NUM_SUBDIR_PER_DIR	Number of subdirectories
18280	MM_NUM_FILES_PER_DIR	Number of files per directory
18290	MM_FILE_HASH_TABLE_SIZE	Hash-table size for files in a directory

Notes:

18300	MM_DIR_HASH_TABLE_SIZE	Hash-table size for subdirectories
18310	MM_NUM_DIR_IN_FILESYSTEM	Number of directories in passive file system
18320	MM_NUM_FILES_IN_FILESYSTEM	Number of files in passive file system
18332	MM_FLASH_FILE_SYSTEM_SIZE	Size of flash file system on PCNC
18342	MM_USER_FILE_MEM_MINIMUM	Minimum part-program memory
18352	MM_U_FILE_MEM_SIZE	End-user memory for part programs/cycles/files
18353	MM_M_FILE_MEM_SIZE	Memory size for cycles/files of the machine manufacturer
18354	MM_S_FILE_MEM_SIZE	Memory size for cycles/files of the NC manufacturer
18355	MM_T_FILE_MEM_SIZE	Memory size for temporary files
18356	MM_E_FILE_MEM_SIZE	Memory size for external files
18360	MM_EXT_PROG_BUFFER_SIZE FIFO	Buffer size for execution from external source (DRAM)
18362	MM_EXT_PROG_NUM	Number of program levels that can be processed simultaneously from external
18370	MM_PROTOC_NUM_FILES	Maximum number of log files
18371	MM_PROTOC_NUM_ETPD_STD_LIST	Number of standard ETPD data lists
18372	MM_PROTOC_NUM_ETPD_OEM_LIST	Number of ETPD OEM data lists
18373	MM_PROTOC_NUM_SERVO_DATA	Number of servo data for log
18374	MM_PROTOC_FILE_BUFFER_SIZE	Log-file buffer size
18375	MM_PROTOC_SESS_ENAB_USER	User-enabled for sessions
18390	MM_COM_COMPRESS_METHOD	Supported compression method
18400	MM_NUM_CURVE_TABS	Number of curve tables (SRAM)
18402	MM_NUM_CURVE_SEGMENTS	Number of curve segments (SRAM)
18403	MM_NUM_CURVE_SEG_LIN	Number of linear curve segments (SRAM)
18404	MM_NUM_CURVE_POLYNOMS	Number of curve table polynomials (SRAM)
18406	MM_NUM_CURVE_TABS_DRAM	Number of curve tables (DRAM)
18408	MM_NUM_CURVE_SEGMENTS_DRAM	Number of curve segments (DRAM)
18409	MM_NUM_CURVE_SEG_LIN_DRAM	Number of linear curve segments (DRAM)
18410	MM_NUM_CURVE_POLYNOMS_DRAM	Number of curve table polynomials (DRAM)
18450	MM_NUM_CP_MODULES	Maximum number of CP modules
18452	MM_NUM_CP_MODUL_LEAD	Maximum number of leading values per CP coupling module
18500	MM_EXTCOM_TASK_STACK_SIZE	Stack size of external communication task (DRAM)
18502	MM_COM_TASK_STACK_SIZE	Stack size in Kbytes of communication task (DRAM)
18510	MM_SERVO_TASK_STACK_SIZE	Stack size of servo task (DRAM)

18512	MM_IPO_TASK_STACK_SIZE	Stack size of IPO task (DRAM)
18520	MM_DRIVE_TASK_STACK_SIZE	Stack size of drive task (DRAM)
18540	MM_PLC_TASK_STACK_SIZE	Stack size of PLC task (DRAM)
18600	MM_FRAME_FINE_TRANS	Fine offset for FRAME (SRAM)
18601	MM_NUM_GLOBAL_USER_FRAMES	Number of globally predefined user frames (SRAM)
18602	MM_NUM_GLOBAL_BASE_FRAMES	Number of global basic frames (SRAM)
18660	MM_NUM_SYNACT_GUD_REAL	Number of configurable real-type GUD variables
18661	MM_NUM_SYNACT_GUD_INT	Number of configurable integer-type GUD variables
18662	MM_NUM_SYNACT_GUD_BOOL	Number of configurable Boolean-type GUD variables
18663	MM_NUM_SYNACT_GUD_AXIS	Number of configurable axis-type GUD variables
18664	MM_NUM_SYNACT_GUD_CHAR	Configurable char-type GUD variable
18665	MM_NUM_SYNACT_GUD_STRING	Configurable STRING-type GUD variable
18700	MM_SIZEOF_LINKVAR_DATA	Size of the NCU link variable memory
18710	MM_NUM_AN_TIMER	Number of global time variables for synchronized actions
18720	MM_SERVO_FIFO_SIZE	Setpoint for buffer size between IPO and closed-loop position control
18780	MM_NCU_LINK_MASK	Activation of NCU link communication
18781	NCU_LINK_CONNECTIONS	Number of internal link connections
18782	MM_LINK_NUM_OF_MODULES	Number of NCU link modules
18790	MM_MAX_TRACE_LINK_POINTS	Size of trace data buffer for NCU link
18792	MM_TRACE_LINK_DATA_FUNCTION	Specifies the contents of NCU link files
18794	MM_TRACE_VDI_SIGNAL	Trace specification of VDI signals
18800	MM_EXTERN_LANGUAGE	Activation of external NC languages
18860	MM_MAINTENANCE_MON	Activate recording of maintenance data
18870	MM_MAXNUM_KIN_CHAINS	Maximum number of trains
18880	MM_MAXNUM_KIN_CHAIN_ELEM	Maximum number of train elements
18890	MM_MAXNUM_3D_PROT_AREAS	Maximum number of elements in 3D protection zones
18892	MM_MAXNUM_3D_PROT_AREA_ELEM	Maximum number of protection-zone elements
18894	MM_MAXNUM_3D_PROT_GROUPS	Maximum number of protection-zone groups
18896	MM_MAXNUM_3D_COLLISION	Maximum number of temporary memory locations for collision check

Notes:

Memory Channel specific Machine Data

Number	Identifier: $MC_	Description
20096	T_M_ADDRESS_EXIT_SPINO	Spindle number as address extension
27900	REORG_LOG_LIMIT	Percentage of IPO buffer for log-file enable
28000	MM_REORG_LOG_FILE_MEM	Memory size for REORG
28010	MM_NUM_REORG_LUD_MODULES	Number of modules for local user variables with REORG
28020	MM_NUM_LUD_NAMES_TOTAL	Number of local user variables
28040	MM_LUD_VALUES_MEM	Memory size for local user variables
28050	MM_NUM_R_PARAM	Number of channel-specific R parameters
28060	MM_IPO_BUFFER_SIZE	Number of NC blocks in the IPO buffer
28070	MM_NUM_BLOCKS_IN_PREP	Number of blocks for block preparation
28080	MM_NUM_USER_FRAMES	Number of settable frames
28081	MM_NUM_BASE_FRAMES	Number of basic frames (SRAM)
28082	MM_SYSTEM_FRAME_MASK	System frames (SRAM)
28083	MM_SYSTEM_DATAFRAME_MASK	System frames (SRAM)
28085	MM_LINK_TOA_UNIT	Allocation of a TO unit to a channel
28090	MM_NUM_CC_BLOCK_ELEMENTS	Number of block elements for compile cycles
28100	MM_NUM_CC_BLOCK_USER_MEM	Size of block memory for compile cycles
28105	MM_NUM_CC_HEAP_MEM	Heap memory in KB for compile cycle applications (DRAM)
28150	MM_NUM_VDIVAR_ELEMENTS	Number of elements for writing PLC variables
28160	MM_NUM_LINKVAR_ELEMENTS	Number of write elements for the NCU link variables
28180	MM_MAX_TRACE_DATAPOINTS	Size of trace data buffer
28200	MM_NUM_PROTECT_AREA_CHAN	Number of modules for channel-specific protection zones
28210	MM_NUM_PROTECT_AREA_ACTIVE	Number of simultaneously active protection zones
28212	MM_NUM_PROTECT_AREA_CONTOUR	Elements for active protection zones (DRAM)
28250	MM_NUM_SYNC_ELEMENTS	Number of elements for expressions in synchronized actions
28252	MM_NUM_FCTDEF_ELEMENTS	Number of FCTDEF elements
28254	MM_NUM_AC_PARAM	Dimension of $AC_PARAM.
28255	MM_BUFFERED_AC_PARAM	$AC_PARAM[ ] is saved in SRAM.
28256	MM_NUM_AC_MARKER	Dimension of $AC_MARKER
28257	MM_BUFFERED_AC_MARKER	$AC_MARKER[ ] is saved in SRAM.
28258	MM_NUM_AC_TIMER	Number of $AC_TIMER time variables (DRAM)
28274	MM_NUM_AC_SYSTEM_PARAM	Number of $AC_SYSTEM_ PARAM for motionsynchronous actions
28276	MM_NUM_AC_SYSTEM_MARKER	Number of $AC_SYSTEM_ MARKER for motionsynchronous actions
28290	MM_SHAPED_TOOLS_ENABLE	Enable tool-radius compensation for contour tools
28300	MM_PROTOC_USER_ACTIVE	Activate logging for a user
28301	MM_PROTOC_NUM_ETP_OEM_TYP	Number of ETP OEM event types
28302	MM_PROTOC_NUM_ETP_STD_TYP	Number of ETP standard event types
28400	MM_ABSBLOCK	Activate block display with absolute values
28402	MM_ABSBLOCK_BUFFER_CONF	Dimension size of upload buffer
28450	MM_TOOL_DATA_CHG_BUFF_SIZE	Buffer for tool data changes (DRAM)
28500	MM_PREP_TASK_STACK_SIZE	Stack size of preparatory task
28520	MM_MAX_AXISPOLY_PER_BLOCK	Maximum number of axis polynomials per block
28530	MM_PATH_VELO_SEGMENTS	Number of memory chips for limiting the tool-path velocity
28535	MM_FEED_PROFILE_SEGMENTS	Number of memory chips for feed profiles
28540	MM_ARCLENGTH_SEGMENTS	Number of memory chips for displaying the arc length function
28560	MM_SEARCH_RUN_RESTORE_MODE	Restore data after a simulation
28580	MM_ORIPATH_CONFIG	Setting for ORIPATH tool orientation trajectory referred to path

Memory Axis/Spindle specific Machine Data

Number	Identifier: $MA_	Description
38000	MM_ENC_COMP_MAX_POINTS	Number of intermediate points with interpolatory compensation
38010	MM_QEC_MAX_POINTS	Number of values for quadrant-error compensati
38120	MM_CPREC_FIR_POINTS	Number of points for characteristic approximation of FIR filters for CPRECON

Notes:

A313: END

Notes:

A407

SINUMERIK ONE

Indexing Axis

Module Description:

In certain applications, an axis is only required to approach specific positions (e.g. location numbers). It is necessary to approach the defined positions, the indexing positions, both in AUTOMATIC and Jog mode.

The relevant axes are called "indexing axes". The positions defined for the indexing axes are known as "coded positions" or "indexing positions".

A servo turret is a typical application for an Indexing Axis.

Module Objective:

At the end of this module you will be able to commission and program an indexing axis both from the NC and the PLC

Content:

Introduction

Commissioning Indexing Axes

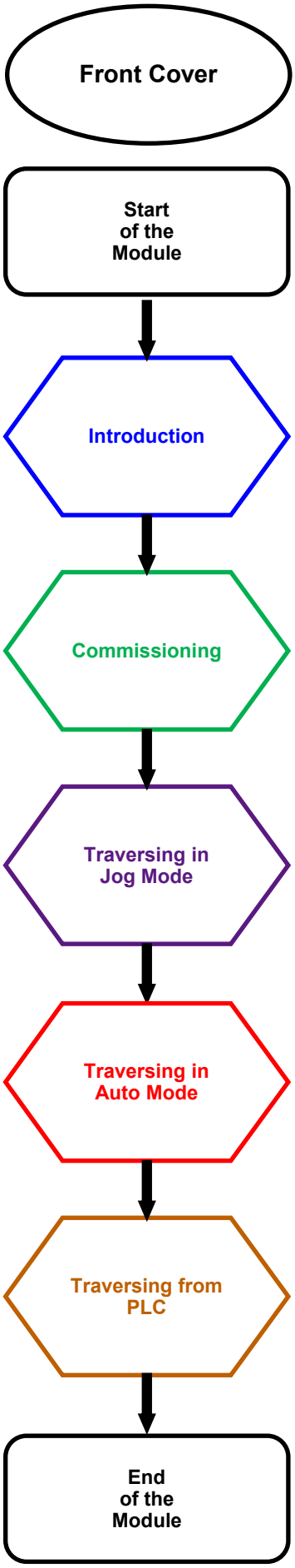
Traversing in Jog Mode

Traversing in Auto Mode

Traversing from PLC

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Introduction
Commissioning
Traversing in Jog Mode
Traversing in Auto Mode
Traversing from PLC
End of the Module

Indexing axes in machine tools

In certain applications, the axis is only required to approach specific grid points (e.g. location numbers). It is necessary to approach the defined grid points, the indexing positions, both in AUTOMATIC and set-up mode.
The relevant axes are called "indexing axes". The positions defined on the indexing axes are known as "coded positions" or "indexing positions".

Applications

Indexing axes are used predominantly in connection with specific types of tool magazines such as tool turrets, tool chain magazines or tool cartridge magazines. The coded positions refer to the individual locations of the tools in the magazines. During a tool change, the magazine is positioned at the location containing the tool to be loaded.

Notes:

Commissioning of an Indexing Axis

Procedure

The procedure for starting up indexing axes is identical to normal NC axes (linear and rotary axes).

Rotary axis

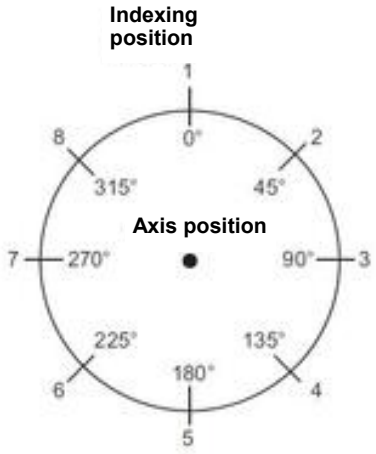
If the indexing axis is defined as a rotary axis (MD30300 $MA_IS_ROT_AX = "1") with modulo 360° conversion (MD30310 $MA_ROT_IS_MODULO = "1"), indexing positions are also approached with modulo 360°. Only positions within the range from 0° to 359.999° can then be entered in the indexing position table. Otherwise alarm 4080 "Incorrect configuration for indexing axis in MD [Name]" is output during power -up.
The position display can be set to modulo 360° as follows:
MD30320 $MA_DISPLAY_IS_MODULO = 1

Example

The assignment of the above machine data is described in the following example.

Indexing axis as rotary axis

Tool turret with 8 locations. The tool turret is defined as a continuously rotating rotary axis. The distances between the 8 turret locations are constant. The first turret location is located at position 0°:



The number of required indexing positions should be entered into MD10900 $MN_INDEX_AX_LENGTH_POS_TAB1

10900	$MN_INDEX_AX_LENGTH_POS_TAB_1	8

The indexing positions for the tool turret are then entered into indexing position table 1, from MD10910[0] to MD10910[7]

10910[0]	$MN_INDEX_AX_POS_TAB_1	0 mm|deg...
10910[1]	$MN_INDEX_AX_POS_TAB_1	45 mm|deg...
10910[2]	$MN_INDEX_AX_POS_TAB_1	90 mm|deg...
10910[3]	$MN_INDEX_AX_POS_TAB_1	135 mm|deg...
10910[4]	$MN_INDEX_AX_POS_TAB_1	180 mm|deg...
10910[5]	$MN_INDEX_AX_POS_TAB_1	225 mm|deg...
10910[6]	$MN_INDEX_AX_POS_TAB_1	270 mm|deg...
10910[7]	$MN_INDEX_AX_POS_TAB_1	315 mm|deg...

In this case Axis 4 is defined as an indexing Axis with Indexing positions in table 1 via axis MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB = 1

30500	$MA_INDEX_AX_ASSIGN_POS_TAB	1

Axis 4 is declared a rotary axis vis axis MD 30300 $_MA_IS_ROT_AX = 1 and modulo conversion is activated via axis MD30310 $MA_ROT_IS_MODULO = 1

30300	$MA_IS_ROT_AX	1
30310	$MA_ROT_IS_MODULO	1

Normally there will also be a ratio between the motor and the rotary axis. In this example 54 revolutions of the motor corresponds to 1 revolution of the turret (360°).

31050[0]	$MA_DRIVE_AX_RATIO_DENOM	1
31050[1]	$MA_DRIVE_AX_RATIO_DENOM	1
31060[0]	$MA_DRIVE_AX_RATIO_NUMERA	54
31060[1]	$MA_DRIVE_AX_RATIO_NUMERA	54

Notes:

Traversing of indexing axes

Indexing axes can be traversed:

- Manually in the setting-up modes JOG and INC
- From a part program with special instructions for coded positions
- From the PLC

Indexing Axis movement in JOG Mode

♦ Jog mode active:

SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD = 1



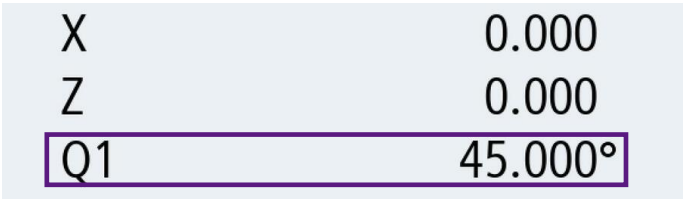
Pressing a "+" or "-" traversing key causes the indexing axis to move in the same way as with conventional JOG INC traversing. When the traversing key is released, the indexing axis traverses to the next indexing position in the direction of traversing.

- Continuous mode active:

SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD = 0

Pressing the traversing key briefly (first rising signal edge) starts the traversing movement of the indexing axis in the desired direction. Traversing continues when the traversing key is released. When the traversing key is pressed again (second rising signal edge), the indexing axis traverses to the next indexing position in the direction of traversing.

Indexing axes are generally traversed in JOG mode (standard setting). Continuous mode plays a less important role.







With
SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD = 1
Every one shot of the axis "+" traverse key moves the axis 45°. If the traversing key is pressed the turret will rotate, and once the traversing key is released the turret will keep rotating until it reaches the next position.

Notes:

Traversing of indexing axes in AUTOMATIC mode

Traversal to selected positions

An axis defined as an indexing axis can be made to approach any selected position from the NC part program in AUTOMATIC mode. This includes positions between the defined indexing positions.

These positions are programmed, in the usual way, in the unit of measurement (mm/inches or degrees) for the axis. The general programming instructions used for this purpose (G90, G91, AC, IC, etc.) are described in the Programming Manuals.

Traversal to "Coded positions"

Special instructions can also be used in the part program to traverse indexing axes to the "coded positions":

Program Code	Effect
CAC	Approach absolute coded position
CACP	Approach absolute coded position in positive direction
CACN	Approach absolute coded position in negative direction
CIC	Approach incremental coded position
CDC	Approach coded position via shortest path

With absolute positioning, the indexing position to be approached is programmed, and with incremental positioning, the number of indexes to be traversed in the "+" or "-" direction is programmed.

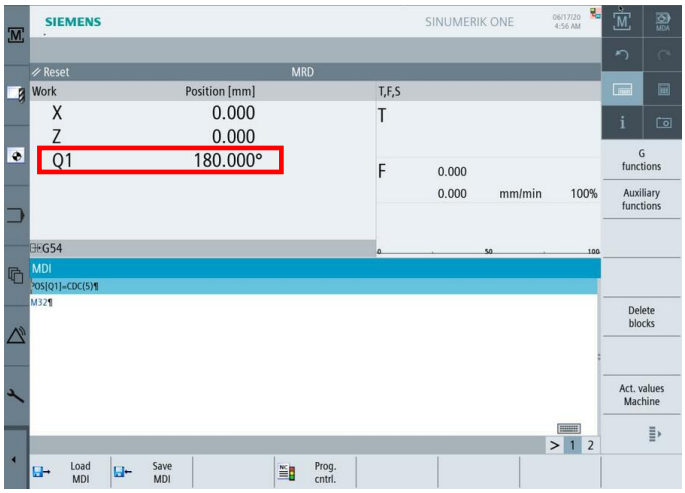
On rotary axes, the indexing position can be approached directly across the shortest path (CDC) or with a defined direction of rotation (CACP, CACN).

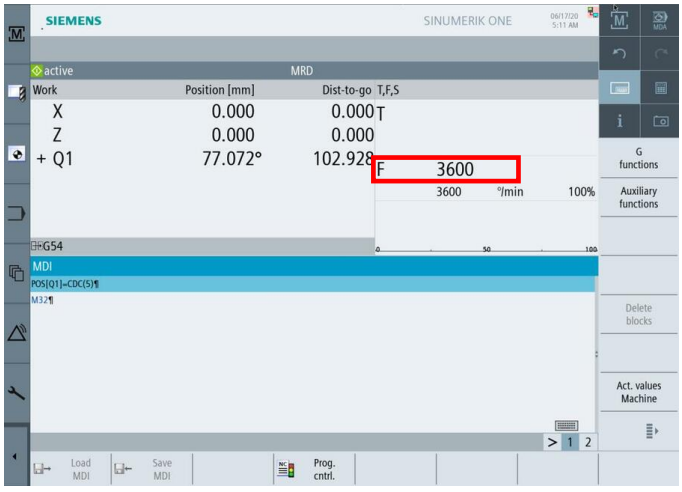
Example 1 :

Turret position number 5

10910[4]	$MN_INDEX_AX_POS_TAB_1	180 mm|deg...

Is approached over the shortest path with program code POS[Q1]=CDC(5).





The turret position 5 is approached at the feedrate set in axis MD32060 $MA_POS_AX_VELO e.g 10 rpm = 3600°/min.

32060	$MA_POS_AX_VELO	10 rpm

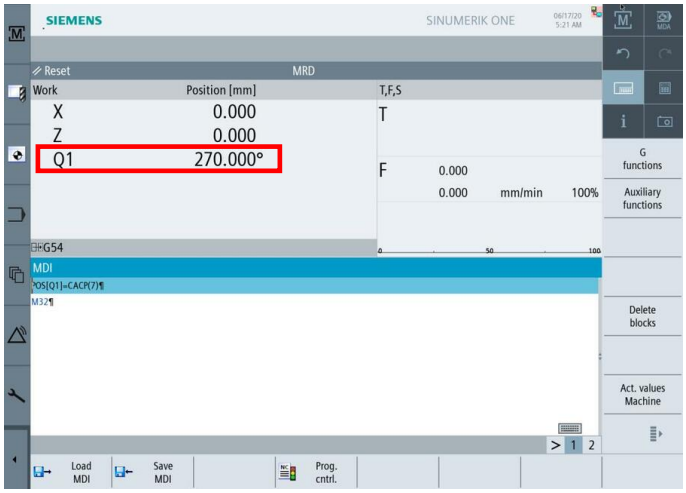
Notes:

Example 2 :

Turret position number 7

10910[6]	$MN_INDEX_AX_POS_TAB_1	270 mm|deg...

Is approached in absolute mode with a positive direction of rotation with program code POS[Q1]=CACP(7).



Hide axis in the actual value windows

It is often desirable in the case of an indexing axis when used as a servo turret to hide the axis display in the actual value window. This can be achieved with channel MD20098 $MC_DISPLAY_AXIS[x]

Bit 0 to 15: Machine (MCS)

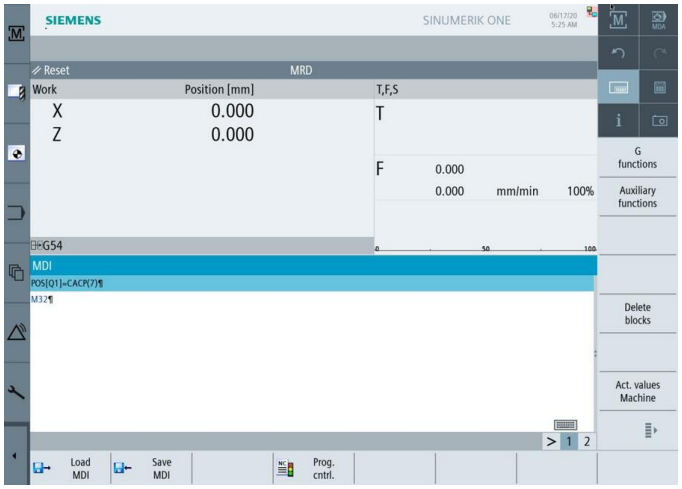
Bit 0= 1 Machine - display machine axis in the actual value windows
0 Machine - hide machine axis in the actual value windows
Bit 1= 1 Machine - display machine axis in the reference point window
0 Machine - hide machine axis in the reference point window
Bit 2= 1 Machine - display machine axis in preset/scratch/parameter work offset windows
0 Machine - hide machine axis in preset/scratch/parameter work offset windows
Bit 3= 1 Machine - display machine axis in the handwheel selection window
0 Machine - hide machine axis in the handwheel selection window
Bit 5= 1 Display spindle in the T,F,S window
0 Hide spindle in the T,F,S window

Bit 16 to 31: Work (WCS)

Bit 16= 1 Work - display geometry axis in the actual value windows
0 Work - hide geometry axis in the actual value windows
Bit 18= 1 Work - display geometry axis in parameter work offset window
0 Work - hide geometry axis in parameter work offset window
Bit 19= 1 Work - display geometry axis in the handwheel selection window
0 Work - hide geometry axis in the handwheel selection window
Bit 20= 1 Work - display position axis in the position/straight line windows
0 Work - hide position axis in the position/straight line windows

With MD20098[2] $MC_DISPLAY_AXIS = 2H then the machine axis is only displayed in the reference point window, and is hidden in all other windows.

20098[2]	$MC_DISPLAY_AXIS	2H



Notes:

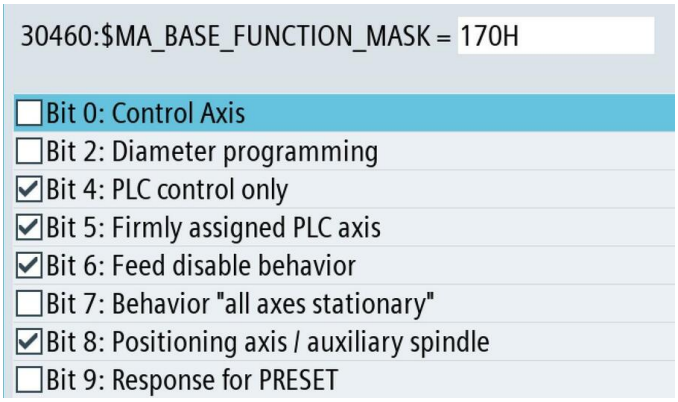
Controlling an Indexing Axis from the PLC

With PLC axis control it is possible to move an indexing axis independently to all other axes.

Using the PLC library block LBP_CtrlAxisSpindle[FC18], SINUMERIK ONE indexing axes can be started from the PLC.

The behaviour of PLC Indexing axis can be manipulated in with machine data:

MD30460 $MA_BASE_FUNCTION_MASK



The example shows a permanently assigned and PLC controlled auxiliary axis

● Bit 4 = 1
The axis is exclusively controlled by the PLC.
● Bit 5 = 1
The axis is a permanently assigned PLC axis.
● Bit 6 = 1
The channel-specific NC/PLC interface signal:
DB21, ... DBX6.0 ("feed disable") is not active for the axis if this is a PLC-controlled axis.
● Bit 7= 1
The channel-specific NC/PLC interface signal:
DB21, ... DBX36.3 ("all axes stationary") is set independently of an axis if this is a PLC-controlled axis.
● Bit 8 = 1:
The axis is a positioning axis / auxiliary spindle

For a PLC-controlled axis:
● The channel-specific NC/PLC interface signal DB21, ... DBX6.0 ("feed disable"), is active if bit 6 = 0
● The channel-specific NC/PLC interface signal DB21, ... DBX36.3 ("all axes stationary") is only set if bit 7 = 0

Starting an indexing axis from the PLC

Activation from PLC
When an Indexing Axis is activated from the PLC, LBP_CtrlAxisSpindle[FC18], is called and supplied with the following parameter data:

● Axis name or axis number
● End position
● Feedrate (with feedrate setting = 0, the feedrate is determined by the setting in machine data MD32060 $MA_POS_AX_VELO):
● Absolute coordinate (G90), incremental coordinate (G91), absolute coordinate along the shortest path for rotary axes (rotary axis name = DC(value))

Prerequisites for the PLC Function:

The axis has to be defined as an Indexing Axis in the NC machine data. MD30500



HMI Requirement:
None

Drive requirement (Sinamics):
None

NCK Requirement:
Indexing Axis

PLC Requirement:
Parameterisation of the FC18 in the PLC

Notes:

Parameter description of FC18

Signal	Type	Format	Range of values	Remark
Start	I	BOOL		Start the function from PLC
Stop	I	BOOL		Stop the function from PLC
Funct	I	BYTE	1 to B#16#0B	1: Position spindle 2: Rotate spindle 3: Oscillate spindle 4: Indexing axis 5: Positioning axis metric 6: Positioning axis inch 7: PosAxis metric with hand-wheel override 8: PosAxis inch with hand-wheel override 9: Rotate spindle with automatic gear stage selection A: Rotate spindle with constant cutting rate (m/min) B: Rotate spindle with constant cutting rate (feet/min)
Mode	I	BYTE	0 to 5	0: Pos to absolute pos 1: Pos incrementally 2: Pos shortest path 3: Pos absolute, positive approach direction 4: Pos absolute, negative approach direction 5: Rotational direction as for M4
AxisNo	I	INT	1 - 31	No. of axis/spindle to be traversed
Pos	I	REAL	0,1469368 I -38 to 0,1701412 I +39	Rotary axis: Degrees Indexing axis: Indexing position Linear axis: mm or inches
FRate	I	REAL	0,1469368 I -38 to 0,1701412 I +39	Rotary axis and spindle: rev/min Linear axis: mm/min or inches/min
InPos	Q	BOOL		1 = Position reached, or function executed
Error	Q	BOOL		1 = error
State	Q	BYTE	0 to 255	Error code

Example:

An example showing the PLC using LBP_CtrlAxisSpindle [FC18], to position an indexing axis to a position specified from MD174 (e.g. from tool management) over the shortest path. (e.g lathe servo turret). The feedrate is not specified therefore the value in MD32060 $MA_POS_AX_VELO will be used.

Funct: = B#16#4 = Indexing axis

Mode: = B#16#2 = Position via shortest path

AxisNo: = Axis number of the Indexing Axis

Pos: = The indexing position that the axis is commanded to move to, supplied from PLC e.g. from tool management.

FRate: = Feedrate for the axis move. If 0 then MD32060 $MA_POS_AX_VELO will be used.

```
FUNCTION FC 114 : VOID
TITLE =
VERSION : 0.1
BEGIN
NETWORK
TITLE = Pos Axis FC18

    A   I  81.6; //
    FP  M 141.0; // Transition marker
    S   M 141.1;
    A   M 141.3;
    FP  M 141.4; // Transition Marker
    R   M 141.1;
    R   M 141.7;
    A   I  81.7;
    FP  M 141.6; // Transition Marker
    S   M 141.7;
    CALL "LBP_CtrlAxisSpindle" (
        Start                   := M141.1,
        Stop                    := M141.7,
        Funct                   := B#16#4,
        Mode                    := B#16#2,
        AxisNo                  := 3,
        Pos                     := MD174,
        FRate                   := 0.000000e+000,
        InPos                   := M  141.3,
        Error                   := M  141.5,
        State                   := MB 142);
END_FUNCTION
```

A407: END

Notes:

A415

SINUMERIK ONE

Axis and Spindle Compensations

Module Description:

This module covers the Axis and Spindle Compensation functions which can increase a machine tools' accuracy if the machine tool is determined to be mechanically sound. Even a machine that is not mechanically sound may be improved substantially as long as any error is repeatable.

Module Objective:

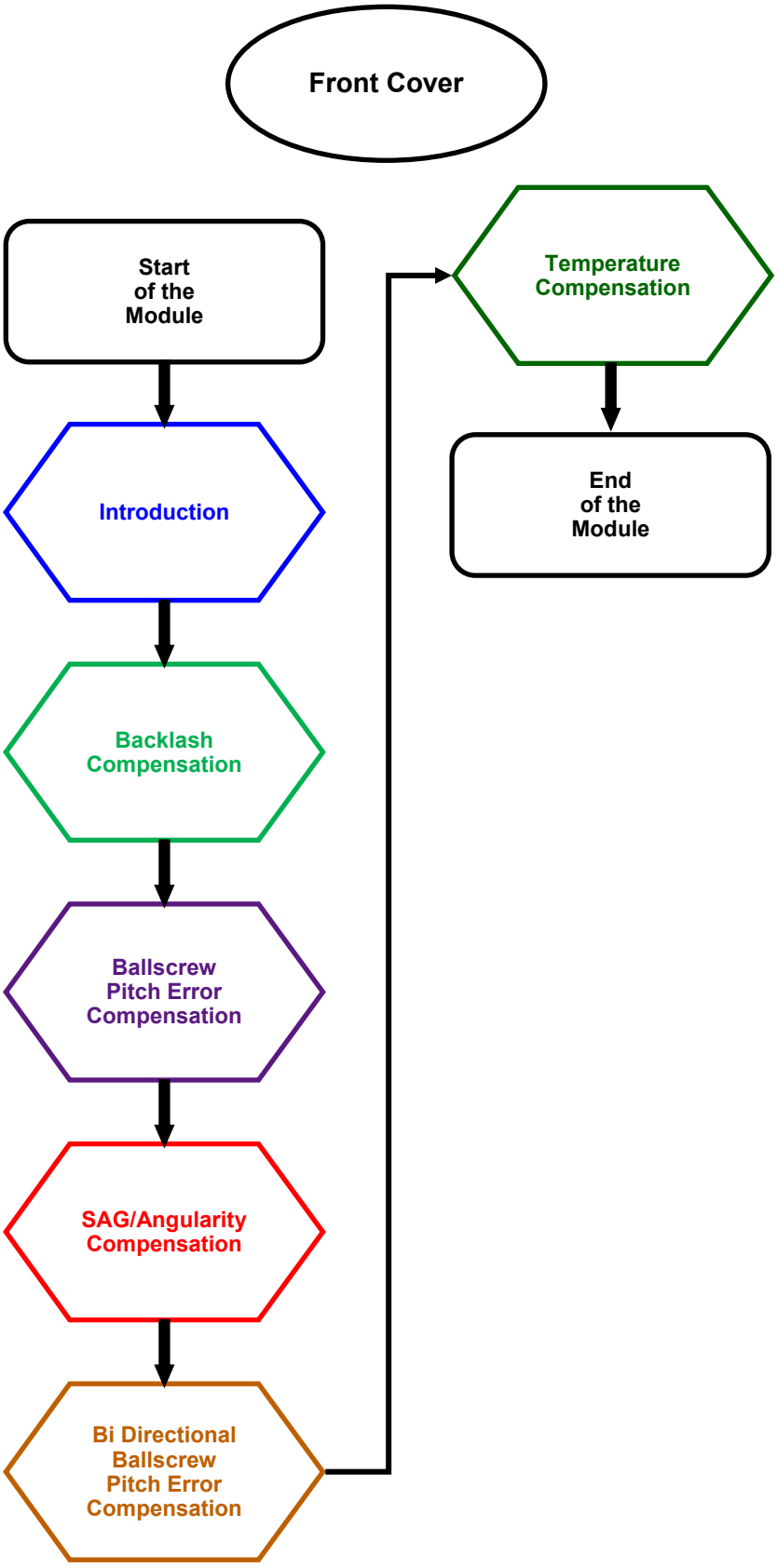
Upon completion of this module you should be able to measure and activate the compensations covered in this module.

Content:

Introduction
Backlash Compensation
Ballscrew Pitch Error Compensation
Sag/Angularity Compensation
Bi Directional Ballscrew Pitch Error Compensation
Temperature Compensation

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Introduction
Backlash Compensation
Ballscrew Pitch Error Compensation
SAG/Angularity Compensation
Bi Directional Ballscrew Pitch Error Compensation
Temperature Compensation
End of the Module

Introduction

Accuracy errors

The accuracy of machine tools is impaired as a result of deviations from the ideal geometry, power transmission faults and measuring system errors. Temperature differences and mechanical forces often result in great reductions in precision when workpieces are machined.
Compensation functions can increase the machine tools' accuracy if the machine tool is determined to be mechanically sound. Even a machine that is not mechanically sound may be improved substantially as long as any error is repeatable.

Compensation functions

Some of these deviations can usually be measured during commissioning and then compensated for during operation on the basis of values read by the positional actual-value encoder and other sensory devices. The Sinumerik ONE controller has compensation functions that are active on an axis for axis basis.

The following compensation functions are available

- Backlash compensation
- Interpolatory compensation
  – Compensation of leadscrew errors and Measuring system errors
  – Compensation of sag and angularity errors
- Temperature compensation
- Dynamic feedforward control (following error compensation)
- Friction compensation
- Conventional friction compensation
- Electronic counterweight

Parameterization

These compensation functions can be set for each machine individually with axis-specific machine data.

Options

Backlash/Ballscrew pitch error compensation
No option required

Sag compensation, multi-dimensional
6FC5800-0BM55-0Yx0

Bi-directional compensation
6FC5800-0BM54-0Yx0

Notes:

Backlash Compensation

Mechanical backlash compensation

When force is transmitted between a moving machine part and its drive (e.g. ball screw), there is normally a small amount of backlash because adjusting mechanical parts so that they are completely free of backlash would result in too much wear and tear on the machine. Thus, backlash (play) can occur between the machine component and the measuring system

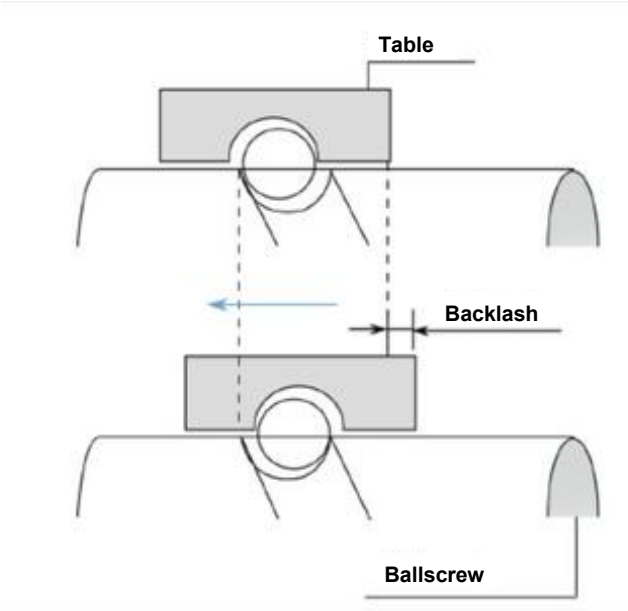
Almost all mechanical systems have backlash, also known as reversal error. A certain amount of backlash is usually necessary to provide lubrication and thermal tolerance. Backlash shows up as lost motion when an axis reverses direction. Backlash can come from several areas

.

- Gearing
- Ball screw nuts
- Ball screw support bearings
- Rack and pinions
- Couplings
- Loose components
- Flexing components

If the backlash is repeatable, it can be compensated for by adding an additional movement to the axis, equal to the amount of the backlash, whenever the axis reverses. Backlash can be tested with indicators, lasers, ballbars or other measuring devices to determine how much motion is lost when an axis reverses direction. This value is entered in the axis specific machine data, so that when an axis changes direction, it compensates by the amount entered into the parameter.

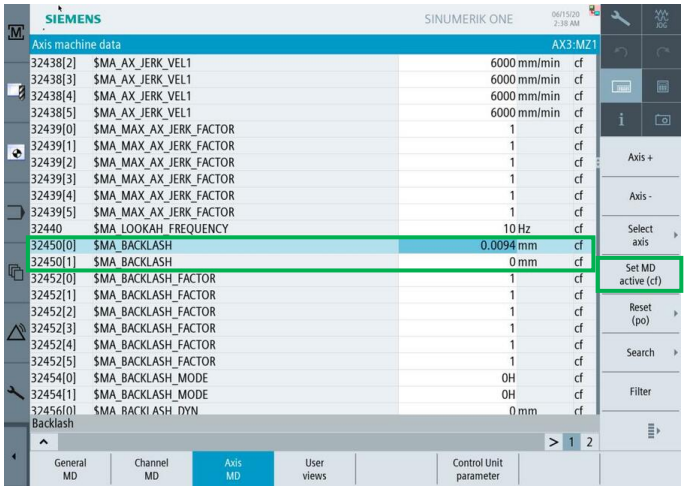
The figure below shows an example of mechanical backlash in a ballscrew nut



Commissioning

To compensate for the mechanical backlash, the axis-specific backlash amount is entered into the machine data.

MD32450[0] or [1] $MA_BACKLASH



Notes:

If the machine axis only has a motor encoder then the backlash figure should be entered into:

MD32450[0] $MA_BACKLASH

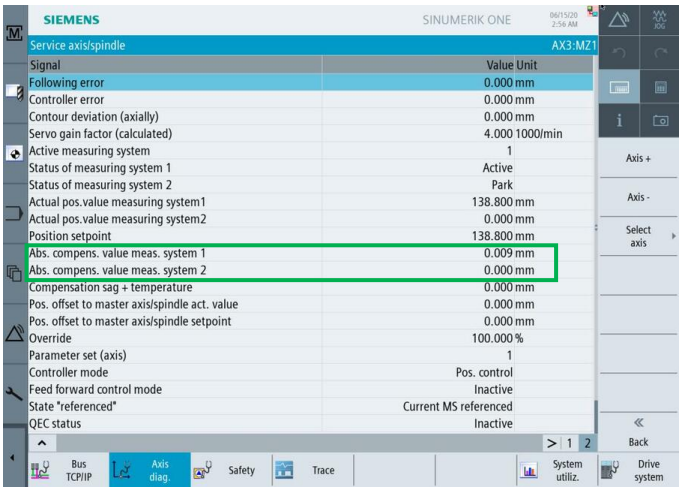
If the machine axis has a second measuring system it is also possible for backlash to occur and so if this is the case the backlash figure should be entered into:

MD32450[1] $MA_BACKLASH

For positive backlash (normal case), the compensation value should be entered as a positive value and for a negative backlash, as negative value.

Display

The compensation value that belongs to the current actual position is displayed in the "Diagnostics" operator area of the "Axis/Spindle Service" window as the total compensation calculated from ballscrew pitch error compensation and mechanical backlash compensation.



Notes:

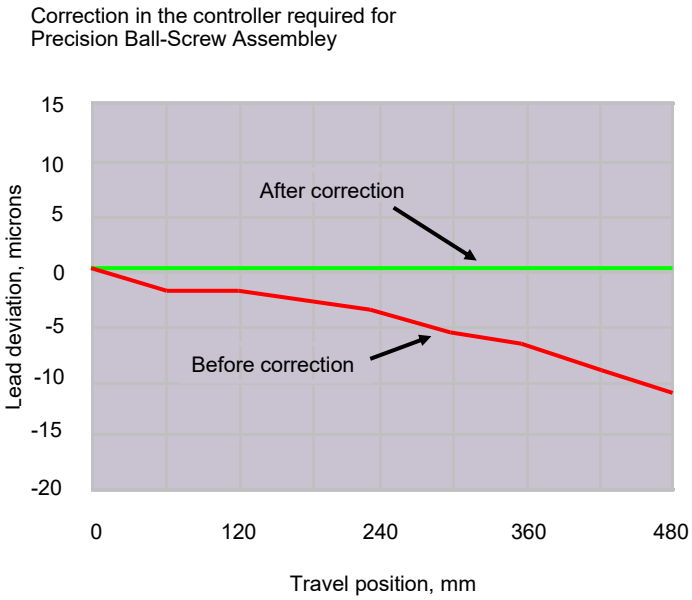
Ballscrew Pitch Error Compensation

Absolute positioning of a machine tool is subject to errors in the manufacturing of the machine's components. All ball screws or gearing have lead (pitch) errors. Even linear scales and encoders are not perfect and have inherent manufacturing errors.

Using a laser interferometer, these errors can be measured at fixed intervals. The errors are then stored in the control and when an axis is interpolating the positional error is divided into smaller increments which effectively creates a straight line between compensation intervals and smoothes out the error compensation output.

The mechanical error of the ball screw can be compensated by the Sinumerik ONE controller for both the indirect measuring system, i.e. the motor encoder, or the direct measuring system, typically a linear scale or high resolution direct encoder.

Typical ball screw error after measurement with a laser interferometer that would require compensation



Entering Compensation Tables

The size of the compensation table, i.e. the number of interpolation points, must first be defined in axis specific machine data. After the next PO, the compensation tables are generated by the NC and preassigned values of "0".

If this takes place later in the commissioning process then the machine data should be modified, and then an archive saved out and reloaded. This ensures that other data is not lost when the static memory is reformatted.

MD38000[0] $MA_MM_ENC_COMP_MAX_POINTS
Should be used if the compensation is to be carried out on the motor encoder.

MD38000[1] $MA_MM_ENC_COMP_MAX_POINTS
Should be used if the compensation is to be carried out on the 2nd encoder/linear scale

38000[0]	$MA_MM_ENC_COMP_MAX_POINTS	25
38000[1]	$MA_MM_ENC_COMP_MAX_POINTS	0

Pitch Error Compensation Data File

This data can be loaded into the control as a part program, and then executed. An example is shown below.

```
MSG("Read Table, continue with NC-START !")
M0
STOPRE
MSG("")
CHANDATA(1)
$MA_ENC_COMP_ENABLE[0,AX1]=0;        Compensation Axis AX1 OFF
NEWCONF
STOPRE
$AA_ENC_COMP[0,0,AX1]=0.000
$AA_ENC_COMP[0,1,AX1]=0.000
$AA_ENC_COMP[0,2,AX1]=0.001
$AA_ENC_COMP[0,3,AX1]=0.002
$AA_ENC_COMP[0,4,AX1]=0.002
$AA_ENC_COMP[0,5,AX1]=0.002
$AA_ENC_COMP[0,6,AX1]=0.002
$AA_ENC_COMP[0,7,AX1]=0.002

Intermediate points

$AA_ENC_COMP[0,18,AX1]=0.007
$AA_ENC_COMP[0,19,AX1]=0.007
$AA_ENC_COMP[0,20,AX1]=0.008
$AA_ENC_COMP[0,21,AX1]=0.009
$AA_ENC_COMP[0,22,AX1]=0.010
$AA_ENC_COMP[0,23,AX1]=0.011
$AA_ENC_COMP[0,24,AX1]=0.012
$AA_ENC_COMP_STEP[0,AX1]=20;         Correction point distance
$AA_ENC_COMP_MIN[0,AX1]=0;           start of comp
$AA_ENC_COMP_MAX[0,AX1]=480;         end of comp
$AA_ENC_COMP_IS_MODULO[0,AX1]=0;     no modulo comp required
$MA_ENC_COMP_ENABLE[0,AX1]=1;        Compensation Axis AX1 ON
NEWCONF
MSG("End")
G4F2
M30
```

! The comp in the table should be in the opposite direction to the error recorded by the laser interferometer, as shown here relative to the example above

Notes:

Before executing the program the compensation must be disabled (already in program above)

MD32700[0] or [1] $MA_ENC_COMP_ENABLE = 0

32700[0]	$MA_ENC_COMP_ENABLE	0
32700[1]	$MA_ENC_COMP_ENABLE	0

Once the program has been executed then the compensation should be enabled (already in program above)

MD32700[0] or [1] $MA_ENC_COMP_ENABLE = 1

32700[0]	$MA_ENC_COMP_ENABLE	1
32700[1]	$MA_ENC_COMP_ENABLE	0

MD32700[0] is for the motor encoder and **MD32700[1]** is for linear scale/2nd encoder

After executing the the file it is possible to see that the configuration was successful in the "Setup" operating area by pressing the "NC" softkey, followed by the “Leadscrew error” softkey and then “Axis +”, “Axis - “ to the relevant axis.



Pitch Error Comp Data File Explanation

“$AA_ENC_COMP_STEP[m,n]=p” defines the increment between compensation points where “m” is the measuring system (0 for motor encoder or 1 for linear scale), “n” is the axis label & “p” is the increment distance between compensation points.

“$AA_ENC_COMP_MIN[m,n]=p” defines the minimum axis position for compensation where “m” is the measuring system (0 for motor encoder or 1 for linear scale), “n” is the axis label & “p” is the minimum compensation value.

“$AA_ENC_COMP_MAX[m,n]=p” defines the maximum axis position for compensation where “m” is the measuring system (0 for motor encoder or 1 for linear scale), “n” is the axis label & “p” is the maximum compensation value.

“$AA_ENC_COMP[m,n,p]=q” defines the compensation values at each compensation point starting at the minimum value to the maximum value incrementing by the step distance. “m” is the measuring system (0 for motor encoder or 1 for linear scale), “n” is the index value starting at 0 for the 1st compensation point, “p” is the axis label & “q” is the compensation value.

“$AA_ENC_COMP_IS_MODULO[m,n]=p” defines whether the compensation values are to be used on a rotary modulo axis. “m” is the measuring system (0 for motor encoder or 1 for linear scale), “n” is the axis label & “p” is 1 to enable or 0 to disable. So on a linear axis “p” should be 0.

! The Sinumerik pitch error compensation feature interpolates between each point so the compensation is not stepped at each compensation point.

!The compensation value at the reference point must always be 0.

! Points outside the actual traversing range can remain as 0.

! The “MIN” & “MAX” value can have the same absolute sign as long as the “MIN” value is less than the “MAX” value.

! Whether the compensation values are entered in metric or inch format is dependant upon :
MD10240 $MN_SCALING_SYSTEM_IS_METRIC.
Normally if the ballscrew pitch is metric the compensation will be carried out in metric and if the ballscrew pitch is inch the compensation will be carried out in inch

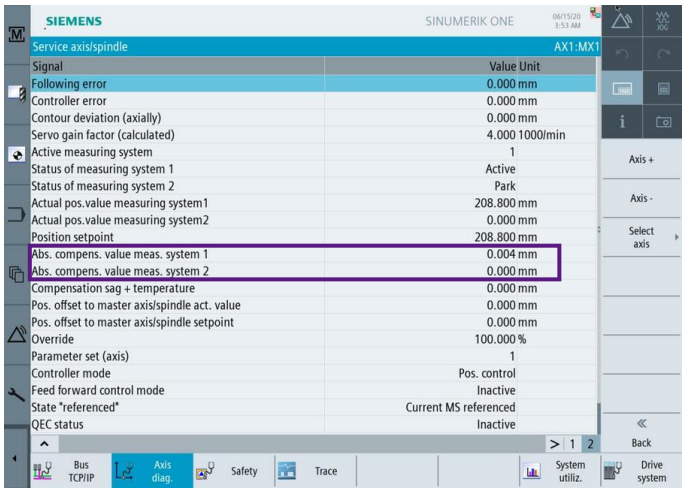
! Carry out an NCK reset for the new data to become effective.

! The compensation data will only become effective once the axis is referenced

Notes:

Display

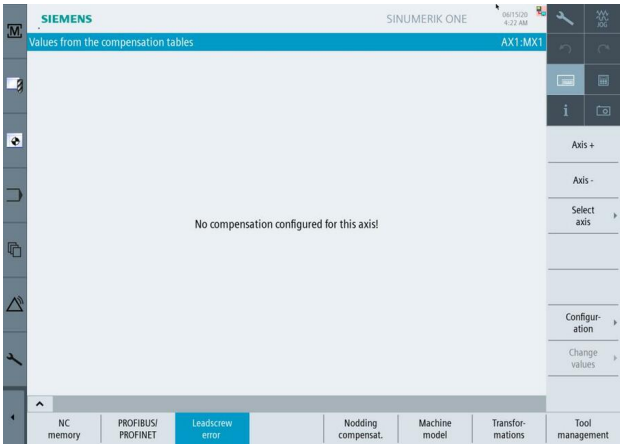
The compensation value that belongs to the current actual position is displayed in the "Diagnostics" operator area of the "Axis/Spindle Service" window as the total compensation calculated from ballscrew pitch error compensation and mechanical backlash compensation.



Manual Configuration of Compensation Table

Instead of transferring the compensation data via the part program or INI file to the NC, you can enter and check the compensation data for all axes or change the configuration in an input screen. The changed data is activated in the NC in one step. The machine data required when activating is set automatically.
MD38000[0][1]$MA_MM_ENC_COMP_MAX_POINTS should be set prior to manual configuration

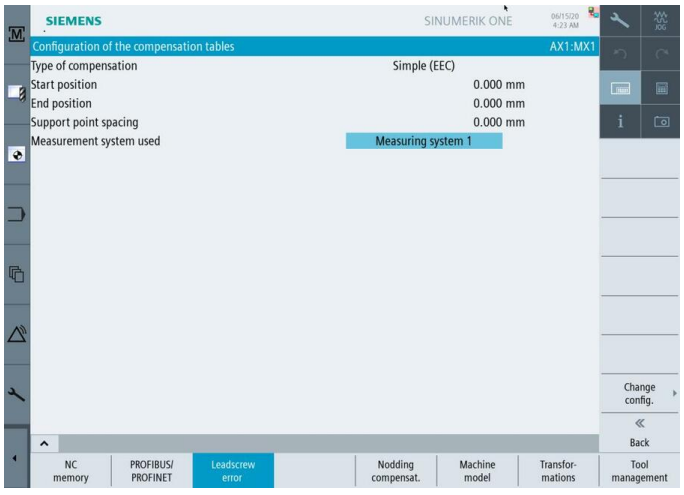
Select the "Setup" operating area and press the "NC" softkey, followed by the “Leadscrew error” softkey and then “Axis +”, “Axis - “ to the relevant axis.



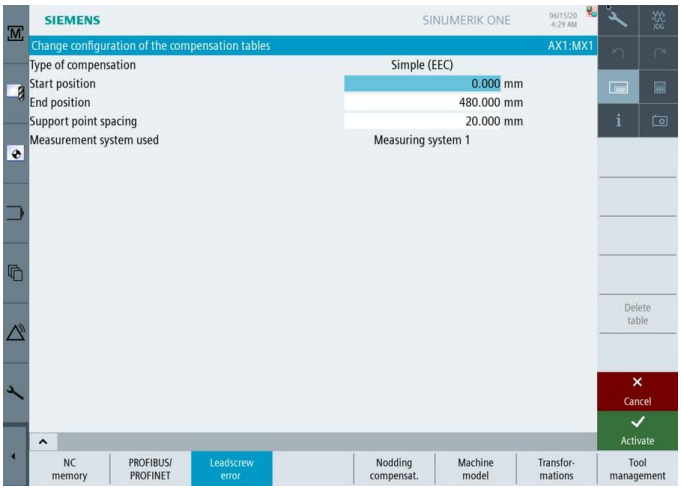
Press the "Configuration" softkey to edit the compensation values for a selected axis.

The "Configuration of the Compensation Tables" window opens.

Press the “Change config” softkey



Notes:



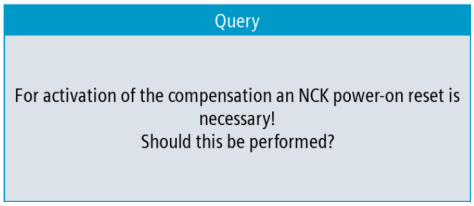
The Type of compensation is Simple (EEC)

Enter the following:

- Start position
- End position
- Support (interpolation) point spacing.

Confirm your entries with the "Activate" softkey.

A dialog box to carry out a NC warm restart is then displayed to activate the compensation. Confirm with "OK".





The "Values of the Compensation Tables" window opens.

Enter the desired compensation values for each interpolation point.

Press the "Activate" softkey again to transfer the compensation values to the NC.

Notes:

Compensation of Sag and Angularity errors

General information

The sag / angularity error compensation (cross error compensation, **CEC**) is a multi-axis compensation. To compensate for sag or angularity errors, the position setpoint of an output or compensation axis is affected here, depending on the current position of an input or basic axis.

Option Required

Sag compensation, multi-dimensional
6FC5800-0BM55-0Yx0

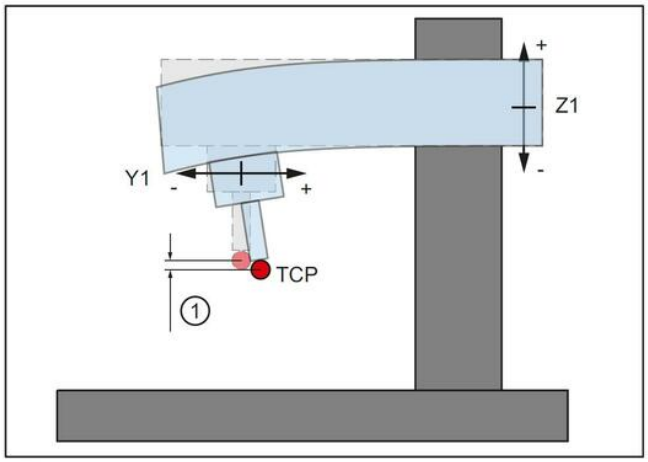
Sag errors

In the case of sag errors, the weight of a moving machine part or work piece leads to a position dependent error in the tool centre point (TCP).

Angularity errors

For angularity errors, a deviation in the angle in the geometric axes in relation to each other leads to a position-dependent error in the tool centre point (TCP).

The figure below shows an example of sag caused by an axes own weight. i.e. there is a position deviation in Z1 because of the sag at the current position when the Y1 axis moves. The more that the Y1 axis is moved in the negative direction, the more error seen in Z1 minus direction.



Sag Compensation Tables

The error must be recorded in the form of a compensation table that contains a compensation value for the Z1 axis for every actual position in the Y1 axis.

The compensation values are entered into the compensation table via interpolation points. When the Y1 axis traverses, the control calculates the actual compensation value in the Z1 axis using linear interpolation between the interpolation points. The Z1 axis is assigned a correction value as the additional target value. A positive compensation value causes the compensation axis to move in the negative traverse direction.

Since it is possible to use the position of an axis as the input quantity (base axis) for several tables, to derive the total compensation value of an axis from several compensation relationships (tables) and to multiply tables, it is also possible to implement sophisticated andcomplex beam sag and angularity error compensation systems.

Entering Sag Compensation Tables

The size of the sag compensation tables in the Sinumerik ONE is defined in general machine data. After the next PO, the compensation table is generated by the NC and preassigned values of "0".

If this takes place later in the commissioning process then the machine data should be modified, and then an archive saved out and reloaded. This ensures that other data is not lost when the static memory is reformatted.

MD18342[0] $MN_MM_CEC_MAX_POINTS
Should be used for setting up the number of compensation points for the first compensation table.

18342[0]	$MN_MM_CEC_MAX_POINTS	128
18342[1]	$MN_MM_CEC_MAX_POINTS	128

Notes:

Sag Error Compensation Data File

This data can be loaded into the control as a part program, and then executed. An example is shown below.

```
MSG("Read Table, continue with NC-START !")
M0
STOPRE
MSG("")
CHANDATA(1)
$MA_CEC_ENABLE[AX3]=0      ;    Axis 3 compensation disabled
$SN_CEC_TABLE_ENABLE[0]=0   ;disable Table 1
NEWCONF
STOPRE
$AN_CEC[0,0]=0.018;    1st compensation value (interpolation point 0) for Z1 +0.018
$AN_CEC[0,1]=0.018;    2nd compensation value (interpolation point 1) for Z1 +0.018
$AN_CEC[0,2]=0.017;    3rd compensation value (interpolation point 2) for Z1 +0.017
$AN_CEC[0,3]=0.017;    4th compensation value (interpolation point 3) for Z1 +0.017
$AN_CEC[0,4]=0.017
$AN_CEC[0,5]=0.016

Intermediate points

$AN_CEC[0,97]=0.002
$AN_CEC[0,98]=0.001
$AN_CEC[0,99]=0.001
$AN_CEC[0,100]=0.000;  last compensation value (interpolation point 101) for Z1 0.000
$AN_CEC_INPUT_AXIS[0]=(AX2);          basic axis Y1
$AN_CEC_OUTPUT_AXIS[0]=(AX3) ;        compensation axis Z1
$AN_CEC_STEP[0]=8;                    distance between interpolation points =8mm
$AN_CEC_MIN[0]=-400;                  compensation starts at Y1 = -400
$AN_CEC_MAX[0]=400;                   compensation ends at Y1 = +400
$AN_CEC_DIRECTION[0]=0;               table applies for both directions of travel of Y1
$AN_CEC_MULT_BY_TABLE[0]=0;           no table multiplier
$AN_CEC_IS_MODULO[0]=0;               compensation without modulo function
$MA_CEC_ENABLE[AX3]=1;                Axis 3 compensation enabled
$SN_CEC_TABLE_ENABLE[0]=1   ;         enable Table 1
NEWCONF
MSG("End")
G4F2
M30
```

Before executing the program the compensation must be disabled (already in program above)

MD32710 $MA_CEC_ENABLE = 0

32710	$MA_CEC_ENABLE	0

Once the program has been executed then the compensation should be enabled (already in program above)

MD32710 $MA_CEC_ENABLE = 1

32710	$MA_CEC_ENABLE	1

Evaluation of the relevant compensation table has to also be enabled with axis specific setting data (already in program above)

SD41300 $SN_CEC_TABLE _ENABLE[<t>] = 1

41300[0]	$SN_CEC_TABLE_ENABLE	1
41300[1]	$SN_CEC_TABLE_ENABLE	0

Sag Error Comp Data File Explanation

"$AN_CEC[m,n]=q" defines the compensation values at each compensation point starting at the minimum value to the maximum value incrementing by the step distance. "m" is the index of the compensation table, "n" is the index value starting at 0 for the 1st compensation point, "q" is the compensation value.

"$AN_CEC_INPUT_AXIS[m]=(p)" defines the name of the axis whose setpoint is to be used as the input for the compensation table, "m" is the index of the compensation table, "p" is the channel axis or the machine axis name.

"$AN_CEC_OUTPUT_AXIS[m]=(p)" defines the name of the axis to which the correction value is switched, "m" is the index of the compensation table, "p" is the channel axis or the machine axis name.

"$AN_CEC_STEP[m]=(p)" defines the distance between the compensation points, "m" is the index of the compensation table, "p" is the distance between the interpolation points. Together with the start and end position, the number of interpolation points is specified by the interpolation point distance.

"$AN_CEC_MIN[m]=p" defines the starting position for the basic axis where the compensation table begins where "m" is the index of the compensation table, "p" is the starting position.

"$AN_CEC_MAX[m]=p" defines the end position for the basic axis where the compensation table ends where "m" is the index of the compensation table, "p" is the end position.

"$AN_CEC_DIRECTION[m]=p" defines whether the compensation table should affect both traverse directions, or only the positive or negative traverse direction of the basic axis, "m" is the index of the compensation table, "p" sets whether the compensation is direction dependent.
p=0 = in both traverse directions
p=1= only in positive traverse direction
p=-1= only in negative traverse direction

"$AN_CEC_MULTI_BY_TABLE[m]=p" defines whether the current compensation value is multiplied by the corresponding compensation value of any other compensation table, "m" is the index of the compensation table, "p" is the number of the table to be used for the multiplication.

An example for table multiplication would be for the compensation of machine foundation sagging on a large machine, causing inclination of the whole machine. For a large boring mill the compensation of the X1 axis could depend on the position of the X1 axis itself, since this determines the angle of inclination β, and on the height Z1 of the drill.
The compensation values of the X1 and Z1 axes would be linked as follows for a compensation:
$\Delta X1 = Z1 * \sin\beta(X1) \approx Z1 * \beta(X1)$

Notes:

Compensation table 1 (index = 0) describes the reaction of axis X1 on axis X1 (sine of the position-dependent tilting angle β(X1)).
Compensation table 2 (index = 1) describes the reaction of axis Z1 on axis X1 (linear).
In table 1, the multiplication of table 1 (index = 0) with table 2 is to be selected:
$AN_CEC_MULT_BY_TABLE[0] = 2

"$AN_CEC_IS_MODULO[m]=p" defines whether modulo compensation is to be implemented with a modulo rotary axis as basic axis, the compensation table used has to be modulo calculated as well, where "m" is the index of the compensation table, "p" sets whether or not the modulo function is used.
p=0 = no modulo function
p=1= compensation with modulo function

!The "sag compensation" function does not become active until the following conditions are fulfilled:

!The function has been activated for the relevant machine axis (compensation axis) with
MD32710 $MA_CEC_ENABLE[<AXi>] = 1

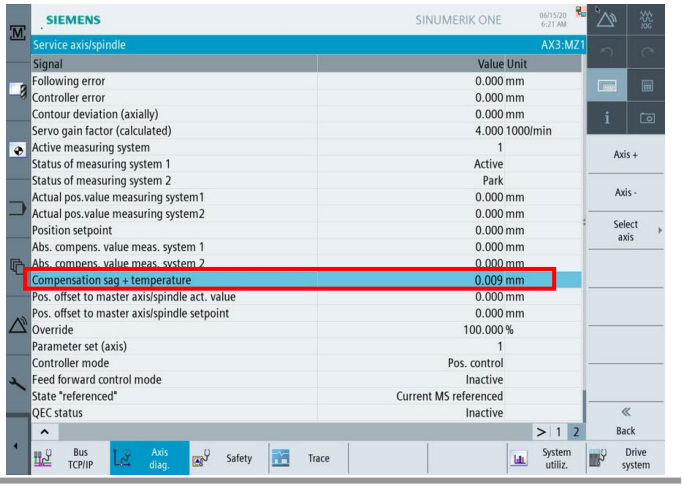
!The compensation values are stored in the static user memory and are active (POWER ON).

!Evaluation of the relevant compensation table has been enabled:
SD41300 $SN_CEC_TABLE_ENABLE[<t>] = 1

!The current measuring system of the base and compensation axes are referenced:

Display

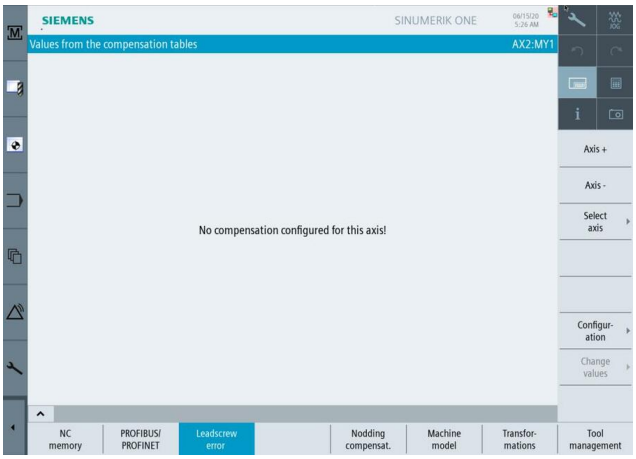
The Sag + temperature compensation value that belongs to the current actual position is displayed in the "Diagnostics" operator area of the "Axis/Spindle Service" window.



Manual Configuration of Sag Compensation Table

Instead of transferring the compensation data via the part program or INI file to the NC, you can enter and check the compensation data for all axes or change the configuration in an input screen. The changed data is activated in the NC in one step. The machine data required when activating is set automatically.**MD18342[0]$MN_MM_CEC_MAX_POINTS** should be set prior to manual configuration

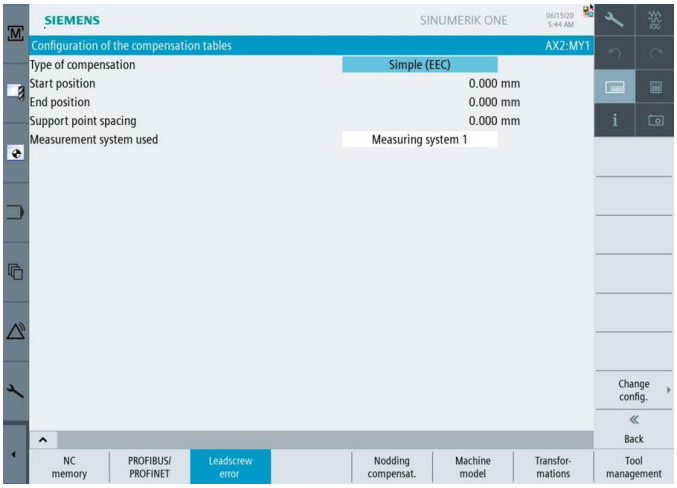
Select the "Setup" operating area and press the "NC" softkey, followed by the "Leadscrew error" softkey and then "Axis +", "Axis - " to the relevant axis.



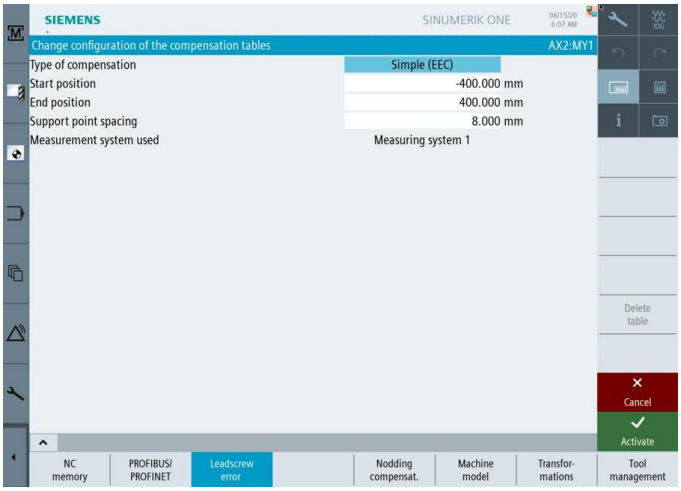
Press the "Configuration" softkey to edit the compensation values for a selected axis.

The "Configuration of the Compensation Tables" window opens.

Press the "Change config" softkey



Notes:

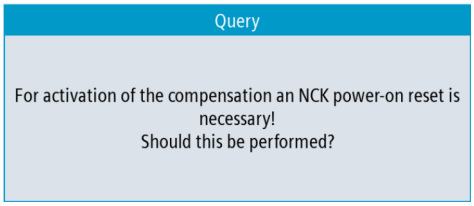


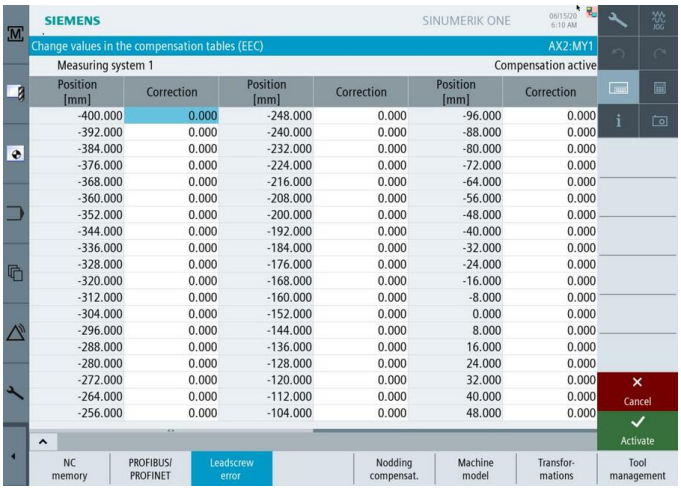
Enter the following:

- Type of compensation: Simple (EEC).
- Start position, End position and Support (interpolation) point spacing.
- Measuring system—if several measuring systems are available, select the desired system from the selection menu.

Confirm your entries with the "Activate" softkey.

A dialog box to carry out a NC warm restart is then displayed to activate the compensation. Confirm with "OK".





The "Values of the Compensation Tables" window opens.

Enter the desired compensation values for the compensation axis at each interpolation point of the base axis.

Press the "Activate" softkey again to transfer the compensation values to the NC.

Notes:

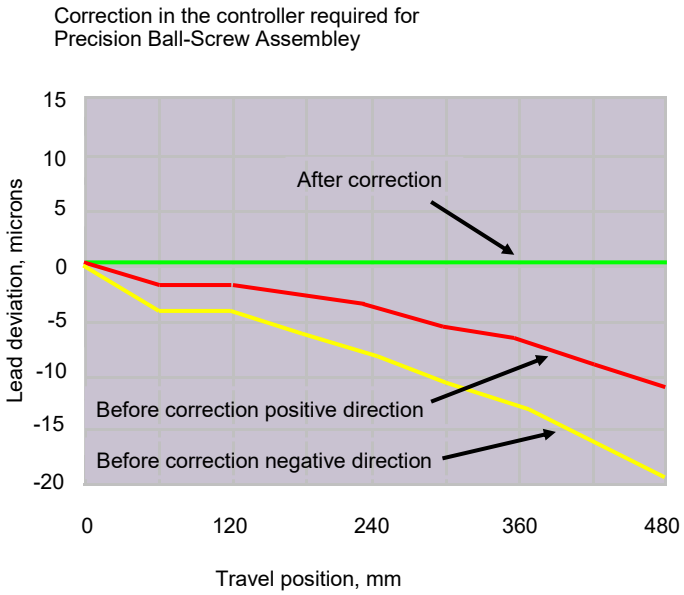
Bi Directional Ballscrew Pitch Error Compensation

If the direction-dependent differences at the compensation points are excessively high, for an inconsistent backlash or for extremely high demands placed on the precision, then it may be necessary to apply bi directional compensation of the ballscrew screw error or measuring system error (for direct position sensing).

For the "Bi directional ballscrew error compensation" two compensation tables are used for each axis. One compensation table for the positive and one compensation table for the negative traversing direction. The deviation at the particular compensation point is entered as difference between the ideal setpoint and measured actual value in the compensation tables. The control automatically calculates compensation values of intermediate values using linear interpolation.

The "Bi directional pitch error compensation" is implemented in the SINUMERIK control as a special case of "sag compensation". Normally with “sag compensation” the position setpoint of an output or compensation axis is affected depending on the current position of an input or basic axis. With "Bi directional ballscrew error compensation" the input and output axes are the same axis.

Typical ball screw error after measurement with a laser interferometer that would require bi directional compensation



Option Required

Sag compensation, multi-dimensional
6FC5800-0BM55-0Yx0

Two compensation tables are used for each axis. One compensation table for the positive and one compensation table for the negative traversing direction.

Entering Sag (bi-directional) Compensation Tables

The size of the sag compensation tables in the Sinumerik ONE is defined in general machine data. After the next PO, the compensation table is generated by the NC and preassigned values of "0".

If this takes place later in the commissioning process then the machine data should be modified, and then an archive saved out and reloaded. This ensures that other data is not lost when the static memory is reformatted.

MD18342[0] $MN_MM_CEC_MAX_POINTS
Should be used for setting up the number of compensation points for the first compensation table, first axis positive direction.

MD18342[1] $MN_MM_CEC_MAX_POINTS
Should be used for setting up the number of compensation points for the second compensation table, first axis negative direction.

18342[0]	$MN_MM_CEC_MAX_POINTS	128
18342[1]	$MN_MM_CEC_MAX_POINTS	128

Notes:

Sag (bi-directional) Error Compensation Data File

This data can be loaded into the control as a part program, and then executed. An example is shown below.

```
MSG("Read Table, continue with NC-START !")
M0
STOPRE
MSG("")
;First Table - positive direction
;Second Table - negative direction
CHANDATA(1)
$MA_CEC_ENABLE[AX1]=0      ;    disable compensation
$SN_CEC_TABLE_ENABLE[0]=0   ;disable Table 1
$SN_CEC_TABLE_ENABLE[1]=0   ;disable Table 2
NEWCONF
STOPRE
;First Axis AX1
;First Table
;Direction plus
$AN_CEC[0,0]=0.0;    1st compensation value (interpolation point 0) for X1 +0.0
$AN_CEC[0,1]=0.018;   2nd compensation value (interpolation point 1) for X1 +0.018
$AN_CEC[0,2]=0.017;   3rd compensation value (interpolation point 2) for X1 +0.017
$AN_CEC[0,3]=0.017;    4th compensation value (interpolation point 3) for X1 +0.017
$AN_CEC[0,4]=0.017

Intermediate points

$AN_CEC[0,21]=0.002
$AN_CEC[0,22]=0.001
$AN_CEC[0,23]=0.001
$AN_CEC[0,24]=0.0;  last compensation value (interpolation point 24) for X1 = 0.0
$AN_CEC_INPUT_AXIS[0]=(AX1);        basic axis X1
$AN_CEC_OUTPUT_AXIS[0]=(AX1) ;   compensation axis X1
$AN_CEC_STEP[0]=20;                  distance between interpolation points =8mm
$AN_CEC_MIN[0]=0;                   compensation starts at X1 = 0
$AN_CEC_MAX[0]=480;                  compensation ends at X1 = +480
$AN_CEC_DIRECTION[0]=1;            table applies for positive direction of travel of X1
$AN_CEC_MULT_BY_TABLE[0]=0;      no table multiplier
$AN_CEC_IS_MODULO[0]=0;             compensation without modulo function
```

```
;First Axis AX1
;Second Table
;Direction minus
$AN_CEC[1,0]=0.0;    1st compensation value (interpolation point 0) for X1 +0.0
$AN_CEC[1,1]=0.028;   2nd compensation value (interpolation point 1) for X1 +0.028
$AN_CEC[1,2]=0.027;   3rd compensation value (interpolation point 2) for X1 +0.027
$AN_CEC[1,3]=0.027;    4th compensation value (interpolation point 3) for X1 +0.027
$AN_CEC[1,4]=0.027

Intermediate points

$AN_CEC[1,21]=0.012
$AN_CEC[1,22]=0.011
$AN_CEC[1,23]=0.011
$AN_CEC[1,24]=0.010;  last compensation value (interpolation point 24) for X1 = 0.010
$AN_CEC_INPUT_AXIS[1]=(AX1);        basic axis X1
$AN_CEC_OUTPUT_AXIS[1]=(AX1) ;   compensation axis X1
$AN_CEC_STEP[1]=20;                  distance between interpolation points =8mm
$AN_CEC_MIN[1]=0;                   compensation starts at X1 = 0
$AN_CEC_MAX[1]=480;                  compensation ends at X1 = +480
$AN_CEC_DIRECTION[1]=-1;           table applies for negative direction of travel of X1
$AN_CEC_MULT_BY_TABLE[1]=0;      no table multiplier
$AN_CEC_IS_MODULO[1]=0;             compensation without modulo function
$MA_CEC_ENABLE[AX1]=1      ;        enable compensation
$SN_CEC_TABLE_ENABLE[0]=1   ;     enable Table 1
$SN_CEC_TABLE_ENABLE[1]=1   ;     enable Table 2
NEWCONF
MSG("End")
G4F2
M17
```

Before executing the program the compensation must be disabled: (already in program above)

MD32710 $MA_CEC_ENABLE = 0

32710	$MA_CEC_ENABLE	0

Once the program has been executed then the compensation should be enabled (already in program above)

MD32710[0] $MA_CEC_ENABLE = 1

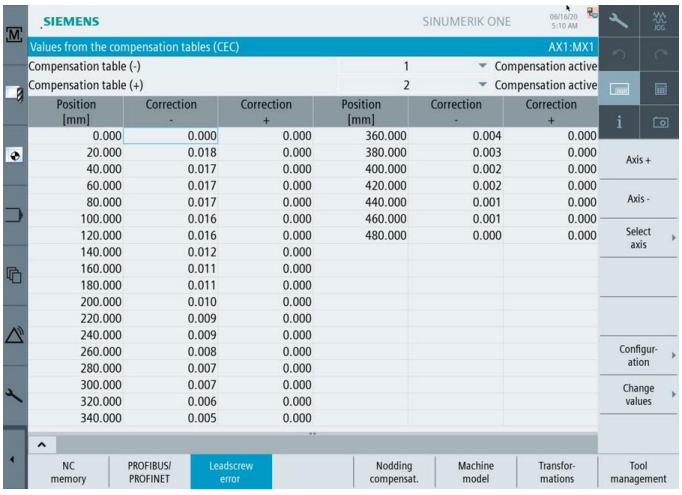
32710	$MA_CEC_ENABLE	1

Evaluation of the relevant compensation table has to also be enabled with axis specific setting data (already in program above)

SD41300 $SN_CEC_TABLE _ENABLE[0] = 1
SD41300 $SN_CEC_TABLE _ENABLE[1] = 1

41300[0]	$SN_CEC_TABLE_ENABLE	1
41300[1]	$SN_CEC_TABLE_ENABLE	1
41300[2]	$SN_CEC_TABLE_ENABLE	0

After executing the the file it is possible to see that the configuration was successful in the "Setup" operating area by pressing the "NC" softkey, followed by the “Leadscrew error” softkey and then “Axis +”, “Axis - “ to the relevant axis.



Notes:

Sag (bi-directional) Error Comp Data File Explanation

"$AN_CEC[m,n]=q" defines the compensation values at each compensation point starting at the minimum value to the maximum value incrementing by the step distance. "m" is the index of the compensation table, "n" is the index value starting at 0 for the 1st compensation point, "q" is the compensation value.

"$AN_CEC_INPUT_AXIS[m]=(p)" defines the name of the axis whose setpoint is to be used as the input for the compensation table, "m" is the index of the compensation table, "p" is the channel axis or the machine axis name.

"$AN_CEC_OUTPUT_AXIS[m]=(p)" defines the name of the axis to which the correction value is switched, "m" is the index of the compensation table, "p" is the channel axis or the machine axis name.

"$AN_CEC_STEP[m]=(p)" defines the distance between the compensation points, "m" is the index of the compensation table, "p" is the distance between the interpolation points. Together with the start and end position, the number of interpolation points is specified by the interpolation point distance.

"$AN_CEC_MIN[m]=p" defines the starting position for the basic axis where the compensation table begins where "m" is the index of the compensation table, "p" is the starting position.

"$AN_CEC_MAX[m]=p" defines the end position for the basic axis where the compensation table ends where "m" is the index of the compensation table, "p" is the end position.

"$AN_CEC_DIRECTION[m]=p" defines whether the compensation table should affect both traverse directions, or only the positive or negative traverse direction of the basic axis, "m" is the index of the compensation table, "p" sets whether the compensation is direction dependent.
p=0 = in both traverse directions
p=1= only in positive traverse direction
p=-1= only in negative traverse direction

"$AN_CEC_MULTI_BY_TABLE[m]=p" defines whether the current compensation value is multiplied by the corresponding compensation value of any other compensation table, "m" is the index of the compensation table, "p" is the number of the table to be used for the multiplication.

"$AN_CEC_IS_MODULO[m]=p" defines whether modulo compensation is to be implemented with a modulo rotary axis as basic axis, the compensation table used has to be modulo calculated as well, where "m" is the index of the compensation table, "p" sets whether or not the modulo function is used.
p=0 = no modulo function
p=1= compensation with modulo function

!The "sag compensation" function does not become active until the following conditions are
fulfilled:

!The function has been activated for the relevant machine axis (compensation axis) with
MD32710 $MA_CEC_ENABLE[<AXi>] = 1

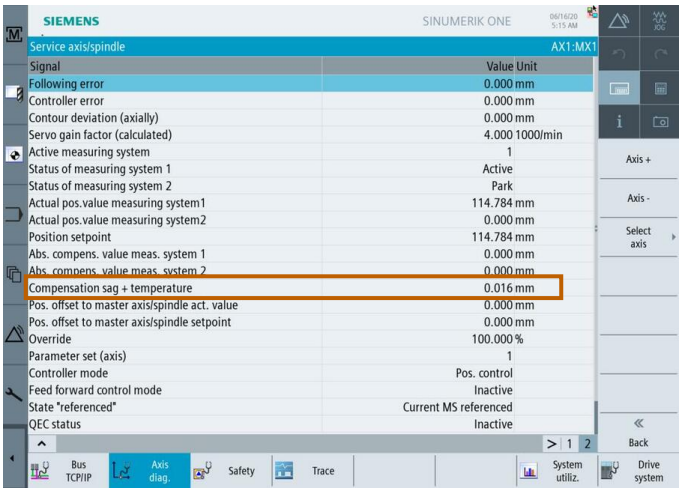
!The compensation values are stored in the static user memory and are active (POWER ON).

!Evaluation of the relevant compensation table has been enabled:
SD41300 $SN_CEC_TABLE_ENABLE[<t>] = 1

!The current measuring system of the base and compensation axes are referenced:

Display

The Sag + temperature compensation value that belongs to the current actual position is displayed in the "Diagnostics" operator area of the "Axis/Spindle Service" window. The "bi-directional comp" can be seen here.

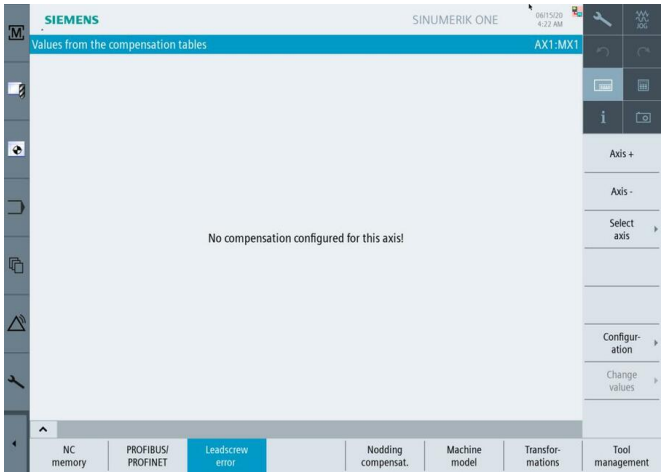


Notes:

Manual Configuration of the Compensation Table

Instead of transferring the compensation data via the part program or INI file to the NC, you can enter and check the compensation data for all axes or change the configuration in an input screen. The changed data is activated in the NC in one step. The machine data required when activating is set automatically. **MD18342[0][1] $MN_MM_CEC_MAX_POINTS** should be set prior to manual configuration

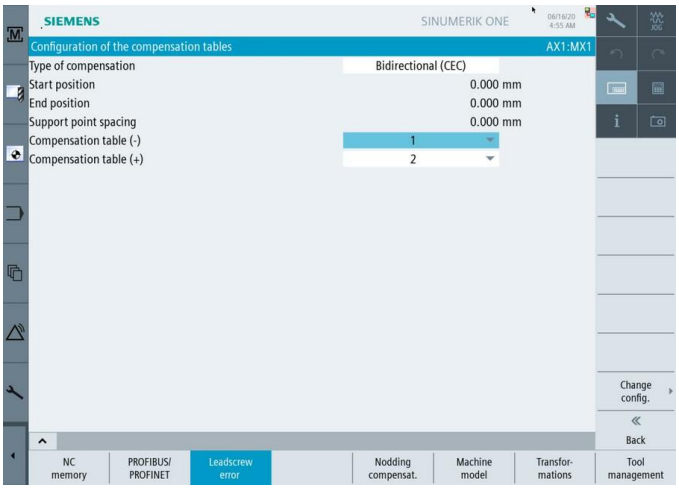
Select the "Setup" operating area and press the "NC" softkey, followed by the “Leadscrew error” softkey and then “Axis +”, “Axis - “ to the relevant axis.



Press the "Configuration" softkey to edit the compensation values for a selected axis.

The "Configuration of the Compensation Tables" window opens. Select the relevant table to be used for positive and negative directions.

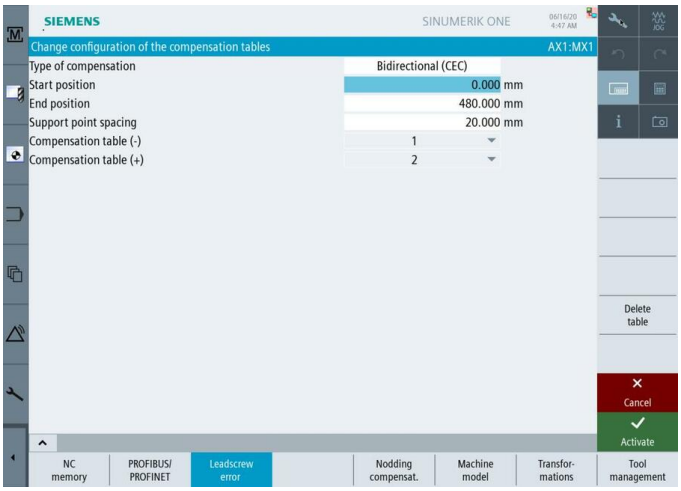
Press the “Change config” softkey



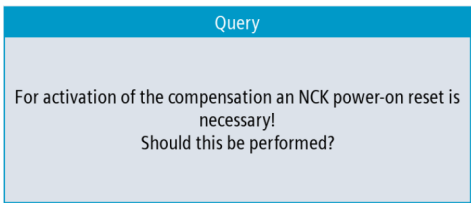
Enter the following:

- Type of compensation: Bidirectional (CEC).
- Start position, End position and Support (interpolation) point spacing.
- Correction table (-) Index of the table in the Negative direction
  Correction table (+) Index of the table in the Positive direction

Confirm your entries with the "Activate" softkey.



A dialog box to carry out a NC warm restart is then displayed to activate the compensation. Confirm with "OK".

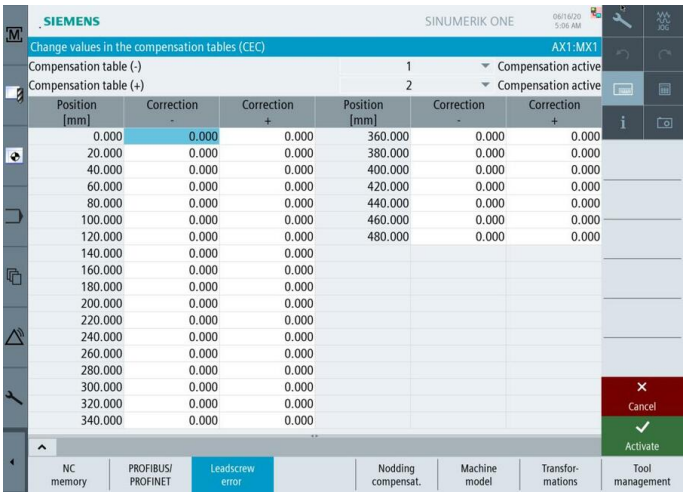


Notes:

The "Values of the Compensation Tables" window can now be opened.

Enter the desired compensation value at each interpolation point of the relevant table, table 1 in the negative direction and table 2 in the positive direction.

Press the "Activate" softkey again to transfer the compensation values to the NC.



Notes:

Temperature Compensation

Most machine tools are subject to continuously changing operating conditions. They are rarely maintained at steady state and the heat generated internally will vary significantly as the machine is used. When this is combined with the effect of environmental temperature variation the result is that the machine tool structure is exposed to complex and changing temperature distributions. The complex structure of a machine tool and the varying temperature distributions produce thermal deformations that are difficult to quantify or predict.

By activating the "temperature compensation" function, actual value changes due to temperature effects can be compensated on an axis-by-axis basis.

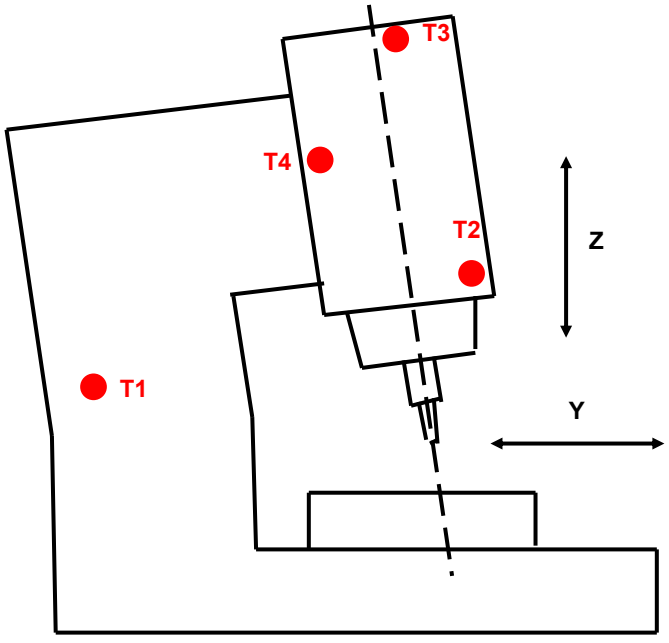
To provide effective temperature compensation, a number of temperature sensors for acquiring a temperature profile are needed in addition to the actual position data from existing encoders. Since temperature-dependent changes occur relatively slowly, the PLC can acquire and pre-process the temperature profile in a minutes cycle, for example.

In order to implement temperature compensation, the machine-tool manufacturer must take measurements at the Tool Centre Point (TCP) as temperature is rising and falling. Error curves /algorithms can then be be produced for different temperatures. This can take several days/weeks depending upon the machine type, and expansion.

It is thus possible for the machine-tool manufacturer to emulate the mathematical and technological relationship between the axis positions and temperature values via the PLC user program and thus calculate the various values for the temperature compensation.

The temperature compensation values are transferred to the NCK using the variable services LBP_WriteVar [FB3] from the PLC library

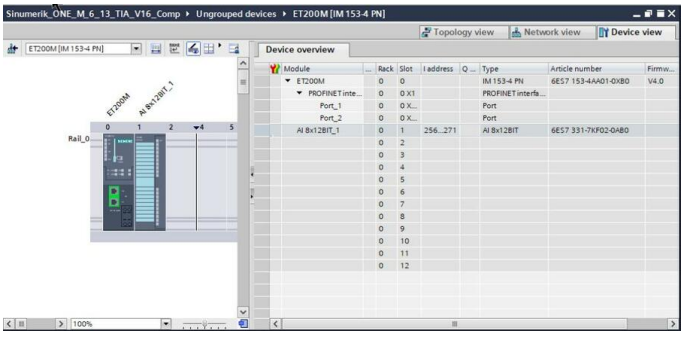
Below is a typical example where temperature compensation is required on a “C” frame vertical machining centre. Heat from high speed spindles etc. causes deformation in the structure requiring comp in Z and Y axes.



An example for VMC with 4 temperature sensors, T1 for ambient temperature and T2, T3, T4 to record temperature rise on different parts of the head.

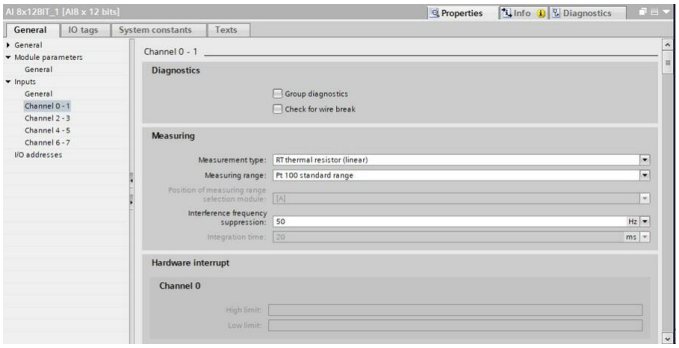
TIA S7 PLC Hardware Configuration

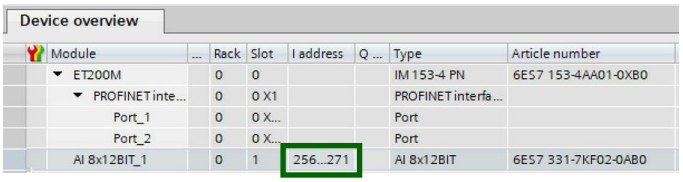
In this example an IM153-4PN 6ES7153-4AA01-0xB0 is configured on profinet, together with an 8 analogue input module 6ES7 331-1KF02-0AB0.
The 4 x Pt100 sensors are connected to the 8 analogue input module 6ES7 331-1KF02-0AB0, and configured as follows in the module properties.



Notes:

Various settings can be changed in the properties of the 8 analogue input module 6ES7 331-1KF02-0AB0, the measuring type RT Thermal resistor (linear) is selected as well as the measuring range Pt 100 standard range.





The input address for the 8 analogue input module starts at peripheral input word (PIW) 256.

Converting from analogue to useful values in PLC

Transfer to Data Block

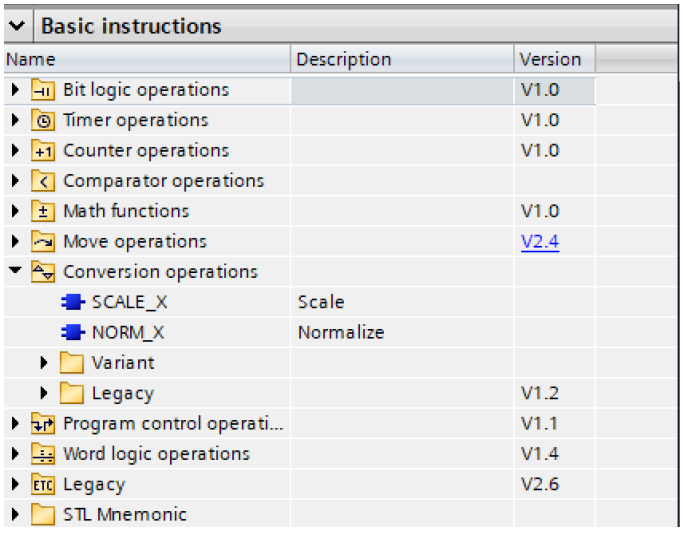
```
// Channel 1 -> Data block
     L    "columnT1Thermistor";        PIW 256
     T    “T1Thermistor”;               DB201.DBW    0

// Channel 2 -> Data block
     L    "headFrontT2Thermistor";     PIW 258
     T    “T2Thermistor”;               DB201.DBW    2

// Channel 3 -> Data block
     L    "headTopT3Thermistor";       PIW260
     T    “T3Thermistor”;               DB201.DBW    4

// Channel 4 -> Data block
     L    "headRearT4Thermistor";      PIW262
     T    “T4Thermistor”;               DB201.DBW    6
```

Convert/Scale raw values to useful/real values



The SFCs SCALE_X and NORM_X can be used in the OEMs PLC to get useful values and then use his algorithms to get the compensation value for each axis.

Notes:

Once the readings from the 4 x Pt100 sensors are converted to useful values the ambient temperature can be subtracted from each of the other 3 head sensor temperature values to obtain the temperature rise at that point.

The OEMs error curves/algorithm can then be used to decide how much compensation is required in each axis.

It is also a good idea to take the readings over several seconds and then average the values so that this has a filtering/ damping affect. A decision can also be made as to whether the compensation is implemented immediately, drip fed or added under certain conditions e.g. tool change.

The compensation values are transferred from the PLC to the axis specific setting data
43900 $SA_TEMP_COMP_ABS_VALUE
using LBP_WriteVar [FB3] from the PLC library, and the NC Variable Selector .

```
CALL  "LBP_writeVar" , "LBP_Writevar_DB"
 Req    :="TrigTemp CompTran"
 NumVar :=3
 Addr1  :="temp_comp".A1_SE_TEMP_COMP; X
 Unit1  :=
 Column1:=
 Line1  :=
 Addr2  :="temp_comp".A2_SE_TEMP_COMP; Y
 Unit2  :=
 Column2:=
 Line2  :=
 Addr3  :="temp_comp".A3_SE_TEMP_COMP, Z
 Unit3  :=
 Column3:=
 Line3  :=
```

Where the comp is being written to. (from NC Var Selector).

Amount of comp

```
Error  :="TempComp TranErr"
Done   :="TempComp TrnDone"
State  :="FB3 Comp TranStat"
SD1    :="comp_var".x_temp_val;   X comp
SD2    :="comp_var".y_temp_val;   Y comp
SD3    :="comp_var".z_temp_val;   Z comp
```

43900	$SA_TEMP_COMP_ABS_VALUE	0.012 mm	im

Temperature Compensation Type

The temperature compensation type “position-independent compensation” should be activated as shown below for the relevant axes

32750 $MA_TEMP_COMP_TYPE = 1

32750	$MA_TEMP_COMP_TYPE	1H

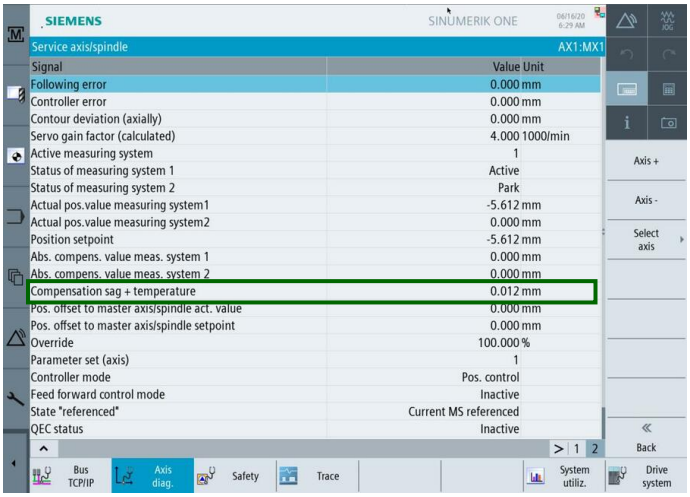
Bit editor

32750:$MA_TEMP_COMP_TYPE = 1H

- [x] Bit 0: Position-independent compensation active
- [ ] Bit 1: Position-dependent compensation active
- [ ] Bit 2: Offset active in tool direction

Display

The Sag + temperature compensation value that belongs to the current actual position is displayed in the "Diagnostics" operator area of the "Axis/Spindle Service" window.



A415: END

Notes:

A421

SINUMERIK ONE

Positioning Axis & Spindles

Module Description:

In addition to axes for machining a workpiece, modern machine tools can also be equipped with axes for auxiliary movements. Axes for auxiliary movements are traversed independently of the path axes at a separate, axis specific feedrate. In the past, many of these axes were moved hydraulically and started by an auxiliary function in the part program. With the closed-loop axis control implemented in the NC, the axis/axes can be addressed by name in the part program and its actual position displayed on the screen

Module Objective:

At the end of this module you will be able to commission/program positioning axes/spindles.

Content:

Introduction

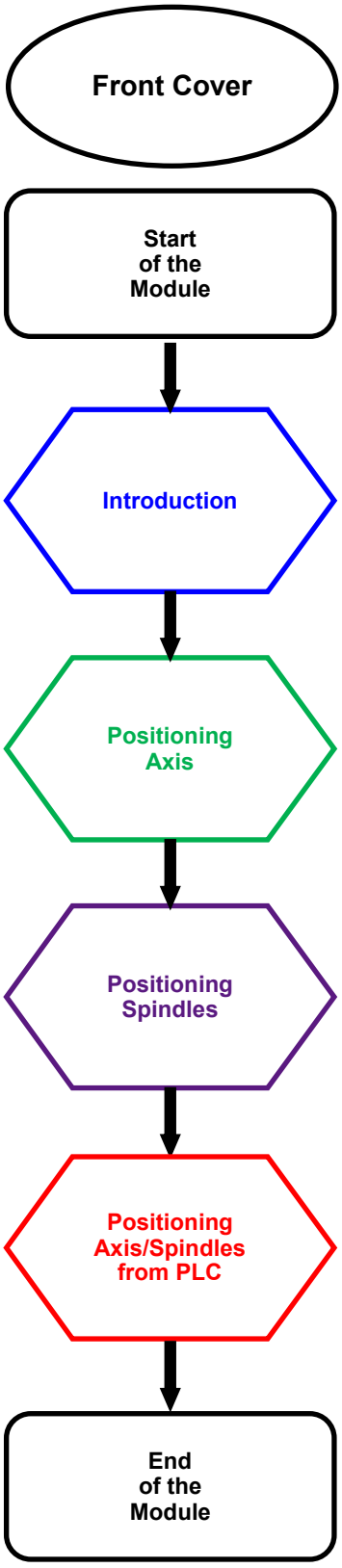
Positioning Axis

Positioning Spindles

Positioning Axis/Spindles fron PLC

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Introduction
Positioning Axis
Positioning Spindles
Positioning Axis/Spindles from PLC
End of the Module

Introduction

Positioning Axis

In addition to axes for machining a workpiece, modern machine tools can also be equipped with axes for auxiliary movements, e.g.:

- Axis for tool magazine
- Axis for tool turret
- Axis for workpiece transport
- Axis for pallet transport
- Axis for loader (also multi-axis)
- Axis for tool changer

The axes for the workpiece machining are called path axes. Within the channel they are guided by the interpolator such that they start simultaneously, accelerate, reach the end point and stop together.

Axes for auxiliary movements are traversed independently of the path axes at a separate, axis specific feedrate. In the past, many of these axes were moved hydraulically and started by an auxiliary function in the part program.

With the closed-loop axis control implemented in the NC, the positioning axis can be addressed by name in the part program and its actual position displayed on the screen.

Positioning Spindles

A spindle can also be operated as a path axis, synchronized axis or positioning axis using the address defined in the machine data. When the axis identifier is specified, the spindle is in axis mode. M70 switches the spindle directly to axis mode.

Axis / Spindle control from the PLC

With PLC axis/spindle control it is possible to move an axis or a spindle independently from all of the other axes. The travel motions are executed separate from the path axes.

Option

Additional 1 positioning axis/auxiliary spindle

6FC5800-0BB00-0Yx0

Notes:

Positioning Axis

Option

Additional 1 positioning axis/auxiliary spindle 6FC5800-0BB00-0Yx0	1

Function

The "positioning axes" function makes it easier to integrate axes for auxiliary movement into the control system:
● during programming:
The axes are programmed together with the axes for workpiece machining in the same part program, without having to sacrifice valuable machining time.
There are special (POS, POSA) traversing instructions.
● during program testing/start-up:
Program testing and start-up are performed simultaneously for all axes.
● during operation:
Operation and monitoring of the machining process commence simultaneously for all axes.
● during system configuring:
A second channel is not required.

Functional restrictions

Optional positioning axes/auxiliary spindles have fewer functions. The following functions are not possible:
● Using the axis as a path axis, geometry axis, or special path axis
● Rigid thread cutting and tapping

Motions and interpolations

Each channel has one path interpolator and at least one axis interpolator with the following interpolation functions:
● for path interpolator:
Linear interpolation (G1), circular interpolation (G2 / G3), spline interpolation, etc.
● for axis interpolator:
If a positioning axis is programmed, an axis interpolator starts in the control (with linear interpolation G1).

Positioning axis or concurrent positioning axis

When axes are provided for auxiliary movements on a machine tool, the required properties will decide whether the axis is to be:
● programmed in an internal part program
● programmed in the same part program as the machining operation
● started exclusively by the PLC during machining ("Concurrent positioning axis").

For “Concurrent positioning axis” the following axis MD should be set = 1

30450	$MA_IS_CONCURRENT_POS_AX	1

Commissioning

As standard, axes are defined as interpolating axes:

MD30460 $MA_BASE_FUNCTION_MASK bit 8 = 0

If an axis is to be operated as a positioning axis/auxiliary spindle, the value for bit 8 must be set to "1":

MD30460 $MA_BASE_FUNCTION_MASK bit 8 = 1

30460:$MA_BASE_FUNCTION_MASK = 100H

- ☐ Bit 0: Control Axis
- ☐ Bit 2: Diameter programming
- ☐ Bit 4: PLC control only
- ☐ Bit 5: Firmly assigned PLC axis
- ☐ Bit 6: Feed disable behavior
- ☐ Bit 7: Behavior "all axes stationary"
- ☑ Bit 8: Positioning axis / auxiliary spindle
- ☐ Bit 9: Response for PRESET

Notes:

Programming positioning axes (POS, POSA, FA, WAITP, WAITMC)

Positioning axes are traversed independently of the path axes at a separate, axis-specific feedrate. There are no interpolation commands. The POS/POSA commands are used to traverse the positioning axes and coordinate the motion sequences at the same time.

WAITP can be used to identify a position in the NC program where the program is to wait until an axis programmed with POSA in a previous NC block reaches its end position.
WAITMC loads the next NC block immediately when the specified wait marker is received.

Syntax

POS[<axis>]=<position>
POSA[<axis>]=<position>
FA[<axis>]=<value>
WAITP(<axis>) ; Programming in a separate NC block.
WAITMC(<wait marker>)

Meaning

POS/POSA:	Move positioning axis to specified position
	POS and POSA have the same functionality but differ in their block change behaviour:
	• POS delays the enabling of the NC block until the position has been reached.
	• POSA enables the NC block even if the position has not been reached.
	<axis>: Name of the axis to be traversed (channel or geometry axis identifier)
	<position>: Axis position to be approached
	Type: REAL
FA:	Feedrate for the specified positioning axis
	<axis>: Name of the axis to be traversed (channel or geometry axis identifier)
	<value>: Feedrate
	Unit: mm/min or inch/min or deg/min
	Note: Up to 5 FA values can be programmed for each NC block.

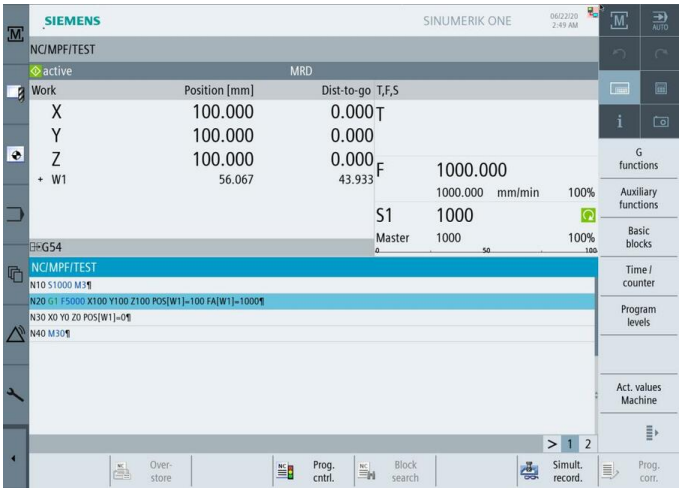
Examples

Example 1:

Positioning axis W1 is programmed together with path axes X, Y, Z, i.e. with the axes that are responsible for workpiece machining.

The commands for the positioning axes and path axes are included in the same NC block. Although they are programmed in the same NC block, the path and positioning axes are not interpolated together and do not reach their end point simultaneously.

The path axes move with a feedrate of 5 m/min, and the positioning axis W1 with a feedrate of 1 m/min. X, Y, Z axes reach position first in block N20, however because the program code POS is used for the positioning axis, block N30 is not processed until the W1 axis reaches its end position of 100mm.



If no feedrate is programmed for the positioning axis W1 then the value in the axis MD32060 $MA_POS_AX_VELO will be used.

32060	$MA_POS_AX_VELO	5000 mm/min

Notes:

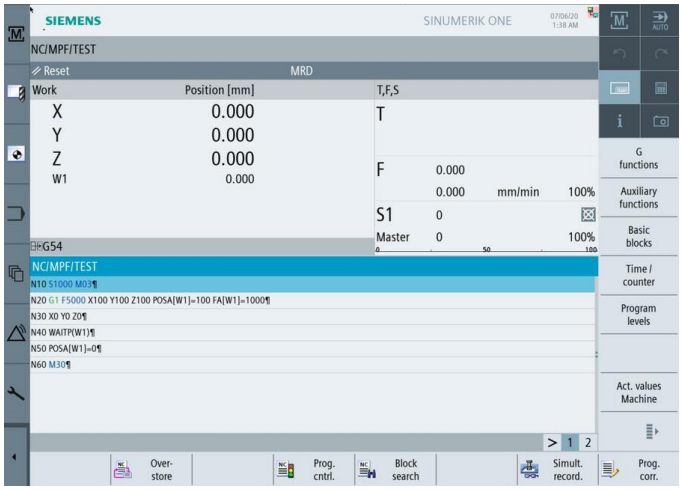
Example 2:

Positioning axis W1 is programmed together with path axes X, Y, Z, i.e. with the axes that are responsible for workpiece machining.

The commands for the positioning axes and path axes are included in the same NC block N20. Although they are programmed in the same NC block, the path and positioning axes are not interpolated together and do not reach their end point simultaneously.

The path axes move with a feedrate of 5 m/min, and the positioning axis W1 with a feedrate of 1 m/min. X, Y, Z axes reach position first in block N20, however because the program code POSA is used for the positioning axis, block N30 is processed immediately.

In block N40 the WAITP(W1) means that block N50 will only be processed once the W1 axis has reached its end position from block N20 (100mm).



Notes:

Programming Positioning of spindles (SPOS, SPOSA, M19, M70, WAITS)

Function

SPOS, SPOSA or M19 can be used to move spindles to specific angular positions, e.g. during tool change.
SPOS, SPOSA and M19 induce a temporary switchover to position-controlled mode until the next M3/M4/M5

Positioning in axis mode

The spindle can also be operated as a path axis, synchronized axis or positioning axis at the address defined in the machine data. When the axis identifier is specified, the spindle is in axis mode. M70 switches the spindle directly to axis mode.

Synchronization

In order to synchronize spindle movements, WAITS can be used to wait until the spindle position is reached.

Syntax

Position spindle:

SPOS=<value>/SPOS[<n>]=<value>
SPOSA=<value>/SPOSA[<n>]=<value>
M19/M<n>=19

Switch spindle over to axis mode:

M70/M<n>=70

Synchronize spindle movements:

WAITS/WAITS(<n>,<m>) ; Programming in a separate NC block.

Meaning

SPOS/ SPOSA:	Set spindle to specified angle		
	SPOS and SPOSA have the same functionality but differ in their block change behaviour:		
	• SPOS delays the enabling of the NC block until the position has been reached.		
	• SPOSA enables the NC block even if the position has not been reached.		
	<n>:	Number of the spindle to be positioned.	
		If a spindle number is not specified or if the spindle number is set to "0", SPOS or SPOSA will be applied to the master spindle.	
	<value>:	Angular position to which the spindle is to be set.	
		Unit: degrees	
		Type: REAL	
		The following options are available for programming the position approach mode:	
		=AC(<value>):	Absolute dimensions
			Range of values: 0 … 359.9999
		=IC(<value>):	Incremental dimensions
			Range of values: 0 … ±99 999.999
		=DC(<value>):	Approach absolute value directly
		=ACN(<value>):	Absolute dimension, approach in negative direction
		=ACP(<value>):	Absolute dimension, approach in positive direction
		=<value>:	as DC(<value>)
M<n>=19:	Set the master spindle (M19 or M0=19) or spindle number <n> (M<n>=19) to the angular position preset with SD43240 $SA_M19_SPOS with the position approach mode preset in SD43250 $SA_M19_SPOSMODE.		
	The NC block is not enabled until the position has been reached.		
M<n>=70:	Switch the master spindle (M70 or M0=70) or spindle number <n> (M<n>=70) over to axis mode.		
	No defined position is approached. The NC block is enabled after the switchover has been performed.		

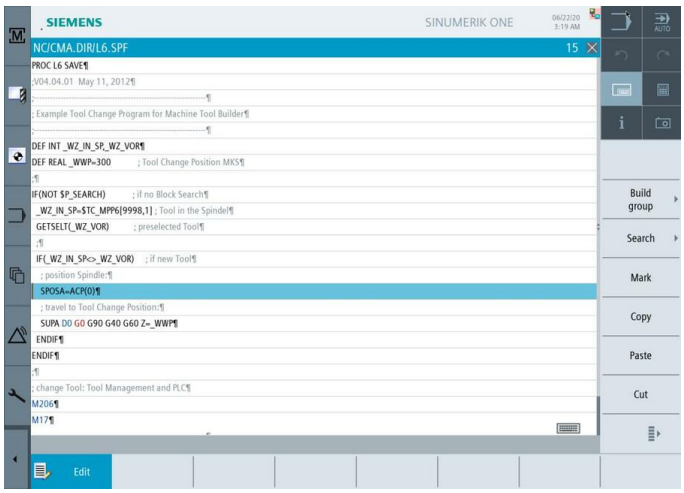
Notes:

WAITS:	Synchronization command for the specified spindle(s)	
	The subsequent blocks are not processed until the specified spindle(s) programmed in a previous NC block with SPOSA has (have) reached its (their) end position(s) (with exact stop fine).	
	WAITS after M5:	Wait for the specified spindle(s) to come to a standstill.
	WAITS after M3/M4:	Wait for the specified spindle(s) to reach their setpoint speed.
	<n>,<m>:	Numbers of the spindles to which the synchronization command is to be applied.
		If a spindle number is not specified or if the spindle number is set to "0", WAITS will be applied to the master spindle.

Examples

Example 1: In this Tool Change cycle L6.SPF the main spindle is positioned to zero degrees in a positive direction with SPOSA=ACP(0). Because SPOSA is used the cycle continues with the next block and it does not wait for the spindle to finish positioning. The Z axis now moves at rapid to the Z axis tool change position. At this point if the spindle has not finished positioning the WAITS command is used so that the cycle does not continue until the spindle is positioned.

Depending upon the mechanical design of the spindle it can be a good idea to always approach the spindle tool change position in the same direction to eliminate any backlash etc.



Notes:

Example 2:

Drill 4 cross holes in turned part.

Cross holes are to be drilled in this turned part. The main spindle is stopped at zero degrees and then successively turned through 90°, the tooling spindle S2 then drills the holes.

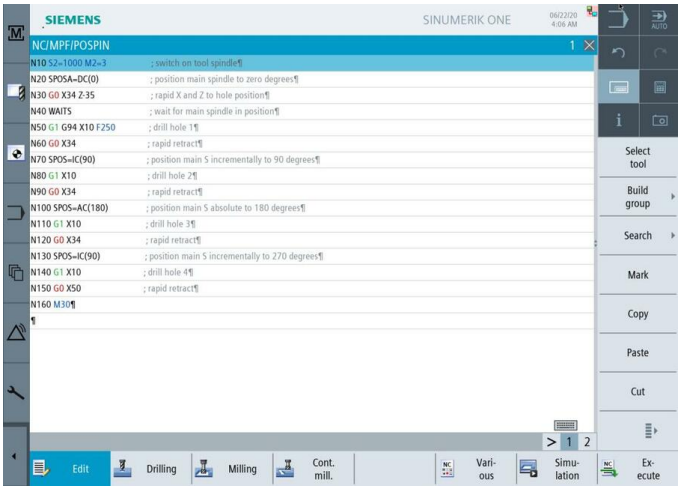
The example shows the main spindle being positioned by approaching absolute value directly in block N20. with SPOSA=DC(0), block N30 will start without waiting for the main spindle to be in position.

The program waits for the main spindle to be in position with block N40 with WAITS

Moving incrementally 90 degrees to absolute position 90 degrees in block N70 with SPOS=IC(90)

Moving to absolute position 180 degrees in block N100 with SPOS=AC(180)

Moving incrementally 90 degrees to absolute position 270 degrees in block N130 with SPOS=IC(90)



Notes:

Controlling positioning axes/spindles from the PLC

With PLC axis control it is possible to move positioning axes/ spindles independently to all other axes.

Using the PLC library block LBP_CtrlAxisSpindle[FC18], SINUMERIK ONE positioning axes/spindles can be started from the PLC.

The behaviour of a PLC controlled axis can be manipulated in with machine data:

MD30460 $MA_BASE_FUNCTION_MASK

30460:$MA_BASE_FUNCTION_MASK = 170H

- ☐ Bit 0: Control Axis
- ☐ Bit 2: Diameter programming
- ☑ Bit 4: PLC control only
- ☑ Bit 5: Firmly assigned PLC axis
- ☑ Bit 6: Feed disable behavior
- ☐ Bit 7: Behavior "all axes stationary"
- ☑ Bit 8: Positioning axis / auxiliary spindle
- ☐ Bit 9: Response for PRESET

The example shows a permanently assigned and PLC controlled auxiliary axis

● Bit 4 = 1
The axis is exclusively controlled by the PLC.
● Bit 5 = 1
The axis is a permanently assigned PLC axis.
● Bit 6 = 1
The channel-specific NC/PLC interface signal:
DB21, ... DBX6.0 ("feed disable") is not active for the axis if this is a PLC-controlled axis.
● Bit 7= 1
The channel-specific NC/PLC interface signal:
DB21, ... DBX36.3 ("all axes stationary") is set independently of an axis if this is a PLC-controlled axis.
● Bit 8 = 1:
The axis is a positioning axis / auxiliary spindle

For a PLC-controlled axis:
● The channel-specific NC/PLC interface signal DB21, ... DBX6.0 ("feed disable"), is active if bit 6 = 0
● The channel-specific NC/PLC interface signal DB21, ... DBX36.3 ("all axes stationary") is only set if bit 7 = 0

Starting a positioning axis/spindle from the PLC

Activation from PLC
When a positioning axis is activated from the PLC, FC18 is called and supplied with the following parameter data:

● Axis name or axis number
● End position
● Feedrate (with feedrate setting = 0, the feedrate is determined by the setting in machine data MD32060 $MA_POS_AX_VELO):
● Absolute coordinate (G90), incremental coordinate (G91), absolute coordinate along the shortest path for rotary axes (rotary axis name = DC(value)

Prerequisites for the PLC Function:

HMI Requirement:
None

Drive requirement (Sinamics):
None

NCK Requirement:
None

PLC Requirement:
Parameterisation of the FC18 in the PLC

Notes:

Parameter description of FC18

Signal	Type	Format	Range of values	Remark
Start	I	BOOL		Start spindle control from PLC
Stop	I	BOOL		Stop spindle control from PLC
Funct	I	BYTE	1 to B#16#0B	1: Position spindle 2: Rotate spindle 3: Oscillate spindle 4: Indexing axis 5: Positioning axis metric 6: Positioning axis inch 7: PosAxis metric with hand-wheel override 8: PosAxis inch with hand-wheel override 9: Rotate spindle with automatic gear stage selection A: Rotate spindle with constant cutting rate (m/min) B: Rotate spindle with constant cutting rate (feet/min)

Signal	Type	Format	Range of values	Remark
Mode	I	BYTE	0 to 5	0: Pos to absolute pos 1: Pos incrementally 2: Pos shortest path 3: Pos absolute, positive approach direction 4: Pos absolute, negative approach direction 5: Rotational direction as for M4
AxisNo	I	INT	1 - 31	No. of axis/spindle to be traversed
Pos	I	REAL	0,1469368 I -38 to 0,1701412 I +39	Rotary axis: Degrees Indexing axis: Indexing position Linear axis: mm or inches
FRate	I	REAL	0,1469368 I -38 to 0,1701412 I +39	Rotary axis and spindle: rev/min Linear axis: mm/min or inches/min
InPos	Q	BOOL		1 = Position reached, or function executed
Error	Q	BOOL		1 = error
State	Q	BYTE	0 to 255	Error code

Example 1:

An example showing the PLC using FC18 to move a positioning axis incrementally 500mm at a feedrate of 1000 mm/min

Funct: = B#16#5 = Positioning axis metric

Mode: = B#16#1 = Position incrementally

AxisNo: = Axis number of the Positioning axis

Pos: = The incremental distance = 500 mm

FRate: = Feedrate for the axis move = 1000 mm/min.

```
FUNCTION FC 114 : VOID
TITLE =
VERSION : 0.1
BEGIN
NETWORK
TITLE = Pos Axis Inc FC18

    A   I  81.6; //
    FP  M 141.0; // Transition marker
    S   M 141.1;
    A   M 141.3;
    FP  M 141.4; // Transition Marker
    R   M 141.1;
    R   M 141.7;
    A   I  81.7;
    FP  M 141.6; // Transition Marker
    S   M 141.7;
    CALL "LBP_CtrlAxisSpindle" (
        Start                 := M141.1,
        Stop                  := M141.7,
        Funct                 := B#16#5,
        Mode                  := B#16#1,
        AxisNo                := 5,
        Pos                   := 5.000000e+002,
        FRate                 := 1.000000e+003,
        InPos                 := M  141.3,
        Error                 := M  141.5,
        State                 := MB 142);
END_FUNCTION
```

Notes:

Example 2:

An example showing the PLC using FC18 to position the spindle to absolute position 0.000 degrees. The feedrate is not specified therefore the value in MD32060 $MA_POS_AX_VELO will be used.

Funct: = B#16#1 = Positioning spindle

Mode: = B#16#0 = Position to absolute position

AxisNo: = Axis number of the spindle

Pos: = The absolute position = 0 degrees

FRate: = Spindle rev/min for axis move = 0. The speed is not specified therefore the value in MD32060 $MA_POS_AX_VELO will be used.

```
FUNCTION FC 114 : VOID
TITLE =
VERSION : 0.1
BEGIN
NETWORK
TITLE = Pos Axis Abs FC18

    A   I  81.6; //
    FP  M 141.0; // Transition marker
    S   M 141.1;
    A   M 141.3;
    FP  M 141.4; // Transition Marker
    R   M 141.1;
    R   M 141.7;
    A   I  81.7;
    FP  M 141.6; // Transition Marker
    S   M 141.7;
    CALL "LBP_CtrlAxisSpindle" (
        Start                   := M141.1,
        Stop                    := M141.7,
        Funct                   := B#16#1,
        Mode                    := B#16#0,
        AxisNo                  := 4,
        Pos                     := 0.000000e+000,
        FRate                   := 0.000000e+000,
        InPos                   := M  141.3,
        Error                   := M  141.5,
        State                   := MB 142);
END_FUNCTION
```

Notes:

A421: END

Notes:

A509

SINUMERIK ONE

Setup Turning

Module Description:

In this module a simple turning machine with the axes X and Z, a main spindle C as well as a driven tool CS3 is to be set up for turning and the subtechnology milling. The individual setup steps are to be checked from time to time to show the result of the parameterization.
The preparation of the tool list, the cycle selections, the search run and various basic settings are also part of the training scope. There will be a detailed description of the selections to be made for JOG-TSM, TRANSMIT/TRACYL and for the main .

Module Objective:

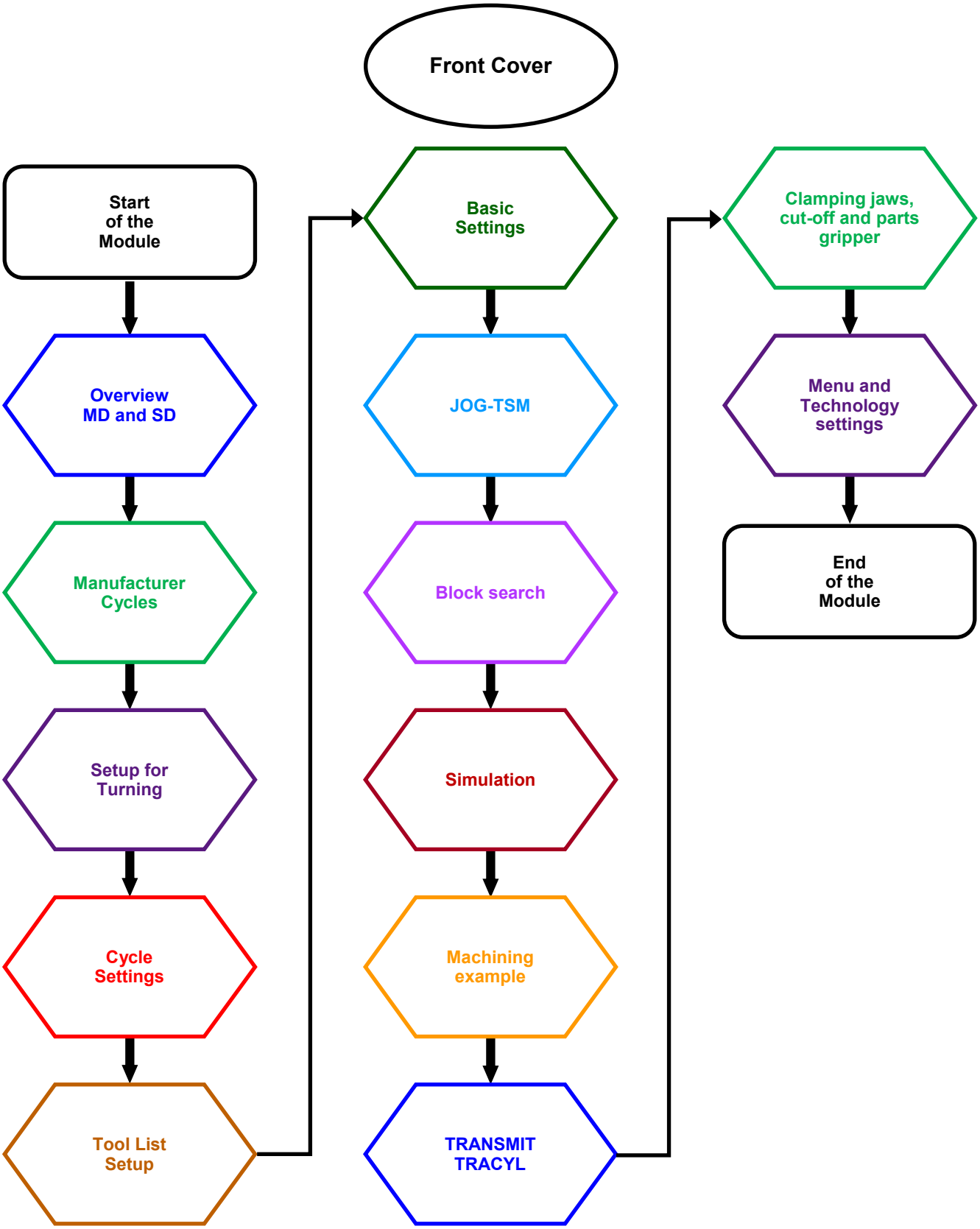
By means of this module you will get to know the meaning of the Machine and Setting data to be used as well as the various means available in order to configure the machine to suit the particular requirements. The functions will be parameterized in several blocks and subsequently tested regarding the function.

Content:

Sinumerik

Overview MD and SD
Manufacturer Cycles
Setup for Turning
Cycle settings
Tool list setup
Basic settings
JOG-TSM
Block search
Simulation
Machining example
TRANSMIT/TRACYL
Clamping jaws, cut-off and parts gripper
Menu and Technology settings

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Overview MD and SD
Manufacturer Cycles
Setup for Turning
Cycle Settings
Tool List Setup
Basic Settings
JOG-TSM
Block search
Simulation
Machining example
TRANSMIT TRACYL
Clamping jaws, cut-off and parts gripper
Menu and Technology settings
End of the Module

[Overview Machine data and setting data]

Basic settings Machine data/setting data

MD	Identifier	Settings	Comment	Description
10260	$MN_CONVERT_SCALING_SYSTEM	1	Exact	Enable basic system conversion
10602	$MN_FRAME_GEOAX_CHANGE_MODE	1	Exact	Frames when changing geometry axes
10610	$MN_MIRROR_REF_AX	0	Exact	Reference axis for mirroring
10714	$MN_M_NO_FCT_EOP	32	Variable	M function for spindle active after reset
10722	$MN_AXCHANGE_MASK	Bit 2 = 1	Exact	Parameters for axis replacement behaviour
11410	$MN_SUPPRESS_ALARM_MASK	Bit 0 = 1 Bit 1 = 1 Bit 15 =1 Bit 20 =1	Variable	Bit 0: 15110: REORG not possible Bit 1: 10763 The path component of the block in the contour plane is zero Bit 15: 5000 Communication order cannot be executed Bit 20: 2900 Reboot is delayed
11450	$MN_SEARCH_RUN_MODE	Bit 0 = 1 Bit 1 = 1 Bit 2 = 1 Bit 6 = 1	Exact	Bit 0: Machining is stopped with the loading of the last action block after search run Bit 1: Automatic ASUP start after output of action blocks Bit 2: The output of the auxiliary functions in the action blocks is suppressed Bit 6: Block search with simple, computing-time-optimized algorithms
11470	$MN_REPOS_MODE_MASK	Bit 3 = 1	Exact	Mask for support of special alarm outputs
17530	$MN_TOOL_DATA_CHANGE_COUNTER	1FH	Exact	Mark tool data change for HMI
18080	$MN_MM_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1	Exact	Bit 0: Memory for TM-specific data is provided Bit 1: Memory for monitoring data (WZMO) is provided Bit 3: Memory to consider adjacent location is provided
20100	$MC_DIAMETER_AX_DEF	X	Variable	Geometry axis with transverse axis function
20110	$MC_RESET_MODE_MASK	Bit 0 = 1 Bit 6 = 1 Bit 14 =1 Bit 16 =0	Exact Exact Exact Variable	Bit 0 = 1 G codes acc. to MD20150 Bit 6 = 1 Current setting for active tool length offset is retained after reset/end-of-part-program Bit 14 = 1 The current setting of the basic frame is retained Bit 16 = 0 Initial setting for master spindle acc. to MD20090
20124	$MC_TOOL_MANAGEMENT_TOOLHOLDER	1	Exact	Tool holder number
20128	$MC_COLLECT_TOOL_CHANGE	0	Exact	No tool change commands after search run
20130	$MC_CUTTING_EDGE_RESET_VALUE	1	Exact	Tool and cutting edge remain activated
20150	$MC_G_CODE_RESET_VALUE	[5] = 2 [15] = 3 [21] = 2 [28] = 2 [48] = 3	Exact Variable Variable Exact Variable	Group 6: Plane selection 2 = 18 Group 16: Feedrate override 3 = CFIN Group 22: Tool offset type 2 = CUT2DF Group 29: Radius/diameter programming 2 = DIAMON Group 49: Point-to-point motion 3 = PTPG0
20152	$MC_G_CODE_RESET_MODE	[5] = 0 [7] = 1 [12] = 0 [14] = 1 [15] = 0 [21] = 0	Exact Exact Exact Exact Exact Variable	Group 6: Plane selection remains active after Reset Group 8: Zero point offset remains active after Reset Group 13: Work piece dimensioning inch/metric default after reset Group 15: Type of feed remains active after Reset Group 16: Feed correction remains active after Reset Group 22: Tool correction type default after reset
20202	$MC_WAB_MAXNUM_DUMMY_BLOCKS	10	Minimum	maximum number of blocks w/o traversing movement with SAR
20240	$MC_CUTCOM_MAXNUM_CHECK_BLOCKS	4	Minimum	Blocks for look-ahead contour calculation with TRC
20250	$MC_CUTCOM_MAXNUM_DUMMY_BLOCKS	5	Minimum	maximum number of blocks without traversing motion in TRC

Notes:

Basic settings Machine data/setting data

MD	Identifier	Settings	Comment	Description
20310	$MC_TOOL_MANAGEMENT_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 14 =1 Bit 16 =1	Exact Exact Exact Exact Exact	Bit 0: Tool management active Bit 1: Tool monitoring function active Bit 3: Consider adjacent location active Bit 14: automatic tool change during RESET and Start Bit 16: Programming T='Location number'
20320	$MC_TOOL_TIME_MONITOR_MASK	Bit 0 = 1	Variable	Time monitoring for tool in tool holder 1
20360	$MC_TOOL_PARAMETER_DEF_MASK	Bit 0 = 1	Variable	Definition of tool parameters
20734	$MC_EXTERN_FUNCTION_MASK	Bit 3 = 1	Variable	Errors in ISO scanner are not output, the block is transferred to the Siemens translator.
22550	$MC_TOOL_CHANGE_MODE	0	Exact	Tool change with T function
24006	$MC_CHSFRAME_RESET_MASK	Bit 0 = 1 Bit 5 = 0	Exact	Active system frames after reset Bit 0 = 1 Actual value setting and scratching remains active Bit 5 = 0 System frame for cycles is deleted
24007	$MC_CHSFRAME_RESET_CLEAR_MASK	Bit 0 = 0 Bit 5 = 1	Exact	Deletion of system frames after reset Bit 0 = 0 Actual value setting and scratching not delete Bit 5 = 1 System frame for cycles delete
24008	$MC_CHSFRAME_POWERON_MASK	Bit 0 = 0 Bit 5 = 0	Exact	Reset channel system frames after power on Bit 0 = 0 Actual value setting and scratching not reset Bit 5 = 0 System frame for cycles not reset
24030	$MC_FRAME_ACS_SET	1	Exact	Adjustment of SZS coordinate system
24040	$MC_FRAME_ADAPT_MODE	7H	Exact	Adaptation of active frames
27860	$MC_PROCESSTIMER_MODE	Bit 0 = 1 Bit 1 = 1 Bit 4 = 1 Bit 5 = 1 Bit 6 = 1 Bit 8 = 1	Variable	Bit 0: Measurement of total operating time is active for all part programs Bit 1: Measurement of current program runtime is active Bit 4: Measurement also with active dry run feed Bit 5: Measurement also with program test Bit 6: $AC_CYCLE_TIME is not deleted on start by ASUP and PROG_EVENT Bit 8: AC_CYCLE_TIME will be deleted with GOTOS
27880	$MC_PART_COUNTER	Bit 0 = 1 Bit 8 = 1 Bit 11 =1	Variable	Bit 0: Counter $AC_REQUIRED_PARTS is activated Bit 8: Counter $AC_ACTUAL_PARTS is active Bit 11: Increment $AC_ACTUAL_PARTS with GOTOS
28000	$MC_MM_REORG_LOG_FILE_MEM	75	Minimum	Memory space for REORG (DRAM)
28010	$MC_MM_NUM_REORG_LUD_MODULES	20	Minimum	Number of blocks for local user variables in REORG (DRAM)
28082	$MC_MM_SYSTEM_FRAME_MASK	Bit 0 = 1 Bit 5 = 1 Bit 6 = 1	Exact	Setting up system frames (SRAM) Bit 0 = 1 System frame for actual value and scratching Bit 5 = 1 System frame for cycles Bit 6 = 1 System frame for transformations
28083	$MC_MM_SYSTEM_DATAFRAME_MASK	Bit 0 = 1 Bit 5 = 0	Exact	Setting up data management system frames (SRAM) Bit 0 = 1 System frame for actual value and scratching Bit 5 = 0 System frame for cycles
28450	$MC_MM_TOOL_DATA_CHG_BUFF_SIZE	400	Exact	Buffer for tool data changes (DRAM)
35040	$MA_SPIND_ACTIV_AFTER_RESET	2	Exact	Own spindle RESET
42440	$SC_FRAME_OFFSET_INCR_PROG	0	Exact	Zero offsets in frames
42442	$SC_TOOL_OFFSET_INCR_PROG	0	Exact	Tool length offsets
42528	$SC_CUTCOM_DECEL_LIMIT	1	Variable	Feed lowering on circles with tool radius compensation
42940	$SC_TOOL_LENGTH_CONST	18	Exact	Change of tool length components with change of active plane
42950	$SC_TOOL_LENGTH_TYPE	2	Exact	Assignment of tool length compensation independent of tool type



Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
51024	$MNS_BLOCK_SEARCH_MODE_MASK_JS	Bit 0 = 1 Bit 1 = 1		Bit mask for available search modes (ShopMill/ShopTurn) Bit 0: Block search with calculation without approach Bit 1: Block search with calculation with approach Bit 3: Skip EXTCALL programs Bit 5: Block search with test run Bit 6 Block search with test run without approach
51028	$MNS_BLOCK_SEARCH_MODE_MASK	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1 Bit 4 = 1		Bit mask for available block search modes Bit 0: Block search with calculation without approach Bit 1: Block search with calculation with approach Bit 3: Skip EXTCALL programs Bit 4: Block search without calculation Bit 5: Block search with test run Bit 6 Block search with test run without approach
51038	$MNS_SET_ACT_VALUE	1		Set actual value selection 0 = Set actual value is not offered. 1 = If a user frame (settable work offset e.g. G54) is active, it will be used. In G500 Set actual values is not offered (system frame is no longer used).
51039	$MNS_PROGRAM_CONTROL_MODE_MASK	Bit 0 = 1		Options for machine - program influence: Bit 0: Program test function available
51040	$MNS_SWITCH_TO_MACHINE_MASK	0H		Automatic operating area switchover to machine Bit 0: When selecting a program in the program manager, the operating area does not automatically switch over to Machine. Bit 1: When switching the type of operation via the MCP, the operating area is not automatically switched over to Machine. Bit 2: Do not switch automatically in Programs during program selection. Bit 3: No auto. Start of block search on selection / execution under Programs.
51226	$MNS_FUNCTION_MASK_SIM	0H		Function mask Simulation Bit 0: No automatic start on simulation selection Bit 1: Deactivate simulation Bit 4: Ignore frame components in X and Y (compatibility) Bit 5: Turning tool can rotate in the tool spindle Bit 6: Enable handwheel as simulation override Bit 7: Interpretation of handwheel values as absolute values Bit 10: Hide zero point symbol Bit 22: Machine model for simulation from kinematic chain (also without collision avoidance)
51228	$MNS_FUNCTION_MASK_TECH	0H		Function mask Cross-technology Bit 0: G code programming without multi-channel data Bit 1: Enable print function of editor Bit 2: With external program selection, test program

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
52000	$MCS_DISP_COORDINATE_SYSTEM	34		0: Vertical lathe, right hand head 6: Vertical lathe, left hand head 33: Horizontal lathe with machining in front of the turning centre 34: Horizontal lathe with machining behind the turning centre
52005	$MCS_DISP_PLANE_MILL	0		Milling: 0: Plane selection on the operator panel
52006	$MCS_DISP_PLANE_TURN	18		Turning: Always G18
52200	$MCS_TECHNOLOGY	1	Exact	Technology 0: no specific configuration 1: turning 2: milling
52201	$MCS_TECHNOLOGY_EXTENSION	2		Extended technology 0: no specific configuration 1: turning: extended technology Turning for Milling 2: milling
52206	$MCS_AXIS_USAGE	[2] = 3 [3] = 1		Meaning of the axes 0 = no special meaning; 1 = tool spindle (driven tool); 2 = C axis of the tool spindle (driven tool); 3 = main spindle (turning); 4 = C axis of the main spindle (turning); 5 = counter spindle (turning); 6 = C axis of the counter spindle (turning); 7 = linear axis of the counter spindle (turning); tailstock (turning); 9 = steady rest (turning); 10 = B axis (turning), 12 = B axis of the counter spindle (turning), 13 = Traverse travel X of the counter spindle (turning)
52207	$MCS_AXIS_USAGE_ATTRIB	0H		Axis attributes Bit 0: Rotates around the 1st geometry axis (applies to rotary axes) Bit 1: Rotates around the 2nd geometry axis (applies to rotary axes) Bit 2: Rotates around the 3rd geometry axis (applies to rotary axes) Bit 3: Positive direction of rotation is counter-clockwise (applies to rotary axes) Bit 4: Displayed direction of rotation for M3 is counter-clockwise (applies to spindles) Bit 5: Direction of rotation M3 corresponds to rotary axis minus (applies to spindles) This bit must be set in the same way as a PLC bit. For ONE / 840D sl: PLC bit DBnn.DBX17.6, nn = 31 + machine axis index For 828D: PLC bit DB38xx.DBX2001.6, xx is machine axis index Bit 6: Show rotary axis as offset target for measuring Bit 7: Offer rotation axis in position pattern Bit 8: Offer rotation axis to accept blank (at milling machines) Bit 9: Spindle is not SPOS-capable Bit 10: Rotary axis rotates around 1st geometry axis (for position pattern only) Bit 11: Rotary axis rotates around 2nd geometry axis (for position pattern only) Bit 12: Rotary axis rotates around 3rd geometry axis (for position pattern only) Bit 13: Position of the axis for blank selection from active WO (for rotary axes) Bit 14: Enable spindle for teach-in in axis mode

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
52210	$MCS_FUNCTION_MASK_DISP	Bit 0 = 1 Bit 1 = 1 Bit 3 = 1		Function mask Display Bit 0: Measuring system for programs always in the base system Bit 1: Front view for turning in school coordinate system Bit 2: Hide “T,S,M” soft key in JOG area Bit 3: Generate automatic end-of-program in MDI (with the “Delete blocks” soft key) Bit 4: Show follow-on tool in T,F,S window Bit 5: Hide softkey “Actual machine values“ Bit 6: Hide tool radius/diameter in T,F,S window Bit 7: Hide tool length in T,F,S window Bit 8: Hide tool icon in T,F,S window Bit 9: Display selection ”Simple input” Bit 10: Do not offer channel in job lists Bit 11: For Work activate the display of the logical spindles Bit 12: Hide mold making view for G code Bit 13: Hide channel axes which are currently located in another channel Bit 14: Display overlay of DRF under machine Bit 15: Display overlay of tool ($AA_TOFF) under machine Bit 16: Select JOG axis via HMI Bit 17: Select increment via HMI Bit 18: Offer select “With set stop override to0%” with configured stop
52212	$MCS_FUNCTION_MASK_TECH	Bit 3 = 1	Exact	Function mask Cross-technology Bit 0: Enable swivel plane/align tool with tool carrier Bit 1: No optimized travel along software limit switches (ShopMill/ShopTurn) Bit 2: Special approach logic for step drill (ShopTurn) Bit 3: Call block search cycle for ShopMill/ShopTurn Bit 4: Approach logic through cycle (ShopTurn) Bit 5: Call block search cycle for SERUPRO Bit 6: Work offset ZV cannot be entered as an absolute value (ShopTurn) Bit 7: Detect the expiry of tool lives (ShopMill/ShopTurn) Bit 8: Manual machine (ShopMill/ShopTurn) Bit 9: Selection/deselection of work offset via softkey Bit 10: Enable swivel plane/align tool using kinematic transformation Bit 11: Switch off layer check for drilling and milling tools (ShopTurn) Bit 13: Approach tool change point Y in Machine (ShopTurn) Bit 14: With approach logic only check tool position with cutting edge (ShopTurn) Bit 15: Do not set $MC TOOL_CARRIER_RESET_VALUE after block search Bit 16: Do not automatically execute swivel plane after block search Bit 17: Do not automatically execute align turning tool after block search Bit 18: Enable Determine total load : empty axis + workpiece
52214	$MCS_FUNCTION_MASK_MILL	0H		Function mask Milling Bit 0: Enable cylinder surface transformation (ShopMill) Bit 1: Offer fixed table for blank clamping (at milling machines) Bit 2: Special handling of slide mill/saw for polyhedron Bit 3: Enable inside/rear machining (ShopTurn) Bit 4: Enable spindle clamping (C axis) (ShopTurn) Bit 5: Enable spindle control of tool spindle via surface Bit 6: Enable spindle control of turning spindle via surface Bit 7: Offer fixed table for blank clamping (on boring mills) Bit 8: Enable damping braking (C axis) (ShopTurn)

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
52216	$MCS_FUNCTION_MASK_DRILL	0H		Function mask Drilling Bit 0: CYCLE84 Unhide technology fields Bit 1: CYCLE840 Unhide technology fields Bit 2: Thread tapping also without encoder (ShopMill/ ShopTurn) Bit 3: Constant cutting speed in rel. to diameter of centering (ShopMill/ShopTurn)
52218	$MCS_FUNCTION_MASK_TURN	0H		Function mask Turning Bit 0: Enable zoom under manual for tool measurement Bit 1: Enable parts gripper for cut-off Bit 2: Enable tailstock Bit 3: Enable spindle control of main spindle via surface Bit 4: Enable spindle control of counterspindle via surface Bit 5: Enable spindle control of tool spindle via surface Bit 6: Enable balance cutting for dual-channel stock removal Bit 7: Retraction during stock removal along contour with G1 Bit 8: Input spindle chuck data in the program Bit 9: Additional entry of tailstock data in the program Bit 10: Enable convex thread Bit 11: Reserved Bit 12: Deactivate thread synchronization Bit 13: Stock rem. contour with CYCLE95 (828D programGUIDE without Advanced Technology) Bit 14: Turning on milling machines: cont. transition between pole position and beta plus
52224	$MCS_FUNCTION_MASK_TECHNOLOGY_0	0H		Technology function mask 0 Bit 0: Hide display of tools/tool functions Bit 1: Hide display of spindles/spindle functions Bit 2: Hide “Synchronous actions” softkey Bit 3: Hide “Retract” softkey Bit 4: Hide “Contour” softkey Bit 5: Hide “Diverse” softkey Bit 6: Hide “Simulation” softkey Bit 7: Hide “Simultaneous recording” softkey Bit 8: Hide “Mold making view” softkey
52229	$MCS_ENABLE_QUICK_M_CODES	0H		Enable fast M functions Bit 0: Coolant OFF Bit 1: Coolant 1 ON Bit 2: Coolant 2 ON Bit 3: Coolant 1 and 2 ON
52230	$MCS_M_CODE_ALL_COOLANTS_OFF	9		M code for all coolants OFF
52231	$MCS_M_CODE_COOLANT_1_ON	8		M code for coolant 1 ON
52232	$MCS_M_CODE_COOLANT_2_ON	7		M code for coolant 2 ON
52233	$MCS_M_CODE_COOLANT_1_AND_2_ON	-1		M code for coolant 1 + 2 ON
52250	$MCS_M_CODE_CHUCK_OPEN	[0]=M131		[0]: M code for Open chuck with non-rotating main spindle [1]: M code for Open chuck with non-rotating counter spindle
52251	$MCS_M_CODE_CHUCK_OPEN_ROT	[0]=M132		[0]: M code for Open chuck with rotating main spindle [1]: M code for Open chuck with rotating counter spindle
52252	$MCS_M_CODE_CHUCK_CLOSE	[0]=M130		[0]: M code for Close chuck main spindle [1]: M code for Close chuck counter spindle

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
52270	$MCS_TM_FUNCTION_MASK	0H		Function mask Tool management Bit 0: Do not permit Create tool on magazine location Bit 1: Load/unload disable when machine is not in reset Bit 2: Load/unload disable for emergency stop Bit 3: Load/unload or relocate tool to/from buffer disabled Bit 4: Load directly in spindle Bit 5: Use grinding configuration file Bit 6: Permit relocating to/from buffer despite disable (Bit3) Bit 7: Create tool using the T number Bit 8: Hide Relocate tool Bit 9: Hide Position magazine Bit 10: Reactivate tool using Position magazine Bit 11: Reactivate tool in all monitoring modes Bit 12: Fade out Reactivate tool
53220	$MAS_AXIS_MCS_POSITION	0		Position of axis in the Machine
53230	$MAS_SIM_START_POSITION	X=400 Z=0		Axis position at start of simulation Only adjust with geometry axes
53240	$MAS_SPINDLE_PARAMETER	MSP [0]=40 [1]=30		[0]: Spindle chuck: Chuck dimension ZC1 [1]: Spindle chuck: Stop dimension ZS1 [2]: Spindle chuck: Jaw dimension ZE2 (counter spindle only) in the data set of the main spindle and the counter spindle
53241	$MAS_SPINDLE_CHUCK_TYPE	0		Spindle jaw type only for counter spindle = 0: Jaw type 1 = 1: Jaw type 2
53242	$MAS_TAILSTOCK_PARAMETER	0		[0] Tailstock diameter [1] Tailstock length
54215	$SNS_TM_FUNCTION_MASK_SET	Bit 0 = 1		Function mask Tool management Bit 0: Diameter display for rotating tools Bit 1: Default direction of rotation for all turning tools is M4 Bit 2: Create tool without name suggestion Bit 3: Input disable for tool name and tool type in loaded tools Bit 4: Input disable for loaded tools, if the channel is not in Reset Bit 5: Calculate tool wear inputs additively Bit 6: Numerical input of the tool ident. Bit 7: Fade out tool monitoring parameters Bit 8: Diameter display for transverse axis - Geometry Bit 9: Diameter display for transverse axis - Wear Bit 10: Enable loading/relocating of tools on buffer locations Bit 11: Creating of new tools in gripper locations is disabled Bit 12: Do not unload measuring tools if "Unload all" function is executed Bit 13: Do not delete tool wear when inputting the geometry value Bit 14: Load/relocate tool. Search without pre-assignment of the magazine
55200	$SCS_MAX_INPUT_FEED_PER_REV	1		Feedrate input upper limit for mm/rev
55201	$SCS_MAX_INPUT_FEED_PER_TIME	10000		Feedrate input upper limit for mm/min
55202	$SCS_MAX_INPUT_FEED_PER_TOOTH	1		Feedrate input upper limit for mm/tooth

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
55212	$SCS_FUNCTION_MASK_TECH_SET	Bit 1 = 1 Bit 2 = 1		Function mask Cross-technology Bit 0: Tool preselection active Bit 1: Calculate thread depth from metric thread lead Bit 2: Refer to the table for thread diameter and depth Bit 3: Delete programs generated by contour cycles
55214	$SCS_FUNCTION_MASK_MILL_SET	Bit 0 = 1 Bit 2 = 1		Function mask Milling Bit 0: Default setting - milling cycles with synchronous operation Bit 2: Depth calculation in milling cycles without parameter SC = 0: Depth calculation in milling cycles is carried out between reference plane + safety clearance and depth = 1: Depth calculation in milling cycles is carried out without inclusion of the safety clearance Note: Bit 2 is active with the following milling cycles: CYCLE61, CYCLE71, CYCLE76, CYCLE77, CYCLE79, CYCLE899, LONGHOLE, SLOT1, SLOT2, POCKET3, POCKET4 Bit 3: Deactivation of alarm 61012 different scaling in the planet
55216	$SCS_FUNCTION_MASK_DRILL_SET	Bit 3 = 1 Bit 4 = 1		Function mask Drilling Bit 0: Reverse direction of spindle rotation during tapping Bit 1: Consider rotation of the tool plane (CYCLE86) Bit 2: Consider swiveled table kinematics (CYCLE86) Bit 3: Monitoring spindle MD 31050/31060 (CYCLE84) Bit 4: Monitoring spindle MD 31050/31060 (CYCLE840) Bit 5: reserved Bit 6: Adapt spindle position to the tool orientation (CYCLE86) Bit 7: Adapt direction of retraction path in the plane to active mirroring (CYCLE86)
55218	$SCS_FUNCTION_MASK_TURN_SET	Bit 0 = 1		Function mask Turning Bit 0: New thread table during thread cutting Bit 1: Reserved (CYCLE93) Bit 2: Reserved (CYCLE93) Bit 3: Chamfer in CYCLE930 as chamfer length (CHF) Bit 4: Retraction from the inside machining (CYCLE951) Bit 5: Disable new smoothing behaviour in thread cutting cycles
55300	$SCS_EASY_SAFETY_CLEARANCE	1		Simple input: Safety clearance
55301	$SCS_EASY_DWELL_TIME	0.6		Simple input: dwell time
55305	$SCS_EASY_DRILL_DEEP_FD1	90		Simple input: percentage 1st federate deep - hole drilling
55306	$SCS_EASY_DRILL_DEEP_DF	90		Simple input: percentage infeed deep - hole drilling
55307	$SCS_EASY_DRILL_DEEP_V1	1.2		Simple input: min. depth infeed deep - hole drilling
55308	$SCS_EASY_DRILL_DEEP_V2	1.4		Simple input: absolute retraction value deep - hole drilling
55309	$SCS_EASY_THREAD_RETURN_DIST	2		Simple input: return distance thread tapping
55400	$SCS_MILL_ENGRAVE_POINT_RAD	0		Engraving cycle CYCLE60: Circ. path radius for generating the char. ”point”

Notes:

Basic settings Technology data

MD	Identifier	Settings	Comment	Description
55500	$SCS_TURN_FIN_FEED_PERCENT	100		Only for ShopTurn cycles with complete machining rough-ing and finishing - Grooving cycles : all - Undercut cycles : “Thread DIN” and “Thread” - Thread cycles : all
55505	$SCS_TURN_ROUGH_O_RELEASE_ DIST	1		Return distance stock removal for external machining
55506	$SCS_TURN_ROUGH_I_RELEASE_ DIST	0,5		Return distance stock removal for internal machining
55510	$SCS_TURN_GROOVE_DWELL_TIM E	-1		negative value in spindle revolutions positive value in seconds
55540	$SCS_TURN_PART_OFF_CTRL_DIS T	0,1		Path for cut-off check
55541	$SCS_TURN_PART_OFF_CTRL_FE ED	0		Feedrate for cut-off check
55542	$SCS_TURN_PART_OFF_CTRL_FO RCE	10		Force in percent for cut-off check, in %
55580	$SCS_TURN_CONT_RELEASE_ANG LE	45		Contour turning: retraction angle
55581	$SCS_TURN_CONT_RELEASE_DIS T	1		Contour turning: retraction value
55582	$SCS_TURN_CONT_TRACE_ANGLE	5		Contour turning: minimum angle for rounding along contour
55583	$SCS_TURN_CONT_VARIABLE_DE PTH	20		Contour turning: percentage for variable cutting depth
55584	$SCS_TURN_CONT_BLANK_OFFSE T	1		Contour turning: blank allowance
55585	$SCS_TURN_CONT_INTERRUPT_TI ME	-1		Contour turning: feed interrupt time negative value in spindle revolutions positive value in seconds
55586	$SCS_TURN_CONT_INTER_RETRA CTION	1		Contour turning: retraction path after feed interrupt
55587	$SCS_TURN_CONT_MIN_REST_MA T_AX1	50		Contour turning: minimum difference dimension residual machining axis 1
55588	$SCS_TURN_CONT_MIN_REST_MA T_AX2	50		Contour turning: minimum difference dimension residual machining axis 2
55595	$SCS_TURN_CONT_TOOL_BEND_R ETR	0,1		Contour plunge turning: retraction path due to tool bending
55596	$SCS_TURN_CONT_TURN_RETRA CTION	0,1		Contour plunge turning: retraction depth prior to turning

Notes:

Setup Turning

Pre-requisite:

Sinumerik Operate is already installed, PLC commissioning has been carried out, the tool management set-up has been completed.

Licencing:

In order for the required function to be achieved on your machine, the following software options must be available:

- "ShopMill/ShopTurn"
  Order No. 6FC5800-0BP17-0YB0
- "Residual material detection and machining"
  Order No. 6FC5800-0BP13-0YB0
- "Simultaneous recording (real-time simulation)"
  Order No. 6FC5800-0BP22-0YB0
- "3D simulation 1 (finished part)"
  Order No. 6FC5800-0BP25-0YB0
- "TRANSMIT and peripheral surface transformation"
  Order No. 6FC5800-0BM27-0YB0

Configuration of the axes and spindles

Machine axis name

MD10000 $MN_AXCONF_MACHAX_ NAME_TAB	Machine axis name
	Change to:

MD10000[0] = MX1
MD10000[1] = MZ1
MD10000[2] = MCS1
MD10000[3] = MCS3
MD10000[4] = MQ1
MD10000[5] = MY1
MD10000[6] = MZS2
MD10000[7] = MCS2

Assignment of geometry axis to channel axis

MD20050 $MC_AXCONF_GEOAX_ ASSIGN_TAB	Assignment of geometry axis to channel axis
	Change to:

MD20050[0] = 1
MD20050[1] = 0
MD20050[2] = 2

Geometry axis name in channel

MD20060 $MC_AXCONF_GEOAX_ NAME_TAB	Geometry axis name in channel
	Change to:

MD20060[0] = X
MD20060[1] = Y
MD20060[2] = Z

Machine axis number valid in channel

MD20070 $MC_AXCONF_MACHAX_ USED	Machine axis number valid in channel
	Change to:

MD20070[0] = 1
MD20070[1] = 2
MD20070[2] = 3
MD20070[3] = 4
MD20070[4] = 5
MD20070[5] = 0
MD20070[6] = 0
MD20070[7] = 0

Channel axis name in channel

MD20080 $MC_AXCONF_CHANAX_ NAME_TAB	Channel axis name in channel
	Change to:

MD20080[0] = X1
MD20080[1] = Z1
MD20080[2] = C
MD20080[3] = CS3
MD20080[4] = Q
MD20080[5] = Y1
MD20080[6] = ZS2
MD20080[7] = CS2

Notes:

Rotary axis / spindle

MD30300 $MA_IS_ROT_AX	Rotary axis / spindle
	Change to:

AX3: MCS1
MD30300 = 1 axis is a rotary axis

AX4: MCS3
MD30300 = 1 axis is a rotary axis

AX5: MQ1
MD30300 = 1 axis is a rotary axis

Modulo conversion for rotary axis / spindle

MD30310 $MA_ROT_IS_MODULO	Modulo conversion for rotary axis / spindle
	Change to:

AX3: MCS1
MD30310 = 1 Modulo-transformation for rotary axis

AX4: MCS3
MD30310 = 1 Modulo-transformation for rotary axis

AX5: MQ1
MD30310 = 1 Modulo-transformation for rotary axis

Modulo 360 degrees displayed for rotary axis or spindle

MD30320 $MA_DISPLAY_IS_MODULO	Modulo 360 degrees displayed for rotary axis or spindle
	Change to:

AX3: MCS1
MD30320 = 1 Actual value display Modulo

AX4: MCS3
MD30320 = 1 Actual value display Modulo

AX5: MQ1
MD30320 = 1 Actual value display Modulo

Assignment of spindle to machine axis

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX	Assignment of spindle to machine axis
	Change to:

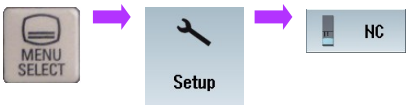
Description:

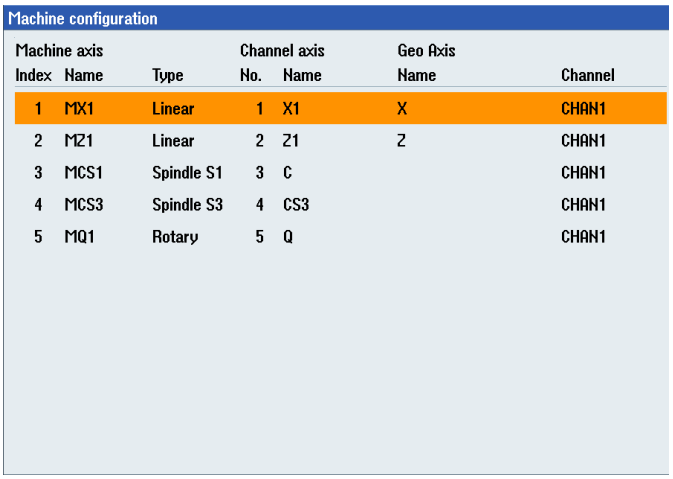
Definition of the spindle. The spindle is defined if the spindle number is entered into this MD.

AX3: MCS1
MD35000 = 1 Assignment of spindle to the machine axis (spindle S1)

AX4: MCS3
MD35000 = 3 Assignment of spindle to the machine axis (spindle S3)

Check of the set axes and spindles:





Notes:

Manufacturer Cycles

For the individual matching the following cycles are included in the standard cycle package:

CUST_TECHCYC.SPF	For matching to the technological cycles.
PROG_EVENT.SPF	Standard cycle for the support of the functionality of: - • Block search when milling is activated • Turning and indexing plane • Tool indexing

Note:

The cycle PROG_EVENT.SPF is a standard Siemens cycle and should not be changed by the manufacturer. For the manufacturer cycles CYCPE1MA.SPF or CYCPE_MA.SPF, the corresponding jump-out markers are prepared at the beginning and at the end of PROG_EVENT.SPF.

At the beginning of the cycle PROG_EVENT

- CYCPE1US in the directory CUS
- CYCPE1MA in the directory CMA

At the end of the cycle PROG_EVENT

- CYCPE_US.SPF in the directory CUS
- CYCPE_MA.SPF in the directory CMA

For these cycles templates are not available. If the CYCPE1MA.SPF and CYCPE_MA.SPF manufacturer cycles are loaded in the NC, the program branches from PROG_EVENT to the manufacturer cycles as appropriate.

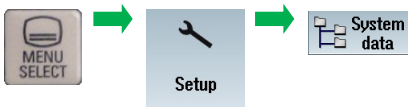
Manufacturer cycle CUST_TECHCYC.SPF

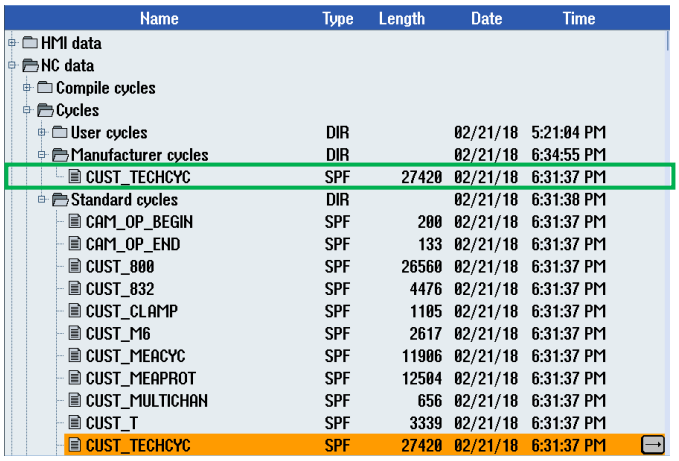
The manufacturer‘s cycle CUST_TECHCYC.SPF is called up by the

- ShopTurn cycles
- Standard cycle CYCLE92 (roughing)

Jump markers (_M1 to _M900), some of them already prepared with entries, have been provided in the cycle CUST_TECHCYC.SPF. These entries must be made to suit the machine by the manufacturer. A listing of the jump markers together with a brief description will be found on the next page. Further details can be found in the sections "JOG-TSM", "Block search" and "Clamping jaws, chucks, part gripper and tailstock".

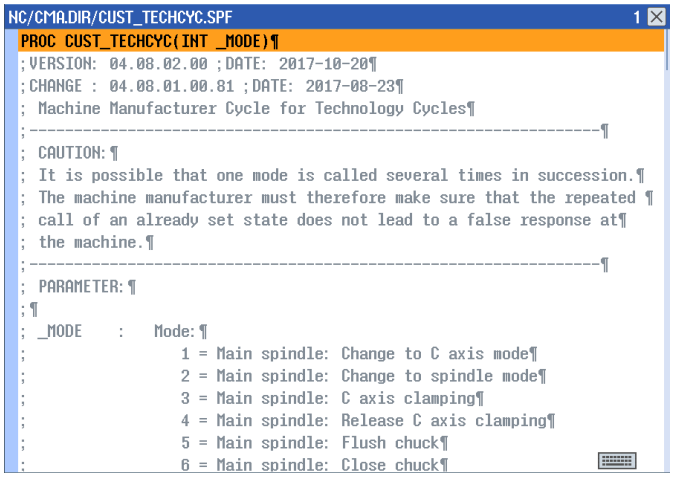
1. Copy the cycle CUST_TECHCYC.SPF from the directory / NC-Data / Cycles / Standard cycles.
2. Paste in the Cycle CUST_TECHCYC.SPF in to following directory: /NC-Data/Cycles/Manufacturer cycles





Notes:

Extract from the CUST_TECHCYC.SPF



```
NC/CMA.DIR/CUST_TECHCYC.SPF                                    1
PROC CUST_TECHCYC(INT _MODE)¶
;VERSION: 04.08.02.00 ;DATE: 2017-10-20¶
;CHANGE : 04.08.01.00.81 ;DATE: 2017-08-23¶
; Machine Manufacturer Cycle for Technology Cycles¶
;-----------------------------------------------------------------¶
; CAUTION:¶
; It is possible that one mode is called several times in succession.¶
; The machine manufacturer must therefore make sure that the repeated ¶
; call of an already set state does not lead to a false response at¶
; the machine.¶
;-----------------------------------------------------------------¶
; PARAMETER:¶
;¶
; _MODE    :    Mode:¶
;                  1 = Main spindle: Change to C axis mode¶
;                  2 = Main spindle: Change to spindle mode¶
;                  3 = Main spindle: C axis clamping¶
;                  4 = Main spindle: Release C axis clamping¶
;                  5 = Main spindle: Flush chuck¶
;                  6 = Main spindle: Close chuck¶
```

Implementation of the matching

In the manufacturer‘s cycle CUST_TECHCYC.SPF the following jump-to markers have been prepared.

- Switching between spindle and C-axis mode of the main spindle: _M1, _M2
  Tool spindle: _M11, _M12
  Counter spindle: _M21, _M22
- Spindle clamping and unclamping:
  Main spindle: _M3, _M4
  Tool spindle: _M13, _M14
  Counter spindle: _M23, _M24
- Chuck flushing, opening and closing:
  Main spindle: _M5, _M6, _M7, _M8
  Counter spindle: _M25, _M26, _M27, M28, _M29
- Axis of the counter spindle after block search:
  _M30
- Coupling and un-coupling of the driven tool (engaging / disengaging the drive):
  _M41, _M42

- Configuration of special functions when switching between the machining planes (no settings are necessary here for TRACYL or TRANSMIT with the C–axis.)
  _M61 to _M68
- Positioning the part receptacle for cut-off operations, moving it in and out.
  _M100, _M101, _M102
- Configuration of special functions for tool changing (these special functions will be called up after the T-command has been put out.)
  _M110, _M111, _M112, _M113, _M114
- Settings for coupling of the main and counter spindle
  _M120
- Setting of specific aspects for starting and ending the program
  _M131, _M135, _M136
- Setting of specific aspects for the tool change after block search:
  _M140, _M141, _M142

Setting-up for turning

Matching of the Coordinate systems to the user interface

Coordinate system position

MD52000 $MCS_DISP_COORDINATE_SYSTEM	Coordinate system position
	Change to: 34

Description:

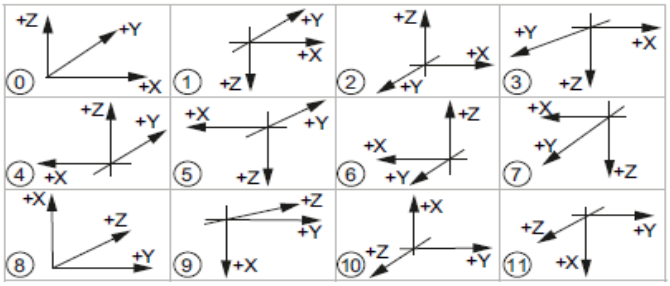
With this MD you adapt the operator panel of the coordinate system
to the machine's coordinate system. Depending on the selected
position, all help screens, the simulation graphics, the simulation and the input fields with the circular direction specified will change automatically.

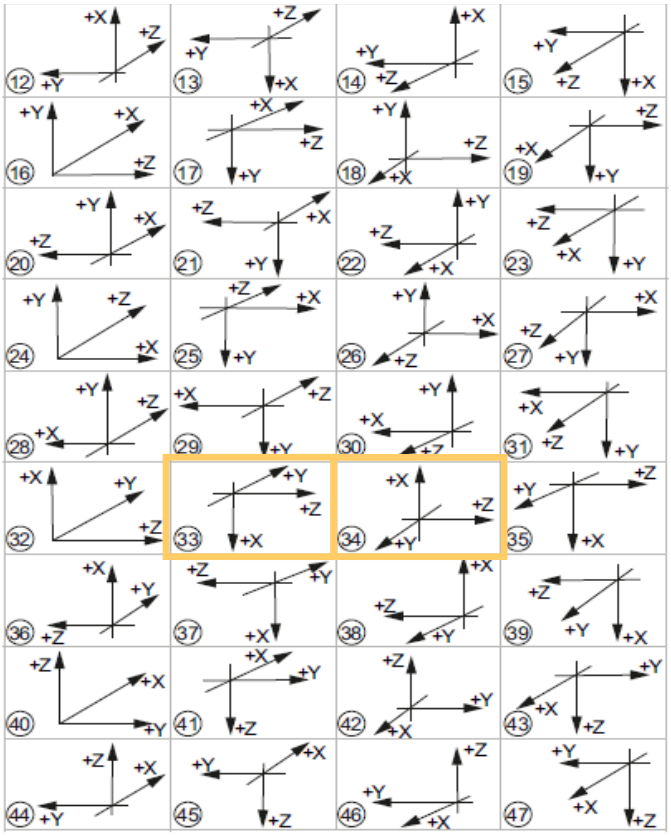
Notes:

The Coordinate system can take one from the shown positions.

Typical settings are:
34: Horizontal lathe with machining behind the turning centre (sloping bed turning machine)
33: Horizontal lathe with machining in front of the turning centre (flat bed turning machine)
0 = Vertical lathe (round-about turning machine) right hand head
6 = Vertical lathe (round-about turning machine) left hand head

Selection table of the possible Coordinate systems





Plane selection Milling and Turning

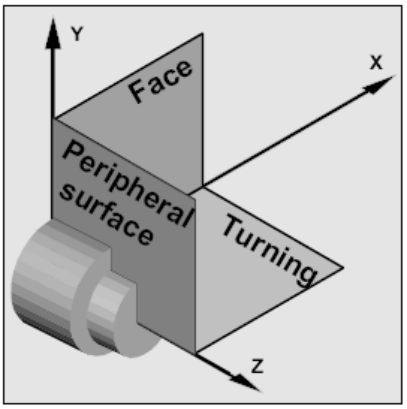
MD52005 $MCS_DISP_PLANE_MILL	Plane selection Milling
	Change to: 0

MD52006 $MCS_DISP_PLANE_TURN	Plane selection Turning
	Change to: 18

Description:

Turning operation: The machining plane is set permanently to G18.
Boring and milling operations: In the standard cycles the machining plane can be set by means of the parameter PL. In the ShopTurn cycles the setting of the MD52005 has no effect.

The machining planes are specified as shown:



Plane	Machining
X/Y	Boring/milling front G17
Z/X	Turning G18
Y/Z	Boring/milling mantle G19

Notes:

Technology

MD52200 $MCS_TECHNOLOGY	Technology
	Change to: 1

Description:

0: No specific configuration
1: Turning
2: Milling
Please also note MD 52201
$MCS_TECHNOLOGY_EXTENSION.

The selection of 1 specifies turning as the main technology. This means, the Softkeys for turning will be shown along the lower edge of the screen.



The Softkey boring is also shown if there are no driven tools. In this case, however, only concentric drilling with a fixed drill is possible.

Extended technology

MD52201 $MCS_TECHNOLOGY_EXTENSION	Extended technology
	Change to: 2

If driven tools are available, it is at this stage possible to also activate the milling technology in addition to turning..

Description:

0: no specific configuration
1: Turning
2: Milling by means of driven tools with TRANSMIT/TRACYL

Example:
Turning machine with milling technology
MD 52200 $MCS_TECHNOLOGY = 1
MD 52201 $MCS_TECHNOLOGY_EXTENSION = 2

The Softkey "Milling" will be shown as well.



Meaning of the axes in the channel

MD52206[x] $MCS_AXIS_USAGE	Meaning of the axes in the channel
	Change to: [2] = 3 [3] = 1

MD52206[2] = 3 Main spindle C
MD52206[3] = 1 Tool spindle CS3

Description:

0 = no specific meaning
1 = Tool spindle (driven tool)
2 = C-axis of the Tool spindle (driven tool)
3 = Main spindle (turning)
4 = C-axis of the main spindle (turning)
5 = Counter spindle (turning)
6 = C-axis of the counter spindle (turning)
7 = Linear axis of the counter spindle (turning)
8 = Tailstock (turning)
9 = Steady (turning)
10 = B-axis (Turning)
12 = B axis of the counter spindle (turning)
13 = Traverse travel X of the counter spindle (turning)

Axis attributes

MD52207[x] $MCS_AXIS_USAGE_ATTRIB	Axis attributes
	Change to: Bit 3 = 0 Bit 4 = 0 Bit 5 = 0

MD52207[2] C

Bit 3 = 0: C+ Direction of rotation clockwise when looking towards the spindle head.

Bit 4 = 0: M3 is clockwise when looking towards the spindle head

Bit 5 = 0: M3 means C+ direction

Note:

If there is a separate C-axis drive, the entry for Bit 3 must be made in the data set of that drive.

Notes:

Description:

Bit 3: Positive direction of rotation is counter-clockwise (for rotary axes)

Bit 4: Displayed direction of rotation for M3 is counter-clockwise (for spindles)

Bit 5: Direction of rotation M3 corresponds to rotary axis minus (for spindles).
This Bit must be set in the same way as PLC interface signal “Invert M3/M4”.
For ONE / 840D sl: PLC bit DBnn.DBX17.6, nn = 31 + machine axis index
For 828D: PLC bit DB38xx.DBX2001.6, xx is machine axis index

2 possible configurations:

- spindle and C-axis with a drive
- Separate C-axis drive

Specifying the direction of rotation

In order for the direction of rotation of the spindle and the C-axis to be shown correctly in the ShopTurn user interface and to ensure that the correct direction of rotation is carried out when programming ShopTurn functions, some settings must be made that correspond with one another. In this case the settings must be chosen according to the actual direction of rotation of the spindle/C-axis of the machine.

In case of C-axis operation with the main spindle

The assignment of the direction of spindle rotation (M3/ M4) to the positive direction of rotation of the C-axis is obtained from the PLC interface signal “Invert M3/M4” of the respective C-axis.
Bit 4 determines, whether M3 and C+ rotate in the same direction (= 0) or in opposition (= 1). It must be ensured under all circumstances that Bit 5 is identical to PLC interface signal “Invert M3/M4”!

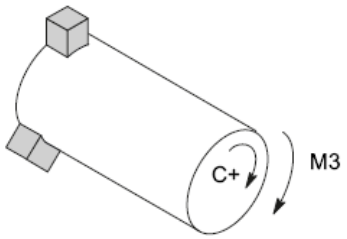
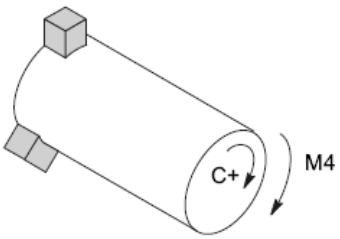
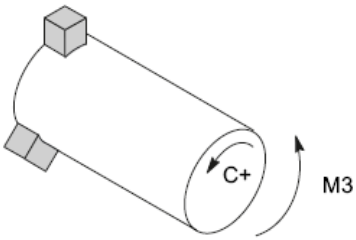
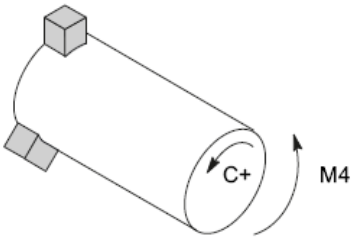
Our example machine:

The machine features no separate drives for the spindle and for the C-axis. This means that the settings in machine datum 52207 must be entered according to the axis index for the main spindle (Index [2]).

Notes:

From this derive the following setting possibilities for the main spindle. The settings for the machine data, however, depend on the direction of viewing the coordinate axis.

No matter as to which of the four combinations of MD52207[2] Bit 3 und Bit 4 shown below the machine manufacturer selects, it is imperative that the settings shown in one line for PLC interface signal “Invert M3/M4” and MD52207[2] Bit 5 are used.

Rotation dir. of the main spindle	52207[2] C Bit 3	52207[2] C Bit 4	PLC bit C “Invert M3/M4“	52207[2] C Bit 5
	**0**	**0**	**0**	**0**
	0	1	1	1
	1	1	0	0
	1	0	1	1

The setting of the machine datum MD52207 Bit 3 requires that viewing takes place in the direction of the negative Z axis. If, however, you prefer to view in the positive Z axis direction, the values must be reversed; i.e. "0" and "1" must be exchanged.

Note:

The MD52207[2] of the main spindle is relevant for

- the readout of the spindle direction in the T,F,S window
- the readout within simulation simultaneous recording
- the rotation direction of the spindle after tool change according to the direction given in the tool list

Notes:

Function mask display

MD52210 $MCS_FUNCTION_MASK_ DISP	Function mask display
Default: Bit 0 = 1 Bit 1 = 1	Change to:

Description:

Bit 0: Meas. system for programs always in the base system
Bit 1: Face view when turning in the school coordinate system
Bit 2: Hide "T,S,M" softkey in the JOG area
Bit 3: Generate automatic end-of-program in MDI (with Softkey "Delete blocks")
Bit 4: Show follow-on-tool in T,F,S-window

Bit 0 = 0:
In a ShopTurn program it is possible to select "mm dimensions" or "inch dimensions" in the program header.

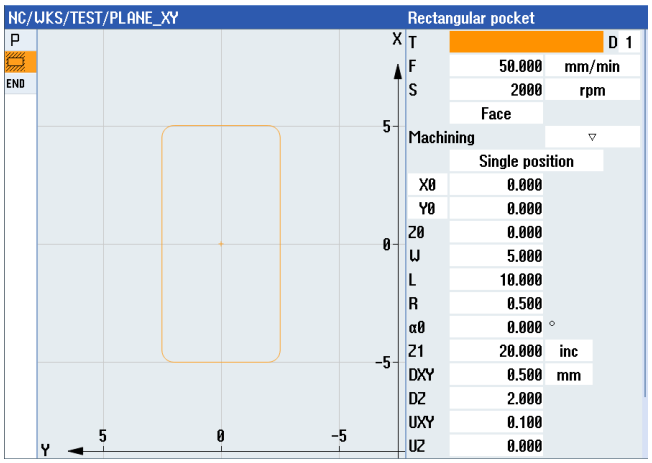
Bit 0 = 1:
The selection field “Unit of meas.” is hidden. The measuring system for programs is always in the base system.





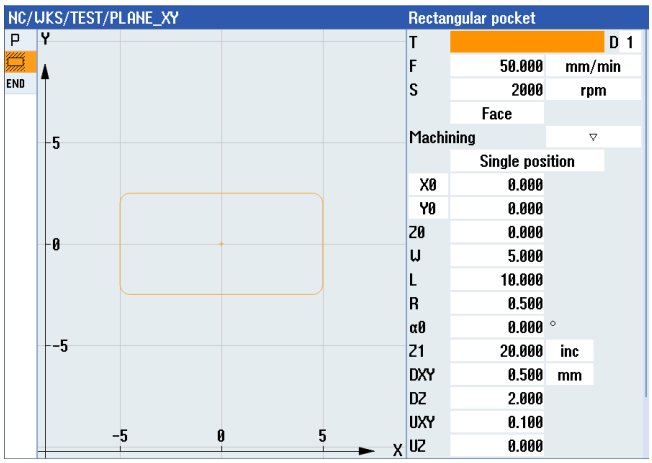


Bit 1=0: Display of the coordinate system when machining the front face (G17)



The display of the axis directions depends on the setting for the position of the coordinate system in MD 52000 $MCS_DISP_COORDINATE_SYSTEM; in this example = 34 for a horizontal lathe with machining behind the turning centre.

Bit 1=1: Display of the coordinate system during face machining (G17) in the ordinary Coordinate system



The axis direction are always displayed in the ordinary coordinate system independently of the setting of the machine datum 52000 $MCS_DISP_COORDINATE_SYSTEM.

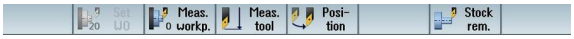
Notes:

Continuation MD52210

Bit 2=0: The Softkey "T,S,M" is shown in the mode JOG . This setting is in line with the basic setting of Shop-Turn



Bit 2=1: The Softkey "T,S,M" is hidden in the mode JOG.



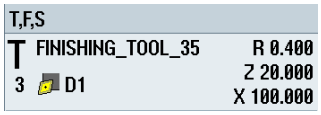
Bit 3=0: Under MDA the command End-of-program will not be generated automatically after "Block delete".



Bit 3=1: Under MDA the M-command entered in MD 10714 $M_NO_FCT_EOP will automatically be generated as End-of-program command after "block delete". In this example configuration it is M32.



Bit 4=0: The follow-up tool is not shown in the T,F,S-window



Bit 4=1: The follow-up tool is shown in the T,F,S-window. This setting makes sense only in conjunction with tool prepare on a machine featuring a double grasp-er. In the configuration chosen for this example the display of the follow-up tool is not possible.

Function mask Cross-technology

MD52212 $MCS_FUNCTION_MASK_TECH	Function mask Cross-technology
	Change to: Bit 3 = 1 Bit 5 = 0

Description:

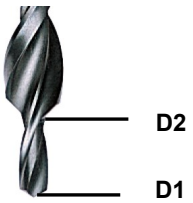
Bit 0: Enable swivel plane/align tool with tool carrier
Bit 1: Optimized travel along software limit switches (ShopMill/ShopTurn)
Bit 2: Special approach logic for step drill (ShopTurn)
Bit 3: Call block search cycle for ShopMill/ShopTurn
Bit 4: Approach logic through cycle (ShopTurn)
Bit 5: Call block search cycle for SERUPRO
Bit 6: Work offset ZV cannot be entered as an absolute value (ShopTurn)

Bit 0 and Bit 1 are meaningless for the machine configuration used here. These Bits are used in the technology Turning in conjunction with a B-axis kinematics.

Bit 2 = 0

The current cutting edge will be taken into account when retracting.

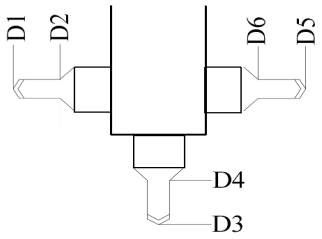
Bit 2 = 1 (Step drill)



The longest cutting edge in direction of the active cutting edge is taken into account when retracting.

Notes:

Continuation MD52212



Bit 5 = 0
Block search cycle for SERUPRO will not be called up.

The setting = 0 is provided for single-channel ShopTurn machines.

Bit 5 = 1
The standard cycle CYCLE207 will be called up after Block search with SERUPRO.

The setting of Bit 3 is not effective in this case. The F_S-ASUP will not be called up.

This setting must be used only for multi-channel machines.

Bit 3 = 0

The ShopTurn cycle F_S_ASUP is not carried out after block search.

Bit 3 = 1

The ShopTurn cycle F_S_ASUP will be carried out after block search. Details regarding this setting are contained in the section "Block search".

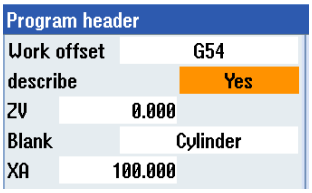
Bit 4 = 0 Fixed value (ShopTurn only)

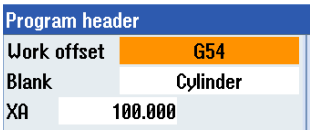
Bit 6 = 0

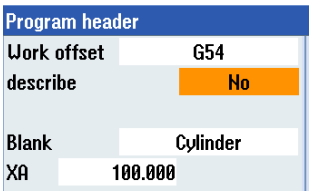
In the program header of a ShopTurn program the parameters "Description of zero point offsets" and "ZV" appear and the operator can choose whether the zero point offset is to be described.

Bit 6 = 1

In the program header of a ShopTurn program the parameters "Description of zero point offsets" and "ZV" will not appear.







Function mask Milling

MD52214 $MCS_FUNCTION_MASK_MILL	Function mask Milling
Default: 0H	Change to:

Description:

Bit 0: Enable cylinder surface transformation (ShopMill)

Bit 1: Offer fixed table for blank clamping (at milling machines)

Bit 2: Special handling of slide mill/saw for polyhedron

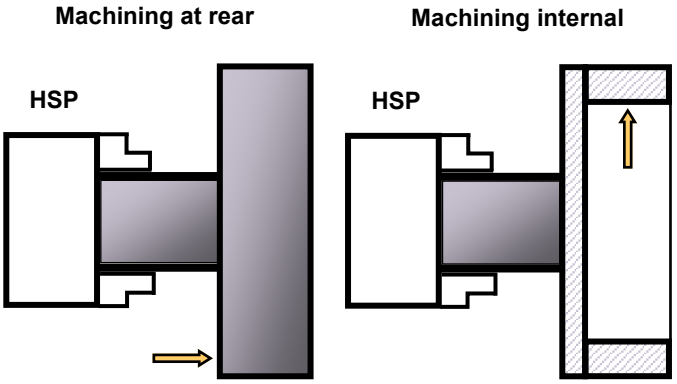
Bit 3: Enable inside/rear machining (ShopTurn)

Bit 4: Enable spindle clamping (C axis) (ShopTurn)

Notes:

Continuation MD52214

Bit 3 = 1 Release of machining internally/at the rear

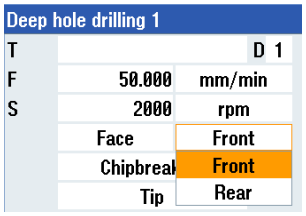


Usage

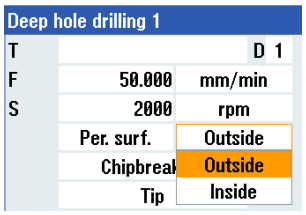
1. Machining at the rear refers to the face of the work piece opposite the front face. The work piece must be clamped in such a manner that machining on the rear using the required tool is possible on the rear of the work piece. The infeed along the Z-axis (e.g. drilling depth) is in the positive direction.
2. Internal machining is possible on large internal work piece diameters using a suitable tool. The infeed along the X-axis (e.g. drilling depth) takes place in the positive direction.

With the release of "machining internally/at rear" an additional input parameter appears in the boring, milling and position masks depending on the face to be machined.

Axial face: front / rear



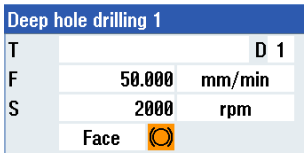
Mantle: external / internal



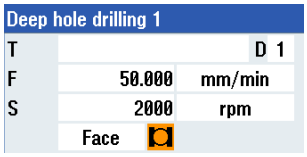
Bit 4 = 1: Enable spindle clamp (C-axis)

With the release of the spindle clamping an additional input parameter will appear in the boring and milling masks "clamping OFF / clamping ON" .

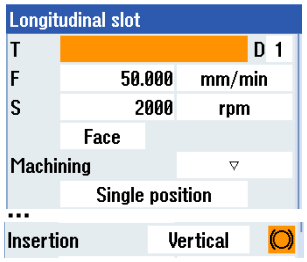
Boring: Clamping OFF



Boring: Clamping ON



Milling: Plunging with clamping OFF



Milling: Plunging with clamping ON



Notes:

Function mask Drilling

MD52216 $MCS_FUNCTION_MASK_ DRILL	Function mask Drilling
Default: 0H	Change to:

Description:

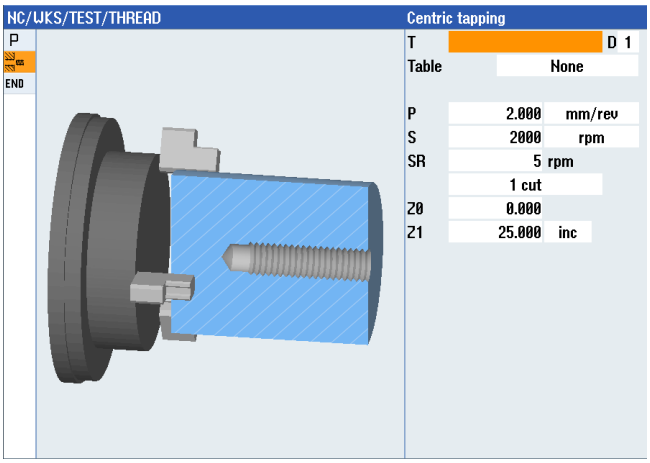
Bit 0: CYCLE84 Show technology fields (without compensating chuck)

Bit 1: CYCLE840 Show technology fields

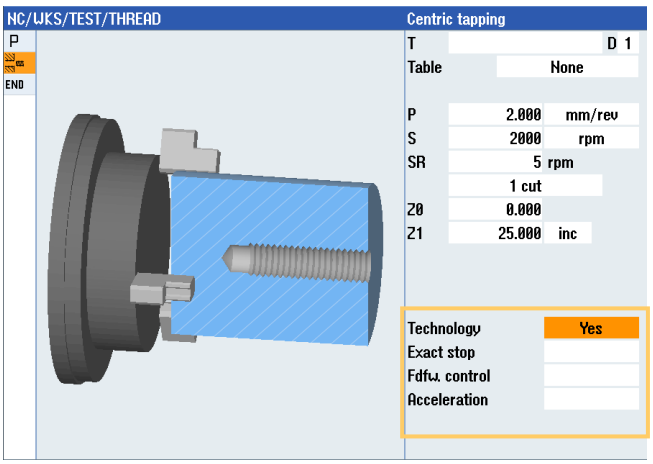
The selection "with compensating chuck (CYCLE840)" is not available for ShopTurn.

Note: The release of the technology requires an axis optimization!

MD52216 Bit 0 = 0 Technology fields will not be shown



MD52216 Bit 0 = 1 Technology fields are being shown



Function mask Turning

MD52218 $MCS_FUNCTION_MASK_ TURN	Function mask Turning
Default: 0H	Change to:

Description:

Bit 0: Enable zoom under manual for tool measurement

Bit 1: Enable parts gripper for cut-off

Bit 2: Enable Tailstock

Bit 3: Enable spindle control of main spindle via surface (only Manual machine)

Bit 4: Enable spindle control of tool spindle via surface (only Manual machine)

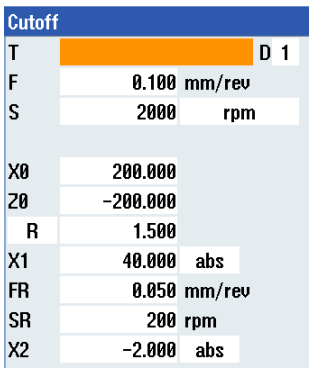
Bit 5: Enable spindle control of counter spindle via surface (only Manual machine)

These settings for bit 3 to 5 are reserved for the Manual machine function, MD52212 bit 8 = 1.

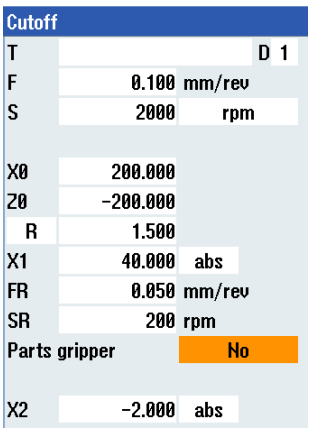
Notes:

Continuation MD52218

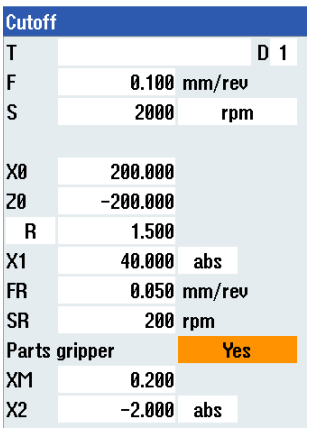
Bit 1 = 0
In the mask "Cut-off" the parameter "Parts gripper" is not shown".



Bit 1 = 1
In the mask "Cut-off" the parameter "Parts gripper" is being shown".



When "Yes" a further parameter "XM" (Depth parts gripper extend) appears.



When the parts gripper is released, there will be jumps from the machining cycles for the cut-off to the Manufacturer‘s cycle CUST_TECHCYC.SPF
- ShopTurn program: Mode 100, 101, 102 and 103
- ProgramGuide program. Mode 101 and 102

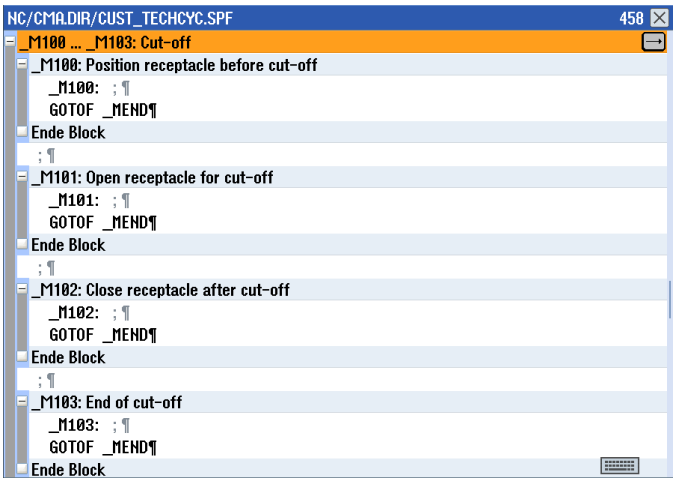
- Mode 100. Positioning the parts gripper before the cut-off
  This call-up occurs only with a ShopTurn-Program, if prior to the cut-off the tool has been positioned (the axis is still at the retraction level).

- Mode 101. Extending the parts gripper before the cut-off
  This jump occurs if the X-axis has attained the position defined under XM for the cut-off.

- Mode 102. Closing the parts gripper after the cut-off.
  This jump occurs when the X-axis during the cut-off has attained the end-position as defined under X2.

- Mode 103. End of cut-off.

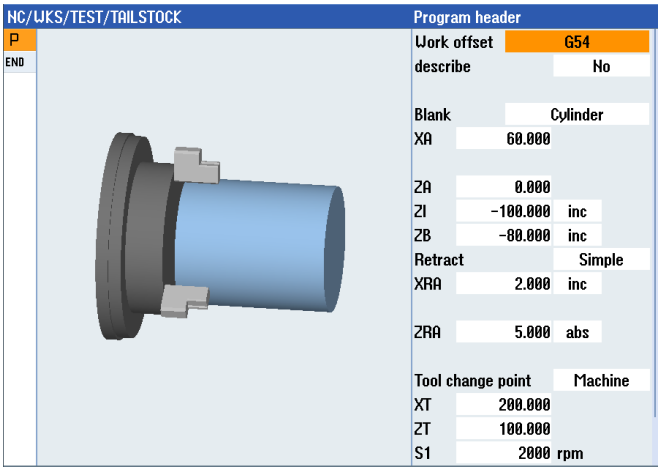
Extract from the Manufacturer‘s cycle CUST_TECHCYC.SPF with the modes 100 - 103, which have been provided for the parts gripper. At this point the machine manufacturer can undertake the machine-specific matching.



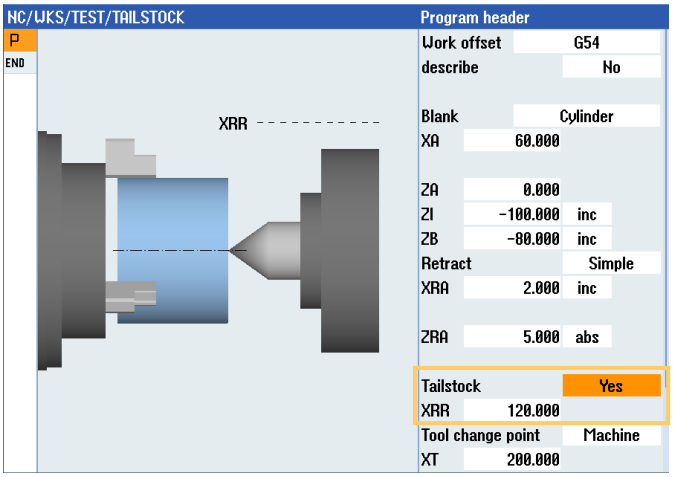
Notes:

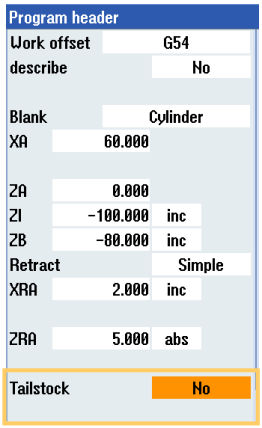
Continuation MD52218

Bit 2 = 0 Tailstock not released



Bit 2 = 1 Tailstock released





Description:

Bit 0: Tool pre-selection active
Bit 1: Calculate thread depth from metric thread lead
Bit 2: Refer to the table for thread diameter and depth

Bit 0: Tool pre-selection activated
Directly after the exchange has been completed, the tool will be prepared
The generation of the follow-up tool occurs only with JobShop cycles

Bit 1: Automatic evaluation of the thread depth for metric threads

Bit 2: Transfer thread diameter and thread depth from table.

Function mask Cross-technology

SD55212 $SCS_FUNCTION_MASK_TECH_SET	Function mask Cross-technology
Default Bit 1 = 1 Bit 2 = 1	Change to:

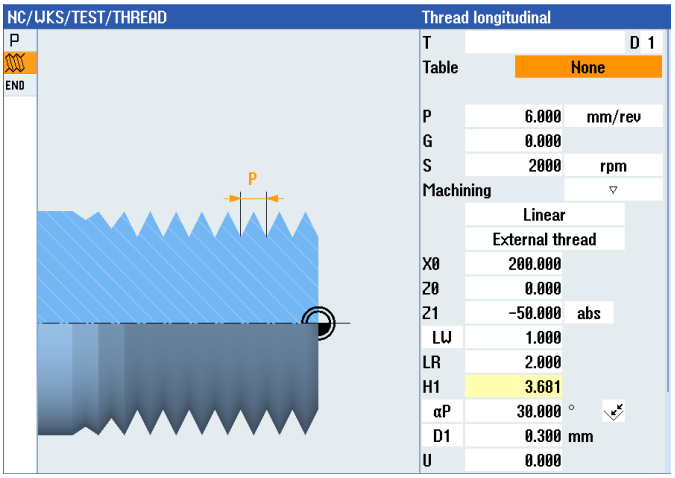
Notes:

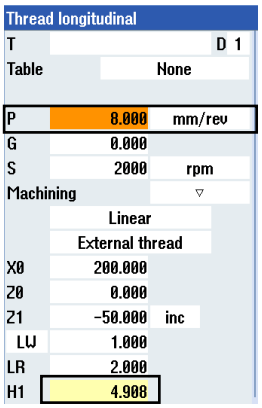
Continuation SD55212

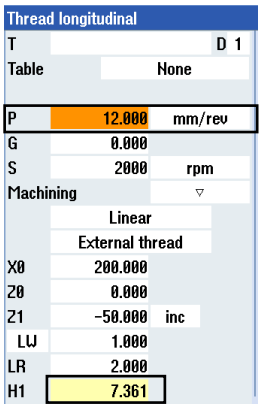
Bit 1 = 0 The thread depth will not be evaluated automatically

The value for the thread depth "H1" must be entered manually from a table.

Bit 1 = 1 The thread depth will be evaluated from the metric thread pitch.



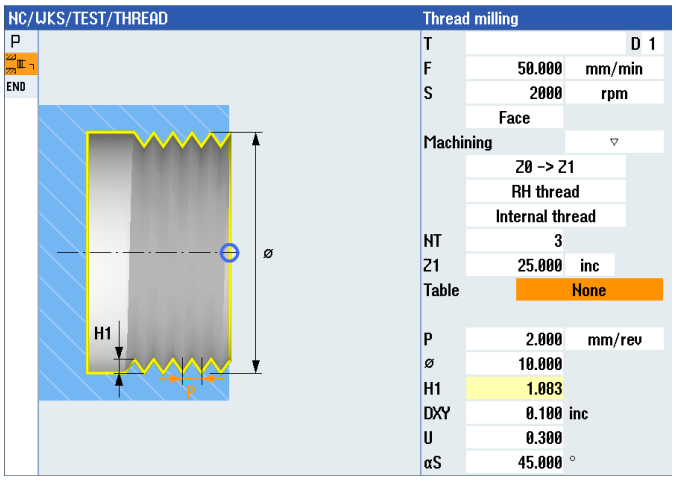


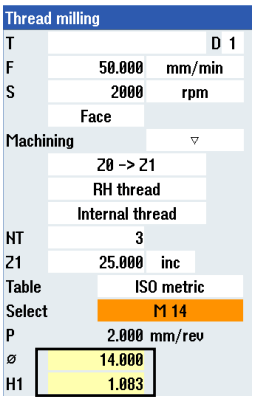


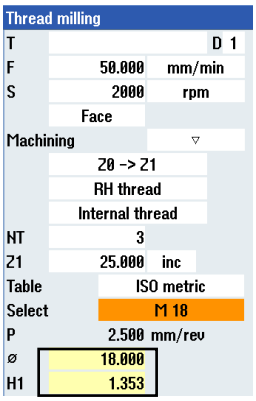
Only if in the table the column "without" is selected and for P the unit "mm/rev" will the thread depth "H1" in the yellow-backed fields be evaluated automatically. These automatically obtained values can be overwritten by the operator. If the thread pitch is entered again, the thread depth will be evaluated once more automatically.

Different factors are stored for the internal and external threads:
- Internal threads: 0,5413
- External threads: 0,6134

Bit 2 = 1 Thread diameter and depth from a table







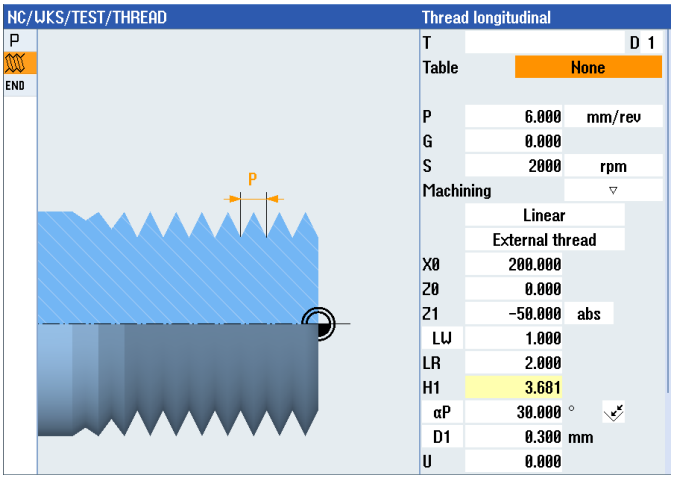
As soon as the line table "ISO metric" is selected, the table is activated for the various threads.

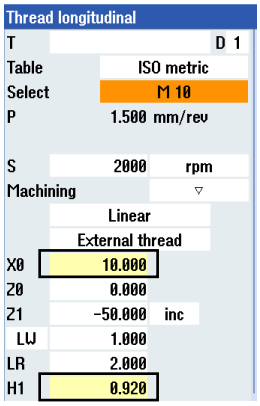
The yellow-backed fields for nominal diameters and thread depth H1 will be filled in with tabular values depending on the chosen thread.

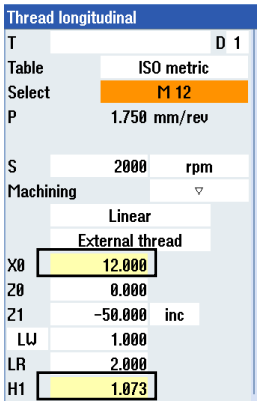
Notes:

Continuation SD55212

Bit 2 = 1 Thread diameter and depth from a table







As soon as the line table "ISO metric" is selected, the table is activated for the various threads.

The yellow-backed fields for nominal diameters and thread depth H1 will be filled in with tabular values depending on the chosen thread, the value H1 being the thread depth.

Function mask Milling

SD55214 $SCS_FUNCTION_MASK_ MILL_SET	Function mask Milling
Default: Bit 0 = 1 Bit 2 = 1	Change to:

Description:

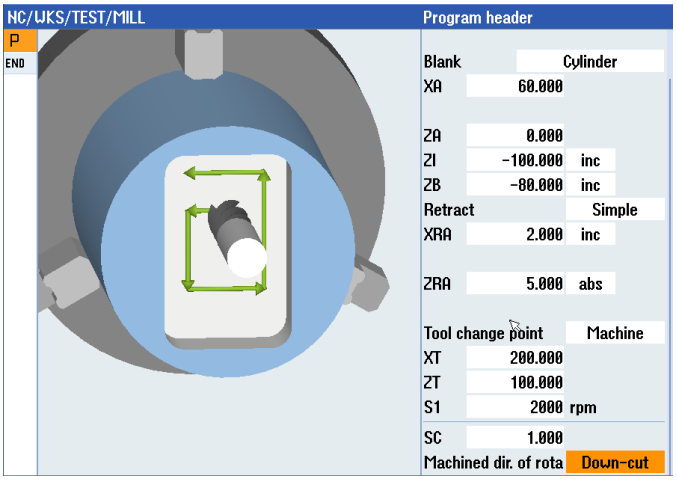
Bit 0: Default setting for milling with synchronous operation

Bit 1: Not assigned

Bit 2: Depth calculation in milling cycles without parameter SC

Bit 0 is used as setting for the initial commissioning. If thereafter the program header for milling is opened, the toggle key can be used to select between climb milling and conventional milling. The selection made here remains activated.

Bit 0 = 1 Default setting for milling down-cut



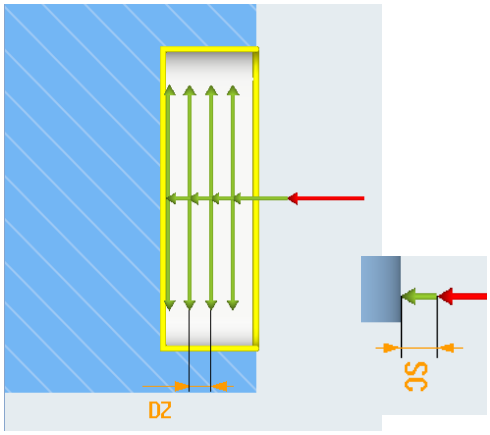


Notes:

Continuation SD55214

Bit 2 = 0
The subdivision of cuts in direction of infeed is obtained including the safety gap SC for mantle and face milling for the path from Z0+SC to Z1.

Bit 2 = 1
The subdivision of cuts in direction of infeed is obtained without the safety gap SC for mantle and face milling for the path from Z0 to Z1.



Note:
The safety gap is specified in the program header.

Function mask Drilling

SD55216 $SCS_FUNCTION_MASK_ DRILL_SET	Function mask Drilling
Default: Bit 3 = 1 Bit 4 = 1	Change to:

Description:

Bit 0: Reverse direction of spindle rotation during tapping

Bit 1: Boring CYCLE86: Taking into account the rotation of the work piece plane when positioning the spindle

Bit 2: Boring CYCLE86: Taking into account the indexed table kinematics (tool carrier) when positioning the spindle

Description:

Bit 3: Tapping CYCLE84: (without compensating chuck). Monitoring the machine data 31050[x] and 31060[x] of the spindle

Bit 4: Tapping CYCLE840: (with compensating chuck). Monitoring the machine data 31050[x] and 31060[x] of the spindle

Bit 5: Reserved

The denominator of the gearbox is contained in MD31050[x] while MD31060[x] contains the numerator. Together the two provide the gear ratio for the spindle. This is required for monitoring the feed obtained from the speed x the pitch.

Description:

Bit 0: Reverse direction of spindle rotation during tapping

Bit 1 = 1: Taking into account the rotation of the tool plane (CYCLE 86)
In the cycle CYCLE86 Boring the spindle is positioned on the parameter SPOS when reaching the boring depth. If SPOS=0 the tool cutting edge points in the positive direction of the 1st axis of the plane (G17 towards X+).

If the setting is varied by using Bit 1 = 1, a rotation about the tool axis (e.g. ROT Z180) will be taken into account for the spindle positioning.

Bit 2 = 1: Taking into account indexed table kinematics (CYCLE 86)
See description Bit 1
With the selection of Bit 2 = 1 an indexed kinematics will be taken into account when positioning the spindle.

Notes:

Continuation SD55216

Description:

Bit 3 = 1: Tapping CYCLE84: Monitoring the machine data 31050 $MA_DRIVE_AX_RATIO_DENOM (divider of gearbox ratio) and 31060 $MA_DRIVE_AX_RATIO_NUMERA (numerator of gearbox ratio) of the spindle for the boring axis.

The monitoring is activated only if MD35590 $MA_PARAMSET_CHANGE_ENABLE = 2
This setting permits a parameter block change by the PLC-interface or by the command SCPARA (servo parameter block programming).
If the monitoring is activated the alarm 61127 [Channel %1:] Block %2 (gear ratio of the tapping axis incorrectly defined) will be set:

Bit 4 = 1: Similar to Bit 3, however, for the cycle CYCLE840

Bit 5: Reserved

Function mask Turning

SD55218 $SCS_FUNCTION_MASK_TURN_SET	Function mask Turning
Default: Bit 0 = 1	Change to:

Description:

Bit 0: New thread table for thread cutting

Bit 1: Reserved (CYCLE93)

Bit 2: Reserved (CYCLE93)

Bit 0: For the types of thread ISO metric, Whitworth BSW, Whitworth BSP and UNC a table is included, which will be used when this Bit is activated.

Cycle settings

Roughing feedrate for complete machining in %

SD55500 $SCS_TURN_FIN_FEED_PERCENT	Roughing feedrate for complete machining in %
Default: 100%	Change to:

Description:

When complete machining has been selected (roughing and finishing) the percentage specified in this datum will be used for the finishing feedrate F .

The selection complete machining (roughing in finishing) is only available in technology turning:

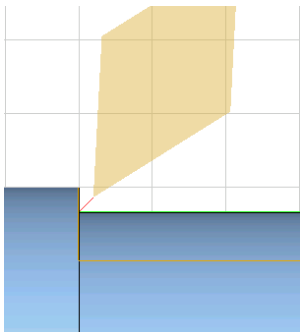
- Grooving cycles “Groove 1”, “Groove 2” and “Groove 3”
- Undercut cycles “Thread DIN” and “Thread”
- Thread cycles “longitudinal”, “taper”, “face” and “chain”

Return distance stock removal for external machining

SD55505 $SCS_TURN_ROUGH_O_RELEASE_DIST	Return distance stock removal for external machining
Default: 1mm	Change to:

Description:

This setting datum states the distance parallel to the axis by which the tool will be retracted during machining. If the safety distance specified in the program is less than this setting datum, then the safety distance will be applicable.



-1: Retraction will always take place by the safety distance SC specified in the program header.

This setting datum is effective only for the roughing cycles 1-3 and not during contouring.

Notes:

Return distance stock removal for internal machining

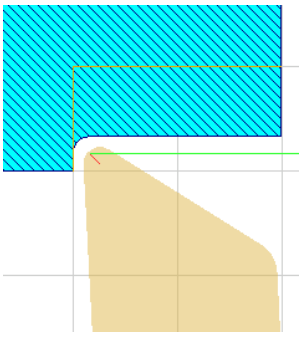
SD55506 $SCS_TURN_ROUGH_I_ RELEASE_DIST	Return distance stock removal for internal machining
Default: 0,5mm	Change to:

Description:

This setting datum states the distance parallel to the axis by which the tool will be retracted during machining. If the safety distance specified in the program is less than this setting datum, then the safety distance will be applicable.

-1: Retraction will always take place by the safety distance SC specified in the program header.

This setting datum is effective only for the roughing cycles 1-3 and not during contouring.



Tool clearance time for grooving at the base (neg. value=rotations)

SD55510 $SCS_TURN_GROOVE_ DWELL_TIME	Tool clearance time for grooving at the base (neg. value=rotations)
Default: -1s	Change to:

Description:

If in the course of a cycle, e.g. plunging and extended roughing (CYCLE952) a dwell period is included, then the value of this setting datum will be used.

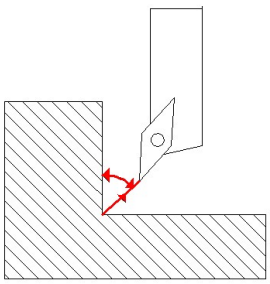
- negative Value: Dwell in spindle revolutions
- positive Value: Dwell in seconds

Contour turning: Retraction angle

SD55580 $SCS_TURN_CONT_ RELEASE_ANGLE	Contour turning: retraction angle
Default: 45Grad	Change to:

Description:

This setting datum specifies the angle at which the tool is lifted off the surface when roughing during contouring.



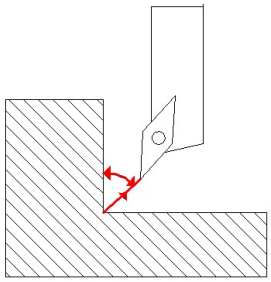
Notes:

Contour turning: retraction value

SD55581 $SCS_TURN_CONT_ RELEASE_DIST	Contour turning: retraction value
Default: 1mm	Change to:

Description:

This setting datum specifies the distance, by which the tool is lifted off the surface in both directions when contouring.

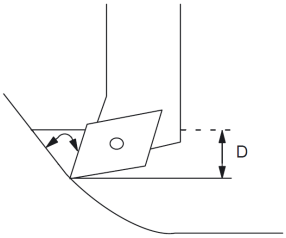


Contour turning: minimum angle for rounding along contour

SD55582 $SCS_TURN_CONT_TRACE_ ANGLE	Contour turning: minimum angle for rounding along contour
Default: 5Grad	Change to:

Description:

This setting datum specifies the angle between cutting edge and the contour as from which the cutting depth is varied such as to remove any remaining material.

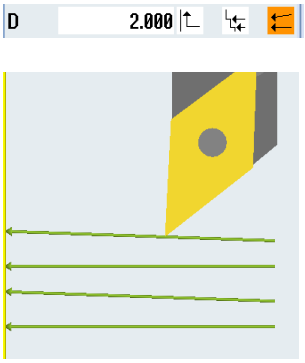


Contour turning: percentage for variable cutting depth

SD55583 $SCS_TURN_CONT_ VARIABLE_DEPTH	Contour turning: percentage for variable cutting depth
Default: 20%	Change to:

Description:

Percentage for variable cutting depth when contouring. The variable cutting depth can be selected for roughing and residual material removal.



Contour turning: Blank allowance

SD55584 $SCS_TURN_CONT_BLANK_ OFFSET	Contour turning: blank allowance
Default: 1mm	Change to:

Description:

This setting datum states the distance (machining allowance) of the raw material as from which switching from G0 to G1 takes place during contouring, in order to compensate for possible blank allowances.

Notes:

Contour turning: Feed interrupt time (neg. values = revolutions)

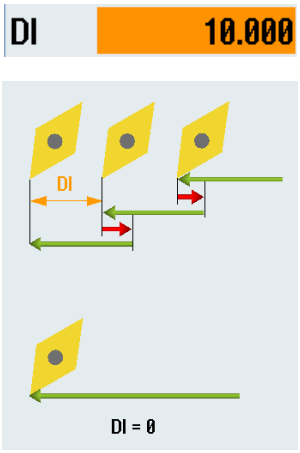
SD55585 $SCS_TURN_CONT_INTERRUPT_TIME	Contour turning: feed interrupt time (neg. values = revolutions)
Default: -1s	Change to:

Description:

Feed interruption times during contouring, contour grooving and plunging

- Negative Value: feed interruption in revolutions
- Positive Value: feed interruption in seconds

This setting datum is effective only if the setting datum 55586 $SCS_TURN_CONT_INTER_RETRACTION equals 0.



Contour turning: Retraction after feed interrupt

SD55586 $SCS_TURN_CONT_INTER_RETRACTION	Contour turning: Retraction path after feed interrupt
Default: 1mm	Change to:

Description:

Retraction path feed interrupt during contour turning, contour grooving and plunge turning:

R>0: retraction path after feed interrupt (the setting datum 55585 $SCS_TURN_CONT_INTERRUPT_TIME is now ineffective!)
=0: no retraction path

Contour turning: minimum difference dimension residual machining axis 1

SD55587 $SCS_TURN_CONT_MIN_REST_MAT_AX1	Contour turning: minimum difference dimension residual machining axis 1
Default: 50%	Change to:

Description:

This MD defines the limit value for stock removal of residual material in the direction of the 1st axis.

Example:

If the MD is set to 50% and the finishing allowance is 0,5 mm, the residual material, which is thinner than 0.25mm, will not be machined in a separate pass, instead it will be removed with the finishing pass.

Note:

If residue machining is programmed and the residual material is less than defined in the setting data 55587 and 55588, in other words there is no residual material, the alarm 61732 "no residual material" will appear..

Contour turning: minimum difference dimension residual machining axis 2

SD55588 $SCS_TURN_CONT_MIN_REST_MAT_AX2	Contour turning: minimum difference dimension residual machining axis 2
Default: 50%	Change to:

Description:

This MD specifies the limiting value for the removal of residual material in the direction of the 2nd axis.

Example:

If this MD is set to 50% and the finishing allowance is 0.5mm, any residual material thinner than 0.25mm will not be removed by means of an extra pass, instead it will be removed together with the finishing pass.

Note:

If residue machining is programmed and the residual material is less than specified in the setting data 55587 and 55588, in other words there is no residual material during roughing, the alarm 61732 "no material to be machined" will appear.

Notes:

Contour plunge turning: Retraction path due to tool bending

SD55595 $SCS_TURN_CONT_TOOL_BEND_RETR	Contour plunge turning: retraction path due to tool bending
Default: 0,1mm	Change to:

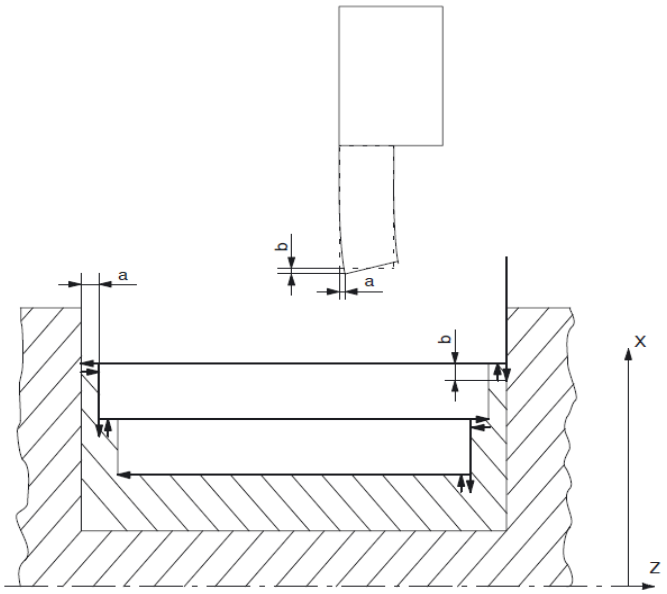
Description:

Due to the bending of the tool when grooving it is not possible to traverse right up to the contour during roughing. This SD specifies the lateral distance to the previous cut, by which the successive cut will be shortened each time.

The sketch right illustrates the settings for the setting data 55595 and 55596.

a: Bending of the tool = Distance to the previous cut = SD55595

b: Tool length difference due to the bending of the tool = Retraction between grooving (plunging to the next depth) and roughing (turning operation) = SD55596



Contour plunge turning: Retraction depth prior to turning

SD55596 $SCS_TURN_CONT_TURN_RETRACTION	Contour plunge turning: retraction depth prior to turning
Default: 0,1mm	Change to:

Description:

Retraction between grooving and turning in order to compensate for the difference in tool length caused by deflection of the tool (see sketch under SD55595).

Upper limit feedrate/rev

SD55200 $SCS_MAX_INP_FEED_PER_REV	Upper limit feedrate/rev
Default: 1 mm/U	Change to:

Description:

Feedrate input upper limit for mm/rev

Upper limit feedrate/min

SD55201 $SCS_MAX_INP_FEED_PER_TIME	Upper limit feedrate/min
Default: 10000 mm/min	Change to:

Description:

Feedrate input upper limit for mm/min

Upper limit feedrate/tooth

SD55202 $SCS_MAX_INP_FEED_PER_TOOTH	Upper limit feedrate/tooth
Default: 1 mm	Change to:

Description:

Feedrate input upper limit for mm/tooth

Notes:

Setting up the tool list

These settings are applicable only to JobShop programs.

Enable fast M functions

MD52229 $MCS_ENABLE_QUICK_M_CODES	Enable fast M functions
Default: 0H	Change to:

Description:

Bit 0: Enable fast M functions: coolant OFF

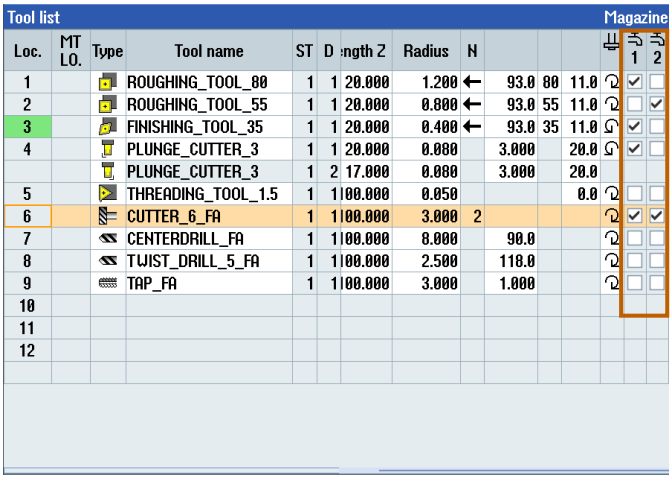
Bit 1: Enable fast M functions: coolant 1 ON

Bit 2: Enable fast M functions: coolant 2 ON

Bit 3: Enable fast M functions: coolants 1 and 2 ON

This MD is used to declare the commands used for the coolant control (see MD52230 - MD52233) as being fast M-commands (M=QU…). In case of fast commands the program progress continues without waiting for the acknowledgement of the M-command by the PLC.

In the JobShop tool list there are input fields for coolant 1 and coolant 2 for each tool. The select key can be used to toggle the function "Coolant ON/OFF". The respective M-functions are put out if the tool is changed by means of the part-program, The M-commands will not be put out if the tool is exchanged by means of the T,S,M mask.



M code for all coolants OFF

MD52230 $MCS_M_CODE_ALL_COOLANTS_OFF	M code for all coolants OFF
Default: 9	Change to:

Description:

M-Code for all coolants OFF

M code for coolant 1 ON

MD52231 $MCS_M_CODE_COOLANT_1_ON	M code for coolant 1 ON
Default: 8	Change to:

Description:

M-Code for coolant 1 ON

M code for coolant 2 ON

MD52232 $MCS_M_CODE_COOLANT_2_ON	M code for coolant 2 ON
Default: 7	Change to:

Description:

M-Code for coolant 2 ON

Notes:

M code for both coolants ON

MD52233 $MCS_M_CODE_COOLANT_ 1_AND_2_ON	M code for both coolants ON
Default: -1	Change to:

Description:

M-Code for Coolant 1 + 2 ON

Already defined: M8 = Coolant 1 ON
M7 = Coolant 2 ON

If in the JobShop tool list both coolants have been selected, the following M-command will be put out.

-1: M-command for coolant 1 OFF is MD52231 and M-command for coolant 2 OFF is MD52232

>0: the M-command selected in this MD.
A separate M-command for both coolants is to be used if the M-commands for coolant 1 and coolant 2 are grouped in an M-commands group, however both coolants are to be activated after a block search.

Function mask Tool management

MD52270 $MCS_TM_FUNKTION_MASK	Function mask Tool management
Default: 0H	Change to:

Description:

Bit 0: Create tool on a magazine location not permitted. Tools can only be created outside the magazine.

Bit 1: Loading/unloading is not possible while the machine is not in Reset.
Tools can only be loaded or unloaded if the respective channel is at rest.

Bit 2: Loading/unloading locked during emergency OFF
Tools can only be loaded or unloaded if the Emergency OFF is not activated.

Bit 3: Loading/unloading tools to/from spindle disabled.
Tools cannot be loaded to or unloaded from the spindle.

Bit 4: Loading takes place in the spindle directly.
The tools are only loaded exclusively in the spindle.

Bit 5: Use grinding configuration file for creating the tool list.
Only grinding tools are offered.

Bit 6: Relocating a tool to/from the buffer (spindle/gripper) is permitted desoite a block (see bit3)

Bit 7: Create tool using the tool number. Specify the tool's T-number when creating the tool.
Note: This function is intended for machines with tool management but without magazine configuration. The default setting must not be changed.

Bit 8: Hide 'Re-locate tool'
The function 'Re-locate tool' is suppressed on the user interface.

Bit 9: Hide 'Magazine positioning'
The function 'magazine positioning' is suppressed on the user interface.

Bit 10: Reactivate tool using Position magazine. Prior to re-activating the tool is positioned at the loading position.

Bit 11: Reactivate tool in all monitoring modes. When re-activating a tool, all monitoring modes enabled in the NC are reactivated for this tool, even the monitoring modes, which have not been set for the relevant tool, but are available in the background only.

Bit 12: Hide 'Tool re-activation'
The function "Tool re-activating" is suppressed on the user interface.

Note:

This MD must be matched to the circumstances of the machine. By hiding those functions on the user interface that are not applicable to the machine, the PLC can be spared unnecessary actions to prevent operation errors..

Notes:

Load station Tool management

MD52274 $MCS_TM_TOOL_LOAD_STATION	Load station Tool management
Default: 0	Change to:

Description:

If more than 1 loading station is configured, a pop-up window appears when a tool is created with a query in which station the tool is to be loaded.

= 0: If a query window is on the user interface or if only one loading station is available there will be no pop-up window

= 1: fixed loading station 1, the window does not appear

= 2: fixed loading station 2, the window does not appear

If only one loading station is configured: The setting = 0
In the turning machine configuration described here there is only one loading station is configured. The query on the right regarding the selection of the loading station does not appear.

Function mask tool management

SD54215 $SNS_TM_FUNCTION_MASK_SET	Function mask Tool management
Default: 0H	Change to: Bit 0 = 1

Description:

Bit 0: Diameter display for rotating tools
It is not the radius value, but the diameter value that is shown for rotating tools.

Bit 1: Default direction of rotation for all turning tools is M4
When turning tools are created, the direction of rotation is preset to M4.

Bit 2: Create tool without name proposal
When creating a new tool the input field for the tool name remains empty.

Bit 3: Input disable for tool name and tool type of loaded tools.
Tool name and tool type cannot be changed for loaded tools.

Description:

Bit 4: Input disable for loaded tools, if the channel is not in Reset.
While the program is in progress the data for loaded tools cannot be changed.

Bit 5: Calculate tool wear inputs additively
The input of wear data will be added to any existing wear values.

Bit 6: Numerical input of the tool ident
No name will be assigned to a new tool being created.
Only numerical characters are permitted .

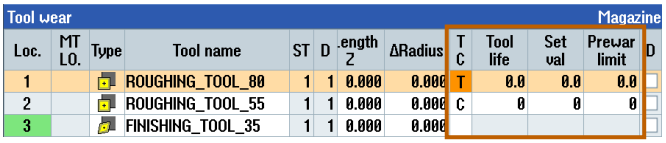
Bit 7: Fade out tool monitoring parameter
The tool monitoring parameter will not be shown on the user interface.

Bit 8: Diameter display for transverse axis - Geometry
The geometry value of the transverse axis is shown as a diameter value.

Bit 9: Diameter display for transverse axis - Wear
The wear value of the transverse axis is shown as diameter value.

Bit 10: Enable loading/ relocating of tools on buffer location
The magazine number can be entered in the loading dialog. This permits access to the buffer station using the magazine number 9998.

Bit 11: Creation of new tools in gripper location is disabled.



Notes:

Continued SD54215

Bit 0 = 0 permanent radius display for all tools

Tool list — Magazine

Loc.	MT LO.	Type	Tool name	ST	D	Length X	Length Z	Radius				
1			ROUGHING_TOOL_80	1	1	100.000	20.000	1.200 ←		93.0	80	1
2			ROUGHING_TOOL_55	1	1	100.000	20.000	0.800 ←		93.0	55	1
3			FINISHING_TOOL_35	1	1	100.000	20.000	0.400 ←		93.0	35	1
4			PLUNGE_CUTTER_3	1	1	100.000	20.000	0.080		3.000		2
			PLUNGE_CUTTER_3	1	2	100.000	17.000	0.080		3.000		2
5			THREADING_TOOL_1.5	1	1	0.000	100.000	0.050				
6			CUTTER_6_FA	1	1	0.000	100.000	3.000	2			
7			CENTERDRILL_FA	1	1	0.000	100.000	8.000		90.0		
8			TWIST_DRILL_5_FA	1	1	0.000	100.000	2.500		118.0		
9			TAP_FA	1	1	0.000	100.000	3.000		1.000		
10												
11												
12												

Bit 0 = 1 Diameter display for rotating tools

Tool list — Magazine

Loc.	MT LO.	Type	Tool name	ST	D	Length X	Length Z	⌀				
1			ROUGHING_TOOL_80	1	1	100.000	20.000	1.200 ←		93.0	80	1
2			ROUGHING_TOOL_55	1	1	100.000	20.000	0.800 ←		93.0	55	1
3			FINISHING_TOOL_35	1	1	100.000	20.000	0.400 ←		93.0	35	1
4			PLUNGE_CUTTER_3	1	1	100.000	20.000	0.080		3.000		2
			PLUNGE_CUTTER_3	1	2	100.000	17.000	0.080		3.000		2
5			THREADING_TOOL_1.5	1	1	0.000	100.000	0.050				
6			CUTTER_6_FA	1	1	0.000	100.000	6.000	2			
7			CENTERDRILL_FA	1	1	0.000	100.000	16.000		90.0		
8			TWIST_DRILL_5_FA	1	1	0.000	100.000	5.000		118.0		
9			TAP_FA	1	1	0.000	100.000	6.000		1.000		
10												
11												
12												

Description:

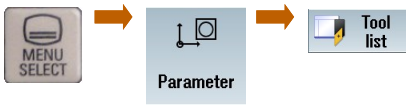
As soon as the cursor is moved to a line with a rotating tool, the term 'Radius' in the header line will be replaced by a diameter symbol.

Bit 1 = 1 M4 Tool 10 has been newly created
Bit 1 = 0 M3 Tool 11 has been newly created

Tool list — Magazine

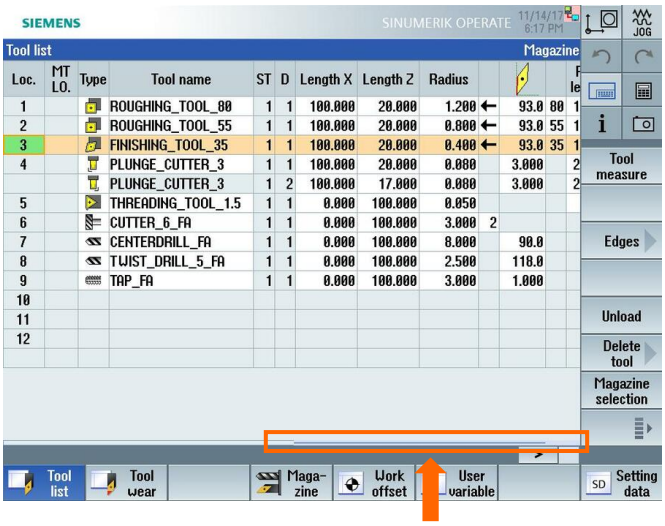
Loc.	MT LO.	Type	Tool name	ST	D	ngth Z	Radius				Pl. leng		1	2
1			ROUGHING_TOOL_80	1	1	20.000	1.200 ←		93.0	80	11.0			
2			ROUGHING_TOOL_55	1	1	20.000	0.800 ←		93.0	55	11.0			
3			FINISHING_TOOL_35	1	1	20.000	0.400 ←		93.0	35	11.0			
4			PLUNGE_CUTTER_3	1	1	20.000	0.080		3.000		20.0			
			PLUNGE_CUTTER_3	1	2	17.000	0.080		3.000		20.0			
5			THREADING_TOOL_1.5	1	1	00.000	0.050				0.0			
6			CUTTER_6_FA	1	1	00.000	3.000	2						
7			CENTERDRILL_FA	1	1	00.000	8.000		90.0					
8			TWIST_DRILL_5_FA	1	1	00.000	2.500		118.0					
9			TAP_FA	1	1	00.000	3.000		1.000					
10			ROUGHING_TOOL_1	1	1	0.000	0.000 ←		95.0	80	11.0			
11			ROUGHING_TOOL_2	1	1	0.000	0.000 ←		95.0	80	11.0			
12														

Checking the tool management

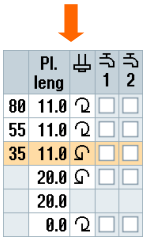


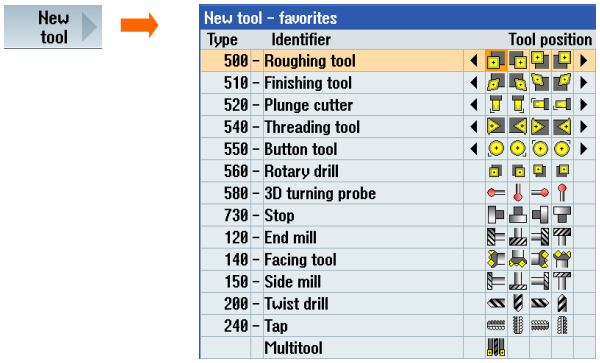
Marked tools can be off-loaded. As soon as the cursor beam is moved to a free magazine location or below the magazine the Softkey "New tool" appears for the creation of a new tool.

Notes:



By horizontal scrolling the fields for the direction of rotation of the tools and the selection of Coolant 1 ON/OFF and Coolant 2 ON/OFF, appear.





The function Multitool, as selected with MD18080 Bit 10 is also included in the selection list for the various types of tools with cutting directions.

Basic settings

Reset-resistant settings

Definition of basic control settings after reset/PP end

MD20110 $MC_RESET_MODE_MASK	Definition of basic control settings after reset/PP end
	Change to: Bit 0 = 1 Bit 6 = 1 Bit 14 = 1

Description:

Determination of the basic control settings after booting and Reset/End-of-part-program with respect to the G-codes (especially the current plane and selectable zero-point offset), tool length and transformation by the setting of the following bits:

Description:

Bit 0: Reset mode
Bit 6: Active tool length compensation
Bit 14: Base frame
Bit 16: Master spindle

The Bits 6 and 16 are only evaluated if Bit 0 = 1.

Meaning of the individual Bits:

Bit 0 = 0:
Initial setting after booting:
- G-Codes as per $MC_GCODE_RESET_VALUES
- Tool length offset not active
- Transformation not active
- no coupled axes groupings active
- no tangential correction active
- no axial rotational feed active
- path rotational feed with master spindle (Default)

Initial setting after Reset or after end-of-part-program :
The current settings will be retained.
With the next part-program Start the following initial settings will become active:

Notes:

Continued MD20110
- G-Codes as per $MC_GCODE_RESET_VALUES
- Tool length offset not active
- Transformation not active
- No coupled-motion groupings active
- No tangential offsets active
- No master value couplings active
- No axial rotational feedrate active
- Path rotational feedrate with master spindle (Default)

Bit 0 = 1:
Initial setting after booting:
- G-Codes as per $MC_GCODE_RESET_VALUES
- Tool length offset active as per $MC_TOOL_RESET_VALUE, $MC_CUTTING_EDGE_RESET_VALUE and $MC_SUMCORR_RESET_VALUE
- Transformation active per $MC_TRAFO_RESET_VALUE
- Geo axis replacement as per $MC_GEOAX_CHANGE_RESET
- No coupled-motion groupings active
- No tangential correction active

Initial setting after reset or after end-of-part-program:
Depending on the setting of $MC_GCODE_RESET_MODE the current settings for the G-groups will be retained or set to the initial values stored in $MC_GCODE_RESET_VALUES.

Description:

Initial setting after Reset or after end-of-part program:
Depending on $MC_RESET_MODE_MASK Bit 6 to 7 the current settings for
- Tool length offset
- Transformation

will either be retained or changed to the initial settings as stored in the MDs.

Depending on Bit 8 and 9 the current settings of coupled motion axes or tangentially corrected axes will either be retained or switched OFF.

Projected synchronous spindle coupling:
The coupling will be de-selected depending on the setting of $MC_COUPLE_RESET_MODE_1.

Non-projected synchronous spindle coupling:
Depending on Bit 10 the coupling will either be switched OFF or retained.
Depending on Bit 14 the initial frame will be retained or de-selected.

Note:
The setting of Bit 0 = 1 must be set for JobShop-uses, such that the settings of the Bits 4 and 16 can be evaluated.

Continued MD20110

Bit 6 = 0:
Initial setting for the active tool length offset after Reset/End-of-part program as per $MC_TOOL_RESET_VALUE, $MC_CUTTING_EDGE_RESET_VALUE, $MC_USEKT_RESET_VALUE and $MC_SUMCORR_RESET_VALUE.
Ist $MC_TOOL_CHANGE_MODE = 1, the tool specified by by $MC_TOOL_PRESEL_RESET_VALUE is additionally pre-selected.
If the tool and/or magazine management is active, it is not the datum $MC_TOOL_RESET_VALUE that is used, instead $MC_TOOL_RESET_NAME will be used.

Bit 6 = 1:
The current setting for the active tool length offset remains in force after reset/End-of-part-program.
If the tool and/or magazine management is active, that tool will be selected which is presently loaded in the master spindle (generally = master tool holder).
If the tool in the master spindle happens to be blocked, the "blocked"-status will be ignored. .

However, it must be remembered that after the end of program or a program interruption either the last programmed value for the master spindle or master tool holder, or the value specified by $MC_SPIND_DEF_MASTER_SPIND or $MC_TOOL_MANAGEMENT_TOOLHOLDER will determine the master spindle or the master tool holder.
(The selection is made by Bit 16 or Bit 17.)

A special case is $MC_CUTTING_EDGE_DEFAULT = -2:
If a tool has been changed on the spindle, but no new correction D has been programmed, then the previous tool is still active in the NCK.
If an interruption takes place in this state, - e.g. by pressing the reset key, the correction will be specified by the lowest D-number of the tool in the master spindle.

Note:
The setting Bit 6=1 is necessary to ensure that after a reset/ End-of-part-program the previously programmed tool and the previously programmed cutting edge remains activated (for instance, if the tool was changed via the T,S,M-mask) and if for example the JobShop-functions in the mode of operation JOG "Zero point work piece" or "Tool measurement" can be carried out with this tool.

Notes:

Continued MD20110

Bit 14 = 0:
The basic frame is cancelled

Bit 14 = 1:
The current setting for the basic frame is retained.

Note:
The setting Bit 14 = 1 is necessary to retain the channel-specific basic offset, which was activated with G500, after a Reset/End-of-part-program.
After activation of the basic offset (JOG-T,S,M-mask - zero point offset - basic reference) the values in the zero point offset base are transferred to the overall base Total WO and then from there to be transferred for activation also to the overall Total WO.

Bit 16 = 0:
Initial setting for the master spindle as per $MC_SPIND_DEF_MASTER_SPIND.

Bit 16 = 1:
The current setting of the master spindle (SETMS) is retained. This Bit in case of $MC_TOOL_MANAGEMENT_TOOLHOLDER = 0 has also an effect on the response of Bit 6.

Note:
The setting of Bit 16 = 0 is important for ShopTurn-machines. The following spindles are used on the machine as configured here:
- Main spindle
- Tool spindle
and in a further module
- Counter spindle
The command SETMS(..) is used within cycles for addressing the spindles (M and S-command).

The previously described settings in MD20110 correspond to the further MD-settings in:

MD20120 $MC_TOOL_RESET_VALUE
MD20130 $MC_CUTTING_EDGE_RESET_VALUE
MD20150 $MC_GCODE_RESET_VALUES
MD20152 $MC_GCODE_RESET_MODE
MD20140 $MC_TRAFO_RESET_VALUE
MD20112 $MC_START_MODE_MASK
MD20121 $MC_TOOL_PRESEL_RESET_VALUE
MD20118 $MC_GEOAX_CHANGE_RESET

Tool edge with length compensation during run-up (reset/end of pp)

MD20130 $MC_CUTTING_EDGE_RESET_VALUE	Tool edge with length compens. during runup (reset/end of pp)
	Change to: 1

Description:

Specification of the cutting edge, which depending on MD20110 $MC_RESET_MODE_MASK and after part program Start depending on MD20112 $MC_START_MODE_MASK is used to select the tool length offset after booting and after Reset /End-of-part-program.

With active tool management and with Bit 0 and Bit 6 set in MD20110 $MC_RESET_MODE_MASK at selection, the last offset of the tool active at power OFF (as a rule the tool in the spindle) is effective after booting.

Corresponds to:
MD20110 $MC_RESET_MODE_MASK
MD20112 $MC_START_MODE_MASK

Note:

This setting is necessary to ensure that after a reset/End-of-part-program the previously programmed tool and the previously programmed cutting edge remains activated (for instance, if the tool was changed via the T,S,M-mask) and if for example the JobShop-functions in the mode of operation JOG "Zero point work piece" or "Tool measurement" can be carried out with this tool.

Notes:

Initial setting of G groups

MD20150[x] $MC_GCODE_RESET_ VALUES	Initial setting of G groups
	Change to: [5] = 2 [15] = 3 [21] = 2 [28] = 2 [48] = 3

MD20150[5] = 2 (G18)
MD20150[15] = 3 (CFIN)
MD20150[21] = 2 (CUT2DF)
MD20150[28] = 2 (DAIMON)
MD20150[48] = 3 (PTPG0)

Description:

Specification of the G-Codes, which after booting and Reset/ End-of-part program are activated depending on
MD20110 $MC_RESET_MODE_MASK and
MD20152 $MC_GCODE_RESET_MODE
as well as after part program Start depending on
MD20112 $MC_START_MODE_MASK.

Note:
(exact) means that this setting is necessary in conjunction with JobShop; (variable) means that this should be seen as a suggestion and that another setting is permissible.

MD20150[5] = 2 (exact)
Group 6: Plane selection
Value: 2 = G18
The is the basic setting for turning.

MD20150[15] = 3 (variable)
Group 16: Feedrate correction on internal and external curves
Value: 3 = CFIN
This setting provokes a constant feedrate on internal curves and acceleration on external curves.

MD20150[21] = 2 (variable)
Group 22: Tool correction type
Value: 2 = CUT2DF
2½-D-tool correction caused by Frame
The tool correction acts relative to the current Frame (oblique plane)

MD20150[28] = 2 (exact)
Group 29: Radius/diameter programming
Value: 2 = DIAMON
Modal diameter programming ON
The effect is independent of the programmed measuring mode (G90/G91).
The diameter programming is shown on the user interface by means of a diameter symbol in front of the X-axis readout.

MD20150[48] = 3 (variable)
Group 49: Point-to-point-motion
Value: 3 = PTPG0
Point-to-point-motion only with G0, otherwise continuous path motion

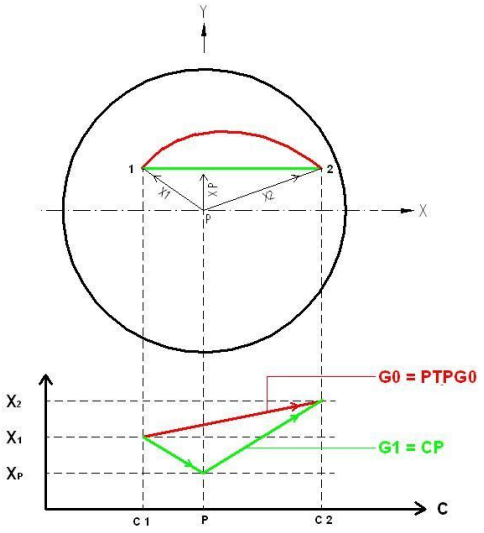
Notes:

Continued MD20150

Description:

The sketch below illustrates the setting PTPG0.
In the course of a facing operation travelling from one Point 1 to another Point 2 is intended.
When travelling at feedrate (G1) an interpolation between the X-axis and the C-axis takes place. This results in the behaviour of the X-axis as shown in the lower sketch in that the axis firstly travels from the starting point X1 in the minus-direction (the distance towards the centre point reduces) to the point XP and then from there in the plus-direction to the destination Point X2. In conjunction with the interpolation with the C-axis in effect a straight line is being traversed from Point 1 to Point 2. This behaviour corresponds to the programming of CP (continuous path).

When travelling at rapid traverse rate (G0) the motion of the X -axis first in the negative and then in the positive direction is not very sensible. By the setting of PTPG0 the axis will move directly from the starting point X1 to the destination point X2. When projected onto the front face of the work piece this represents a spiral-like motion.



Notes:

Reset response of G groups

MD20152[x] $MC_GCODE_RESET_MODE	Reset response of G groups
	Change to: [5] = 0 [7] = 1 [12] = 0 [14] = 1 [15] = 0 [21] = 0

Description:

This machine datum is evaluated only if in MD20110 $MC_RESET_MODE_MASK the Bit 0 is set. For each entry in MD20150 $MN_GCODE_RESET_VALUES (that is for each G group) this MD is used to specify whether

= 0: in case of a Reset/End-of-part-program the setting as per $MC_GCODE_RESET_VALUES is used again
or
= 1: the current setting is retained after Reset/End-of-part-program

Note:
(exact) means that this setting is necessary in conjunction with JobShop; (variable) means that this should be seen as a suggestion and that another setting is permissible.

MD20152[5] = 0 (exact)
Group 6: Plane selection
Value: 0
A plane switch (e.g. G17 for facing) programmed in the part-program remains active after Reset/ End-of-part-program.

MD20152[7] = 1 (exact)
Group 8: Selectable zero point offset
Value: 1
A zero point offset (e.g. G54) programmed in the part program remains active after Reset/End-of-part-program. This permits for example the activation of a zero point offset in the mode-of-operation JOG_T,S,M-mask Parameter zero point offset.

Description:

MD20152[12] = 0 (exact)
Group 13: Work piece dimensioning inch/metric
Value: 0
In the user interface it is possible in the mode-of-operation JOG Basic strip ">" Settings per Softkey "Switching Inch" to toggle the measuring system from metric to inch and vice versa.

MD20152[14] = 1 (exact)
Group 15: Type of feed
Value: 1
The last type of feed G93 to G97 to be programmed remains active after Reset/End-of-part -program.
In the user interface it is possible in the mode-of-operation JOG - Position to position the axes with rapid traverse or at feedrate in mm/min or mm/rev. The type of feed programmed in this mask remains active.

MD20152[15] = 0 (exact)
Group 16: Feed correction on internal or external curves
Value: 0
After Reset/End-of-part-program the feed correction defined in MD20150[15] is activated once more.

MD20152[21] = 0 (variable)
Group 22: Tool correction type
Value: 0
After Reset/End-of-part-program the tool correction defined in MD20150[15] is activated once more. CUT2DF in this configuration example.

Notes:

Definition of tool parameters

MD20360 $MC_TOOL_PARAMETER_ DEF_MASK	Definition of tool parameters
	Change to: Bit 0 = 1

Description:

Definition of the effect of tool parameters.
Bit Nr. Meaning when the Bit is set.

Bit 0: For turning and grinding tools, the wear parameter of the transverse axis is included in the calculation as a diameter value.

Bit 1: For turning and grinding tools, the tool length component of the transverse axis is included in the calculation as a diameter value.

Bit 2: If a wear component or a length component is included in the calculation as a diameter value, the tool may only be used in the plane that was active when the tool was selected. If the bit is set, a plane change raises an alarm

Bit 3: Zero point offsets of Frames in the transverse axis will be evaluated as diameter values.

Bit 4: PRESET value is evaluated as diameter value.

Bit 5: Evaluate external zero point offsets in the transverse axis as diameter values.

Bit 6: Read actual values of the transverse axis as diameter value (AA_IW, AA_IEN, AA_COMM, AA_IB, Caution: but not AA_IM)

Bit 7: Readout of all actual values of the transverse axis as diameter values independently of the G-Code of the Group 29 (DIAMON / DIAMOF)

Bit 8: Display of the distance to go in WCS always as a radius.

Bit 9: During DRF-handwheel traverse of an axis only half of the specified increment is actually travelled. (providing MD11346 $MN_HANDWH_TRUE_DISTANCE = 1)

Bit 10: Permit activation of the tool component of an active indexable tool carrier even if no tool is active..

Bit 11: The tool parameter $TC_DP6 is not interpreted as tool radius, but as a tool diameter.

Bit 12: The tool parameter $TC_DP15 is not interpreted as wear on the tool radius, but as wear of the tool diameter.

Bit 13: During JOG of circles, the circle centre Coordinate is always a radius value, see SD42690 SC_JOG_CIRCLE_CENTRE.

Bit 14: Absolute values of the transverse axis for cycle masks as a radius

Bit 15: Incremental values of the transverse axis for cycle masks as diameters.

Note:

For turning machines it is sensible to enter and evaluate the tool wear correction value for turning tools in the X-axis as a diameter, since the wear is determined on the work piece diameter.

Maximum number of blocks w/o traversing movement with SAR

MD20202 $MC_WAB_MAXNUM_ DUMMY_BLOCKS	Maximum number of blocks w/o traversing movement with SAR
	Change to: 10

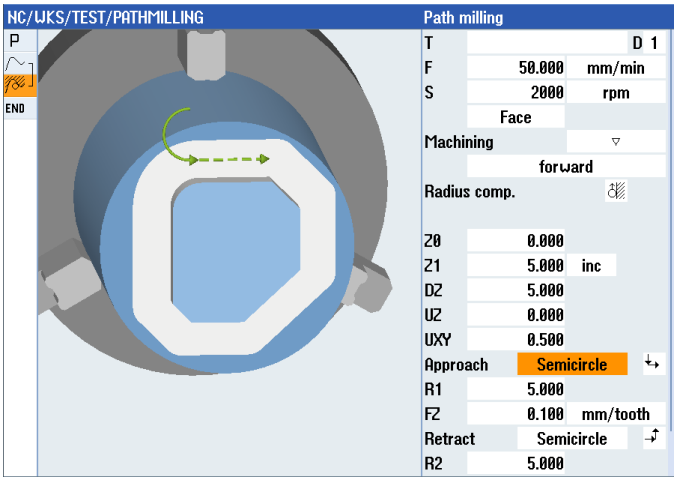
Description:

Maximum number of blocks that can appear between the SAR (soft approach and retraction) block and the traversing block which determines the direction of the approach or retraction tangent.

This setting is required in conjunction with the JobShop-cycles. A typical use is illustrated by the screenshot below taken from a continuous path milling operation on a contour using the approach and leaving strategy "Semi-circle".

Notes:

Continued MD20202



Blocks for look-ahead contour calculation with TRC

MD20240 $MC_CUTCOM_MAXNUM_ CHECK_BLOCKS	Blocks for look-ahead contour calculation with TRC
Default: 4	Change to:

Description:

Indicates the maximum number of blocks with traversing information at the offset plane that are considered simultaneously for collision detection with active TRC (tool radius compensation).

Note:

The Standard setting should not be undercut.

Maximum number of blocks without traversing motion in TRC

MD20250 $MC_CUTCOM_MAXNUM_ DUMMY_BLOCKS	Maximum number of blocks without traversing motion in TRC
	Change to: 5

Note:

The value 5 represents a value based on experience for JobShop installations.

Time monitoring for tool in tool holder

MD20320 $MC_TOOL_TIME_MONITOR _MASK	Time monitoring for tool in tool holder
	Change to: Bit 0 = 1

Description:

The setting Bit 0 = 1 is required for the life-time monitoring of the tool in the tool holder 1. The machine configuration used here only uses one tool holder.

Notes:

Active system frames after reset

MD24006 $MC_CHSFRAME_RESET_MASK	Active system frames after reset
	Change to: Bit 0 = 1 Bit 5 = 0

Description:

Bit-mask for the reset setting of the channel-specific system frames, which will be evaluated with the channel.

Bit 0: System frame active for preset. actual value/ scratching

Bit 5: System frame for cycles

Deletion of system frames after reset

MD24007 $MC_CHSFRAME_RESET_CLEAR_MASK	Deletion of system frames after reset
	Change to: Bit 0 = 0 Bit 5 = 1

Description:

Bit-mask for deleting channel-specific system frames in the data management on Reset..

Bit 0 = 0: The system frame for setting actual values and scratching (basic reference) will not be deleted with Reset/ End-of-part-program.

Bit 5 = 1: The system frame for cycles will be deleted with Reset/End-of-part-program.

Reset channel system frames after power ON

MD24008 $MC_CHSFRAME_POWERON_MASK	Reset channel system frames after power ON
	Change to: Bit 0 = 0 Bit 5 = 0

Description:

This machine data defines whether channel-specific system frames are reset in the data management on Power On. That is offsets and rotations are set to 0, scalings to 1. Mirroring is disabled.

The selection can be made separately for individual system frames.

Bit 0: System frame for set actual value and scratching is deleted after Power On.

Bit 5: System frame for cycles retained after Power On.

Adaptation of active frames

MD24040 $MC_FRAME_ADAPT_MODE	Adaptation of active frames
	Change to: Bit 0 = 1 Bit 1 = 1 Bit 2 = 1

Description:

Bit-mask for matching the active frame to the axis constellation

Bit 0: Adjustment of rotations.

Bit 1: Rectangular shear angle

Bit 2: Geo axes are not scaled

Note:

This setting is absolutely necessary for the function the ShopTurn cycles in conjunction with TRANSMIT/TRACYL.

Notes:

Activation and impact of program runtime measurement

MD27860 $MC_PROCESSTIMER_ MODE	Activation and impact of program runtime measurement
	Change to: Bit 0 = 1 Bit 1 = 1 Bit 4 = 1 Bit 5 = 1 Bit 6 = 1 Bit 8 = 1

Description:

Under the function program runtime timers are provided as system variables. While the NCK-specific timers are always activated (for timings since the last control unit boot), the channel-specific timers must be started by means of this datum.

Meaning:

Bit 0 = 0
No timing of the total runtime for all part programs

Bit 0 = 1
The timing of the total runtime for all part programs is active ($AC_OPERATING_TIME)

Bit 1 = 0
No timing of the current program runtime

Bit 1 = 1
The measurement of the current program run-time is active ($AC_CYCLE_TIME)

Bit 2 = 0
No timing of the tool cutting time

Bit 2 = 1
Timing of the tool cutting time is active ($AC_CUTTING_TIME)

Bit 3
Reserved

Bits 4,5 only if Bit 0, 1 = 1:

Bit 4 = 0 No timing during active test run feed motion

Bit 4 = 1 Timing also during active test run feed motion

Bit 5 = 0 No timing during program test

Bit 5 = 1 Timing also during program test

Description:

Bit 6 only if Bit 1 = 1:

Bit 6 = 0
Delete $AC_CYCLE_TIME also after Start by ASUP and PROG_EVENTs.

Bit 6 = 1
$AC_CYCLE_TIME will not be deleted after Start by ASUP and PROG_EVENTs.

Bit 7 only if Bit 2 = 1:

Bit 7 = 0 $AC_CUTTING_TIME counts only with active tool.

Bit 7 = 1 $AC_CUTTING_TIME counts independently of tool.

Bit 8 only if Bit 1 = 1:

Bit 8 = 0
$AC_CYCLE_TIME will not be deleted in case of a jump with GOTOS to the program start.

Bit 8 = 1
$AC_CYCLE_TIME will be deleted in case of a jump with GOTOS to the program start.

Bit 9 only if Bit 0, 1 = 1:

Bit 9 = 0
$AC_OPERATING_TIME, $AC_CYCLE_TIME: No timing when Override = 0.

Bit 9 = 1
$AC_OPERATING_TIME, $AC_CYCLE_TIME: Timing also if Override = 0.

Bit 10 to 31
Reserved

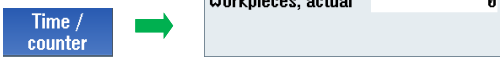
Notes:

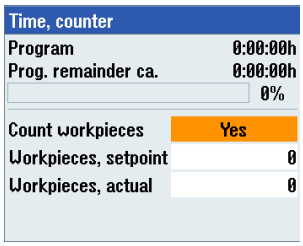
Work piece counters

A difference is made between the following work piece counters:

- Number of the command work pieces
  System variable $AC_REQURED_PARTS
  Setting by means of Bit 0 and 1
- Total number of work pieces (actual work pieces) finished since the Start
  System variable $AC_TOTAL_PARTS
  Setting by means of Bit 4, 5 and 7
- Number of all work pieces (actual work pieces) finished since the Start
  System variable $AC_ACTUAL_PARTS
  Setting by means of Bit 8, 9 und 11
- Number of work pieces finished as per the own strategy
  System variable $AC_SPECIAL_PARTS
  Setting by means of Bit 12, 13 and 15

In the user interface of the mode-of-operation AUTO the mask shown alongside can be opened by selecting Time/ Counter





The following attribution applies:
- Param "Workpiece Command" = $AC_REQUIRED_PARTS
- Param "Work pieces Actual" = $AC_ACTUAL_PARTS

Activation of workpiece counter

MD27880 $MC_PART_COUNTER	Activation of workpiece counter
	Change to: Bit 0 = 1 Bit 8 = 1 Bit 11 = 1

Bit 0 = 1: Counter $AC_REQUIRED_PARTS is activated
Further meaning of Bit 1-3 only if Bit 0 =1 and $AC_REQUIRED_PARTS > 0:

Further meaning of Bit 1 - 3 only if Bit 0 = 1 and $AC_REQUIRED_PARTS > 0:

Bit 1 = 0: Alarm/VDI-output if $AC_ACTUAL_PARTS corresponds to $AC_REQUIRED_PARTS

Bit 1 = 1: Alarm/VDI-output if $AC_SPECIAL_PARTS corresponds to $AC_REQUIRED_PARTS

Bit 2: reserved

Bit 3: reserved

Bit 4 = 1: Counter $AC_TOTAL_PARTS is active

Further meaning of Bit 5 - 7 only if Bit 4 = 1 and $AC_TOTAL_PARTS > 0:

Bit 5 = 0: Counter $AC_TOTAL_PARTS is increased by the value 1 after a VDI-output of M02/M30.

Bit 5 = 1: Counter $AC_TOTAL_PARTS is increased by the value 1 after the output of the M-command from MD PART_COUNTER_MCODE[0]

Bit 6 : reserved

Bit 7 = 1: Counter $AC_TOTAL_PARTS is increased by the value 1 after a jump back with GOTOS

Bit 8 = 1: Counter $AC_ACTUAL_PARTS is active

Further meaning of Bit 9 - 11 only if Bit 8 = 1 and $AC_REQUIRED_PARTS > 0:

Bit 9 = 0: Counter $AC_ACTUAL_PARTS is increased by the value 1 after a VDI-output of M02/M30

Bit 9 = 1: Counter $AC_ACTUAL_PARTS is increased by the value 1 after the output of the M-command from MD PART_COUNTER_MCODE[1]

Bit 10: reserved

Bit 11 = 1: Counter $AC_ACTUAL_PARTS is increased by the value 1 after a jump back with GOTOS.

Notes:

Continued MD27880

Bit 12 = 1: Counter $AC_SPECIAL_PARTS is active

Further meaning of Bit 13-15 only if Bit 12 =1 and $AC_REQUIRED_PARTS > 0:

Bit 13 = 0: Counter $AC_SPECIAL_PARTS is increased by the value 1 after a VDI-output of M02/M30

Bit 13 = 1: Counter $AC_SPECIAL_PARTS is increased by the value 1 after output of the M-command from MD PART_COUNTER_MCODE[2]

Bit 14: reserved

Bit 15 = 1: Counter $AC_SPECIAL_PARTS is increased by the value 1 after a jump back with GOTOS.

Application example:

MD27880 PART_COUNTER = 101H
Bit 0 = 1: Counter $AC_REQUIRED_PARTS "Work pieces Command" activate
Bit 8 = 1: Counter $AC_ACTUAL_PARTS "Work pieces Actual" activate

MD 27882[1] PART_COUNTER_MCODE = 33

Part program:

In such a JobShop-Program the program repeat can be switched ON in the automatically generated End-of-program block by means of the parameter "Repeat". In this case the command GOTOS will be generated.

G-Code-Program_1:	G-Code-Program_2:
…	...
M33; Repeat	GOTOS; Repeat
M30	M30

The number of work pieces is to be set in the mode-of-operation AUTO in the mask “Time/Counter” by the parameter “Workpieces, setpoint”.

Relevant PLC interface signals:

Activating the part program repetition is done by setting the PLC interface signal "Program branches control" to 1.
- For ONE / 840D sl: PLC bit DB21.DBX384.0
- For 828D: PLC bit DB3200.DBX16.0

When the actual number of workpieces matches the number of setpoint the PLC interface signal "Workpiece setpoint reached" is set to 1.
- For ONE / 840D sl: PLC bit DB21.DBX317.1
- For 828D: PLC bit DB3300.DBX4001.1

Memory space for REORG (DRAM)

MD28000 $MC_MM_REORG_LOG_FILE_MEM	Memory space for REORG (DRAM)
	Change to: 75

Number of blocks for local user variables in REORG (DRAM)

MD28010 $MC_MM_NUM_REORG_LUD_ MODULES	Number of blocks for local user variables in REORG (DRAM)
	Change to: 20

Projecting of channel-specific system frames

MD28082 $MC_MM_SYSTEM_FRAME_MASK	System frames (SRAM)
	Change to: Bit 0 = 1 Bit 5 = 1 Bit 6 = 1

Description:

Bit-mask for the projecting of channel-specific system frames, which are evaluated in the channel.

Bit 0 = 1 System frame for actual value set and scratching
Bit 5 = 1 System frame for cycles
Bit 6 = 1 System frame for transformations

Note:
The system frames for actual value set and scratching (basic reference) and for cycles are required for JobShop installations.

Projecting of channel-specific system frames in the data management

MD28083 $MC_MM_SYSTEM_DATAFRAME_MASK	System frames (SRAM)
	Change to: Bit 0 = 1 Bit 5 = 0

Notes:

Work offsets in frames

SD42440 $SC_FRAME_OFFSET_INCR_PROG	Work offsets in frames
	Change to: 0

Note:

This setting is required for JobShop installations.

Tool length compensations

SD42442 $SC_TOOL_OFFSET_INCR_PROG	Tool length compensations
	Change to: 0

Note:

This setting is required for JobShop installations.

Feed reduction on circles with tool radius compensation

SD42528 $SC_CUTCOM_DECEL_LIMIT	Feed reduction on circles with tool radius compensation
	Change to: 1

Note:

This setting is required for JobShop installations. It avoids the "stopping" of the axes in such circular sections if the diameter of the milling cutter is only marginally smaller than the machining radius.

Change of tool length components with change of active plane

SD42940 $SC_TOOL_LENGTH_CONST	Change of tool length components with change of active plane
	Change to: 18

Note:

The setting = 18 is required for ShopTurn installations, see also SD42950 $SC_TOOL_LENGTH_TYPE.

Assignment of tool length compensation independent of tool type

SD42950 $SC_TOOL_LENGTH_TYPE	Assignment of tool length compensation independent of tool type
	Change to: 2

Note:

The setting = 2 is required for ShopTurn installations, see also SD42940 $SC_TOOL_LENGTH_CONST.

With the setting the setting data
SD42940 $SC_TOOL_LENGTH_CONST and
SD42950 $SC_TOOL_LENGTH_TYPE
the tool lengths are rigidly assigned to the geometry axes independently of the tool type:
Length 1 = X
Length 2 = Z
Lange 3 = Y

Notes:

JOG-TSM

In the operation mode JOG the input masks for "T,S,M", "Position" and "Roughing" are available. These permit simple functions to be carried out such
- T,S,M: Tool changes, spindle functions, work offset selections
- Position: Axes and spindles positioning
- Roughing: Roughing of a workpiece.

After the data have been entered and NC-Start initiated the file MA_JOG_STEP1.MPF will be generated in the directory "Workpieces - TEMP" and carried out in the program-manager.

To ensure that such functions as e. g. tool selection and spindle Start will remain activated also after reset and end-of, respective machine data are required, which will be described below.
The settings will be described with the T,S,M-Mask; the input masks Position and Roughing do not require special settings.

Tool changes

Definition of basic control settings after reset/PP end

MD20110 $MC_RESET_MODE_MASK	Definition of basic control settings after reset/PP end
	Change to: Bit 6 = 1

Bit 0 = 1 Reset mode
Bit 6 = 1 Active tool length compensation
Bit 14 = 1 Base frame

Description:

Bit 6 = 1:

The current setting for the active tool length correction remains active also after Reset/end-of-part program.
If the tool or magazine management is active, that tool is selected which happens to be in the master spindle (generally referred to as master-tool holder).
If the tool in the master spindle is locked, the "locked" status will be ignored.

It must be borne in mind that after end-of-program, program abort either the last programmed value for the master spindle respectively the master tool holder or else the value specified by $MC_SPIND_DEF_MASTER_SPIND respectively $MC_TOOL_MANAGEMENT_TOOLHOLDER for the master spindle respectively the master-tool holder will be used.

Activation of tool management functions

MD20310 $MC_TOOL_MANAGEMENT_MASK	Activation of tool management functions
	Change to: Bit 14 = 1

Bit 0 - 3 must be set as in MD18080 $MN_MM_TOOL_MANAGEMENT_MASK

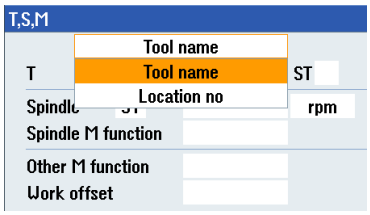
Bit 0 = 1 Magazine management active
Bit 1 = 1 Monitoring functions active
Bit 3 = 1 Adjacent location treatment
Bit 14 = 1 autom. tool change during Reset and Start
Bit 16 = 1 Programming T = "Location number"

Description:

Bit 14 = 1: Reset-Mode
Tool and correction selection as per the settings of MD20110 $MC_RESET_MODE_MASK and MD20112 $MC_START_MODE_MASK.

Note:

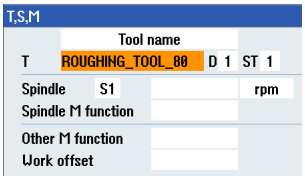
In case of turning machines featuring a turret the setting with Bit 16=1 "Programming T=Location number" the programming of the turret place can be selected in addition to the tool name. In the T,S,M-mask appears the following selection means:



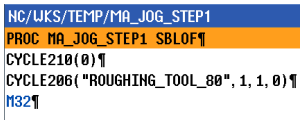
Notes:

Continuation of tool change

Selection and transfer of a tool from the list of tools in the T,S,M-mask (the direct input of the name is also possible)



With NC-Start the part-program MA_JOG_STEP1 is generated in directory Work pieces - TEMP and the tool change is carried out.

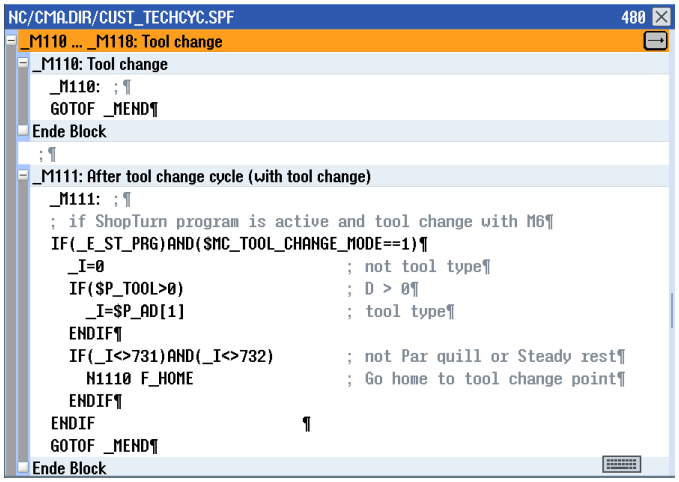


Description:

- CYCLE210(0)
  checks the configuration settings for ShopTurn, as for instance axis and spindle assignment.
- CYCLE206
  obtains the tool name and the cutting edge number as transfer parameters.
- M32
  The M-command defined under MD10714

The cycle CYCLE206 called up with the tool change in turn calls the manufacturer cycle CUST_TECHCYC in the mode 110 and 111.

Extract from the CUST_TECHCYC.SPF:

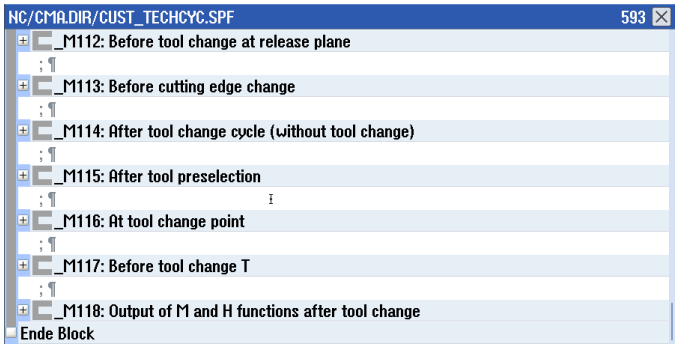


Description:

_M110: This mode is called up after the tool change. No entries are prepared.

_M111: This jump is provided in case the tool change takes place from a ShopTurn-program and the tool change is set up with the M-command (MD22550 $MC_TOOL_CHANGE_MODE = 1). The ShopTurn-cycle F_HOME.SPF for the approach of the tool changing position will be called up.

Further jump accesses in the CUST_TECHCYC are provided for the tool change, which, however, are called up only during the program execution.



Description:

_M112: Is called up before the tool change when the retraction plane is reached (the retraction plane is defined for the X and Z-axis in the part program)

_M113: Is called up if just a cutting edge, but no tool, has been programmed.
Here a special case is being considered where on one tool there is a cutting edge for external and another or internal machining. If now a change between these two cutting edges is programmed (together with a simultaneous change from external to internal machining) there will be no approach of the tool changing point from the ShopTurn cycles. The change of the cutting edge takes place close to the work piece and the starting point of the internal machining will be approached along the shortest possible path. In order to avoid collisions with the work piece the ShopTurn cycle F_HOME for the approach of the tool changing point is called up.

_M114: Is called up after the tool change.

Notes:

Spindle

These settings are required to enable spindles to be started from the T,S,M mask.

M function for spindle active after reset

MD10714 $MN_M_NO_FCT_EOP	M function for spindle active after reset
	Change to: 32

Suggestion M32

Description:

With Start in the manual mode-of-operation (T,S,M-mask, position) the part-program MA_JOG_STEP1 will be generated and executed.
This part-program will be finished with the M-function in MD10714. The setting of MD35040 = 2 ensures that the spindle remains active also after the end-of-program.

Limitations see MD10715 $MN_M_NO_FCT_CYCLE

Own spindle RESET

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET	Own spindle RESET
	Change to: 2

AX3: MCS1
MD35040 = 2 Own spindle-RESET (Main spindle)

AX4: MCS3
MD35040 = 2 Own spindle-RESET (Tool spindle)

Description:

This machine data defines the response of the spindle after Reset and program end (M2, M30).

This MD is only active in the spindle mode open-loop control mode. In the positioning or oscillation mode, the spindle is always stopped.

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET = 0:

- Spindle stops (with M2/M30 and channel and mode group reset)
- Program is aborted

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET= 1:

- Spindle does not stop
- Program is aborted

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET= 2:

- Spindle does not stop at the M function configured via MD10714 $MN_M_NO_FCT_EOP (e.g. M32).
- However, the spindle stops at channel or mode group reset.

Not relevant to:

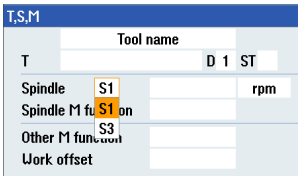
- Spindle modes other than open-loop control mode.

Notes:

Start main spindle via the T,S,M-mask

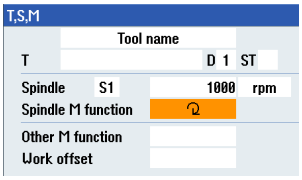
T,S,M-mask with inputs for spindle speed and direction of rotation of the main spindle S1

Main spindle S1 select



Enter speed and direction of rotation

and actuate NC-Start



MA_JOG_STEP1 is generated

```
NC/WKS/TEMP/MA_JOG_STEP1
PROC MA_JOG_STEP1 SBLOF¶
CYCLE210(0)¶
¶
SETMS(1)¶
¶
CUST_TECHCYC(4)¶
CUST_TECHCYC(2)¶
M1=3 G972 S1=1000 ¶
M32¶
```

Description:

- SETMS(1)
  declares the main spindle S1 as master spindle
- CUST_TECHCYC(4)
  Call-up of CUST_TECHCYC in mode 4 "Main spindle C-axis clamping release"
- CUST_TECHCYC(2)
  Call-up of CUST_TECHCYC in mode 2 "Main spindle switch to spindle mode-of-operation"
- M1=3 G972 S1=1000
  Starting of the spindle

Note on spindle direction of rotation:
The assignment of the direction arrow for the spindle M-function M3/M4 is effected via the MD52207 $MCS_AXIS_USAGE_ATTRIB.

Listed here are all the CUST_TECHCYC jump accesses envisaged for the main spindle.

Extract from the CUST_TECHCYC.SPF:

```
NC/CMA.DIR/CUST_TECHCYC.SPF 191
_M1 ... _M9: Main spindle
  _M1: Main spindle: Change to C axis mode
  ;¶
  _M2: Main spindle: Change to spindle mode
    _M2:  ;¶
    IF($P_SMODE[_F_S_NR[0]]<>1); Spindle is not in spindle mode¶
      ;N20 M[_F_S_NR[0]]=          ; Change to spindle mode¶
    ENDIF¶
    GOTOF _MEND¶
  End of group
  ;¶
  _M3: Main spindle: C axis clamping
  ;¶
  _M4: Main spindle: Release C axis clamping
    _M4:  ;¶
    ;N40 M[_F_S_NR[0]]=          ; Release C axis clamping¶
    ;N45 $AC_PRTIME_A_INC=1.5    ; increase auxillary time by 1.5 se
conds¶
    GOTOF _MEND¶
  End of group
```

Description:

For the switching of the main spindle into the C-axis mode or spindle mode and for clamping or unclamping of the main spindle the following jump accesses into the manufacturer cycle CUST_TECHCYC.SPF are prepared:

_M1: Switching the main spindle into the C-axis mode.

_M2: Switching from the C-axis mode into the spindle mode

_M3: Main spindle clamp

_M4: Main spindle unclamp

Notes:

Continued, Start main spindle via the T,S,M-mask

Description:

_M1: Switching to the C-axis mode
When switching into the C-axis mode there is a difference whether in the C-axis mode the main spindle is driven by the spindle drive or by a separate feed drive motor.

1. No separate feed drive available:
The following entry is prepared:

```
IF (($P_SMODE[_F_S_NR[0]]<>2)AND($P_SMODE
[_F_S_NR[0]]<>4))OR($P_PROG_EVENT==5)
  N10 SPOS[_F_S_NR[0]]=0
ENDIF
```

Variable _F_S_NR: The spindle index of the main, the tool and the counter spindle can be freely defined by the machine manufacturer. In order to permit the CUST_TECHCYC to be generally valid for all applications, the spindle index will be determined via the variable _F_S_NR from the given axis configurations (set by means of MD52206 $MCS_AXIS_USAGE). The following applies:
_F_S_NR[0] holds the spindle index of the main spindle
_F_S_NR[1] holds the spindle index of the tool spindle
_F_S_NR[2] holds the spindle index of the counter spindle

From this derives the following logic:
The IF-query is fulfilled, if the main spindle is not in the mode-of-operation 2 = positioning and not in the mode-of-operation 4 = axis mode-of-operation or the Prog_Event is in the mode 5 = block search.
The subsequent line positions the main spindle to 0 degrees:
SPOS[_F_S_NR[0]]=0
The query ensures that the main spindle is not positioned again, if it had already been switched into the axis mode beforehand.

2. Separate feed drive fitted:
The IF-query described above and the subsequent ENDIF-command must be invalidated and replaced by the logic for the engaging of the separate feed drive. This should preferably be done in a separate manufacturer cycle, e.g. AXIS_C_ON.SPF.

```
_M1: Main spindle: Change to C axis mode
  AXIS_C_ON
GOTOF _MEND
```

Example for the manufacturer cycle AXIS_C_ON.SPF:

```
PROC AXIS_C_ON
IF $P_SEARCH==1 GOTOF _END
IF $P_SIM==1 GOTOF _END
; Do not execute cycle during block search and simulation
IF $A_IN[24]==1
;If feedback from the PLC, that the drive is already coupled,
;do not execute cycle;
; Release digital inputs and outputs via MD10350/MD10360
GOTOF _END
ENDIF
M[_F_S_NR[0]]=41  ;Select gearbox stage 1 for main spindle
SPOSA[[_F_S_NR[0]]=0  ;Position main spindle
WAITS(1)        ;wait until the spindle position is reached
STOPRE
G0 C0 ;C-axis positioning
STOPRE
M64             ;M-command for engaging the C-axis
MARK1:
IF $A_IN[24]==1 GOTOF MARK2
;Query whether the engaging has been completed
GOTOB MARK1 ;     Jump back to engaging completed
MARK2:
  _END:
IF $P_SIM==1          ;Simulation active
M[_F_S_NR[0]]=70   ;Switch main spindle into the axis mode
ENDIF
M17
```

_M2: Switching of the main spindle to the spindle mode:
1. No separate feed drive available:
The following entry is prepared :

```
IF($SP_MODE[_F_S_NR[0]]<>1
  ;N20 M[_F_S_NR[0]]=
ENDIF
```

The line N20 will be executed only if the main spindle is not in the spindle mode and can be supplemented by "5". By means of the command M5 the counter spindle is switched back to the spindle mode.

Notes:

Continued: Extract from the CUST_TECHCYC.SPF

2. Separate feed drive available:
As described for the previous mode a further manufacturer cycle should be used for disengaging the separate feed drive: AXIS_C_OFF.SPF and this is called up in the mode _M2.

```
_M2: ; Main spindle: Change to spindle mode
  AXIS_C_OFF
GOTOF _MEND
```

Example for the manufacturer cycle AXIS_C_OFF.SPF:

```
PROC AXIS_C_OFF
IF $P_SEARCH==1 GOTOF _END
IF $P_SIM==1 GOTOF _END
; Do not carry out cycle during block search and simulation
M65              ;M-command for disengaging C-axis
MARK1:
IF $A_IN[23]==1 GOTOF MARK2
;Query, whether the disengaging has been completed
GOTOB MARK1
MARK2:
ENDIF
_END:
M17
```

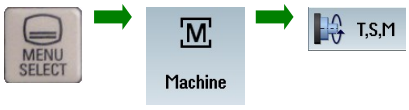
_M3: Main spindle: C axis clamping
The following entry is prepared:
;N30 M[_F_S_NR[0]]=
;N35 $AC_PRTIME_A_INC=1.5

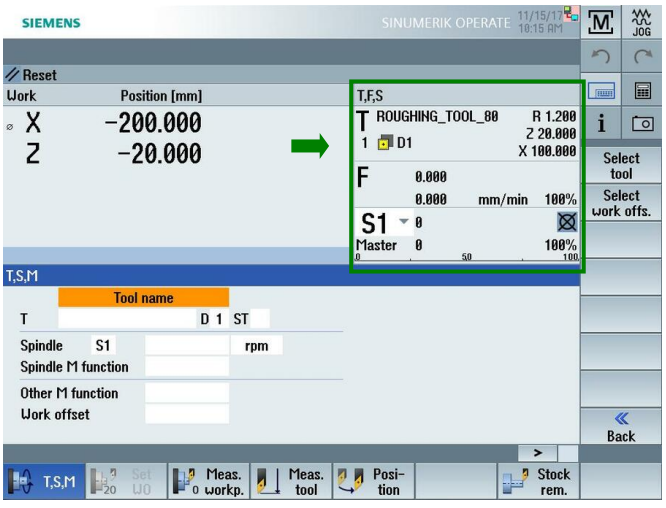
Enter the M-command for clamping the main spindle in the line N30. The line N35 is prepared for matching the machining time evaluated during the simulation to the real machining time (delay caused by the activation of the clamping must be taken into account).

_M4: Main spindle: Release C axis clamping
The following entry is prepared:
;N40 M[_F_S_NR[0]]=
;N45 $AC_PRTIME_A_INC=1.5

Enter the M-command for the clamping of the counter spindle in line N40. The line N45 is prepared for matching the machining time evaluated during the simulation to the real machining time (delay caused by the un-clamping must be taken into account).

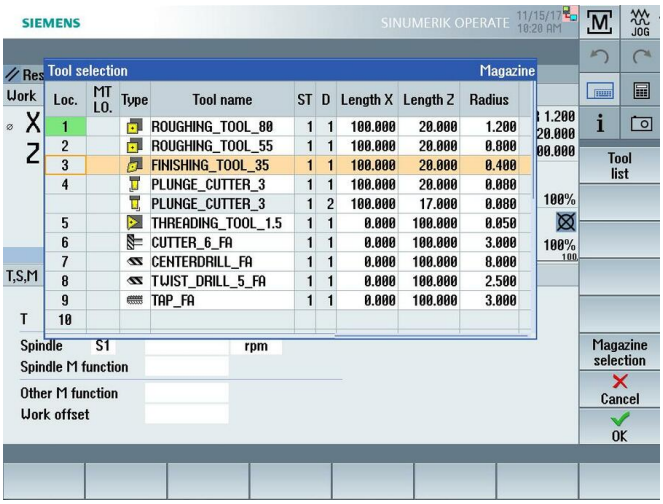
Monitoring the tool change

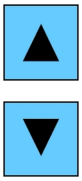




Shown in the T,F,S-field is the tool currently loaded in the spindle.





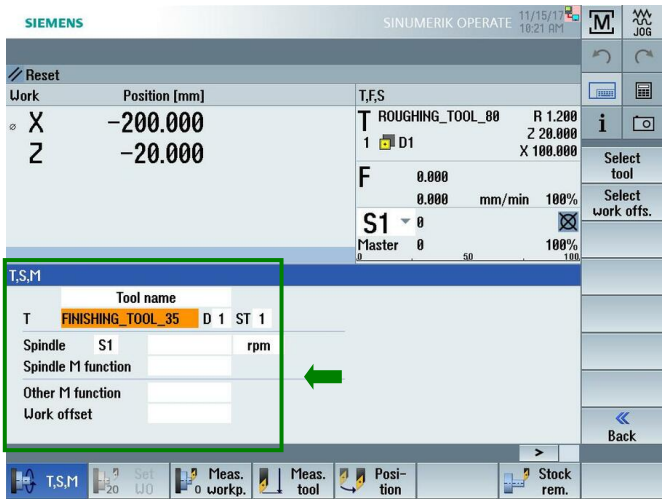


Use the cursor keys to select the required tool and thereafter transfer it by means of the Soft-Key "OK".

Notes:

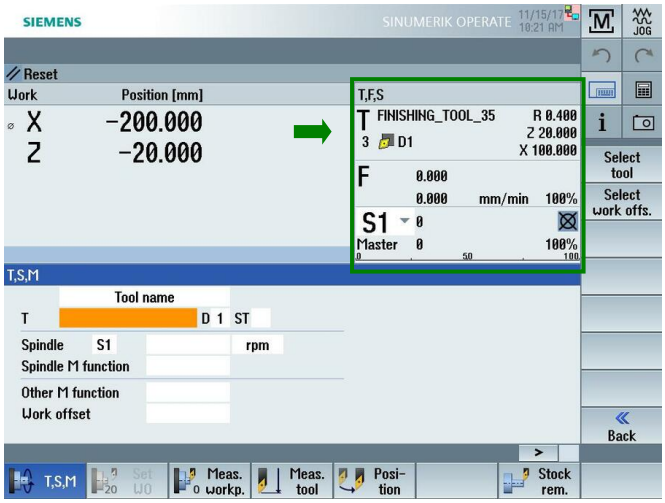
Loading the new tool





The selected tool is now displayed in the T,S,M mask.





The tool will be loaded into the T,F,S mask with Cycle Start. The PLC-tool change functions provided by the machine manufacturer are executed in the background for this process.

Gear step

Parameterize gear stage change

MD35010 $MA_GEAR_STEP_CHANGE_ENABLE	Parameterize gear stage change
	Change to: Bit 0 = 1

Description:

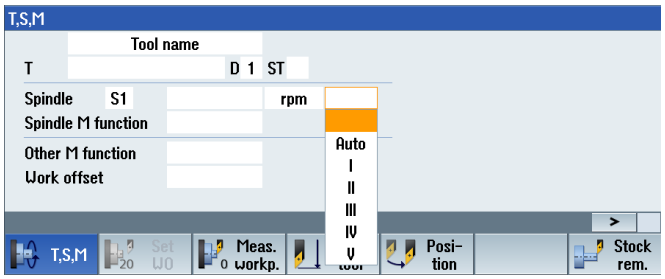
The selection field Gear stage appears only if in MD35010 for the spindle Bit 0 = 1 is set.

Number of gear stages

MD35090 $MA_NUM_GEAR_STEPS	Number of gear stages
Default: 5	Change to:

Description:

Number of prepared gear stages.



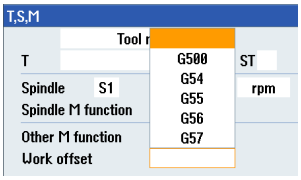
With NC-Start the program MA_JOG_STEP1 will be generated - in this example for the selection of the gear stage 2 of the main spindle M1=42.

```
NC/WKS/TEMP/MA_JOG_STEP1
PROC MA_JOG_STEP1 SBLOF¶
CYCLE210(0)¶
M1=42 ¶
M32¶
```

Notes:

T,S,M-mask: Zero point offset

In the T,S,M-mask to select a zero point offset by means of the parameter Zero point offset (G54,…) or it can be de-selected by means of the selection "Basic setting".



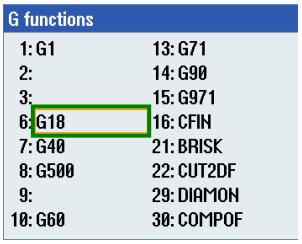
In order for the selection and de-selection of the Zero point offset remains activated also after reset/end of part-program, the setting of MD20152[7]=1 is required.

Machining level

The initial setting of the machining level of a turning machine is G18 as per MD20150[5] = 2.

Depending on the setting of MD20152[5] the switching of the plane remains operative after a reset/end-of-program.

MD20152[5]=0



Inch/Metric-switching

Below follows once more a summation of the machine data settings required for Inch/Metric-switching. The details to these MDs can be found in section "Basic settings".

Enable basic system conversion

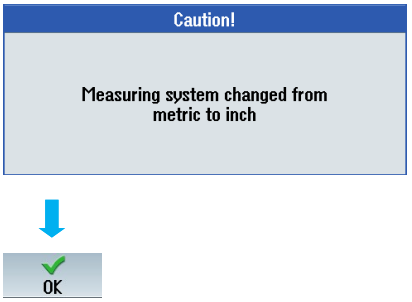
MD10260 $MN_CONVERT_SCALING_SYSTEM	Enable basic system conversion
	Change to: 1

Description:

Switching the measuring systems with HMI-Softkey
0. Hide Softkey inch/metric switching
1. Display Softkey inch/metric switching

Selection of Inch/Metric-switching:





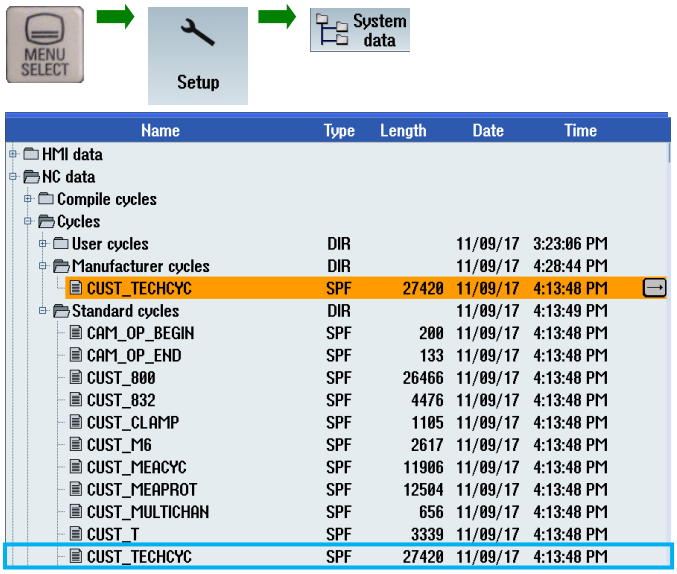
Note:

MD 20152[12] = 0 must be set for the position readouts to be converted.

Notes:

Manufacturer cycle CUST_TECHCYC.SPF

In the cycle CUST_TECHCYC.SPF function marks (_M1 to _M900) are prepared and documented. Match the cycle if necessary under manufacturer cycle.



Overview of the CUST_TECHCYC-jump accesses

Description:

This section provides an overview of the CUST_TECHCYC-jump accesses, including a note where the description can be found.

- **Main spindle:**
  - 1 - 2 Switching axis/spindle mode: Section JOG-TSM
  - 3 - 4 Clamping ON/OFF: Section JOG-TSM
  - 5 - 9 Chuck open/close, chuck flushing: Section Jaws, chuck, …
- **Tool spindle:**
  - 11 - 12 Switching axis/spindle mode: Section JOG-TSM
  - 13 - 14 Clamping ON/OFF: Section JOG-TSM
- **Counter spindle:**
  - 21 - 22 Switching axis/spindle mode: Module Sub spindle
  - 23 - 24 Clamping ON/OFF: Module Sub spindle
  - 25 - 29 Chuck open/close, chuck flushing Module Sub spindle

- 30: Positioning the 4th axis after block search: Section Block search
- 41 - 42: Engaging/disengaging a driven tool: Section JOG-TSM, CUST_TECHCYC
- 61 - 68: Switching turning/milling, mantle-front face C, mantle-front face Y, B-axis and block search: Section JOG-TSM, CUST_TECHCYC
- 100 - 103: Part receptacle: Section Jaws, chuck, ...
- 110 - 118: Tool change: Section JOG-TSM, CUST_TECHCYC
- 120 - 126: Engaging of main/counter spindle: Module Sub spindle
- 131 - 136: ShopTurn program-control: Section: JOG-TSM, CUST_TECHCYC
- 140 - 142: Block search: Section Block search

Note:

In order to clarify the call-ups of the CUST_TECHCYC it is possible to display the call-up mode by adding a message during commissioning:

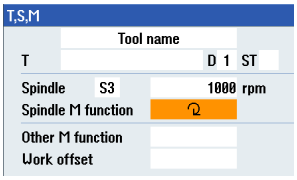
```
; MODE:
MSG ("CUST_TECHCYC-Mode = "<<_MODE)
M0
STOPRE
MSG()
```

Notes:

Tool spindle

After the selection of the tool spindle S3, input of the speed, selection of the direction of rotation and NC-Start the program shown alongside is generated.

The CUST_TECHCYC (42) is called up after tool spindle Stop.




```
NC/WKS/TEMP/MA_JOG_STEP1
PROC MA_JOG_STEP1 SBLOF¶
CYCLE210(0)¶
¶
SETMS(3)¶
¶
CUST_TECHCYC(14)¶
CUST_TECHCYC(41)¶
CUST_TECHCYC(12)¶
M3=3 G972 S3=1000 ¶
M32¶
```

Extract from the CUST_TECHCYC.SPF

```
NC/CMA.DIR/CUST_TECHCYC.SPF 264
_M11 ... _M14: Tool spindle
  _M11: Tool spindle: Change to C axis mode
    ;¶
  _M12: Tool spindle: Change to spindle mode
    _M12: ;¶
    IF($P_SMODE[_F_S_NR[1]]<>1); Spindle is not in spindle mode¶
      ;N120 M[_F_S_NR[1]]=          ; Change to spindle mode ¶
    ENDIF¶
    GOTOF _MEND¶
  Ende Block
    ;¶
  _M13: Tool spindle: C axis clamping
    ;¶
  _M14: Tool spindle: Release C axis clamping
    _M14: ;¶
    ;N140 M[_F_S_NR[1]]=          ; Release C axis clamping¶
    ;N145 $AC_PRTIME_A_INC=1.5    ; increase auxillary time by 1.5 s
econds¶
    GOTOF _MEND¶
  Ende Block
```

```
NC/CMA.DIR/CUST_TECHCYC.SPF 400
_M41 ... _M42: Driven tool
  _M41: Engage driven tool
    _M41: ;¶
    ;N410 M[_F_S_NR[1]]=          ; Engage driven tool¶
    GOTOF _MEND¶
  Ende Block
    ;¶
  _M42: Disengage driven tool
  Ende Block
```

Description:

_M11 - _M14
These jump accesses for the tool spindle relate to the jump accesses _M1 - _M4 for the main spindle and have already been described there. The entries differ only with respect to the spindle index in the variable _F_S_NR.

_M41: Engaging a driven tool
The following entry is prepared:
;N410 M[_F_S_NR[1]]=

The line N410 must be completed with the M-command for engaging the driven tool.

_M42: Disengaging a driven tool
The following entry is prepared:
N420 M[_F_S_NR[1]]=5 ; Stop spindle
;N425 M[_F_S_NR[1]]=

In block N420 the tool spindle is stopped. The line N425 must be completed with the M-command for disengaging the driven tool.

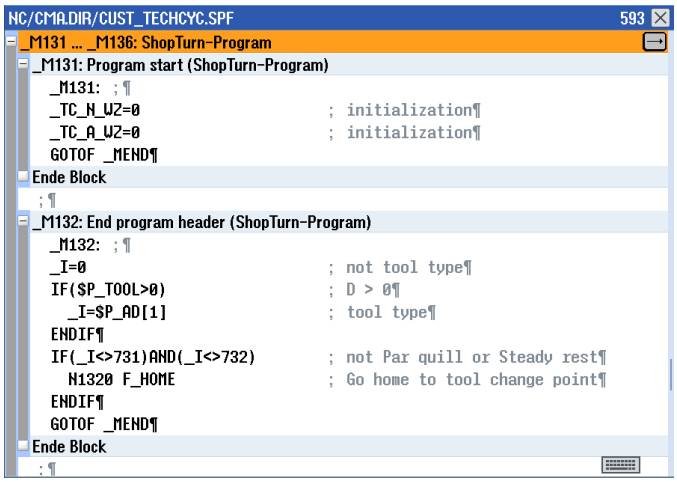
Note:

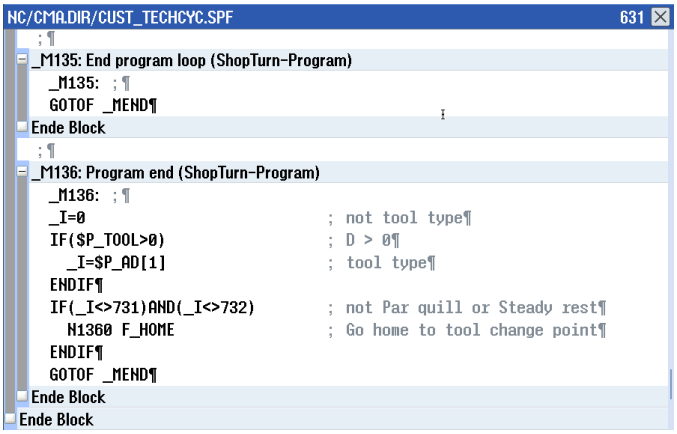
In the program sequence the CUST_TECHCYC call-ups are obtained from the ShopTurn cycles depending on the programmed technology - in case of driven tools by the boring and milling cycles. For this all shown jump accesses are possible..
The call-ups in mode 11 (switching the tool spindle into the C -axis mode) and mode 13 (tool spindle clamp) are not possible in the T,S,M-mask.

Notes:

Program control

Extract from the CUST_TECHCYC.SPF





Description:

The modes 131 - 136 will be called up only by a ShopTurn program.

_M131: At the beginning of a program
The following entry is prepared:
_TC_N_WZ=0
_TC_A_WZ=0

The variables _TC_N_WZ and _TC_A_WZ are meaningful only in conjunction with a B-axis.

_M132: at the end of the program header
The following entry is prepared:
_I=0
IF($P_TOOL>0)
_I=$P_AD[1]
ENDIF
IF(_I<>731)AND(_I<>732)
_N1320 F_HOME
ENDIF

Block N1320 calls up the ShopTurn-cycle F_HOME for the approach of the tool changing position defined in the program header.

_M135: at the end of a program loop of a ShopTurn program
No entry is prepared.

_M136: at the end-of-program of a ShopTurn program
The following entry is prepared:
_I=0
IF($P_TOOL>0)
_I=$P_AD[1]
ENDIF
IF(_I<>731)AND(_I<>732)
_N1360 F_HOME
ENDIF

Block N1360 calls up the ShopTurn-cycle F_HOME for the approach of the tool changing position defined in the program header.

Notes:

Block search

Parameterization for search run

MD11450 $MN_SEARCH_RUN_MODE	Parameterization for search run
	Change to: Bit 0 = 1 Bit 1 = 1 Bit 2 = 1 Bit 6 = 1

Description:

Bit 0: Response to output of alarm 10208

Bit 1: Automatic ASUP start after action blocks

Bit 2: Response to Auxf output (spindle)

Bit 6: Block search with simple, computing-time-optimized algorithms

Function mask Cross-technology

MD52212 $MCS_FUNCTION_MASK_TECH	Function mask Cross-technology
	Change to: Bit 3 = 1

Description:

Bit 3: Call block search cycle for ShopMill/ShopTurn

Note:

The setting of Bit 3=1 ensures that the tool programmed after the block search is duly changed and that the spindle rotates in the tool-assigned direction at the speed programmed in the destination block. For installations without ShopMill this Bit needs not to be set, however, the machine manufacturer must ensure that the tool and spindle functions are gathered and put out during the block search.

Tool change commands to PLC after search run

MD20128 $MC_COLLECT_TOOL_CHANGE	Tool change commands to PLC after search run
	Change to: 0

Description:

This MD is of any meaning only with activated magazine management (MD18080 $MN_MM_TOOL_MANAGEMENT_MASK, MD20310 $MC_TOOL_MANAGEMENT_MASK).
It specifies whether after block search with evaluation tool changing commands, tool preparation commands (generally tool changing commands) are put out to the PLC or not put out to the PLC.

Note:

This setting is necessary for ShopMill/ShopTurn.

1: Tool changing commands and tool preparation commands are gathered and passed on to the PLC with program Start when the search run has attained the destination block.

0: all tool and/or magazine specific commands that were gathered during the block search will not be put out to the PLC with the next program Start! That means that also programmed POSM, TCI, TCA will not be passed on to the PLC.

Note 1:

Without an active magazine management the tool change-M-code will not be gathered, if it is not assigned to an auxiliary function group. With an active magazine management this corresponds to an MD-value = 0.

Note 2:

The value = 0 is sensible for instance if after reaching the search destination the gathered tool change commands are transferred to the PLC in an ASUP-program with the aid of the commands GETSELT, GETEXET.

Notes:

Simulation

Adjustment of SZS Coordinate system

MD24030 $MC_FRAME_ACS_SET	Adjustment of SZS Coordinate system
	Change to: 1

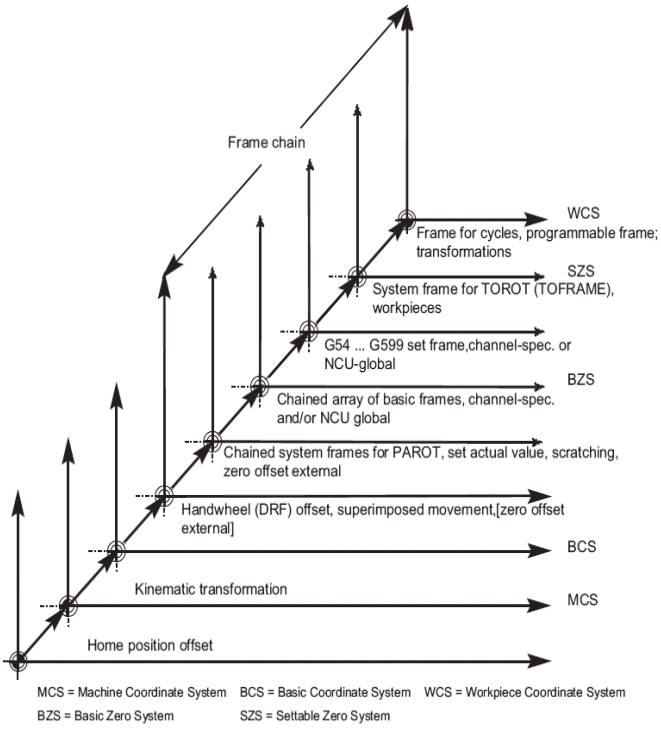
Description:

0: SZS results from the WCS transformed with $P_CYCFRAME and $P_PFRAME.

1: SZS results from the WCS transformed with $P_CYCFRAME.

The position readouts show the SZS-Coordinate system if "WCS" is selected.

Depiction of the Coordinate systems and the Frame chain

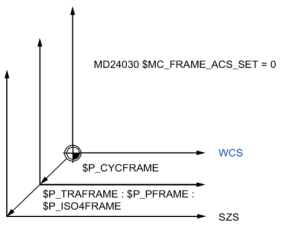


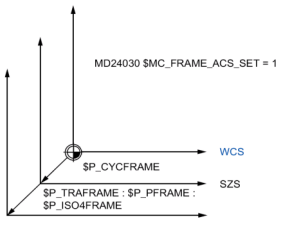
Description:

The SZS-Coordinate system is different to the WCS Coordinate system by

- Cycle frame
- Programmable frames (G54, …)
- Transformation (TRANS, ROT, SCALE)

With the setting of MD24030 $MC_FRAME_ACS_SET = 1 the programmable frames are taken into account in the position readouts.





Axis position at start of simulation

MD53230 $MAS_SIM_START_POSITION	Axis position at start of simulation
	Change to:

AX1: MX1
MD53230 = e.g. 400 mm

AX2: MZ1
MD53230 = e.g. 0 mm

Description:

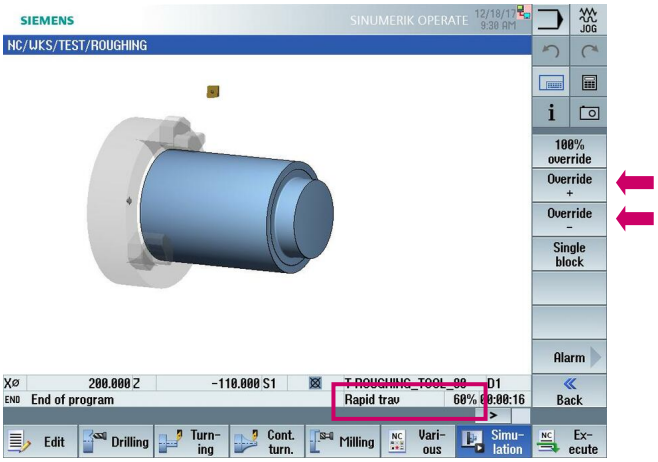
Axis position at the Start of the simulation.
The simulation is possible only, if for at least one geo-axis a value not equal to 0 has been set.

Note:

For the further axes and spindles this MD should be left on 0.

Notes:

Simulation depiction



Tipp:

To permit better observation of the simulation it is advisable to reduce the rapid traverse rate by means of the override switch. The percentage settings do not represent a linear function. A setting of about 60 - 80% are practicable.

Alternatively to the vertical Softkeys the override setting can also be obtained using the following keys:

Override increase: CTRL and ▲

Override reduction: CTRL and ▼

Override maximum: CTRL and M

Machining example

Machining example Main spindle

Example:

By means of a simple machining example with the main spindle the sequence is to be tested.

Tool list

Preparation:

The tools used in this example must have been edited and must have the correct cutting edge orientation.

Tool list — Magazine

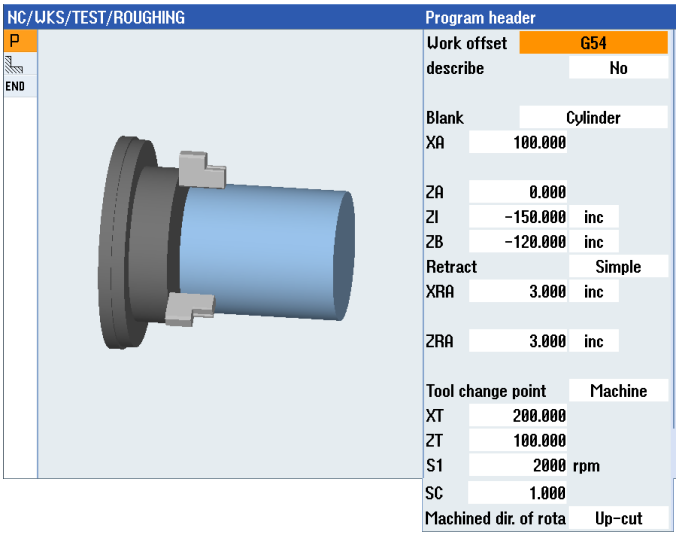
Loc.	MT LO.	Type	Tool name	ST	D	Length X	Length Z	Radius			
1			ROUGHING_TOOL_80	1	1	100.000	20.000	1.200	←	93.0	80
2											

Machining program: Roughing

NC/UKS/TEST/ROUGHING 1

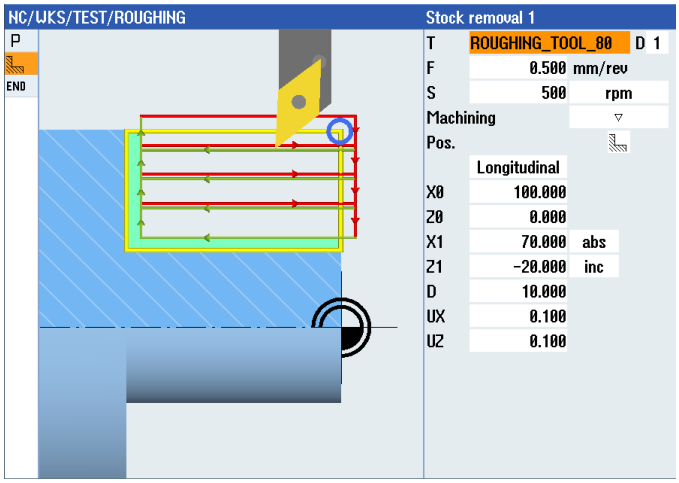
P	Program header	G54 Cylinder
	Stock removal	T=ROUGHING_TOOL_80 F=0.5/rev S=500rev
END	End of program	

Program header with workpiece definition:

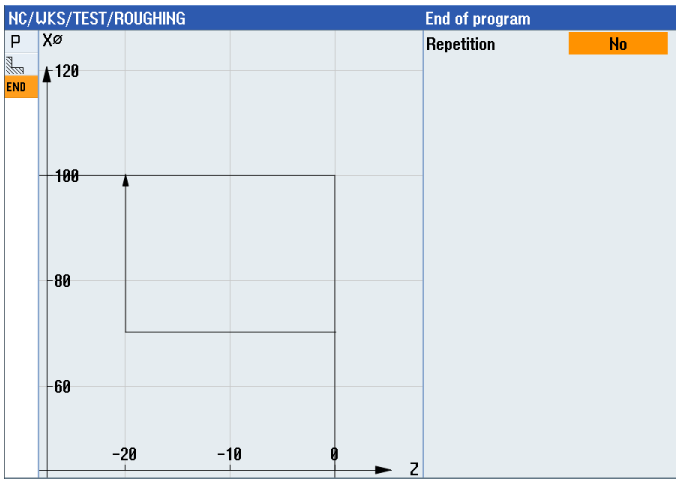


Notes:

Stock removal and technology data:



Program end and repetition:



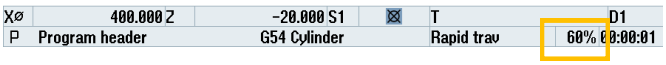
Test of the machining

A check of the machining sequences can be made having completed the inputs and selecting simulation.



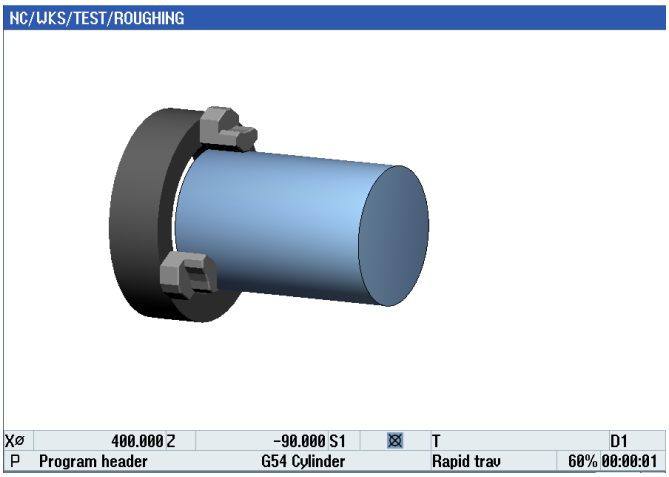
Alteration of the speed to about 60%.





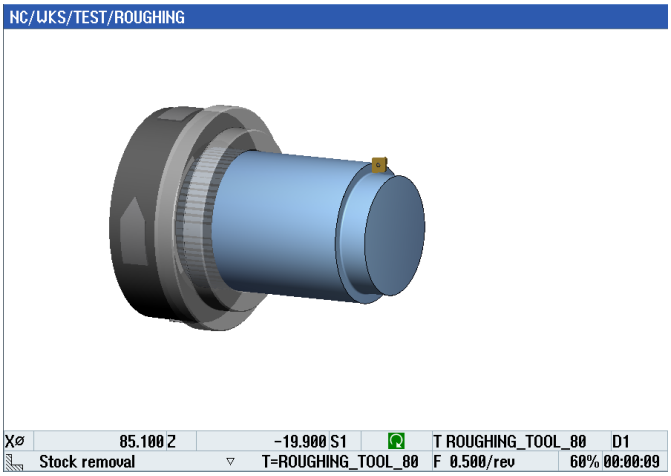
The machining progress is illustrated below in several stages.

Start, work piece in the main spindle

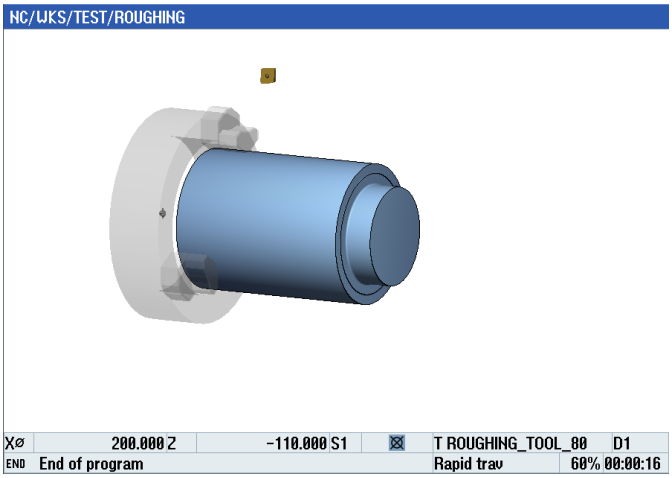


Notes:

Cut main spindle



End of the machining



TRANSMIT/TRACYL for the main spindle

General aspects

By means of the functions cylinder mantle transformation and front face machining both the mantle as well as the front face of the work piece can be machined.

The functions cylinder mantle transformation (TRACYL) and front face machining (TRANSMIT) are a Software-Option.

The functions must be set up additionally by means of the machine data.

Note:

There is a difference between the set-ups of TRANSMIT and TRACYL depending on whether a Y-axis is available or not. Described here is the set-up as applicable when no Y-axis is available.

There are 2 differing sets of data, both of which must be set up; - one for TRANSMIT and one for TRACYL:

- Front face machining main spindle: Transformation 1
  MD 24100 $MC_TRAFO_TYPE_1=256 (with C–axis)
- Cylinder mantle transformation main spindle: Transformation 3 without slot side face correction (with C–axis):
  MD 24300 $MC_TRAFO_TYPE_3=512

Notes:

Note:

With the function TRANSMIT/TRACYL a non-existent Y-axis is simulated by means of an interpolation between X/C (in case of TRANSMIT) respectively Z/C (in case of TRACYL). In this configuration example there is, apart from the main spindle, no separate C-axis as feed drive for the C-axis mode. Therefore, the channel index of the main spindle must be entered for the C-axis into the configuration data for TRANSMIT/TRACYL.

As a reminder; this is how the machine is already configured:

	Index
20080[0] $MC_AXCONF_CHANAX_NAME_TAB = X1	1
20080[1] $MC_AXCONF_CHANAX_NAME_TAB = Z1	2
20080[2] $MC_AXCONF_CHANAX_NAME_TAB = C	3
20080[3] $MC_AXCONF_CHANAX_NAME_TAB = CS3	4
20080[4] $MC_AXCONF_CHANAX_NAME_TAB = Q	5

These MDs were already set up and described.

MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 1
MD24040 $MC_FRAME_ADAPT_MODE = 7H

The data blocks for TRANSMIT and TRACYL described below are conceived for the configured horizontal lathe with machining behind the turning centre and with the axis configuration as shown above. Nevertheless such requirements as the direction of rotation of the C-axis in case of the transformation or the input of an offset might require an adaptation this set of data.

The tables on the next pages gives an overview of the data blocks for TRANSMIT and TRACYL.

Front face machining (TRANSMIT)

- Front face machining main spindle: Transformation 1 (with C–axis)

MD	Identifier	Settings	Comment	Description
24100	$MC_TRAFO_TYPE_1	256		Definition of the 1st transformation in the channel
24110	$MC_TRAFO_TYPE_IN_1	[0]=1 (X) [1]=3 (C) [2]=2 (Z) [3]=0 [4]=0		Axis assignment for transformation 1
24120	$MC_TRAFO_GEOAX_ASSIGN_TAB_1	[0]=1 (X) [1]=3 (C) [2]=2 (Z)		Assignment of geometry axes to channel axes for transformation 1
24900	$MC_TRANSMIT_ROT_AX_OFFSET_1	0		Offset of rotary axis for the 1st TRANSMIT transformation
24905	$MC_TRANSMIT_ROT_AX_FRAME_1	1		Rotary axis offset TRANSMIT 1
24910	$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1	0		Sign of rotary axis for 1st TRANSMIT transformation
24911	$MC_TRANSMIT_POLE_SIDE_FIX_1	1		Restriction of working range in front of / behind the pole, 1. TRANSMIT
24920	$MC_TRANSMIT_BASE_TOOL_1	0		Vector of base tool for 1st TRANSMIT transformation

Note: After all parameters have been entered a HMI reboot is required for the simulation.

Notes:

Front face machining with TRANSMIT

Offset of rotary axis for the 1st TRANSMIT transformation

MD24900 $MC_TRANSMIT_ROT_AX_OFFSET_1	Offset of rotary axis for the 1st TRANSMIT transformation
Default: 0 Grad	Change to:

Description:

Indicates the offset of the rotary axis for the first agreed TRANSMIT transformation in degrees in relation to the neutral position while TRANSMIT is active.

The rotary position of the XY-plane of the Cartesian Coordinate system in relation to the defined zero position of the rotary axis C can be entered here.

Rotary axis offset TRANSMIT 1

MD24905 $MC_TRANSMIT_ROT_AX_FRAME_1	Rotary axis offset TRANSMIT 1
	Change to: 1

Description:

0: the axial offset of the rotary axis is not taken into account.

1: the axial offset of the rotary axis will be taken into account.

2: the axial offset of the rotary axis will be taken into account up to the SZS.

If the work piece is to be rotated during machining with TRANSMIT, then this cannot be achieved by programming the rotary axis in the channel, since this is occupied by a geometry axis (Y). An offset of the rotary axis in the transformation frame (TRANS, ROT, SCALE) will be taken into account during TRANSMIT if the setting is = 1.

Corresponds to:
MD24030 $MC_FRAME_ACS_SET

Sign of rotary axis for 1st TRANSMIT transformation

MD24910 $MC_TRANSMIT_ROT_SIGN_IS_PLUS_1	Sign of rotary axis for 1st TRANSMIT transformation
	Change to: 0

Description:

Indicates the sign with which the rotary axis is taken into account in the TRANSMIT transformation for the first agreed TRANSMIT transformation for each channel.

Corresponds to:
MD24960 $MC_TRANSMIT_ROT_SIGN_IS_PLUS_2

Note:

The requirements at the machine such as direction of rotation of the spindle/C-axis might necessitate different settings to the ones shown here.

Restriction of working range in front of / behind the pole, 1st TRANSMIT

MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1	Restriction of working range in front of / behind the pole, 1st TRANSMIT
	Change to: 1

Description:

Restriction of the working range in front of/behind the pole or no restriction, i.e. travelling through the pole.

The assigned values have the following meaning:

0: No restriction of the working range. Travelling through the pole.

1: Working range of the linear axis for positions >= 0,
(if the tool length correction parallel to the linear axis = 0)
If on the turning machine travelling in the X-axis beyond the turning centre (-X) is not possible.

2: Working range of the linear axis for positions <=0,
(if the tool length correction parallel to the linear axis = 0)

Notes:

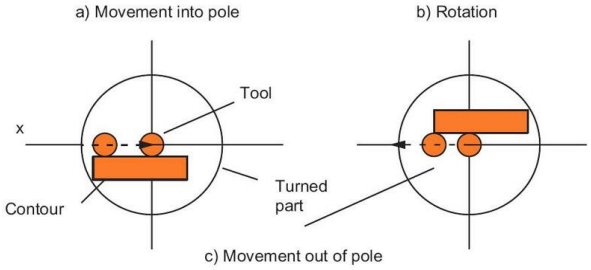
Continuation of Restriction of the working range in front of / behind the pole, 1st TRANSMIT

The sketch below shows the behaviour with the setting of MD24911 = 1.

This setting is usual, since normally no or very little traverse is available in the X-axis beyond the turning centre.

Machining takes place in the following steps:

- Traverse along the X-axis to the pole
- Rotation of the rotary axis by 180 degrees. During this motion the X-axis remains stationary
- Machining of the remainder of the block. The X-axis motion is now away from the centre in the positive direction.



Vector of base tool for 1st TRANSMIT transformation

MD24920[x] $MC_TRANSMIT_BASE_TOOL_1	Vector of base tool for 1st TRANSMIT transformation
Default: 0, 0, 0 mm	Change to:

MD24920[0] =0
MD24920[1] =0
MD24920[2] =0

Description:

This states a basic offset of the tool zero point for the 1st TRANSMIT-transformation. Only if there is an offset from the turning centre will this MD be required. For the standard machining the default applies.

The index i takes on the values 0, 1, 2 for the 1st to the 3rd geometry axis.

Corresponds to:
MD24970 $MC_TRANSMIT_BASE_TOOL_2

Creating a machining program for front face machining with the main spindle.

Tool list

Tool list — Magazine

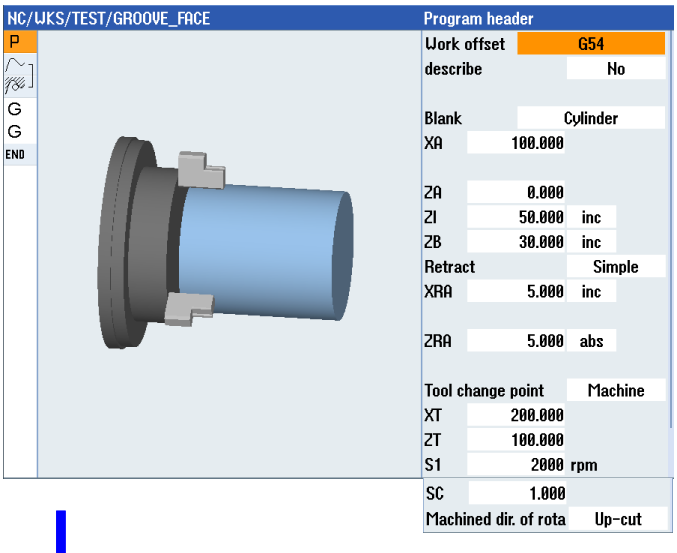
Loc.	MT LO.	Type	Tool name	ST	D	Length X	Length Z	Radius			
1			ROUGHING_TOOL_80	1	1	100.000	20.000	1.200	←	93.0	80
2			CUTTER_6_FA	1	1	0.000	100.000	6.000	2		
3											

Machining program

NC/WKS/TEST/GROOVE_FACE 1

P	Program header	G54 Cylinder
	Contour	FACE_GROOVE
	Path milling	T=CUTTER_6_FA F=50/min S=2000rev Z0=0 Z1=5inc
G	TRAFOOF¶	
G	G0 C0¶	
END	End of program	

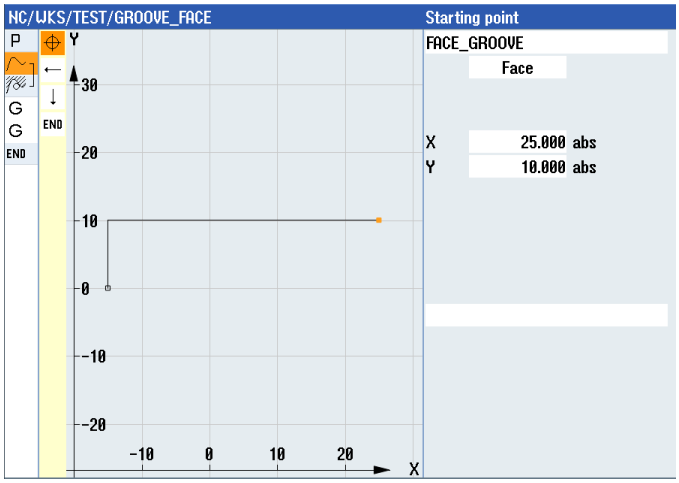
Program header with workpiece definition:



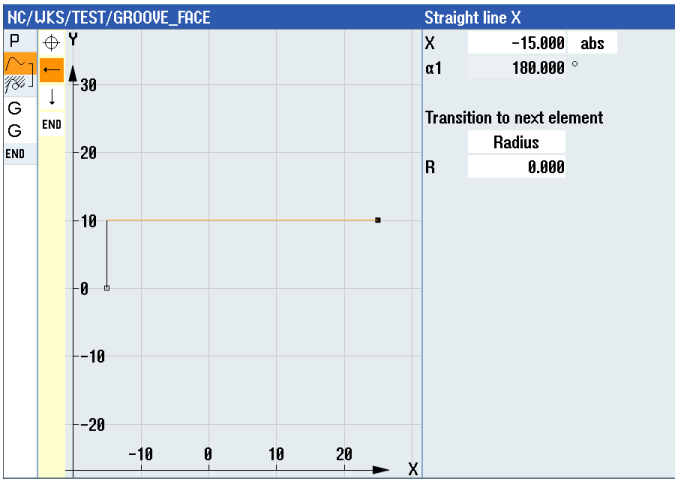
Notes:

Create contour description:

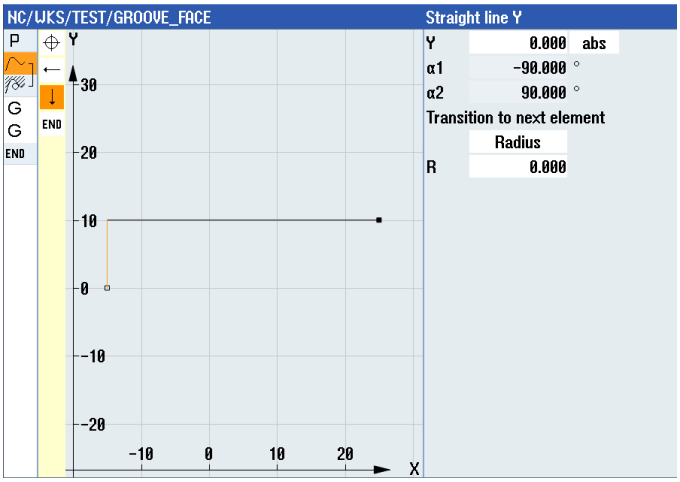
- Front face, Starting point X=25 abs, Y =10 abs



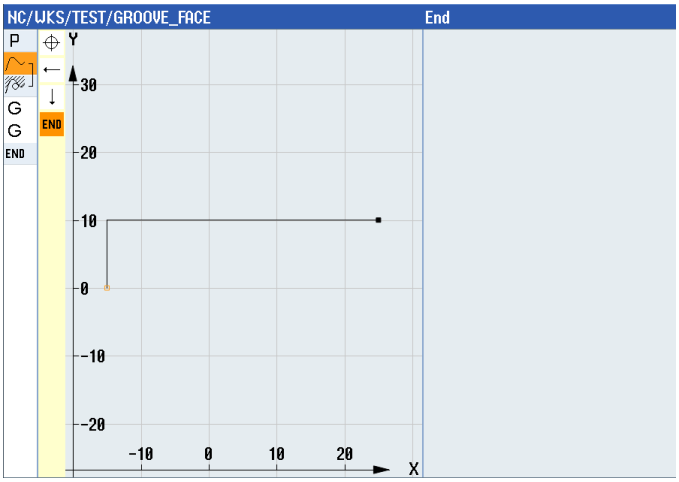
- Straight X=-15 abs



- Straight Y=0 abs

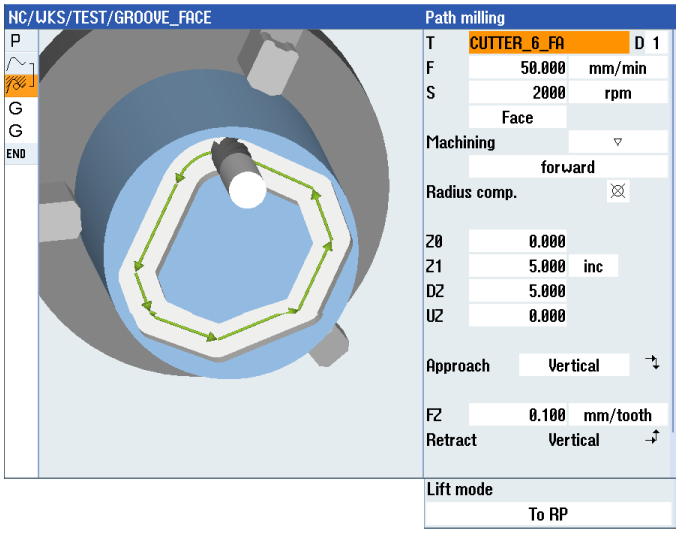


- End point

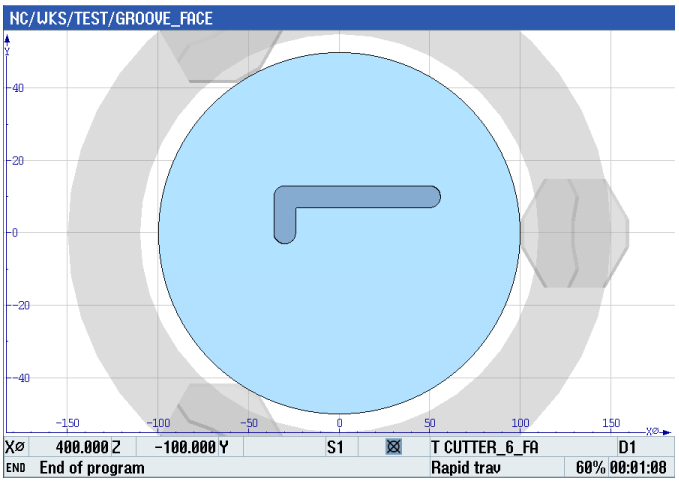


Notes:

Continuous path milling on front face and technology data:



Starting the simulation:



Note:

In order for the depiction of the simulation to correspond to the depiction of the contour programming, the transformation has been deactivated (TRAFOOF) following the continuous path milling and the C-axis positioned into the basic position (G0 C0). Otherwise the depiction would be shown in the last C-axis position to be approached =180 degrees; in other words rotated by 180 degrees.

Cylinder mantle transformation (TRACYL)

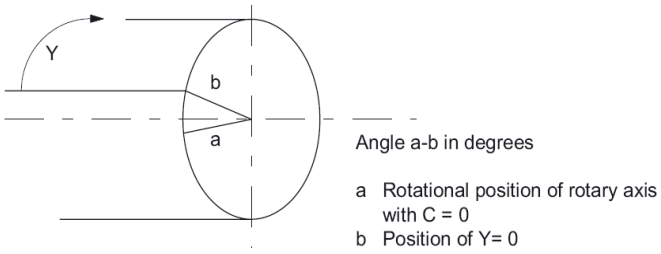
- Cylinder mantle transformation main spindle: Transformation 3 (without slot side face correction with C–axis)

MD	Identifier	Set-tings	Comment	Description
24300	$MC_TRAFO_TYPE_3	512		Definition of the 3rd transformation in the channel
24310	$MC_TRAFO_TYPE_IN_3	[0]=1 (X) [1]=3 (C) [2]=2 (Z) [3]=0 [4]=0		Axis assignment for transformation 3
24320	$MC_TRAFO_GEOAX_ASSIGN_TAB_3	[0]=1 (X) [1]=3 (C) [2]=2 (Z)		Assignment of geometry axes to channel axes for transformation 3
24800	$MC_TRACYL_ROT_AX_OFFSET_1	0		Offset of rotary axis for the 1st TRACYL transformation
24805	$MC_TRACYL_ROT_AX_FRAME_1	1		Rotary axis offset TRACYL 1
24810	$MC_TRACYL_ROT_SIGN_IS_PLUS_1	1		Sign of rotary axis for 1st TRACYL transformation
24820	$MC_TRACYL_BASE_TOOL_1	0		Vector of base tool for 1st TRACYL transformation

Note: After all parameters have been entered a HMI reboot is required for the simulation.

Notes:

Mantle machining with TRACYL (without slot side face correction)



Offset of rotary axis for the 1st TRACYL transformation

MD24800 $MC_TRACYL_ROT_AX_ OFFSET_1	Offset of rotary axis for the 1st TRACYL transformation
Default: 0 Grad	Change to:

Description:

Indicates the offset of the rotary axis for the first agreed TRACYL transformation in degrees in relation to the neutral position while TRACYL is active.

The rotary position of the mantle plane in relation to the rotary axis C can be entered here.

Rotary axis offset TRACYL 1

MD24805 $MC_TRACYL_ROT_AX_ FRAME_1	Rotary axis offset TRACYL 1
	Change to: 1

Description:

0: the axial offset of the rotary axis will not be taken into account.

1: the axial offset of the rotary axis will be taken into account.

2: the axial offset of the rotary axis will be taken into account up to the SZS.

If the work piece is to be rotated during machining with TRANSMIT, then this cannot be achieved by programming the rotary axis in the channel, since this is occupied by a geometry axis (Y). An offset of the rotary axis in the transformation frame (TRANS, ROT, SCALE) will be taken into account during TRANSMIT if the setting is = 1.

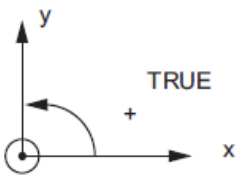
Corresponds to:
MD24030 $MC_FRAME_ACS_SET

Sign of rotary axis for 1st TRACYL transformation

MD24810 $MC_TRACYL_ROT_SIGN_ IS_PLUS_1	Sign of rotary axis for 1st TRACYL transformation
	Change to: 1

Description:

Indicates the sign with which the rotary axis is taken into account in the TRACYL transformation for the first agreed TRACYL transformation.



Corresponds to:
MD24860

$MC_TRACYL_ROT_SIGN_IS_PLUS_2

Notes:

Vector of base tool for 1st TRACYL transformation

MD24820[x] $MC_TRACYL_BASE_ TOOL_1	Vector of base tool for 1st TRACYL transformation
Default: 0, 0, 0 mm	Change to:

MD24820[0] =0
MD24820[1] =0
MD24820[2] =0

Description:

Indicates a basic offset of the tools zero for the 1st TRACYL transformation. The offset is referenced to the geometry axes valid when TRACYL is active. The basic offset is included with and without selection of the tool length compensation. Programmed length corrections have an additive effect with respect to the basic tool.
The index i takes on the values 0, 1, 2 for the 1st to 3rd geometry axes.

Corresponds to:
MD24870 $MC_TRACYL_BASE_TOOL_2

Creating a machining program for mantle machining with the main spindle.

Tool list

Tool list — Magazine

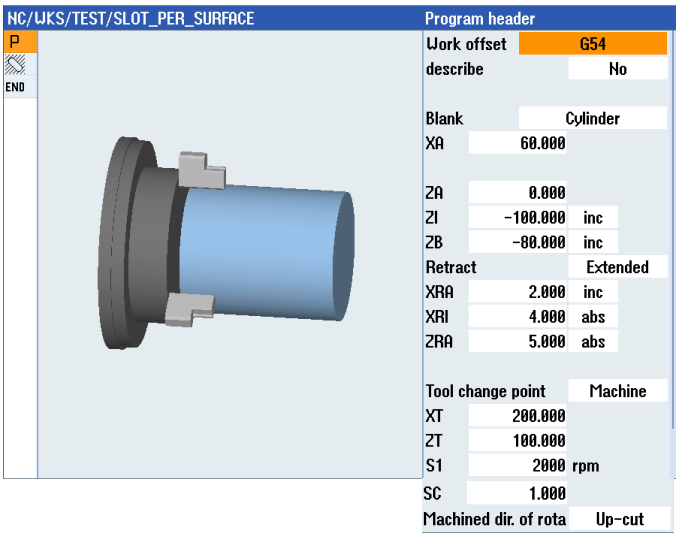
Loc.	MT LO.	Type	Tool name	ST	D	Length X	Length Z	Radius			
1			ROUGHING_TOOL_80	1	1	100.000	20.000	1.200	←	93.0	80
2			CUTTER_6_FA	1	1	0.000	100.000	6.000	2		
3			CUTTER_6_MA	1	1	0.000	100.000	6.000	2		
4											

Machining program

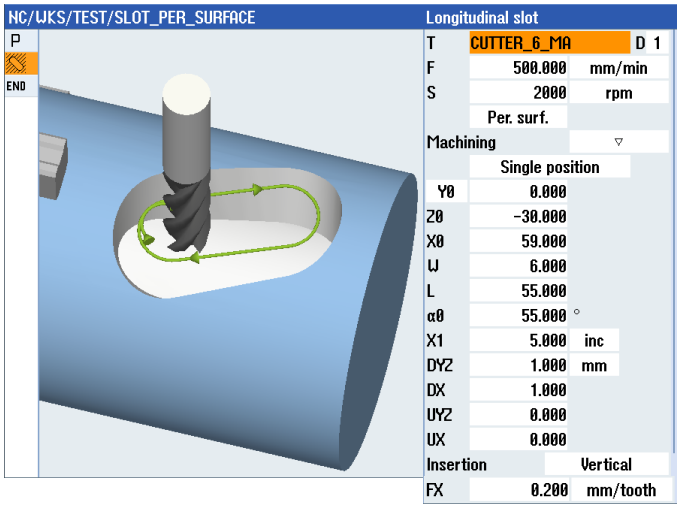
NC/WKS/TEST/SLOT_PER_SURFACE 1

P	Program header	G54 Cylinder
	Longitudinal slot	T=CUTTER_6_MA F=500/min S=2000rev Y0=0 Z0=-30
END	End of program	

Program header with workpiece definition:



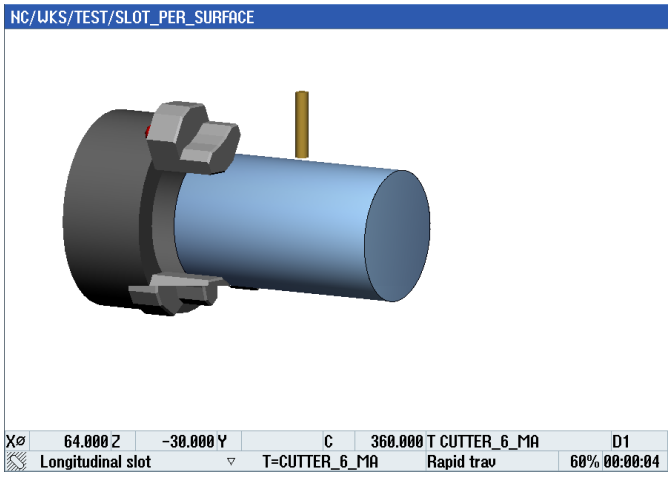
Longitudinal slot on mantle and technology data:

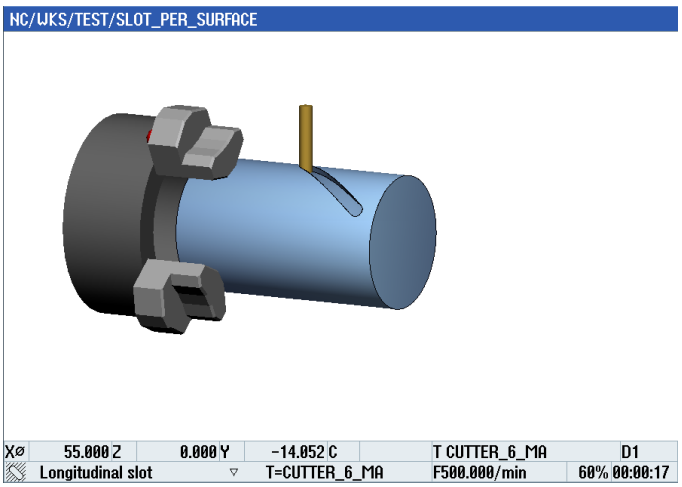


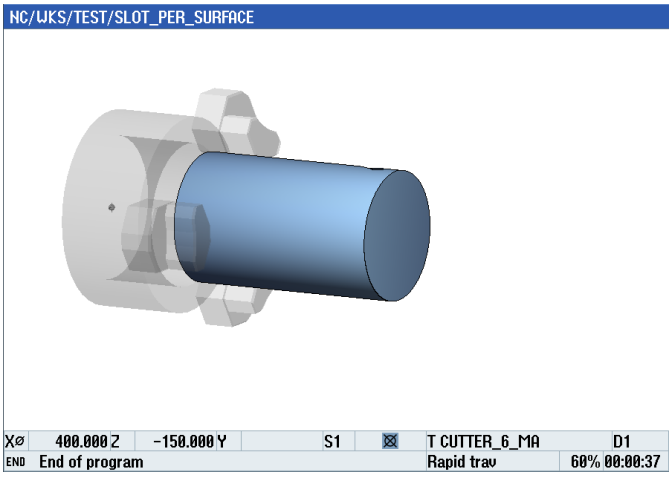
Notes:

Testing the machining









Chuck, jaws, part receptacle and tailstock

This section describes

- the functions for opening closing and flushing of the chuck for the main spindle
- The setting and dimensions of the jaws for the main spindle
- The settings for part receptacle and
- The settings of the tailstock

Chuck

Machine data for the definition of M-commands

Notes:

M code for open chuck with non-rotating spindle

MD52250[x] $MCS_M_CODE_CHUCK_OPEN	M code for Open chuck with non-rotating spindle
Default: -	Change to:

Description:

e.g.: "M34" or "M1=34"

Elements:
MD52250[0] = Main spindle
MD52250[1] = Counter spindle

M code for open chuck with rotating spindle

MD52251[x] $MCS_M_CODE_CHUCK_OPEN_ROT	M code for Open chuck with rotating spindle
Default: -	Change to:

Description:

e.g.: "M35" or "M1=35"

Elements:
MD52251[0] = Main spindle
MD52251[1] = Counter spindle

M-code for Close chuck

MD52252[x] $MCS_M_CODE_CHUCK_CLOSE	M- code for Close chuck
Default: -	Change to:

Description:

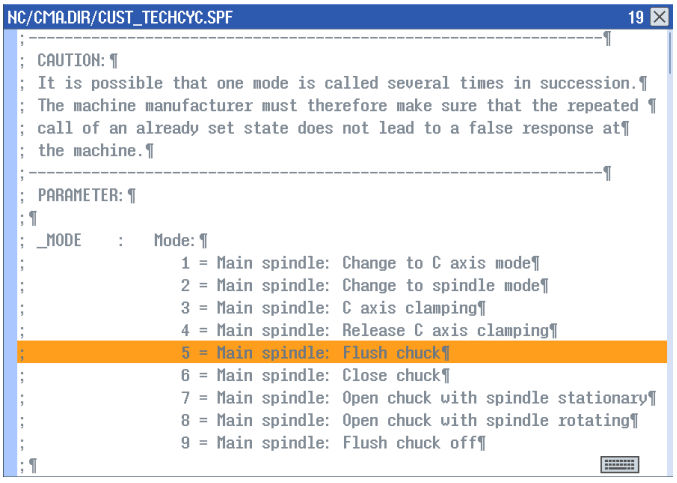
e.g.: "M36" or "M1=36"

Elements:
MD52251[0] = Main spindle
MD52251[1] = counter spindle

Chuck flushing, close, open

For the control of the chuck functions there are jump exits from the ShopTurn cycles into the manufacturer-cycle CUST_TECHCYC.SPF.

Extract from the CUST_TECHCYC.SPF



Notes:

Continuation of Chuck flushing, close, open

```
NC/CMA.DIR/CUST_TECHCYC.SPF                                    223
_M5: Main spindle: Flush chuck
  _M5:  ;¶
  GOTOF _MEND¶
Ende Block
  ;¶
_M6: Main spindle: Close chuck
  _M6:  ;¶
  IF(ISNUMBER($MCS_M_CODE_CHUCK_CLOSE[0]))¶
    N60 EXECSTRING("M"<<$MCS_M_CODE_CHUCK_CLOSE[0])¶
  ELSE¶
    N65 EXECSTRING($MCS_M_CODE_CHUCK_CLOSE[0])¶
  ENDIF¶
  GOTOF _MEND¶
Ende Block
  ;¶
_M7: Main spindle: Open chuck with spindle stationary
  ;¶
_M8: Main spindle: Open chuck with spindle rotating
  ;¶
_M9: Main spindle: Flush chuck off
```

Note:

The definition of the M-commands for opening and closing the chuck of the main and counter spindle does not occur in CUST_TECHCYC.SPF but by way of machine data. This is necessary in order for the opening and closing of the chuck and the associated work piece movement to be shown correctly in the simulation also with G-programs. The M-command for flushing the chuck occurs directly in Mode 5 since this is not necessary for the simulation.

Description:

The machine data are evaluated in the modes _M6, _M7 and _M8. If the machine data contain a numerical value only, then this will be put out as M-function without an extension. If the machine datum contains an auxiliary function with an extension, e.g. M1=34, then this will be executed out as a string.

Jaws for the main spindle

Spindle chuck data

MD53240[x] $MAS_SPINDLE_ PARAMETER	Spindle chuck data
Default: 0, 0, 0 mm	Change to:

Description:

for AX3: MCS1

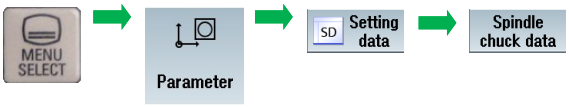
MD53240[0] =0 → e.g. 40 mm
Chuck dimensions = ZC1 in the mask Spindle chuck data

MD53240[1] =0 → e.g. 30 mm
Stop dimensions = ZS1 in the mask Spindle chuck data

MD53240[2] =0 → e.g. 22 mm
Jaws dimensions = ZE2 in the mask Spindle chuck data

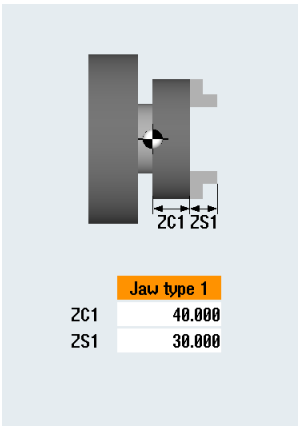
In case of the main spindle the selection for the type of jaws is not available. This machine datum 53240[2] is meaningless.

The dimensions depend on the used jaws and can be changed by the machine operator.



ZC1: The parameter ZC1 is used for manual measuring of tools if the chuck of the main spindle is used as reference. Furthermore for the simulation of the faceplate.

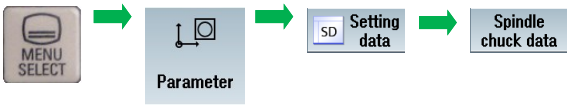
ZS1: The parameter ZS1 is only used in the simulation for the depiction of the jaws.

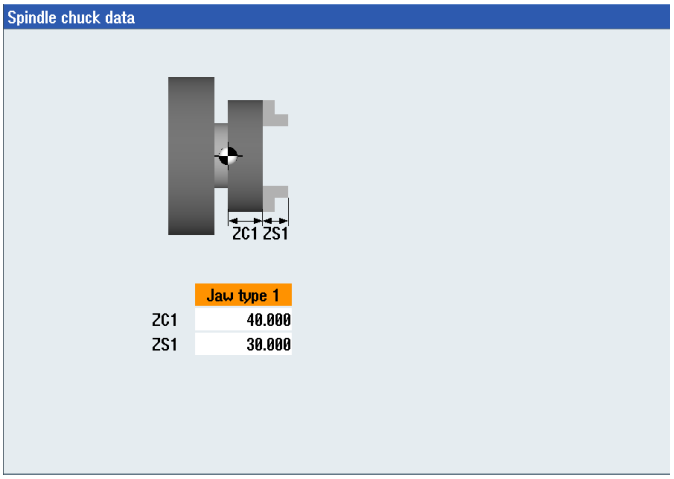


The inputs for the parameters "ZC1" for chuck dimensions and "ZS1" for the dead stop dimensions are written into the machine data MD53240[0] and MD53240[1] of the data block of the C-axis. These entries are then transferred to the data block of the spindle if there exists a separate C-axis as feed drive for the C-axis mode apart from the main spindle.

Notes:

Check of the entries made in the window "Spindle chuck data"





The entries for the parameters "ZC1" chuck dimension and "ZS1 stop dimension for the counter spindle are identical in MD53240.

Note:

The input field ZE1 for the jaw dimensions be shown only if the jaw type 2 has been selected.

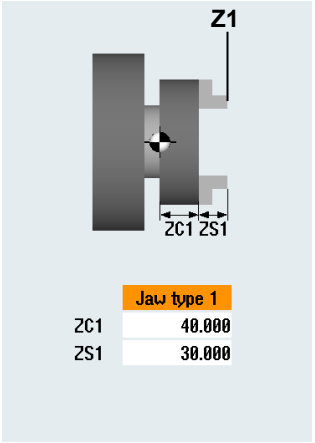
Spindle chuck data

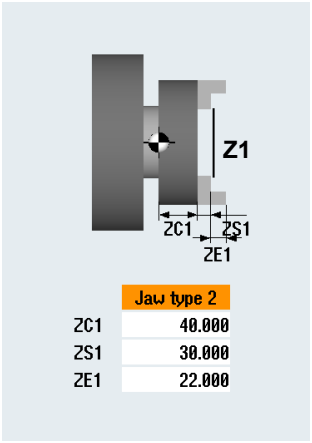
MD53241 $MAS_SPINDLE_CHUCK_TYPE	Spindle jaw type
Default: 0	Change to:

Description:

for AX3: MCS1

0 = Jaw type 1 = Clamping from outside
1 = Jaw type 2 = Clamping from inside





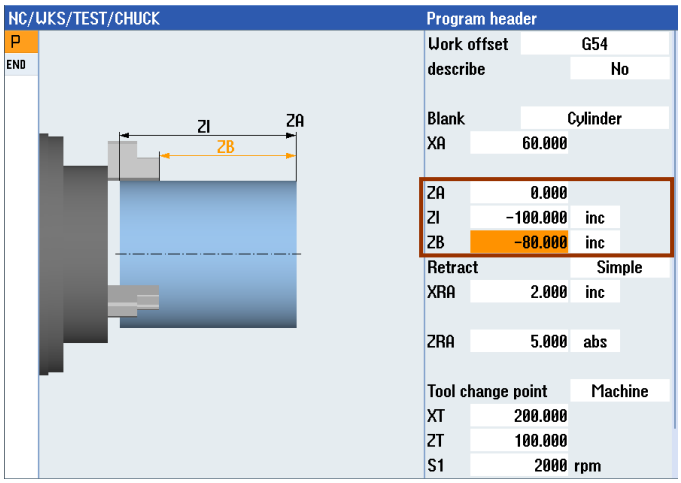
Jaw type 1:

Jaw type 1: Z1 lies in the WKS on the front edge of the clamping jaws; so if for instance Z1 = -10, the work piece will be inserted by 10 mm.

Jaw type 2:

Jaw type 2: Z1 lies on the inner jaw step; so for Z1 = 0, the work piece will be inserted right up to the jaw step.

Notes:



Example:

In case of a work piece ZI = 150mm long, ZB is theoretically 120mm, since 30mm were entered in ZS1 as the dead stop dimension. For safety reasons, however, a further allowance must be made to ensure that the tools do not damage the surface of the chuck jaws.

Part receptacle

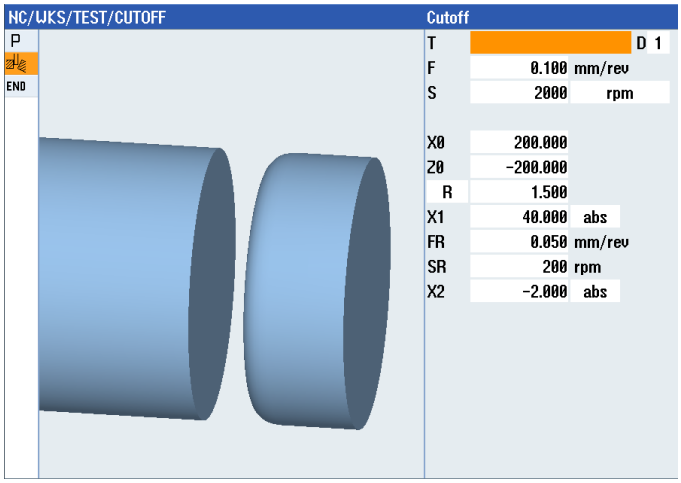
Function mask Turning

MD52218 $MCS_FUNKTION_MASK_TURN	Function mask Turning
	Change to: Bit 1 = 1

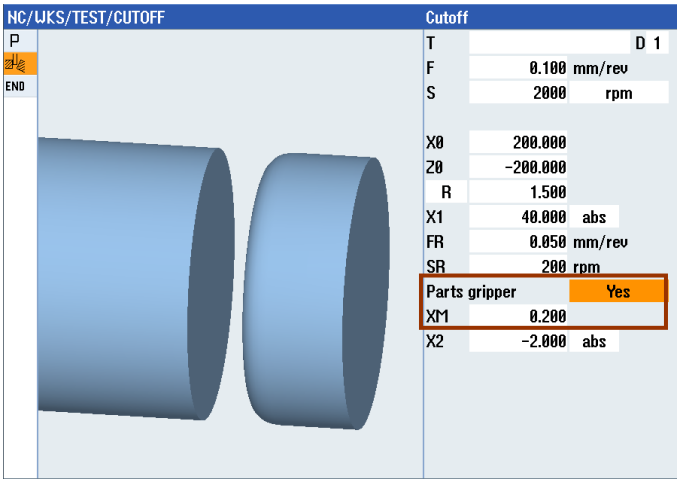
Description:

Bit 1: Release of the part receptacle

MD52218 Bit 1 = 0



MD52218 Bit 1 = 1



Notes:

Continuation of Part receptacle

Extract from the CUST_TECHCYC.SPF

NC/CMA.DIR/CUST_TECHCYC.SPF 54

```
;                30 = 4th axes after block search (e.g. sub spindle line
ar axis, tailstock, steady rest)¶
;¶
;                41 = Engage driven tool¶
;                42 = Disengage driven tool¶
;¶
;                61 = Change to turning¶
;                62 = Change to peripheral surface machining C¶
;                63 = Change to face machining C¶
;                64 = Change to peripheral surface machining Y¶
;                65 = Change to face machining Y¶
;                66 = Change to face machining B (swiveling)¶
;                67 = Deselect face machining B (swiveling)¶
;                68 = After block search on ST-block before swiveling¶
;¶
;               100 = Position receptacle before cut-off¶
;               101 = Open receptacle for cut-off¶
;               102 = Close receptacle after cut-off¶
;               103 = End of cut-off¶
;¶
```

NC/CMA.DIR/CUST_TECHCYC.SPF 458

```
_M100 ... _M103: Cut-off
_M100: Position receptacle before cut-off
  _M100:  ;¶
  GOTOF _MEND¶
Ende Block
  ;¶
_M101: Open receptacle for cut-off
  _M101:  ;¶
  GOTOF _MEND¶
Ende Block
  ;¶
_M102: Close receptacle after cut-off
  _M102:  ;¶
  GOTOF _MEND¶
Ende Block
  ;¶
_M103: End of cut-off
  _M103:  ;¶
  GOTOF _MEND¶
Ende Block
```

Description:

No entries are prepared in the CUST_TECHCYC-jump accesses for the part receptacle.

_M100: This mode is used only in a ShopTurn-program, The call-up occurs when the tool in the X-axis is on the retraction level at the height of the cut-off position.

_M101: This mode is used both in ShopTurn programs as also in a programGuide program. It is called up when during the cut-off the position XM defined in the input mask for the extension of the part receptacle has been reached.

_M102: This mode is used both in ShopTurn programs as also in a programGuide program. It is called up when the destination depth of the cut-off is reached; it can be used for swivelling the part receptacle into position.

Program mask Cutting-off cycle

ShopTurn program:

Cutoff

T	D 1	
F	0.100	mm/rev
S	2000	rpm
X0	200.000	
Z0	-200.000	
R	1.500	
X1	40.000	abs
FR	0.050	mm/rev
SR	200	rpm
Parts gripper	Yes	
XM	0.200	
X2	-2.000	abs

programGuide program:

Cutoff

SC	1.000	
F	0.100	
DIR		
S	2000	rpm
X0	200.000	
Z0	-200.000	
R	1.500	
X1	40.000	abs
FR	0.050	Dis./rev
SR	200	rpm
Parts gripper	Yes	
XM	0.200	
X2	-2.000	abs

Description:

The parameter ”Parts gripper” appears in the input masks with the release of the part receptacle in MD52218 Bit 1 = 1. If this is set to "yes" a further parameter XM appears for the definition of the position for the extension of the part receptacle. The destination depth, defined in parameter X2, is at the same time the position for positioning the part receptacle.

Notes:

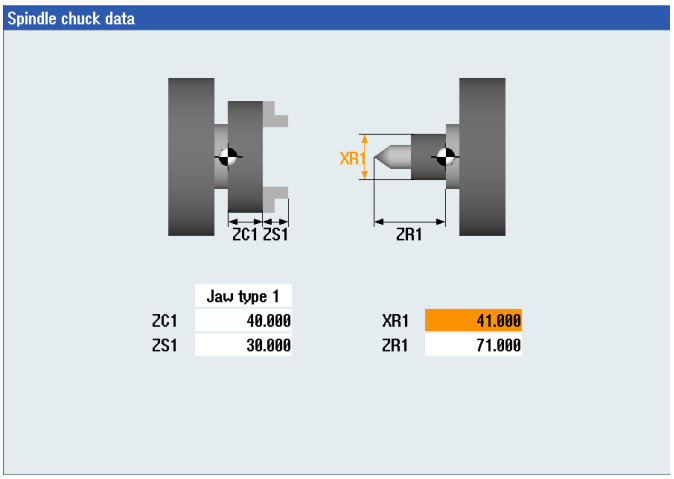
Tailstock activation

Software-Option
For the depiction of the tailstock the option "ShopMill/ ShopTurn" is required.



Function mask Turning

MD52218 $MCS_FUNKTION_MASK_TURN	Function mask Turning
	Change to: Bit 2 = 1



Description:

Bit 2: Release of tailstock

The tailstock dimensions are set up in the mask "Spindle chuck data".

Tailstock data

MD53242[x] $MAS_TAILSTOCK_PARAMETER	Tailstock data
Default: 0 mm	Change to:

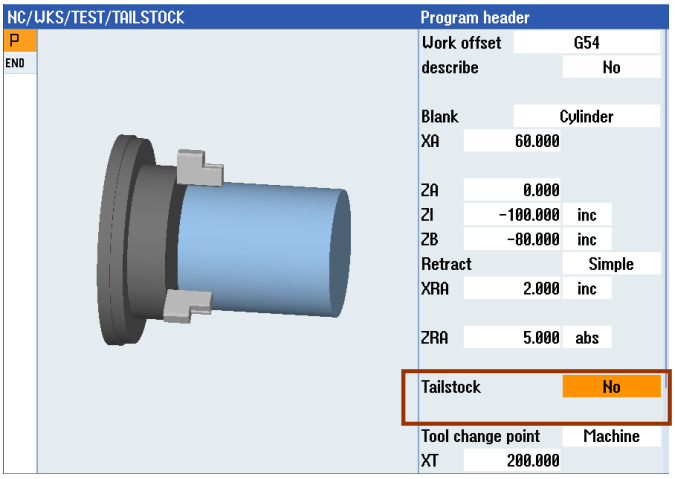
AX3: MCS1
MD53242[0] = XR1 Tailstock diameter
MD53242[1] = ZR1 Tailstock length

Description:

The inputs into the parameters "XR1" for tailstock diameter and "ZR1" for tailstock length must be entered into the machine data MD53242[0] and MD53242[1] of the data set for the C-axis. These entries are then transferred to the data set of the spindle. The tailstock dimensions are used in the simulation for the depiction of the tailstock.

Tailstock programming

After release of the tailstock in MD52218 the parameter "Tailstock" appears additionally in the program header of a ShopTurn-Program.



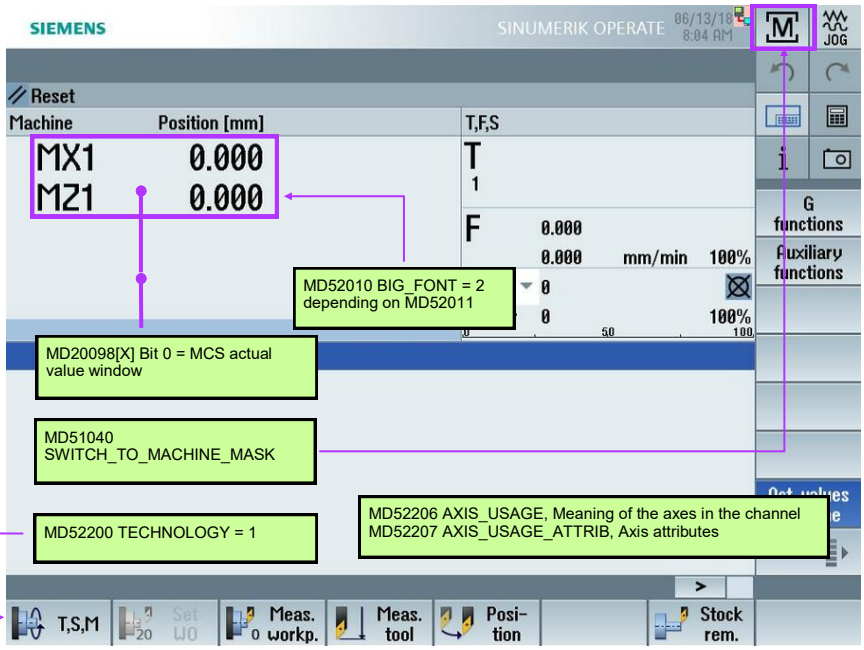
With tailstock "yes" a further parameter XRR for the definition of the retraction level in the X-axis for the prevention of collisions with the tailstock.

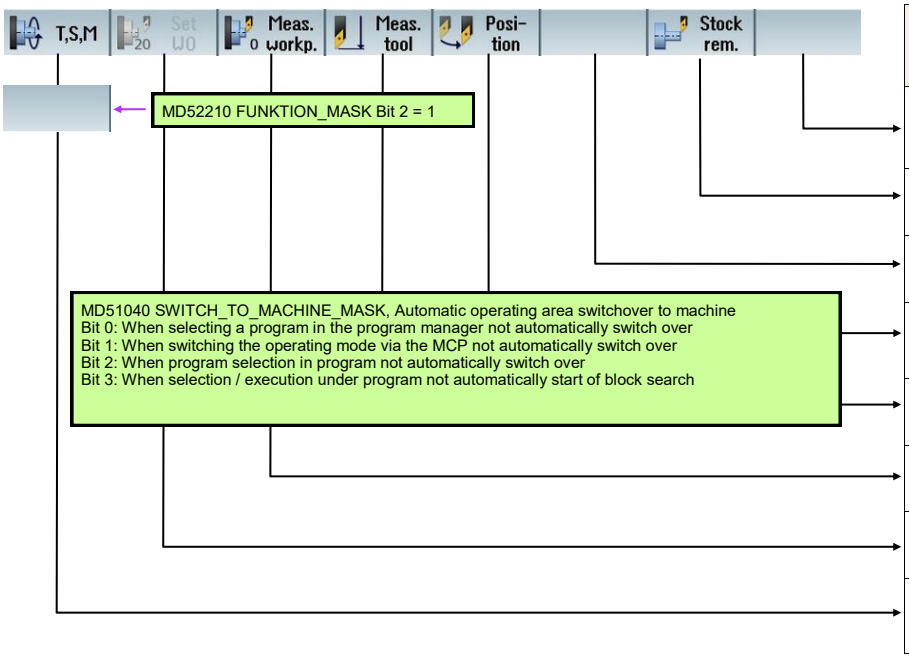


Notes:

Menu and Technology settings

Machine —> JOG

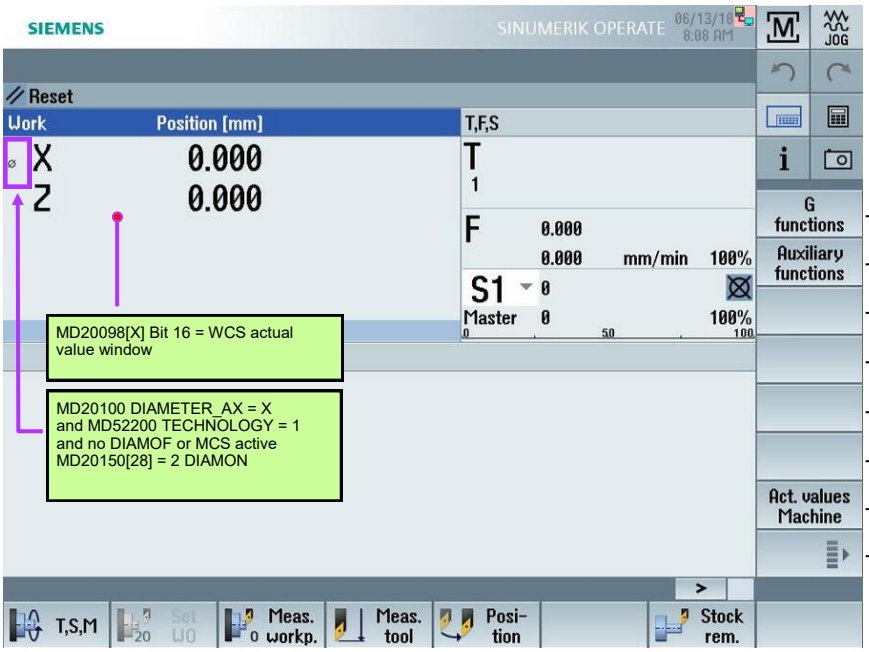




Softkey selection	Description
MD52200 = 1	=1: Technology Turning
MD52200 = 1	=1: Technology Turning
MD52200 = 1	=1: Technology Turning
MD52210 Bit 2 = 0	Bit 2 = 0 Show Softkey "T,S,M" in JOG area

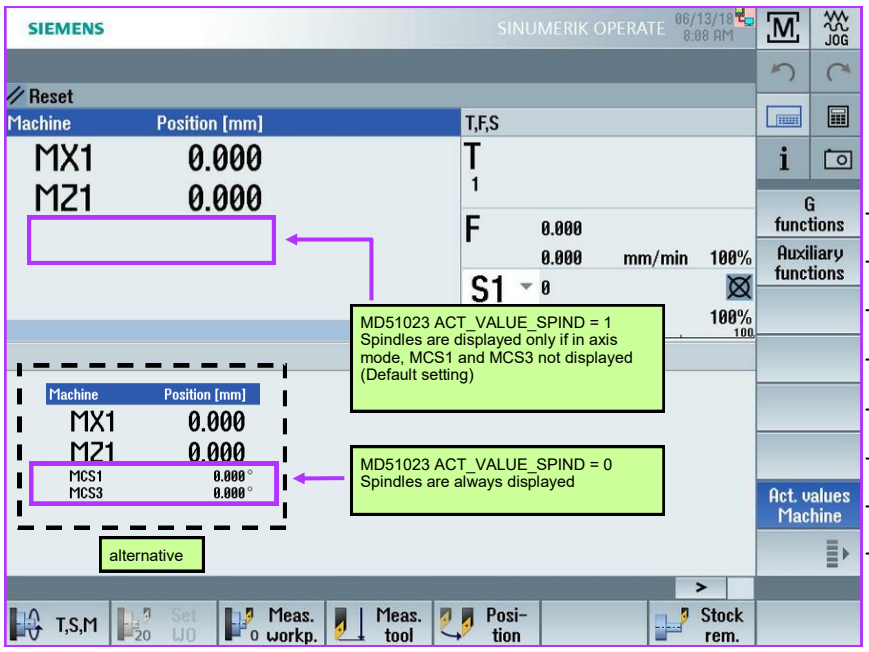
Notes:

Machine —> JOG —> WCS



Softkey selection	Description
Switchover MCS/WCS	Switchover Softkey from MCS to WCS via PLC

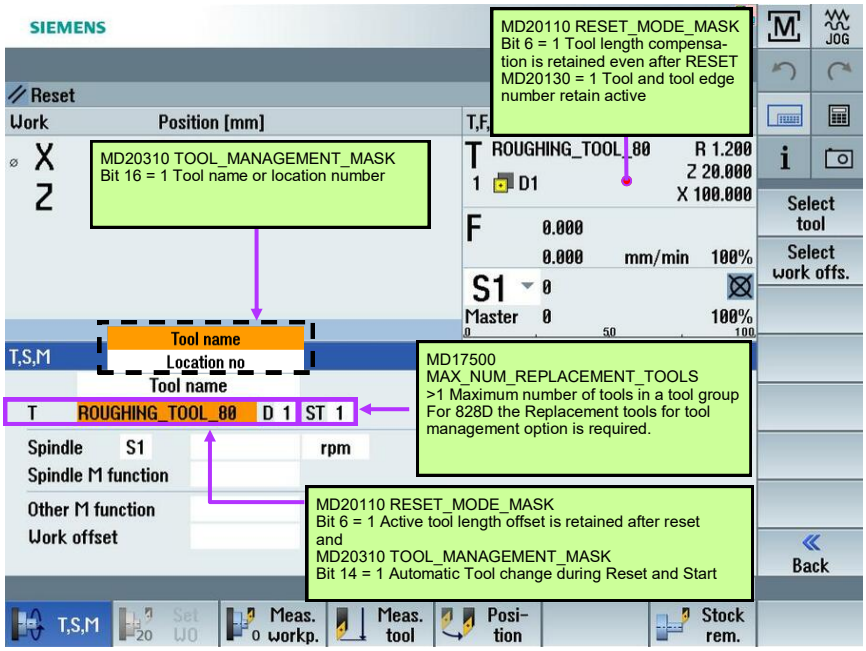
Machine —> JOG —> MCS



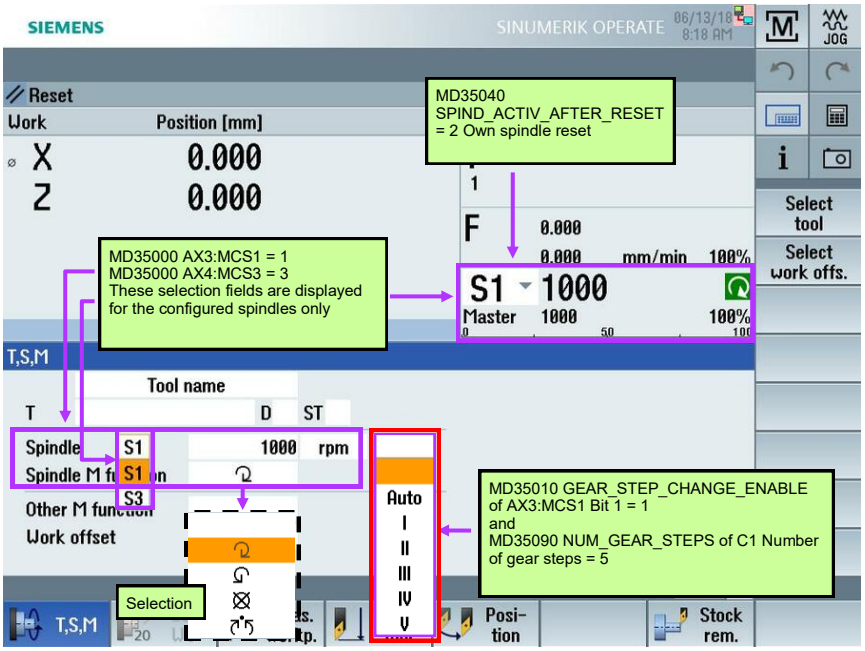
Softkey selection	Description
Switchover MCS/WCS	Switchover Softkey from MCS to WCS via PLC

Notes:

Machine —> JOG —> T,S,M

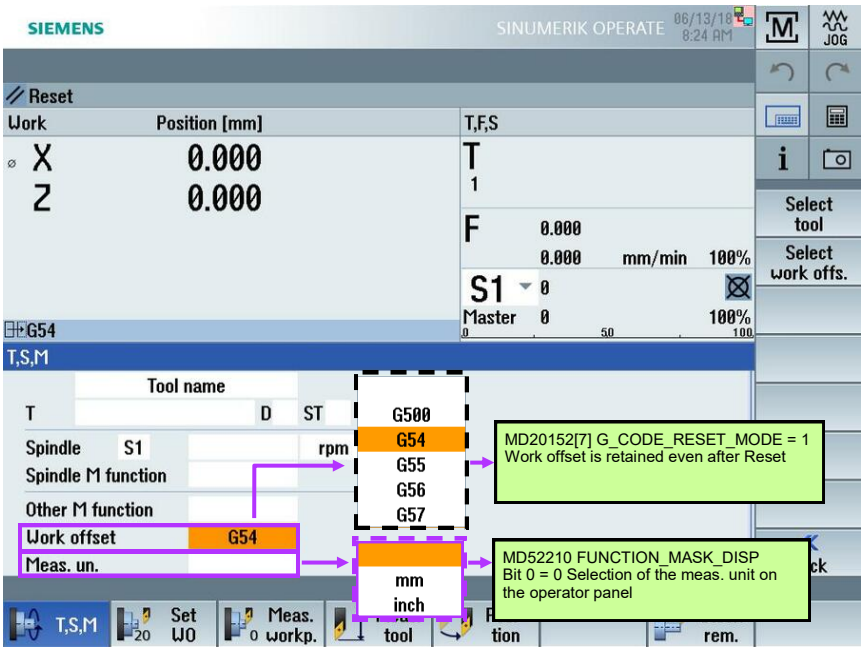


Machine —> JOG —> T,S,M

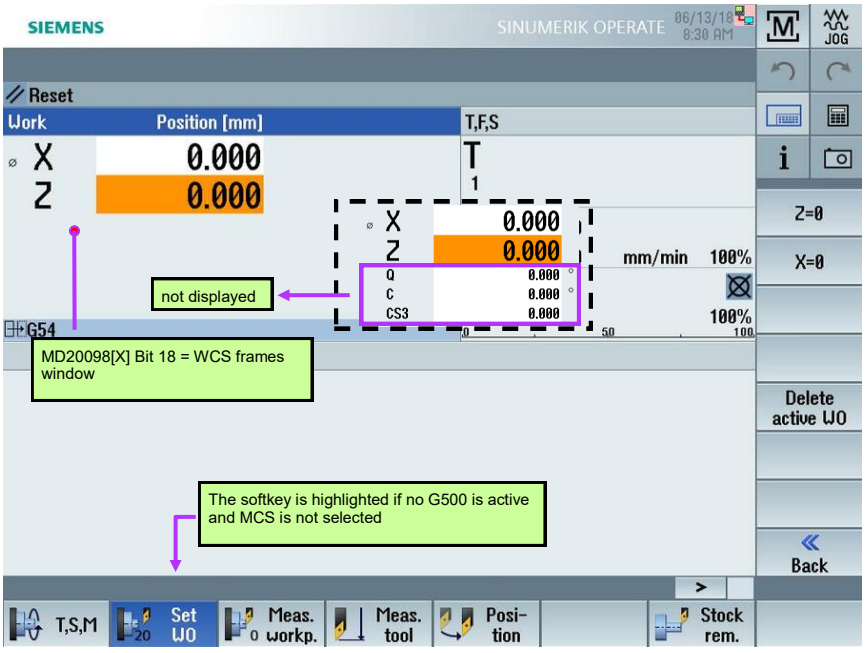


Notes:

Machine —> JOG —> T,S,M

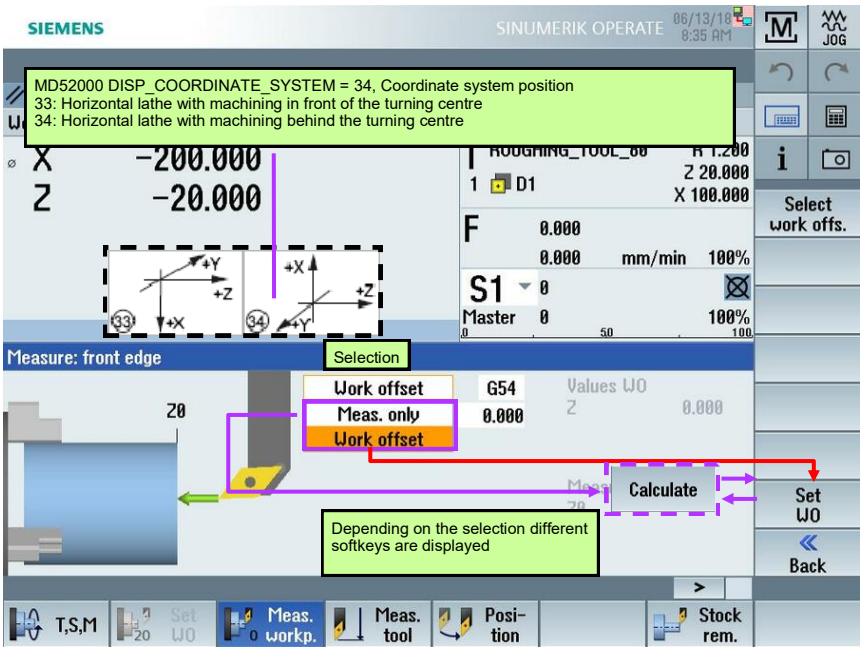


Machine —> JOG —> Set WO

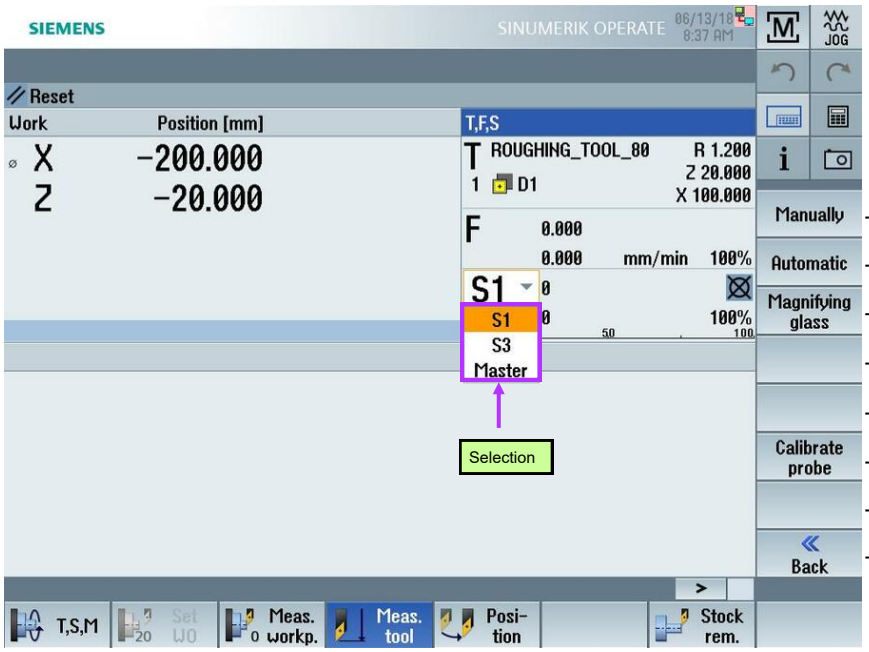


Notes:

Machine —> JOG —> Measure workpiece



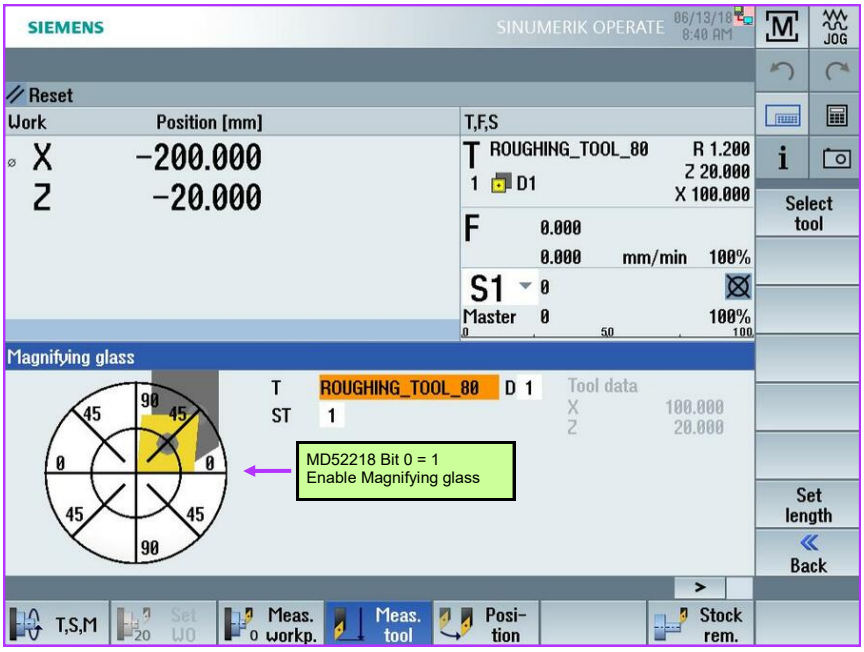
Machine —> JOG —> Measure tool



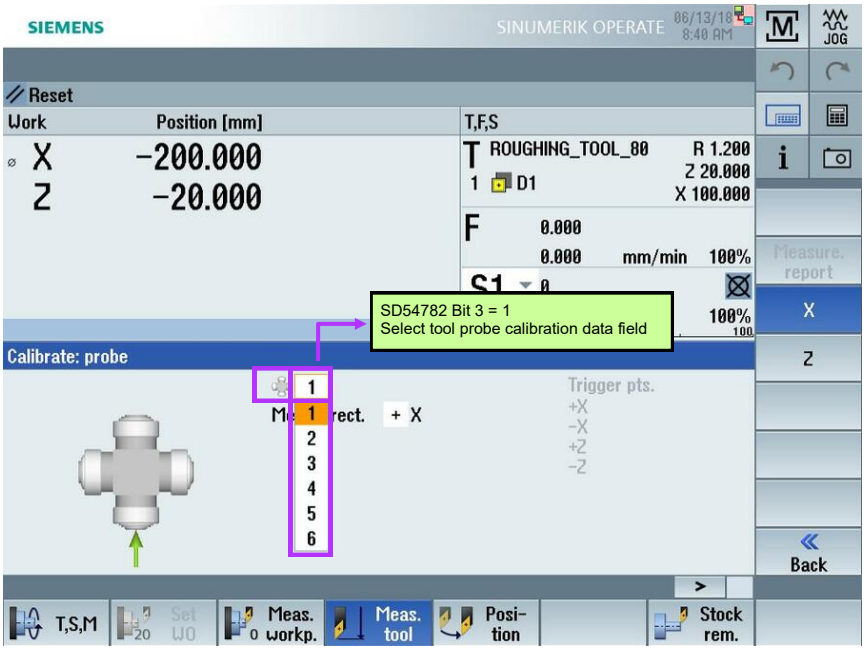
Softkey selection	Description
SD54782 Bit 2 = 1	Bit 2 = 1 Enable automatic tool measurement
MD52218 Bit 0 = 1	Bit 0 =1 Enable zoom under manual for tool measurement
SD54782	Bit 2 = 1 Enable automatic tool measurement

Notes:

Machine —> JOG —> Measure tool

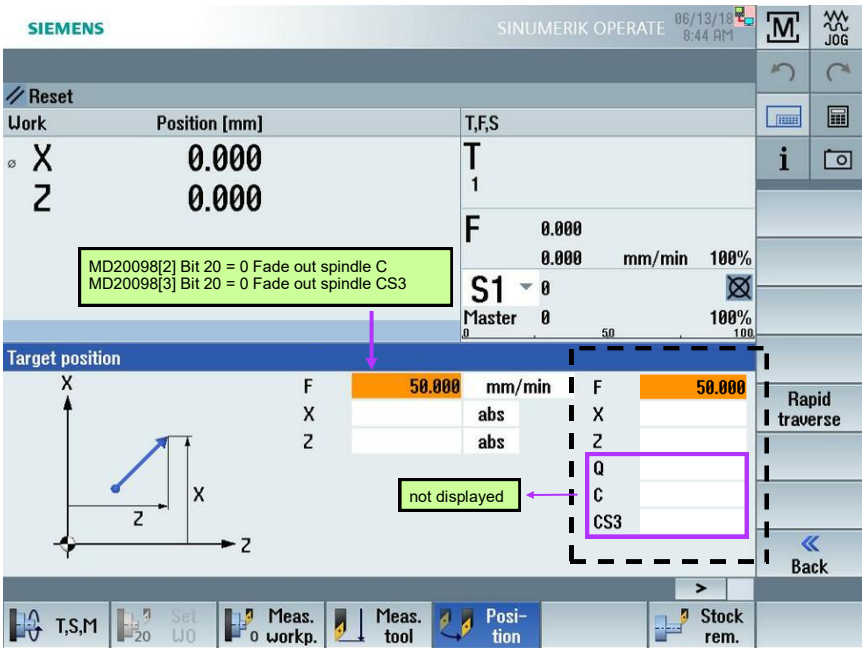


Machine —> JOG —> Measure tool

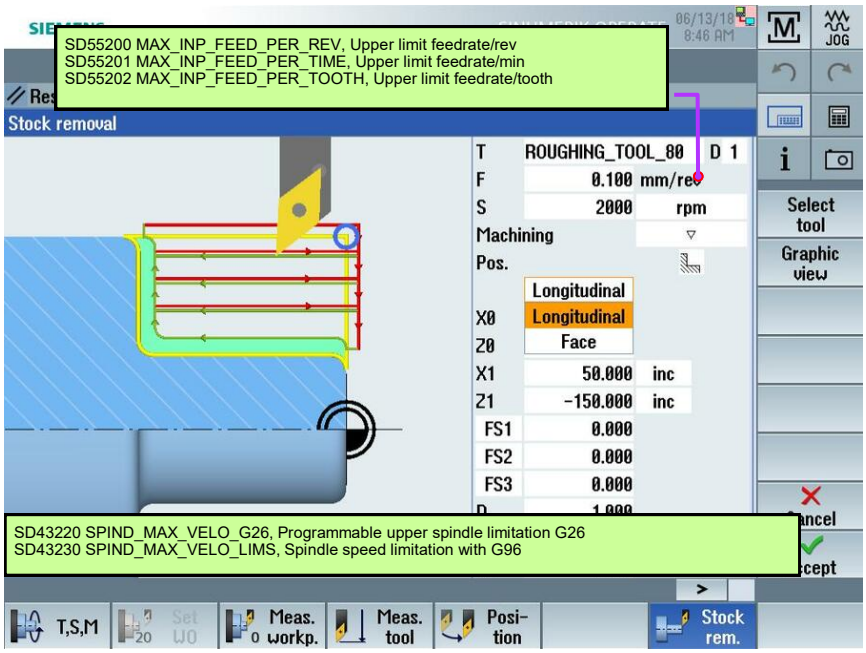


Notes:

Machine —> JOG —> Position

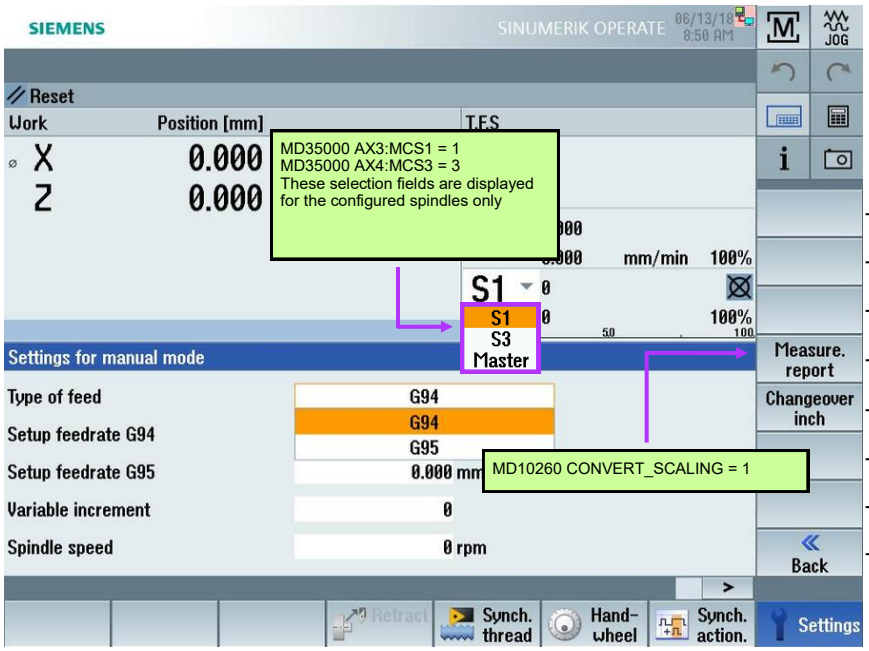


Machine —> JOG —> Stock removal

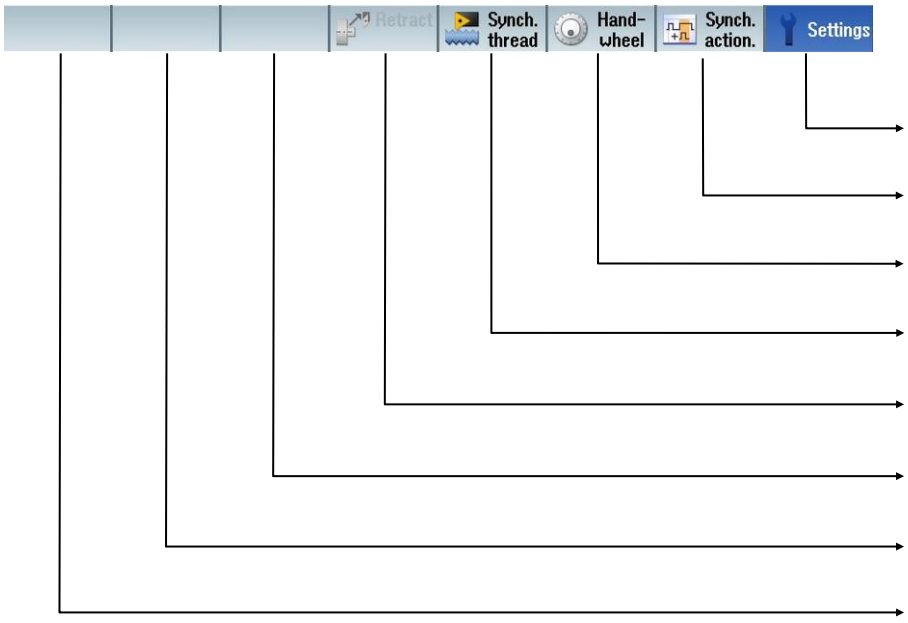


Notes:

Machine —> JOG —> Settings



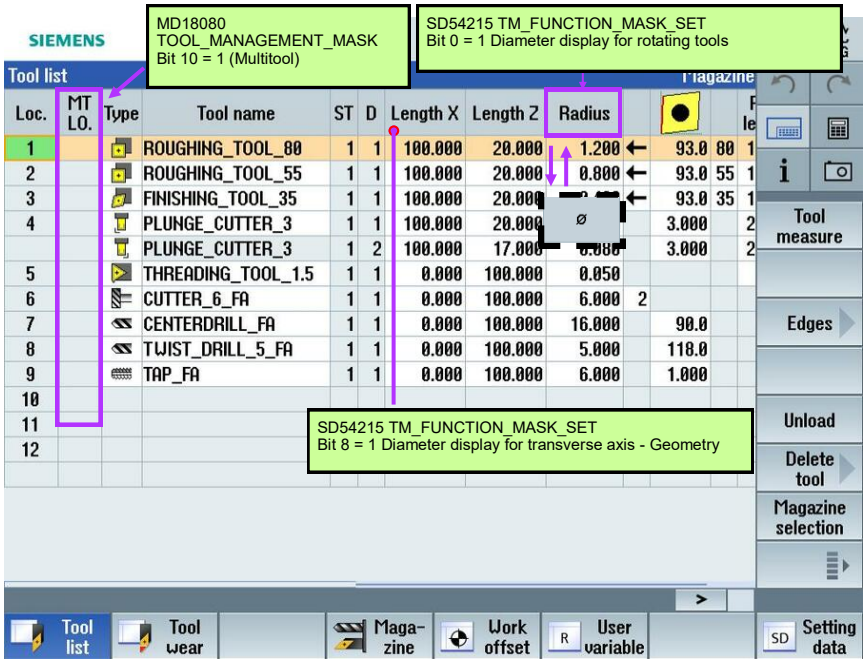
Softkey selection	Description
MD10260 = 1	1: Enable basic system conversion

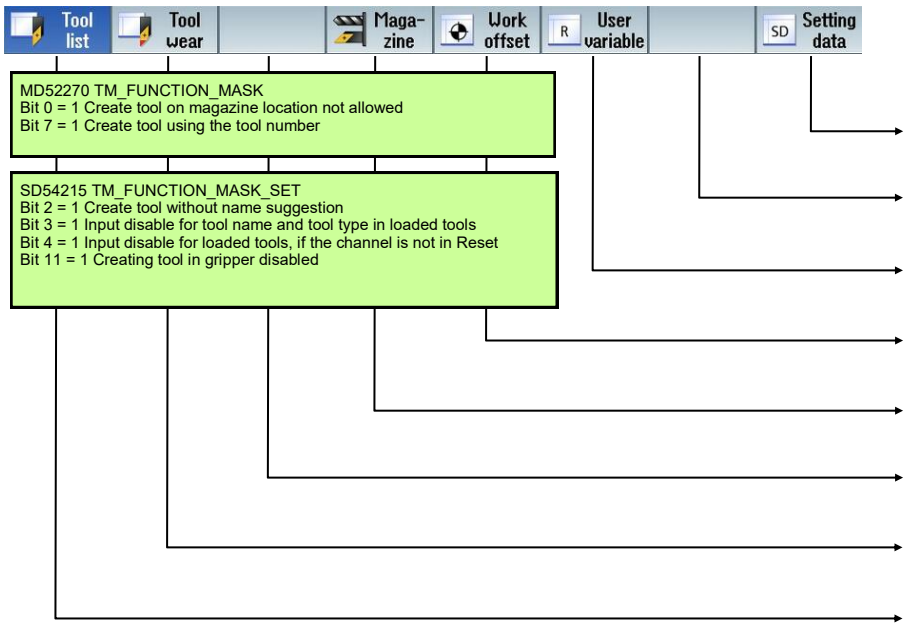


Softkey selection	Description
Option (828D only)	Extended operating functions
MD51067 = 1	=1: Show handwheel window (Handwheel must be configured in MD11350,11351,11352)
MD52200 = 1	=1: Technology Turning

Notes:

Parameter —> Tool list

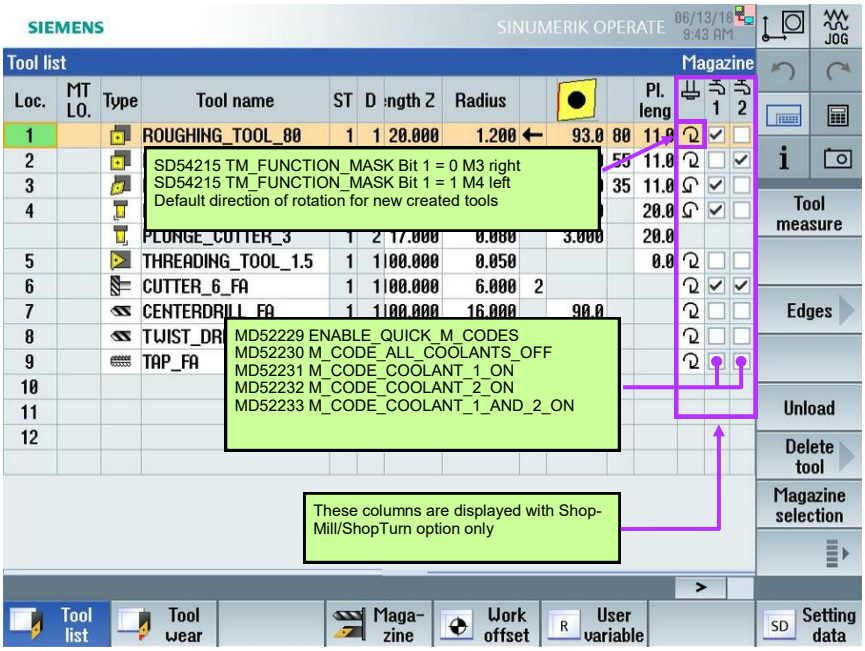




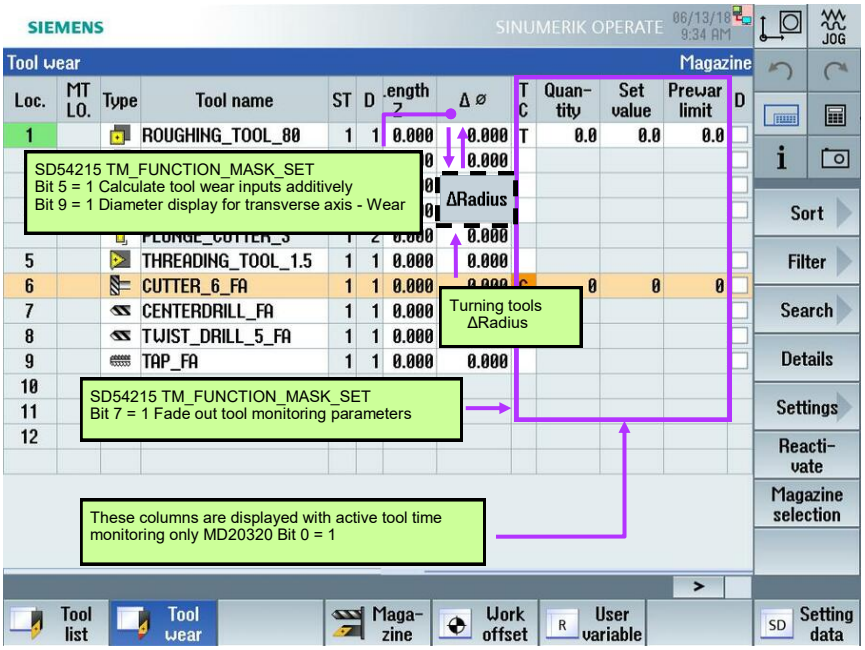
Softkey selection	Description

Notes:

Parameter —> Tool list



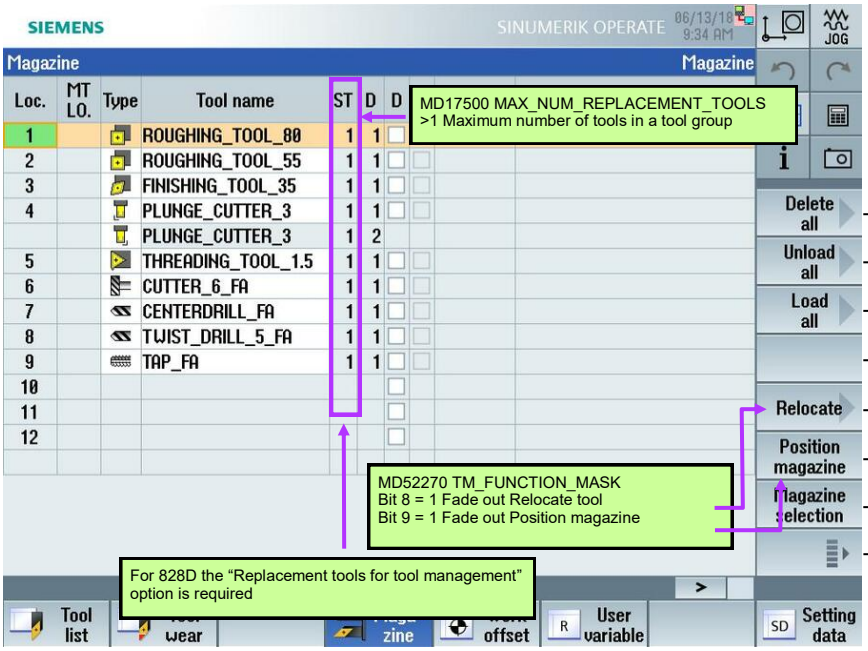
Parameter —> Tool wear



Softkey selection	Description
MD52270	Bit 12 = 0 Enable Reactivate tool

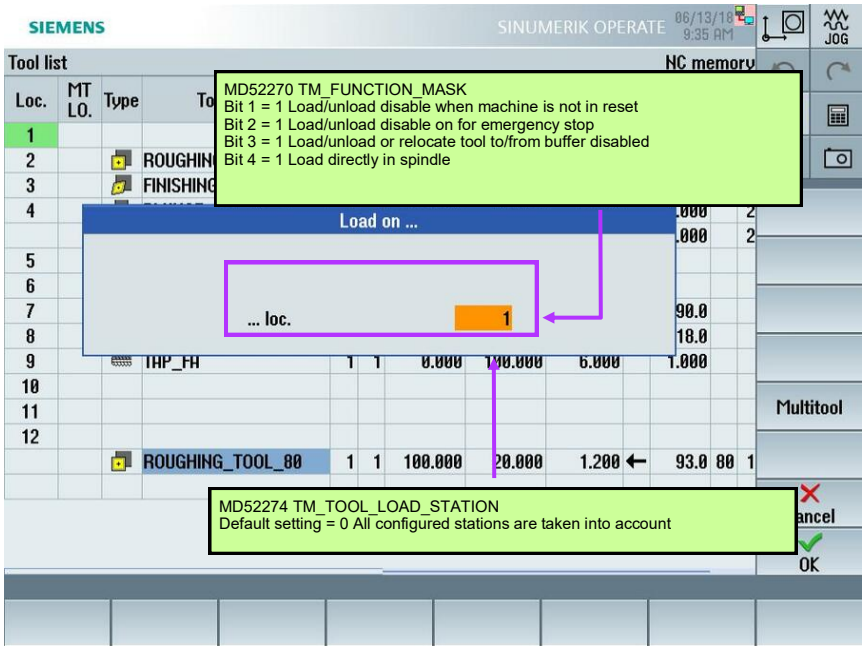
Notes:

Parameter —> Magazine



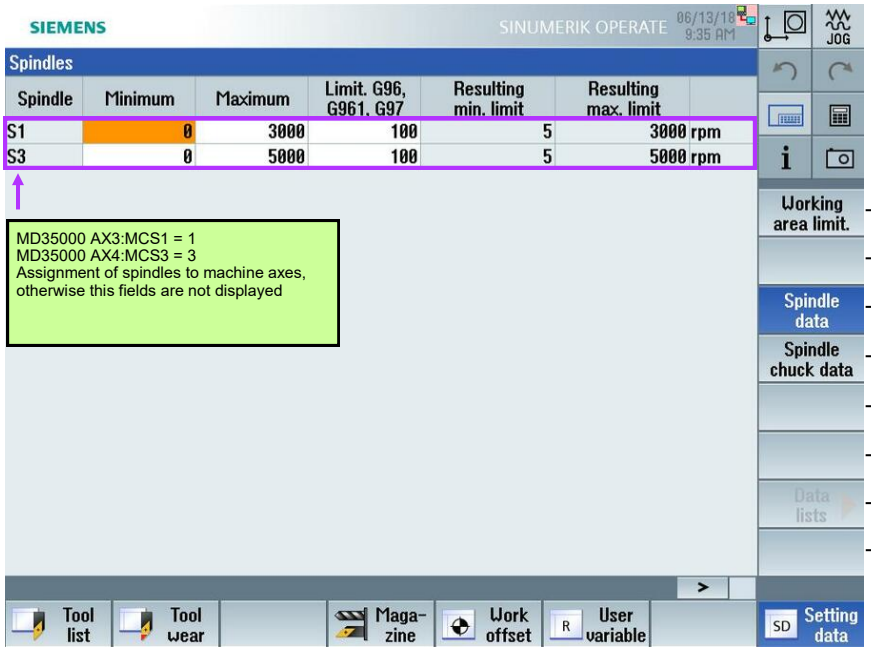
Softkey selection	Description
MD52270	Bit 8 = 0 Enable Relocate tool
MD52270	Bit 9 = 0 Enable Position magazine

Parameter —> Tool load



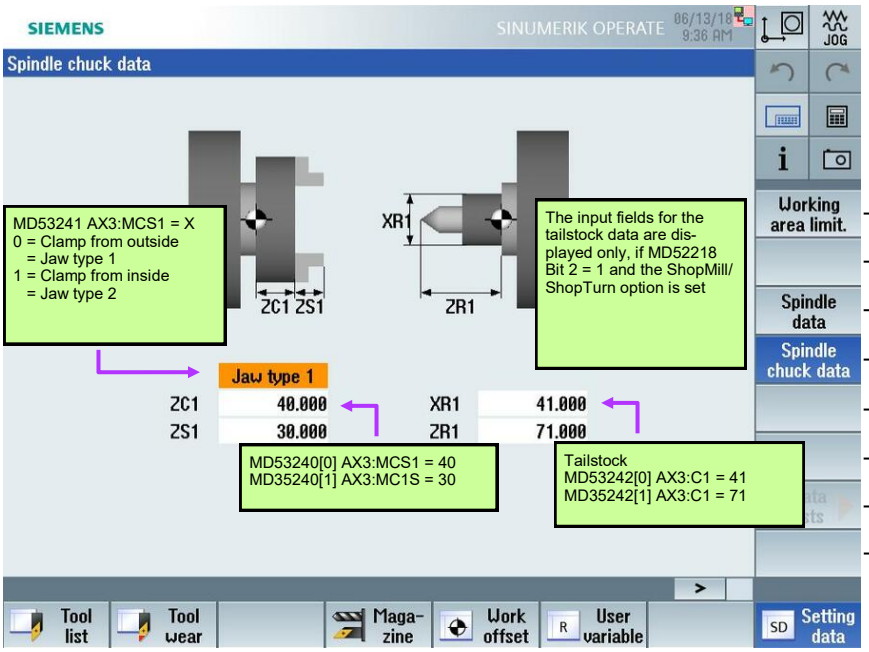
Notes:

Parameter —> Setting data



Softkey selection	Description
MD52200 = 1	=1: Technology Turning

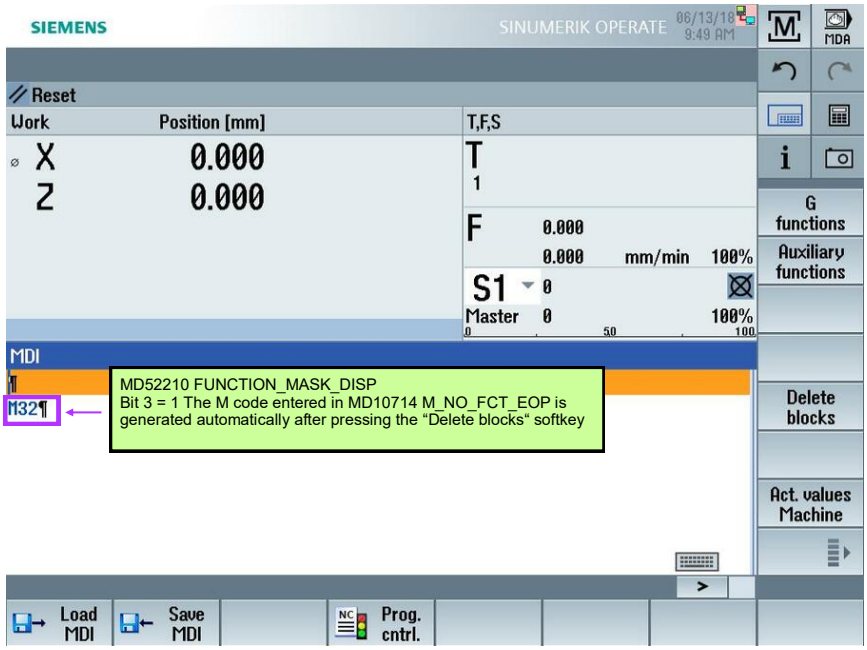
Parameter —> Setting data

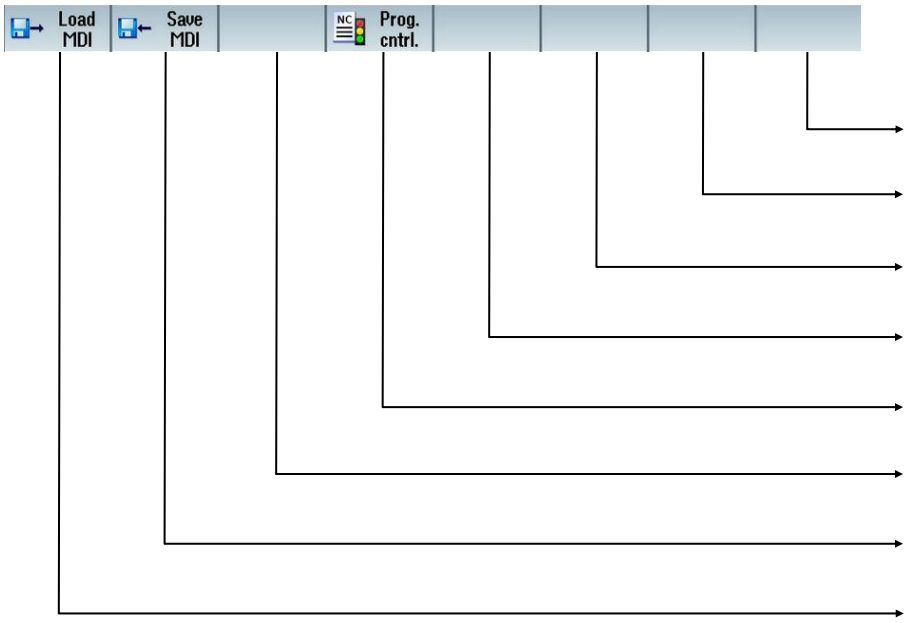


Softkey selection	Description
MD52200 = 1	=1: Technology Turning

Notes:

Machine —> MDA

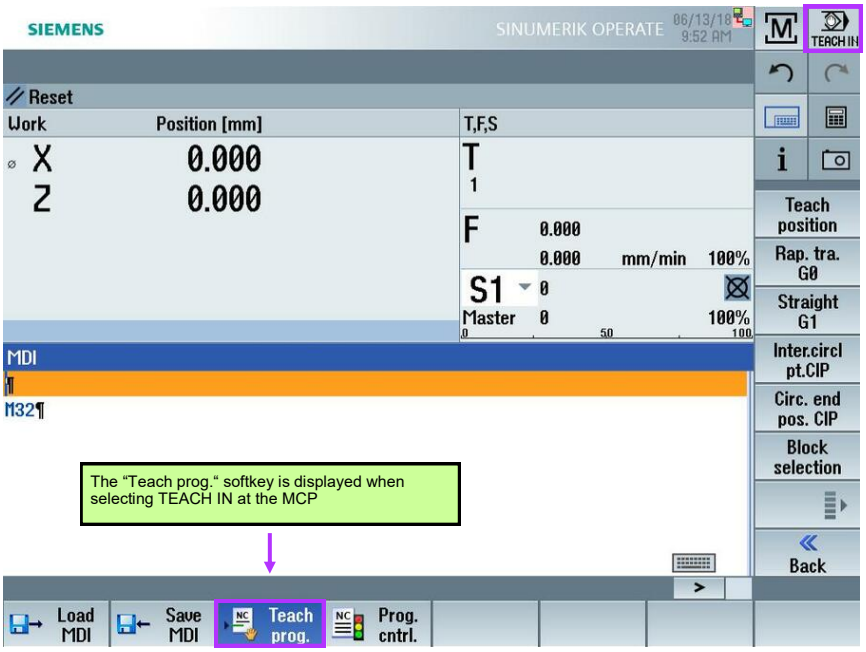


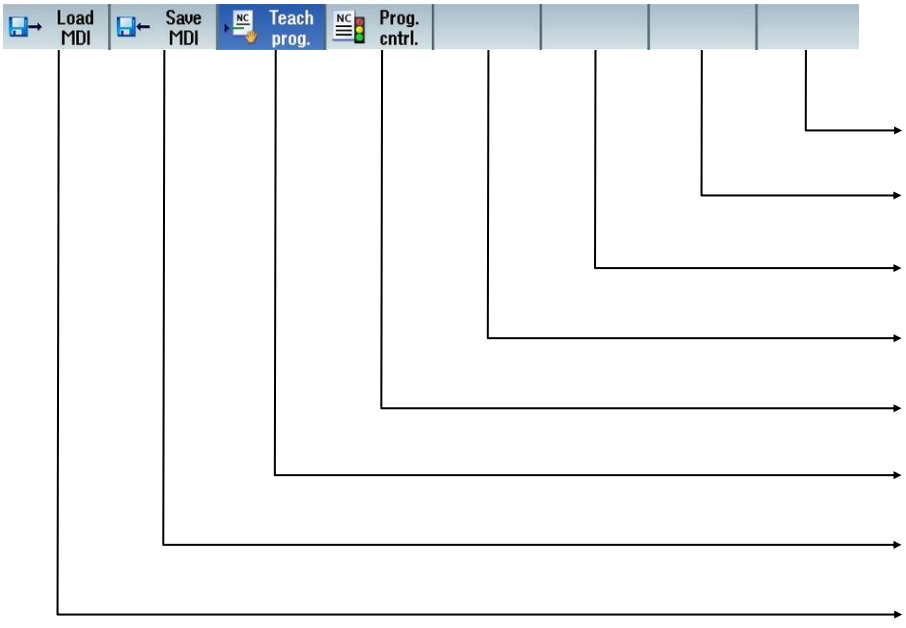


Softkey selection	Description
Option (828D only)	Extended operating functions
Option (828D only)	Extended operating functions

Notes:

Machine —> TEACH IN —> Teach program

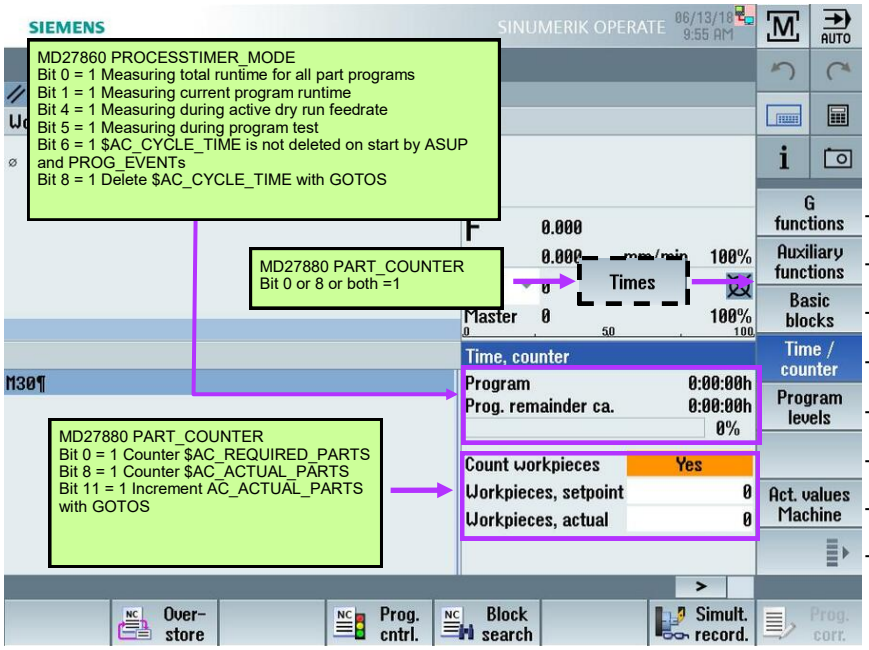




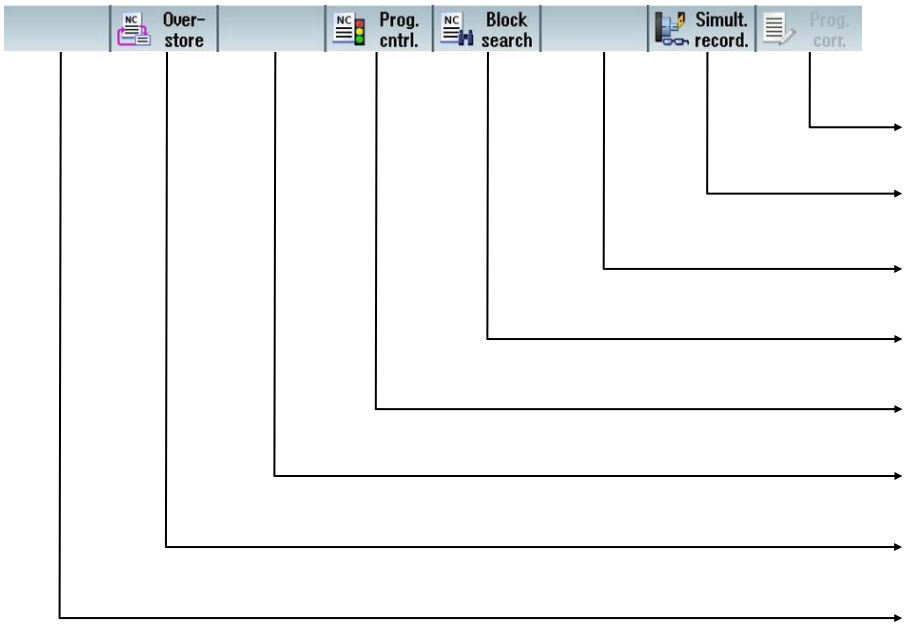
Softkey selection	Description
Option (828D only)	Extended operating functions
Option (828D only)	Extended operating functions
Option (828D only)	Extended operating functions

Notes:

Machine —> AUTO —> Time / Counter



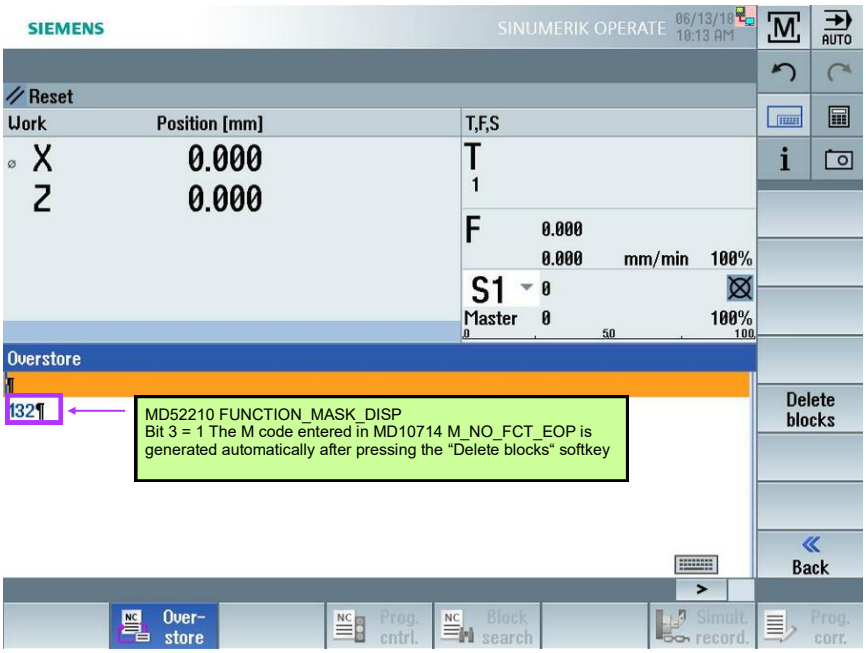
Softkey selection	Description
MD27880 Bit 0 oder Bit 8 = 1	=1: Sotkey Time/Counter =0: Softkey Times (Default setting)



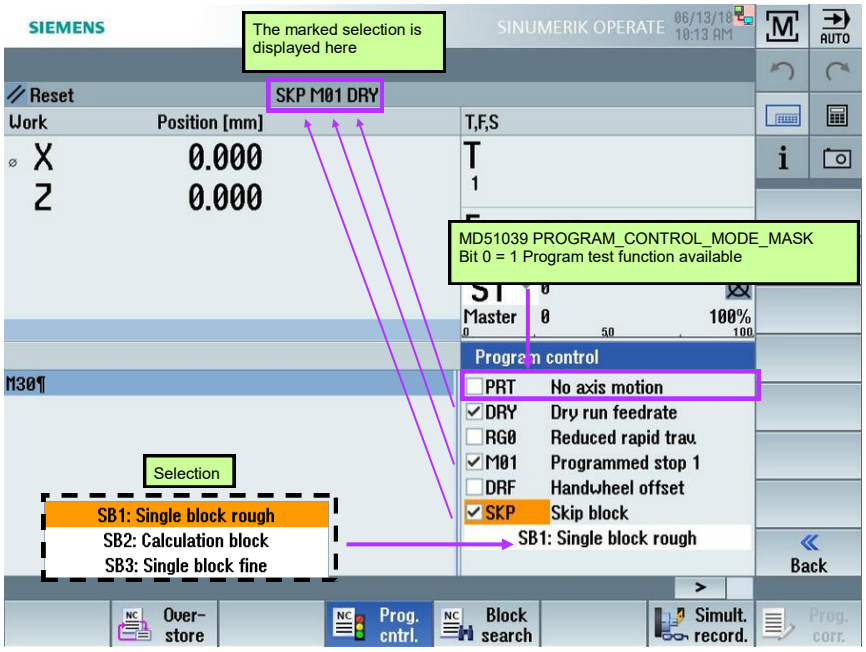
Softkey selection	Description
Option	Simultan. recording (real-time simulation)
Option (828D only)	Extended operating functions

Notes:

Machine —> AUTO —> Overstore

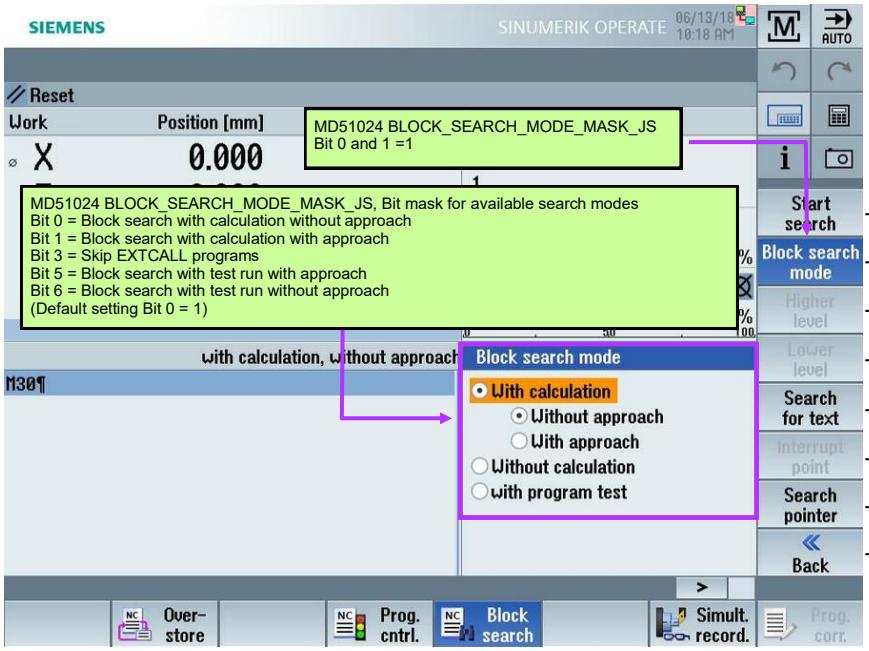


Machine —> AUTO —> Program control



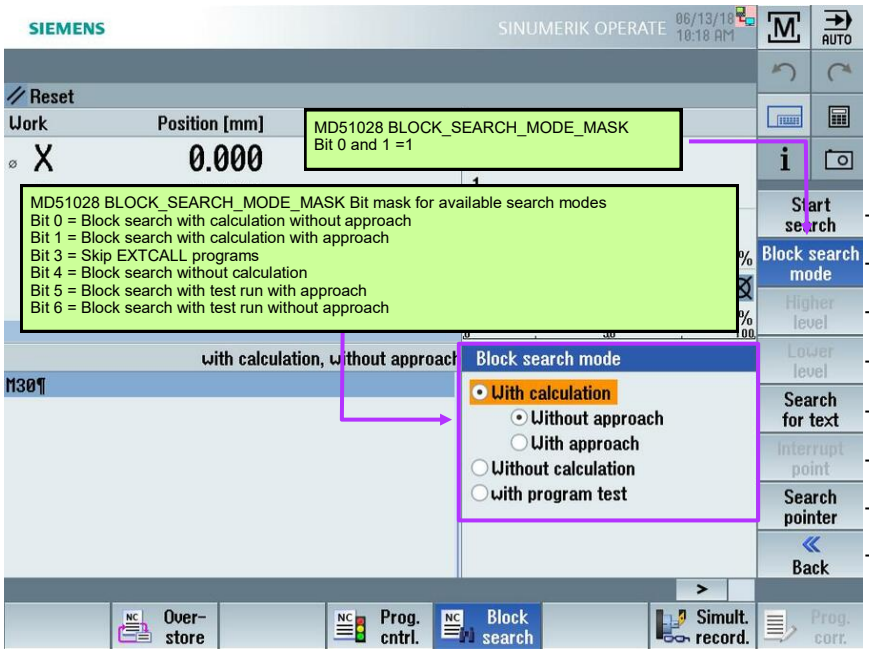
Notes:

Machine —> AUTO —> Block search (ShopTurn)



Softkey selection	Description
MD51024 Bit 0 and Bit 1 = 1	Bit mask for available search modes in ShopTurn program
Option (828D only)	Extended operating functions

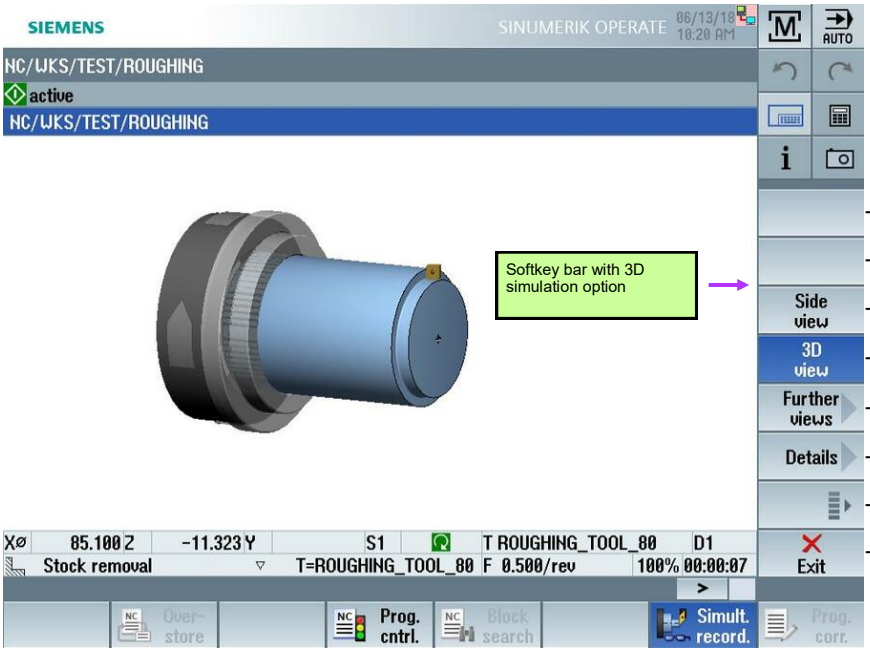
Machine —> AUTO —> Block search (G Code program)



Softkey selection	Description
MD51028 Bit 0 and Bit 1 = 1	Bit mask for available search modes in G code program
Option (828D only)	Extended operating functions

Notes:

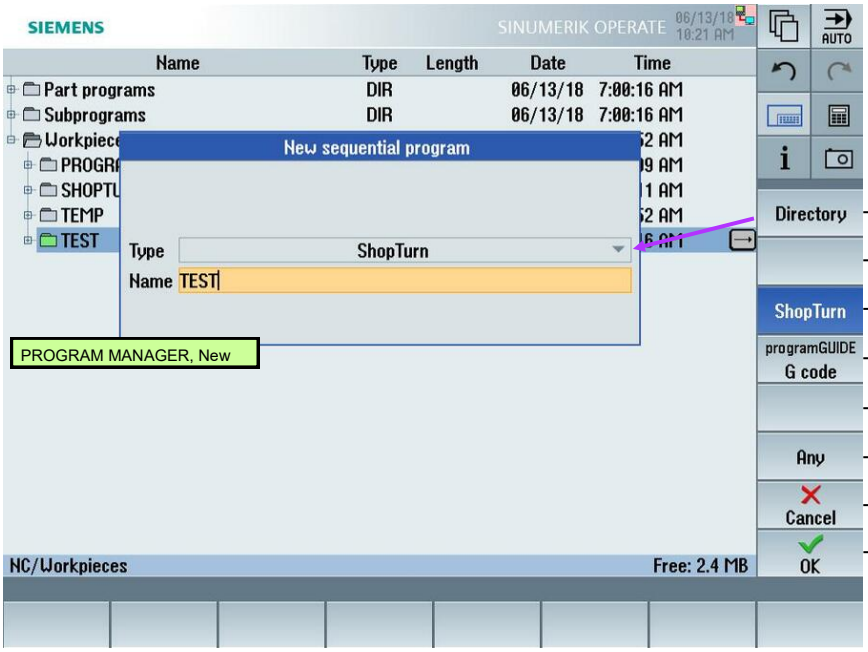
Machine —> AUTO —> Simultaneous recording



Softkey selection	Description
Option	3D simulation 1 (finished part)

Notes:

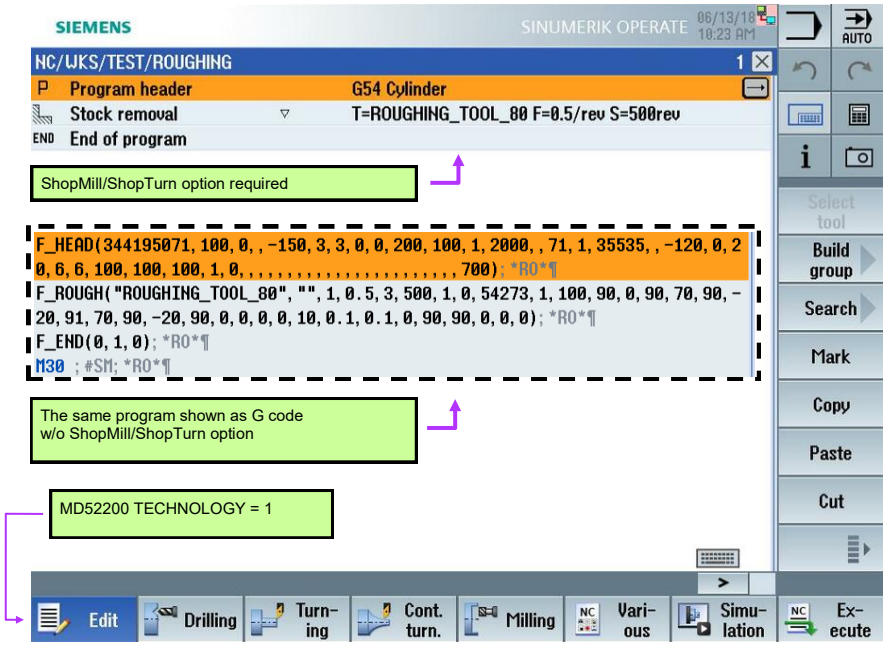
Program manager —> New program



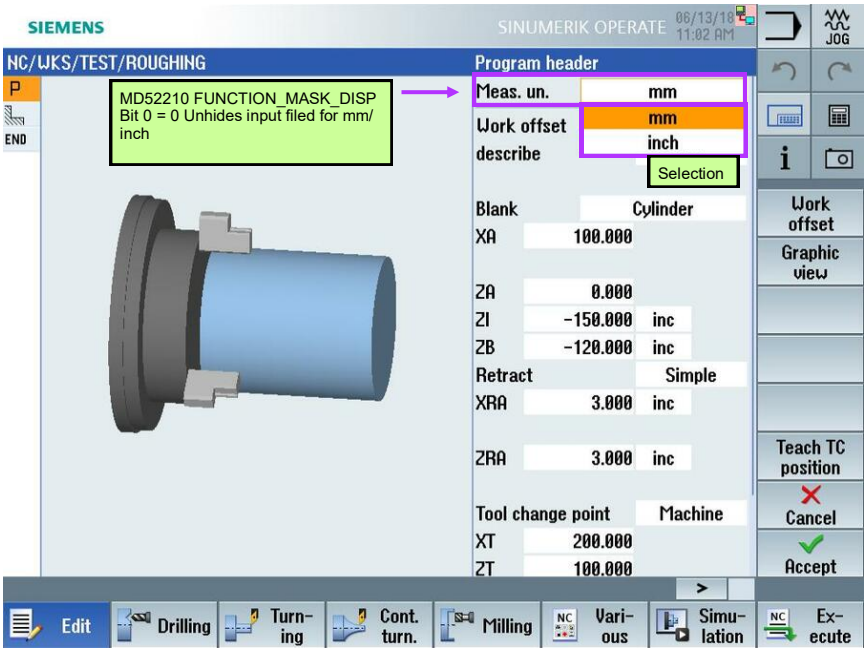
Softkey selection	Description
Option	3D simulation 1 (finished part)

Notes:

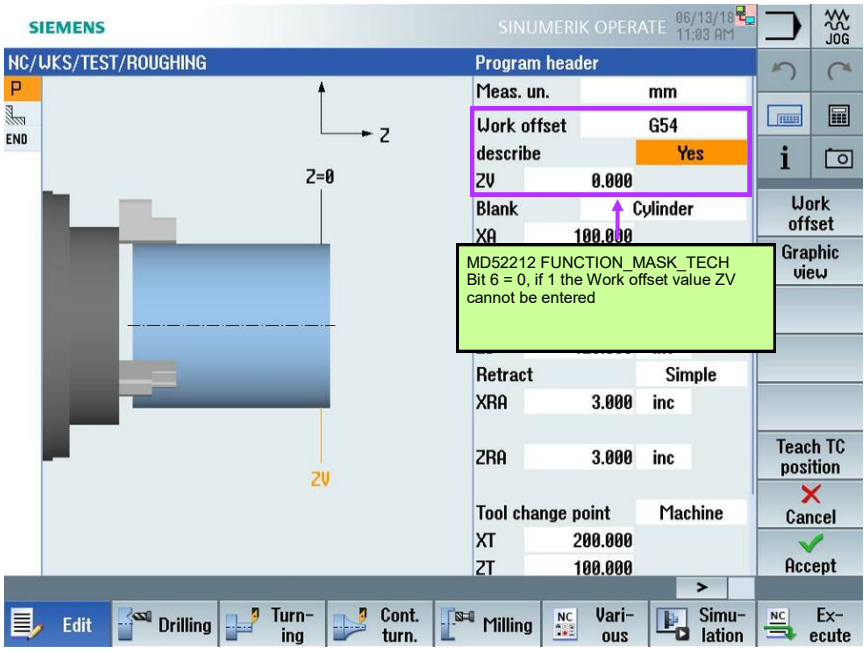
Program —> Edit


SIEMENS
SINUMERIK OPERATE
NC/WKS/TEST/ROUGHING
Program header
G54 Cylinder
Stock removal
T=ROUGHING_TOOL_80 F=0.5/rev S=500rev
End of program
ShopMill/ShopTurn option required
F_HEAD(344195071,100,0,,-150,3,3,0,0,200,100,1,2000,,71,1,35535,,-120,0,20,6,6,100,100,100,1,0,,,

Program —> Edit —> Program header

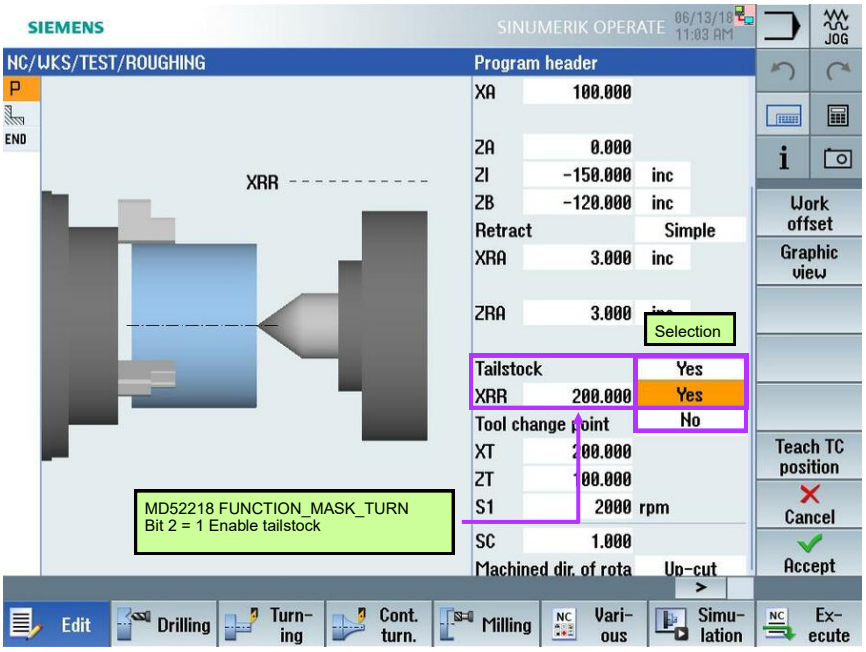


Program —> Edit —> Program header

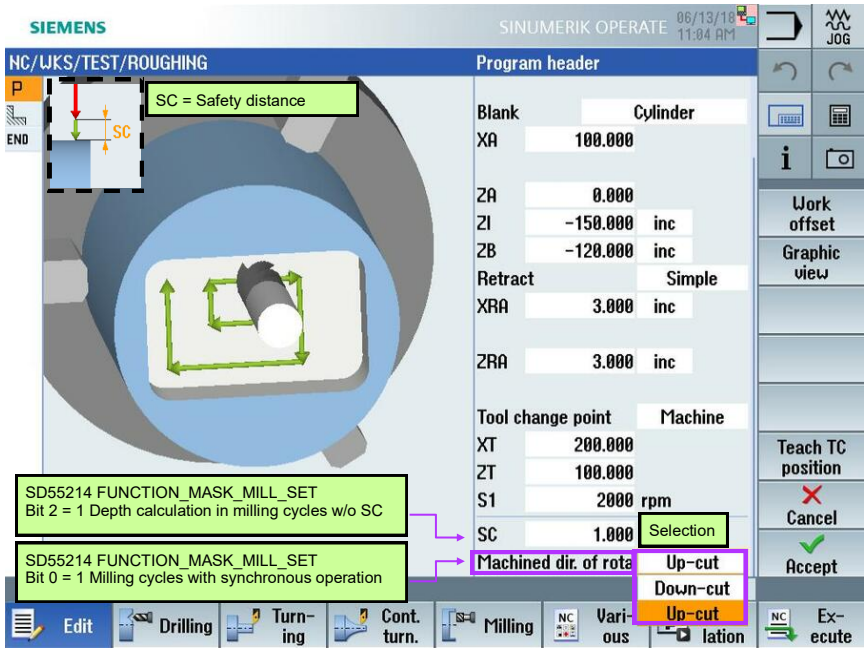


Notes:

Program —> Edit —> Program header

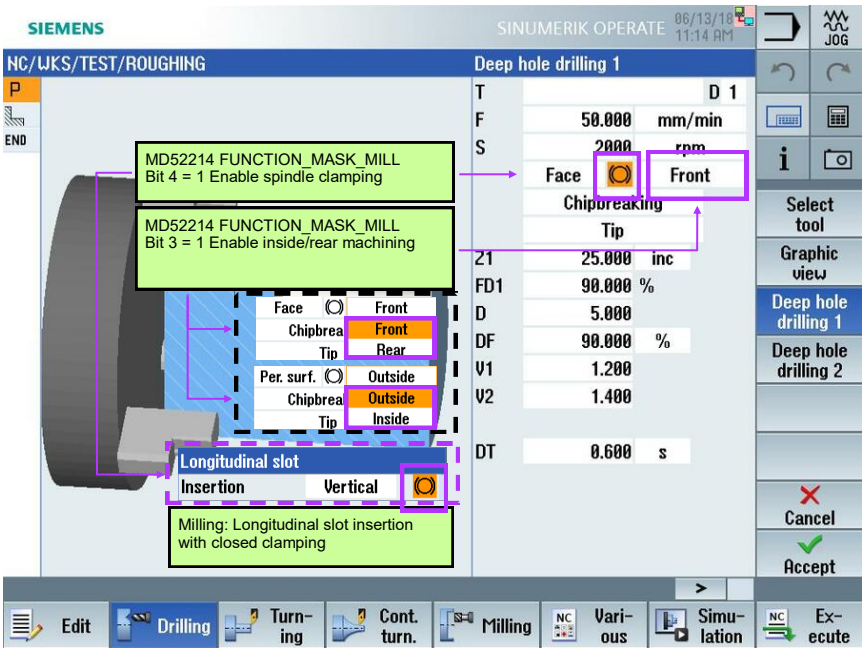


Program —> Edit —> Program header

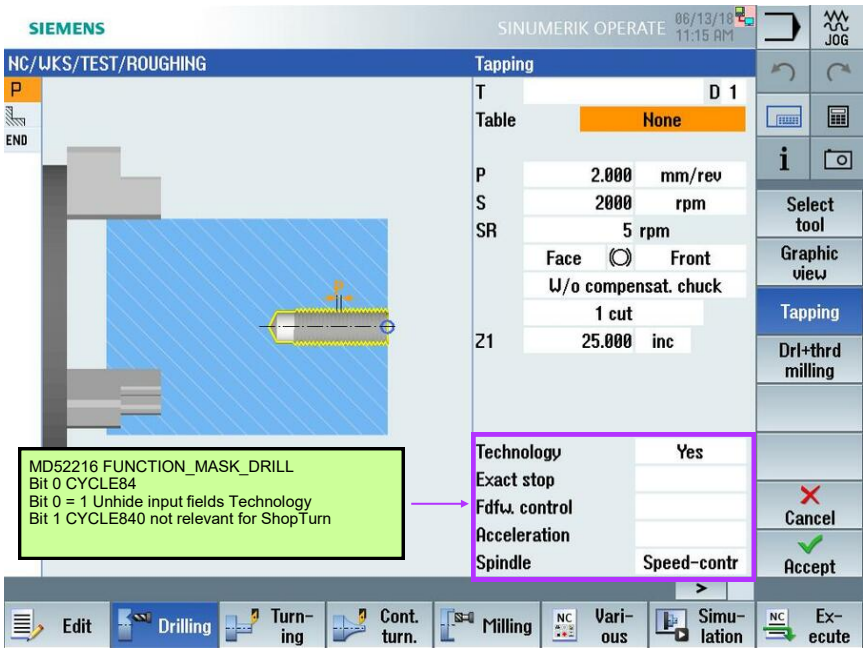


Notes:

Program —> Drilling

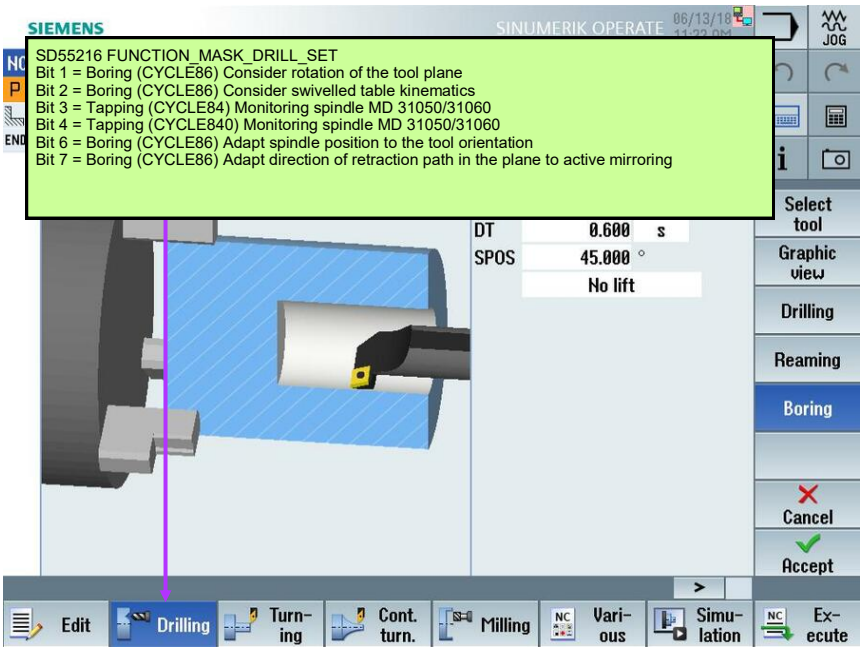


Program —> Drilling



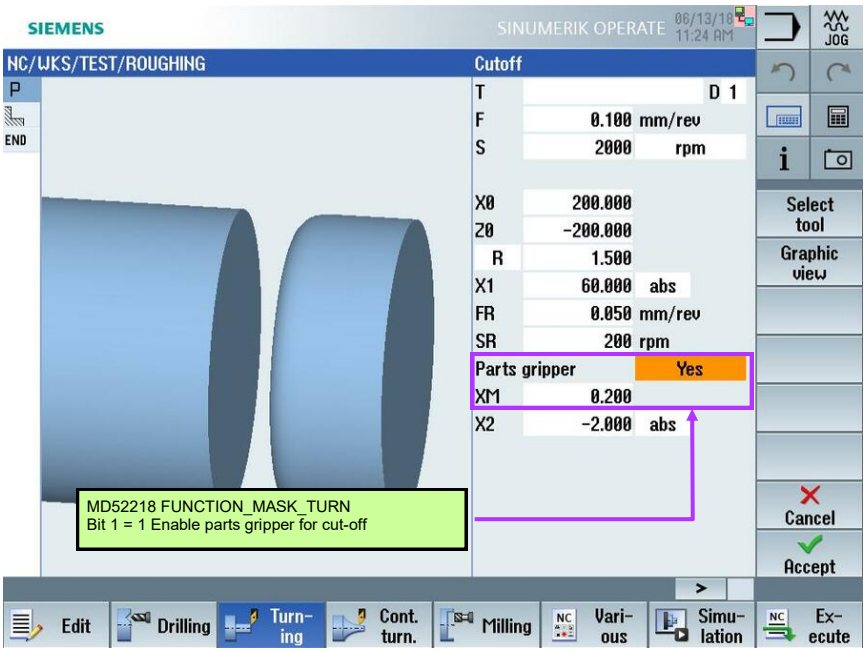
Notes:

Program —> Drilling

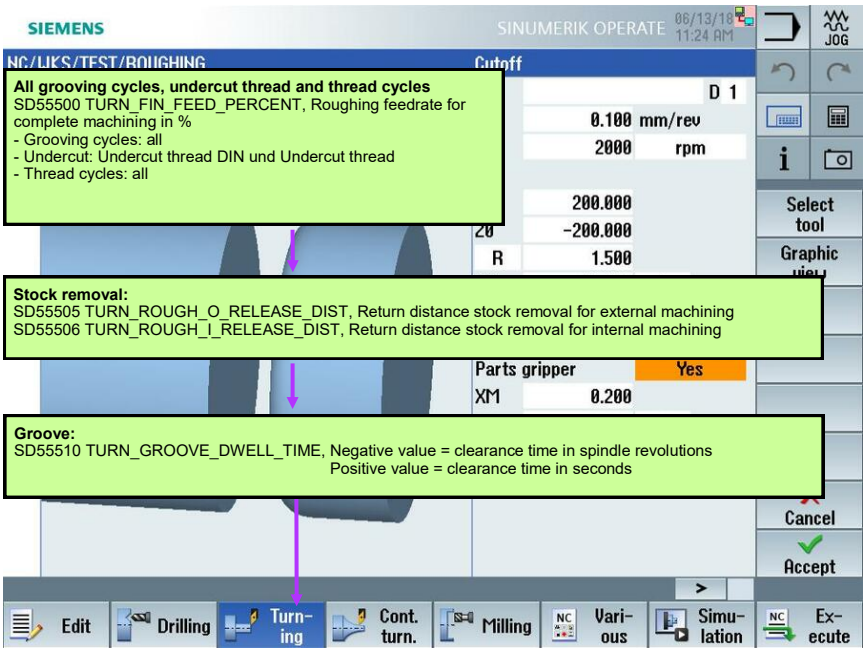


Notes:

Program —> Turning

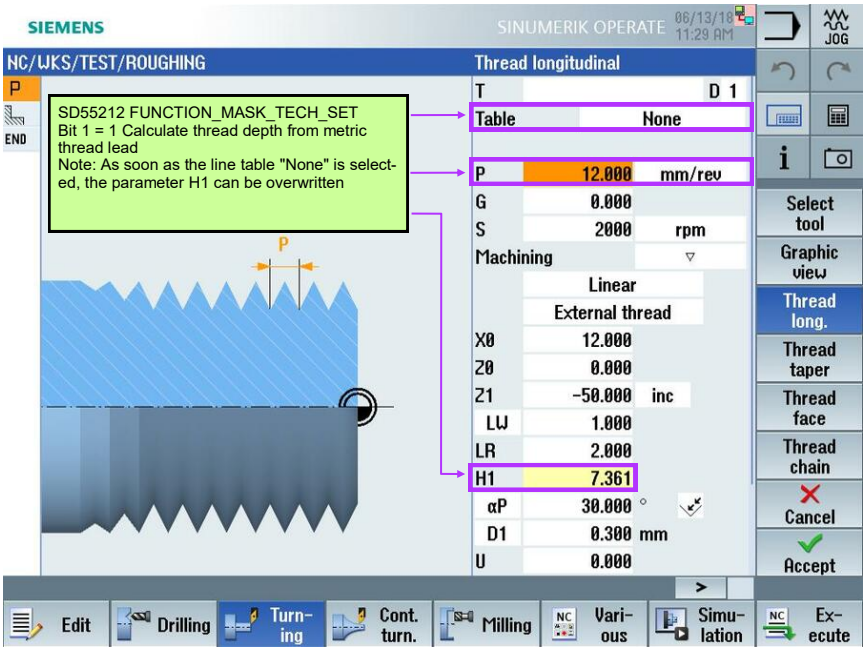


Program —> Turning

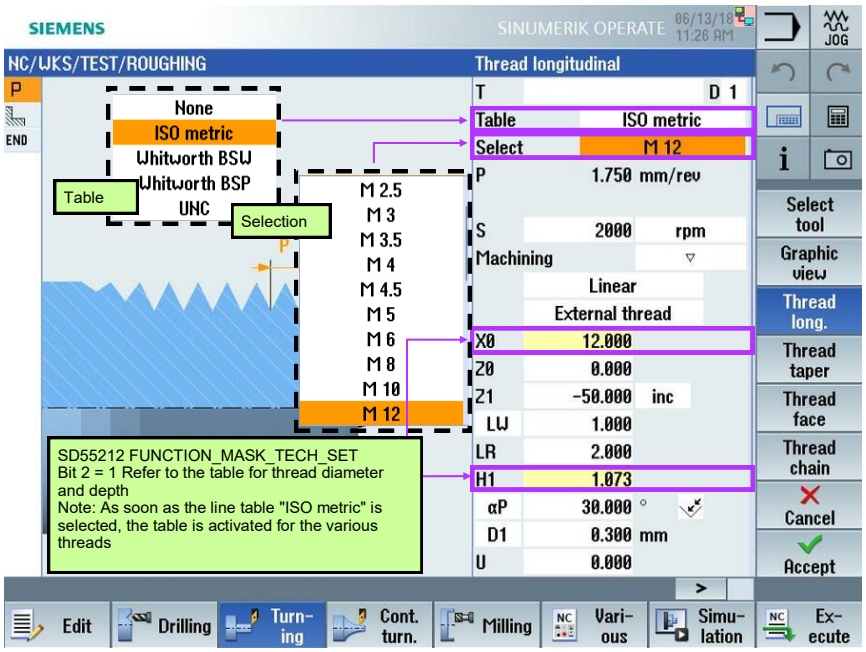


Notes:

Program —> Turning

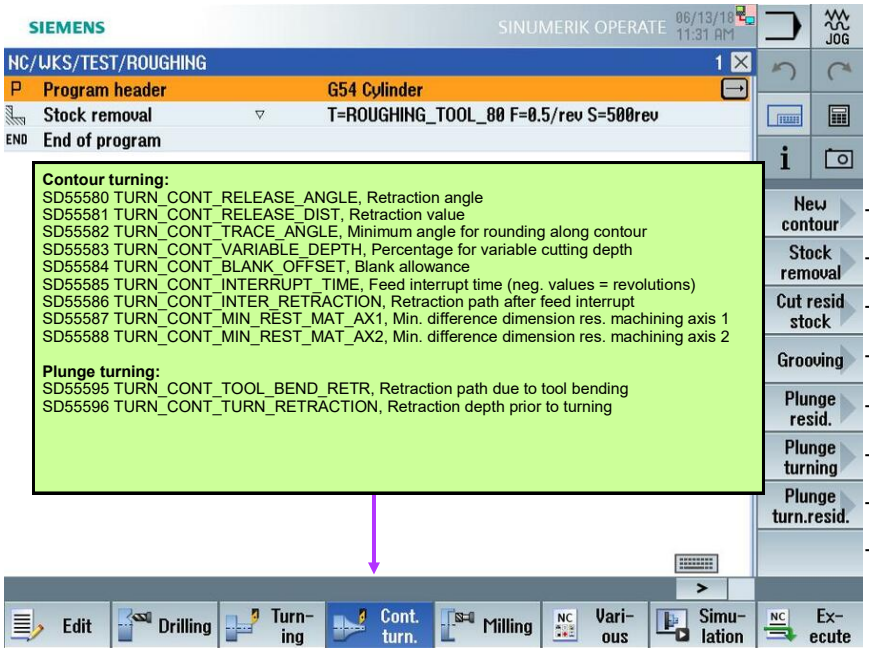


Program —> Turning



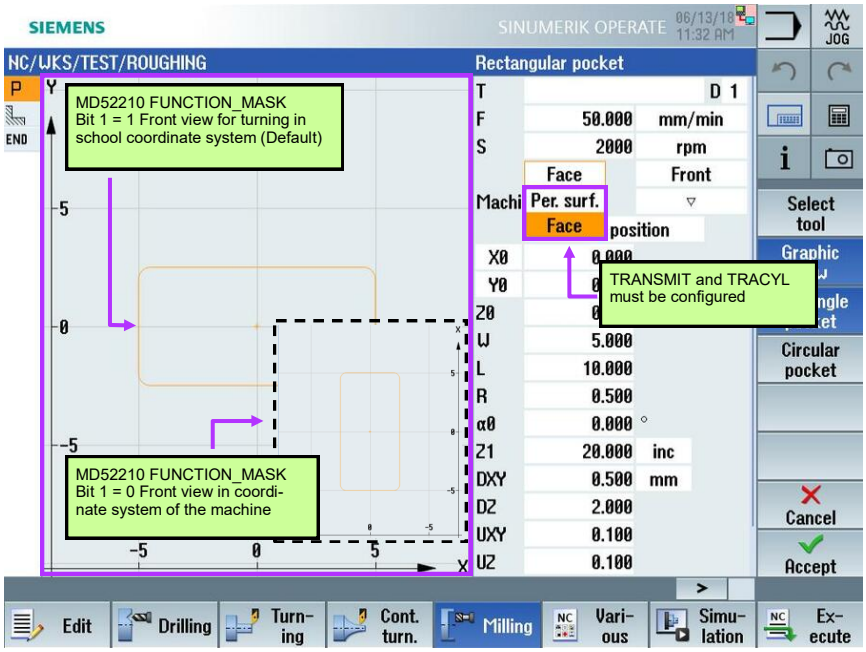
Notes:

Program —> Contour turning



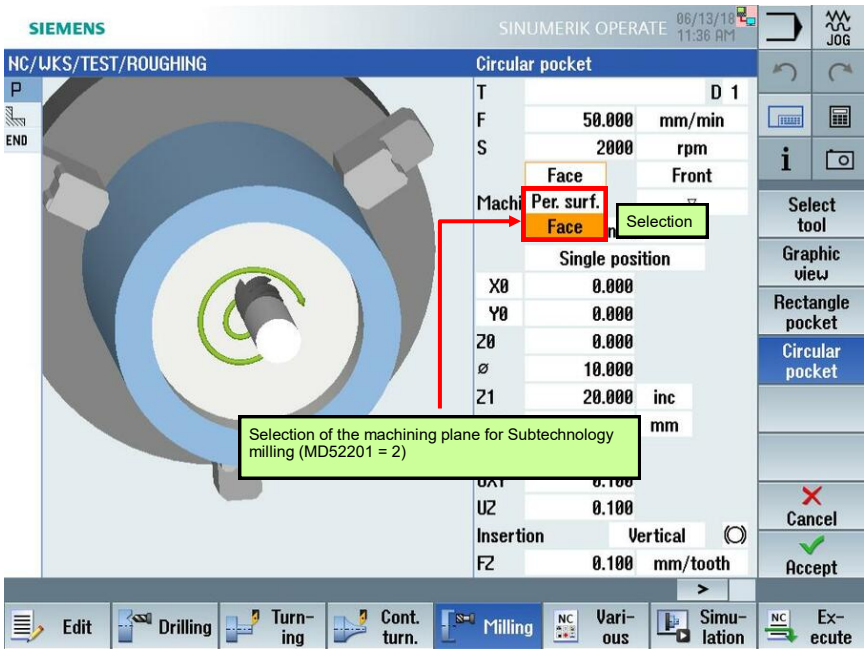
Softkey selection	Description
Option	Residual material detection and machining
Option	Residual material detection and machining
Option	Residual material detection and machining

Program —> Milling

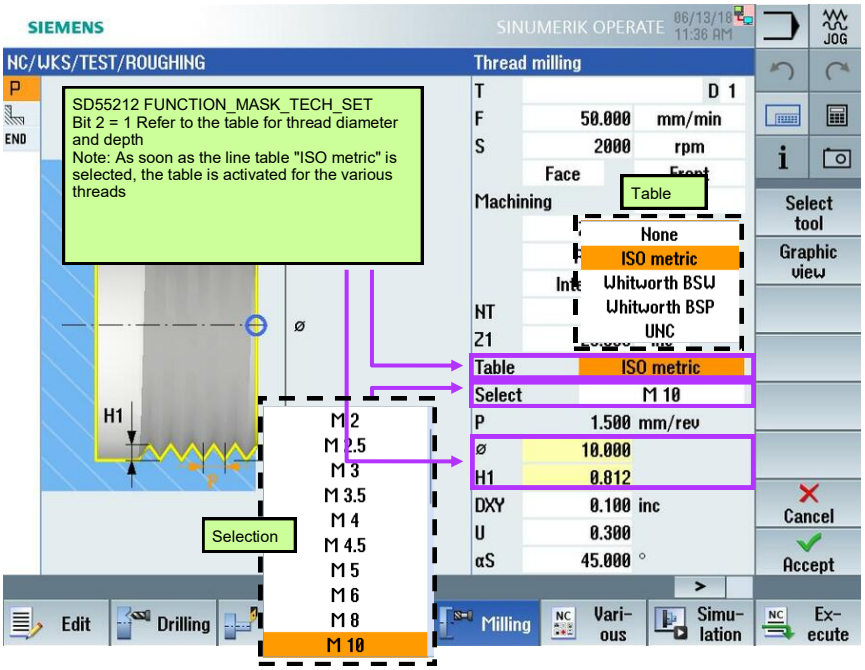


Notes:

Program —> Milling

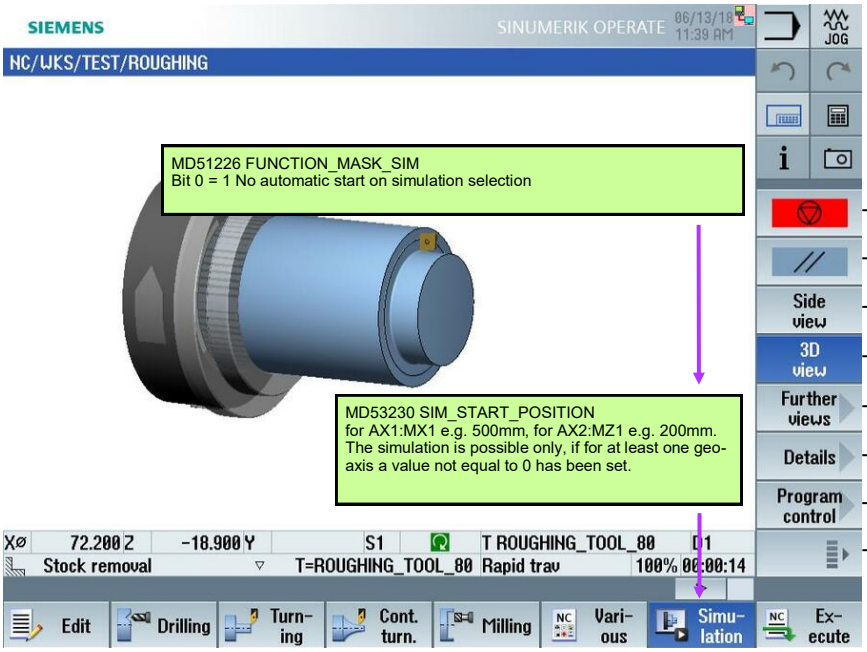


Program —> Milling



Notes:

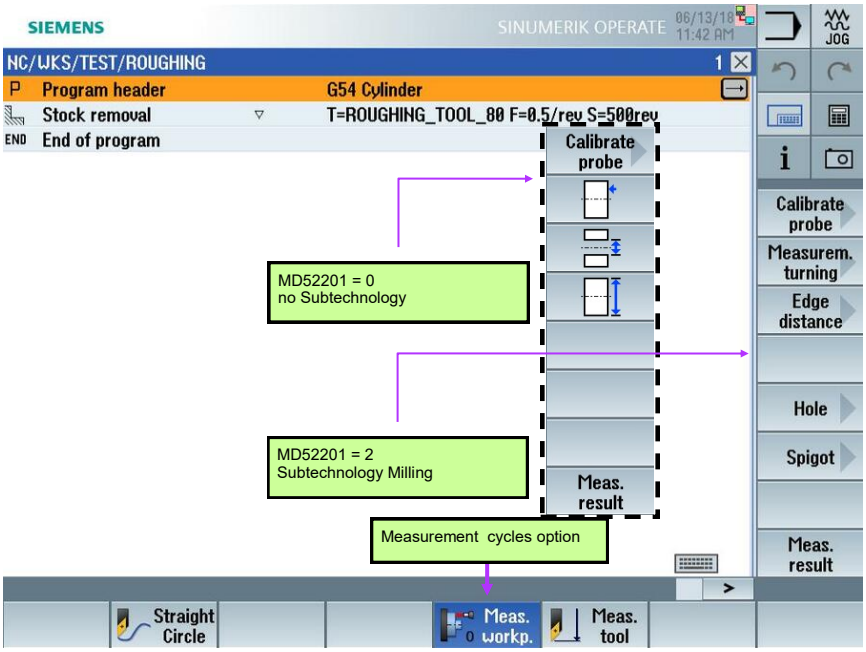
Program —> Simulation



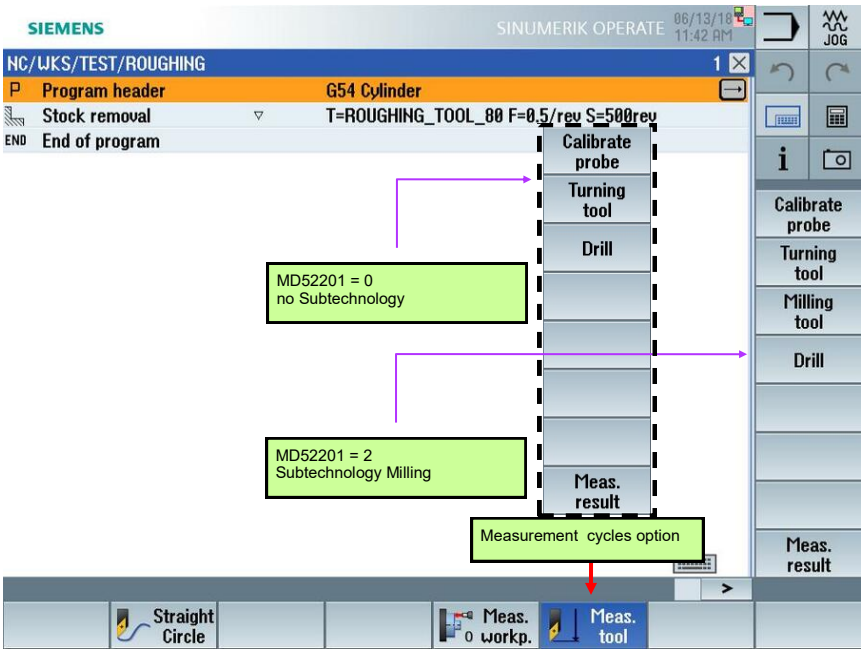
Softkey selection	Description
Option	3D simulation 1 (finished part)

Notes:

Program —> Measure workpiece



Program —> Measure tool



A509: END

Notes:

A511

SINUMERIK ONE

Setup Y-axis

Module Description:

A turning machine with the linear axes X and Z as well as a main spindle C and a driven tool CS3, now an additional Y-axis is to be added and tested.

The way of checking the undertaken parameterization is shown by means of examples and illustrations.

Module Objective:

By means of this module you will learn which MD's must be adapted in order to retrofit a Y-axis to an operative machine and at the end check it's function. In case of the transformations the special aspects regarding the slot-wall correction will be pointed out.

Content:

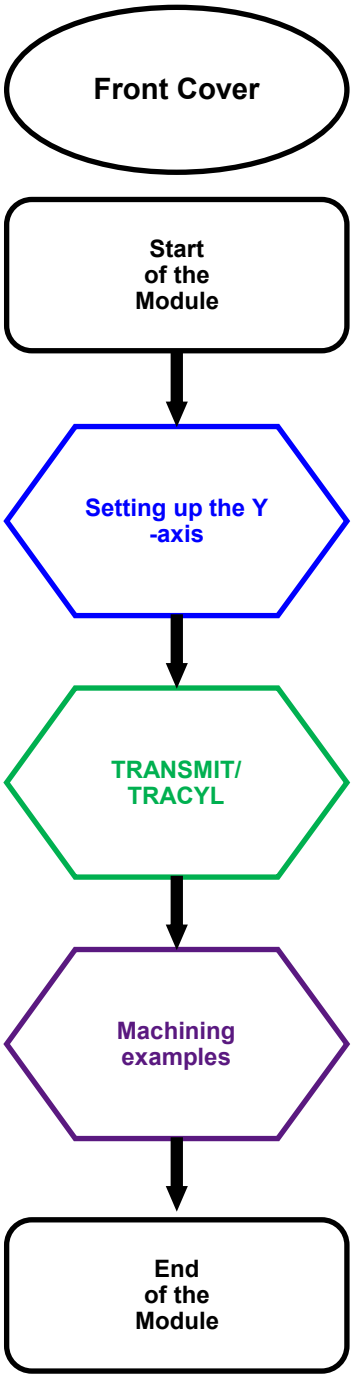
Setting up the Y-axis

TRANSMIT/TRACYL

Machining examples

Sinumerik

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.


Front Cover
Start of the Module
Setting up the Y -axis
TRANSMIT/ TRACYL
Machining examples
End of the Module

Setting up the Y-axis

MD10000[x] $MN_AXCONF_MACHAX_NAME_TAB	Machine axis name
	Change to:

MD10000[0] = MX1
MD10000[1] = MZ1
MD10000[2] = MCS1
MD10000[3] = MCS3
MD10000[4] = MQ1
MD10000[5] = MY1
MD10000[6] = MZS2
MD10000[7] = MCS2

MD20050[x] $MC_AXCONF_GEOAX_ASSIGN_TAB	Assignment of geometry axis to channel axis
	Change to:

MD20050[0] = 1
MD20050[1] = 6
MD20050[2] = 2

New:
MD20050[1] = 6

MD20060[x] $MC_AXCONF_GEOAX_NAME_TAB	Geometry axis to channel
	Change to:

MD20060[0] = X
MD20060[1] = Y
MD20060[2] = Z

MD20070[x] $MC_AXCONF_MACHAX_USED	Machine axis number valid in channel
	Change to:

MD20070[0] = 1
MD20070[1] = 2
MD20070[2] = 3
MD20070[3] = 4
MD20070[4] = 5
MD20070[5] = 6
MD20070[6] = 0
MD20070[7] = 0

New:
MD20070[5] = 6

MD20080[x] $MC_AXCONF_CHANAX_NAME_TAB	Channel axis name in channel
	Change to:

MD20080[0] = X1
MD20080[1] = Z1
MD20080[2] = C
MD20080[3] = CS3
MD20080[4] = Q
MD20080[5] = Y1
MD20080[6] = ZS2
MD20080[7] = CS2

Notes:

Meaning of the axes in the channel

MD52206[x] $MCS_AXIS_USAGE	Meaning of the axes in the channel
	Change to:

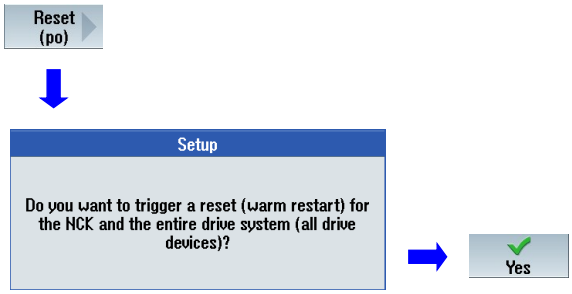
MD52206[0] = 0
MD52206[1] = 0
MD52206[2] = 3
MD52206[3] = 1
MD52206[4] = 0
MD52206[5] = 0
MD52206[6] = 0
MD52206[7] = 0

New:
MD52206[5] = 0

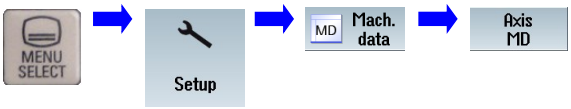
Description:

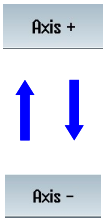
0 = No special meaning
1 = Tool spindle (driven tool)
2 = Extension spindle (driven tool)
3 = Main spindle (Turning)
4 = C-axis of the main spindle (Turning)
5 = Counter spindle (Turning)
6 = C-axis of the counter spindle (Turning)
7 = Linear axis of the counter spindle (Turning)
8 = Tailstock (Turning)
9 = Steady (Turning)
10 = B-axis (Turning)

Validation of the changed MD's



Matching the MD's for the Y-axis





Scrolling the axes with Softkey Axis +/-)

AX6: MY1
MD30130 = 0 *for simulation (command value)*
MD30240 = 0 *Simulation (actual value encoder)*
MD30300 = 0
MD30310 = 0
MD30320 = 0
MD30350 = 1 *for axis simulation*
MD34000 = 0 *axis without a reference point stop*

Text in Italics mark entries for simulated axes.

Some of the undertaken changes will be activated only after renewed Power ON (Softkey “Reset (po))“ valid.



The axis “Y1“ is now also visible in the machine configuration.

All axes, spindles and rotary axes can be traversed in both directions by means of the operating elements.

Notes:

Check of the installed axes and spindles:



Machine configuration

Machine axis Index	Name	Type	Channel axis No.	Name	Geo Axis Name	Channel
1	MX1	Linear	1	X1	X	CHAN1
2	MZ1	Linear	2	Z1	Z	CHAN1
3	MCS1	Spindle S1	3	C		CHAN1
4	MCS3	Spindle S3	4	CS3		CHAN1
5	MQ1	Rotary	5	Q		CHAN1
6	MY1	Linear	6	Y1	Y	CHAN1

The axis MY1 is now available.



Machine	Position [mm]
MX1	0.000
MY1	0.000
MZ1	0.000

The Y-axis is now available.

As a reminder:

MD52010 $MCS_DISP_NUM_AXIS_BIG_FONT is set to 3, hence the 3 geo-axes are shown in large type.

Setting up the oblique Y-axis

Note:

The "Oblique Y-axis“ must be regarded as a special case. In order to be able to use this function, the option “Oblique axis“ is required - Order No. 6FC5800-0BM28-0YB0.
The "Oblique Y-axis“ is not part of this documentation.

Function

If your turning machine features an oblique axis (i.e. an axis that is not perpendicular to the axes X and Z) you can still programme the entire machining operation in Cartesian co-ordinates. The control unit transforms the Cartesian co-ordinates into the motions of the oblique axis using the function oblique axis (TRAANG).

Notes:

TRANSMIT/TRACYL

General aspects

The basic information and machine data settings have already been described in module “Setup Turning”. Only the differences and specialities regarding the transformations TRANSMIT and TRACYL in conjunction with the Y-axis.

Note:

There is a difference between the set-ups of TRANSMIT and TRACYL depending on whether a Y-axis is available or not. Described here is the set-up as applicable when an Y-axis is available.

There are 2 differing sets of data, both of which must be set up; - one for TRANSMIT and one for TRACYL:

- Front face machining main spindle: Transformation 1
  MD 24100 $MC_TRAFO_TYPE_1=257 (with Y–axis)
- Cylinder mantle transformation main spindle: Transformation 3 with slot side wall correction and Y–correction:
  MD 24300 $MC_TRAFO_TYPE_3=514

The following axes are being assumed for the configuration of TRANSMIT/TRACYL:

	Index
20080[0] $MC_AXCONF_CHANAX_NAME_TAB = X1	1
20080[1] $MC_AXCONF_CHANAX_NAME_TAB = Z1	2
20080[2] $MC_AXCONF_CHANAX_NAME_TAB = C	3
20080[3] $MC_AXCONF_CHANAX_NAME_TAB = CS3	4
20080[4] $MC_AXCONF_CHANAX_NAME_TAB = Q	5
20080[5] $MC_AXCONF_CHANAX_NAME_TAB = Y1	6

The tables on the next pages gives an overview of the data blocks for TRANSMIT and TRACYL.

Slot side wall correction

Description:

If the machine features a real Y-axis, an extended TRACYL-variant "Type 514“ can be projected. This permits the machining of slots with side wall correction: Slot side wall and bottom are perpendicular to one another - even if the cutter diameter is less than the slot width. This under other circumstances is possible only if the diameter is the same as the slot width.

The following sketches show the machining of a slot

- without slot side wall correction: By rotation of the C-axis the slot walls are not parallel
- With slot side wall correction: The slot width is obtained by the displacement of the Y-axis - the slot walls are now parallel to one another

The slot side wall correction is available only for

- longitudinal slots
- transverse slots
- slots in any direction (contours)

The transformation of the type 514 in case of a machine with a further linear axis makes use of the redundancy in order to obtain a better tool correction. Applicable for the second linear axis (Y-axis) is now:

- a reduced working range and
- that the second linear axis should not be used for execution of the part-program.

A programming example will be found on the following pages.

Notes:

TRANSMIT/TRACYL

- Front face machining main spindle: Transformation 1 (with Y–axis)
- Cylinder mantle transformation main spindle: Transformation 3 (with slot side face correction and Y-correction)

MD	Identifier	Set-tings	Comment	Description
24100	$MC_TRAFO_TYPE_1	257		Definition of the 1st transformation in the channel
24110	$MC_TRAFO_TYPE_IN_1	[0]=1 (X) [1]=3 (C) [2]=2 (Z) [3]=6 (Y) [4]=0		Axis assignment for transformation 1
24120	$MC_TRAFO_GEOAX_ASSIGN_TAB_1	[0]=1 (X) [1]=3 (C) [2]=2 (Z)		Assignment of geometry axes to channel axes for transformation 1
24300	$MC_TRAFO_TYPE_3	514		Definition of the 3rd transformation in the channel
24310	$MC_TRAFO_TYPE_IN_3	[0]=1 (X) [1]=3 (C) [2]=2 (Z) [3]=6 (Y) [4]=0		Axis assignment for transformation 3
24320	$MC_TRAFO_GEOAX_ASSIGN_TAB_3	[0]=1 (X) [1]=3 (C) [2]=2 (Z)		Assignment of geometry axes to channel axes for transformation 3

Note: After all parameters have been entered a HMI reboot is required for the simulation.

Validation of the changed MD's

After all parameters have been entered they must be validated by means of the Soft-Key “Set MD active (cf)“.



Furthermore a re-boot is required for the simulation and the graphical simulation. This can be achieved either by means of the Softkey "Reboot HMI“ or by switching the machine OFF and back ON again.



Notes:

Machining examples

In the drilling and milling cycles and in the contour descriptions the following machining faces can be selected in case of a real Y-axis:

- Mantle C
- Mantle Y
- Front face C
- Front face Y

Tool list

Tool list — Magazine

Loc.	MT LO.	Type	Tool name	ST	D	Length X	Length Z	Radius			
1			ROUGHING_TOOL_80	1	1	100.000	20.000	1.200	←	93.0	80
2			CUTTER_6_FA	1	1	0.000	100.000	6.000	2		
3			CUTTER_6_MA	1	1	0.000	100.000	6.000	2		
4											

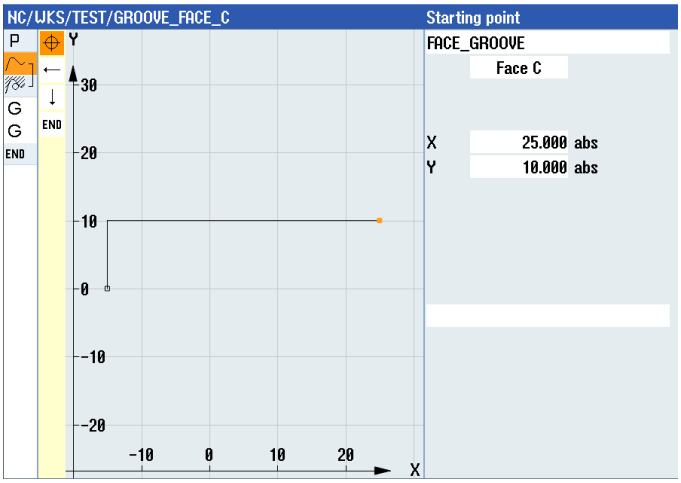
1st Machining example on the front face C

NC/WKS/TEST/GROOVE_FACE_C 1

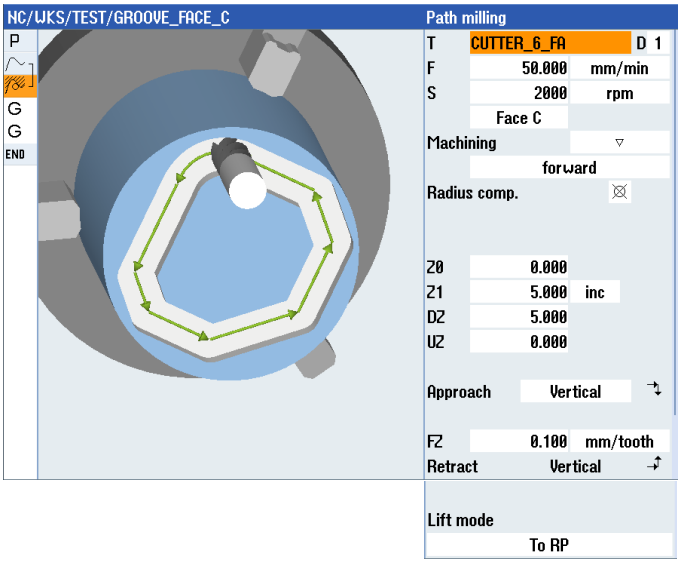
P	Program header	G54 Cylinder
	Contour	FACE_GROOVE
	Path milling	T=CUTTER_6_FA F=50/min S=2000rev Z0=0 Z1=5inc
G	TRAFOOF¶	
G	G0 C0¶	
END	End of program	

Create contour description:

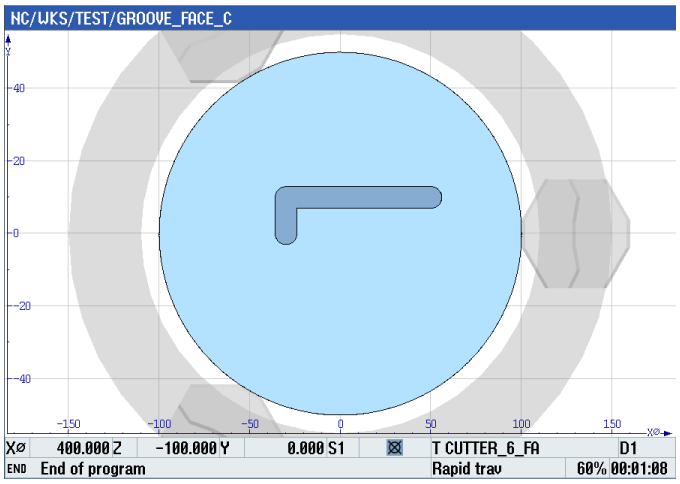
- Front face C, Starting point X=25 abs, Y =10 abs
- Straight X=-15 abs
- Straight Y=0 abs



Continuous path milling on front face C and technology data:



Starting the simulation:



Note:

In order for the depiction of the simulation to correspond to the depiction of the contour programming, the transformation has been deactivated (TRAFOOF) following the continuous path milling and the C-axis positioned into the basic position (G0 C0). Otherwise the depiction would be shown in the last C-axis position to be approached =180 degrees; in other words rotated by 180 degrees.

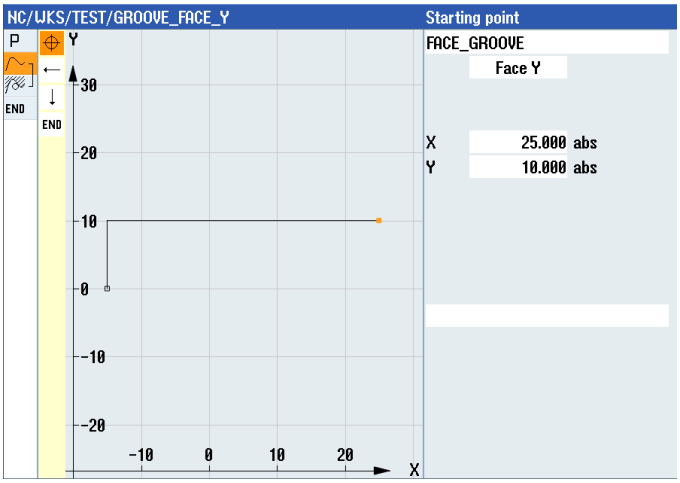
Notes:

2nd machining example front face Y

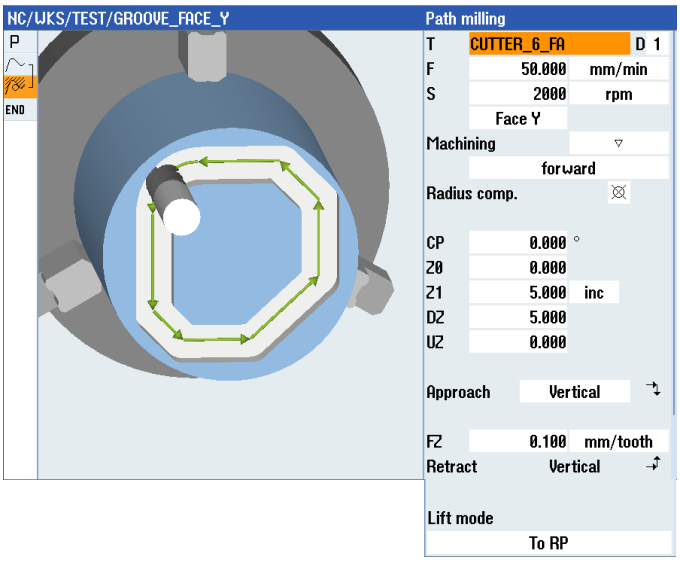


Create contour description:

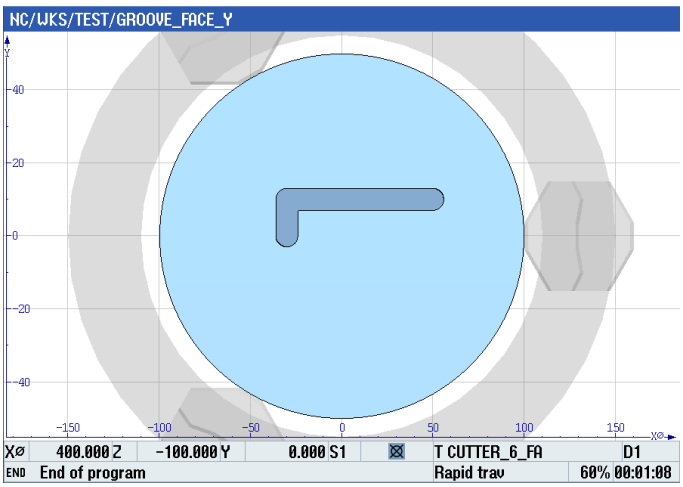
- Front face Y, Starting point X=25 abs, Y =10 abs
- Straight X=-15 abs
- Straight Y=0 abs



Continuous path milling front face Y and technology data:



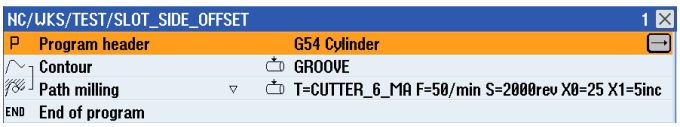
Start simulation:



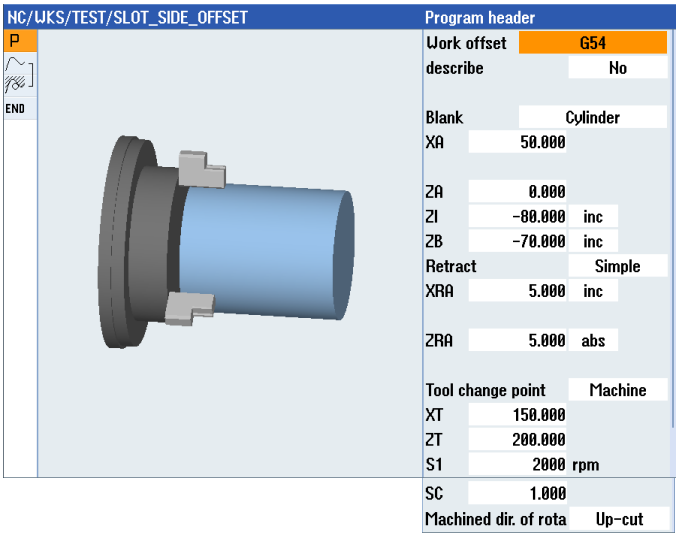
The depiction in the simulation corresponds to the depiction in contour input mask.

Notes:

3rd programming example for mantle face machining with slot side wall correction



Program header with blank definition:



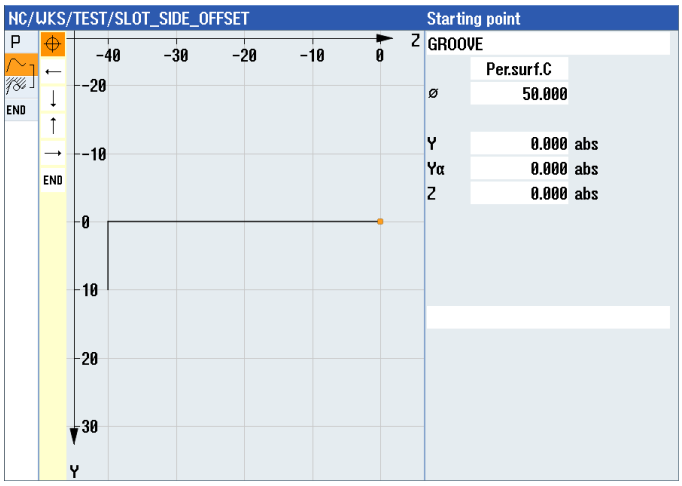
Create contour description:

A slot with parallel walls is to be milled. For this it is not the contour of the slot that is to be programmed, instead it is an imaginative central path of a bolt being moved along the slot that has to be programmed.

- Mantle C, starting point Y=0 abs, Z=10 abs
- Straight Z=-40 abs
- Straight Y=10 abs
- Straight Y=0 abs
- Straight Z=0 abs

For the slot side wall correction it is most important, that the contour is programmed back to the starting point since the slot mill traverses along this contour with the necessary Y-axis offset.

Contour description:

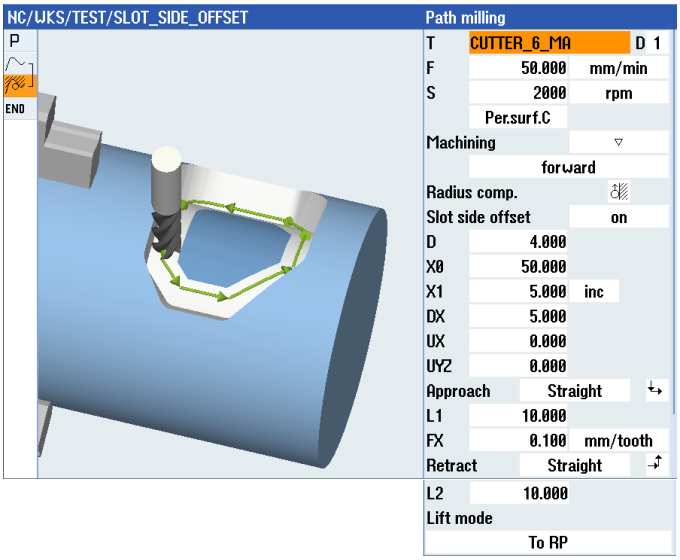


Notes:

Continuous path milling and technology data:

Note:

The slot side wall correction must be used only in conjunction with the radius correction. The input parameter D determines the width of the slot = offset to the programmed path.



Machining sequence:

Tool "CUTTER_6_MA", diameter = 6 mm
Offset to the programmed contour D = 4 mm
Resulting slot width: 8 mm (= 2x D)

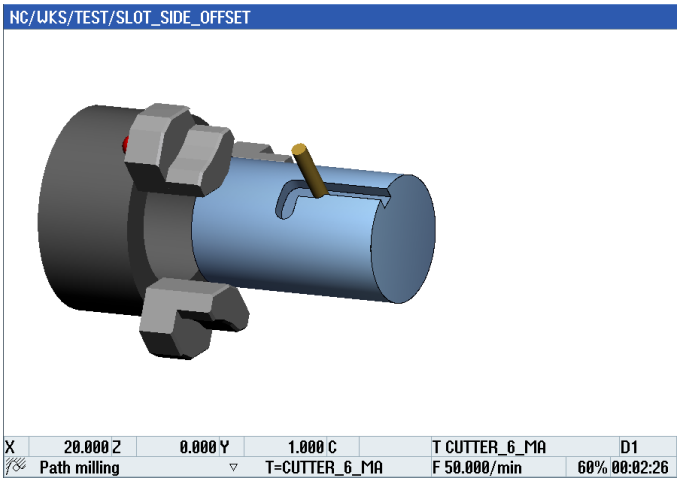
In case of cylinder mantle curve transformation with slot side wall correction the axis used for the correction (Y-axis) should be set to 0 (y=0) such that the slot will be machined centrally to the programmed slot centre line.

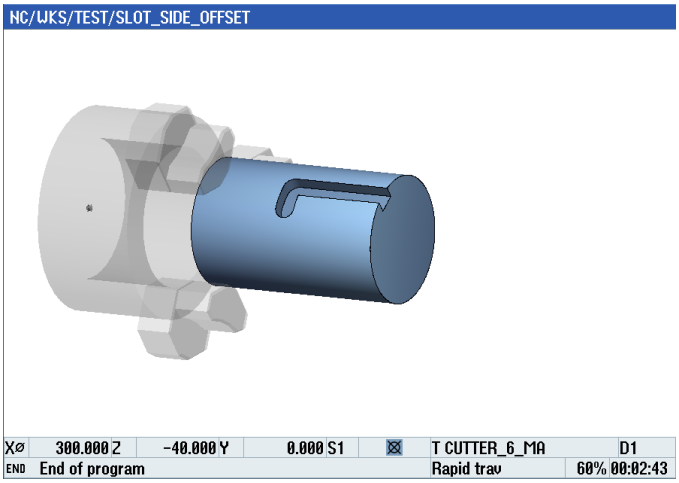
Travelling paths:

The entry into the slot takes place at a position of -2 mm (D - tool diameter), the return path takes place at a Y-position of +2 mm (-D + tool diameter).
The difference in the Y-axis is therefore 4mm. If the slot-mill diameter of 6 mm is added the slot width of 10 mm is obtained.

The simulation below shows the retraction from the slot.

Start simulation:





Note:

In the mask "Path milling" the radius correction "to the left of the contour" is selected. Alternatively it is also possible to select the radius correction "to the right of the contour". The only difference is the direction of traverse in the slot, i.e. when entering the slot the Y-axis is positioned on +2 mm, on the way back on -2 mm. The result on the work piece is identical.

A511: END

Notes:

A513

SINUMERIK ONE

Setup Counter spindle/Tailstock

Module Description:

In this module a counter spindle will first of all be set up and shown as to which settings are required for the axis ZS2 and the spindle CS2. In a subsequent step the technologies TRANSMIT will be set up for the counter spindle and the technology TRACYL for the available axes/spindles. After that follows the configuration for "Traverse to fixed stop" as well as for cutting-off. Lastly the setting up of the counter spindle tailstock will be described.

A simple machining example using the main and the counter spindle, supported by illustrations, will be used to thoroughly check the functionality.

A more complex machining example for facing with the main spindle, followed by circumferential machining using the counter spindle and finally a facing operation with the counter spindle will be used to check the functionality of TRANSMIT and TRACYL. This example, too, will be illustrated by pictures showing the machining progress.

Module Objective:

With the aid of this module you will learn matching of general MD, channel MD and axis MD, in order to extend the functionality of the machine and to equip the counter spindle with the technology TRANSMIT/ TRACYL. The travelling to a fixed stop as well as the cutting-off and the setting-up of a tail stock for the counter spindle complete the functionality of the machine.

Content:

Sinumerik

Counter spindle set-up

TRANSMIT TRACYL for the counter spindle set-up

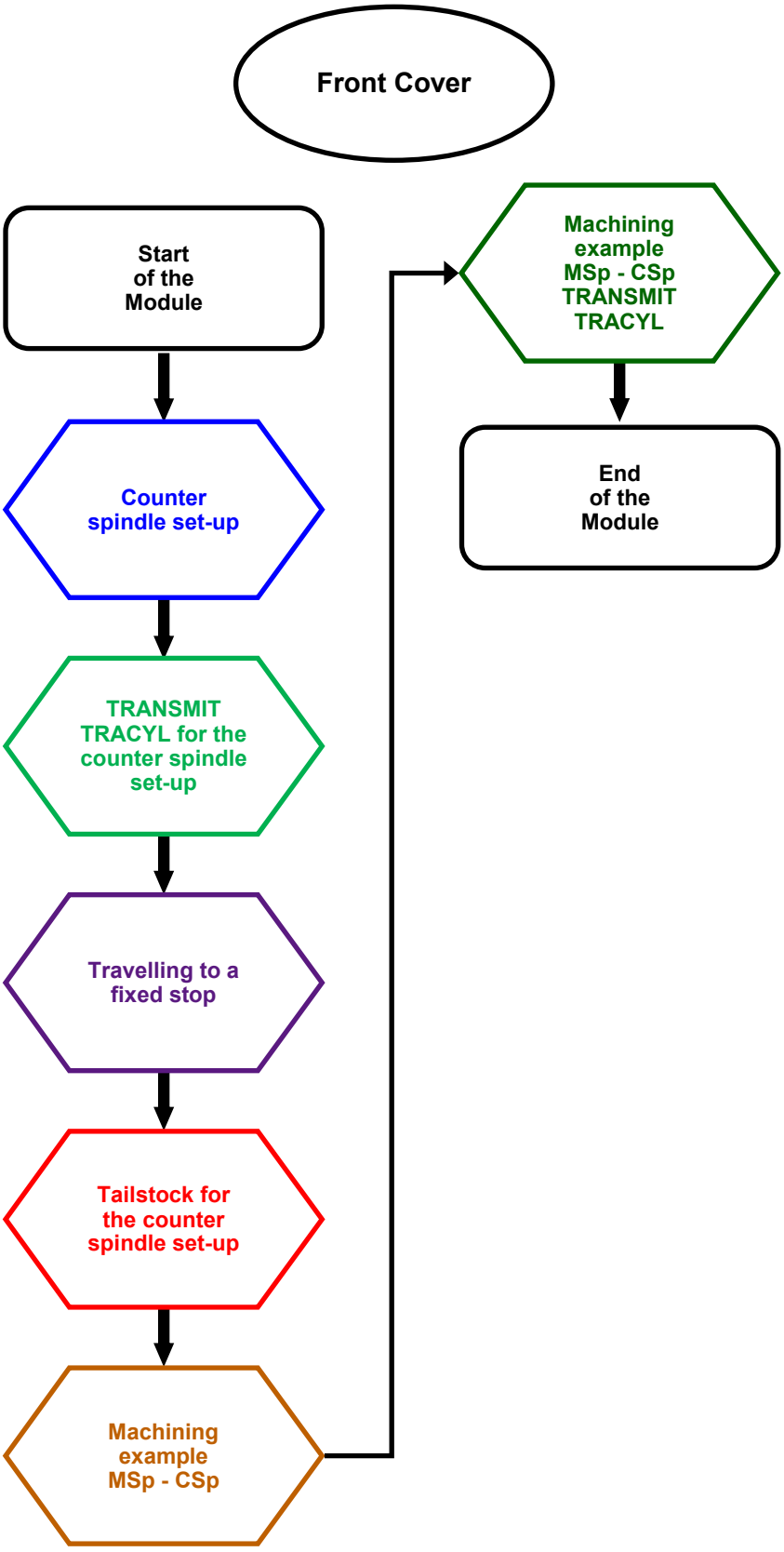
Travelling to a fixed stop

Tailstock for the counter spindle set-up

Machining example MSp - CSp

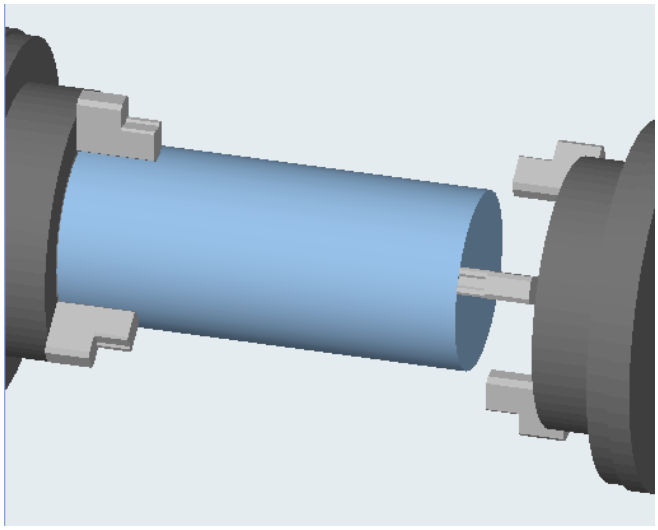
Machining example MSp - CSp TRANSMIT TRACYL

This document was produced for training purposes.
Siemens assumes no responsibility for its contents.

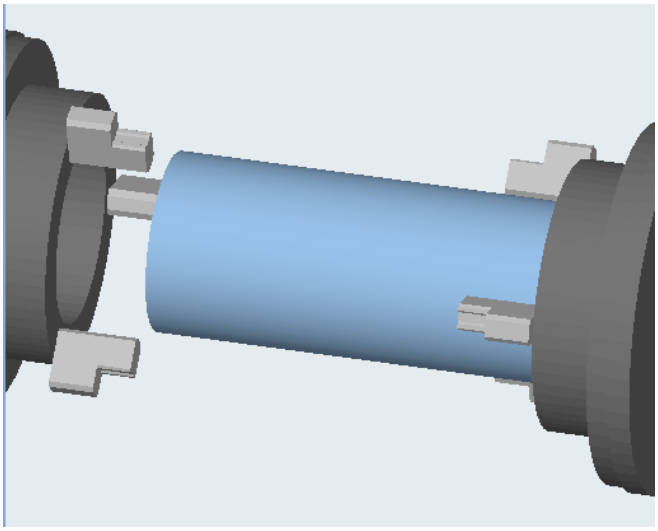

Front Cover
Start of the Module
Counter spindle set-up
TRANSMIT TRACYL for the counter spindle set-up
Travelling to a fixed stop
Tailstock for the counter spindle set-up
Machining example MSp - CSp
Machining example MSp - CSp TRANSMIT TRACYL
End of the Module

Counter spindle set-up

If your machine features a counter spindle, you will be able to machine work pieces using turning, drilling and milling operations on all faces without having to re-load the work piece manually.



Before machining the rear face, the counter spindle must grab the work piece, pull it away from the main spindle and move it to the new machining location.



Setting-up the axis and spindle

Note:

The spindle denominations C2 and C3 are not permissible, since these denominations are already used in the numerical range of MD10620 – MD10640.

If these abbreviations are used, the error message 4020 "Denomination C2 used repeatedly in the machine datum $MN_AXCONF_MACHAX_NAME_TAB“

MD10000[6] = MZS2
MD10000[7] = MCS2

MD20070[6] = 7
MD20070[7] = 8

MD20080[6] = ZS2
MD20080[7] = CS2

Meaning of the axes in the channel

MD52206[x] $MCS_AXIS_USAGE	Meaning of the axes in the channel
	Change to: [6] = 7 [7] = 5

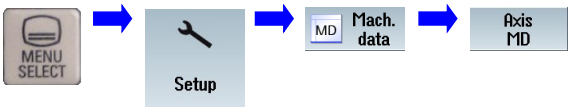
MD52206[6] = 7 Linear axis of the counter spindle ZS2
MD52206[7] = 5 Counter spindle CS2

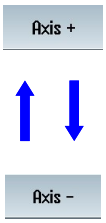
Description:

0 = No special meaning
1 = Tool spindle (driven tool)
2 = Extension spindle (driven tool)
3 = Main spindle (turning)
4 = C-axis of the main spindle (turning)
5 = Counter spindle (turning)
6 = C-axis of the counter spindle (turning)
7 = Linear axis of the counter spindle (turning)
8 = Tailstock (turning)
9 = Steady (turning)
10 = B-axis (turning)
12 = B axis of the counter spindle (turning)
13 = Traverse travel X of the counter spindle (turning)

Notes:

Matching of the axis MD for rotary axis/spindle





(Scrolling of axes using Softkey axis +/-)

AX8: MCS2 (Counter spindle)
MD30130 = 0 *for Simulation (com. value)*
MD30240 = 0 *Simulation (actual value encoder)*
MD30300 = 1 axis is rotary axis
MD30310 = 1 Modulo transformation for rotary axis
MD30320 = 1 actual value readout Modulo
MD30350 = 1 *for axis simulation*
MD34000 = 0 *axis without reference cam*
MD35000 = 2 Assignment spindle to machine axis
(spindle S2)

AX7: MZ2 (Linear axis of the counter spindle)
MD30130 = 0 *for simulation (command value)*
MD30240 = 0 *Simulation (actual value Encoder)*
MD30300 = 0
MD30310 = 0
MD30320 = 0
MD30350 = 1 *for axis simulation*
MD34000 = 0 *axis without reference cam*

Text in italics shows entries for simulated axes.

Note:

In the axis MD's 32000, 32010, 32020, 35100, 35110, 35120, 35130, 36200 the speeds of the - possibly - available gearbox steps and their minimum and maximum limits must be entered depending on the machine execution.

Check of the installed axes and spindles:



Machine configuration

Machine axis			Channel axis		Geo Axis	
Index	Name	Type	No.	Name	Name	Channel
1	MX1	Linear	1	X1	X	CHAN1
2	MZ1	Linear	2	Z1	Z	CHAN1
3	MCS1	Spindle S1	3	C		CHAN1
4	MCS3	Spindle S3	4	CS3		CHAN1
5	MQ1	Rotary	5	Q		CHAN1
6	MY1	Linear	6	Y1	Y	CHAN1
7	MZS2	Linear	7	ZS2		CHAN1
8	MCS2	Spindle S2	8	CS2		CHAN1

The axis MZS2 and the spindle MCS2 is now available.



Machine	Position [mm]
MX1	0.000
MY1	0.000
MZ1	0.000
MZS2	0.000

The ZS2-axis is now available.

As a reminder:

MD52010 $MCS_DISP_NUM_AXIS_BIG_FONT is set to 3, hence the 3 geo-axes are shown in large type.

Notes:

Spindle

These settings are required to enable spindles to be started from the T,S,M mask.

M function for spindle active after reset

MD10714 $MN_M_NO_FCT_EOP	M function for spindle active after reset
	Change to: 32

Suggestion M32

Description:

With Start in the manual mode-of-operation (T,S,M-mask, position) the part-program MA_JOG_STEP1 will be generated and executed.
This part-program will be finished with the M-function in MD10714. The setting of MD35040 = 2 ensures that the spindle remains active also after the end-of-program.

Limitations see MD10715 $MN_M_NO_FCT_CYCLE

Own spindle RESET

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET	Own spindle RESET
	Change to: 2

AX3: MCS1
MD35040 = 2 Own spindle-RESET (Main spindle)

AX4: MCS3
MD35040 = 2 Own spindle-RESET (Tool spindle)

AX8: MCS2
MD35040 = 2 Own spindle reset (Counter spindle)

Description:

This machine data defines the response of the spindle after Reset and program end (M2, M30).

This MD is only active in the spindle mode open-loop control mode. In the positioning or oscillation mode, the spindle is always stopped.

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET = 0:

- Spindle stops (with M2/M30 and channel and mode group reset)
- Program is aborted

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET= 1:

- Spindle does not stop
- Program is aborted

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET= 2:

- Spindle does not stop at the M function configured via MD10714 $MN_M_NO_FCT_EOP (e.g. M32).
- However, the spindle stops at channel or mode group reset.

Not relevant to:

- Spindle modes other than open-loop control mode.

Notes:

Travelling direction

The following settings must be made to inform the user interface and the cycles in which direction the spindles and rotary axes are rotating.
The direction of rotation of an NC-rotary axis is specified via the MD32100 $MA_AX_MOTION_DIR.

These settings are:

Specific to machine manufacturers.

MD32100 $MA_AX_MOTION_DIR	Traversing direction (not control direction)
Default: 1	Change to:

0, 1: Direction reversal
-1: No direction reversal

Description:

By means of this MD the direction of motion of the machine can be reversed. The conventions, however, will not be lifted in this case; i.e. the rules will be upheld.

Note:

For SINAMICS-Drives it is recommended to utilise the direction reversal in the drive. See P1821 under the control Unit MD (Drive MD).

With reference to the work piece the main and the counter spindle must have the same direction-of-rotation and speed in order to enable the transfer of the work piece from the main to the counter spindle under synchronous running conditions.

The machine manufacturer cycle CUST_TECHCYC.SPF is called up in the ShopTurn cycles.
Here the default for the coupling of the main and counter spindle can be changed Mark _M120.

Specifying the direction of rotation

MD52207[x] $MCS_AXIS_USAGE_ATTRIB	Axis attributes
	Change to: Bit 3 = 1 Bit 4 = 1 Bit 5 = 0

MD52207[7] CS2

Bit 3 = 1: C+ Direction of rotation is counter-clockwise (for rotary axes).

Bit 4 = 1: M3 Indicated direction of rotation for M3 is counter -clockwise (for spindles)

Bit 5 = 0: M3 Direction of rotation M3 means rotary axis minus (for spindles).

Note:

If there is a separate C-axis drive, the entry for Bit 3 must be made in the data set of that drive.

As a reminder: Here once more the axis indexes

MD52206[6] = 7 Linear axis of the counter spindle ZS2
MD52206[7] = 5 Counter spindle CS2

Note:

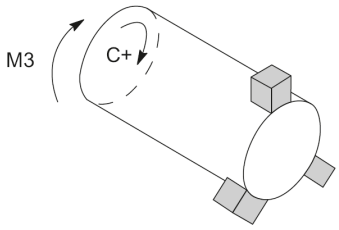
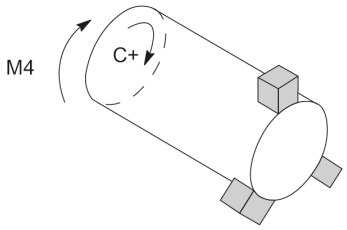
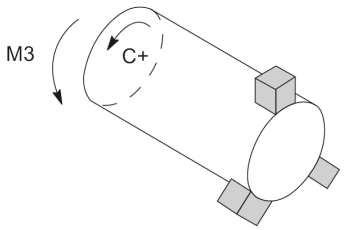
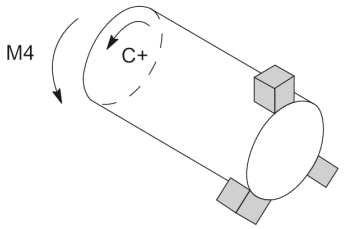
In order to be able to use the counter spindle on your machine, the following software-options will be required:

- "Travelling to a fixed stop with Force Control"
  Order No. 6FC5800-0BM01-0YB0
  Is required if cutting-off in the counter spindle is envisaged.
- “Generic coupling "CP-BASIC“
  Order No. 6FC5800-0BM72-0YB0

Notes:

From this derive the following setting possibilities for the counter spindle. The settings of the machine data, however, depend on the direction of viewing the co-ordinate axis.

No matter as to which of the four combinations of MD52207[7] Bit 3 und Bit 4 shown below the machine manufacturer selects, it is imperative that the settings shown in one line for PLC interface signal “Invert M3/M4” and MD52207[7] Bit 5 are used.

Direction of rotation counter spindle	52207[7] CS2 Bit 3	52207[7] CS2 Bit 4	PLC bit CS2 “Invert M3/M4“	52207[7] CS2 Bit 5
	**1**	**1**	**0**	**0**
	1	0	1	1
	0	0	0	0
	0	1	1	1

The setting of the machine datum MD52207 Bit 3 assumes that the direction of viewing is in direction of the negative Z-axis . If, however, you are looking in the positive direction of the co-ordinate axis, the values must be reversed; i.e. "0" and "1" must be interchanged.

Note:

The MD52207[7] of the counter spindle affects

- the readout of the spindle direction in the user interface
- the depiction in the simulation and for sketching
- the direction of rotation of the spindle after a tool change according to the direction of rotation from the tool.

Notes:

TRANSMIT and TRACYL for the counter spindle set-up

General aspects

The basic information and machine data settings have already been described in module “Setup Turning”. Only the differences and specialities regarding the transformations TRANSMIT and TRACYL in conjunction with the counter spindle.

Note:

There is a difference between the set-ups of TRANSMIT and TRACYL depending on whether a Y-axis is available or not. Described here is the set-up as applicable when an Y-axis is available.

There are 2 differing sets of data, both of which must be set up; - one for TRANSMIT and one for TRACYL:

- Front face machining counter spindle: Transformation 2
  MD 24200 $MC_TRAFO_TYPE_2=257 (with Y–axis)
- Cylinder mantle transformation counter spindle: Transformation 4 With slot side face correction and Y–correction:
  MD 24400 $MC_TRAFO_TYPE_4=514

The tables on the next pages gives an overview of the data blocks for TRANSMIT and TRACYL.

Note:

With the function TRANSMIT/TRACYL a non-existent Y-axis will be simulated by an interpolation between X/C (in case of TRANSMIT) or Z/C (in case of TRACYL). In this configuration example there is beside the counter spindle no separate C-axis as feed drive for the C-axis mode-of-operation. For this reason the channel index of the counter spindle must be entered for the CS2-axis into the configuration data for TRANSMIT/TRACYL..

As a reminder: the machine is already configured thus:

	Index
20080[0] $MC_AXCONF_CHANAX_NAME_TAB = X1	1
20080[1] $MC_AXCONF_CHANAX_NAME_TAB = Z1	2
20080[2] $MC_AXCONF_CHANAX_NAME_TAB = C	3
20080[3] $MC_AXCONF_CHANAX_NAME_TAB = CS3	4
20080[4] $MC_AXCONF_CHANAX_NAME_TAB = Q	5
20080[5] $MC_AXCONF_CHANAX_NAME_TAB = Y1	6
20080[6] $MC_AXCONF_CHANAX_NAME_TAB = ZS2	7
20080[7] $MC_AXCONF_CHANAX_NAME_TAB = CS2	8

These MD's were already configured in the module “Setup Turning” in section TRANSMIT/TRACYL for the main spindle.

MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 1
MD24040 $MC_FRAME_ADAPT_MODE = 7H

The sets of data for TRANSMIT and TRACYL described below are as required for the horizontal lathe with machining behind the turning centre as configured here and the axis configuration as described above. Nevertheless the conditions on the machine might require matching of this set of data, such as for instance the direction of rotation of the C-axis in case of the transformation or the input of an offset.

Notes:

Front face machining (TRANSMIT)

- Front face machining counter spindle: Transformation 2 (with Y–axis)

MD	Identifier	Settings	Comment	Description
24200	$MC_TRAFO_TYPE_2	257		Definition of the 2nd transformation in the channel
24210	$MC_TRAFO_TYPE_IN_2	[0]=1 (X) [1]=8 (CS2) [2]=2 (Z) [3]=6 (Y) [4]=0		Axis assignment for transformation 2
24220	$MC_TRAFO_GEOAX_ASSIGN_TAB_2	[0]=1 (X) [1]=8 (CS2) [2]=2 (Z)		Assignment of geometry axes to channel axes for transformation 2
24950	$MC_TRANSMIT_ROT_AX_OFFSET_2	0		Offset of rotary axis for the 2nd TRANSMIT transformation
24955	$MC_TRANSMIT_ROT_AX_FRAME_2	1		Rotary axis offset TRANSMIT 2
24960	$MC_TRANSMIT_ROT_SIGN_IS_PLUS_2	0		Sign of rotary axis for 2nd TRANSMIT transformation
24961	$MC_TRANSMIT_POLE_SIDE_FIX_2	1		Restriction of working range in front of / behind the pole, 2. TRANSMIT
24970	$MC_TRANSMIT_BASE_TOOL_2	0		Vector of base tool for 2nd TRANSMIT transformation

Note: After all parameters have been entered a HMI reboot is required for the simulation.

Front face machining (TRANSMIT)

Offset of rotary axis for the 2nd TRANSMIT transformation

MD24950 $MC_TRANSMIT_ROT_AX_OFFSET_2	Offset of rotary axis for the 2nd TRANSMIT transformation
Default: 0 Grad	Change to:

Description:

States the offset of the rotary axis in degrees in relation to the zero position while TRANSMIT is active for the second agreed TRANSMIT-transformation.

Corresponds to:
MD24900 $MC_TRANSMIT_ROT_AX_OFFSET_1

Rotary axis offset TRANSMIT 2

MD24955 $MC_TRANSMIT_ROT_AX_FRAME_2	Rotary axis offset TRANSMIT 2
	Change to: 1

Description:

0: axial rotary axis offset is not considered

1: axial rotary axis offset is considered

2: axial rotary axis offset is considered until SZS.

The SZS-frames include transformed rotations around the rotary axis.

Note:

This setting is necessary in conjunction with JobShop.

Notes:

Sign of rotary axis for 2nd TRANSMIT transformation

MD24960 $MC_TRANSMIT_ROT_ SIGN_IS_PLUS_2	Sign of rotary axis for 2nd TRANSMIT transformation
	Change to: 0

Description:

Indicates the sign with which the rotary axis is taken into account in the TRACYL transformation for the 2nd agreed TRACYL transformation for each channel. The sign of the virtual Y-axis is meant in this context.

Corresponds to:
MD24910 $MC_TRANSMIT_ROT_SIGN_IS_PLUS_1

Note:

Match to the machine conditions.

Restriction of working range before/behind the pole, 2. TRANSMIT

MD24961 $MC_TRANSMIT_POLE_ SIDE_FIX_2	Restriction of working range before/behind the pole, 2. TRANSMIT
	Change to: 1

Description:

Restriction of the working range before/behind the pole or else no restriction; i.e. travelling through the pole.

The assigned values have the following meaning:

1: Working area of linear axis for positions >= 0,
(if tool length correction parallel to the linear axis = 0)

2: Working area of linear axis for positions <=0,
(if tool length correction parallel to the linear axis = 0)

0: No restriction of working area. Traversal through pole.

Vector of base tool for 2nd TRANSMIT transformation

MD24970[x] $MC_TRANSMIT_BASE_ TOOL_2	Vector of base tool for 2nd TRANSMIT transformation
Default: 0, 0, 0 mm	Change to:

MD24970[0] = 0
MD24970[1] = 0
MD24970[2] = 0

Description:

States a basic offset for the tool zero point offset for the 2nd TRANSMIT-transformation. This MD will only be required. For the standard application the default is applicable.

The index i takes the values 0, 1, 2 for the 1st to the 3rd geometry axis.

Corresponds to:
MD24920 $MC_TRANSMIT_BASE_TOOL_1

Notes:

Cylinder mantle transformation (TRACYL)

- Cylinder mantle transformation main spindle: Transformation 4 (with slot side face correction and Y-correction)

MD	Identifier	Settings	Comment	Description
24400	$MC_TRAFO_TYPE_4	514		Definition of the 4th transformation in the channel
24410	$MC_TRAFO_TYPE_IN_4	[0]=1 (X) [1]=8 (CS2) [2]=2 (Z) [3]=6 (Y) [4]=0		Axis assignment for transformation 4
24420	$MC_TRAFO_GEOAX_ASSIGN_TAB_4	[0]=1 (X) [1]=8 (CS2) [2]=2 (Z)		Assignment of geometry axes to channel axes for transformation 4
24850	$MC_TRACYL_ROT_AX_OFFSET_2	0		Offset of rotary axis for the 2nd TRACYL transformation
24855	$MC_TRACYL_ROT_AX_FRAME_2	1		Rotary axis offset TRACYL 2
24860	$MC_TRACYL_ROT_SIGN_IS_PLUS_2	1		Sign of rotary axis for 2nd TRACYL transformation
24870	$MC_TRACYL_BASE_TOOL_2	0		Vector of base tool for 2nd TRACYL transformation

Note: After all parameters have been entered a HMI reboot is required for the simulation.

Mantle machining with TRACYL for the counter spindle

Offset of rotary axis for the 2nd TRACYL transformation

MD24850 $MC_TRACYL_ROT_AX_ OFFSET_2	Offset of rotary axis for the 2nd TRACYL transformation
Default: 0 Grad	Change to:

Description:

Indicates the offset of the rotary axis in degrees in relation to the neutral position for the 2nd agreed TRACYL transformation for each channel.

Irrelevant:
if no TRACYL is active

Rotary axis offset TRACYL 2

MD24855 $MC_TRACYL_ROT_AX_ FRAME_2	Rotary axis offset TRACYL 2
	Change to: 1

Description:

0: axial rotary axis offset is not considered.

1: axial rotary axis offset is considered.

2: axial rotary axis offset is considered until SZS.

The SZS frames include transformed axial rotary axis offsets.

Notes:

Sign of rotary axis for 2nd TRACYL transformation

MD24860 $MC_TRACYL_ROT_SIGN_ IS_PLUS_2	Sign of rotary axis for 2nd TRACYL transformation
	Change to: 1

Description:

Indicates the sign with which the rotary axis is taken into account in the TRACYL transformation for the 2nd agreed TRACYL transformation for each channel.

Corresponds to:
MD24810 $MC_TRACYL_ROT_SIGN_IS_PLUS_1

Note:

Match to the machine conditions.

Vector of base tool for 2nd TRACYL transformation

MD24870[x] $MC_TRACYL_BASE_ TOOL_2	Vector of base tool for 2nd TRACYL transformation
Default: 0, 0, 0 mm	Change to:

MD24870[0] = 0
MD24870[1] = 0
MD24870[2] = 0

Description:

Indicates a basic offset of the tools zero for the 2ndTRACYL transformation. The offset is referenced to the geometry axes valid when TRACYL is active. The basic offset is included with and without selection of the tool length compensation. Programmed length corrections have an additive effect with respect to the basic tool.
The index i takes the values 0, 1, 2 for the 1st to 3rd geometry axes.

Corresponds to:
MD24820 $MC_TRACYL_BASE_TOOL_1

Activation of the undertaken alterations

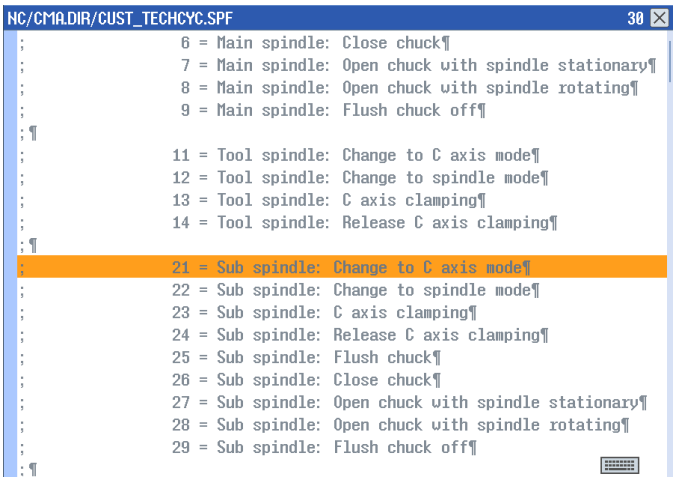
After all parameters have been entered and validated by means of the softkey “Set MD active (cf)“ one further step is necessary.



Turn the main switch OFF once and turn it back ON again to cause the control to re-boot. By this the control unit is matched to the undertaken alterations.

Switching between axis mode and clamping

Extract from the CUST_TECHCYC.SPF



Notes:

Continue: Switching between axis mode and clamping

```
NC/CMA.DIR/CUST_TECHCYC.SPF                                    297
_M21 ... _M29: Sub spindle
_M21: Sub spindle: Change to C axis mode
  _M21: ;¶
  ; Spindle is not in C axis mode or after block search¶
  IF(($P_SMODE[_F_S_NR[2]]<>2)AND($P_SMODE[_F_S_NR[2]]<>4))OR($P_PROG
_EVENT==5)¶
    N210 SPOS[_F_S_NR[2]]=0       ; Spindle in C axis mode¶
  ENDIF¶
  GOTOF _MEND¶
Ende Block
  ;¶
_M22: Sub spindle: Change to spindle mode
  _M22: ;¶
  IF($P_SMODE[_F_S_NR[2]]<>1); Spindle is not is spindle mode¶
    ;N220 M[_F_S_NR[2]]=          ; Change to spindle mode¶
  ENDIF¶
  GOTOF _MEND¶
Ende Block
```

```
  ;¶
_M23: Sub spindle: C axis clamping
  _M23: ; Attention: The spindle must stay in C axis mode! Do not use
 M5!¶
  ;N230 M[_F_S_NR[2]]=            ; C axis clamping¶
  ;N235 $AC_PRTIME_A_INC=1.5      ; increase auxillary time by 1.5 s
econds¶
  GOTOF _MEND¶
Ende Block
  ;¶
_M24: Sub spindle: Release C axis clamping
  _M24: ;¶
  ;N240 M[_F_S_NR[2]]=            ; Release C axis clamping¶
  ;N245 $AC_PRTIME_A_INC=1.5      ; increase auxillary time by 1.5 s
econds¶
  GOTOF _MEND¶
Ende Block
```

Description:

For the switching of the counter spindle into the C-axis mode-of-operation and for clamping or unclamping of the counter spindle the following jumps into the manufacturer cycle CUST_TECHCYC.SPF have been prepared:

_M21: Switching of the counter spindle into the C-axis mode-of-operation.
_M22: Switching of the C-axis mode to the spindle mode
_M23: Counter spindle clamping
_M24: Counter spindle release clamping

Description:

_M21: Switching to the C-axis mode
When switching to the C-axis mode there is a difference, whether the C-axis mode is carried out with the drive of the counter spindle or by means of a separate feed drive.

1.No separate feed drive fitted:
The following entry is provided :
IF (($P_SMODE[_F_S_NR[2]]<>2)AND($P_SMODE [_F_S_NR[2]]<>4))OR($P_PROG_EVENT==5)

Variable _F_S_NR: The spindle index of the main, the tool and the counter spindle.
In order to enable the CUST_TECHCYC-draft to be generally valid for all applications the spindle index is determined by means of the variable _F_S_NR from the existing axis configurations (setting via MD52206 $MCS_AXIS_USAGE.
The following applies:
_F_S_NR[0] contains the spindle index of the main spindle
_F_S_NR[1] contains spindle index of the tool spindle
_F_S_NR[2] contains spindle index of the counter spindle

From this derives the following logic:
The IF-query is fulfilled, if the counter spindle is not in the mode-of-operation 2 = positioning and not in the mode-of-operation 4 = axis mode or the program event is in the mode 5 = block search.

The following line positions the counter spindle to 0 degrees:
SPOS[_F_S_NR[2]]=0
This query ensures that the counter spindle is not positioned again if it was switched into the axis mode beforehand.

2. Separate feed drive fitted:
The IF-query described above and the subsequent ENDIF-command must be replaced by the logic for the coupling of the separate feed drive. This should preferably be effected in a separate manufacturer cycle, e.g. AXIS_CS2_ON.SPF.
_M21: Sub spindle: Change to C axis mode
  AXIS_CS2_ON
GOTOF _MEND

Example for the manufacturer cycle AXIS_CS2_ON.SPF:
PROC AXIS_CS2_ON
IF $P_SEARCH==1 GOTOF _END
IF $P_SIM==1 GOTOF _END
; Cycle not to be carried out during block search and
; simulation
IF $A_IN[24]==1
;Cycle not to be carried out on report by the PLC, that the feed drive is already coupled
GOTOF _END
ENDIF
M[_F_S_NR[2]]=41 ;Select counter spindle gear stage 1
SPOSA[[_F_S_NR[2]]=0 ;Positioning of counter spindle
WAITS(1) ;Wait until spindle position is reached

Notes:

Continuation of extract from CUST_TECHCYC.SPF

```
STOPRE
G0  CS2=0     ;CS2-axis positioning
STOPRE
M66           ;M-command for coupling the CS2-axis
MARK1:
IF $A_IN[24]==1 GOTOF MARK2
;Query whether coupling has taken place
GOTOB MARK1 ;     Jump back until coupling is completed
MARK2:
 _END:
IF $P_SIM==1          ;Simulation active
M[_F_S_NR[2]]=70   ;Switch counter spindle to the axis mode
ENDIF
M17
```

_M22: Switching the counter spindle to the spindle mode:

1. no separate feed drive :

The following entry has been prepared:

```
IF($SP_MODE[_F_S_NR[2]]<>1
  ;N220 M[_F_S_NR[2]]=
ENDIF
```

Line N220 can be extended by „5“. The counter spindle will be switched back to the spindle mode with the command M5

2. Separate feed drive fitted:

As described in the previous mode a further manufacturer cycle for un-coupling the separate feed drive is used: AXIS_CS2_OFF.SPF and this is called up in _M22.

```
_M22: ; Sub spindle: Change to spindle mode
  AXIS_CS2_OFF
GOTOF _MEND
```

Example for manufacturer cycle AXIS_CS2_OFF.SPF:

```
PROC AXIS_CS2_OFF
IF $P_SEARCH==1 GOTOF _END
IF $P_SIM==1 GOTOF _END
; Cycle not to be carried out with block search and simulation
M67           ;M-command for un-coupling the CS2-axis
MARK1:
IF $A_IN[24]==0 GOTOF MARK2
;Query, whether un-coupling is completed
GOTOB MARK1
MARK2:
ENDIF
_END:
M17
```

_M23: Sub spindle: C axis clamping

The following entry has been prepared:

```
  ;N230 M[_F_S_NR[2]]=
  ;N235 $AC_PRTIME_A_INC=1.5
```

In line N230 enter the M-command for clamping the counter spindle. Line N235 has been prepared for matching the time evaluated in the simulation to the real machining time.

_M24: Sub spindle: Release C axis clamping

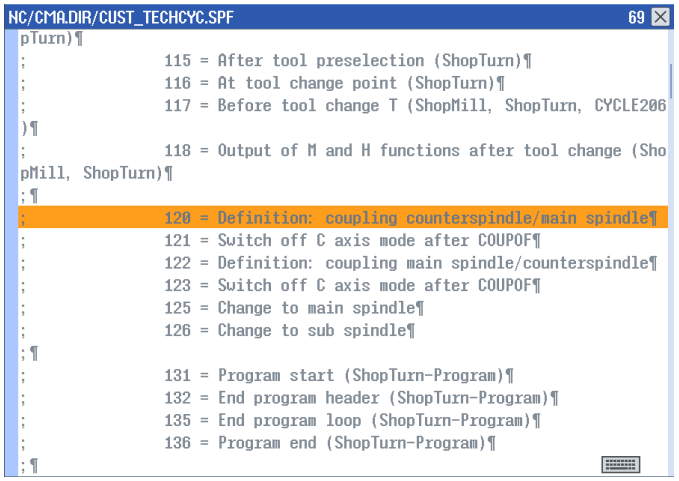
The following entry has been prepared:

```
  ;N240 M[_F_S_NR[2]]=
  ;N245 $AC_PRTIME_A_INC=1.5
```

In line N240 enter the M-command for clamping the counter spindle. Line N235 has been prepared for matching the time evaluated in the simulation to the real machining time.

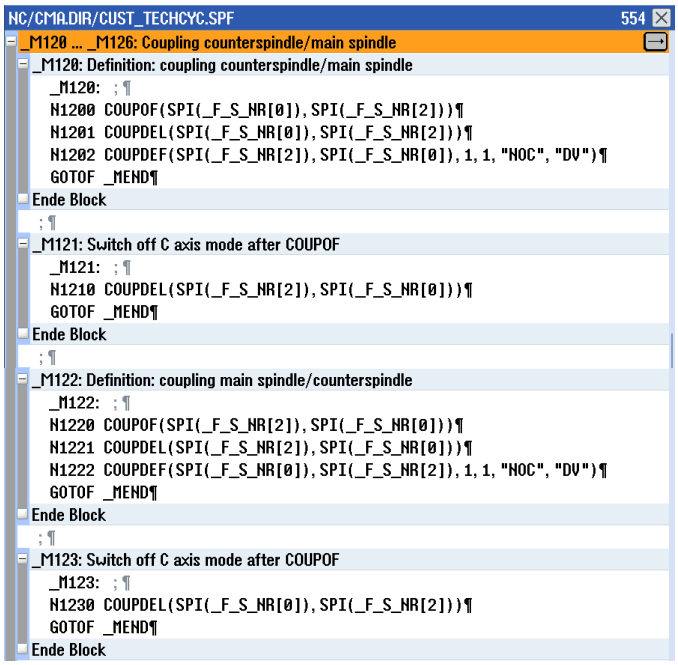
Coupling

Extract from CUST_TECHCYC.SPF



Notes:

Continued: Coupling



Description:

Definition of coupling counter spindle/main spindle

_M120: Definition of the coupling of spindle [2], spindle [0]

_M121: End of C-axis mode-of-operation after COUPOF

_M122: Definition of the coupling spindle [0], spindle [2]

_M123: Switch of C axis mode after CUPOF

_M125: Change to main spindle

_M126: Change to counter spindle

_M120: Definition: Coupling counter spindle/main spindle
N1200 COUPOF(SPI(_F_S_NR[0]),SPI(_F_S_NR[2]))
N1201 COUPDEL(SPI(_F_S_NR[0]),SPI(_F_S_NR[2]))
N1202 COUPDEF(SPI(_F_S_NR[2]),SPI(_F_S_NR[0]),1,1,
”NOC”,”DV”)
The prepared COUPDEF-command for coupling the main and counter spindle can be made to suit by the machine manufacturer, for example to change the direction of rotation between the two spindles.

_M125 and _M126:
No entries prepared here.

Chuck

Machine data for definition of the M-commands

M code for Open chuck with non-rotating spindle

MD52250[x] $MCS_M_CODE_CHUCK_OPEN	M code for opening chuck with non-rotating spindle
Default:	Change to:

MD52250[0] Main spindle
MD52250[1] Counter spindle

Description:

M-Code for opening chuck with non-rotating spindle.
e.g.: "M34" or "M1=34"

M code for Open chuck with rotating spindle

MD52251[x] $MCS_M_CODE_CHUCK_OPEN_ROT	M code for opening chuck with rotating spindle
Default:	Change to:

MD52251[0] Main spindle
MD52251[1] Counter spindle

Description:

M-Code for open chuck with rotating spindle.
e.g.: "M35" or "M1=35"

M code for Close chuck

MD52252[x] $MCS_M_CODE_CHUCK_CLOSE	M code for closing the chuck
Default:	Change to:

MD52252[0] Main spindle
MD52252[1] Counter spindle

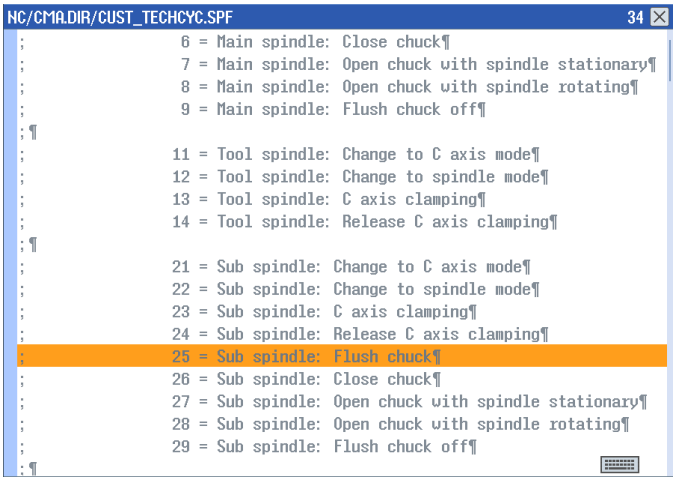
Description:

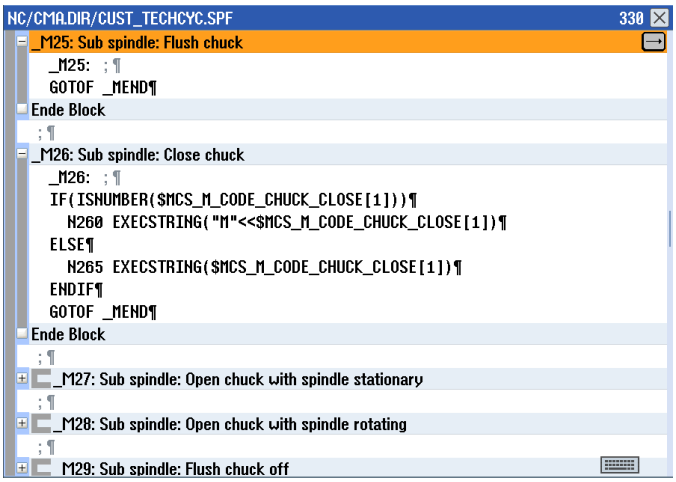
M-Code for Close chuck
e.g.: "M36" or "M1=36"

Notes:

Chuck flushing, closing, opening

Extract from the CUST_TECHCYC.SPF





Description:

Chuck of the counter spindle opening, closing and flushing

_M25: Counter spindle chuck flushing

_M26: Counter spindle chuck closing
if chuck [1] not closed, then closing

_M27: Counter spindle, chuck opening while stationary
if chuck [1] not open, then opening

_M28: Counter spindle, chuck opening while rotating
if chuck [1] not open while rotating, then opening

_M29: Counter spindle, de-selection of chuck flushing

Note:

The M-commands for opening and closing the chucks of the main and the counter spindle are not defined by CUST_TECHCYC.SPF. but by the machine data. This is necessary for the opening and closing of the chucks and the associated work piece motions to be correctly shown in the simulation of G-code programmes.

Description:

The machine data are evaluated in the modes _M26, _M27 and _M28 . If the machine data contain only a numerical value then this will be put out as an M-function without an extension. If the machine datum contains an auxiliary function with extension, e.g.. M1=34, then this will be carried out as a string.

Notes:

Clamping jaws counter spindle

Spindle chuck data

MD53240[x] $MAS_SPINDLE_PARAMETER	Spindle chuck data
Default: 0, 0, 0 mm	Change to:

Description:

for AX8: MCS2

MD53240[0] = e.g. 50 mm
Chuck dimension = ZC2 in the mask Spindle chuck data

MD53240[1] = e.g. 40 mm
Stop dimension = ZS2 in the mask Spindle chuck data

MD53240[2] = e.g. 22 mm
Jaw dimension = ZE2 in the mask Spindle chuck data

The dimension for ZE2 takes effect only when the jaw type 2 is selected; Z1 would be shifted by this amount.

Note 1

Alteration to the machine data will automatically be transferred to the menu display Spindle chuck data and vice versa.

In case of machines that feature a Y-axis, a parking position must be created for this spindle to ensure collision-free working. The default 0 is usually adequate.

Note 2

In order to show the counter spindle firstly without tailstocks, it is sensible to temporarily take out a setting of the basic machine!

MD52218 Bit 2 = 0

AX3: MCS1
MD53242[0] = 0
MD53242[1] = 0

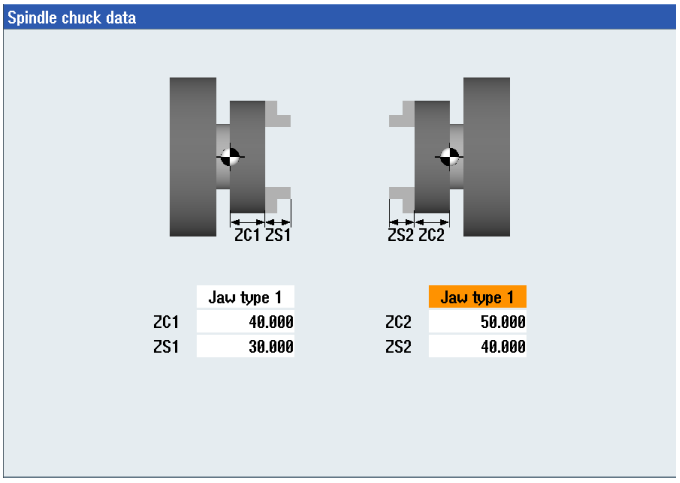
Check of the entries made in the window "Spindle chuck data"



The entries for the parameters "ZC2" chuck dimension and "ZS2 stop dimension for the counter spindle are identical in MD53240.

Note:

The input field ZE2 for the jaw dimensions be shown only if the jaw type 2 has been selected.



Notes:

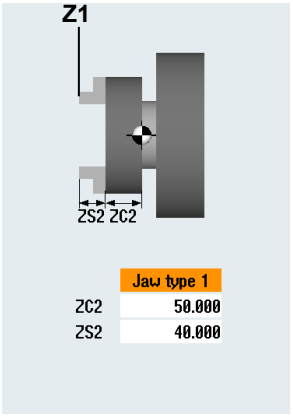
Spindle chuck data

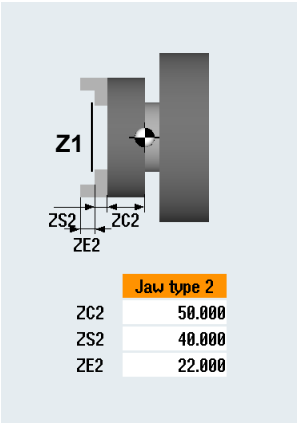
MD53241 $MAS_SPINDLE_CHUCK_TYPE	Spindle jaw type
Default: 0	Change to:

Description:

for AX8: MCS2

0 = Jaw type 1 = Clamping from outside
1 = Jaw type 2 = Clamping from inside





Jaw type 1:

Jaw type 1: Z1 lies in the WKS on the front edge of the clamping jaws; so if for instance Z1 = -10, the work piece will be inserted by 10 mm.

Jaw type 2:

Jaw type 2: Z1 lies on the inner jaw step; so for Z1 = 0, the work piece will be inserted right up to the jaw step.

Travelling to a fixed stop

Travelling to a fixed stop is required to enable the controlled transfer of a work piece, that was being machined in the main spindle, to the counter spindle for continued machining.

Travel to fixed stop mode

MD37000 $MA_FIXED_STOP_MODE	Travel to fixed stop mode
Default: 0	Change to:

Description:

Activation of sub functions of "Travel to fixed stop".

Release of Travelling to a fixed stop

Bit 0: Reserved

Release of the safe brake test
Bit 1 = 0: Safe brake test not available
Bit 1 = 1: Safe brake test can be executed under the control of the PLC.

Note:

When enabling these functions the user must ensure that Travelling to a fixed stop and Safe brake test are not called up simultaneously.

Notes:

Retract position Z for counter spindle

SD55232 $SCS_SUB_SPINDLE_REL_POS	Retract position Z for counter spindle
	Change to: e.g. 700 mm

Description:

Z-retract position for the counter spindle (ZS2).

Enable of the fixed stop alarms

MD37050 $MA_FIXED_STOP_ALARM_MASK	Enable of the fixed stop alarms
	Change to: 2

for AX7: MZS2

MD37050 = 2 (Linear axis of the counter spindle)

Description:

In this example Z2 is the linear axis of the counter spindle. Hence the parameter list for AX7: MZS2 must be selected and there MD37050 = 2 must be set. .

The machine datum determines whether the alarms:
20091 "Fixed stop not reached",
20094 "Fixed stop aborted" and
25042 "FOC: Standstill monitoring" are output.

MD = 0
Suppresses alarm 20091 "Fixed stop not reached"
MD = 2
Suppression of alarms
20091 "Fixed stop not reached" and
20094 "Fixed stop aborted"
MD = 3
Suppression of alarm 20094 "Fixed stop aborted"
Add value 8
Suppression of alarm 25042 "FOC: Standstill monitoring"

Note:

Independently of the setting of the alarm mask it is possible to read errors during approach of the fixed stop from the status variable $AA_FXS.

MD = 1 (Default)
Alarms 20091, 20094 and 25042 will be raised

The alarms 20091 "Fixed stop not reached" and 20094 "Fixed stop aborted" will be suppressed, since the function Travelling to a fixed stop is also used for checking cutting-off operations (see description below).

In order to recognise erroneous behaviour all the same when travelling to a fixed stop, these alarms are replaced by a cycle alarm:

61254 [channel %1:] block %2: Error while travelling to a fixed stop

1. Work piece reception by counter spindle

In setting data described below the paths, the feedrate, and the force for the travelling to a fixed stop will be defined.

Counter spindle: path for travel to fixed stop

SD55550 $SCS_TURN_FIXED_STOP_DIST	Counter spindle: path for travel to fixed stop
Default: 10 mm	Change to:

Counter spindle: Feedrate for travel to fixed stop

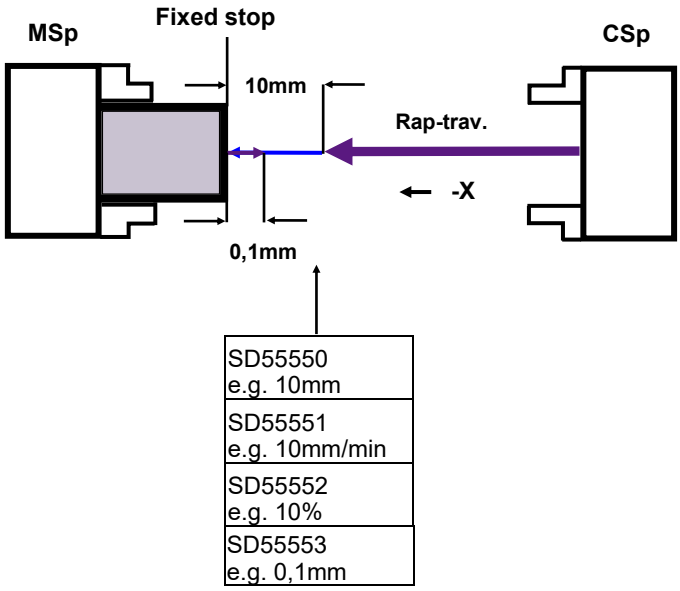
SD55551 $SCS_TURN_FIXED_STOP_FEED	Counter spindle: feedrate for travel to fixed stop
Default: 0 mm/min	Change to: 10 mm/min

Notes:

Counter spindle: Force for travel to fixed stop in %

SD55552 $SCS_TURN_FIXED_STOP_ FORCE	Counter spindle: force for travel to fixed stop in %
Default: 10 %	Change to:

Between the travelling to a fixed stop and the chucking the counter spindle can retract a small amount in order to relieve pressure on the work piece.



Counter spindle: Retraction path prior to chucking after fixed stop

SD55553 $SCS_TURN_FIXED_STOP_ RE-TRACTION	Counter spindle: retraction path prior to chucking after fixed stop
	Change to: e.g. 0,1

Sequence:

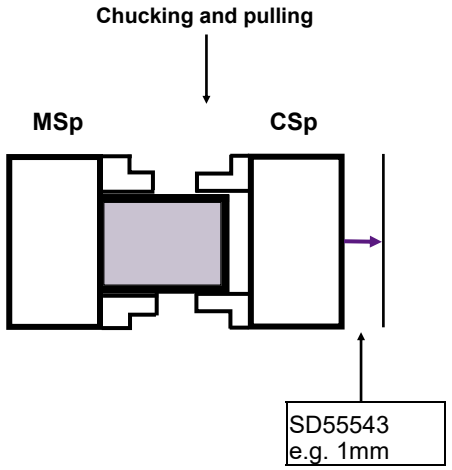
1. The counter spindle travels with G0 to the work piece in the main spindle until the distance specified in SD55550 has been reached.
2. The remaining distance will be travelled at the feedrate defined in SD55551 and with the reduced force as defined in SD55552 right up to the work piece.
3. The counter spindle retracts by the amount defined in SD 55553 in order to relieve the work piece.
4. Chucking the work piece in the counter spindle.

2. Cut-off

After chucking there is the possibility to cut off the work piece. Prior to this the counter spindle can be retracted a small amount together with the work piece to create a tensile force in the work piece. This relieves the tool a little bit during the parting.

Retraction path prior to cut-off with counter spindle

SD55543 $SCS_TURN_PART_OFF_RETRACT ION	Retraction path prior to cut-off with counter spindle
Default: 0 mm	Change to: 1 mm



Notes:

Cut-off check

After the cut-off a check can be carried out and during turning the function "Travelling to a fixed stop" can be used. The cut-off check can either be set up by means of the following channel specific setting data or it can be deactivated:

Path for cut-off check

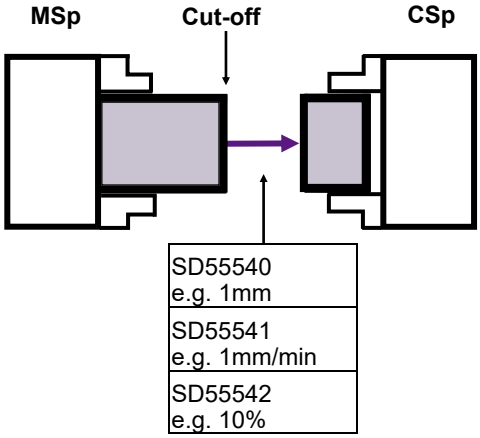
SD55540 $SCS_TURN_PART_OFF_CTRL_DIST	Path for cut-off check
Default: 0.1 mm	Change to: 1 mm

Feedrate for cut-of check

SD55541 $SCS_TURN_PART_OFF_CTRL_FEED	Feedrate for cut-off check
Default: 0 mm/min	Change to: 1 mm/min

Force for cut-off check in %

SD55542 $SCS_TURN_PART_OFF_CTRL_FORCE	Force for cut-off check in %
Default: 10 %	Change to:



Sequence:

1. The counter spindle travels by the amount defined in SD55543 to obtain a tensile force.
2. Cutting-off
3. For cut-off check the counter spindle retracts at the feedrate defined in SD55541 with reduced force as per SD55542 the distance from SD55540.

Notes:

Note:

The function „Travelling to a fixed stop“ is also used for the cutting-off check. The cutting off is taken to be successful, if the travelling to a fixed stop fails.

The alarms 20091 and 20094 are suppressed by changing the configuration of MD37050 to 2.

Alarm	Alarm Text
20091	The axis %1 has not reached the fixed stop
20094	Axis %1 end stop has been aborted.

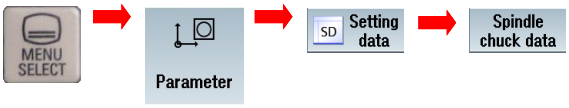
In order to recognise a mal-function during the cut-off check, these alarms are replaced by the cycle alarm 61255:

[channel %1:] Block %2: Error on cut-off: Tool breakage?

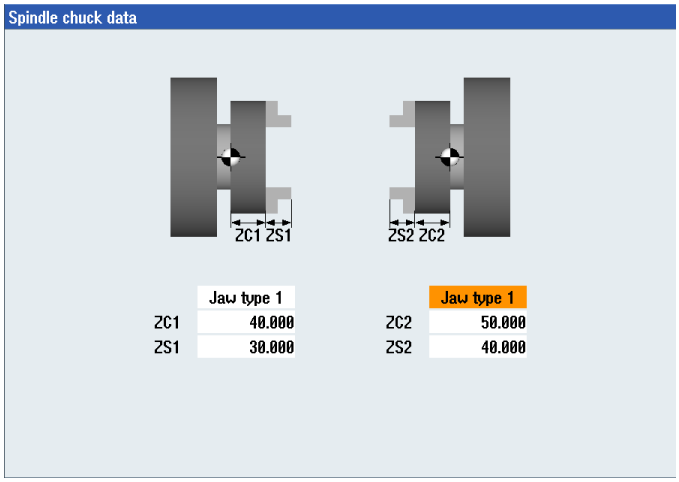
Setting up the counter spindle tailstock

On machines that feature a counter spindle, the counter spindle can be used as a tailstock by clamping a quill-point in the chuck. The mask „Spindle chuck data“ is laid out for both the definition of the tailstock dimensions (diameter and length) of the main and also of the counter spindle.

Opening the input mask Spindle chuck data



Depiction of the spindle chuck data for “Tailstock release“



Release of tailstock depiction

Function mask Turning

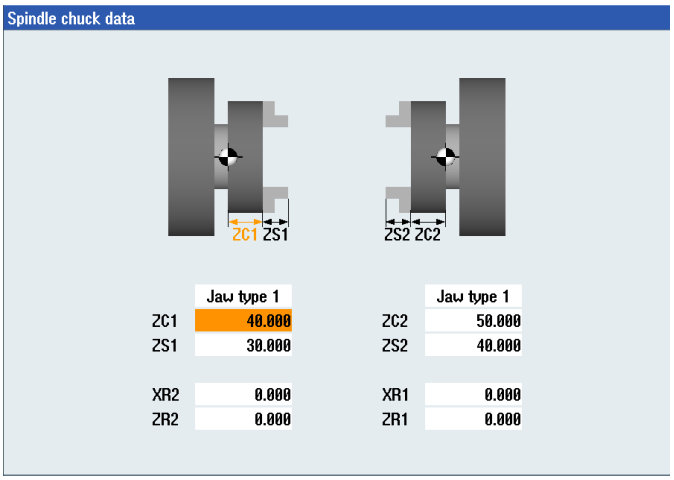
MD52218 $MCS_FUNKTION_MASK_TURN	Funktionsmaske Drehen
	Change to: Bit 2 = 1

Bit 2: Enable tailstock

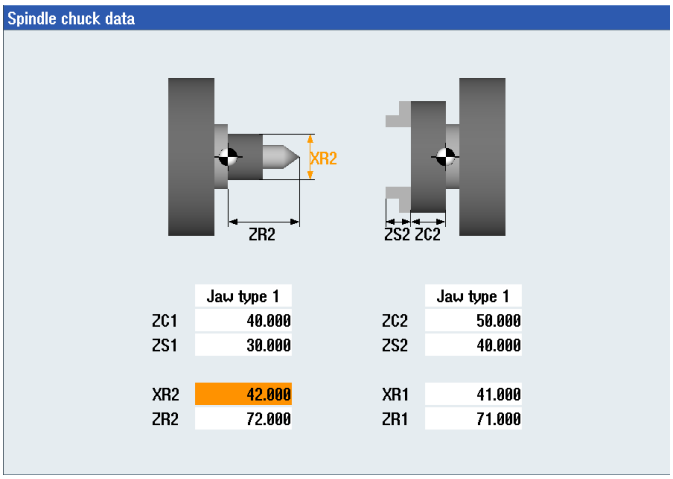
The tailstock is displayed when in the "program" tailstock is selected.

Notes:

Depiction of the spindles after “Tailstock release“ and Cursor position on ZC1. The fields for XR1, ZR1, XR2 and ZR2 have not yet been filled in.



Depiction of the tailstock after “Tailstock release“, cursor on position XR2 and the tailstock data have been entered manually.



The mechanical dimensions must be entered in the input fields XR1, ZR1, XR2 and ZR2.

Tailstock data for the counter spindle

MD53242[x] $MAS_TAILSTOCK_PARAMETER	Tailstock data
Default: 0 mm	Change to:

AX8: MCS2
MD53242[0] = XR2 Tailstock diameter
MD53242[1] = ZR2 Tailstock length

Tailstock data for the main spindle

MD53242[x] $MAS_TAILSTOCK_PARAMETER	Tailstock data
Default: 0 mm	Change to:

AX3: MCS1
MD53242[0] = XR1 Tailstock diameter
MD53242[1] = ZR1 Tailstock length

Note:

The entries in the mask Spindle chuck data correspond directly to the entries in MD53242[0] and MD53242[1] of AX3: MCS1 and AX8: MCS2.

Notes:

Machining example Main spindle - Counter spindle

Example:

By means of a simple machining example with the main spindle followed by the counter spindle the sequence is to be tested.

The programming of the operations with the counter spindle is carried out in the same manner as for the main spindle, for instance the location of cutting. For the machining with the counter spindle the Z-axis is mirrored (within the cycles). Only the cutting edge position must suit the actual cutting direction. This way of programming permits part programmes to be run both with the main spindle and also with the counter spindle.

Tool list

Preparation:

The tools used in this example must have been edited and must have the correct cutting edge orientation.

Tool list — Magazine

Loc.	MT LO.	Type	Tool name	ST	D	Length X	Length Z	Radius			
1			ROUGHING_TOOL_80	1	1	100.000	20.000	1.200	←	93.0	80
2			CUTTER_6_FA	1	1	0.000	100.000	6.000	2		
3			CUTTER_6_MA	1	1	0.000	100.000	6.000	2		
4			ROUGHING_80_CSP	1	1	100.000	20.000	1.200	→	93.0	80

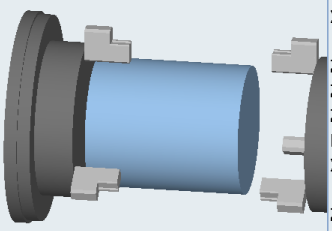
Machining program: Counter spindle

NC/WKS/TEST/COUNTERSPINDLE 1

P	Program header	G54 Cylinder
	Stock removal	T=ROUGHING_TOOL_80 F=0.5/rev S=500rev
	Counterspindle	Complete S1→S2 G55
	Stock removal	T=ROUGHING_80_CSP F=0.5/rev S=500rev Longitudinal
END	End of program	

Program head with workpiece definition:

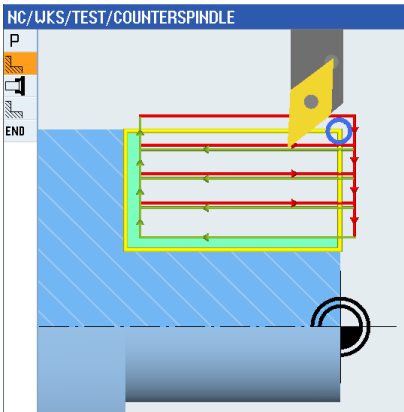
NC/WKS/TEST/COUNTERSPINDLE — Program header



Work offset	G54	
describe	No	
Blank	Cylinder	
XA	100.000	
ZA	0.000	
ZI	-150.000	inc
ZB	-120.000	inc
Retract	Simple	
XRA	3.000	inc
ZRA	3.000	inc
Tool change point	Machine	
XT	200.000	
ZT	100.000	
S1	2000	rpm
S2	2000	rpm
SC	1.000	
Machined dir. of rota	Up-cut	
ZS2W	700.000	

Stock removal and technology data main spindle:

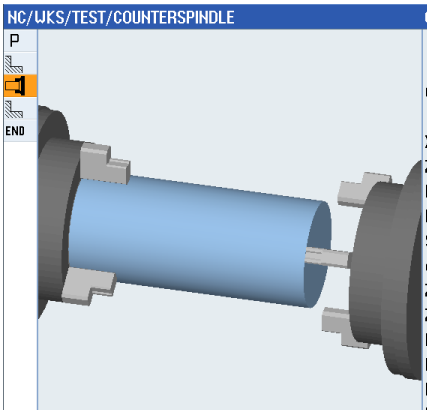
NC/WKS/TEST/COUNTERSPINDLE — Stock removal 1



T	ROUGHING_TOOL_80	D 1
F	0.500	mm/rev
S	500	rpm
Machining		
Pos.		
	Longitudinal	
X0	100.000	
Z0	0.000	
X1	70.000	abs
Z1	-20.000	inc
D	10.000	
UX	0.100	
UZ	0.100	

Transfer to the counter spindle:

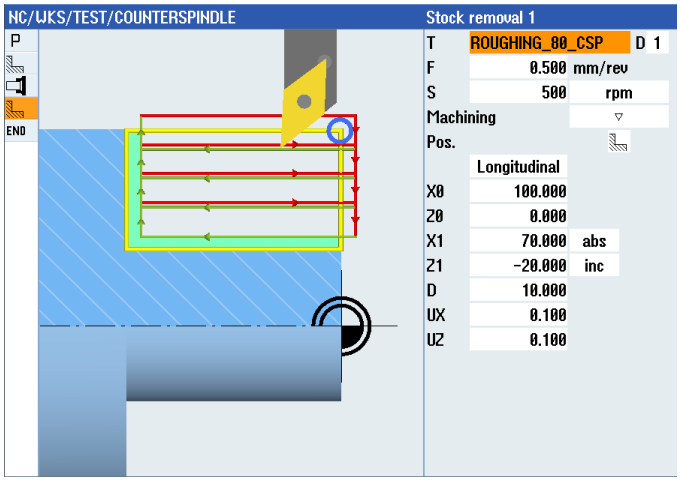
NC/WKS/TEST/COUNTERSPINDLE — Counterspindle



	Complete acceptance	
	Main to counterspindle	
Grip		
	Machine	
XP	200.000	
ZP	0.000	
Flush chuck	No	
DIR		
S	2000	rpm
α1	0.000	°
Z1	-29.000	
ZR	-28.000	inc
FR	50.000	mm/min
Fixed stop	No	
Draw		
Draw blank	No	
Cut-off cycle	No	
Rear		
Work offset	G55	
describe	Yes	
ZV	-150.000	inc
ZS2W	700.000	

Notes:

Stock removal and technology data counter spindle:



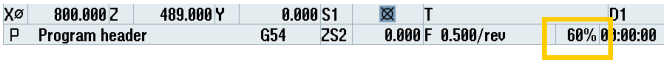
Test of the machining

A check of the machining sequences can be made having completed the inputs and selecting simulation.



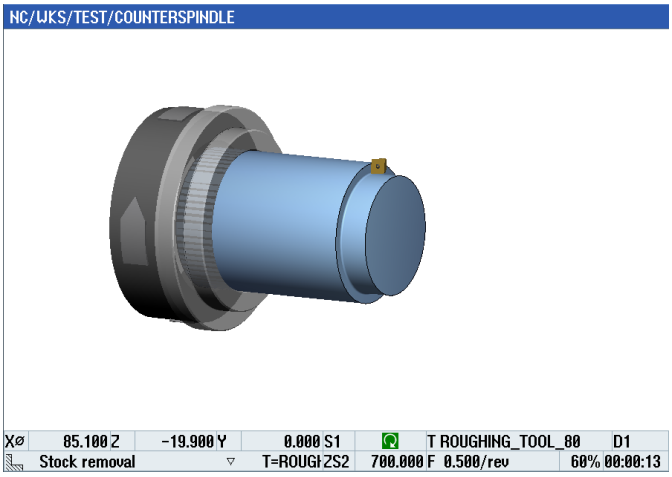
Alteration of the speed to about 60%.



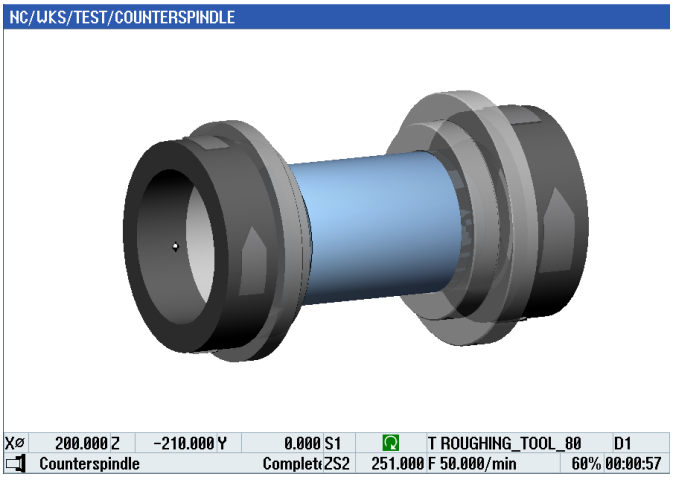


The machining progress is illustrated below in several stages.

Cut main spindle

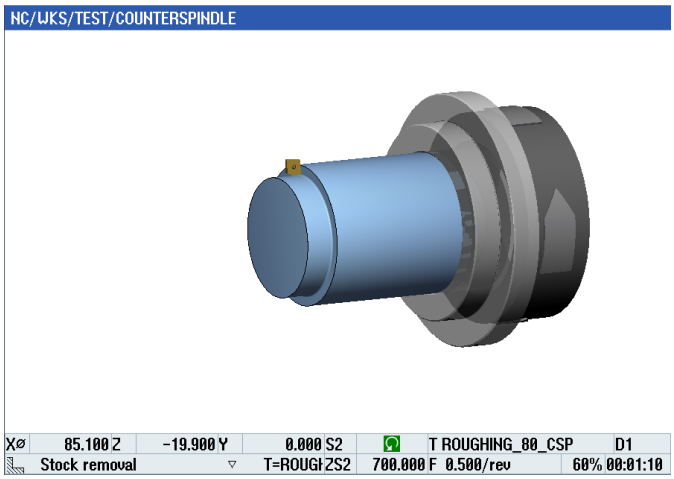


Transfer to the counter spindle

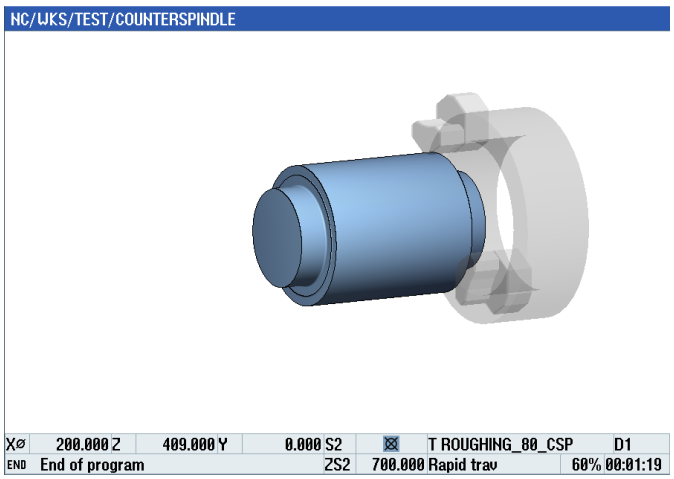


Notes:

Cut counter spindle



Machining completed



Machining example main spindle with TRANSMIT and counter spindle with TRACYL and TRANSMIT

Example:

By means of a simple machining program for facing with the main spindle followed by turning and facing with the counter spindle the sequence is to be tested.

Tool list

Preparation:

The tools used in this example must have been edified and must have the correct cutting edge orientation.

Tool list — Magazine

Loc.	MT LO.	Type	Tool name	ST	D	Length X	Length Z	Radius			
1			ROUGHING_TOOL_80	1	1	100.000	20.000	1.200	←	93.0	80
2			CUTTER_6_FA	1	1	0.000	100.000	6.000	2		
3			CUTTER_6_MA	1	1	0.000	100.000	6.000	2		
4			ROUGHING_80_CSP	1	1	100.000	20.000	1.200	→	93.0	80
5			CUTTER_6_FA_CSP	1	1	0.000	100.000	6.000	2		
6											

Machining program: MSp and CSp

NC/WKS/TEST/TRANSMIT_TRACYL_CSP 1

P	Program header	G54 Cylinder
	Contour	FACE_GROOVE
	Path milling	T=CUTTER_6_FA F=50/min S=2000rev Z0=0 Z1=5inc
	Counterspindle	Complete S1→S2 G55
	Contour	FACE_GROOVE_CSP
	Path milling	T=CUTTER_6_FA_CSP F=50/min S=2000rev Z0=0
	Longitudinal slot	T=CUTTER_6_MA F=500/min S=1600rev Y0=0 Z0=-30
END	End of program	

Note:

In the last two program steps the symbol for the front face machining with the counter spindle is the same as that for machining on the main spindle. For the machining with the counter spindle the Z-axis is mirrored.

Reason:

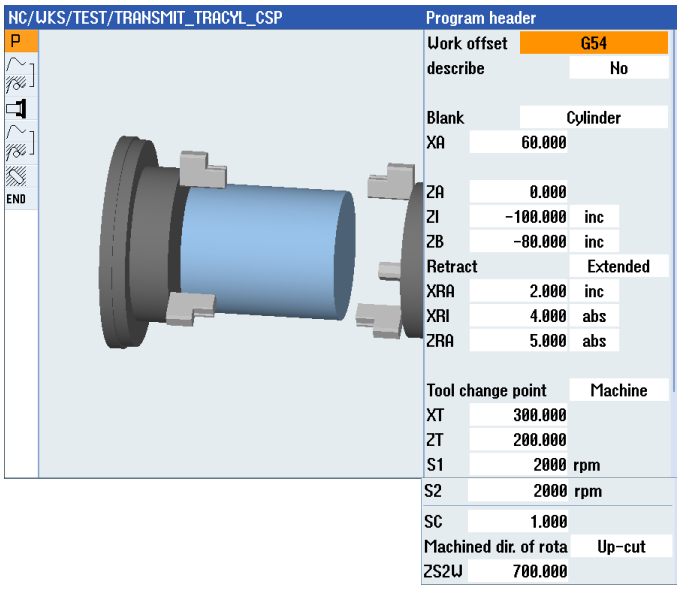
The programming is generally speaking the same as for machining with the main spindle. Hence these programs can be run on the main as well as on the counter spindle.

Machining time:

At the end of the graphic simulation the established machining time will be entered In the bottom program line.

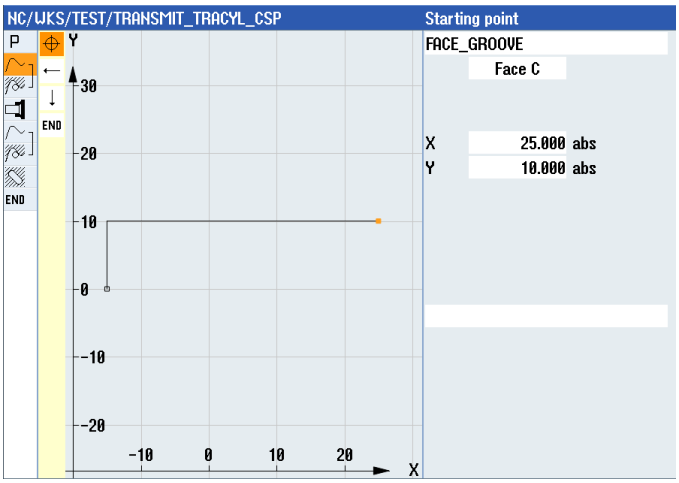
Notes:

Program head with workpiece definition:

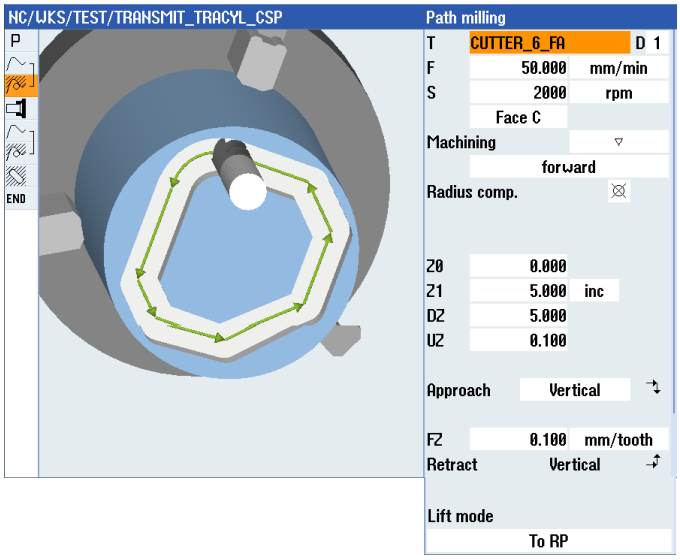


Create contour description:

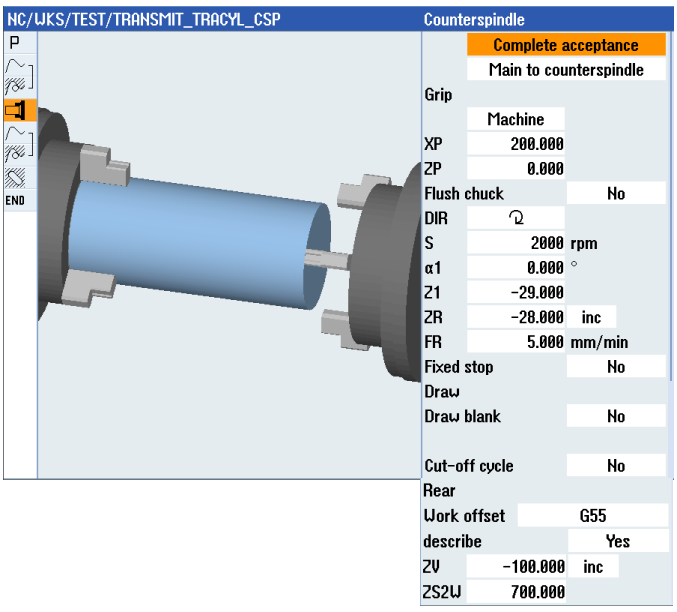
- Front face C, Starting point X=25 abs, Y =10 abs
- Straight X=-15 abs
- Straight Y=0 abs



Continuous path milling front face C main spindle:



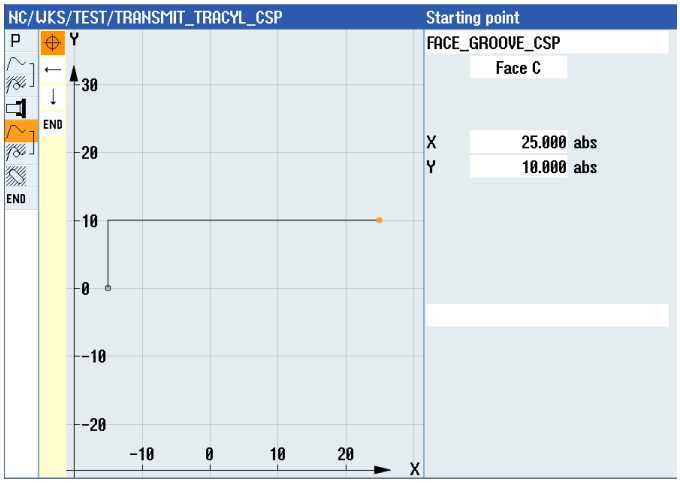
Transfer to the counter spindle:



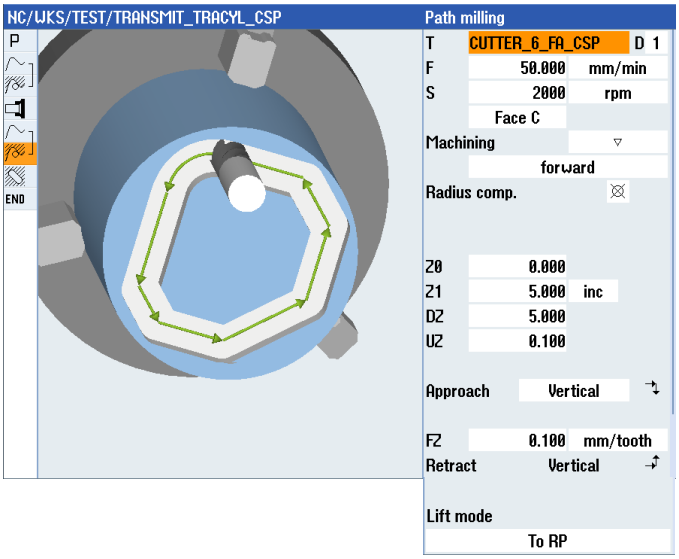
Notes:

Create contour description:

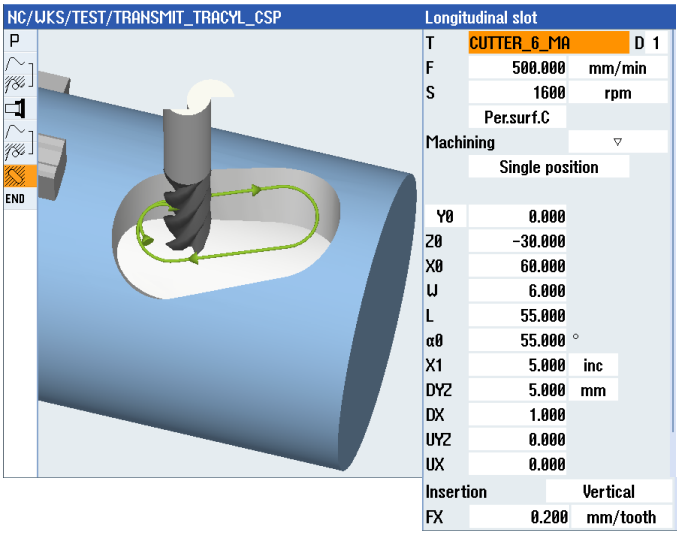
- Front face C, Starting point X=25 abs, Y =10 abs
- Straight X=-15 abs
- Straight Y=0 abs



Continuous path milling front face C counter spindle:



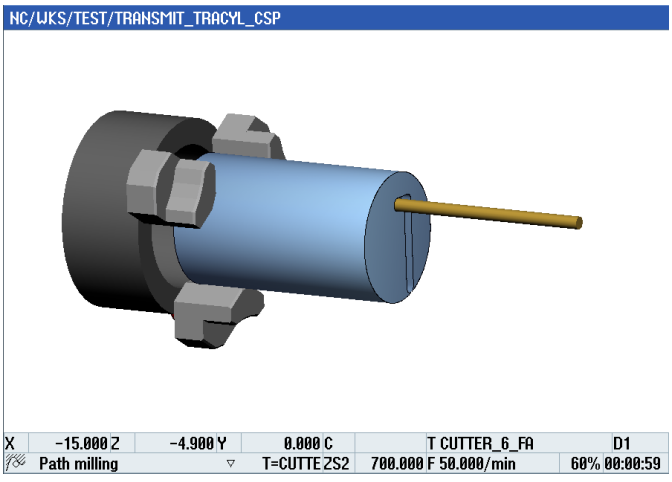
Longitudinal slot Per. surface C counter spindle:



Start simulation:

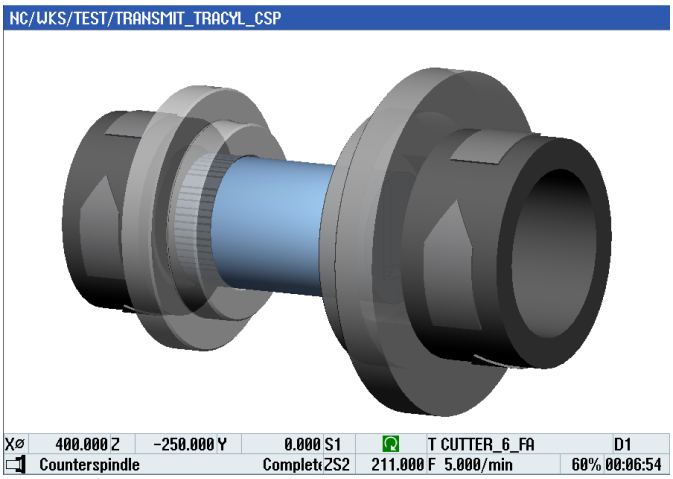
The machining progress is illustrated below in several stages.

Start, work piece in the main spindle

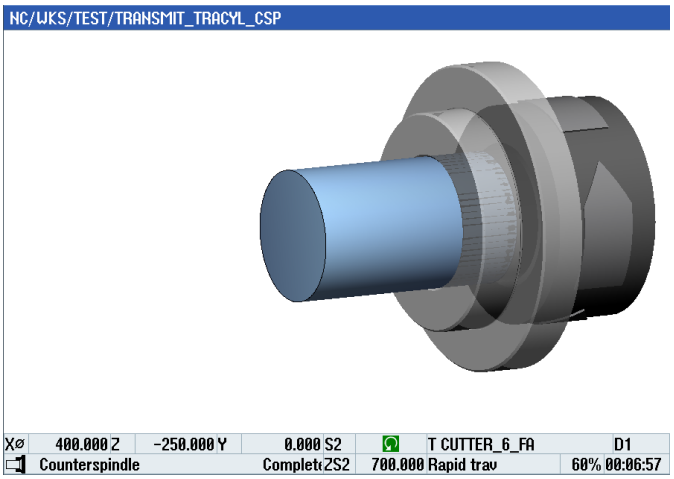


Notes:

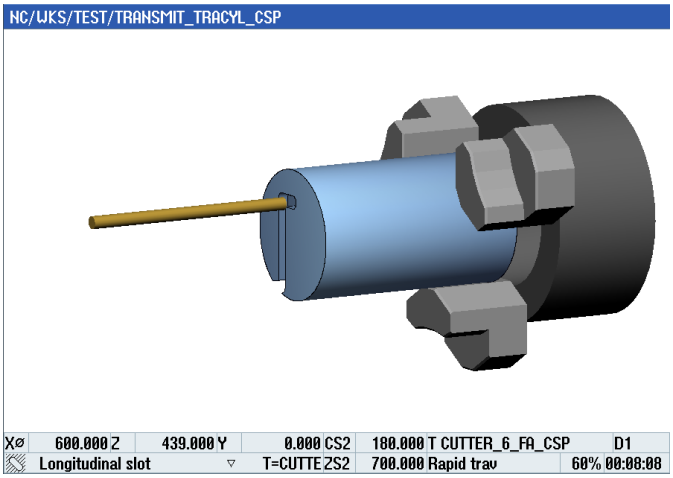
Transfer of the work piece to the counter spindle



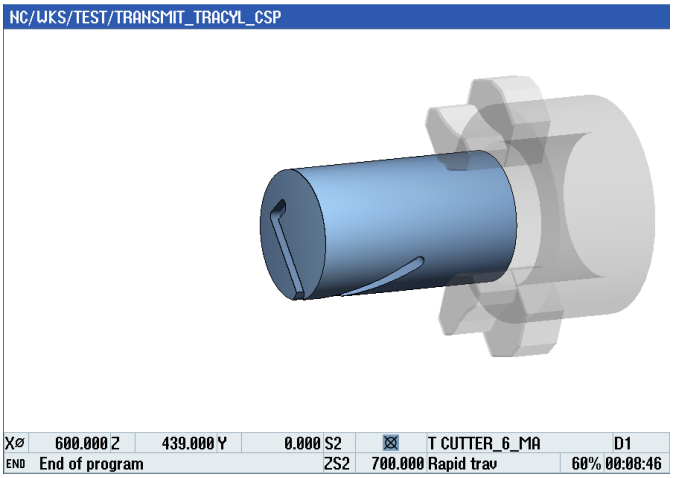
Retraction of the counter spindle



Work piece in the counter spindle



End of the machining



A513: END

Notes:

SIEMENS

MACHINE TEST PROTOCOL FOR SINUMERIK

Product / Version	Comment
SINUMERIK ONE **SINUMERIK 840D sl** **SINUMERIK 828D**	Virtual and Real

Machine Description	**Machine Features**

Electrical Cabinet Protocol Attached	☐	Safety Integrated Protocol Attached	☐

Title **VMC 600**	DEX Machine Tool Incorporated
Document no.	Date
	Protocol version 6.1

Copyright © Siemens AG 2020 All Rights Reserved.

Contents

1 Responsible persons: 3
1.1 Schedule 3
1.2 First commissioning contacts 3
2 Machine Test summary 4
2.1 Summary 4
2.2 Result 4
2.3 Release of the machine recommended (YES/NO/Reason): 4
2.3.1 Exception List 4
2.3.2 The machine has been released for sale with the following exceptions: 4
2.4 Changes to standard data – as reference data 4
2.5 Data Backup 4
3 Change Log 5
3.1 Changes to the machine. 5
Components contained in the machine configuration. 5
3.2 Control 5
4 Test objective 6
4.1 Note 6
5 Functions tested 7
Detailed Test 8
5.1 Machine Area 8
5.1.1 Jog → Ref 8
5.1.2 Jog → MCP 9
5.1.3 Jog → HMI Soft keys 11
5.1.4 MDA → Teach-in 11
5.1.5 MDA → HMI Softkeys 12
5.1.6 MDA → MDA → Teach-in 13
5.1.7 AUTO 14
5.2 Technology 15
5.2.1 Milling General 15
5.2.2 Milling 3+2 15
5.2.3 Milling 5 Axis 16
5.2.4 Turning Basic 16
5.2.5 Turning Advanced 17
5.3 Parameter 18
5.3.1 Machine Builder Options 18
5.3.2 Measure tool 18
5.3.3 Setting data 18
5.3.4 Magazine 18
5.3.5 Tool List 19
5.4 Geometry 19
5.4.1 Circularity test HMI 19
5.4.2 Ball-bar circularity test 19
5.5 Safety Integrated 19

Copyright © Siemens AG 2020 All Rights Reserved.

5.5.1 Safety related test 19
5.6 Password protection 20
5.6.1 Securing data areas 20
5.7 Various 21

1 Responsible persons:

1.1 Schedule

Start date of First Commissioning:

End date of First Commissioning:

1.2 First commissioning contacts

Name	Company / Department	Contact information
Mr. Peter Writer	*Machine Builder ABC*	*PLC Program Contact*

Copyright © Siemens AG 2020 All Rights Reserved.

2 Machine Test summary

2.1 Summary

2.2 Result

2.3 Release of the machine recommended (YES/NO/Reason):

2.3.1 Exception List

2.3.2 The machine has been released for sale with the following exceptions:

Function:	Test Number	Remark

2.4 Changes to standard data – as reference data

Extend table as necessary

Machine data name + ID number	Original Value	New Value

2.5 Data Backup

Create data backup after completing the test and attach to the protocol.

Appendices:

- (NC Archive)
- (PLC Archive)
- (Drive Archive)
- (HMI Archive)
- (Complete archive) (Backup)(Setup)
- (Clone TGZ file for series production)
  (Image for restore)
- (Ghost image IPC)
- (License key data)

Archive format dependent upon control type:

SINUMERIK ONE	DSF Format
SINUMERIK 840D sl	ARC Format
SINUMERIK 828D	ARD Format

Copyright © Siemens AG 2020 All Rights Reserved.

3 Change Log

3.1 Changes to the machine.

This table describes features added and tested after the protocol was completed, it serves as a confirmation for the Machine Tool Builder that prior to machine delivery all features were tested without exception.

Date	Change	Reason	Tested	Responsible person	Signature	Contact information
			☐			
			☐			
			☐			
			☐			
			☐			
			☐			
			☐			

Components contained in the machine configuration.

Machine

Type of Machine	
Serial No.	
Configuration	
Axes	
Spindle	
Miscellaneous	

3.2 Control

NC type	SINUMERIK ONE / Sinumerik 840D sl / Sinumerik 828D
Serial No.	
Machine control panel	
Software version:	
User software	
PLC Application	

Copyright © Siemens AG 2020 All Rights Reserved.

4 Test objective

4.1 Note

The test plan extends the commissioning documentation of Sinumerik Operate.

The application is tested with the current test plan, which helps to achieve for the most part an error-free interaction of SINUMERIK with the machine. No guarantee for fault-free machine application can be guaranteed.

Siemens is not responsible for the PLC safety related functions which are implemented on the Machine. The Machine tool builder is responsible in satisfying the receiving countries "Safety regulations regarding CNC machine tools" regulations.

Basic knowledge of operation and programming of Sinumerik is required for carrying out the tests. In case of some test tasks, it is necessary to start an automatic program or to trigger machine functions. It is, therefore, explicitly indicated that all operations that trigger machine functions should only be carried out by the employee of the machine manufacturer, who is authorized to do this.

The necessary programs are to be created directly at the machine.

ALL RELEVANT TESTS HAVE TO BE CARRIED OUT WITH THE FOLLOWING CONDITIONS:	
1) Siemens Code (DIN) and ShopMill/ShopTurn with →	**Metric system of units**
2) Siemens Code (DIN) and ShopMill/ShopTurn with →	**Inch System of Units**
3) ISO / dialect with →	**Metric system of units – if activated**
4) ISO / dialect with →	**Inch system of units – if activated**

Copyright © Siemens AG 2020 All Rights Reserved.

5 Functions tested

Confirm the functions as per the test plan

Test.	Function	Result / Accepted/To Do
5.1	**Machine**	
5.1.1	Jog → Reference	*All tests completed*
5.1.2	Jog → MCP	*To Do´s in list*
5.1.3	Jog → HMI Soft keys	
5.1.4	MDA → Teach-in	
5.1.5	MDA → HMI Soft keys	
5.1.6	MDA → MDA → Teach-in	
5.1.7	AUTO → DRF, Jog	
5.2	**Technology**	
5.2.1	Milling General	
5.2.2	Milling 3+2	
5.2.3	Milling 5 Axis	
5.2.4	Turning Basic	
5.2.5	Turning Advanced	
5.3	Parameter	
5.3.1	Machine Builder Options	
5.3.2	Measure Tool	
5.3.3	Setting data	
5.3.4	Magazine	
5.3.5	Tool List	
5.4	**Machine Geometry**	
5.4.1	Circularity Test HMI	
5.4.2	Ball bar circularity Test	
5.5	**Safety integrated**	
5.5.1	Safety Integrated protocol	
5.6	General	
5.6.1	Securing data areas	
5.7	Various	

Copyright © Siemens AG 2020 All Rights Reserved.

Detailed Test
5.1 Machine Area
5.1.1 Jog → Ref

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	3	Reference not necessary (Absolute measuring system) (Start-up in JOG)	☐		☐			X	X	X
T+M	4	Prevention of Axis faults, when starting in axis operation, the Spindles with axis functionality should be synchronized in jog prior to NC Start	☐		☐			X	X	X
T+M	5	Additional equipment requiring reference point approach should be referenced prior to the first NC Start (Turret). “Alarm prevention”	☐		☐			X	X	X
T+M	9	The feedrate and Spindle enable signals should not be active immediately after power-on of the machine. The operator must enable these signals with the respective key presses on the MCP	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.2 Jog → MCP

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	If the axis is resting on the Hardware limit switch, it should be possible to JOG the axis away in the opposite direction.	☐		☐			X	X	X
T+M	2	It should be possible in the RESET state to change to JOG mode	☐		☐			X	X	X
T+M	3	When changing from REF to Jog - Operator must select again the required axis. There should be no pre-selected axis.	☐		☐			X	X	X
T	5	With Turning machines, it should also be possible to move more axes simultaneously. The machine builder has to realize this in the PLC.	☐		☐			X	X	X
M	6	With Milling machines, it is also possible to Jog axes simultaneously; however, the machine builder has to realize this.	☐		☐			X	X	X
T+M	7	The machine builder should also ensure that the rotary axis can be jogged, including rotary axis functionality with Spindles. C+ button = spindle direction M4 C- button = spindle direction M3	☐		☐			X	X	X
T+M	8	The feedrate override should be effective on all axes which can be Jogged. All axes should come to standstill when the feed override is set to zero.	☐		☐			X	X	X
T+M	9	When the feed stop button is pressed the Jog plus and minus buttons should have no effect on any axis motion. The axis should all come to a stop.	☐		☐			X	X	X
T+M	10	When the spindle stop button is pressed the Jog plus and minus buttons should have no effect on the spindle motion. The spindles should all come to a stop.	☐		☐			X	X	X
T	11	When Spindle direction buttons are available, the spindle should move clockwise or anti-clockwise dependent upon which button is pressed. The default speed is defined in the machine data.	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

T+M	12	The Spindle override should correct the currently controlled spindle/s.	☐		☐			X	X	X
T+M	13	Do all axis move according to the work plane defined in the DIN 6625 standard.	☐		☐			X	X	X
T+M	14	When direction keys plus and minus are pressed simultaneously, the axis in motion should stop immediately. Further motion can only be achieved when both buttons are once again released, and the desired direction button pressed again	☐		☐			X	X	X
T+M	15	WCS >><< MCS With key not pressed (MCS) the axis should be moved using the Machine axis interface, and the axis will move on an individual basis, without active rotations. With key pressed (WCS) the axis should be moved using the Channel axis interface, and the axis will move on an individual basis, with active rotations.	☐		☐			X	X	X
T+M	16	Repos With interruption in Automatic, it should be possible to switch to Jog, move axis away from the contour, subsequently return to Automatic / Repos and Jog the axis back to the previous position.	☐		☐			X	X	X
T+M	17	TCS Tool Coordinate System Jog should be activated dependent upon machine kinematic	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.3 Jog → HMI Soft keys

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	T,S,M → T-D Select a tool from the tool list, using SK TOOL and IN MANUAL. Press cycle start tool should be transferred into the spindle or positioned before the work piece. The tool must not rotate if already present in the spindle or present before the work piece. Repeat the process with other tools from the magazine. Prior to changing from a driven tool, the rotating tool should be stopped prior to changing.	☐		☐			X	X	X
T+M	2	Spindle → Set a spindle speed and activate "Cycle Start" (together with spindle M function) Repeat the procedure with further spindles.	☐		☐			X	X	X
T+M	3	Spindle gear change→ Change gear, if a gear box is available on the machine.	☐		☐			X	X	X
T+M	4	Spindle direction → Program M3 and M4 on all spindles which are available on the machine and ensure for the programmed code that spindle rotates in the correct direction.	☐		☐			X	X	X
T+M	5	Measure Tool → Manual → Test with all types of possible tools Automatic → Test with all types of possible tools	☐		☐			X	X	X
T+M	6	Position → All axis must be available, ensure that it is possible to move the spindle (C-Axis function) without first running in spindle operation	☐		☐			X	X	X
T+M	7	MCS / WCS → Should change only the display from MCS/WCS	☐		☐			X	X	X
T+M	8	Hand wheel → Test in both MCS and WCS Syncron-actions→. show currently active SYNACT Settings → Test and acknowledge functionality of the Setting data's	☐		☐			X	X	X

5.1.4 MDA → Teach-in

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Teach-in → It should be possible to Jog the axis in MDA - Teach-in **OPTION 828D**	☐		☐			X	X	X
	2		☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.5 MDA → HMI Softkeys

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Program Control → When the PRT checkbox is set the axis should work internally, and not physically move the axis. The signal should work only in Automatic mode - when changing to Jog/MDA the signal should not be active except when an external switch is used.	☐		☐			X	X	X
T+M	2	Program Control → DRY - When selected, the axis should move with the feedrate which is defined in Settings. This should only be available in Automatic.	☐		☐			X	X	X
T+M	3	Program Control → M01 - When selected, the next occurrence of M01 in the NC program should cause the NC program to stop and wait for the next NC start.	☐		☐			X	X	X
T+M	4	Program Control → When DRF is selected it should be possible in the Automatic mode of operation to move the axis with the Hand wheel, the position will be stored in the DRF offset. **OPTION 828D**	☐		☐			X	X	X
T+M	5	Program Control → SKP - When selected, the next NC block with the Skip block instruction will not be executed.	☐		☐			X	X	X
T+M	6	G functions → G - Functions for the reset state of the machine - To be checked for: Turning Milling	☐		☐			X	X	X
T+M	7	Auxiliary functions → Machine tool builder should check that the functions which he does not allow in MDA are not processed.	☐		☐			X	X	X
T+M	8	>> Extend SK → All G functions G - Functions for the reset state of the machine - To be checked for: Turning Milling	☐		☐			X	X	X
T+M	9	> Extend menu → Hand wheel → Move axis using Hand wheel in the machine axis interface or using Geometry interface in connection with Teach-in mode **OPTION 828D**	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.6 MDA → MDA → Teach-in

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	MDA → Tool change in MDI window - T01 + Cycle start for Turning machines T01 M06 + Cycle start for Milling machines.	☐		☐			X	X	X
T+M	2	MDA → It should not be possible to Jog the axis in MDA - exception: Teach-in is active	☐		☐			X	X	X
T+M	**3**	Teach-in → Hand wheel → Move axis using Hand wheel in the machine axis interface or using Geometry interface in connection with Teach-in mode **OPTION 828D**	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.1.7 AUTO

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Move axis using Hand wheel/Jog/Inc in the machine axis interface to generate a DRF offset **OPTION 828D**	☐		☐			X	X	X
T+M	2	Jog in Automatic → Jog Mode Mask is set to 0 as standard, which means you are not able to Jog the axis in Automatic, with the exception of Teach-in. Jog Mode Mask is set to 1, means you are able to Jog the axis in Automatic. The machine builder should explicitly set this machine data when he requires this functionality.	☐		☐			X	X	X
T+M	3	AUTO → Cycle start	☐		☐			X	X	X
T+M	4	AUTO → Cycle Stop	☐		☐			X	X	X
T+M	5	AUTO → Cycle single block	☐		☐			X	X	X
T+M	6	AUTO → Program reset	☐		☐			X	X	X
T+M	7	AUTO → Feed Stop in Program effect	☐		☐			X	X	X
T+M	8	AUTO → Spindle stop in Program - effect	☐		☐			X	X	X
T+M	9	AUTO → Feed override = 0	☐		☐			X	X	X
T+M	10	AUTO → Spindle override = 50	☐		☐			X	X	X
T+M	11	AUTO → Effect of E-Stop	☐		☐			X	X	X
T+M	12	Program control → PRT - Program Test	☐		☐			X	X	X
T+M	13	Program control → DRY - Dry run feedrate	☐		☐			X	X	X
T+M	14	Program control → M01 - Program stop	☐		☐			X	X	X
T+M	15	Program control → DRF - Differential hand wheel offset **OPTION 828D**	☐		☐			X	X	X
T+M	16	Program control → SKP - Block skip	☐		☐			X	X	X
T+M	17	Block search → M,S,T,H - Effect Search to Tool	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.2 Technology

5.2.1 Milling General

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
M	2	Customer Cycles CUST800 etc.	☐		☐			X	X	X
M	3	Settings for Technology and measuring cycles	☐		☐			X	X	X
M	4	Settings for Tool management views	☐		☐			X	X	X
M	5	Cycle settings when multitasking Turning on Milling machines Milling on Turning machines	☐		☐			X	X	
M	6	Specific customer cycle CUST832	☐		☐			X	X	X
M	7	3/5 Axis M Dynamics MD Test	☐		☐			X	X	X
M	8	Tool change with DRF Deactivation M6	☐		☐			X	X	X
M	9	Cycle 996/9960 Measuring Kinematic	☐		☐					
M	10		☐		☐					
M	11		☐		☐					
M	12		☐		☐					

5.2.2 Milling 3+2

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
M	1	Settings for Swiveling	☐		☐			X	X	X
M	2	Tool change with active CYCLE800 / Transmit / Tracyl	☐		☐			X	X	X
M	3	Settings Transmit / Tracyl	☐		☐			X	X	X
M	4	Reset State with Head/ Table / Mixed kinematic	☐		☐			X	X	
M	5	Power off state / Head / Table / Mixed kinematic	☐		☐			X	X	

Copyright © Siemens AG 2020 All Rights Reserved.

5.2.3 Milling 5 Axis

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
M	1	Settings for Traori	☐		☐			X	X	
M	2	Settings for CUST832	☐		☐			X	X	X
M	3	Correct Trafo Type set Or Trafo based upon Kinematic Chain	☐		☐			X	X	X
M	4	Tool Change with active Traori	☐		☐			X	X	
M	5	Reset State with Head/ Table / Mixed kinematic	☐		☐			X	X	
M	6	Power off state / Head / Table / Mixed kinematic	☐		☐			X	X	
M	7	Settings for Orientation / ORIAXES reset ORIEULER ORIMKS ORIWKS reset	☐		☐			X	X	

5.2.4 Turning Basic

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T	1	Is the coordinate system displayed correctly in the editor?	☐		☐			X	X	X
T	2	Are the softkeys for the standard cycles displayed in the editor?	☐		☐			X	X	X
T	3	Is the “plunge-cutting” and “plunge-turning” softkey displayed in the editor?	☐		☐			X	X	X
T	4	Are the softkey for the remaining-material machining displayed in the editor?	☐		☐			X	X	X
T	5	Is the “Tailstock” parameter displayed in the program header? (only in ShopTurn)	☐		☐			X	X	X
T	6	Is the “parts gripper” parameter displayed in the cut-off cycle?	☐		☐			X	X	X
T	7	Does the parts gripper in the cut-off cycle function correctly?	☐		☐			X	X	X
T	10	Is a program simulated correctly with a open working-area door?	☐		☐			X	X	X
T	11	Is the simulation aborted when the “Reset” key is pressed?	☐		☐			X	X	X
T	12	Is the calculated program run time updated during the simulation?	☐		☐			X	X	X
T	13	Is a program executed correctly when you press “Cycle Start”?	☐		☐			X	X	X
T	14	Is the “Base block” softkey displayed?	☐		☐			X	X	X
T	15	Is the” Simultaneous recording” softkey displayed?	☐		☐			X	X	X
T	16	Does “Block search” work correctly?	☐		☐			X	X	X
T	17	Does” REPOS” work correctly?	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.2.5 Turning Advanced

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T	1	Is a Face milling contour milled correctly on main spindle?	☐		☐			X	X	X
T	2	Is a Face milling contour milled correctly on counter spindle?	☐		☐			X	X	X
T	3	Is a Peripheral milling contour milled correctly on main spindle?	☐		☐			X	X	X
T	4	Is a Peripheral milling contour milled correctly on counter spindle?	☐		☐			X	X	X
T	5	Is a Y-Axis milling contour milled correctly on main spindle?	☐		☐			X	X	X
T	6	Is a Y-Axis milling contour milled correctly on counter spindle?	☐		☐			X	X	X
T	7	Is a B-Axis milling contour milled correctly on main spindle?	☐		☐			X	X	
T	8	Is a B-Axis milling contour milled correctly on counter spindle?	☐		☐			X	X	
T	9	Is a position pattern drilled/tapped correctly on main spindle?	☐		☐			X	X	X
T	10	Is a position pattern drilled/tapped correctly on counter spindle?	☐		☐			X	X	X
T	11	Does the counter spindle cycle correctly? (only in ShopTurn)	☐		☐			X	X	X
T	12	Does turning with B axis on the main spindle correctly?	☐		☐			X	X	
T	13	Does turning with B axis on the counter spindle correctly?	☐		☐			X	X	
T	14	Are the softkeys for the measuring cycles displayed in the editor?	☐		☐			X	X	X
T	17	Does a workpiece measurement program work correctly?	☐		☐			X	X	X
T	18	Does the measurement result pictures display correctly?	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.3 Parameter

5.3.1 Machine Builder Options

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Test the commissioning of machine builder options **EASY EXTEND 828D**	☐		☐			X	X	X
T+M	2	Test the activation / deactivation of machine builder options **EASY EXTEND 828D**	☐		☐			X	X	X
T+M	3	Test Regular Maintenance Tasks **MAINTENANCE PLANNER 828D**	☐		☐			X	X	X

5.3.2 Measure tool

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Measure the length of the Tool	☐		☐			X	X	X
T+M	2	Measure the Diameter / radius of the tool	☐		☐			X	X	X

5.3.3 Setting data

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Spindle data - minimum spindle speed should be set to a value not equal to 0	☐		☐			X	X	X

5.3.4 Magazine

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Relocate a tool in the Magazine	☐		☐			X	X	X
T+M	2	Position the Magazine	☐		☐			X	X	X
T+M	3	Rename a tool	☐		☐			X	X	X
T+M	4	Disable magazine locations -	☐		☐			X	X	X
T+M	5	Test oversize tools	☐		☐			X	X	X
T+M	6	Test tools with fixed locations	☐		☐			X	X	X
T+M	7	Test interruption scenarios in tool changer ■ Reset the channel ■ Emergency stop ■ Isolator switch	☐		☐			X	X	X
M	8	Test Hand loaded Tools in program	☐		☐			X	X	X
T+M	9	L6 Optimized, Spindle Tool = Pre-selected Tool	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.3.5 Tool List

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Enter a new tool directly in the spindle	☐		☐			X	X	X
T+M	2	Enter a new tool directly in the Magazine	☐		☐			X	X	X
T+M	3	Enter a new tool directly in the Buffer	☐		☐			X	X	X
T+M	4	Delete a tool directly in Spindle	☐		☐			X	X	X
T+M	5	Delete a tool from the Magazine	☐		☐			X	X	X
T+M	6	Delete a tool from the buffer	☐		☐			X	X	X
T+M	7	Unload the tool in the spindle	☐		☐			X	X	X
T+M	8	Machine with sub-spindle should be possible to create 2 tools for 1 turret position (different tool types !!!)	☐		☐			X	X	X
T+M	9	It must be possible to program/call a tool with tool name (e.g. T="name") and also turret position T1	☐		☐			X	X	X
T+M	10	Relocate a tool in the Spindle	☐		☐			X	X	X
T+M	11	Tool Change in Inch and Metric	☐		☐			X	X	X

5.4 Geometry

5.4.1 Circularity test HMI

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Geometry axis	☐		☐			X	X	X
T+M	2	Thread boring	☐		☐			X	X	X

5.4.2 Ball-bar circularity test

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Machine axis	☐		☐			X	X	X
T+M	2	Geometry axis	☐		☐			X	X	X

5.5 Safety Integrated

5.5.1 Safety related test

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Safety integrated acceptance Test.	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.6 Password protection

5.6.1 Securing data areas

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	1	Key switch → Tools Tool wear length and radius Tool length and radius	☐		☐			X	X	X
T+M	2	Key switch → Create a tool	☐		☐			X	X	X
T+M	3	Key switch → Delete a tool	☐		☐			X	X	X
T+M	4	Programming → Open program	☐		☐			X	X	X
T+M	5	Programming → Edit program	☐		☐			X	X	X
T+M	6	Programming → Select program (NC)	☐		☐			X	X	X
T+M	7	Offsets → fine coarse	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

5.7 Various

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	3	Clone backup (TGZ) available for Series machines.	☐		☐			X	X	X
T+M	4	Country specific safety regulations included in PLC program + door interlocks etc.	☐		☐			X	X	X
T+M	5	NCK, HMI, PLC and Drive archives tested.	☐		☐			X	X	X
T+M	6	TGZ file for series production tested	☐		☐			X	X	X
T+M	7	Adaptive Friction compensation active	☐		☐			X	X	X
T+M	8	Top Surface date set and tested	☐		☐			X	X	X
T+M	9	Customer Cycles optimized and tested	☐		☐			X	X	X
T+M	10	Top Speed data set and Tested	☐		☐			X	X	X
T+M	11	Save Data VRam copy	☐		☐			X	X	X
+M	12	PLC Written according to 828D design document	☐		☐					X
T+M	13	ESR has been commissioned	☐		☐			X	X	X
T+M	14	ELG has been commissioned	☐		☐			X	X	
T+M	15	CA ECO Commissioned	☐		☐			X	X	X
T+M	16	CA Advanced	☐		☐			X	X	
T+M	17	Sidescreen activated	☐		☐			X	X	X
T+M	18	Display Manager activated	☐		☐			X	X	X
T+M	19	WinCC installed and configured	☐		☐			X	X	
T+M	20	Spindle Load monitoring	☐		☐			X	X	
T+M	22	Axis optimization with AST	☐		☐			X	X	X
T+M	23	Manual axis optimization	☐		☐			X	X	X
T+M	24	Compile cycles loaded	☐		☐			X	X	
T+M	25	Tools for Sinumerik integrate	☐		☐			X	X	
T+M	26	OPC UA Server commissioned	☐		☐			X	X	X

Copyright © Siemens AG 2020 All Rights Reserved.

Technology	Test No	Description	Tested	OEM Alternative / or Comments / not relevant	Tested	Reason for OEM Alternative	To Do	Valid for Sinumerik		
								ONE	840Dsl	828D
T+M	29	ILC Intelligent Load control	☐		☐			X	X	X
T+M	30	IDC Intelligent Dynamic Control	☐		☐			X	X	X
T+M	31	APC ECO	☐		☐			X	X	X
T+M	32	Nodding Compensation	☐		☐			X	X	X
T+M	33	IPC Data backup	☐		☐			X	X	
T+M	34	Deactivate Softkeys	☐		☐			X	X	
T+M	35	Create Machine data user views	☐		☐			X	X	
T+M	36	IPC/PCU Ghost image	☐		☐			X	X	
T+M	37	CNC Lock Feature	☐		☐					X
+M	38	Standard password changed	☐		☐			X	X	X
T+M	39	Machine registration data	☐		☐			X	X	X
T+M	40	License key back-up	☐		☐			X	X	X
T+M	41	1st Setup complete	☐		☐			X	X	X
T+M	42	2nd Setup complete	☐		☐			X	X	X
T+M	43	Warranty log file	☐		☐			X	X	X
T+M	44		☐		☐					
T+M	45		☐		☐					
T+M	46		☐		☐					
T+M	47		☐		☐					
T+M	48		☐		☐					
T+M	49		☐		☐					
T+M	50		☐		☐					
T+M	51		☐		☐					

Copyright © Siemens AG 2020 All Rights Reserved.