

# Blocks for Simatic S7 – TIA and Profibus-Connection

## SERVOSTAR® 300/400/600 and S700

### Table of contents

Notes for the TIA project „Sv14_v3c_V14_SP1_1516“ for Servostar® S300/400/600 and S700.....	2
TIA Device configuration .....	6
Servostar settings with DriveGui .....	8
Block overview .....	10
Notes .....	10
Input and output interface of Axis_01_FB (FB14) .....	11
Axis_01_FB (FB14).....	11
IF_DB (DB70) .....	11
Symbolic addressing from the Hardware-Interface to the Axis_01_FB ( FB14) - Interface.....	12
Complete overview of interface from Axis_01_FB (FB14).....	14
Program examples for converting the Servostar internal units to user / SI-units .....	18
GenCycleOB (OB1) - Network 11: "IF_DB".ActualPosition .....	18
GenCycleOB (OB1) – Netzwerk 12: "IF_DB".ActualSpeed.....	18
GenCycleOB (OB1) – Netzwerk 13: "IF_DB".ActualCurrent .....	18
GenCycleOB (OB1) – Netzwerk 20: "IF_DB".ActualPressure .....	19
GenCycleOB (OB1) – Netzwerk 21: "IF_DB".ActualFlow .....	19
Commissioning Axis_01_FB (FB14).....	20
S7-PLC – Switching ON/OFF .....	20
S7-PLC – Switching Run/Stop.....	20
iAck =1 ( Acknowledge ) .....	20
Initialization .....	20
Set operating mode.....	20
Enable operation .....	20
Jogging mode .....	21
Referencing run ( Homing run ) .....	21
Start of an stored EEPROM or RAM MotionTask .....	21
Start of the DirectMotionTask ( RAM and has the number 0 ) .....	21
Digital speed operating mode.....	23
Digital torque operating mode .....	23
Electronic gear operating mode.....	23
Servopump digital – velocity - operating mode .....	24
Macro program.....	25
Writing parameter .....	28
Reading parameter .....	28
Additional Profibus functions .....	30
Watch table_1 .....	30
Example for a sequence „Initialisation and Enable Servostar“ .....	31
Example for a sequence „Start MotionTasks in the Servostar“.....	33
Example for the “RAM MotionTask 201 in the Servostar“ .....	36
Blocks for Servostar parameters .....	38
Axis_01_WriteDataDB (DB20).....	38
Axis_01_ReadDataDB (DB21).....	38
Axis_01_Write (FB16).....	39
Axis_01_Read (FB17) .....	40
Axis_01_Compare (FB18).....	41
Tips and information:.....	42
Error messages and warnings from the blocks .....	42
Override via Profibus .....	42
Activating the sin^2 motion profile for DirectMotionTask No. 0.....	43
Bitcoding of error – PNU 1001 - ERRCODE .....	46
Bit coding of warnings and manufacturer specific state register – PNU 1002 - DRVSTAT .....	47
Switch ON and Start MotionTask timing diagram.....	48
Kill MotionTask timing diagram.....	49

## Notes for the TIA project „Sv14\_v3c\_V14\_SP1\_1516“ for Servostar<sup>®</sup> S300/400/600 and S700

### Record of Document Revisions:

Revision/Ausgabe	Comment/Bemerkungen	Author/Autor
A, 04/2018	first edition / Erstausgabe	KeH

### Trademarks

- Servostar is a registered trademark of Kollmorgen Corporation
- Simatic, Step7, TIA-Portal are registered trademarks of Siemens AG
- Profibus and ProfiNet are registered trademarks of PROFIBUS and PROFINET International(PI)

This document applies to the Profibus-DP connection of the Kollmorgen servo amplifiers Servostar<sup>®</sup> S300/400/600 and S700 to a Simatic S7-1500 PLC controller.

In the following text, the servo amplifiers Servostar<sup>®</sup> S300/400/600 and S700 are simply referred to as „Servostar“.



**Never use the S7-project „Sv14\_v3c\_V14\_SP1\_1516“ without modifications in an application.  
The S7 project „Sv14\_v3c\_V14\_SP1\_1516“ is an example how the Servostar can be integrated in an S7 project. This project example must always be adapted to the existing application.**



**KOLLMORGEN Europe GmbH assumes not liability for damages and precludes all claims arising from the use of the S7- project „Sv14\_v3c\_V14\_SP1\_1516“ or program components from it.**



**Take care and follow the security advice of the supplier of the different components.**

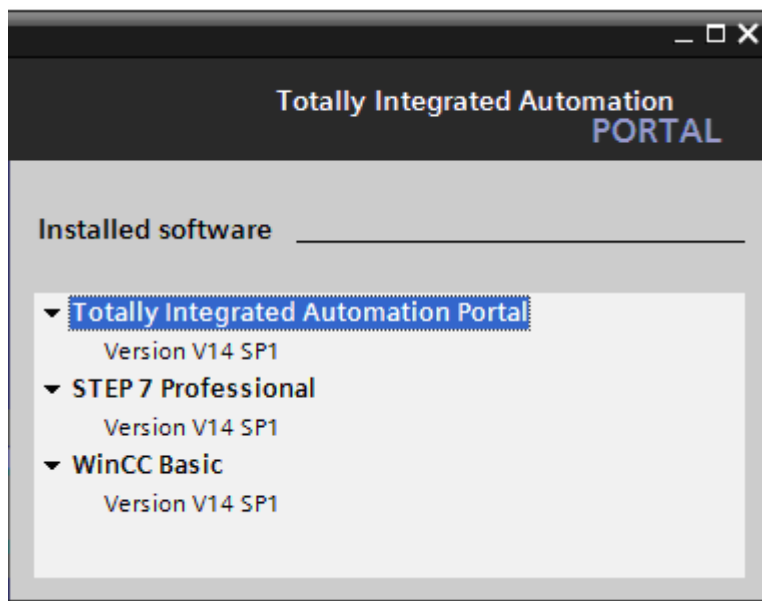
This manual assumes a good knowledge of Simatic TIA, controller and drive technology.

For properly behaviour from the complete Servoaxis consist of S7-1500 PLC and Servostar and motor the commissioning is acceptably realized !!!

All references to the manual refer to the technical description „srprof\_e.pdf“.

The terms and abbreviations used correspond to the Servostar setup software DriveGui or the manual in the English language.

The TIA project „Sv14\_v3c\_V14\_SP1\_1516“ was created in English with Mnemonics International and the version:



The TIA project „Sv14\_v3c\_V14\_SP1\_1516“ consists of the following components:

#### PLC\_1 ( CPU1516-3 PN/DP )

- Devices & networks configuration for a Simatic S7-CPU1516 – 6ES7 516-3AN00-0AB0
- program blocks
- PLC tags
- PLC data types
- Watch and force tables

#### Ungrouped devices ( Servostar )

- Slave\_1

The screenshot displays the Siemens TIA Portal interface for the project "Sv14\_v3c\_V14\_SP1\_1516". The main workspace shows a network diagram with two devices: "PLC\_1" (CPU 1516-3 PN/DP) and "Slave\_1" (Servostar). They are connected via a "PROFIBUS\_1" network. The "Network overview" table on the right lists the devices and their addresses.

Device	Type	Address in subnet
S71500/ET200MP-Station_1	S71500/ET200MP station	
PLC_1	CPU 1516-3 PN/DP	
PROFINET-Schnittstell...	PROFINET interface	192.168.0.1
Port_1	Port	
Port_2	Port	
PROFINET-Schnittstell...	PROFINET interface	192.168.1.1
Port_1	Port	
DP-Schnittstelle_1	DP interface	2
GSD-Geraet_1	GSD device	
Slave_1	Servostar	12

The "Properties" window for "Slave\_1" is open, showing the "General" tab. The "Catalog information" section includes:

- Short designation: Servostar
- Description: (koll045d.gsd)
- Article number:
- Firmware version: Z01
- GSD file: koll045d.gsd

## System and clock memory – Enable the use of system / clock memory byte – PLC Tags are default

**System and clock memory**

**System memory bits**

☒ Enable the use of system memory byte

Address of system memory byte (MBx): 1

First cycle: %M1.0 (FirstScan)

Diagnostic status changed: %M1.1 (DiagStatusUpdate)

Always 1 (high): %M1.2 (AlwaysTRUE)

Always 0 (low): %M1.3 (AlwaysFALSE)

**Clock memory bits**

☒ Enable the use of clock memory byte

Address of clock memory byte (MBx): 0

10 Hz clock: %M0.0 (Clock\_10Hz)

5 Hz clock: %M0.1 (Clock\_5Hz)

2.5 Hz clock: %M0.2 (Clock\_2.5Hz)

2 Hz clock: %M0.3 (Clock\_2Hz)

1.25 Hz clock: %M0.4 (Clock\_1.25Hz)

**PLC tags**

Name	Tag table	Data type	Address	Retain	Access	Write	Visibl...
System_Byte	Standard-Variab...	Byte	%MB1				
Clock_Byte	Standard-Variablen...	Byte	%MB0				
FirstScan	Standard-Variablen...	Bool	%M1.0				
DiagStatusUpdate	Standard-Variablen...	Bool	%M1.1				
AlwaysTRUE	Standard-Variablen...	Bool	%M1.2				
AlwaysFALSE	Standard-Variablen...	Bool	%M1.3				
Clock_10Hz	Standard-Variablen...	Bool	%M0.0				
Clock_5Hz	Standard-Variablen...	Bool	%M0.1				
Clock_2.5Hz	Standard-Variablen...	Bool	%M0.2				
Clock_2Hz	Standard-Variablen...	Bool	%M0.3				
Clock_1.25Hz	Standard-Variablen...	Bool	%M0.4				
Clock_1Hz	Standard-Variablen...	Bool	%M0.5				
Clock_0.625Hz	Standard-Variablen...	Bool	%M0.6				
Clock_0.5Hz	Standard-Variablen...	Bool	%M0.7				
System_1_4	Standard-Variablen...	Bool	%M1.4				
System_1_5	Standard-Variablen...	Bool	%M1.5				
System_1_6	Standard-Variablen...	Bool	%M1.6				
System_1_7	Standard-Variablen...	Bool	%M1.7				
<Add new>							



## TIA Device configuration

Among other things please note this settings:

- \* symbolic System constants – Hardware identifier
  - see also PLC tags > show all tags > System constants
- \* Input Address / Output Address
- \* GSD-file-version
- \* Subnet, Address and Transmission speed
  - The Servostar adjust itself automatically to the transmission speed ( baud rate )
- \* Watchdog

### symbolic System constants – Hardware identifier

The screenshot shows the Siemens TIA Portal interface. The 'Project tree' on the left lists the project structure, including 'Sv14\_v3c\_V14\_SP1\_1516' and 'PLC\_1 [CPU 1516-3 PN/DP]'. The 'Network view' in the center shows a connection between 'PLC\_1' and 'Slave\_1 Servostar' via 'PROFIBUS\_1'. The 'Network overview' on the right lists the devices and their addresses. The 'GSD-Geraet\_1 [Device]' properties window is open, showing the 'System constants' tab with a table of hardware identifiers.

Name	Type	Hardware identi.	Used by	Comment
Slave_1-DPSlave	Hw_DpSlave	259	PLC_1	
Slave_1-Head	Hw_Interface	258	PLC_1	
Slave_1-4_Wort_AE_AA_PKW	Hw_SubModule	261	PLC_1	
Slave_1-6_Wort_AE_AA_PZD	Hw_SubModule	262	PLC_1	

## Input Address / Output Address and GSD-file-version

**Device overview**

Module	Rack	Slot	I address	Q address	Type
Slave_1	0	0			Servostar
4 Wort AE_AA_PKW	0	1	256...263	256...263	4 Wort AE/AAKo...
6 Wort AE_AA_PZD	0	2	264...275	264...275	6 Wort AE/AAKo...
	0	3			
	0	4			
	0	5			
	0	6			
	0	7			
	0	8			
	0	9			
	0	10			
	0	11			
	0	12			

**Slave\_1 [Module] Properties**

**General**

**Catalog information**

Short designation: Servostar

Description: (koll045d.gsd)

Article number:

Firmware version: Z01

GSD file: koll045d.gsd

Install date: Wednesday, September 13, 2017 12:18

Installed by:

## Subnet, Address and Transmission speed

**Slave\_1 [Module] Properties**

**General**

**PROFIBUS address**

Interface networked with

Subnet: PROFIBUS\_1

Add new subnet

**Parameters**

Address: 12

Highest address: 126

Transmission speed: 1.5 Mbps

## Watchdog

**Slave\_1 [Module] Properties**

**General**

**Watchdog**

☒ Watchdog activated

## Servostar settings with DriveGui

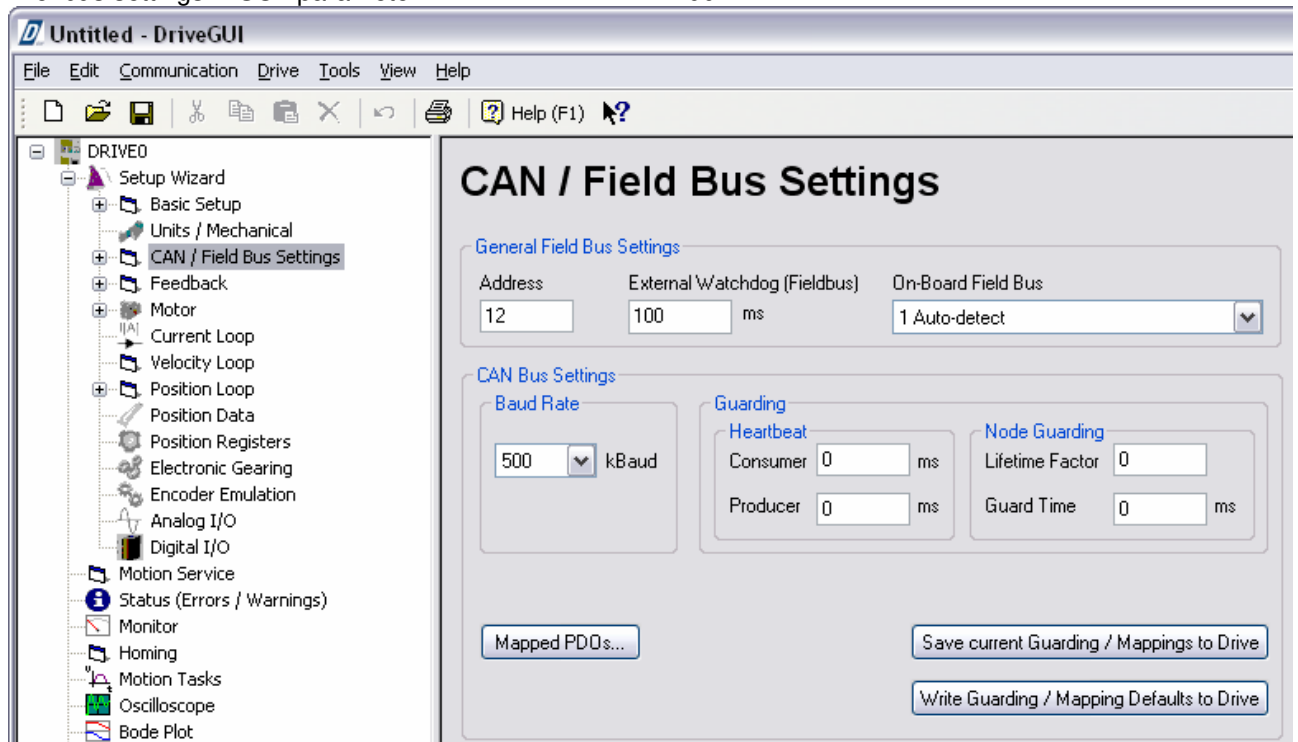
For commissioning the Servostar S400 / S600 the Software Drive is used.  
It is not described here more in detail.



For commissioning the Servostar S300 / S700 the Software DriveGui is used.  
DriveGui-Version:

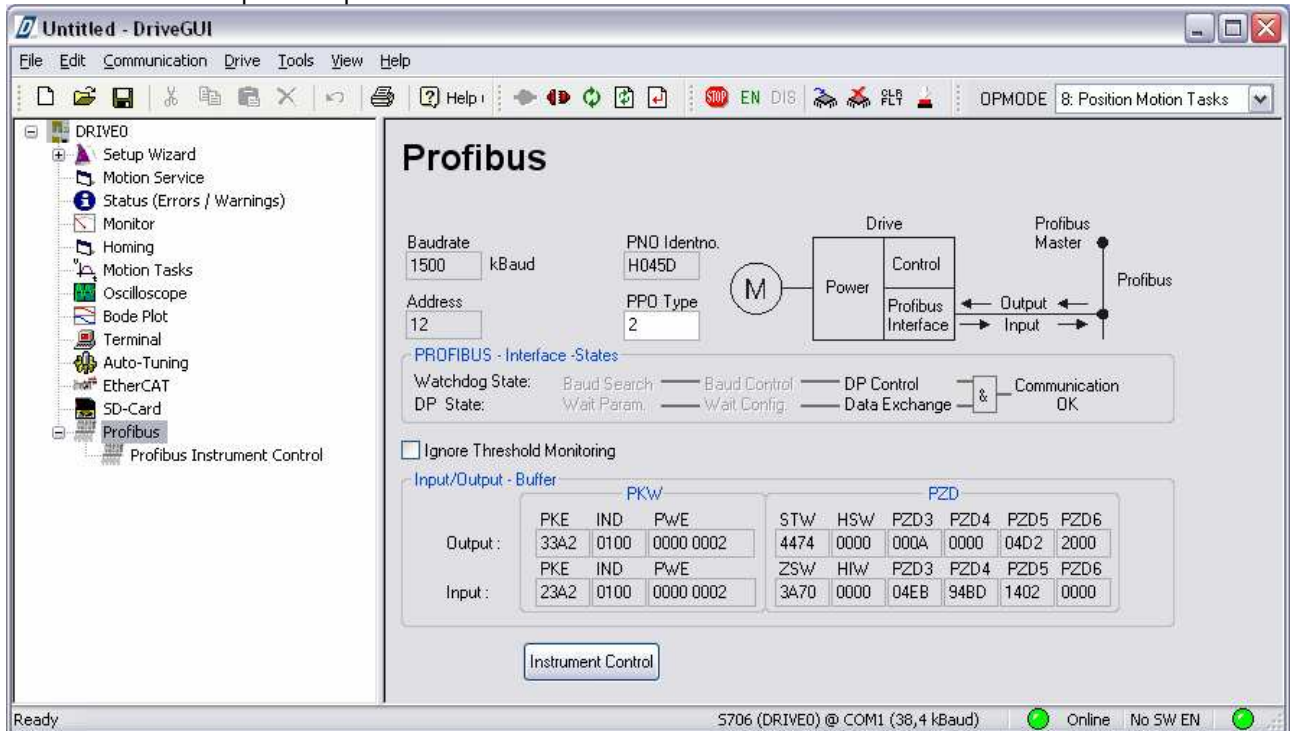


Profibus settings: ASCII parameter ADDR 12 – EXTWD 100

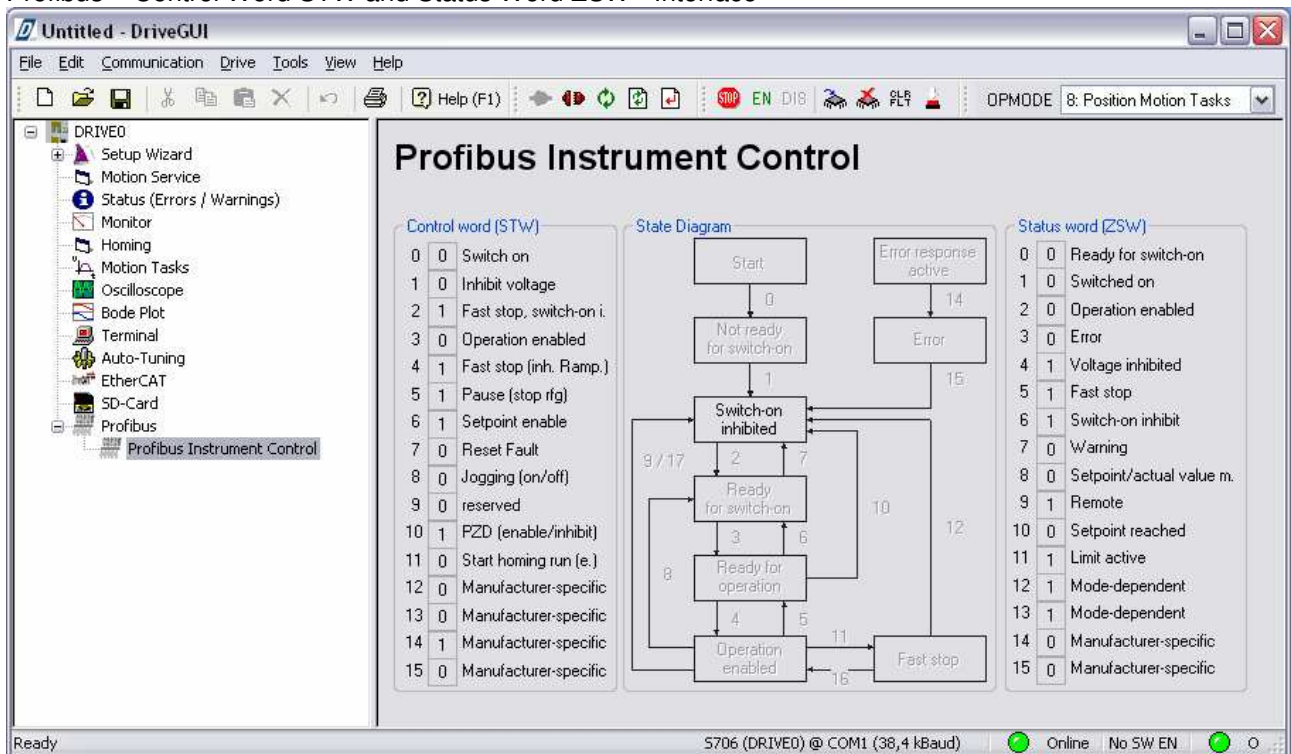




## Profibus – Data – Input / Output or Send / Receive – PKW and PZD - Interface



## Profibus – Control Word STW and Status Word ZSW - Interface



## Block overview

Axis_01_FB	FB14/DB14	Handling block between S7-PLC and Servostar
Axis_01_Write	FB16/DB16	Block for the FB14 to write PNU data to Servostar
Axis_01_WriteData	DB20	Contains the PNU data that are written to Servostar with FB16
Axis_01_Read	FB17/DB17	Block for the FB14 to read PNU data from Servostar
Axis_01_ReadData	DB21	Contains the PNU data that are read from Servostar with FB17
Axis_01_Compare	FB18/DB18	Module for PNU data comparison, e.g. DB20 and DB21
IF_DB	DB70	Interface data block e.g. for an operator panel / touch panel

All blocks can be renamed or renumbered upon demand.  
 Afterwards, "Compile -> Software (rebuild all)" must be performed.  
 The resulting conflicts may have to be corrected manually in the blocks.

The Axis\_01\_FB (FB14) uses the following blocks:

D\_ACT\_DP – Deactivating and Activating DP Slave / ProfiNet IO device  
 DPRD\_DAT – Reading consistent data of a DP Slave / ProfiNet IO device  
 DPWR\_DAT – Writing consistent data of a DP Slave / ProfiNet IO device  
 BLKMOV – Copy memory area  
 FILL – Fill memory area  
 and the system block:  
 BITSUM ( FC99 ) – Count number of set bits

The Axis\_01\_Write (FB16), Axis\_01\_Read (FB17), Axis\_01\_Compare (FB18) uses the system blocks:  
 ReadFromArrayDB ( FC901 ) – Read from ARRAY data block  
 WriteToArrayDB ( FC902 ) – Write to ARRAY data block

## Notes

- The Axis\_01\_FB (FB14) is programmed as multi instance and, therefore, can be called multiple times in FBs as subprogram block without separate instance data block.
- If Axis\_01\_FB (FB14) is not implemented as multi instance, then Axis\_01\_FB (FB14) must be called up for every Servostar amplifier with a separate instance data block DB.
- Axis\_01\_FB (FB14) is programmed in FBD ( function chart ) with mnemonics International and symbols with comments in ENGLISH.
- The PLC data types enables an uniform database and an object-oriented programming in TIA.
- The STRUCT data types enables simple addressing of entire data areas via pointer.
- The S7 project „Sv14\_v3c\_V14\_SP1\_1516“ contains a completely functional S7-PLC program for Servostar - Profibus.
- The S7 project „Sv14\_v3c\_V14\_SP1\_1516“ is migrated into the Simatic TIA-Portal from the Step7 project „Sv14\_v3c“ and afterwards reprogrammed completely in FBD.
- For S7-1200 PLC is the completely functional S7 project “Sv14\_v3c\_V14\_SP1\_1214C” available.

With a S7-1500 PLC and a Servostar with Profibus option card and motor, the complete Servoaxis can be quickly and easily commissioned using the Watch table\_1.

## Input and output interface of Axis 01\_FB (FB14)

The interface of the blocks FB14, FB16, FB17, FB18 and DB20, DB21, DB70 uses PLC data types.

M\_Axis\_01\_Request  
M\_Axis\_01\_MaMsg – Machine messages / Error messages  
M\_Axis\_01\_State  
M\_Axis\_01\_InData  
M\_Axis\_01\_OutData  
M\_Axis\_01\_PnuStruct  
M\_Axis\_01\_RcvSend - Receive and Send Profibus

The PLC data types ensures that the data structure in the S7 project has an uniform common database. A modification at the data structure is done once in the PLC data types. The entire data structure of the S7 project is automatically updated with "Compile -> Software (rebuild all)".

Please check afterwards that the Initial values from the PLC data types are transferred to the Actual values from the DB. If not then for this DB a separate initialization or correct adjustment from the block properties is required.

## Axis 01\_FB (FB14)

Input variables are preceded by an "i" (Input) and output variables by an "o" (Output).  
This makes the program code easier to read.

VAR_INPUT		
iData	M_Axis_01_InData	
iRequest	M_Axis_01_Request	
iPnu	Struct	
Write	M_Axis_01_PnuStruct	
..Read	M_Axis_01_PnuStruct	
...		
VAR_OUTPUT		
oMaMsg	M_Axis_01_MaMsg	
oState	M_Axis_01_State	
oData	M_Axis_01_OutData	
oAxis	M_Axis_01_RcvSend	
...		

## IF DB (DB70)

ToMachine	IfGroupToUnit	
FromMachine	IfUnitToGroup	
Request	M_Axis_01_Request	
State	M_Axis_01_State	
...		
iData	M_Axis_01_InData	
oData	M_Axis_01_OutData	
oMaMsg	M_Axis_01_MaMsg	
oAxis	M_Axis_01_RcvSend	
...		

## Symbolic addressing from the Hardware-Interface to the Axis\_01\_FB (FB14) - Interface

TIA generate automatically symbolic System constants – Hardware identifiers  
 – see also PLC tags > show all tags > system constants

The numeric address can modified unintentional by TIA after “Compile -> Hardware (rebuild all )”.  
 Therefore use in the program always the symbol address !!!

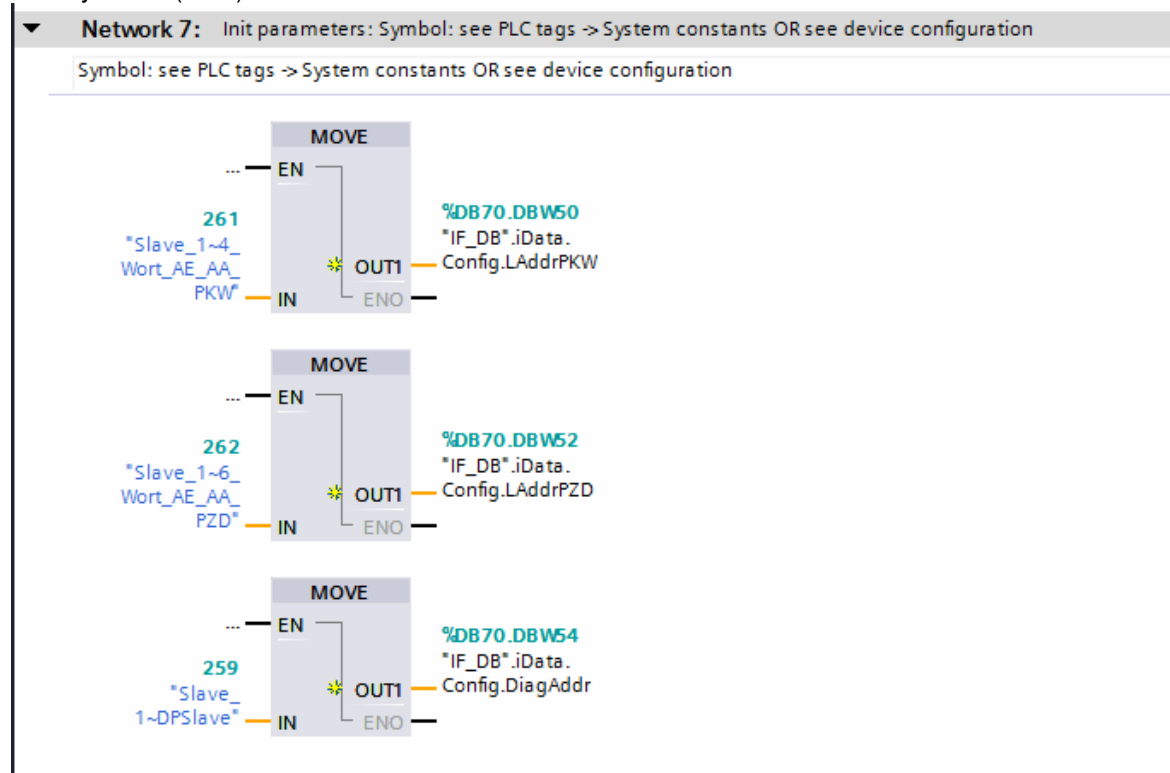
The Input Address / Output Address is adjusted in the Devices & networks configuration  
 See chapter TIA-Device configuration

With drag&drop or copy&paste you can insert the symbolic address from the system constants or from the device configuration easy to the Input of the MOVE

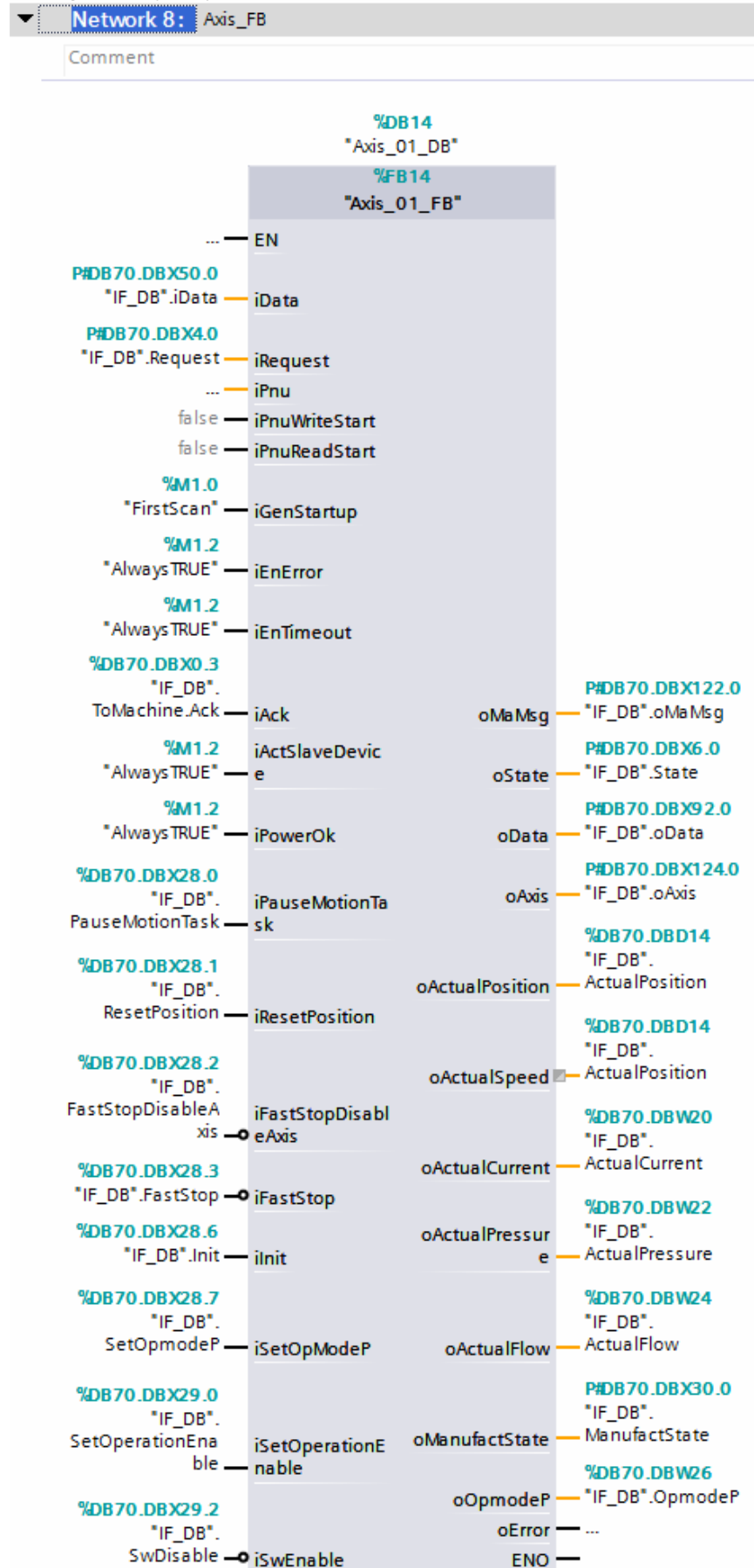
Every Servostar need the mapping from his device configuration to the Axis\_01\_FB (FB14)  
 by the DataType HardwareInterface.

Note: With Copy&Paste from parts from a project into another project the symbolic address is not always the same. The TIA device configuration generate then e.g. an extension xxxPKW\_1, xxxPZD\_1, xxxSlave\_1.  
 Therefore check always at PLC tags > show all tags > system constants that all symbols are definitely and unique assigned. In the TIA device configuration or in the plc program the mapping could be corrected and the unused symbol should afterwards deleted. If the mapping is executed correct the S7 PLC and the Servostar communicate over the Profibus and the Axis\_01\_FB (FB14) answer oState.CommunicationOk =1

### GenCycleOB (OB1)



## GenCycleOB (OB1)



## Complete overview of interface from Axis 01 FB (FB14)

VAR_INPUT		
iData	STRUCT	Data: HWConfig, OpMode, RefJogSpeed, MotionTask
Config	STRUCT	
LAddrPKE	HW_Inf	StartAddressIn in HW-Config periphery dez
LAddrPZD	HW_Inf	StartAddressOut in HW-Config periphery dez
DiagAddr	HW_Inf	DiagnosticAddress in HW-Config periphery hex
TO_Reference	Time	Timeout referencing
TO_Position	Time	Timeout positioning
OpModeP	DWORD	PNU 930 (2 =MotionTask, 1 =VelocityDigital, -2 =TorqueDigital ... )
JogSpeed	WORD	PNU 1889
RefSpeed	WORD	PNU 1896
MotionTask	STRUCT	
Number	WORD	0 =DirectMotionTask
DirectSpeed	DWORD	PNU 1791
DirectPosition	DWORD	PNU 1790
DirectType	WORD	PNU 1785
DigitalSpeed	STRUCT	
Ncmd	WORD	PNU1886 - Ncommand ( Ncmd16 = Ncmd * 2 <sup>15</sup> / VOSPD )
DigitalTorque	STRUCT	
Icmd	WORD	PNU1870 - Icommand ( I[A] = Icmd * IpeakAmplifier[A] / 3280 )
DigitalPump	STRUCT	
QPRcmd	WORD	QPR command ( pressure 1 <-> 10 mbar )
QFcmd	WORD	QF command ( flow 1 <-> 0,1 l/min )
iRequest	STRUCT	Requests: Ref Pos StartStopCancelMotionTask Jog
StartRef	BOOL	Start reference movement, static 1
StartIcmd	BOOL	Start I command digital torque, P4, static 1
StartMotionTask	BOOL	Start motion task ( direct motion task =0 ), P4
StopMotionTask	BOOL	Stop motion task, P4->P3
CancelMotionTask	BOOL	Cancel motion task, P3
StartNcmd	BOOL	Start N command digital speed, P4, static 1
JogPlus	BOOL	Jog positive, static 1
JogMinus	BOOL	Jog negative, static 1
MacroInput	BOOL	MacroProgramInput, static 1, PROSTW & 0x200
StartQcmd	BOOL	Start Q command digital speed, P4, static 1
Res_1_2	BOOL	
Res_1_3	BOOL	
Res_1_4	BOOL	
Res_1_5	BOOL	
Res_1_6	BOOL	
Res_1_7	BOOL	
iPnu	STRUCT	
Write	STRUCT	
Number	WORD	
Index	WORD	
Value	DWORD	
Read	STRUCT	
Number	WORD	
Index	WORD	
Value	DWORD	
iPnuWriteStart	BOOL	Request PnuWriteStart pulse 0->1
iPnuReadStart	BOOL	Request PnuReadStart static 1
iGenStartup	BOOL	Startup cycle after PLC startup
iEnError	BOOL	Enable error messages
iEnTimeout	BOOL	Enable timeout movement referencing and positioning
iAck	BOOL	Acknowledge WarningsErrors
iActSlaveDevice	BOOL	Activate Slave or Device
iPowerOk	BOOL	All powersupplies are ok
iPauseMotionTask	BOOL	Pause for Motion Task, P4
iResetPosition	BOOL	Reset position, set ActualPosition to RefPosition ( ROFFS )
iFastStopDisableAxis	BOOL	FastStop with disable axis, P4->P1
iFastStop	BOOL	FastStop without disable axis, P4->P11
iInit	BOOL	Initialize axis with disable axis
iSetOpModeP	BOOL	Set operating mode Profibus
iSetOperationEnable	BOOL	Set axis state machine to P4_OperationEnabled
iSwEnable	BOOL	Software enable axis



VAR_OUTPUT		
oMaMsg	STRUCT	Error messages
ErrTO_Ref	BOOL	Error timeout reference
ErrTO_Pos	BOOL	Error timeout positioning
ErrNoReferenceSet	BOOL	Error if reference is not set and request
ErrWrongOpMode	BOOL	Error wrong operation mode selected and request
ErrActSlaveDevice	BOOL	Error activating slave or device
ErrCfgInput	BOOL	Error configuration input
ErrRcv	BOOL	Error receiving data
ErrCfgOutput	BOOL	Error configuration output
ErrSend	BOOL	Error sending data
ErrAxis	BOOL	Error from axis
ErrReadWrite	BOOL	Error request Read and Write together
ErrNotEnabled	BOOL	Error if not enabled and request
ErrRes_1_4	BOOL	
ErrRes_1_5	BOOL	
ErrRes_1_6	BOOL	
ErrCmd	BOOL	Error more than one request command active
oState	STRUCT	AxisState
ReferencingActive	BOOL	Axis is referencing
JoggingActive	BOOL	Axis is jogging
VelocityIsZero	BOOL	Axis velocity is zero
MotionTaskActive	BOOL	Axis motion task is active
InPosition	BOOL	Axis is in position
Res_0_5	BOOL	
Res_0_6	BOOL	
Res_0_7	BOOL	
ReferenceOk	BOOL	Axis is referenced
CommunicationOk	BOOL	Axis communication Profibus is ok
InitOK	BOOL	Axis initialisation is ok
InitError	BOOL	Axis initialisation error
OpModeP_Ok	BOOL	Axis opmode Profibus is ok
OpModeP_Error	BOOL	Axis opmode Profibus error
WarningActive	BOOL	Axis warning active
WarnPositionError	BOOL	Axis warning position error
P0_NotReadySwitchOn	BOOL	state diagram
P1_SwitchOnInhibited	BOOL	state diagram
P2_ReadyForSwitchOn	BOOL	state diagram
P3_ReadyForOperation	BOOL	state diagram
P4_OperationEnabled	BOOL	state diagram
P11_FastStopActive	BOOL	state diagram
P13_ErrorReaction	BOOL	state diagram
P14_ErrorActive	BOOL	state diagram
PnuWriteOk	BOOL	Pnu write done and ok
PnuWriteError	BOOL	Pnu write not done and error
PnuReadOk	BOOL	Pnu read done and ok
PnuReadError	BOOL	Pnu read not done and error
Res_3_4	BOOL	
Res_3_5	BOOL	
Res_3_6	BOOL	
Res_3_7	BOOL	
ResponseTelegram_PKW_PWE	DWORD	Axis response telegram after Pnu Rcv or Send
oData	STRUCT	AxisData
Canceled	STRUCT	
DirectMotionTask	STRUCT	
STW	WORD	
Speed	DWORD	
Position	DWORD	
TaskType	WORD	
ActualSpeed	INT	
ActualPosition	DINT	
StartPosition	DINT	
PnuRead	STRUCT	
Number	WORD	
Index	WORD	
Value	DWORD	

oAxis	STRUCT	
Rcv	STRUCT	
PKW		
PKE	WORD	
IND	WORD	
PWE1	WORD	
PWE2	WORD	
PZD	STRUCT	
ZSW	STRUCT	
SetpointActualValMonitor	BOOL	only in Opmode POSITION: Following error
Remote	BOOL	not working, set to 1
SetpointReached	BOOL	only in Opmode POSITION: At Position
LimitActive	BOOL	at the moment not working
ModeDependentx	BOOL	used in ASCII-Modus
ModeDependenty	BOOL	used in ASCII-Modus
ModeDependentz	BOOL	used in ASCII-Modus
Reserved	BOOL	reserved
ReadyForSwitchOn	BOOL	
SwitchedOn	BOOL	
OperationEnabled	BOOL	
Error	BOOL	see ASCII-Kommando ERRCODE
VoltageInhibit	BOOL	
FastStop	BOOL	
SwitchOnInhibit	BOOL	
Warning	BOOL	see ASCII-Kommando STATCODE
HIW	WORD	
PZD3	WORD	
PZD4	WORD	
PZD5	WORD	
PZD6	WORD	
Send	STRUCT	
PKW	STRUCT	
PKE	WORD	
IND	WORD	
PWE1	WORD	
PWE2	WORD	
PZD	STRUCT	
STW	STRUCT	
JoggingOnOff	BOOL	OpMode dependent
Reserved	BOOL	
PZDenableInhibit	BOOL	
StartHomingRun	BOOL	OpMode dependent
ResetPosition	BOOL	
AckWarning	BOOL	Acknowledge warnings
MoTaskDirectOrMoTaskNr	BOOL	Only in OpModes Position: 0=MotionTaskNumber 1=DirectMotionTask
DigitalRevolutionSpeed	BOOL	OpMode dependent, digital velocity
SwitchOn	BOOL	
InhibitVoltage	BOOL	
FastStopSwitchOn	BOOL	1>0Axis FastStopWithEmgyRamp, AxisWillDisabled- STOPMODEECDIS
OperationEnabled	BOOL	
FastStopWithEmgyRamp	BOOL	1>0 Axis fast stop with emergency ramp
PauseStopRfg	BOOL	OpMode dependent, 1>0 Axis stop
SetpointEnable	BOOL	OpMode dependent
ResetFault	BOOL	Reset errors
HSW	WORD	
PZD3	WORD	
PZD4	WORD	
PZD5	WORD	
PZD6	WORD	

oActualPosition	DINT	Actual position, valid if PZD channel is active 1*)
oActualSpeed	INT	Actual speed, valid if PZD channel is active, OpmodePb 1,2
oActualCurrent	INT	Actual current, valid if PZD channel is active, OpmodePb -2
oActualPressure	INT	Actual pressure, valid if PZD channel is active, OpmodeP -7
oActualFlow	INT	Actual flow, valid if PZD channel is active, OpmodeP -7
oManufactState	STRUCT	comment valid for OpMode positioning
Pos3reached	BOOL	
Pos4reached	BOOL	
AxisInternalInitReady	BOOL	
x0_3	BOOL	
VelocityIsZero	BOOL	
SafetyRelayOpen	BOOL	
AxisEnabled	BOOL	
AxisErrorExist	BOOL	
MotionTaskActive	BOOL	
ReferenceDoneAndOK	BOOL	
ReferenceSwitchOn	BOOL	
InPositionWindow	BOOL	
LatchPositionDone	BOOL	
x1_5	BOOL	
Pos1reached	BOOL	
Pos2reached	BOOL	
oOpmodeP	WORD	Actual OpmodeP ProfibusDP ( Pnu 930)
oError	BOOL	Error is active

1\*) The actual position in the Servostar is transferred in the process data only in incremental internal units to the S7-PLC ( $2^{20}$  incr. per motor revolution). The conversion from internal units to user units can be performed in the S7-PLC, depending on the position resolution defined in the Servostar by the DriveGui software.

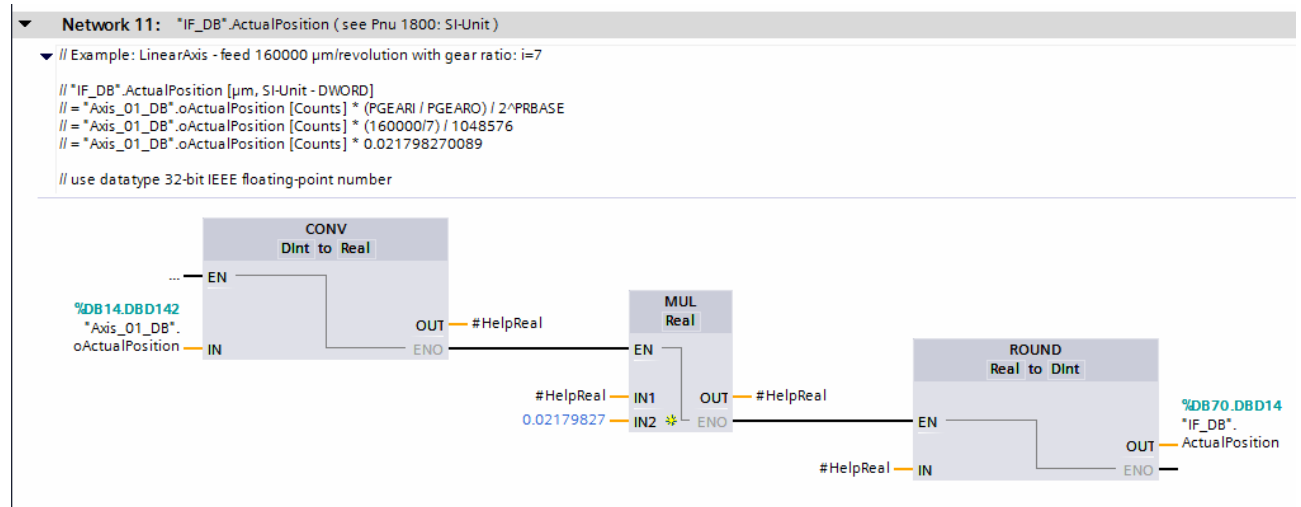
Example with PRBASE 20:

Resolution = 5000 incr / 3 revolutions

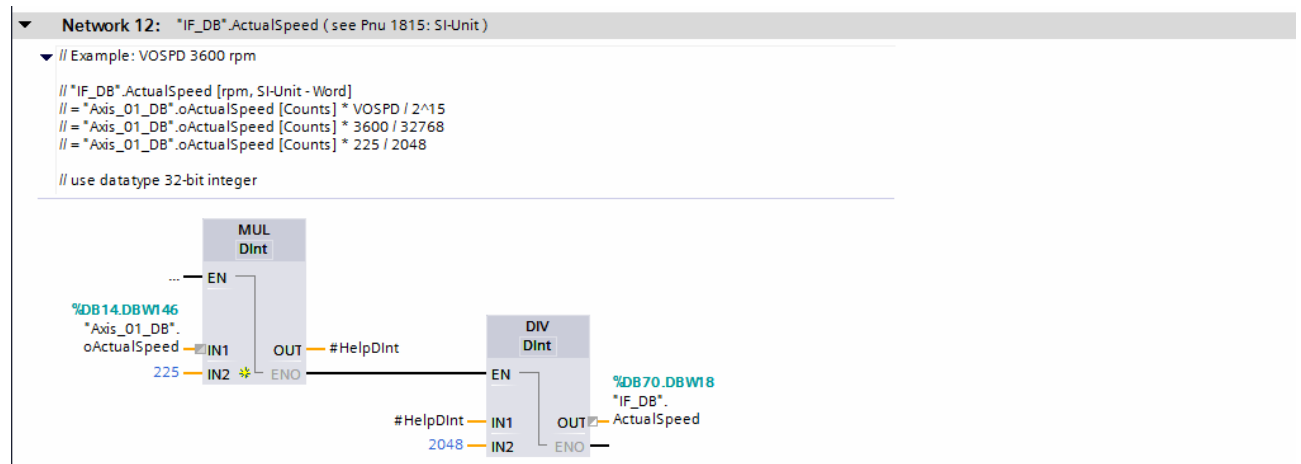
=> Position in user units = Actual\_Position x 5000 / (3 x  $2^{20}$ )

## Program examples for converting the Servostar internal units to user / SI-units

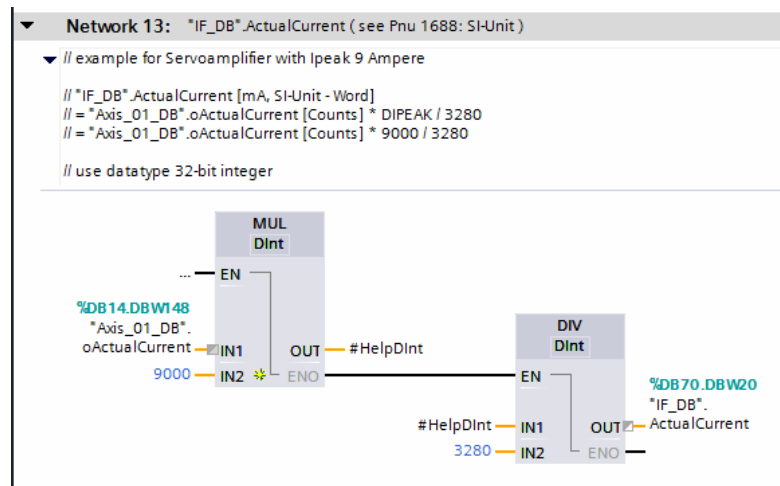
### GenCycleOB (OB1) - Network 11: "IF\_DB".ActualPosition



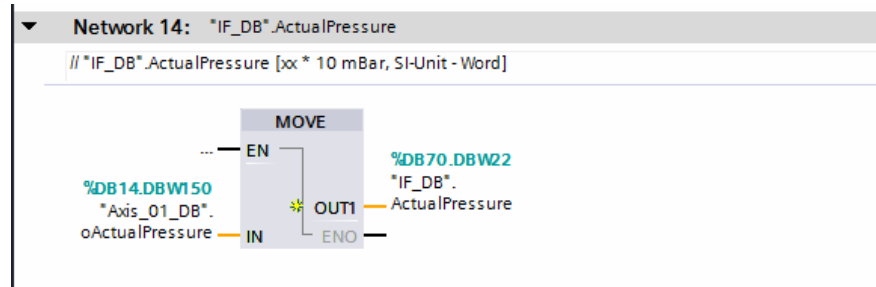
### GenCycleOB (OB1) – Netzwerk 12: "IF\_DB".ActualSpeed



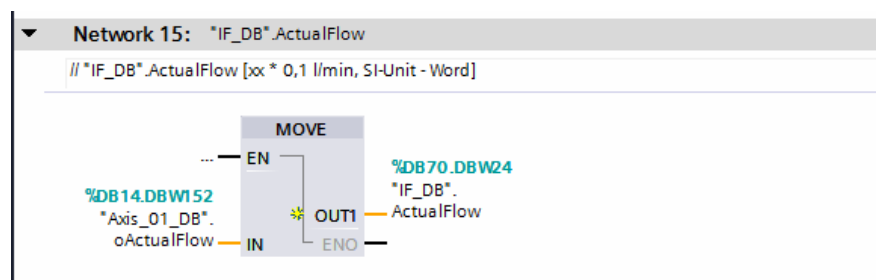
### GenCycleOB (OB1) – Netzwerk 13: "IF\_DB".ActualCurrent



### GenCycleOB (OB1) – Netzwerk 20: "IF\_DB".ActualPressure



### GenCycleOB (OB1) – Netzwerk 21: "IF\_DB".ActualFlow



## Commissioning Axis 01\_FB (FB14)

### S7-PLC – Switching ON/OFF

If the S7-PLC Profibus-Master fails, the watchdog timeout starts in the Servostar (EXTWD) and generate the warning „n04 – node guarding ( watch dog )“.

After startup, the S7-PLC performs a software reset (FirstScan - M1.0) and deletes all warnings and error messages and states in the PLC program, but not in the Servostar.

If the Servostar boot up faster than the S7-PLC the Servostar generate the warning n04.

With the ASCII-Parameter S1DLY it is possible to increase the boot up time e.g. to avoid the warning n04.

After acknowledge the warning n04 the Servostar is able for operation.

### S7-PLC – Switching Run/Stop

If the state change from STOP -> RUN the S7-PLC performs a software reset (FirstScan - M1.0) and deletes all warnings and error messages and states in the PLC program, but not in the Servostar.

### iAck =1 ( Acknowledge )

Reset the warnings and errors in the Servostar and in the S7-PLC program in the blocks

Axis\_01\_FB (FB14), Axis\_01\_Write (FB16), Axis\_01\_Read (FB17), Axis\_01\_Compare (FB18).

### Initialization

During initialization, a zero telegram ( Send.PKW and Send.PZD ) is transmitted from S7-PLC to Servostar.

The Servostar disable immediately and the state machine change to oState.P1\_SwitchOnInhibited =1

- Set iInit =1 ( a pulse is generated internally )
- Wait until oState.InitOk =1
- If oState.InitOk is not =1 or oState.InitError =1 the initialization was not successful.
- For the error cause see Servostar receive telegram.
- The initialization is monitored with a timeout of 1 second.

### Set operating mode

Programmed operating mode: Positioning, Digital velocity, Digital torque, Electronic gear, Digital ServoPump  
After 24VDC switch-on, the Servostar is always in the safe internal operation mode -126 and “locked”.

Via Profibus there is a communication established between the S7-PLC and Servostar only from STW and ZSW and the RCV - PZD2, PZD3, PZD4, PZD5, PZD6 ( receive telegram ) contains all the value “0”.

The S7-PLC must first change and recheck the operating mode.

- Write the desired operating mode in iData.Config.OpModeP  
( see manual: Positioning: 2 ; Digital velocity: 1 ; Digital torque: -2 ;  
Electronic gear: -4 , Digital Servopump: -7 )
- Set iSetOpModeP =1 ( a pulse is generated internally )
- Wait until oState.OpModeP\_Ok = 1
- The activated operating mode will be visible in oOpModeP
- Now make the recheck for the desired OpmodeP
- If oState.OpModeP\_Ok is not =1 or oState.OpModeP\_Error =1  
or oOpModeP is not the desired OpmodeP then setting the operating mode was not successful.
- For error cause see Servostar receive telegram.
- Now the RCV - PZD2, PZD3, PZD4, PZD5, PZD6 is displayed depending on the operating mode  
with values from e.g. oActualPosition, oActualSpeed, oManufactState.
- Setting the operating mode is monitored with a timeout of 1 sec.

### Enable operation

After 24VDC switch-on, the Servostar is in oState.P1\_SwitchOnInhibited =1.

iFastStop, iFastStopDisableAxis must be =0 and iSwEnable must be =1.

At the Servostar, HardwareEnable must be =1 and if present AS/STO-Enable must be =1.

Only S700: A build in safety card must be in state „RUN“.

The DC link voltage is applied and the display of Servostar shows „Pxx“ without warnings or errors.

- Set iSetOperationEnable =1 ( a pulse is generated internally )
- Wait until oState.P4\_OperationEnabled =1
- If oState.P4\_OperationEnabled is not =1, the enable operation was not successful.

The Servostar is now enabled and able to generate force / torque and move a motor.



Requirement for operation mode positioning

- The operation is enabled (oState.P4\_OperationEnabled =1)
- AS/STO- and Hardware-Enable is present
- No warning and error-free and the display from Servostar shows „Exx“
- The operation mode positioning (oState.OpModeP\_Ok = 1 und oOpModeP = 2 dec ) is activated

Jogging mode

- Write the desired jogging speed to "iData.Config.JogSpeed"
- Set iRequest.JogPlus =1 -> The motor moves positive and oState.JoggingActive =1
- Set iRequest.JogMinus =1 -> The motor moves negative and oState.JoggingActive =1

Referencing run ( Homing run )

The referencing type is normally adjusted with DriveGui and stored in the Servostar.

- Write the desired referencing run speed to "iData.Config.RefSpeed"
- Set iRequest.StartRef =1 -> The motor move and oState.ReferencingActive =1
- Wait until oState.ReferenceOk =1 and oState.ReferencingActive again =0
- Set iRequest.StartRef =0 – The Servostar is referenced ( homed ).

With iEnTimeout =1 und iData.Config.TO\_Reference xxx ms [TIME] the referencing run is monitored with a timeout.

Start of an stored EEPROM or RAM MotionTask

Additional requirement: The Servostar is referenced.

The DriveGui is use to create MotionTasks and store them in the Servostar EEPROM.

Only the parameters of the RAM MotionTask can be changed with the S7-PLC in the Servostar even in the state "P4\_OperationEnabled" and while the motor is moving (see ASCII - MTMUX)

- Write the number of the MotionTask to iData.MotionTask.Number
- Set iRequest.StartMotionTask =1 ( a pulse is generated internally )
- Wait until oState.InPosition =0 and oState.MotionTaskActive =1
- Wait until oState.InPosition =1 and oState.MotionTaskActive =0

The Servostar executed the MotionTask. With iEnTimeout =1 and iData.Config.TO\_Position xxx ms [TIME], the MotionTask run is monitored with a timeout.

Actual position, actual speed and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

Start of the DirectMotionTask ( RAM and has the number 0 )

Additional requirement: The Servostar is referenced.

- Write 0 to iData.MotionTask.Number
- Write target position to iData.MotionTask.DirectPosition
- Write target speed to iData.MotionTask.DirectSpeed
- Write direkt type to iData.MotionTask.DirectType
- Set iRequest.StartMotionTask =1 ( a pulse is generated internally )
- Wait until oState.InPosition =0 and oState.MotionTaskActive =1
- Wait until oState.InPosition =1 and oState.MotionTaskActive =0

The Servostar executed the MotionTask. With iEnTimeout =1 and iData.Config.TO\_Position xxx ms [TIME], the DirectMotionTask run is monitored with a timeout.

Actual position, actual speed and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

While the MotionTask is running, the next MotionTask can already be copied to iData.MotionTask with his record and then started immediately with Set iRequest.StartMotionTask =1 (pulse).

With BLKMOV and pointer the complete Data.MotionTask [STRUCT] can be copied easily.

See "Switch ON and Start MotionTask timing diagram"

A started MotionTask can be paused with `iPauseMotionTask = 1`.  
The Servostar remain in `oState.P4_OperationEnabled = 1`.

A started MotionTask can be stopped with `iRequest.StopMotionTask = 1`.  
The Servostar change to `oState.P3_ReadyForOperation = 1`.  
A stopped MotionTask remains active in the controller, i.e. if the controller is switched back to `oState.P4_OperationEnabled = 1`, the MotionTask is continued.  
With `iSwEnable = 0`, the stopped MotionTask is killed, i.e. if the controller now is switched back to `oState.P4_OperationEnabled = 1`, the MotionTask is no longer continued.  
See: "Kill MotionTask timing diagram"

In the GenCycleOB (OB1) is an example to parameterize parts of a MotionTask and transmit with `Axis_01_WRITE (FB16)` from S7-PLC into the Servostar with the parameter channel.

Note: The ASCII parameter INPT0 PNU 1904 set the time in ms for which the signal `oManufactState.InPositionWindow` is reset at a defined time after the MotionTask start (see manual ch. VII.1).  
So the interface timing behaviour between the S7-PLC and the Servostar is always the same.

#### Note about positions and speeds – MotionTaskType O C – PNU 1785

0x0000 hex (bit 13 = 0)  
Absolute positioning with preset of speeds and positions only in incremental internal 32 bit and 16 bit units and trapezoidal motion profile.

0x2000 hex (bit 13 = 1)  
Absolute positioning with preset of speeds and positions in user units and trapezoidal motion profile.

0x2003 hex (bits 0, 1, 13 = 1)  
Relative positioning with preset of speeds and positions in user units and trapezoidal motion profile.

0x12000 hex ( bit 13 = 1 and bit 16 = 1 )  
Absolute positioning with preset of speeds and positions in user units and sin<sup>2</sup> motion profile.

0x10000 hex ( bit 16 )  
The Bit 16 cannot be set directly in the PZD channel.  
See Activating the sin<sup>2</sup> motion profile for DirectMotionTask

#### Note about Feedback with Wake&Shake (W&S)

Actual position, actual speed and the manufacturer state are cyclically transmitted in the RCV.PZD channel after Set operating mode is done ( `oState.OpModeP_Ok = 1` und `oOpModeP = xxx dec` ) and the Wake&Shake procedure is finished.

Remark: `n14 = 1` Scanning for MPHASE (e.g. `FBTYPE=7`)  
Is set while start-up of the drive and is cleared after the drive was enabled and MPHASE was determined with Wake&Shake.

If you receive an error from Set operating mode please enable first the Servostar ( `iSetOperationEnabled = 1` ) and then the Wake&Shake procedure will be execute.  
Start afterwards Set operating mode.

### Digital speed operating mode

The digital speed operating mode (oState.OpModeOk =1 and oOpmodeP =1 dec) is activated and the operation is enabled (oState.P4\_OperationEnabled =1).

- Write target speed to iData.DigitalSpeed.Ncmd (16 bit)
- Set iRequest.StartNcmd =1  
The motor moves with the preset target speed.  
The target speed can be changed at any time.
- Set iRequest.StartNcmd =0
- The motor decelerates via the set speed ramps until standstill.

Actual position, actual speed (16 bit) and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

The speed value (16 bit) is calculated using the following formula:

PNU 1886 - Ncommand ( Ncmd16 = Ncmd \* 2<sup>15</sup> / VOSPD )

### Digital torque operating mode

The digital torque operating mode (oState.OpModeOk = 1 and oOpmodeP = -2 dec) is activated and the operation is enabled (oState.P4\_OperationEnabled =1).

- Write target current to iData.DigitalTorque.lcmd
- Set iRequest.Startlcmd =1
- The Servostar injects the preset target current in the motor.
- The target current can be changed at any time.
- Set iRequest.Startlcmd =0 and the Servostar outputs the target current 0 Ampere.

Actual position, actual current and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

The current value is calculated using the following formula:

PNU 1870 - Icommand ( I[A] = lcmd \* IpeakAmplifier[A] / 3280 )

The ASCII parameter ICMDVLIM - PNU 1989 can be used to limit the speed of the motor to a maximum value. This prevents the motor from overspeeding if the load is too low.

### Electronic gear operating mode

The operation mode Electronic gear (oState.OpModeP\_Ok = 1 und oOpmodeP = -4 dec ) is activated and the operation is enabled (oState.P4\_OperationEnabled =1).

Actual position, actual speed and the manufacturer state are cyclically transmitted in the RCV.PZD channel.

### Servopump digital – velocity - operating mode

only for S300/S700 - see "Applikationsschrift – Servopumpe - s700\_servopumpe\_d.pdf"

The operation mode Servopump digital (oState.OpModeP\_Ok = 1 und oOpmodeP = -7 dez ) is activated and the operation is enabled (oState.P4\_OperationEnabled =1).

Note: While activating the opmode -7, automatically QENA set to 1.

By switching to another operation mode than -7, the Servopump will be deactivated (QENA=0).

- Write target pressure to iData.DigitalPump.QPFRcmd  
OR
- Write target flow to iData.DigitalPump.QPFRcmd
- Set iRequest.StartQcmd =1
- The Servostar injects the preset target pressure / flow.
- The target pressure / flow can be changed at any time.
- Set iRequest.StartQcmd =0 and the Servostar outputs the target pressure / flow 0.

Actual pressure, actual flow, actual current and actual position are cyclically transmitted in the RCV.PZD channel.

With this operation modes the most requirements can solved in a machine.

Further operation modes are not implemented in the Axis\_01\_FB (FB14), but can very easily additional programmed.

### Macro program

A macro program can execute in the Servostar further functions.

For macro programming the Software MacroStar is used.

In the parameter channel could be used for the data exchange between S7 PLC and the Servostar the ASCII parameter DPRVARxx and DPVxx for Macro und PLC programs in the parameter channel.

e.g. DPRVAR1 - PNU 2022 (dec) IND = 1 (dec)

In the real time channel exist additional iRequest.MacroInput

– see Axis\_01\_FB (FB14) – Network 88: #Axis.Send.PZD.STW.MacroInput

For special requirements it could be necessary to change and expand e.g. the Axis\_01\_FB (FB14) program code.

The complete SEND and RECEIVE data could be used also by the macro program.

Thereby the MacroStar compiling operated error-free it is necessary that the file variables.cfg contain:

```
PROSTW,2,""  
PROZSW,2,""  
PROFIN0,2,""  
PROFIN1,2,""  
PROFIN2,2,""  
PROFIN3,2,""  
PROFIN4,2,""  
PROFIN5,2,""  
PROFIN6,2,""  
PROFIN7,2,""  
PROFIN8,2,""  
PROFIN9,2,""  
PROFOUT0,2,""  
PROFOUT1,2,""  
PROFOUT2,2,""  
PROFOUT3,2,""  
PROFOUT4,2,""  
PROFOUT5,2,""  
PROFOUT6,2,""  
PROFOUT7,2,""  
PROFOUT8,2,""  
PROFOUT9,2,""
```

Example for S7-PLC und Macro programming in S300/S700:

The motor turns endless and stop followed directly into a defined transfer position without jerk with a smooth  $\sin^2$  brake ramp. The Target position and VJOG velocity will either transmit from the S7-PLC as DirectMotionTask parameter or could be stored as DPRVAR1 and DPRVAR2 at the S300/S700-EEPROM.

The complete motion is to be checked at the machine with the DriveGui-Oscilloscope.  
( Position, Position error, Actual current, Actual velocity )

S300/S700-Parameter:

```
OPMODE 8 ( oOpModeP = 2 dez )
PGEARI 3600
PGEARO 1
POSCNFG 1 (Axis type MODULO )
DREF 16
SRND 0
ERND 36000
DPRVAR1 18500
DPRVAR2 3000
DPRVAR3 73728
DPRVAR4 50
DPRVAR5 50
```

S7-PLC Program:

```
// Start iAck
// Start iInit
// Start iSetOpModeP =2
// Start iOpEnable
// Start Reference run
// Start DirektMotionTask to TargetPosition with VJOG-Velocity
```

Activate the Macro programm with Tag\_20 ( M200.0 )





## Makro program:

```

PROGRAM PLCINIT

LONG INP1:=0;
LONG INP2:=0;
LONG INP3:=0;
LONG VALUE1:= 1048576;
LONG VALUE2:= 1;
LONG VALUE3:= 35999;
LONG RESULT:= 1;
LONG TEMPVAR1:= 0;
LONG TEMPVAR2:= 0;

END_PROGRAM

//*****
PROGRAM PLCMAIN

END_PROGRAM

//*****

PROGRAM PLC250

// Profi-STW - Bit 9 - MacroInput
// Bit 9 0->1: JOGMOVE with VJOG-Velocity: iData.MotionTask.Direct.Speed or DPRVAR2
// Bit 9 1->0: STOP at TargetPosition: iData.MotionTask.Direct.Position or DPRVAR1 < ERND !!!
// O_ACC with DPRVAR4
// O_DEC with DPRVAR5

// O_P:= DPRVAR1; // TargetPosition
// O_V:= DPRVAR2; // VJOG-Velocity
O_C:= O_C | 0x12000; // DPRVAR3; // 73728dez <-> 0x12000hex: Sin^2 and SI-Units
O_ACC:= DPRVAR4;
O_DEC:= DPRVAR5;

IF O_V <> 0 THEN
    TEMPVAR1:= 7179; // GO
ELSE
    TEMPVAR1:= 8888; // XX
END_IF;

IF TEMPVAR1 = 7179 THEN
    TEMPVAR2:= PROSTW&0x200;
    IF TEMPVAR2 = 0x200 THEN
        INP2:= 0;
        INP3:= 0;
        IF INP1 = 0 THEN
            INP1:= 1;
            VJOGIO:=O_V; // Set VJOG speed
            SETPTR(TRJ,G_MOVEJOGIO); //Start moving
        END_IF;
    ELSE
        IF INP1 = 1 THEN
            INP2:= 1;
            VALUE2:= O_P;
            MULDIV(VALUE1,VALUE2,VALUE3,RESULT);
        END_IF;
        INP1:= 0;
        IF INP2 = 1 THEN
            IF INP3 = 0 THEN
                IF PFB < RESULT THEN
                    INP3:= 1;
                END_IF;
            END_IF;
            IF INP3 = 1 THEN
                IF PFB >= RESULT THEN
                    INP2:= 0;
                    INP3:= 0;
                    MOVEP_NR:=0;
                    SETPTR(TRJ,G_STARTMOVE);
                END_IF;
            END_IF;
        END_IF;
    END_IF;
END_IF;

END_PROGRAM

```

## page 28 of 49

The S7-PLC could parametrize the Servostar and store this data with SAVE and COLDSTART in the EEPROM.

At some parameters is after a change also a SAVE and afterwards COLDSTART necessary.

Please do this modification ever in the safe state from the Servoaxis.

Disable first the Servostar:

Set iSwEnable =0

Check that oState.P1\_SwitchOnInhibited =1

Set the Hardware-Enable Input =0

Set the hardware input to AS/STO-Enable =0

Start then a initialisation ( Zero-telegram )

Set ilnit =1 ( a pulse is generated internally )

Wait until oState.InitOk =1

Change now the parameters: execute with WRITE

Start now the SAVE – command – execute with READ

See ASCII parameter SAVE – PNU 1835

Note: The maximum permissible SAVE write cycles in an EEPROM is limited !!!

Wait 5 seconds – The parameters will now stored in the Servostar-EEPROM.

COLDSTART – command – execute with READ.

The Servostar boot now and make a Newstart / Restart.

The S7-PLC reports meanwhile a „Slave-Servostar – communication error“

Wait until the Servostar ist running again in the Profibus: oStateCommunicationOK =1

Reset now the warnings and errors in the Servostar and in the S7-PLC program in the blocks

iAck =1 ( Acknowledge )

Now execute READ and COMPARE to check that the Servostar stored correct the values in the EEPROM.

In the Servostar must now execute set operating mode and execute enable operation.

### Additional Profibus functions

The DP slave Servostar is activated or deactivated in Axis\_01\_FB (FB14) with D\_ACT\_DP with running S7-PLC.

- iActSlaveDevice =1, Servostar is activate and connect with data exchange in the Profibus.
  - iActSlaveDevice =0, Servostar is deactivated and data exchange is terminated. The watchdog timeout starts in the Servostar (EXTWD) and generate warning „n04 – node guarding ( watch dog )“.
- The Servostar can now unplug from the Profibus without an error message at the S7-PLC.

The Profibus communication is monitored with DPRD\_DAT and DPWR\_DAT for

- configuration errors
- Slave failure
- telegram errors

The entire SEND and RCV data telegrams are output in oAxis [STRUCT]

The Servostar state machine is output in oState [STRUCT]

With relative positioning, the data can be stored in oData.Canceled [STRUCT] if a MotionTask is canceled. At a later time (e.g. after EMERGENCY OFF – manual access - door OPEN / door CLOSED), the S7-PLC can use it to correct the MotionTask data and finish the relative MotionTask with corrected data.

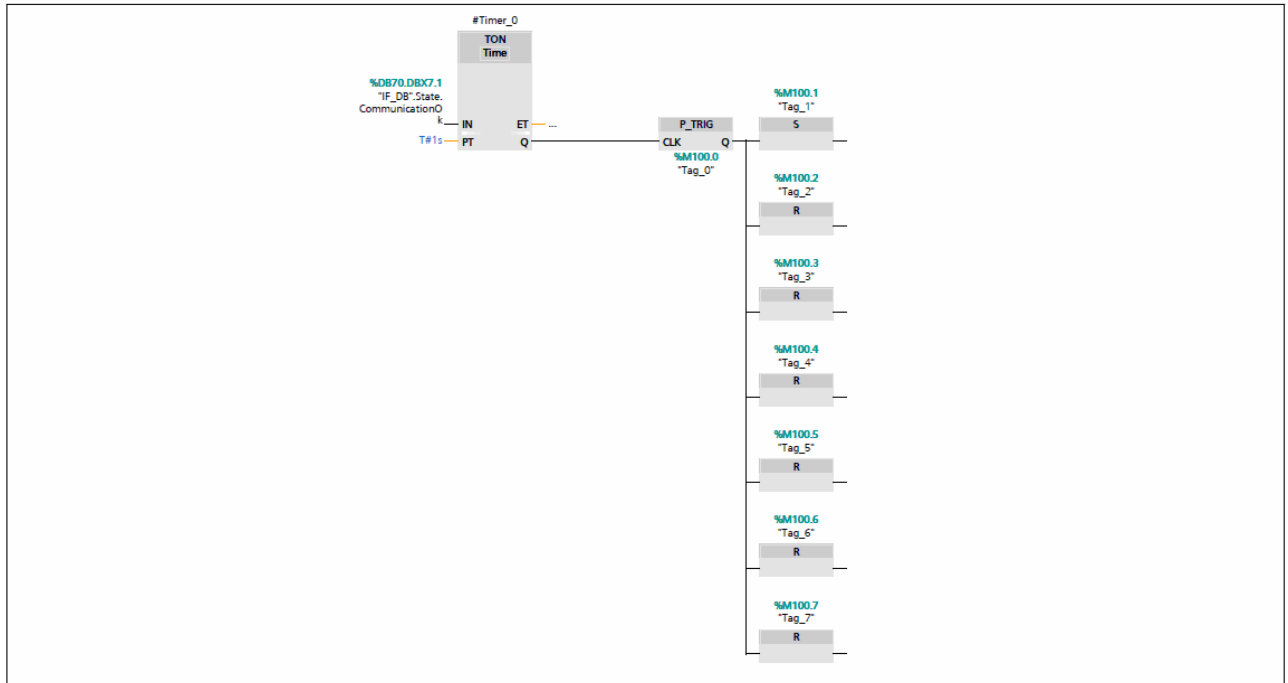
### Watch table 1

With the Watch table 1, the blocks can be controlled and monitored.

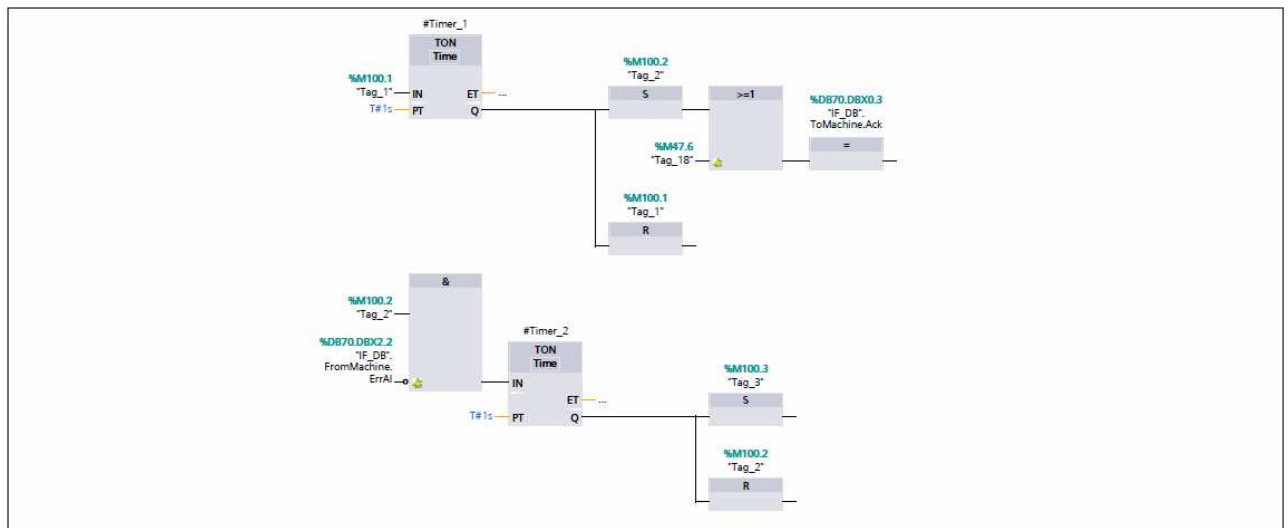
Sv14_v3c_V14_SP1_1516 - PLC 1 [CPU 1516-3 PN/DP] - Watch and force tables - Watch table_1									
	Name	Address	Display format	Monitor value	Modify value		C...	Tag comment	
1	*IF_DB*.ToMachine.Ack	%DB70.DBX0.3	Bool	<input type="checkbox"/> FALSE	FALSE			Acknowledge error	
2	*IF_DB*.Init	%DB70.DBX28.6	Bool	<input type="checkbox"/> FALSE	FALSE				
3	*IF_DB*.SetOpModeP	%DB70.DBX28.7	Bool	<input type="checkbox"/> FALSE	FALSE				
4	*IF_DB*.SetOperationEnable	%DB70.DBX29.0	Bool	<input type="checkbox"/> FALSE	FALSE				
5	*IF_DB*.Request.StartRef	%DB70.DBX4.0	Bool	<input type="checkbox"/> FALSE	FALSE			Start reference movement, static 1	
6	*IF_DB*.Request.StartIcmd	%DB70.DBX4.1	Bool	<input type="checkbox"/> FALSE	FALSE			Start I command digital torque, P4, static 1	
7	*IF_DB*.Request.StartMotionTask	%DB70.DBX4.2	Bool	<input type="checkbox"/> FALSE	FALSE			Start motion task ( direct motion task =0 )...	
8	*IF_DB*.Request.StopMotionTask	%DB70.DBX4.3	Bool	<input type="checkbox"/> FALSE	FALSE			Stop motion task, P4->P3, pulse 0->1	
9	*IF_DB*.Request.CancelMotionTask	%DB70.DBX4.4	Bool	<input type="checkbox"/> FALSE	FALSE			Cancel motion task, P3, pulse 0->1	
10	*IF_DB*.Request.StartNcmd	%DB70.DBX4.5	Bool	<input type="checkbox"/> FALSE	FALSE			Start N command digital speed, P4, static 1	
11	*IF_DB*.Request.JogPlus	%DB70.DBX4.6	Bool	<input checked="" type="checkbox"/> TRUE	TRUE			Jog positive, static 1	
12	*IF_DB*.Request.JogMinus	%DB70.DBX4.7	Bool	<input type="checkbox"/> FALSE	FALSE			Jog negative, static 1	
13	*IF_DB*.Request.MacroInput	%DB70.DBX5.0	Bool	<input type="checkbox"/> FALSE	FALSE			MacroProgramInput, static 1, PROSTW & 0...	
14	*IF_DB*.Request.StartQcmd	%DB70.DBX5.1	Bool	<input type="checkbox"/> FALSE	FALSE			Start Q command digital speed, P4, static 1	
15	*IF_DB*.PauseMotionTask	%DB70.DBX28.0	Bool	<input type="checkbox"/> FALSE	FALSE				
16	*IF_DB*.FastStopDisableAxis	%DB70.DBX28.2	Bool	<input type="checkbox"/> FALSE	FALSE				
17	*IF_DB*.FastStop	%DB70.DBX28.3	Bool	<input type="checkbox"/> FALSE	FALSE				
18	*IF_DB*.SwDisable	%DB70.DBX29.2	Bool	<input type="checkbox"/> FALSE	FALSE				
19									
20	*IF_DB*.iData.Config.OpModeP	%DB70.DBW64	DEC+/-	2	2			PNU 930 (2 =MotionTask, 1 =DigitalSpeed...	
21	*IF_DB*.iData.Config.JogSpeed	%DB70.DBW68	DEC	100	100			PNU 1889	
22	*IF_DB*.iData.Config.RefSpeed	%DB70.DBW70	DEC	100	100			PNU 1896	
23	*IF_DB*.iData.MotionTask.Number	%DB70.DBW72	DEC	0	0			0 =DirectMotionTask	
24	*IF_DB*.iData.MotionTask.DirectSpeed	%DB70.DBW74	DEC	100	100			PNU 1791	
25	*IF_DB*.iData.MotionTask.DirectPosition	%DB70.DBW78	DEC+/-	12345	12345			PNU 1790	
26	*IF_DB*.iData.MotionTask.DirectType	%DB70.DBW82	Bin	2#0010_000...	2#0010_00...			PNU 1785	
27	*IF_DB*.iData.DigitalSpeed.Ncmd	%DB70.DBW84	DEC+/-	547	547			PNU1886 - Ncommand ( Ncmd16 = Ncm...	
28	*IF_DB*.iData.DigitalTorque.Icmd	%DB70.DBW86	DEC+/-	200	200			PNU1870 - Icommand ( I[A] = Icmd * Ipea...	
29	*IF_DB*.iData.DigitalPump.QPRcmd	%DB70.DBW88	DEC+/-	43	43			1 <-> 10 mBar	
30	*IF_DB*.iData.DigitalPump.QFcmd	%DB70.DBW90	DEC+/-	12	12			1 <-> 0,1 l/min	
31									
32	*IF_DB*.FromMachine.ErrAl	%DB70.DBX2.2	Bool	<input type="checkbox"/> FALSE				Error or alarm at unit active	
33	*IF_DB*.FromMachine.Warning	%DB70.DBX2.3	Bool	<input type="checkbox"/> FALSE				Warning at unit active	
34									
35	*IF_DB*.OpModeP	%DB70.DBW26	DEC+/-	2					
36	*IF_DB*.ActualPosition	%DB70.DBW14	DEC+/-	412908					
37	*IF_DB*.ActualSpeed	%DB70.DBW18	DEC+/-	602					
38	*IF_DB*.ActualCurrent	%DB70.DBW20	DEC+/-	0					
39	*IF_DB*.ActualPressure	%DB70.DBW22	DEC+/-	0					
40	*IF_DB*.ActualFlow	%DB70.DBW24	DEC+/-	0					

# Example for a sequence „Initialisation and Enable Servostar“

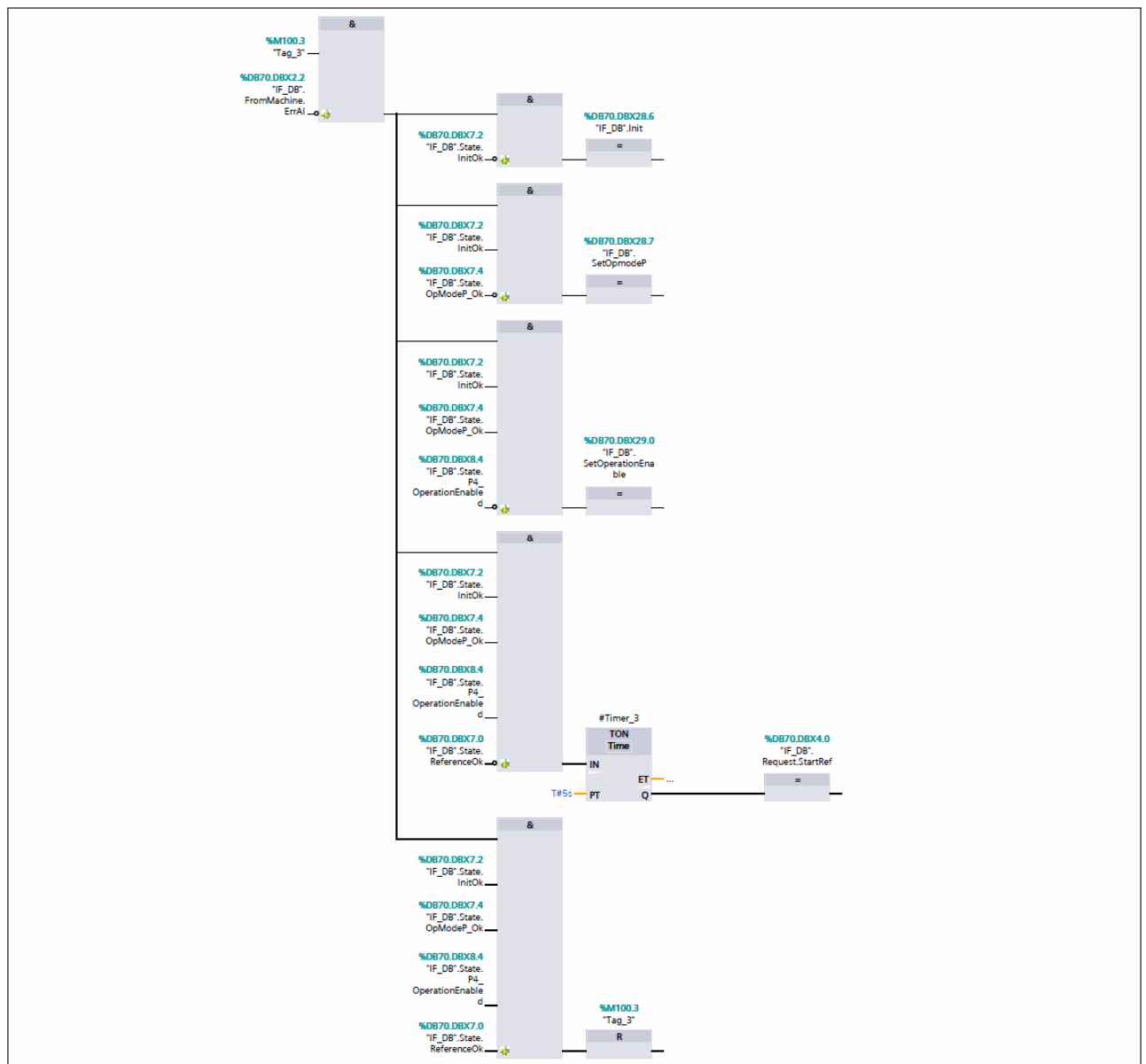
## Network 1: Initialisation Sequence Enable and Referencing



## Network 2: "IF\_DB".ToMachine.Ack



### Network 3: Sequence Enable and Referencing



Timer\_3: Delay is required for e.g. until Wake&Shake procedure is finished

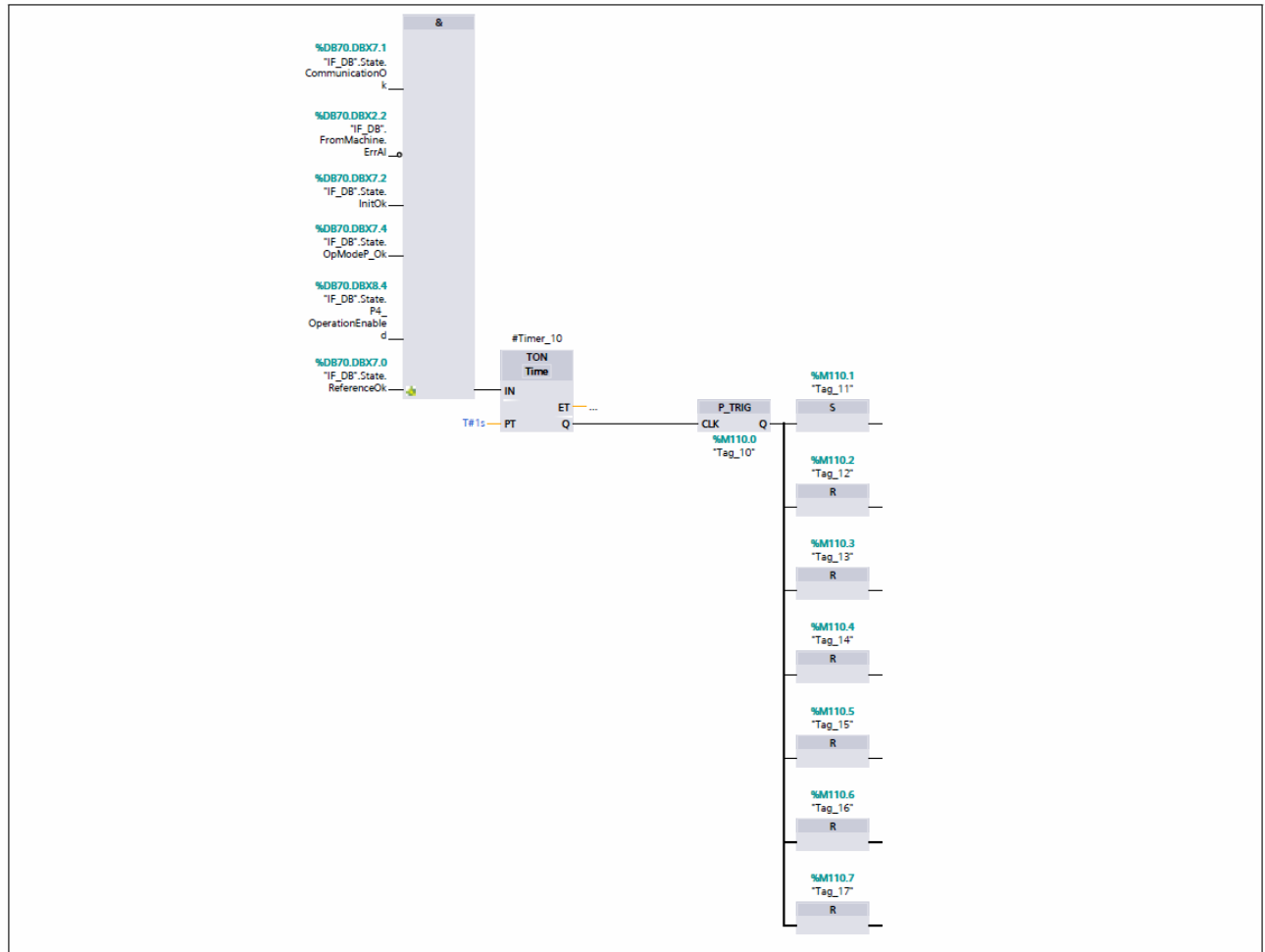
See "Switch ON and Start MotionTask timing diagram"



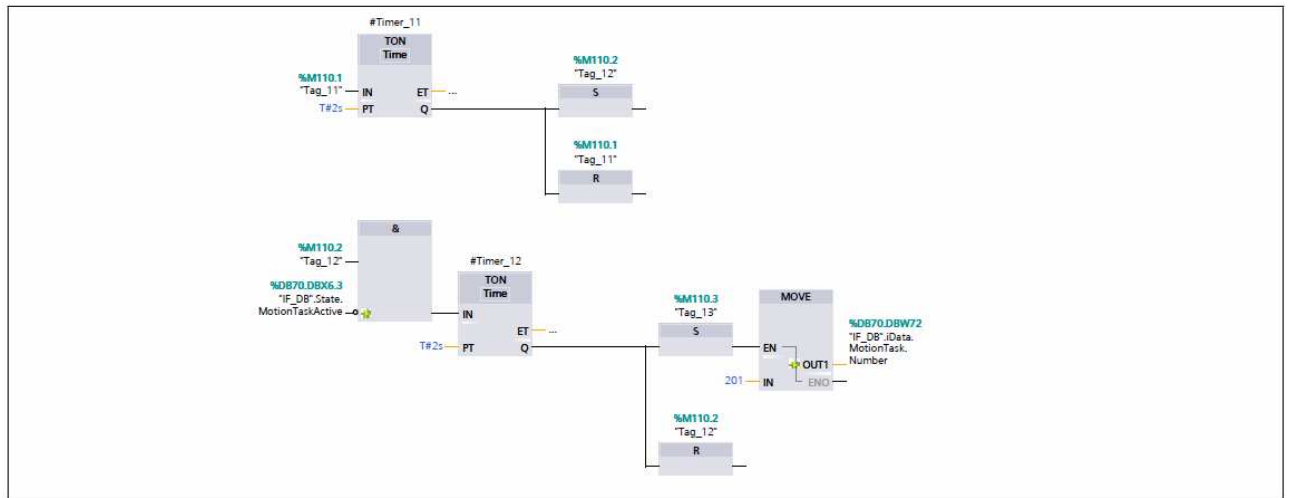
### Example for a sequence „Start MotionTasks in the Servostar“

```
// IF Ready then Write Data from PLC to Servostar with pulse from M110.1 and FB16
// and then Start RAM_MotionTask 201,210 and 211
// remark: the RAM_MotionTask 201 has a following RAM_MotionTask 202
```

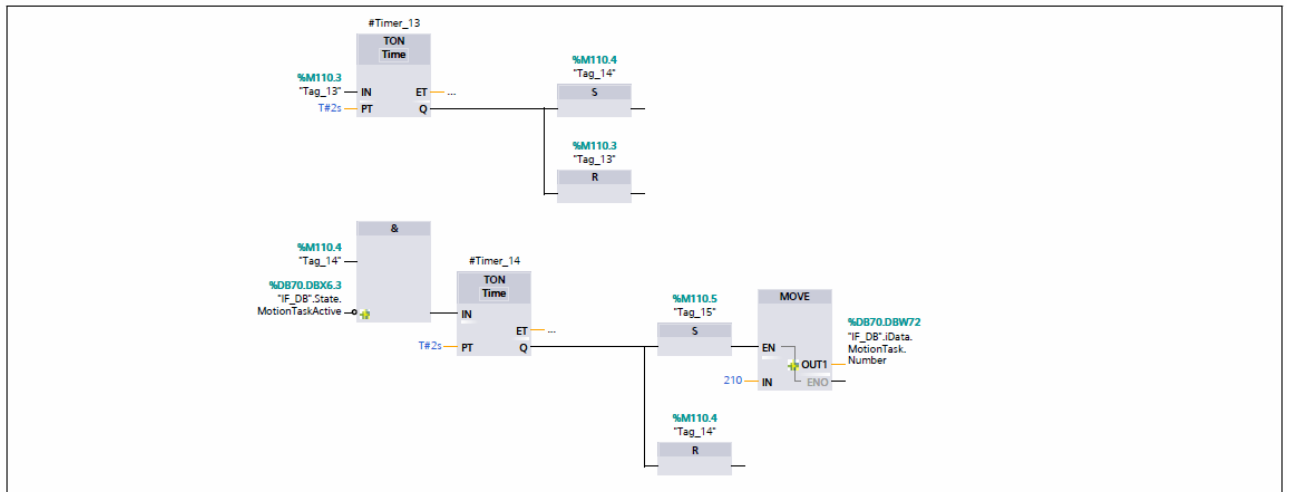
#### Network 4: Initialization Sequence Start Motion Tasks

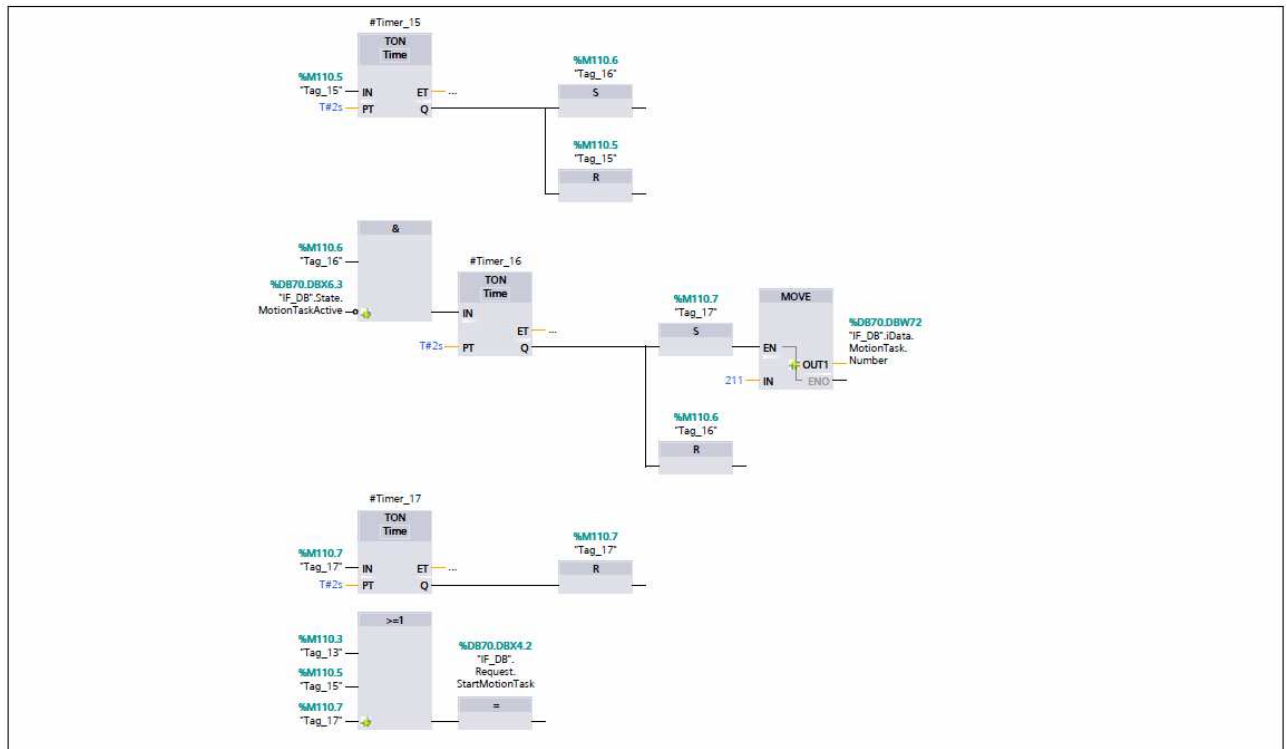


Network 5: "IF\_DB".Request.StartMotionTask No 201



Network 6: "IF\_DB".Request.StartMotionTask No 210

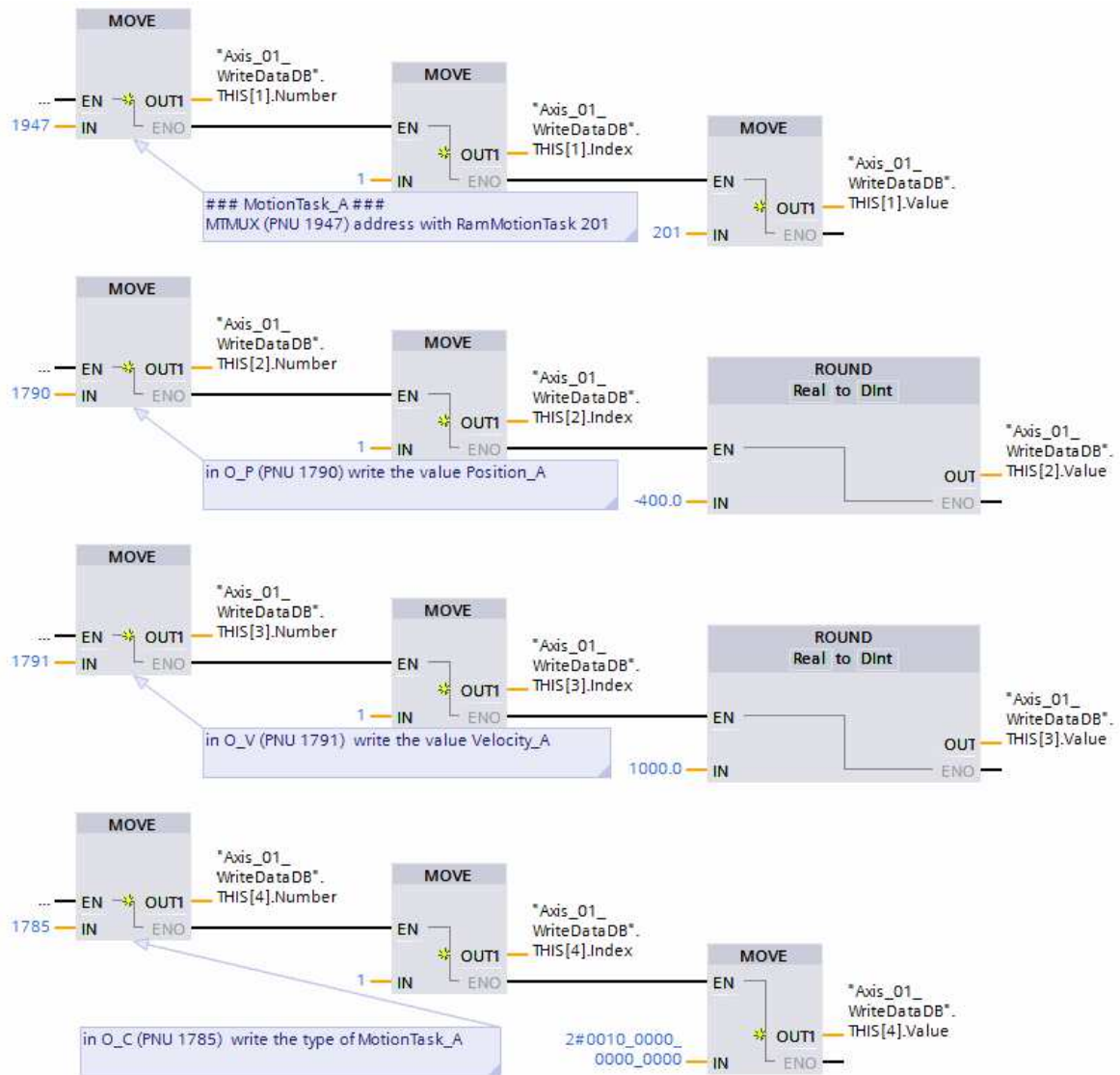


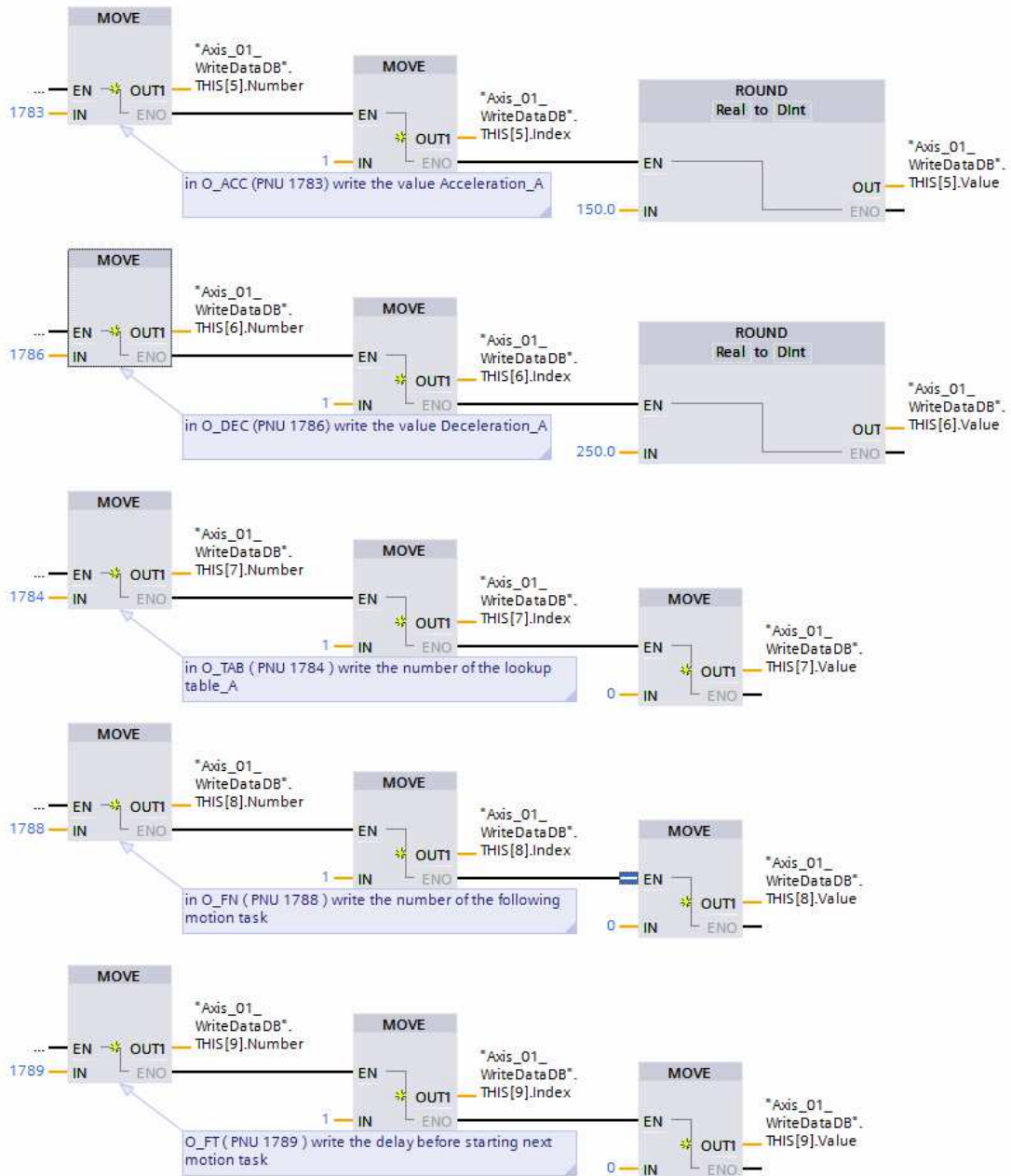
**Network 7: "IF\_DB".Request.StartMotionTask No 211**


See "Switch ON and Start MotionTask timing diagram"

### Example for the “RAM MotionTask 201 in the Servostar“

```
// ORDER 201 - RAM MotionTask
// PNU 1947 Index 1 - MTMUX
// PNU 1790 Index 1 - O_P target position/path for the motion task
// PNU 1791 Index 1 - O_V target speed/velocity
// PNU 1785 Index 1 - O_C type of motion task (control word)
// PNU 1783 Index 1 - O_ACC acceleration ramp /starting acceleration
// PNU 1786 Index 1 - O_DEC braking ramp / deceleration
// PNU 1784 Index 1 - O_TAB number of the lookup table
// PNU 1788 Index 1 - O_FN number of following motion tasks
// PNU 1789 Index 1 - O_FT delay before starting next motion task
```





Note for the ASCII parameter MTMUX - PNU 1947 – sample at GenCycleOB (OB1):

With DriveGui maybe there are no values visible in the MotionTask table No. 201,210,211  
Then make simply a recheck from the values with the DriveGui terminal:

--> ORDER 201

201 -400 1000 -1 -1 -1 -1 -1 400 -1

--> ORDER 210

210 -200 1500 -1 100 50 -1 -1 -1 -1

--> ORDER 211

211 -500 850 -1 150 250 -1 -1 -1 -1

-->

## Blocks for Servostar parameters

The S7 project „Sv14\_v3c\_V14\_SP1\_1516“ contains even more function blocks and data blocks that write complete data areas from the S7-PLC to the Servostar or read them from the Servostar in the S7-PLC und compare in the S7-PLC.

A record consists of 3 parameters:

- PNU – parameter number – indicates the respective parameter  
z.Bsp PNU 1783 – acceleration time O\_ACC1
- Index indicates what the transmitted value is about  
e.g Index=1 - actual value or Index=3 upper limit.
- Value – contains the transmitted value.

### Axis\_01\_WriteDataDB (DB20)

contains the data for writing with Axis\_01\_Write (FB16) and the Axis\_01\_FB (FB14) from the S7-PLC over the non real time PKW parameter channel in Profibus to the Servostar.

#### Declaration of Axis\_01\_WriteDataDB (DB20) – for 100 values

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Axis_01_WriteDataDB	Array[0..100] of *M_Axis_01_PnuStruct*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### Axis\_01\_ReadDataDB (DB21)

contains the data for reading with Axis\_01\_Read (FB17) and the Axis\_01\_FB (FB14) from the Servostar over the non real time PKW parameter channel in Profibus to the S7-PLC.

#### Declaration of Axis\_01\_ReadDataDB (DB21) – for 100 values

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Axis_01_ReadDataDB	Array[0..100] of *M_Axis_01_PnuStruct*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

See GenCycleOB (OB1) and Watch table\_1

### Axis\_01\_Write (FB16)

controls the Axis\_01\_FB (FB14) to write data from S7-PLC to the Servostar

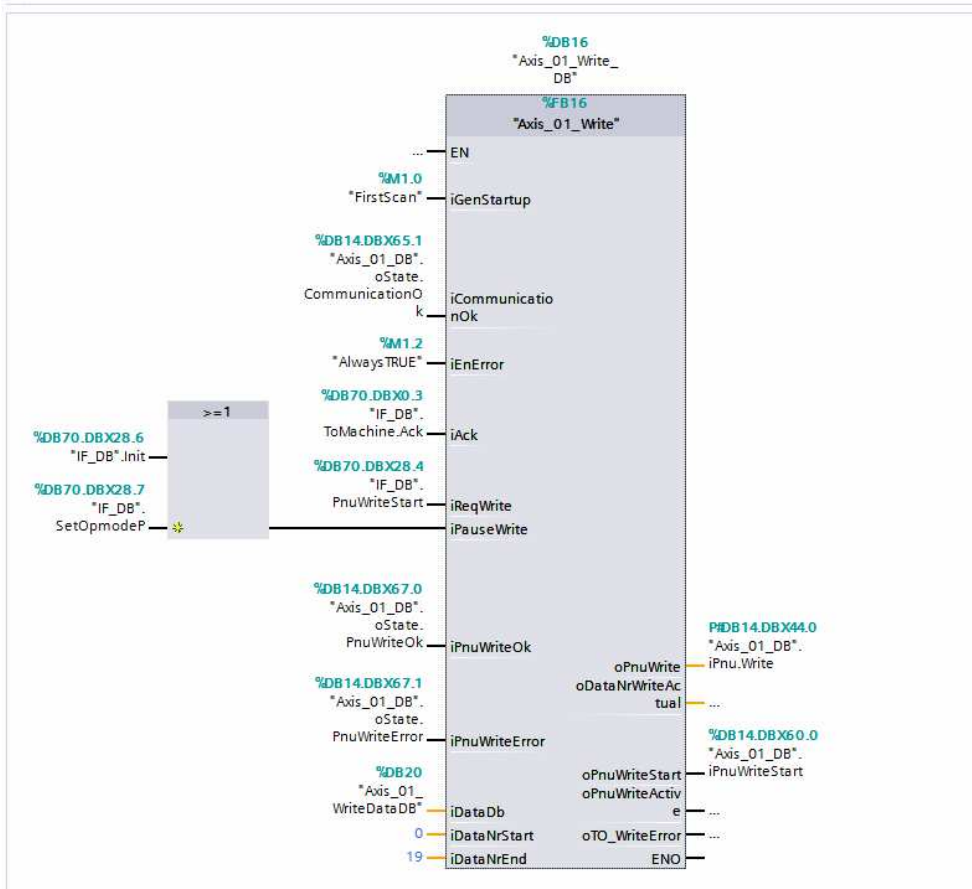
#### Input and output interface of Axis\_01\_Write (FB16)

VAR_INPUT		
iGenStartup	BOOL	Startup cycle after PLC startup
iCommunicationOk	BOOL	Axis communication Profibus is OK
iEnError	BOOL	Enable error messages
iAck	BOOL	Acknowledge WarningsErrors
iReqWrite	BOOL	RequestWrite
iPauseWrite	BOOL	PauseWrite (necessary for SetOpmode or InitAxis)
iPnuWriteOk	BOOL	PnuWriteOk =1 succesful
iPnuWriteError	BOOL	PnuWriteError =1 not succesful
iDataDBNr	WORD	WriteDataDBNumber
iDataNrStart	INT	DataSetNumberStart - first number from data to write
iDataNrEnd	INT	DataSetNumberEnd - last number from data to write

VAR_OUTPUT		
oPnuWrite	STRUCT	PnuWrite to Axis_FB actual with Number, Index, Value
Number	WORD	
Index	WORD	
Value	DWORD	
oDataNrWriteActual	INT	Data number is writing actual
oPnuWriteStart	BOOL	Request write start for Axis_FB
oPnuWriteActive	BOOL	Writing DataBlock is active
oTO_WriteError	BOOL	Timeout writing is active but not working

#### Network 3: Axis\_Write

Write: "Axis\_01\_WriteDataDB" from Data[0] until Data[19]





## Axis\_01\_Read (FB17)

controls the Axis\_01\_FB (FB14) to read data from the Servostar to the S7-PLC

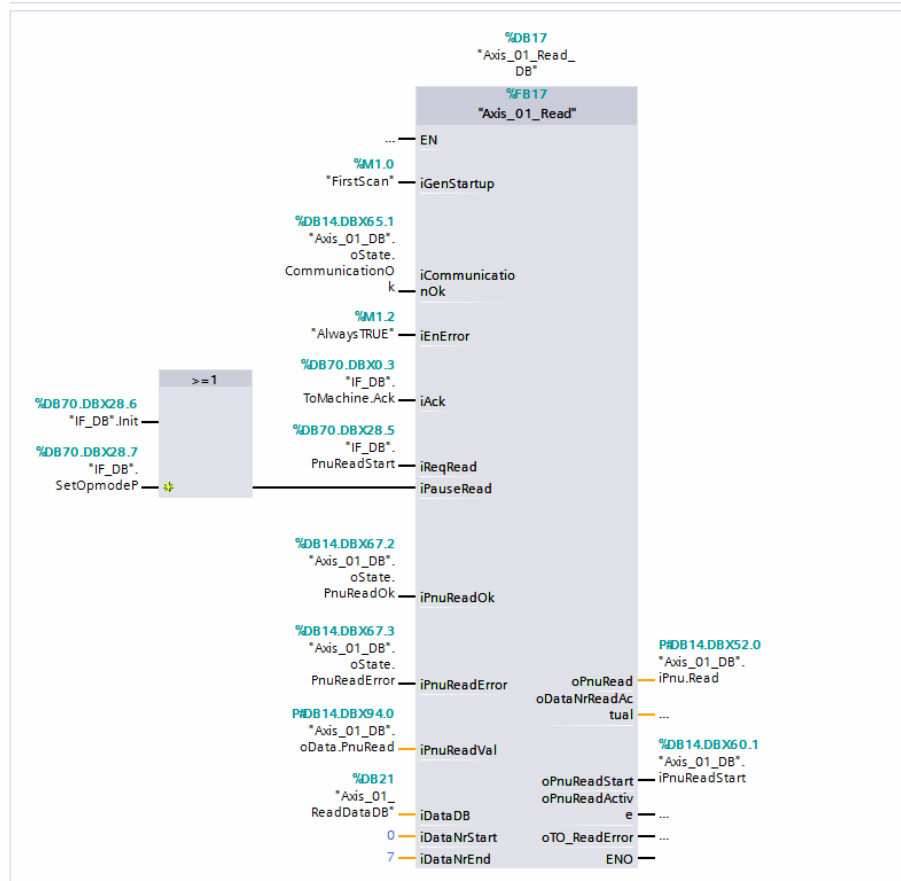
### Input and Output interface from the Axis\_01\_Read (FB17)

VAR_INPUT		
iGenStartup	BOOL	Startup cycle after PLC startup
iCommunicationOk	BOOL	Axis communication Profibus is OK
iEnError	BOOL	Enable error messages
iAck	BOOL	Acknowledge WarningsErrors
iRequestRead	BOOL	RequestRead
iPauseWrite	BOOL	PauseRead (necessary for SetOpmode or InitAxis)
iPnuWriteOk	BOOL	PnuReadOk =1 succesful
iPnuWriteError	BOOL	PnuReadError =1 not succesful
iPnuReadVal	STRUCT	PnuRead from Axis_FB actual with Number, Index, Value
Number	WORD	
Index	WORD	
Value	DWORD	
iDataDBNr	WORD	ReadDataDBNumber
iDataNrStart	INT	DataNumberStart - first number from data to read
iDataNrEnd	INT	DataNumberEnd - last number from data to read

VAR_OUTPUT		
oPnuRead	STRUCT	PnuRead from Axis_FB actual with Number, Index
Number	WORD	
Index	WORD	
oDataNrReadActual	INT	Data number is reading actual
oPnuReadStart	BOOL	Request read start for Axis_FB
oPnuReadActive	BOOL	Reading DataBlock is active
oTO_ReadError	BOOL	Timeout reading is active but not working

#### Network 5: Axis\_Read

Read: "Axis\_01\_ReadDataDB" from Data[0] until Data[7]



Note: Simultaneous Read and Write request at Axis\_01\_FB (FB14) generates the error oMaMsg.ErrReadWrite

### Axis\_01\_Compare (FB18)

Compares a data area from the written data from Axis\_01\_WriteDataDB (DB20) with the read data from Axis\_01\_ReadDataDB (DB21).

Only one PNU [STRUCT] is compared for each S7-PLC cycle.

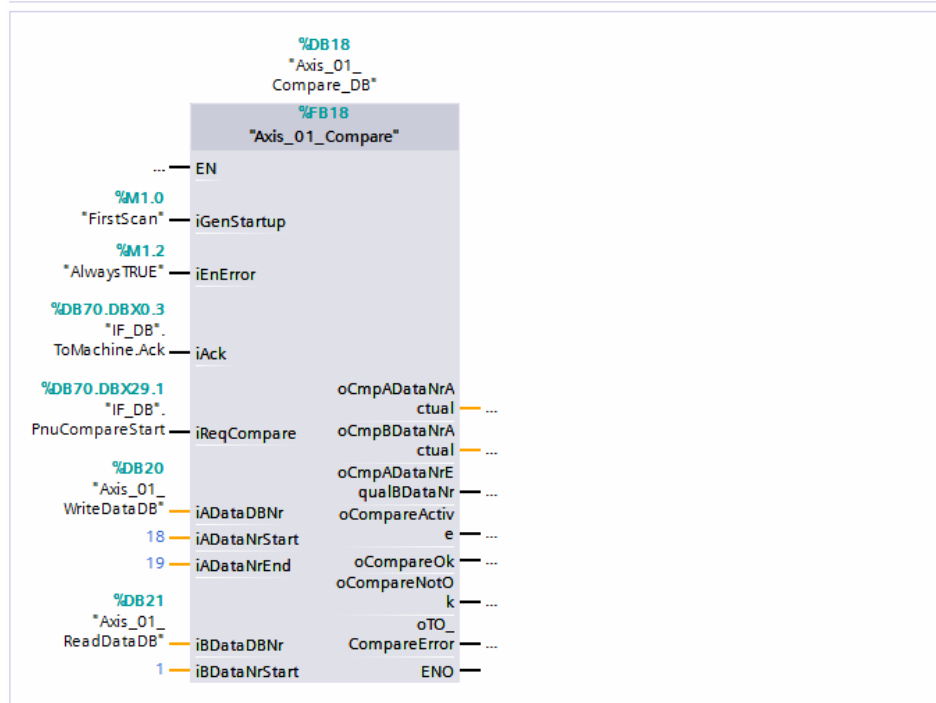
#### Input and output interface of Axis\_01\_CompareDB (FB18)

VAR_INPUT		
iGenStartup	BOOL	Startup cycle after PLC startup
iEnError	BOOL	Enable error messages
iAck	BOOL	Acknowledge WarningsErrors
iReqCompare	BOOL	Request compare
iADaDBNr	WORD	ADaDBNumber
iADaNrStart	INT	ADaNumberStart - first number from data to compare
iADaNrEnd	INT	ADaNumberEnd - last number from data to compare
iBdaDBNr	WORD	BdaDBNumber
iBdaNrStart	INT	BdaNumberStart - first number from data to compare

VAR_OUTPUT		
oCmpADaNrActual	INT	Compare A Data number is actual
oCmpBdaNrActual	INT	Compare B Data number is actual
oCmpADaNrEqualBdaNr	BOOL	Compare A Data number is equal B Data number
oCompareActive	BOOL	comparing DataBlocks is active
oCompareOk	BOOL	Compare is ok, datas are equal
oCompareNotOk	BOOL	Compare is not ok, datas are not equal
oTO_CompareError	BOOL	Timeout compare is active but not working

#### Network 6: Axis\_Compare

- ▼ iADaDBNr: "Axis\_01\_WriteDataDB" from Data[18] until Data[19]
- iBdaDBNr: "Axis\_01\_ReadDataDB" from Data[1] until Data[19-18+1=2]



With the Watch table\_1, the blocks Axis\_01\_FB (FB14), Axis\_01\_Read (FB16), Axis\_01\_Write (FB17) and Axis\_01\_Compare (FB18) can be controlled and monitored.

## Tips and information:

### Error messages and warnings from the blocks

must be implemented so that the S7-PLC program can respond to them.

oMaMsg	STRUCT	Error messages
ErrTO_Ref	BOOL	Error timeout reference
ErrTO_Pos	BOOL	Error timeout positioning
ErrNoReferenceSet	BOOL	Error if reference is not set and request
ErrWrongOpMode	BOOL	Error wrong operation mode selected and request
ErrActSlave	BOOL	Error activating slave
ErrCfgInput	BOOL	Error configuration input
ErrRcv	BOOL	Error receiving data
ErrCfgOutput	BOOL	Error configuration output
ErrSend	BOOL	Error sending data
ErrAxis	BOOL	Error from axis
ErrReadWrite	BOOL	Error request Read and Write together
ErrNotEnabled	BOOL	Error if not enabled and request
ErrRes_1_4	BOOL	
ErrRes_1_5	BOOL	
ErrRes_1_6	BOOL	
ErrCmd	BOOL	Error more than one request command active
oTO_WriteError	BOOL	Timeout writing is active but not working
oTO_ReadError	BOOL	Timeout reading is active but not working
oTO_CompareError	BOOL	Timeout reading is active but not working

### Override via Profibus

See ASCII parameter OVERRIDE

The override-function allows controlling the speed of a MotionTask, the reference speed and the jogging speed.

OVRIDE=0 Override function deactivated

OVRIDE=3 Profibus for the digital Override function activated

See ASCII parameter DOVRIDE:

When activating the digital Override function, this parameter is used to predefine the digital Override factor.

The following scaling applies:

DOVRIDE=0 MotionTask speed = 0 %

DOVRIDE=8192 MotionTask speed = 100 %

The Override function is not possible for sin^2 – ramps !

### Activating the sin<sup>2</sup> motion profile for DirectMotionTask No. 0

The motion profile Sin<sup>2</sup> enables a smooth, jerk-free accelerating and decelerating. This protects the mechanical system (gears, spindles). Pendulum movements etc. are suppressed.

For Servostar S400/S600, see ASCII list, it is not described here more in detail.

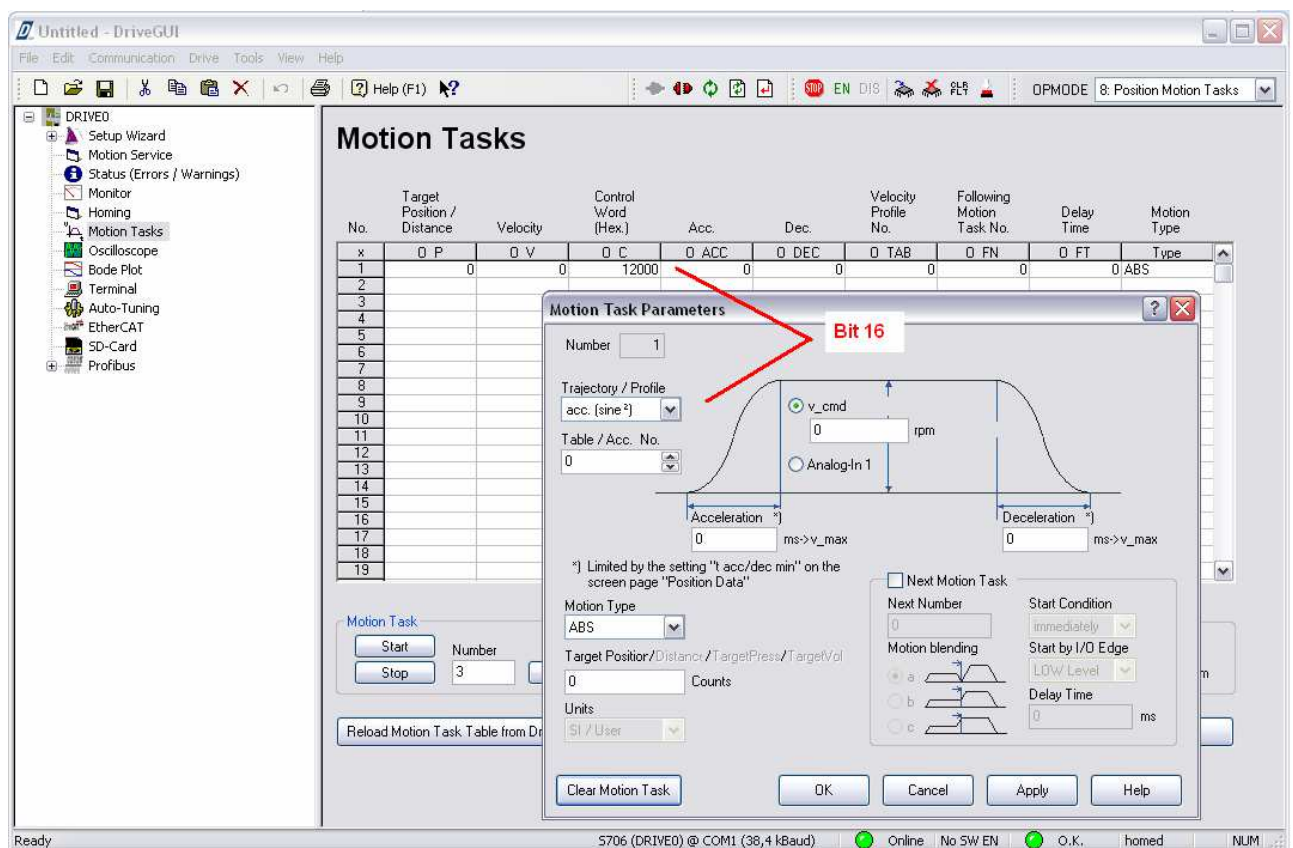
For Servostar S300/S700 – Profibus see ASCII parameter O\_C

The O\_C (Profibus PNU 1785) command defines the MotionTaskType for the direct MotionTask No.0. If bit 16 is set by O\_C, then the MotionTask has sine<sup>2</sup> profile. Bit 9 must be set to 0. From the O\_C are only the Bits 0 to 15 directly addressed in DirectMotionTaskType (PZD5), thereby the bit 16 must be set separately in S300/S700 by another way.

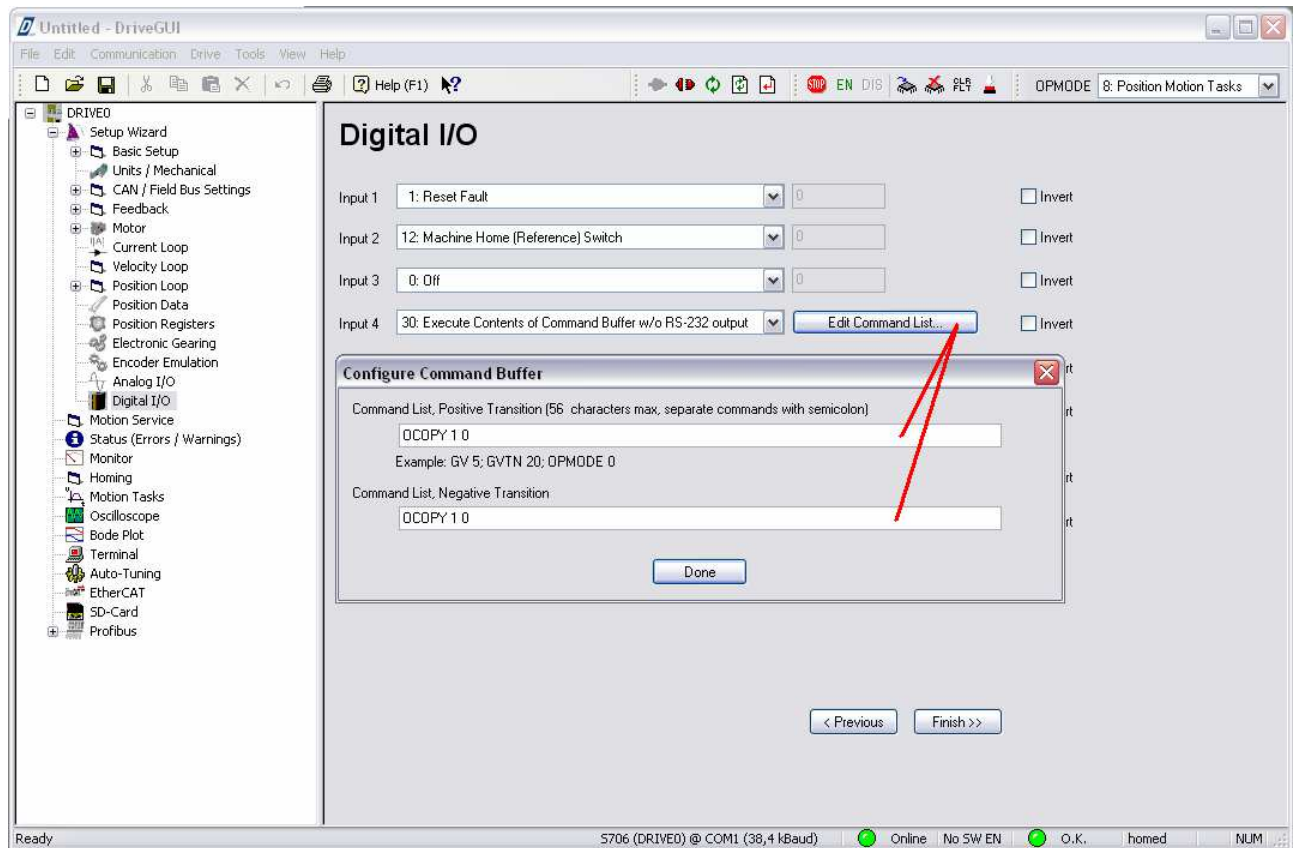
Further the O\_C command is a RAM parameter and thereby not storable in the Servostar.

O\_C - Bit 16 can be set in this ways:

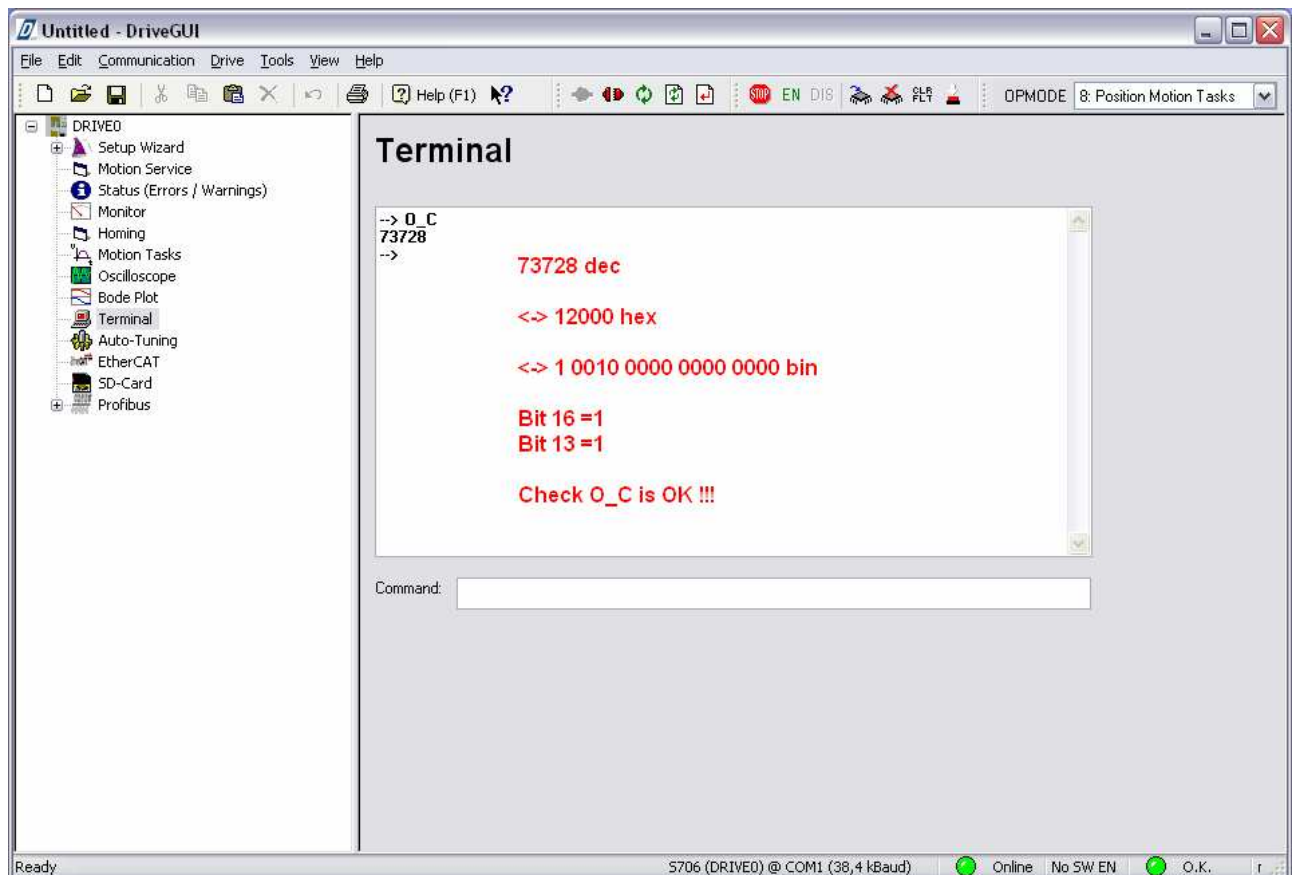
- 1.) Use the PKW parameter channel with PNU 1785 to write the value 10000hex to S300/S700.
- 2.) After switching ON the 24-VDC supply voltage, the S300/S700 automatically copies at EEPROM MotionTask with Bit 16 =1 ( with motion profile Sin<sup>2</sup> ), e.g. no.1, to the DirectMotionTask No. 0. This allows the S7 program, to remain unchanged and PNU 1785 parameter download is not necessary. In addition, the acceleration ramp O\_ACC and deceleration ramp O\_DEC can also be set for smooth motion. For value 0, the value PTMIN applies or the largest ramp in each case.



An unused digital Input (not wired) is assigned the OCOPY function from EEPROM MotionTask e.g. No.1 to direct MotionTask No. 0



SAVE and COLDSTART and then check the content of O\_C with DriveGui terminal



### 3.) ASCII command sequence

The command ASCII INxHCMD / INxLCMD is used to define an ASCII command sequence. This command sequence will always be carried out when a rising/falling edge is detected at the input x that has been configured with the function INxMODE=30.

A command sequence consists of individual ASCII commands, separated by a semicolon (;)

The maximum length of this command sequence is 56 characters.

With DriveGui-Terminal:

```
IN4MODE 30
IN4HCMD O_C 73728; O_ACC 100; O_DEC 200
IN4LCMD O_C 73728; O_ACC 100; O_DEC 200
SAVE
COLDSTART
```

Recheck with DriveGui-Terminal:

```
ORDER 0
0 0 0 73728 100 200 0 0 0 0
```

### 4. ) Macro program

Example:

```
O_C:= O_C | 0x12000; // DPRVAR3; // 73728dez <-> 0x12000hex: Sin^2 and SI-Units
```

Recheck with DriveGui-Terminal:

```
ORDER 0
0 0 0 73728 0 0 0 0 0 0
```

Note: The recheck with DriveGui terminal - there is also possible this result with the same functionality:

```
ORDER 0
0 -1 -1 73728 -1 -1 -1 -1 -1 -1
```

# Bitcoding of error – PNU 1001 - ERRCODE

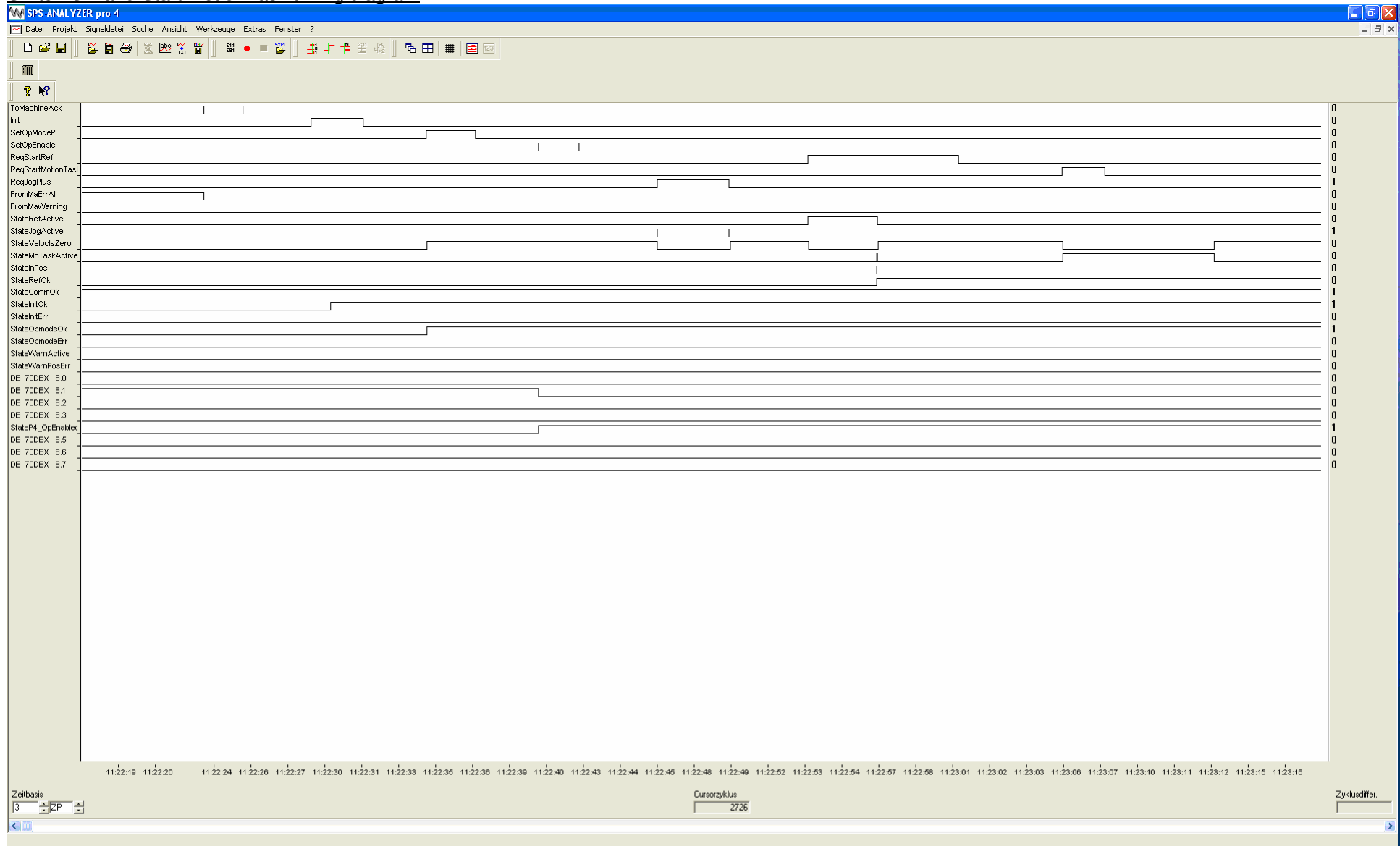
Axis_PNU1001	MD 1000	DWORD	Faults / Error
Axis_PNU1001_Bit31_F32	M 1000.7	BOOL	System error
Axis_PNU1001_Bit30_F31	M 1000.6	BOOL	Error SafetyCard
Axis_PNU1001_Bit29_F30	M 1000.5	BOOL	Error Emergency stop timeout
Axis_PNU1001_Bit28_F29	M 1000.4	BOOL	Slot card error
Axis_PNU1001_Bit27_F28	M 1000.3	BOOL	EtherCAT synchronization error
Axis_PNU1001_Bit26_F27	M 1000.2	BOOL	STO error
Axis_PNU1001_Bit25_F26	M 1000.1	BOOL	Error during reference travel – hardware limit switch
Axis_PNU1001_Bit24_F25	M 1000.0	BOOL	Commutating error
Axis_PNU1001_Bit23_F24	M 1001.7	BOOL	Warning error changed to error
Axis_PNU1001_Bit22_F23	M 1001.6	BOOL	Error in CAN communication
Axis_PNU1001_Bit21_F22	M 1001.5	BOOL	Error reserved
Axis_PNU1001_Bit20_F21	M 1001.4	BOOL	Error handling error
Axis_PNU1001_Bit19_F20	M 1001.3	BOOL	Error slot error
Axis_PNU1001_Bit18_F19	M 1001.2	BOOL	Error collapse in DC link voltage
Axis_PNU1001_Bit17_F18	M 1001.1	BOOL	Ballast error (defective ballast transistor)
Axis_PNU1001_Bit16_F17	M 1001.0	BOOL	Error A/D converter
Axis_PNU1001_Bit15_F16	M 1002.7	BOOL	Error network BTB
Axis_PNU1001_Bit14_F15	M 1002.6	BOOL	Error I2tmax exceeded
Axis_PNU1001_Bit13_F14	M 1002.5	BOOL	Error output stage: Ground fault, short circuit or ballast short circuit
Axis_PNU1001_Bit12_F13	M 1002.4	BOOL	Error ambient temperatur
Axis_PNU1001_Bit11_F12	M 1002.3	BOOL	Error reserved
Axis_PNU1001_Bit10_F11	M 1002.2	BOOL	Error brake
Axis_PNU1001_Bit9_F10	M 1002.1	BOOL	Cable break ROD interface or timing problem of master slave controller booting
Axis_PNU1001_Bit8_F09	M 1002.0	BOOL	Error EEPROM
Axis_PNU1001_Bit7_F08	M 1003.7	BOOL	Error overspeed
Axis_PNU1001_Bit6_F07	M 1003.6	BOOL	Error internal supply voltages
Axis_PNU1001_Bit5_F06	M 1003.5	BOOL	Error motor temperature
Axis_PNU1001_Bit4_F05	M 1003.4	BOOL	Error undervoltage
Axis_PNU1001_Bit3_F04	M 1003.3	BOOL	Feedback error
Axis_PNU1001_Bit2_F03	M 1003.2	BOOL	Contouring error when executing external trajectory
Axis_PNU1001_Bit1_F02	M 1003.1	BOOL	Error overvoltage
Axis_PNU1001_Bit0_F01	M 1003.0	BOOL	Error heat sink temperature



# Bit coding of warnings and manufacturer specific state register – PNU 1002 - DRVSTAT

Axis_PNU1002	MD 1004	DWORD	Warnings AND manufacturer-specific state register
Axis_PNU1002_Bit31	M 1004.7	BOOL	Error present
Axis_PNU1002_Bit30	M 1004.6	BOOL	Output stage enabled
Axis_PNU1002_Bit29	M 1004.5	BOOL	Safety relay tripped (STO)
Axis_PNU1002_Bit28	M 1004.4	BOOL	Velocity = 0
Axis_PNU1002_Bit27	M 1004.3	BOOL	-
Axis_PNU1002_Bit26	M 1004.2	BOOL	Initialization complete (internal initialization of amplifier finished)
Axis_PNU1002_Bit25	M 1004.1	BOOL	Position 4 reached (see above)
Axis_PNU1002_Bit24	M 1004.0	BOOL	Position 3 reached (see above)
Axis_PNU1002_Bit23	M 1005.7	BOOL	Position 2 reached (see above)
Axis_PNU1002_Bit22	M 1005.6	BOOL	Position 1 reached (see above)
Axis_PNU1002_Bit21	M 1005.5	BOOL	-
Axis_PNU1002_Bit20	M 1005.4	BOOL	Position latch made
Axis_PNU1002_Bit19	M 1005.3	BOOL	In Position
Axis_PNU1002_Bit18	M 1005.2	BOOL	Actual position = home position (reference switch is occupied)
Axis_PNU1002_Bit17	M 1005.1	BOOL	Reference point set (after a reference travel or absolute encoder)
Axis_PNU1002_Bit16	M 1005.0	BOOL	MotionTask active - motion set, jogging mode, reference move
Axis_PNU1002_Bit15_n16	M 1006.7	BOOL	Warning 16: Reserve
Axis_PNU1002_Bit14_n15	M 1006.6	BOOL	Warning 15: Speed current table INXMODE 35 error
Axis_PNU1002_Bit13_n14	M 1006.5	BOOL	Warning 14: SinCos commutation not completed
Axis_PNU1002_Bit12_n13	M 1006.4	BOOL	Warning 13: Expansion card does not work properly
Axis_PNU1002_Bit11_n12	M 1006.3	BOOL	Warning 12: HIPERFACE® or EnDat®: Motor default values have been loaded
Axis_PNU1002_Bit10_n11	M 1006.2	BOOL	Warning 11: Limit switch NSTOP activated
Axis_PNU1002_Bit9_n10	M 1006.1	BOOL	Warning 10: Limit switch PSTOP activated
Axis_PNU1002_Bit8_n09	M 1006.0	BOOL	Warning 9: No reference point was set for Start motion task
Axis_PNU1002_Bit7_n08	M 1007.7	BOOL	Warning 8: A defective motion task was started
Axis_PNU1002_Bit6_n07	M 1007.6	BOOL	Warning 7: Software limit switch 2 exceeded
Axis_PNU1002_Bit5_n06	M 1007.5	BOOL	Warning 6: Software limit switch 1 exceeded
Axis_PNU1002_Bit4_n05	M 1007.4	BOOL	Warning 5: Power supply phase missing
Axis_PNU1002_Bit3_n04	M 1007.3	BOOL	Warning 4: Watchdog (fieldbus) active
Axis_PNU1002_Bit2_n03	M 1007.2	BOOL	Warning 3: Set contouring error exceeded
Axis_PNU1002_Bit1_n02	M 1007.1	BOOL	Warning 2: Set braking power reached
Axis_PNU1002_Bit0_n01	M 1007.0	BOOL	Warning 1: I²t reporting threshold exceeded

## Switch ON and Start MotionTask timing diagram



Kill MotionTask timing diagram

