



**SCHLEICHER**

Ein Unternehmen der Wieland Gruppe

# Mit Sicherheit in die Zukunft

## Operating Manual

### XCx

Compact Controllers for PLC and  
CNC Solutions

Operating Manual XCx Version 07/05  
Part No. R4.322.2140.0 (322 383 83)

## Operating Manual XCx

### Target Group

These operating manual have been written for trained personnel with specialised knowledge. There are special demands on the selection and training of the personnel who work on the automation system. Suitable personnel are, for example, electricians and electrical engineers who have had the relevant training (see also Safety-related information, "Selection and Qualification of Personnel").

### Applicability of these operating manual

Version Hardware XX / Software XX

### Previous versions of these operating manual

06/02 08/02 04/03 01/04 08/04 02/05 04/05 06/05

### Where can I obtain manuals?

You can download all our programming and operating manuals free of charge from our web site at <http://www.schleicher-electronic.com> or order them by writing to the following address (please quote order no.):

SCHLEICHER Electronic GmbH & Co. KG  
Pichelswerderstraße 3-5  
D-13597 Berlin  
Germany

Bezeichnung	Artikel-Nr.
MULTIPROG Programmiersystem nach IEC 61131-3 deutsch	322 158 43
Getting Started MULTIPROG deutsch	322 380 14
MULTIPROG Programming System acc. IEC 61131-3 englisch	322 158 44
Getting Started MULTIPROG englisch	322 380 15
Inbetriebnahmehinweise für Feldbussysteme	322 152 48
Commissioning Field Bus Systems	322 152 49
CNC-Programmierung XCx und ProNumeric deutsch	322 381 61
CNC Programming XCx and ProNumeric englisch	322 381 62
RIO Erweiterungsmodule deutsch	322 154 14
RIO Expansion Modules englisch	322 154 15

### Copyright by

SCHLEICHER Electronic GmbH & Co. KG  
Pichelswerderstraße 3-5  
D-13597 Berlin  
Germany  
Phone +49 (0) 30 33005 - 330  
Fax +49 (0) 30 33005 - 305  
Hotline +49 (0) 30 33005 - 304  
Internet <http://www.schleicher-electronic.com>

Errors and omissions excepted. Subject to modifications.

## Contents

1	Overview .....	8
2	Installation .....	12
2.1	XCx.....	12
2.1.1	Installation position.....	12
2.1.2	Installation clearances.....	12
2.1.3	Fixing the XCx to DIN rail.....	13
2.1.4	Fixing with screws .....	15
2.1.5	Connecting the voltage supply XCx .....	16
2.2	XRIO Installation .....	18
3	Displays, Connections and Controls.....	21
3.1	Overview .....	21
3.2	LED Displays.....	22
3.3	CompactFlash .....	24
3.4	X7 PRG RS232 Interface for Programming Devices.....	24
3.5	X6 ETH RJ 45 EtherNet Connection .....	24
3.6	X5 CANopen Field Bus Connection .....	25
3.7	X4 X RIO Connection .....	25
3.8	X3 USB Connection (Universal Serial Bus).....	25
3.9	X1 RS422 Interface for Connecting Operator Panels and Displays.....	26
3.10	X2 Connection for Supply Voltage, Direct I/O and BUSY Contact .....	26
3.11	Mode Switch.....	26
3.12	RESET Button.....	27
3.13	IRDa Infrared Interface.....	27
4	Quick Introduction .....	28
4.1	Starting up Communication .....	28
4.1.1	Communication connections .....	28
4.1.2	Starting up the serial connection via the RS232 interface.....	28
4.2	Starting up the Ethernet Interface .....	30
4.2.1	Output current IP address of XCx via the RS232 interface .....	30
4.2.2	Setting the IP address of the service PC.....	30
4.2.3	Setting up and checking the Ethernet connection .....	31
4.3	Changing the IP Address of the XCx.....	31
4.4	Installing the MULTIPROG, AddOns and OPC Server .....	33
4.4.1	Installing MULTIPROG.....	33
4.4.2	Starting MULTIPROG and opening a new project .....	37
4.4.3	Setting the Ethernet connection in MULTIPROG .....	38
4.4.4	Compiling a project and sending to the XCx .....	39
4.5	Accessing the I/O Level with XRIO .....	41
4.5.1	XRIO connection .....	41
4.5.2	Example of an XRIO configuration .....	41
4.5.3	Inserting the XRIO configuration .....	42
4.5.4	The first I/O access by the PLC program .....	45
4.5.5	Diagnosis.....	46
4.6	CANopen for Remote I/O .....	47
4.6.1	Specifications .....	47
4.6.2	Connection and wiring.....	48
4.6.3	Settings on the RIO 8 I/O CANopen module.....	49
4.6.4	Declaring the I/O driver for CANopen.....	49
4.6.5	Declaring network variable in MULTIPROG.....	51
4.6.6	Configuring the CANopen network with ProCANopen .....	52
4.6.7	Installing ProCANopen .....	52
4.6.8	Integrating ProCANopen in MULTIPROG .....	53
4.6.9	First connections with ProCANopen.....	54
4.7	The Web-Server Functions of the XCx.....	58
4.7.1	General functions and concept.....	58
4.7.2	Schleicher-specific applet.....	58
4.7.3	Declaring variables for visualization .....	59
4.7.4	Application example .....	59
4.7.5	Browser / components.....	59



5	XRIO.....	60
5.1	Recognition of XRIO Configuration by XCx Operating System.....	61
5.1.1	The sysinfo.txt file XRIO section .....	62
5.2	Addressing I/O Modules.....	64
5.3	XRIO Flags in Shared Memory Validity of Process Data .....	64
5.4	The XRIO Driver.....	65
5.5	Calculating Estimated Transmission Time .....	65
5.6	Expansion Modules from the RIO System .....	67
5.6.1	Overview .....	67
5.6.2	Tables of module types and module classes .....	70
5.7	Expansion Module XRIO .....	71
5.7.1	Displays and Connections.....	71
5.7.2	Technical data .....	71
6	CANopen Field Bus.....	72
6.1	Fundamentals.....	72
6.1.1	Process Data Objects (PDO) .....	73
6.1.2	Service Data Objects (SDO) .....	73
6.1.3	Nodeguarding.....	73
6.1.4	Lifeguarding.....	74
6.2	CANopen-Specific PLC Addresses .....	75
6.2.1	Parameterizing and diagnosis data .....	75
6.2.2	Network variables .....	75
6.2.3	Access to network variables and I/O configuration .....	76
6.2.4	Accessing direct network variables .....	78
6.2.5	Accessing parameterizing and diagnosis functions.....	78
6.3	SDO Function Blocks .....	79
6.4	Parameterizing and Diagnosis Functions.....	80
6.4.1	Function 1 Set and read CANopen node number .....	81
6.4.2	Function 2 Set and read data transmission rate.....	81
6.4.3	Function 7 Output CANopen error number .....	82
6.4.4	Function 8 Read CANopen firmware version .....	82
6.4.5	Function 20 Set CANopen startup delay .....	82
6.5	Error messages .....	83
6.6	Expansion Module XF-CAN .....	84
6.6.1	Displays and Connections.....	84
6.6.2	Technical data .....	84
7	The XCx Multi-Tasking System.....	85
7.1	Overview .....	85
7.2	User Tasks .....	86
7.2.1	Cyclical tasks.....	86
7.2.2	Event tasks.....	87
7.2.3	System tasks .....	88
7.2.4	Default task .....	90
7.3	User Task Information.....	91
7.4	Task Priorities.....	93
7.5	Tasks and Watchdogs.....	94
7.6	Inserting Tasks and Assigning Programs.....	95
8	The PLC.....	97
8.1	Programming.....	97
8.2	PLC Operating States and Starting Behaviour .....	97
8.2.1	Operating states .....	97
8.2.2	Changing operating states with MULTIPROG.....	98
8.2.3	PLC starting behaviour after power supply is switched on.....	99
8.3	System Variables .....	99
8.4	Libraries and Function Blocks in MULTIPROG.....	100
8.4.1	CANopen_Vxxx library .....	102
8.4.2	CFB_Vxxx Library .....	102
8.4.3	CNC_Vxxx library .....	103
8.4.4	Date_Time library .....	103
8.4.5	MMI Library .....	104
8.4.6	PLC library.....	104



8.4.7	Profibus_Vxxx Library .....	104
8.4.8	Serial Library .....	105
8.4.9	SchleicherLib_Vxxx library .....	105
8.4.10	XCx7_Vxxx Library .....	105
8.5	PLC Operating System ProConOS .....	106
8.5.1	ProConOS.INI initialization file .....	106
9	The CNC .....	107
9.1	Configuration of the shared RAM PLC-CNC .....	107
9.2	Important Notes on the Multi-Tasking System and CNC .....	108
10	SLM Drive Module XP-SLM .....	109
10.1	General .....	109
10.1.1	Displays and Connections .....	110
10.1.2	Assignment of Sockets on Module (Top View) .....	111
10.1.3	Technical data for SLM drive module XP-SLM .....	113
10.1.4	Interfaces .....	114
10.1.5	Application example .....	116
10.1.6	Wiring Example with MultiAx .....	117
10.1.7	Wiring example with M'Ax .....	119
11	SERCOS Module XP-SRC .....	121
11.1	Displays and Connections .....	122
11.2	Technical Data .....	123
11.3	Application Example .....	123
12	PROFIBUS-DP Modules .....	125
12.1	PROFIBUS-DP Master XF-DPM .....	125
12.2	Displays and Connections XF-DPM .....	126
12.3	Technical Data XF-DPM .....	126
12.4	PROFIBUS-DP Slave XF-DPS .....	127
12.5	Displays and Connections XF-DPS .....	128
12.6	Technical Data XF-DPS .....	128
12.7	PROFIBUS-DP Configuration Data .....	129
13	Error messages .....	131
13.1	Structure of Error Messages .....	131
13.2	PLC Error Messages .....	133
13.2.1	0x0100 PLC runtime error .....	133
13.2.2	0x0110 General PLC errors .....	138
13.2.3	0x0120 XRIO errors .....	139
13.3	CNC Error Messages .....	140
13.3.1	0x0200 General CNC errors .....	141
13.3.2	0x0210 Subsystem-specific CNC errors .....	142
13.3.3	0x0211 Decoder errors .....	143
13.3.4	0x0212 Error in automatic record insertion .....	148
13.3.5	0x0213 Error in tool nose radius compensation .....	149
13.3.6	0x0214 Error in interpolation preparation .....	149
13.3.7	0x0220 Axis errors .....	151
13.3.8	0x0221 SERCOS diagnosis status .....	153
13.3.9	0x0222 SERCOS status class 1 .....	154
13.3.10	0x0223 Manufacturer-specific SERCOS errors .....	155
13.3.11	0x0224 Manufacturer-specific SERCOS error texts .....	155
13.3.12	0x0230 Drive errors .....	156
13.4	CAN Error Messages .....	161
13.4.1	0x0300 CAN-specific operating system errors .....	161
13.4.2	0x0301 CAN driver errors .....	162
13.5	User-defined Error Messages .....	163
13.5.1	0x0400 User-defined error messages .....	163
13.6	NC Memory Access Errors .....	164
14	Technical data .....	166
15	Dimensions .....	168
16	Appendix .....	169
16.1	Trademarks .....	169
17	Safety-related Information .....	170
17.1	Correct Use of the System .....	170



17.2	Selection and Qualification of Personnel.....	170
17.3	Configuring, Programming, Installation, Commissioning and Operation.....	171
17.4	Maintenance.....	171
17.5	High Voltages.....	171
17.6	Dealing With Used Batteries .....	172
18	Pictures and Index .....	173

## Document conventions

This manual uses the following signs to indicate a safety-related warning:



Possible injuries or damage to the equipment if relevant safety precautions were not to be taken.

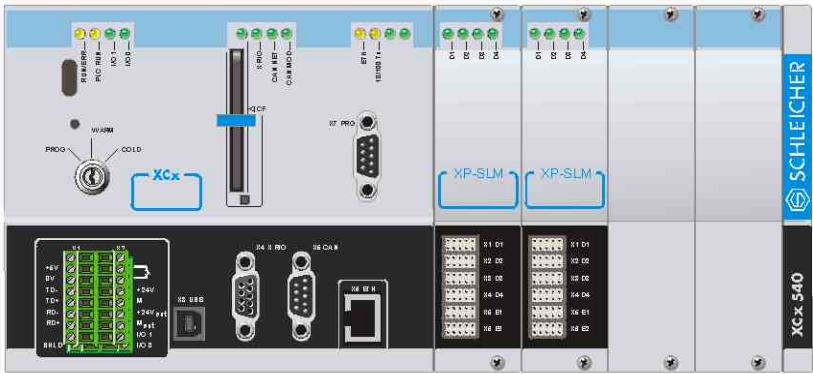
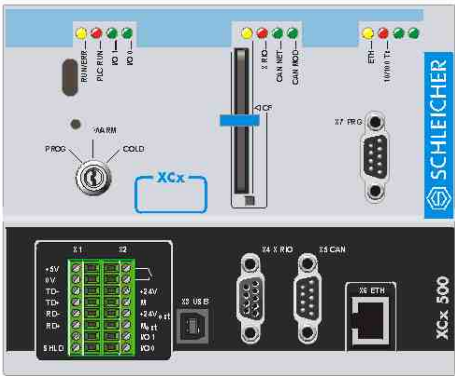
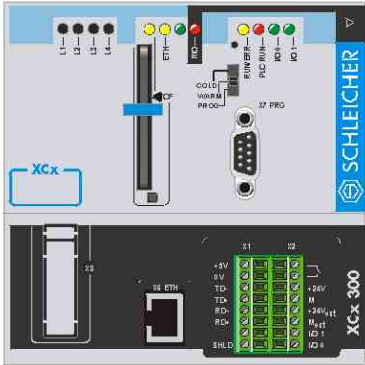


Important information on the handling of the automation system or the respective part in the operating manual.

Other objects are presented:

Object	Example
File names	MANUAL.DOC
Menus / Menu Items	<i>Einfügen / Graphik / Aus Datei [Insert / Graphic / From file]</i>
Paths / Directories	<i>C:\Windows\System</i>
Hyperlinks	<u><a href="http://www.schleicher-de.com">http://www.schleicher-de.com</a></u>
Program listings	MaxTsdr_9.6 = 60 MaxTsdr_93.75 = 60
Keys	<Esc> <Enter> (press first key, let go and press next key) <Ctrl+Alt+Del> (press all keys at the same time)

# Overview



For realizing of I/O's, the powerful RIO modules are used.

## XCx 300

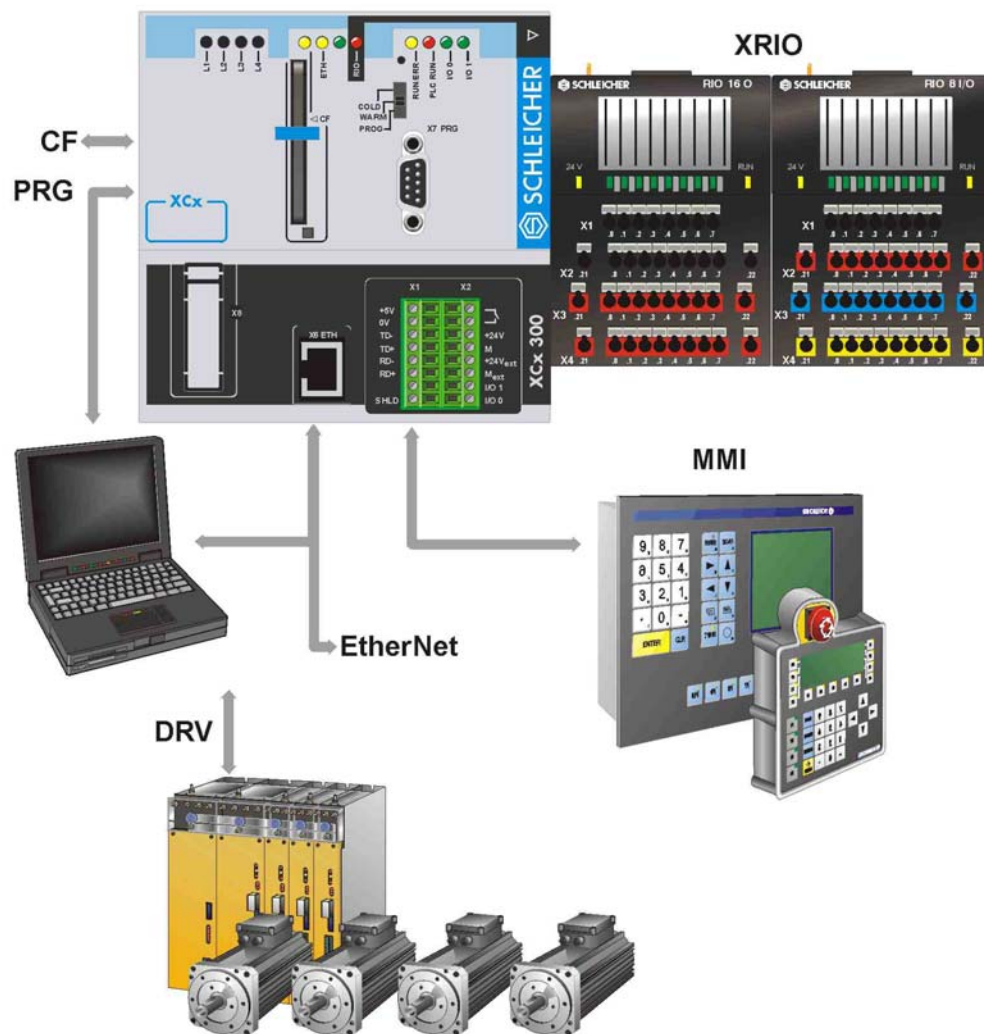


Figure 1: XCx 300 Overview

CF	CompactFlash makes it easy to save user programs and change the operating system
PRG	RS 232 programming device interface
DRV	Option: functional expansion with one expansion module, here connection of digital servo drives
EtherNet	RJ-45 connection for networking and programming via TCP/IP
MMI	RS 422 operator panel interface
XRIO	Direct interfacing of RIO modules for flexible implementation of direct digital and analog I/Os, counters and positioning of NC axes.

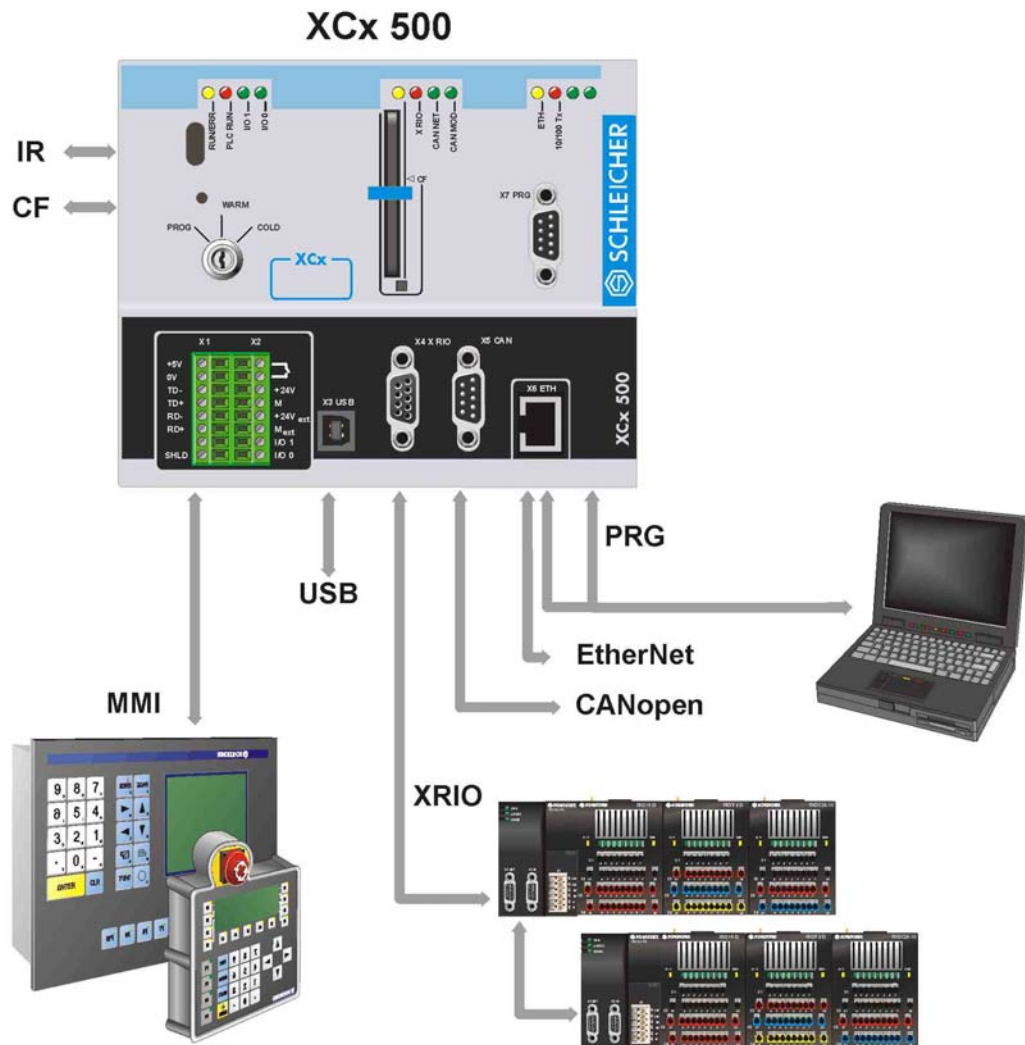


Figure 2: XCx 500 Overview

IR	IRDa infrared interface for diagnosis
CF	CompactFlash makes it easy to save user programs and change the operating system
MMI	RS 422 operator panel interface
USB	USB port as additional programming device interface
XRIO	Interfacing of RIO bus nodes for flexible implementation of direct digital and analog I/Os, counters and positioning of NC axes.
CANopen	CANopen field bus connection
EtherNet	RJ-45 connection for networking and programming via TCP/IP
PRG	RS 232 programming device interface

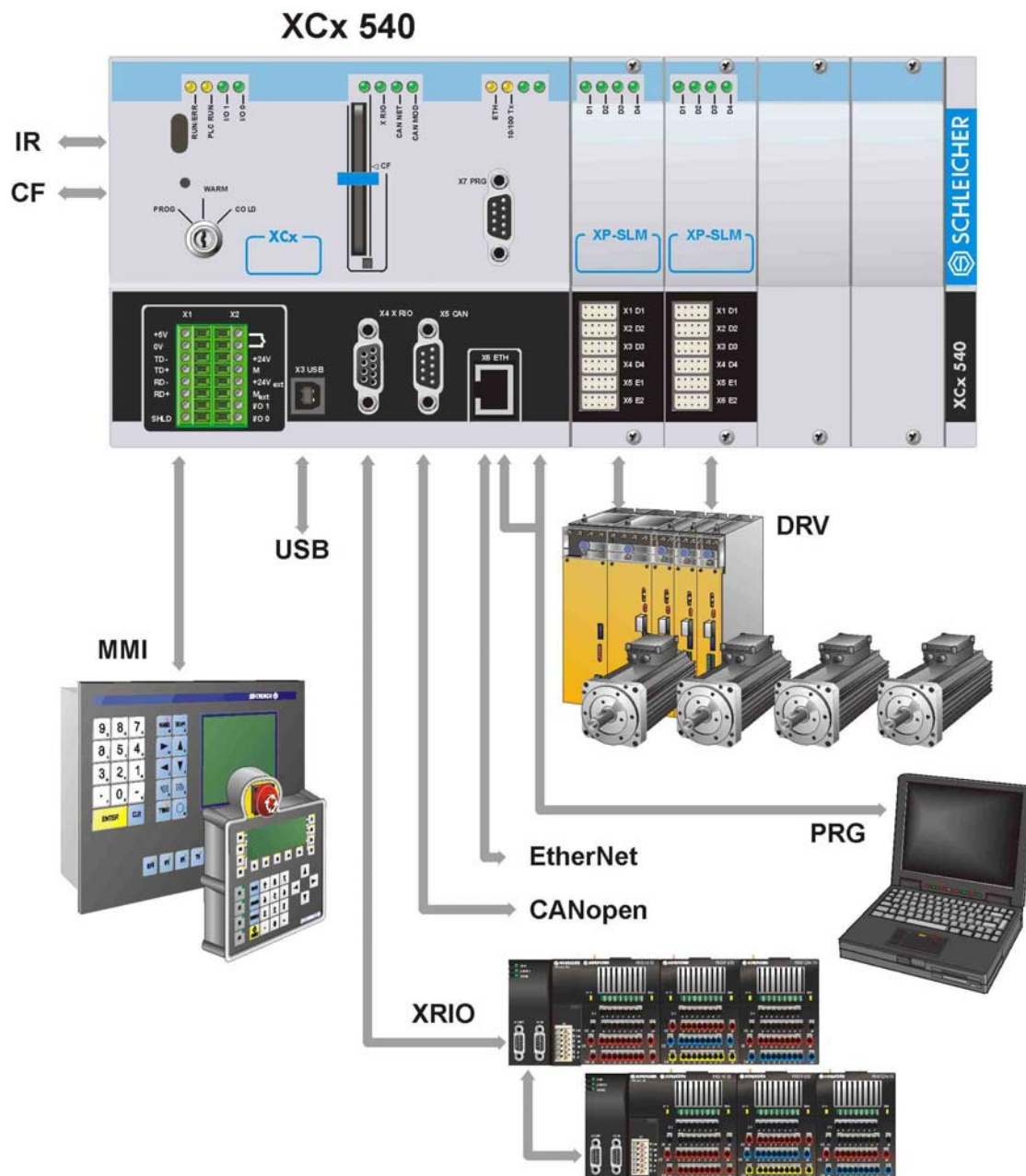


Figure 3: XCx 540 Overview

IR	IRDa infrared interface for diagnosis
CF	CompactFlash makes it easy to save user programs and change the operating system
MMI	RS 422 operator panel interface
USB	USB port as additional programming device interface
XRIO	Interfacing of RIO bus nodes for flexible implementation of direct digital and analog I/Os, counters and positioning of NC axes.
CANopen	CANopen field bus connection
EtherNet	RJ-45 connection for networking and programming via TCP/IP
PRG	RS 232 programming device interface
DRV	Functional expansion with expansion modules, here connection of digital servo drives via an SML interface (Speed Loop Module)



## 2 Installation

### 2.1 XCx

The XCx must be installed in earthed and closed metal cases (e.g. enclosures or cabinets) on an electrically conductive metal carrier board.



To protect the devices against electrostatic discharge, operating personnel must discharge any electrostatic charge they may be carrying before opening enclosures or cabinets.

#### 2.1.1 Installation position

The XCx can be installed in any position.

#### 2.1.2 Installation clearances

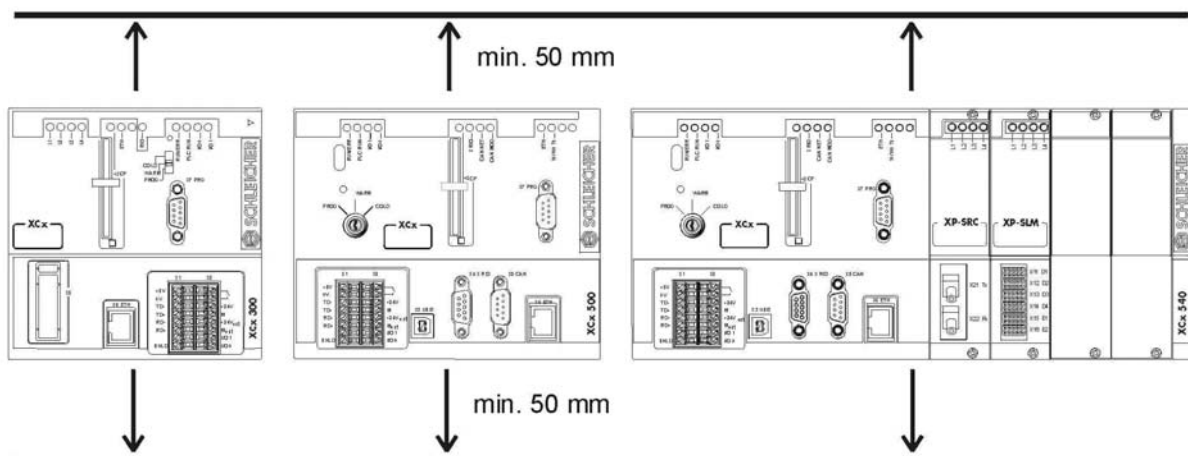


Figure 4: Installation clearances



### 2.1.3 Fixing the XCx to DIN rail

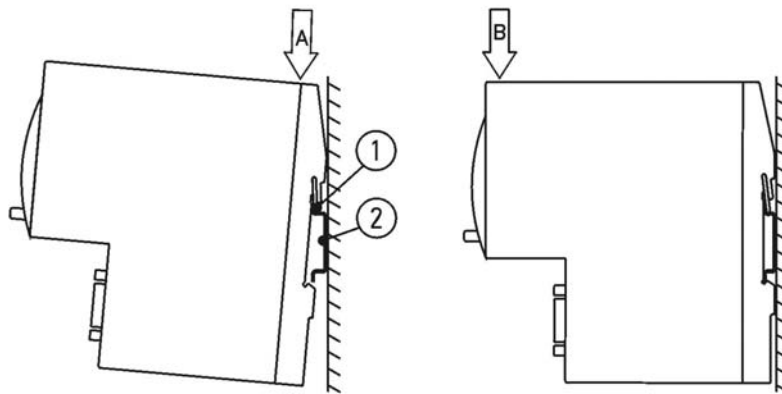
Use DIN rail type TS 35 mm / 7.5 to DIN EN 50022.

The DIN rail must be attached with a good electrical connection to an electrically conducting carrier board.



DIN rail installation is sufficient for normal mechanical loads.  
For higher mechanical loads please refer to "Fixing with screws" (only XCx 500 and XCx 540).

### Assembly

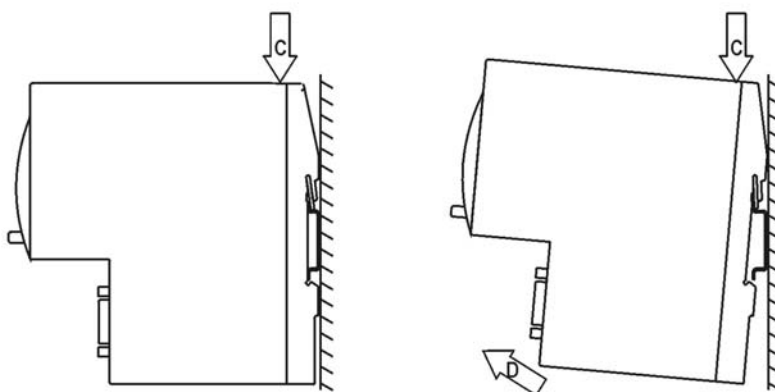


**A** Place device with guide (1) on DIN rail (2).

**B** Press until the device engages.

Figure 5: Fixing the XCx on DIN rail

### Disassembly



**C** Push device down.

**D** Pull away while pushing down.

Figure 6: Disassembly the XCx



The RIO expansion modules must be fixed on the same DIN rail with the XCx 300.

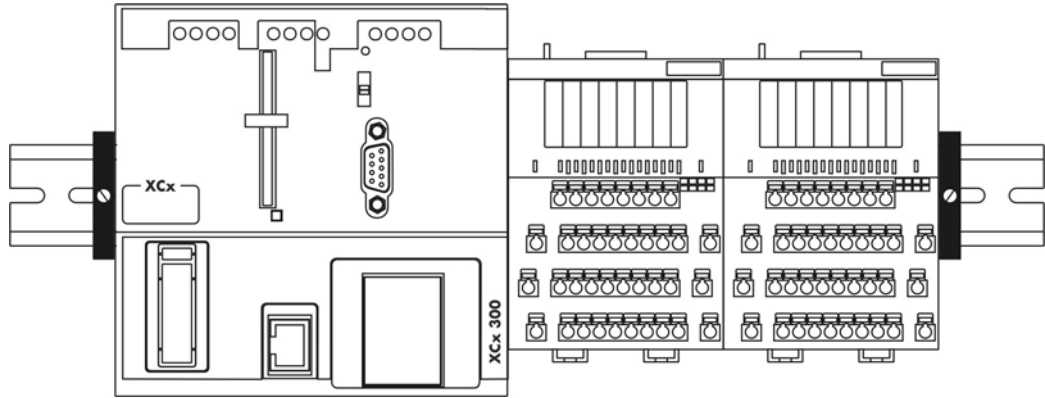


Figure 7: Installation RIO expansion modules with XCx 300

### 2.1.4 Fixing with screws



Only XCx 500 and XCx 540 are suitable for fixing with screws.

The controller must be attached with a good electrical connection to an electrically conducting carrier board.

Use 4 M4 machine screws with toothed lock washers.

The fixing holes for fixing with screws are on the left and right sides of the base plate of the device. They are easily accessible from outside.

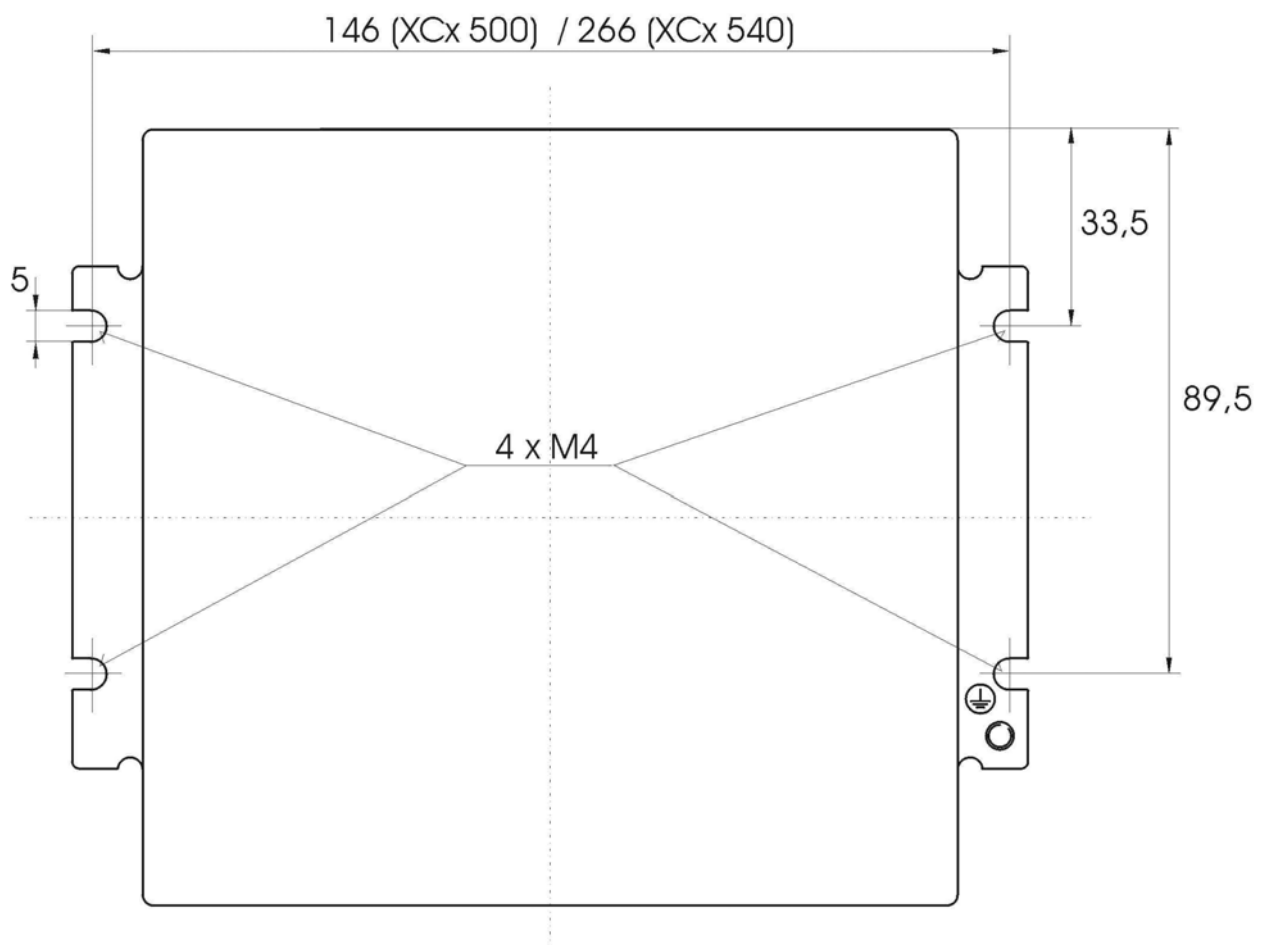


Figure 8: Dimensions of holes for fixing with screws

### 2.1.5 Connecting the voltage supply XCx

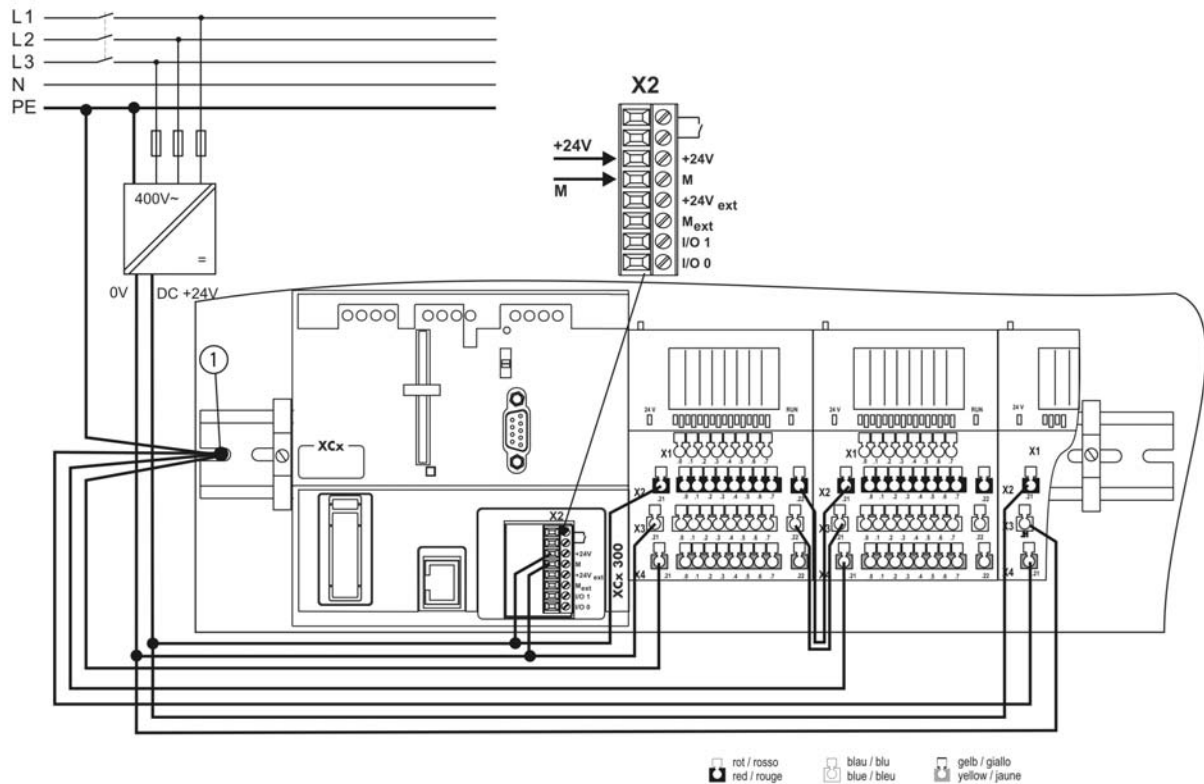


Figure 9: Connecting the voltage supply XCx 300

If no shield rail is used the metal carrier board can be connected to ground (PE) at a central point (1).  
See also section XRIO Installation.

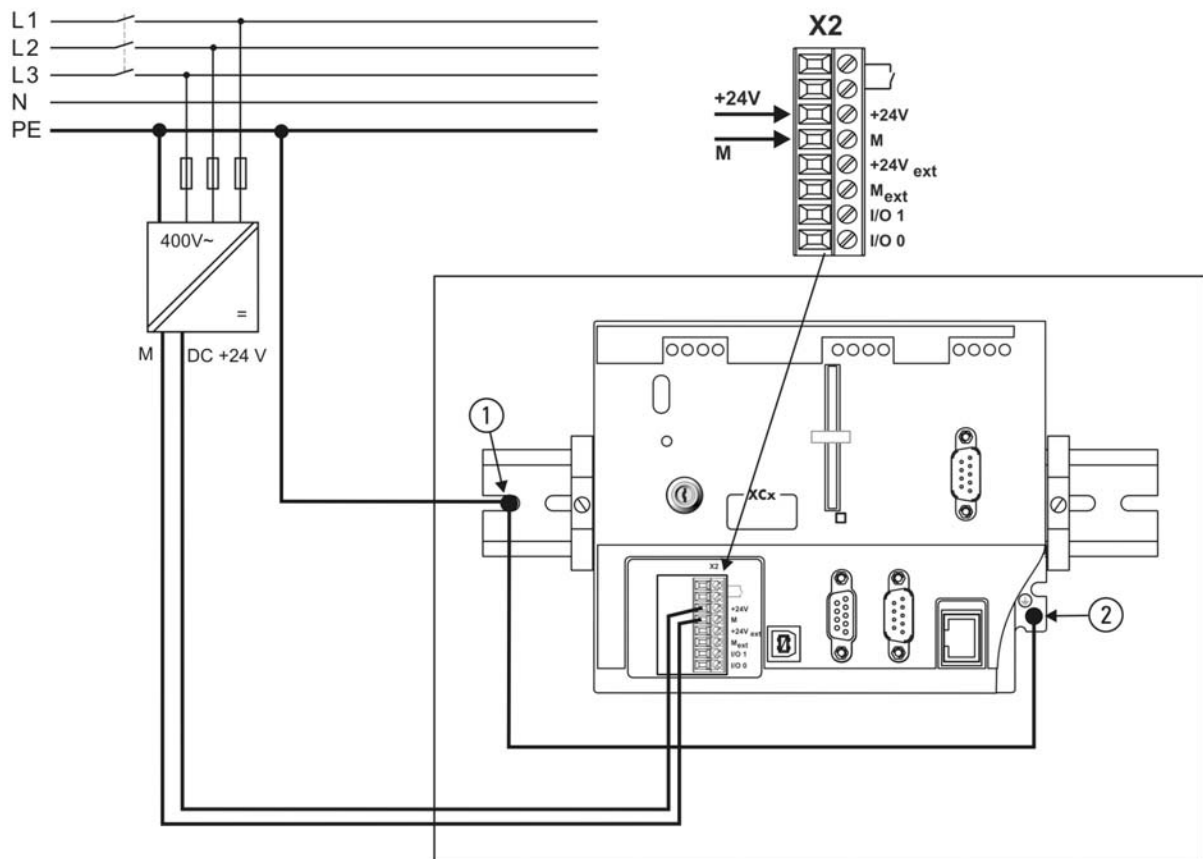


Figure 10: Connecting the voltage supply XCx 500 and XCx 540

If no shield rail is used the metal carrier board can be connected to ground (PE) at a central point (1).

The earthable point (2) on the back of the device can be connected if required using an M4 machine screw with a toothed lock washer.

## 2.2 XRIO Installation

See operating manual RIO Bus Couplers part no.: 32215700 section Installation and section XRIO.

If the XCx 300 is used pay attention to the following features:

Install the modules directly next to each other and secure them with an end clip.

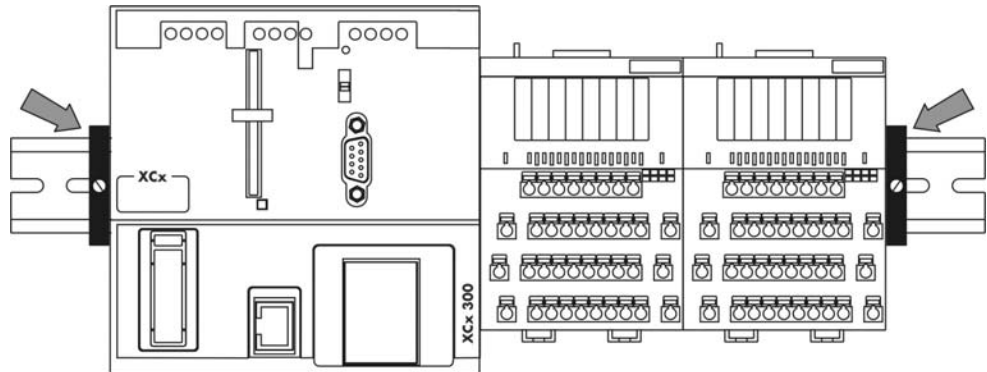


Figure 11: End clips with XCx 300

The orange slide contacts on top of the module connect the modules to the bus coupler.



Make sure that the slide contacts are open when actuating (opening) the spring terminals in order to reduce mechanical wear on the contact points.

Align the modules before you close the slide contacts. Do not use force when closing the slide contacts.  
Close the slide contacts before commissioning the unit.

Do not open the slide contacts during operation.

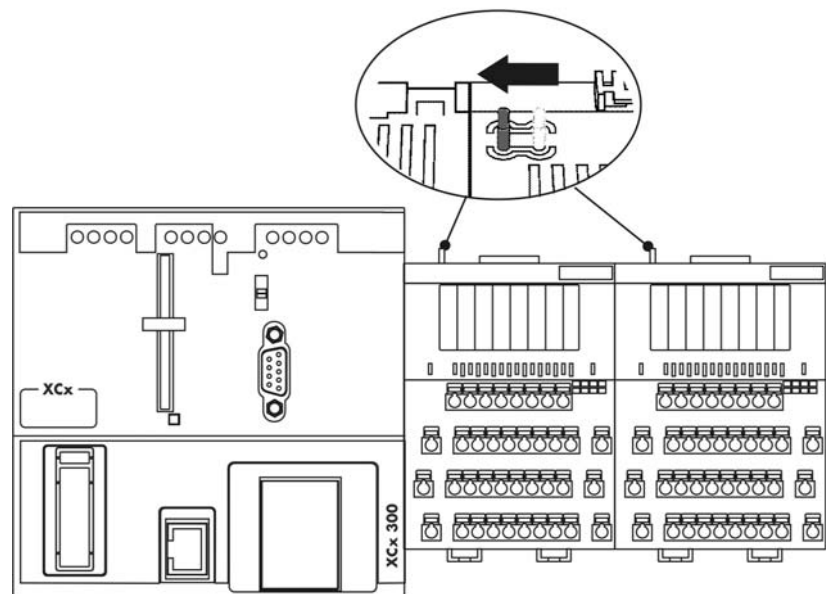
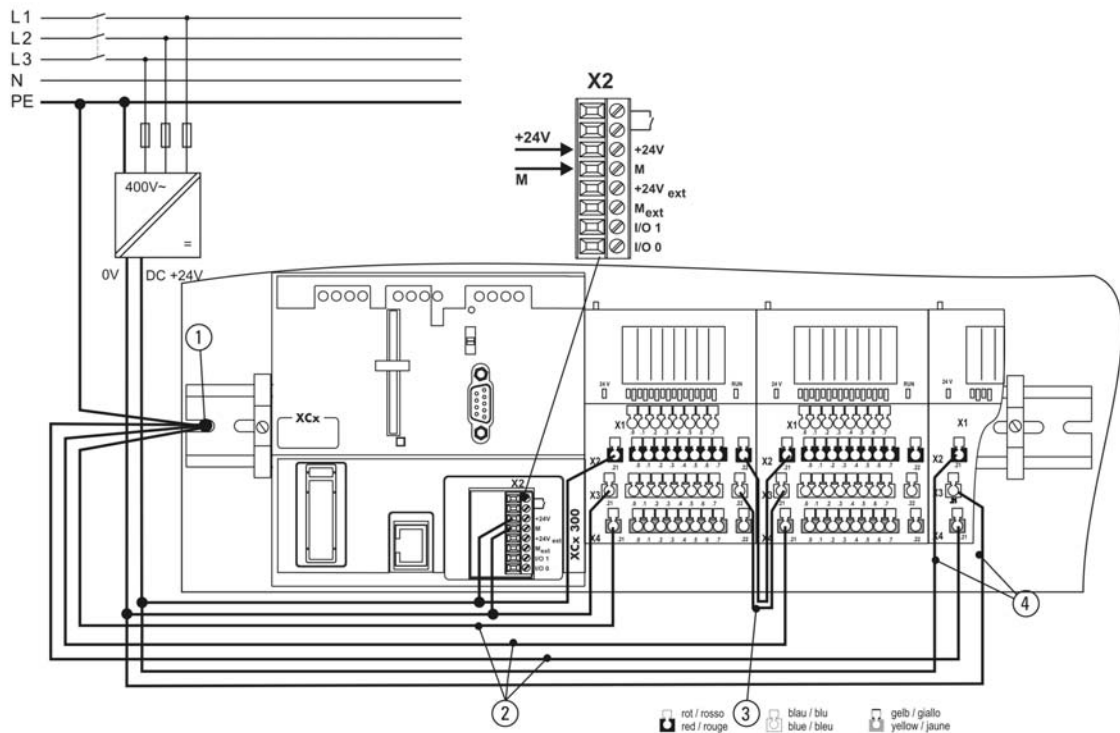


Figure 12: Slide contacts RIO with XCx 300

## Connecting the power supply



The diagram shows a bus coupler with expansion modules of the RIO 8 I/O type.

For details on the diagram refer to the following section.

### Connecting the DIN rail to ground (1)

The DIN rail which holds the modules must have a broad-surface connection to ground providing adequate conduction.

### Connecting the expansion modules to ground (2)

The 8 I/O expansion module have a connecting terminal marked with the ground symbol. To increase interference immunity this terminal must be connected to ground (or PE potential) with a conductor which is as short as possible ( $2.5\text{mm}^2$ ).

The contact spring located in the module clamp base is used to divert EMC interference. This spring provides the connection between the PCB shield potential and the DIN rail. Do not fit modules without a contact spring or with a defective contact spring.

### Relaying the power supply (3)

To provide an optimum wiring the power supply can be relayed from one module to the next. When using modules with digital outputs the current on each relay terminal must not exceed a specific value. If the maximum current is exceeded an additional power supply feed must be provided (see below).



### Additional power supply feeds (4)

When relay terminals are used to relay the power supply from one module to the next the current on each relay terminal must not exceed a specific value.

---



Make sure that the load on a terminal does **not exceed  $I_{max} = 8A$** .

If the maximum current may be exceeded you need to provide additional power supply feeds.

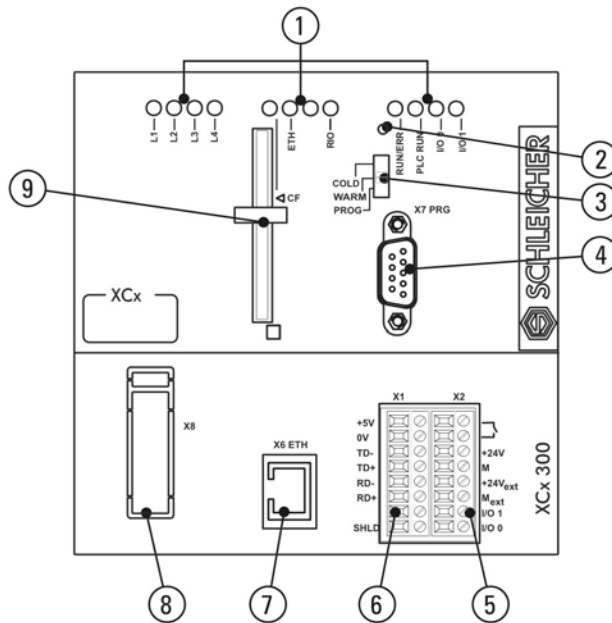
---



### 3 Displays, Connections and Controls

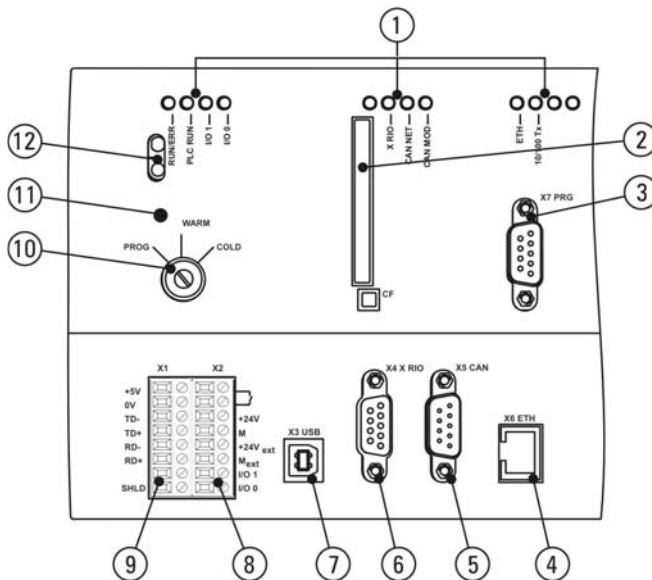
#### 3.1 Overview

##### XCx 300



- 1 LED Displays
- 2 RESET Button
- 3 Mode Switch
- 4 X7 PRG RS232 Interface for programming devices
- 5 X2 Connection for supply voltage, direct I/O and BUSY contact
- 6 X1 RS422 Interface for connecting operator panels and displays
- 7 X6 ETH RJ 45 EtherNet Connection
- 8 Slot for expansion module
- 9 CompactFlash

##### XCx 500 / XCx 540



- 1 LED Displays
- 2 CompactFlash
- 3 X7 PRG RS232 Interface for programming devices
- 4 X6 ETH RJ 45 EtherNet Connection
- 5 X5 CANopen Field Bus Connection
- 6 X4 X RIO direct interfacing with RIO modules
- 7 X3 USB for additional programming devices
- 8 X2 Connection for supply voltage, direct I/O and BUSY contact
- 9 X1 RS422 Interface for connecting operator panels and displays
- 10 Mode Switch
- 11 RESET Button
- 12 IRDa Infrared Interface for diagnosis applications



### 3.2 LED Displays

#### XCx 300

LED-designation	Colöör	Status	Meaning
<b>L1 / L2 / L3 / L4</b>			unused (free for expansion module)
<b>CF</b>			Compact Flash
	green	ON	Memory access
	red	ON	Access error
<b>ETH</b>			EtherNet network
	green	Flashing	Network access
	red	ON	Network error
<b>XRIO</b>			RIO direct connection
	green	ON	Operational
		Flashing	Pre-Operational
	yellow	ON	Bus error
	red	ON	Frame error
<b>RUN/ERROR</b>			CPU status
		OFF	No operating voltage
	green	ON	Operating voltage ok, no error
	red	ON	CPU not running (watchdog)
		Flashing	CPU has detected a fatal error
<b>PLC RUN</b>			SPS status
		OFF	PLC stop
	yellow	ON	PLC running
		Flashing	PLC running, but outputs shut down (ready-for-operation relay released)
<b>I/O 0</b> <b>I/O 1</b>			Combination I/O
	yellow	OFF	Input / output not set
		ON	Input / output set

Error messages are stored in the active error buffer and in the error logbook. Error messages include error numbers and additional informations.

Use the <Ctrl+?> keys in the SCHLEICHER-Dialog software to read the active error buffer and the log book.

## XCx 500 / 540

LED designation	Colour	Status	Meaning
<b>RUN/ERROR</b>			CPU status
		OFF	No operating voltage
	green	ON	Operating voltage ok, no error
	red	ON	CPU not running (watchdog)
		Flashing	CPU has detected a fatal error
<b>PLC RUN</b>			PLC status
		OFF	PLC stop
	yellow	ON	PLC running
		Flashing	PLC running, but outputs shut down (ready-for-operation relay released)
<b>I/O 1 I/O 2</b>			Combination I/O
	yellow	OFF	Input / output not set
		ON	Input / output set
<b>CF</b>			Compact Flash
	green	ON	Memory access
	red	ON	Access error
<b>XRIO</b>			RIO direct connection
	green	ON	Operational
		Flashing	Pre-Operational
	red		Frame error
<b>CAN NET</b>			CAN network status
		OFF	CAN state prepared
	green	ON	CAN state operational
		Flashing	CAN state pre-operational
	red	ON	Bus Off
		Flashing	CAN error
<b>CAN MOD</b>			CAN module status
	green	ON	CAN stack initialized
		Flashing	Invalid CAN configuration
	red	ON	Control unit not ready, or serious error
		Flashing	Error in controller
<b>ETH</b>			EtherNet network
	green	Flashing	Network access
	red	ON	Network error
<b>10/100 Tx</b>			EtherNet transmission rate
	yellow	ON	Transmission rate 100 Mbit/s
		OFF	Transmission rate 10 Mbit/s

Error messages are stored in the active error buffer and in the error logbook. Error messages include error numbers and additional information.

Use the <Ctrl+?> keys in the SCHLEICHER-Dialog software to read the active error buffer and the log book.



### 3.3 CompactFlash

The operation system, important configuration files, the PLC program (boot project and PLC source) and the CNC files are stored on the CF card.

CF card access is indicated by the LED CF.

The CF card is hot pluggable. Of course pay attention to the following definitions:



Unplug the CF card only if:

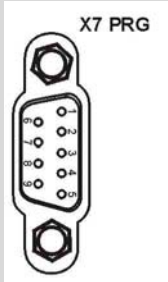
- no CF card access (LED CF must be dark)
- the controller stand in the STOP operation mode

Plug the CF card only if:

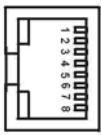
- the controller stand in the STOP operation mode

Lost of data or watch dog events can occur, when not pay attention to this definitions.

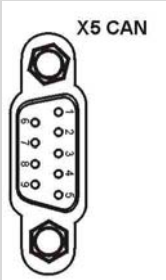
### 3.4 X7 PRG RS232 Interface for Programming Devices

X7 Subminiature, 9-pin, plug connector				
	Pin	Designation		Explanation
	1	(DCD)	(Data Carrier detect)	not used
	2	RD	Receive Data	Input
	3	SD	Send Data	Output
	4	DTR	Data Terminal ready	Bridge to pin 6
	5	GND	Logic Ground	Not for shield
	6	DSR	Data set ready	Bridge to pin 4
	7	RTS	Request to send	Bridge to pin 8
	8	CTS	Clear to send	Bridge to pin 7
	9	(Ri)	(Ring Indicator)	not used

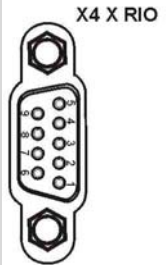
### 3.5 X6 ETH RJ 45 EtherNet Connection

X6 RJ 45			
	Pin	Designation	Explanation
	1	TX+	Transmitted data plus
	2	TX-	Transmitted data minus
	3	RX+	Received data plus
	4	nc	Not connected
	5	nc	Not connected
	6	RX-	Received data minus
	7	nc	Not connected
	8	nc	Not connected

### 3.6 X5 CANopen Field Bus Connection

X5 Subminiature, 9-pin, plug connector			
	Pin	Designation	Explanation
	1	NC	Not connected
	2	CAN_L	
	3	V-	Ground
	4	nc	Not connected
	5	Drain	Shield connection optional
	6	V-	Ground
	7	CAN_H	
	8	nc	Not connected
	9	V+	Power supply

### 3.7 X4 X RIO Connection


X4 Subminiature, 9-pin, socket connector			
	Pin	Designation	Explanation
	1	<b>RD+</b>	Received data plus
	2	<b>TD+</b>	Transmitted data plus
	3	<b>0 V</b>	Ground Power supply
	4		Not connected
	5	<b>+5 V</b>	Power supply
	6	<b>RD-</b>	Received data minus
	7	<b>TD-</b>	Transmitted data minus
	8		Not connected
	9		Not connected

### 3.8 X3 USB Connection (Universal Serial Bus)

In preparation

Type B female connector corresponding to Universal Serial Bus Specification Revision 2.0 (USB2.0)

The USB connector to host is always type A, the connector to device type B. This prevents wiring mistakes and ring connections.

X3 USB connection socket type B			
	Pin	Designation	Explanation
	1	VCC (VBus)	Power supply via bus (bus-driven)
	2	- Data	Data-
	3	+ Data	Data+
	4	Ground (Gnd)	Power supply ground

### 3.9 X1 RS422 Interface for Connecting Operator Panels and Displays

X1 8-pin screw terminal block		
	Designation	Explanation
	<b>+5 V</b>	Supply voltage +5 V
	<b>0 V</b>	Supply voltage 0 V
	<b>TD-</b>	Transmitted data minus
	<b>TD+</b>	Transmitted data plus
	<b>RD-</b>	Received data minus
	<b>RD+</b>	Received data plus
		Not connected
	<b>SHLD</b>	Shield

### 3.10 X2 Connection for Supply Voltage, Direct I/O and BUSY Contact

X2 8-pin screw terminal block		
	Designation	Explanation
		BUSY contact (PLC running)
	<b>+24 V</b>	Supply voltage DC +24 V
	<b>M</b>	Supply voltage ground
	<b>+24 V ext</b>	Supply voltage DC +24 V direct inputs/outputs
	<b>Mext</b>	Supply voltage ground direct inputs/outputs
	<b>I/O 1</b>	Direct input/output* 1
	<b>I/O 2</b>	Direct input/output* 2
		*Connection can be used as input or output

### 3.11 Mode Switch

The mode switch is a three-position key-operated switch. These three positions determine the start up behaviour of the controller. The key-operated switch prevents unauthorized manipulation of the start up behaviour.

Mode switch		
	Position	Explanation
	<b>PROG</b>	PLC stop, programming mode (Only in this position is the RESET button operational)
	<b>WARM</b>	PLC warm start
	<b>COLD</b>	PLC cold start (reinitialisation retain variables)

### 3.12 RESET Button

The RESET button causes a hardware reset, equivalent to power off.  
The RESET button functions only in PROG mode (see Mode Switch).

### 3.13 IRDa Infrared Interface

In preparation

## 4 Quick Introduction

The XCx startup described in this section can be carried out without in-depth knowledge. In order to achieve success quickly, you have to follow the startup steps exactly, and keep to the specified conditions (e.g. I/O configuration).

Important subjects such as XRIO, task system and CANopen are dealt with in more detail in later sections.

### 4.1 Starting up Communication

#### 4.1.1 Communication connections

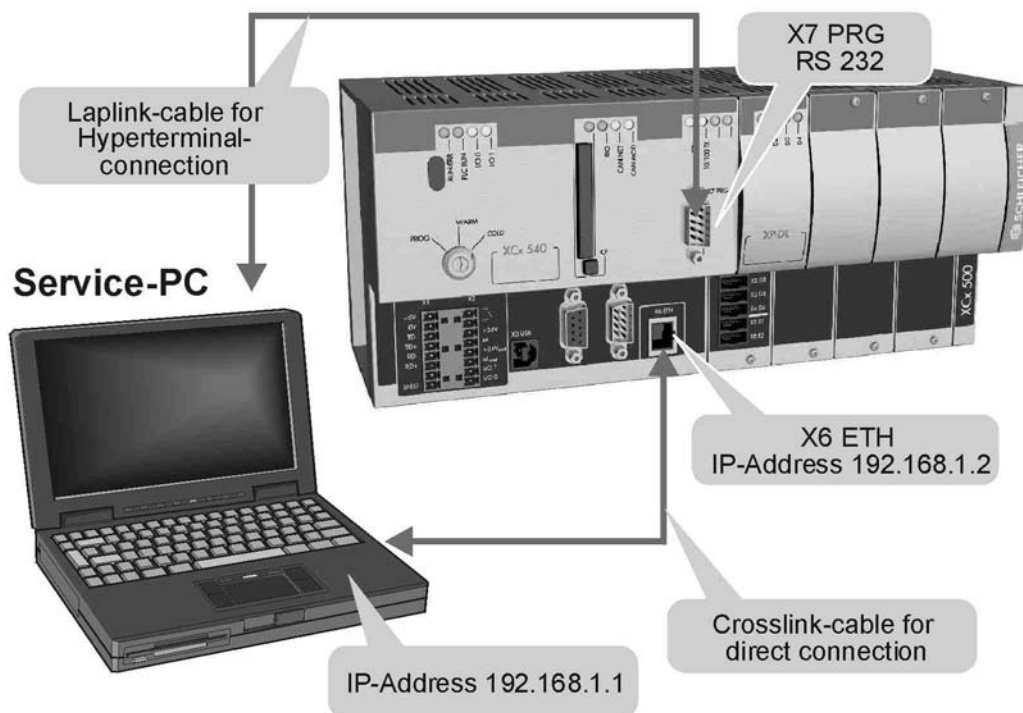


Figure 13: Connecting the service PC to the XCx

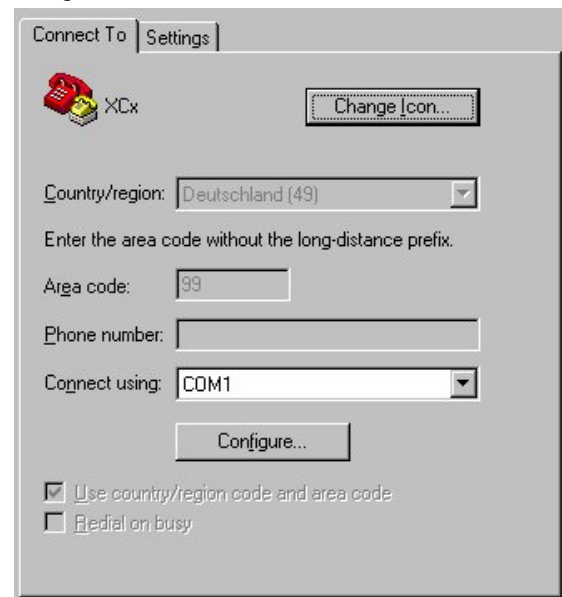
#### 4.1.2 Starting up the serial connection via the RS232 interface

A connection via the serial interface is very useful when you first start up the XCx. It is easy to get it running, and it allows you to see the first reactions of the XCx.

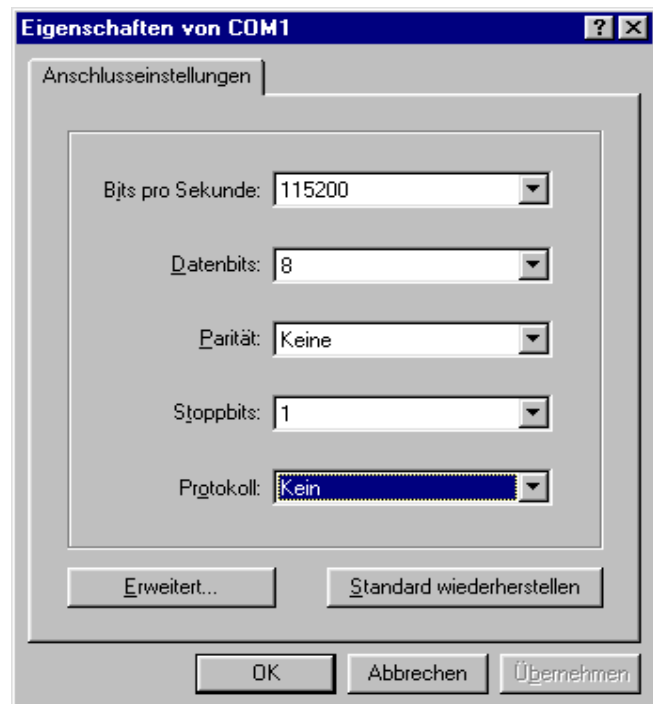
- Connect the cable between COM1 or COM2 on the service PC and X7 PRG on the XCx.
- Go to *Start/Programs/Accessories/Hyperterminal* on the PC and start *Hypertrm.exe*. Enter a name, for example XCx, and select a symbol.



- In Properties of .../Connect To select direct connection via COM1 and then click the Configure button.



- Set the following parameters in *Properties of COM1*.



- Switch the XCx on or RESET.  
The boot log appears in the hyperterminal dialog window while the controller is starting up.



### 4.2 Starting up the Ethernet Interface

We recommend using the Ethernet connection for programming with MULTIPROG, in order to gain maximum performance during configuring, programming and startup.

You will need a PC with a network card for Ethernet and a cross-link cable for direct connection to the XCx.

#### 4.2.1 Output current IP address of XCx via the RS232 interface

Enter the instruction *version* in the *Hyperterminal* dialog window.

The version information is displayed.

```
-> version
VxWorks .....
Kernel: .....
Made on .....
Boot line:
ata=.....e=192.168.1.2:ffffff00 u=target pw=target
tn=X
Cx .....
value = .....
->
```

The IP address (192.168.1.2), subnet mask (255.255.255.00), username (u=target) and password (pw=target) are in the line e=.

#### 4.2.2 Setting the IP address of the service PC

1. The IP address of the service PC is set via *Start/Settings/Control Panel/Network/Configuration*.
2. Select TCP/IP and click *Properties*.  
Enter IP address: **192.168.1.1**  
Enter subnet mask: **255.255.255.0**
3. Now the PC must be restarted.
4. Output the IP address to check the service PC  
Windows® 9x    *Start/Run/winipcfg*  
                  At network card info select <Name of network card> instead of *PPP-Adapter*.  
Windows® NT    At *Start/Programs/MS-DOS Prompt*  
                  enter *ipconfig* and start.

**192.168.1.1** should now appear as the IP address of the network card.

**255.255.255.0** should now be displayed at Subnet Mask.

Acknowledge with OK or close window.

### 4.2.3 Setting up and checking the Ethernet connection

1. Set the XCx lockswitch to PROG and connect the Ethernet connection on the PC to the X6 ETH Ethernet interface on the XCx using a cross-link cable.
2. Select *Start/Programs/MS-DOS Prompt* on the PC.
3. Enter: *doskey* (Doskey installed appears. Use as help for another test).
4. Enter :*ping 192.168.1.2* (this IP address is preset when the XCx is delivered).
5. Start with OK
6. If everything is OK the following appears:  
pinging ...  
Response from 192.168.1.2.....  
Response ...  
Response ...  
Response ...

If the XCx is not recognized an error message will appear. Check the IP address and the cable connections.

### 4.3 Changing the IP Address of the XCx

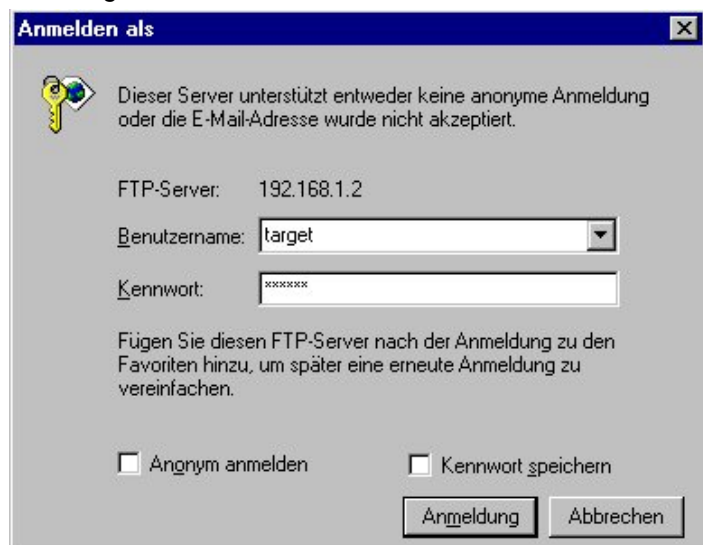
If the XCx is operated as described above via a crosslink cable you do not have to change the IP address.

- To change it, start a program for FTP access (File Transfer Protocol) on the PC (e.g. Windows® Internet Explorer 6).



For FTP access the TCP/IP connection to the XCx has to be active. (see "Checking the IP address with ping")

- Enter *ftp://192.168.1.2* in the address line and start.
- To log on, enter the user name and the password for the XCx:  
*User name = target*  
*Password = target*



- Now the content of the compact flash should be in Windows®



Internet Explorer.

- Copy the *BootLine.ini* file from the \OS directory to the hard disk.
- Edit the *BootLine.ini* file.

```
InetOnEthernet  =192.168.1.1
InetMask        =255.255.255.0
UserName        =target
Password        =target
```

- Save the file and copy it back to the compact flash on the XCx. If the file cannot be copied, you will have to cancel the read-only attribute in the *BootLine.ini* file on the compact flash. To do this enter and execute the instruction attrib "*bootline.ini*", "-R" using the hyperterminal via the RS232 interface.
- Set the XCx lockswitch to PROG and switch the XCx off and on again (or RESET).

## 4.4 Installing the MULTIPROG, AddOns and OPC Server

### 4.4.1 Installing MULTIPROG



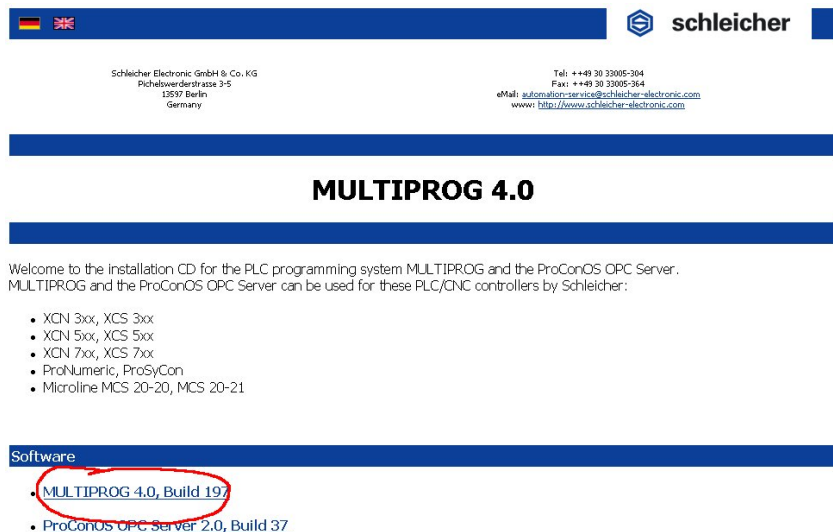
The entire programming software has the components: MULTIPROG, OPC server, AddOns for MULTIPROG and Schleicher Dialog.

All components must be installed in this order and separate.

Two CD ROM are delivered with the controller:

Name	Content	Part no.
MULTIPROG	<ul style="list-style-type: none"> <li>MULTIPROG programming software</li> <li>OPC-Server</li> </ul>	320 385 65
Service Pack	<ul style="list-style-type: none"> <li>Software for all Schleicher controllers</li> <li>AddOns for MULTIPROG</li> <li>Schleicher Dialog</li> <li>Other resources like manuals and service information.</li> </ul>	320 157 96

- Put the CD named MULTIPROG in the drive.  
The auto run function of the CD will start the Internet Explorer.



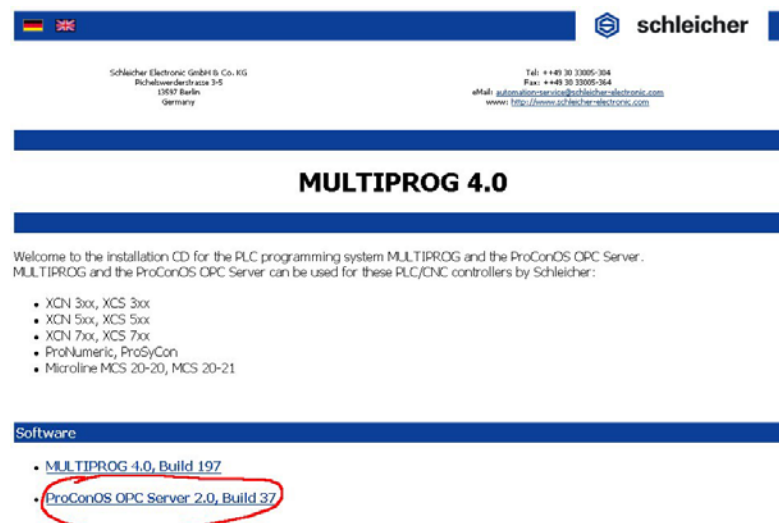
- Pay attention to "Installation guides".  
Choose *MULTIPROG* and start this.



All installation settings remain unchanged.



If MULTIPROG 1.2 for Schleicher MicroLine and ProNumeric/ProSycon controllers is already installed, the installed version must not be overwritten. In this case you will have to install MULTIPROG in a different folder.


- Next step install OPC server.



- Choose ProConOS OPC Server and start this. The OPC server should be installed in the MULTIPROG path.
- Restart the PC after OPC installation.

- Next step: Install *AddOns* for MULTIPROG.  
Put the CD named Service Pack in the drive.  
The auto run function of the CD will start the Internet Explorer.


**schleicher**

Schleicher Electronic GmbH & Co. KG  
Pichelswerderstrasse 3-5  
13597 Berlin  
Germany

Tel: ++49 30 33005-304  
Fax: ++49 30 33005-364  
eMail: [automation-service@schleicher-electronic.com](mailto:automation-service@schleicher-electronic.com)  
www: <http://www.schleicher-electronic.com>

[Software](#)   [Documentation](#)   [Service information](#)   [EDS, GSD files](#)   [Training material](#)

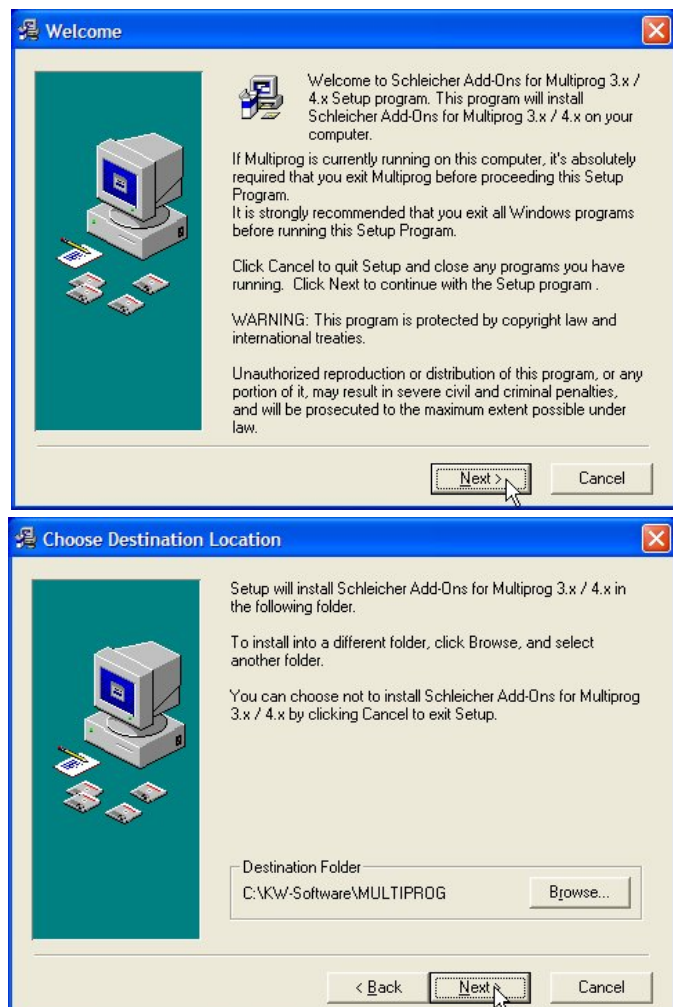
## Schleicher Service Pack 05.17

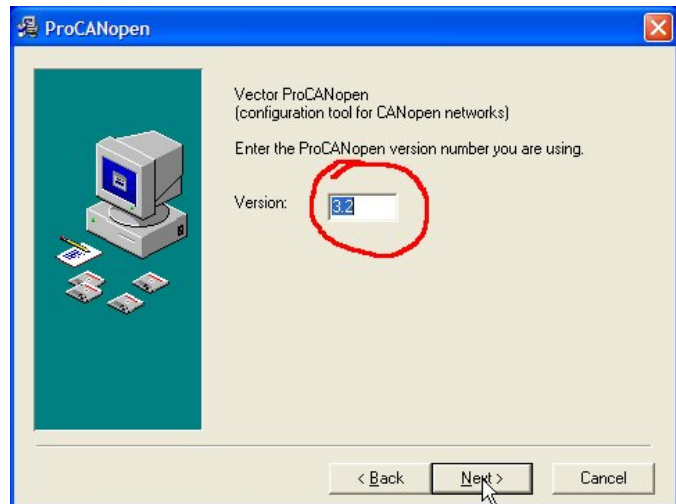
Welcome to the Schleicher Service Pack CD, version 05.17 (25 April 2005). On this CD you will find the current software for the controller families XCx and ProNumeric / ProSyCon as well as the associated documentation.

**Software**

- XCx 300, software version 05.02/2
  - [Controller software](#)
  - [AddOns for Multiprog Dialog](#)

- Choose AddOns for MULTIPROG and start this.





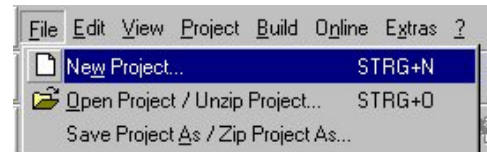
- Enter the current ProCANopen version (here 3.2). You will need ProCANopen later for starting up the CANopen network.

You do not have to restart the PC at this point.

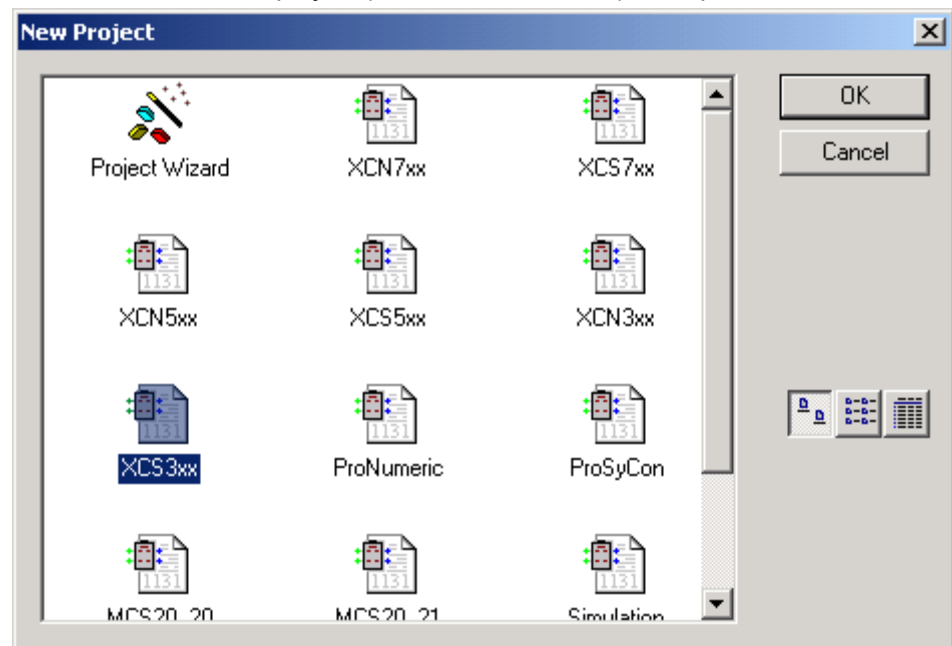


## 4.4.2 Starting MULTIPROG and opening a new project

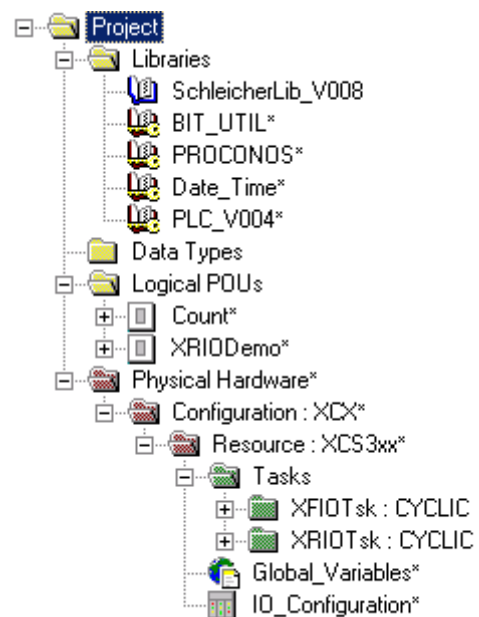
- Start MULTIPROG and select *File/New Project*.



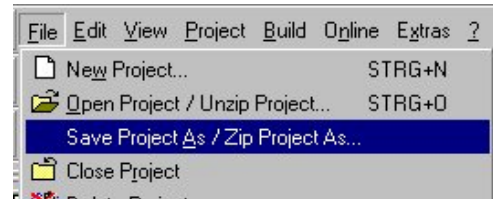
- Select a project (XCS3xx in this case) and open with *OK*.



- Answer Yes to all questions concerning conversion of libraries.
- If the project has opened successfully the project tree will appear in the project window.

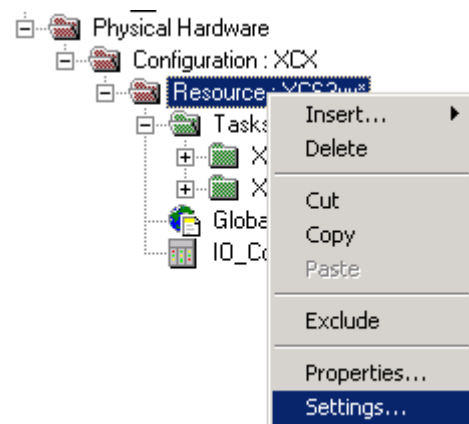


Save the project under a new name with *File/Save Project As*.

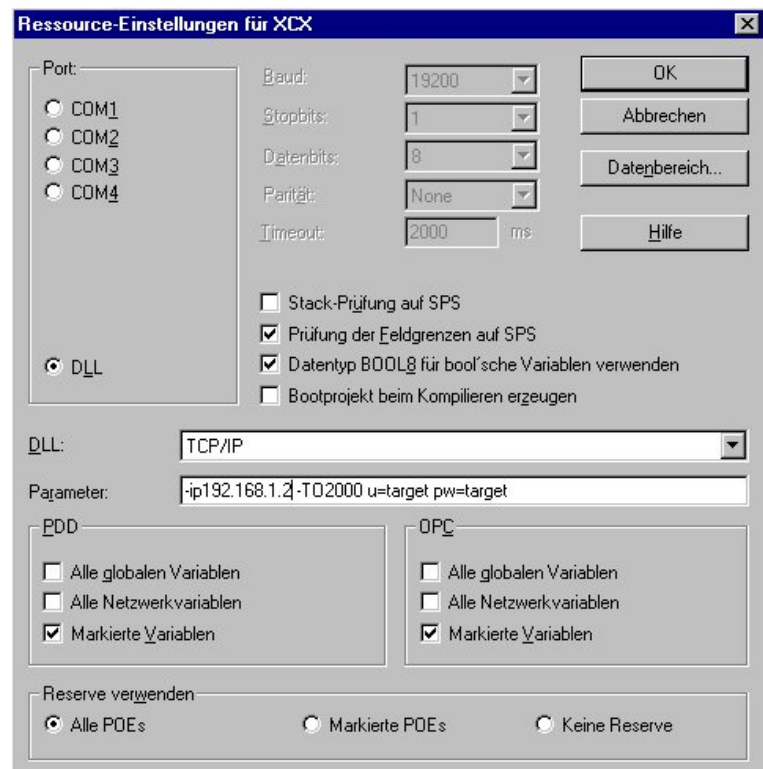


### 4.4.3 Setting the Ethernet connection in MULTIPROG

- In the *Resource* project tree: Click with the right mouse button on *XCS3xx* to open the context menu.



- Then select *Settings* and set the parameters as shown below.



Meaning of parameters:

-ip192.168.1.2 = IP address of XCx

-TO2000 = Time Out 2000 ms

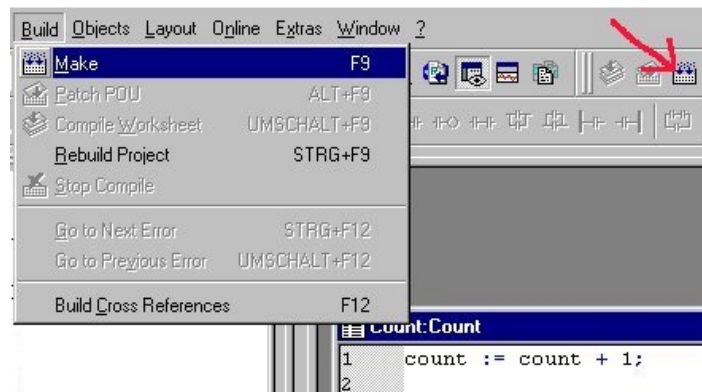
u=target = Username

pw=target = Password

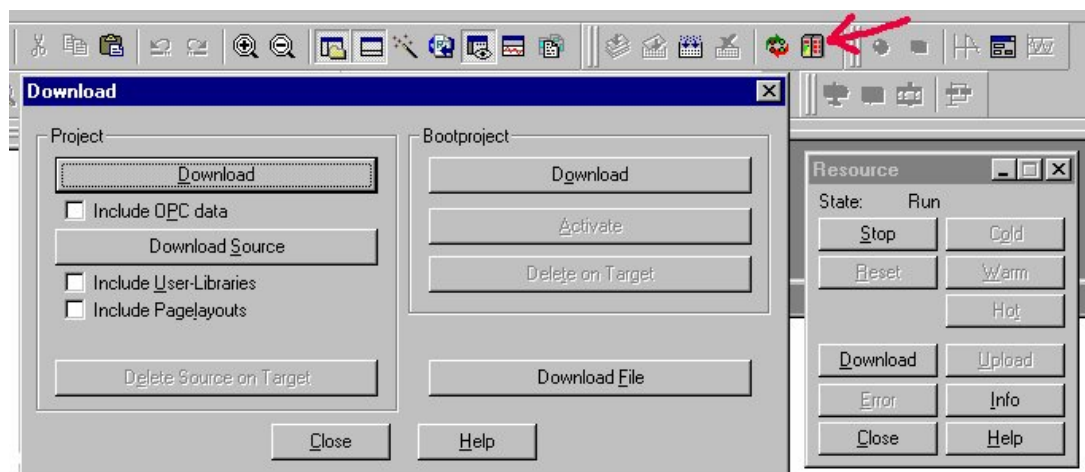
All other settings remain unchanged.

#### 4.4.4 Compiling a project and sending to the XCx

- Select *Code/Make* to compile the project.  
(or <F9> or *Make* button, see arrow)

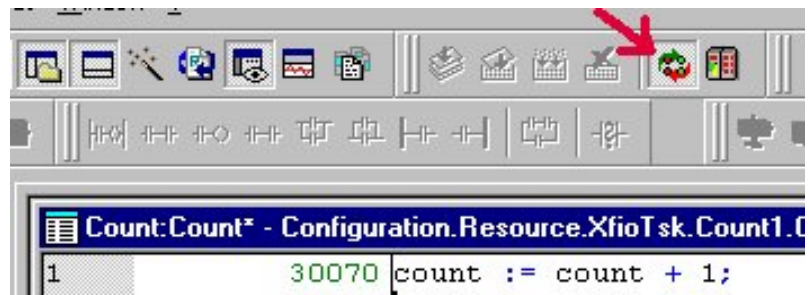


- The progress of compiling is displayed in the message window below.  
Error messages, warnings and other information are displayed here.  
Click the selection box for error, warning, etc., to display the message in more detail.  
If errors are displayed, you can jump directly to the line of the PLC program that caused the error by double-clicking in the error line.
- Send the project via the Ethernet connection.
  1. Click on the *Project Control Dialog* button (see arrow).
  2. Click on the *Download* button in the XCx control dialog to open the *Download* dialog window.
  3. Under project, select *Download* again, overwrite the existing project in the XCx.
  4. Use the *Cold* (cold start) button in the control dialog to start the program on the XCx. The yellow PLC RUN LED lights up.





With the Debug on/off button (see arrow) you can display the content of the variables online on the worksheet.



## 4.5 Accessing the I/O Level with XRIO

### 4.5.1 XRIO connection

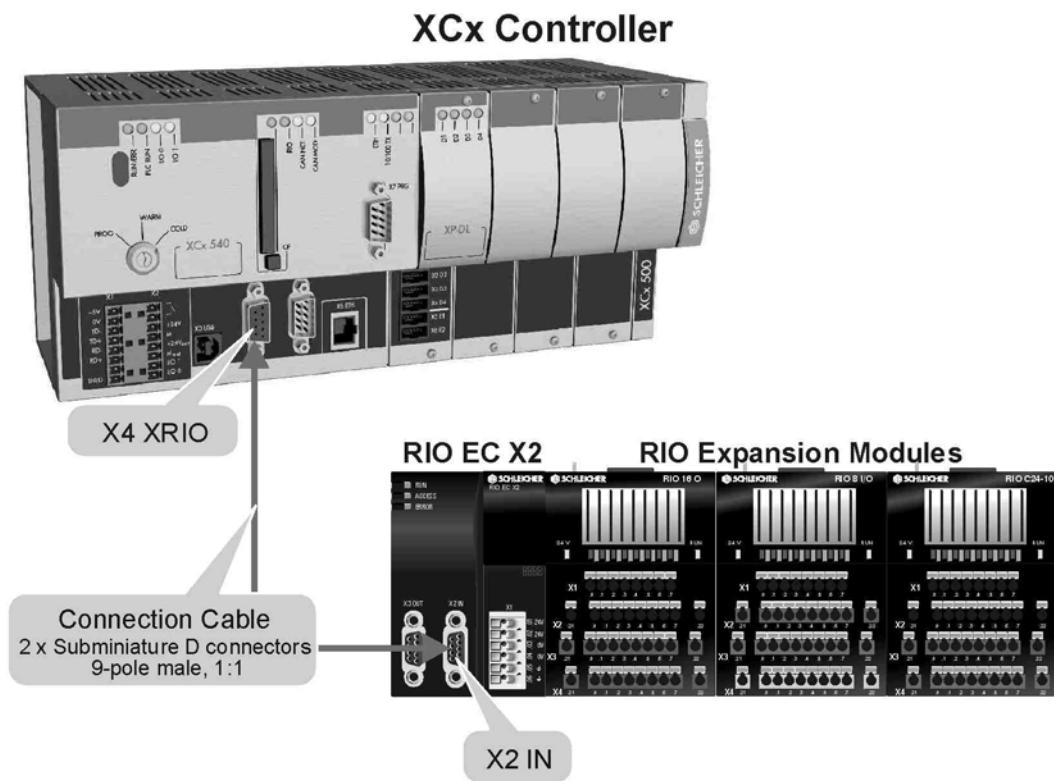
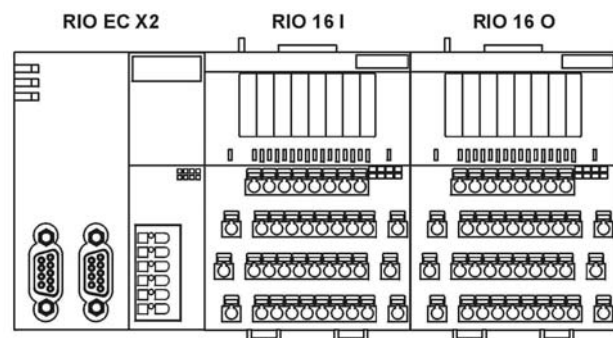


Figure 14: XRIO connection to XCx

### 4.5.2 Example of an XRIO configuration

This configuration example is used in the following sections.



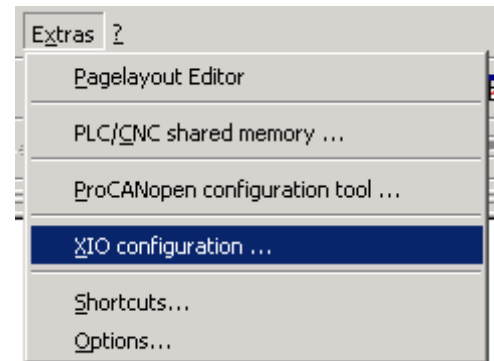
Module	RIO 16I	RIO 16O
Inputs not debounced	IB0, IB1 IX0.0...IX0.7, IX1.0...IX1.7	-
Inputs debounced	IB2, IB3 IX2.0...IX2.7, IX3.0...IX3.7	-
Outputs	-	QB0, QB1 QX0.0...QX0.7, QX1.0...QX1.7



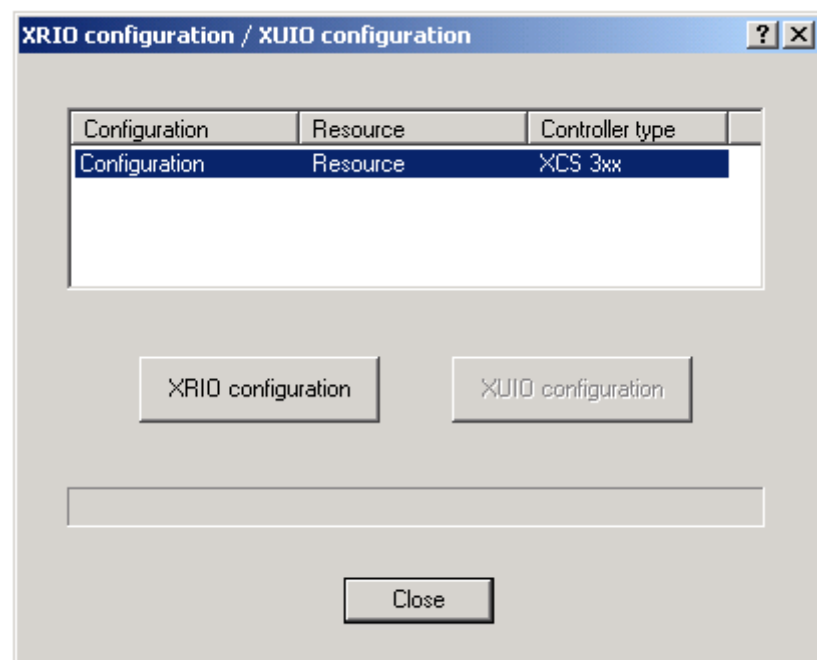
### 4.5.3 Inserting the XRIO configuration

When you install *AddOns* (see MULTIPROG installation) an XRIO configurator is provided. It completely automates XRIO driver insertion and XRIO variable declaration.

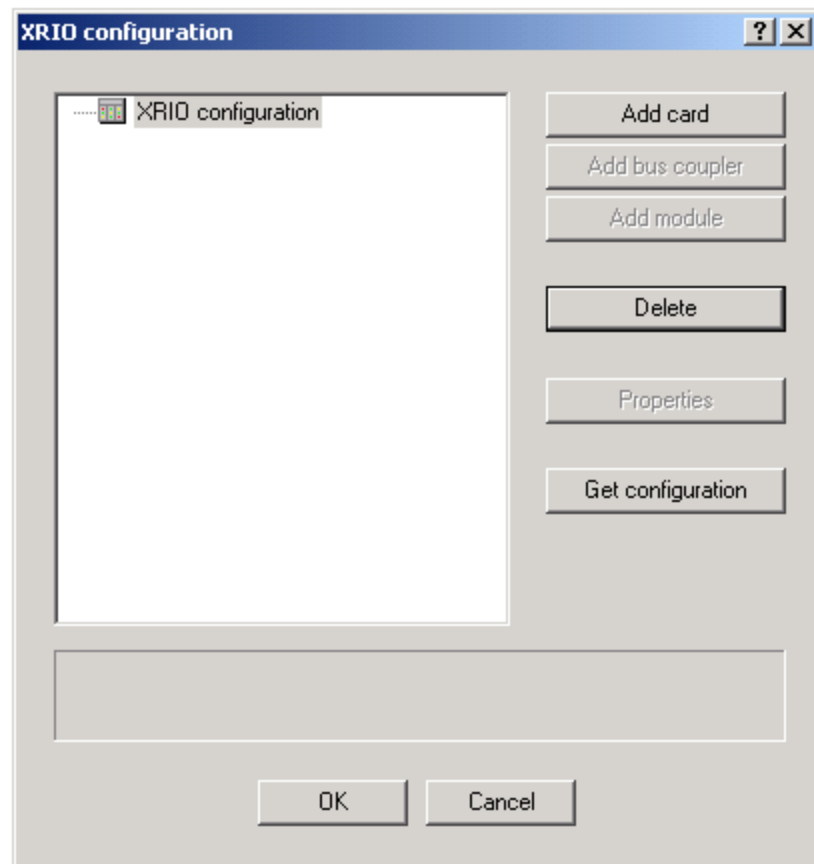
- You can start the configurator via *Extras/XRIO configuration*.



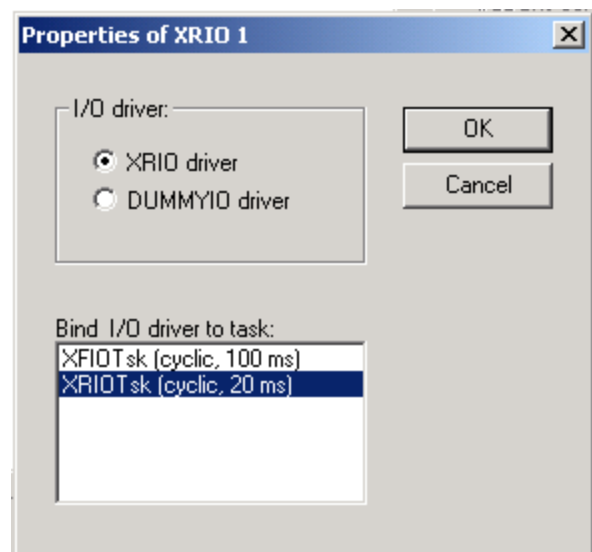
- Select XCS3xx resource and confirm by clicking XRIO configuration.



- Click on the *Get configuration* button to read the actual XRIO configuration.

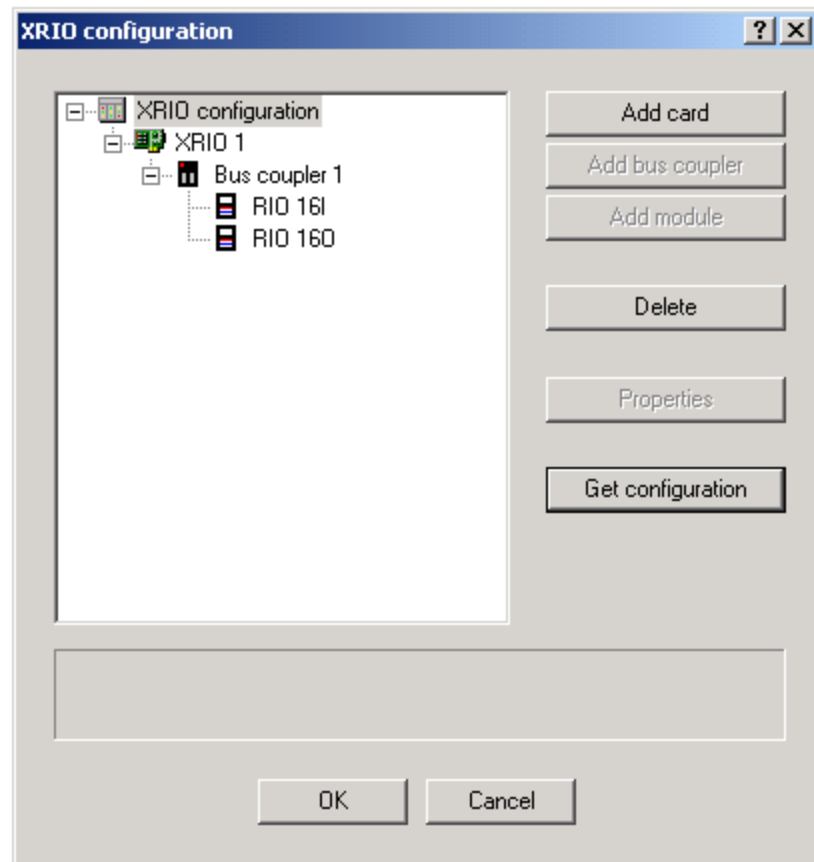


- You only have to select the task (here XRIOTsk) to which the XRIO driver is to be bound.  
If no XRIO hardware available it is possible to choose *DUMMYIO driver* to programming and testing a virtual XRIO configuration.





- Confirm the settings by clicking OK. The drivers are parameterized and the variable declaration is inserted.



- In the MULTIPROG project tree the variable declaration for the configuration described above has been entered under Global\_Variables.



Global_Variables:Configuration.Resource									
Name	Type	Address	Description	Usage	Init	Retain	PDD	OPC	
Global_Variables									
XFIO_Variables									
XRIO_Variables									
xrio1IB0	BYTE	%IB 0	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL	VAR_GLOB...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
xrio1IB1	BYTE	%IB 1	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL	VAR_GLOB...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
xrio1IB2d	BYTE	%IB 2	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL	VAR_GLOB...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
xrio1IB3d	BYTE	%IB 3	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL	VAR_GLOB...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
xrio1QB0	BYTE	%QB 0	XRIO card 1, module 2, bc 0, RIO 16O, DIGITAL	VAR_GLOB...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
xrio1QB1	BYTE	%QB 1	XRIO card 1, module 2, bc 0, RIO 16O, DIGITAL	VAR_GLOB...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Network_Variables									
PLC_Common									

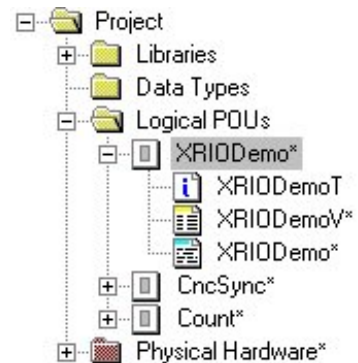
For the RIO 16 I input module the non-debounced inputs (xrio1IB0 and xrio1IB1) are declared first, followed by the debounced inputs (xrio1B2d and xrio1B3d).



#### 4.5.4 The first I/O access by the PLC program

For the demonstration of I/O access, the first 8 inputs (non-debounced) on the RIO 16 I should be assigned to the first 8 outputs on the RIO 16 O.

A POU named *XRIODemo* already exists, and an instance of the POU is inserted in the *XRIOTsk* task.

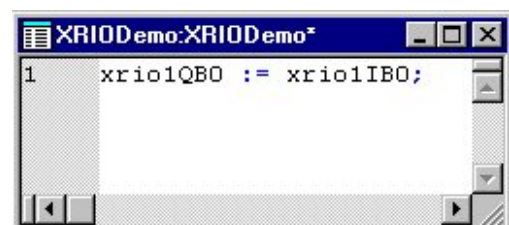


The worksheets of the POU are empty. They will accept the local I/O variable declaration and the PLC program.

- First the I/O variables have to be declared. Click on *XRIODemoV* and make the following entries.  
It is important to describe usage with `VAR_EXTERNAL`, because the I/O variables are predefined as global variables. You can also easily copy and paste (`<Ctrl+C>`, `<Ctrl+V>`) the entries from the *Global\_Variables* worksheet to this worksheet.

XRIODemoV:XRIODemo					
	Name	Type	Usage	Description	Access
Default					
	xrio1IB0	BYTE	VAR_EXTERNAL	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL	
	xrio1QB0	BYTE	VAR_EXTERNAL	XRIO card 1, module 2, bc 0, RIO 16O, DIGITAL	

- In the next step you can create the PLC program on the *XRIODemo* worksheet.



After the project has been compiled and downloaded you can start the XCx.

After startup the green RUN and ACCESS LEDs on the XRIO coupler light up. Inputs 0 to 7 on the RIO 16 I can be controlled with DC 24 V. The corresponding output on the RIO 16 O is then set.



### 4.5.5 Diagnosis

Diagnosis can be carried out with the XRIO\_STATE function block from the PLC\_Vxxx firmware library. It has to be called cyclically in a task.

Example for function block call (program in ST):

```
XRIO_STATE_1(ENABLE:=enable,CARD:=card,STATE:=state
,MODULES:=modules);
state      := XRIO_STATE_1.STATE;
modules    := XRIO_STATE_1.MODULES;
error      := XRIO_STATE_1.ERROR;
enable     := FALSE;
```

Variables:

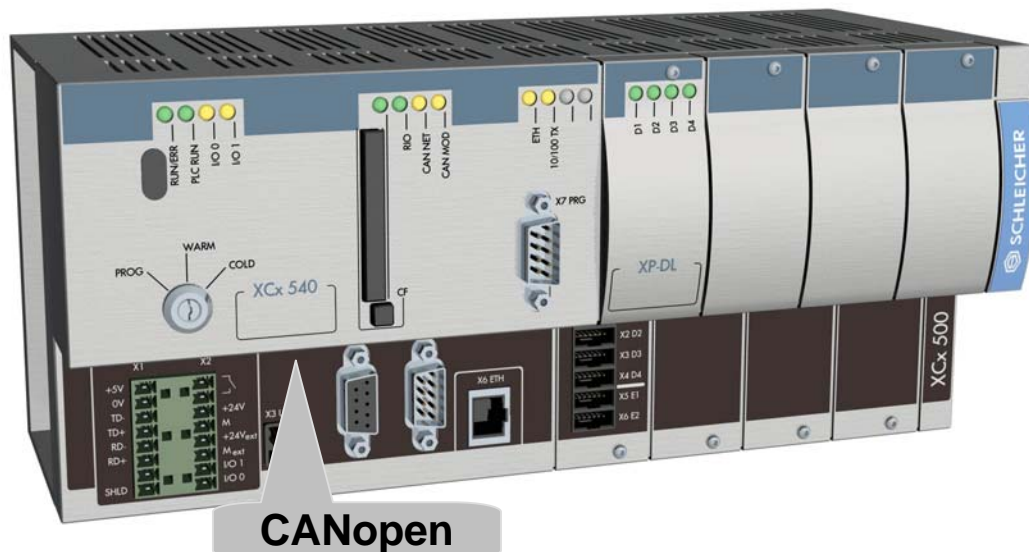
card	USINT
enable	BOOL
error	WORD
modules	XRIO_ModInfoType
state	XRIO_StateType

TYPE definition:

```
XRIO_ModuleInfo  : STRUCT
    id            : UDINT;
    EBytes        : UINT;
    ABytes        : UINT;
    modClass      : UINT;
    generall      : UINT;
END_STRUCT (* XRIO_ModInfoType *);

XRIO_ModInfoType : ARRAY [1..16] OF
XRIO_ModuleInfo;
XRIO_StateType   : STRUCT
    nofModules    : INT;
    activeErrBits : USINT;
    staticErrBits : USINT;
    overload      : UDINT;
    _24VFail      : UDINT;
    stateErrCount : UDINT;
END_STRUCT (* XRIO_StateType *);
```

## 4.6 CANopen for Remote I/O



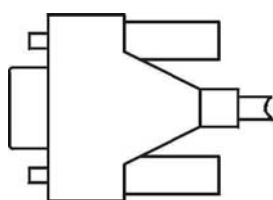
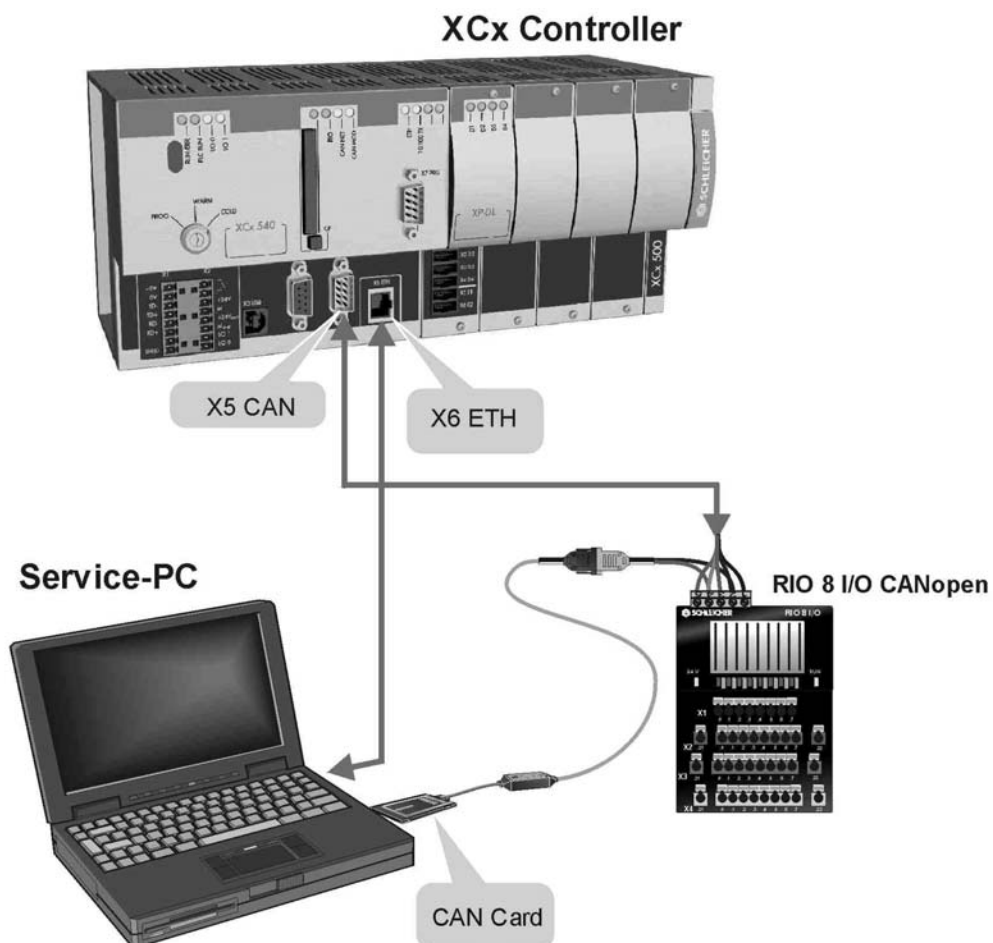
### 4.6.1 Specifications

- CANopen operates with two types of telegram:  
**SDO** (service data objects) are telegrams that have to be **confirmed** by the recipient,  
**PDO** (process data objects) are telegrams that **do not** have to be **confirmed** by the recipient.
- The PDOs for data exchange are defined during network configuration and given a COB ID. The recipient of a message always knows which telegram is addressed to which node.
- Certain components only use default mapping and work with fixed COD IDs, which are defined in the CANopen Definition.
- The standard communication for PDO is **COS** (Change Of State): A PDO is only sent if the information in the PDO changes.

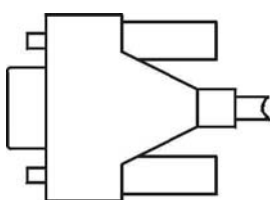


### 4.6.2 Connection and wiring

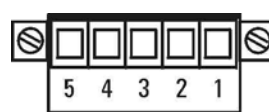
The described setup is a minimum configuration, which serves as an example for further startup steps.



Subminiature, 9-pin,  
socket connector  
to CAN card in PC



Subminiature, 9-pin,  
socket connector  
to XCx X5



Open style connector  
on RIO 8 I/O CANopen

Pin	Pin	Pin	
	3	1	0 V
2	2	2*	CAN_L
	5	3	Drain
7	7	4*	CAN_H
	9	5	+24 V DC

\* A 120 ohm terminator must be connected between pin 2 and 4 on the RIO 8 I/O CANopen.

#### 4.6.3 Settings on the RIO 8 I/O CANopen module

On the RIO 8 I/O CANopen compact module set the node number to 2 and the data transmission rate to 125 kbaud.

Set the DIP switches on top of the module as follows:

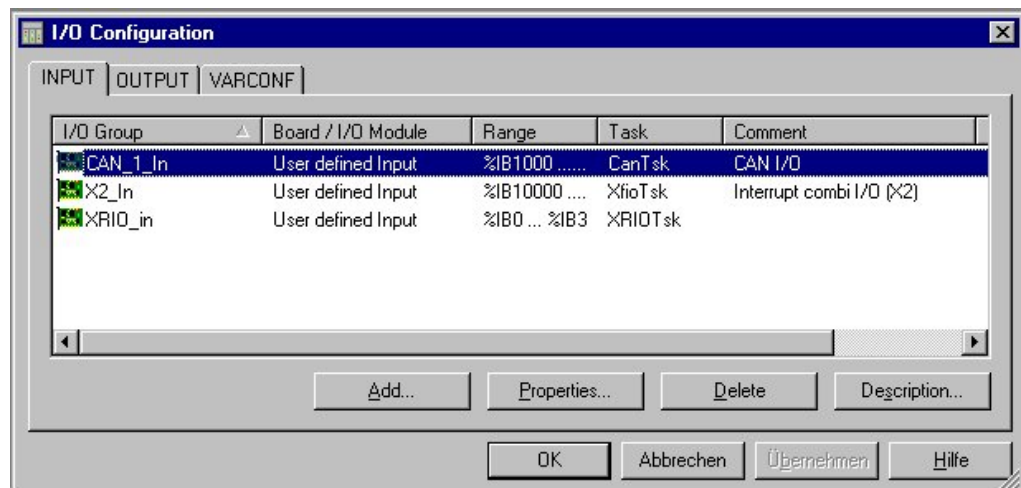
	Node number							Data transmission rate		
Switch	1	2	3	4	5	6	7	8	9	10
Position	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF



#### 4.6.4 Declaring the I/O driver for CANopen

The I/O driver for the CANopen network is declared at the same place and in the same way as the XRIO driver.

- The IO\_Configuration container is at the end of the project tree.



I/O configurations CAN\_1\_In and CAN\_1\_Out are already in place there. For this example you have to enter the following parameters:

- As *Start address* enter the logical addresses IB1000 for CAN\_1\_In and QB1000 for CAN\_1\_Out.
- The *Task* (with which the I/O address space will be synchronized) must be *CanTsk*.
- In the *Length* parameter, declare the number of I/O bytes to be exchanged in the CANopen network (here 4, because minimal double word spacing is being used).



The 'Properties' dialog box for 'CAN 1 Ir' contains the following fields and controls:

- Name:** CAN 1 Ir
- Task:** <default>
- Logical addresses:**
  - Start address:** %B 1000
  - Length:** 4
  - End address:** %B 1003
- Refresh:**
  - ☒ by task
  - ☐ manual
- Device:**
  - ☒ Driver
  - ☐ Memory
- Board / I/O Module:** User defined Input
- Comment:** CAN I/O

Buttons on the right: OK, Cancel, Description..., Driver Parameter...

- Driver name CANIO must be set in *Driver parameters*, the data type is DWORD.

The 'Driver information of standard device' dialog box contains the following fields and controls:

- Driver name:** CANIO
- Parameter 1:** 0
- Parameter 2:** 0
- Parameter 3:** 0
- Parameter 4:** 0
- Datatype:** DWORD

Buttons on the right: OK, Cancel, Description...

## 4.6.5 Declaring network variable in MULTIPROG

- The required variables are predefined in the Network\_Variables folder in *Global\_Variables* in the project tree window.

Global_Variables:Configuration.Resource - Configuration.Resource.Global_Variables							
Name	Type	Usage	Description	Address	Init	Re	
+ Global_Variables							
+ I/O_Variables							
- Network_Variables							
IB1000	USINT	VAR_GLOBAL		%IB 1000			
IB1001	USINT	VAR_GLOBAL		%IB 1001			
IB1002	USINT	VAR_GLOBAL		%IB 1002			
IB1003	USINT	VAR_GLOBAL		%IB 1003			
IW1000	UINT	VAR_GLOBAL		%IW 1000			
IW1002	UINT	VAR_GLOBAL		%IW 1002			
ID1000	UDINT	VAR_GLOBAL		%ID 1000			
QB1000	USINT	VAR_GLOBAL		%QB 1000			
QB1001	USINT	VAR_GLOBAL		%QB 1001			
QB1002	USINT	VAR_GLOBAL		%QB 1002			
QB1003	USINT	VAR_GLOBAL		%QB 1003			
QW1000	UINT	VAR_GLOBAL		%QW 1000			
QW1002	UINT	VAR_GLOBAL		%QW 1002			

- I/O bits are declared in the "I\_O\_Variables" worksheet with addresses IX1000.0.. and QX1000.0 ... (QX1000.7 is used in the example, to make the result visible on the RIO 8 I/O.)

Global_Variables:Configuration.Resource - Configuration.Resource.Global_Variables							
Name	Type	Usage	Description	Address	Init	Re	
+ Global_Variables							
- I/O_Variables							
CAN_1_Ix0	BOOL	VAR_GLOBAL	CANopen 1. N...	%IX 1000.0			
CAN_1_Qx7	BOOL	VAR_GLOBAL	CANopen 1. N...	%QX 1000.7			
XRIO_1_Ix0	BOOL	VAR_GLOBAL	XRIO Modul 1 ...	%IX 0.0			





### 4.6.6 Configuring the CANopen network with ProCANopen

- To configure a network you will need the ProCANopen software. You will also require a CAN field bus card such as "CANcardY" in the service PC.
- The properties and capabilities of the components are declared in an "eds" file (electronic data sheet). The EDS file must be copied to the Eds subdirectory of ProCANopen.
- ProCANopen maps the (mappable) objects of the node. E.g. output bytes (representing input bits from RIO) of the RIO modules are linked to XCx input bytes.
- Additional information for the field bus is also configured.  
Which node is the "NMT manager"?  
Which node is the "configuration manager"?  
"Guarding" and "guarding time".  
"Sync time" and "sync window length".
- After the network has been configured they can be saved in the network.  
Saving in the network means that the node selected as "configuration manager" (normally XCx) receives the information about how the network is to be configured via CANopen. The "configuration manager" saves the information (e.g. XCx on flash disk) and the XCx configures the network when it is switched on.
- Once the network has been configured the "NMT manager" can start the CAN network. Starting the network means status "operational" and data exchange by PDOs (process data objects).

### 4.6.7 Installing ProCANopen

- To install ProCANopen please follow the documentation supplied with the software and the CAN card.
- You have to install the drivers and the ProCANopen software in two steps.
- Copy the current EDS files for XCS and XCN from the CD to .\ProCANopen\EDS\...
- If you want dialog language English: Edit the line language=0049 in file \....\proCANopen\exe\vector.ini to =001.



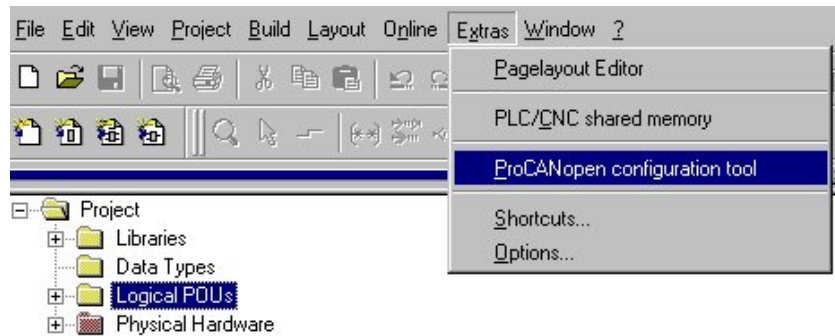
- Notes for users who have already installed ProCANopen version 2.1 and MULTIPROG 1.2 for Schleicher MicroLine and ProNumeric/ProSycon controllers:
- You only need the update version of ProCANopen.
- Do not overwrite the installed version!
- Install ProCANopen V3.2 in a new path on the hard disk, e.g. \ProCANopen3.
- If you are using a CANCardX you may need to update the firmware and the options on the card. Please note the serial number of the card and contact your local supplier.
- Depending on your PC operating system, you may need to install a number of drivers. Some new drivers are incompatible with the older ProCANopen version 2.1. This means that ProCANopen version 2.1 with the new V3.x drivers will not function online with the field bus.
- Configuration files configured with ProCANopen V2.1 can still be used with ProCANopen V3.2.



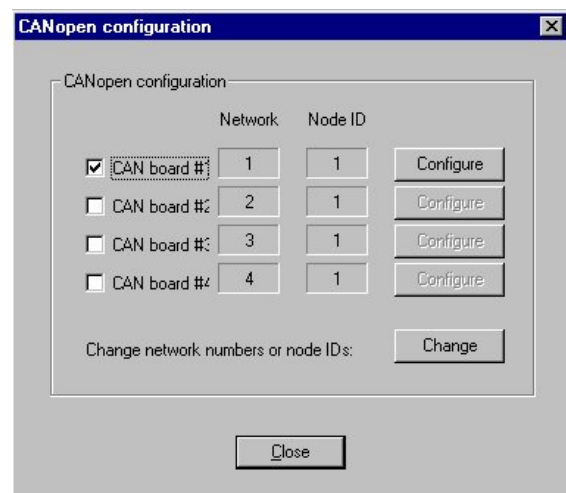
## 4.6.8 Integrating ProCANopen in MULTIPROG

Installing the Prodoc\_Plus\_Addon "AddOns.EXE" prepares MULTIPROG so that ProCANopen can be started directly from MULTIPROG.

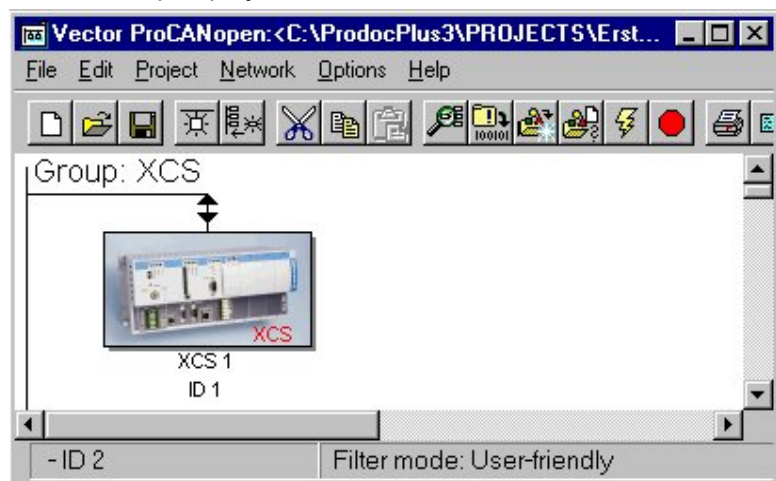
- Select *Extras/ProCANopen configuration tool*.



- Then you can select the XCx CAN card. In the example only the single standard card is used. Node number (NodeID) 1 can be retained.



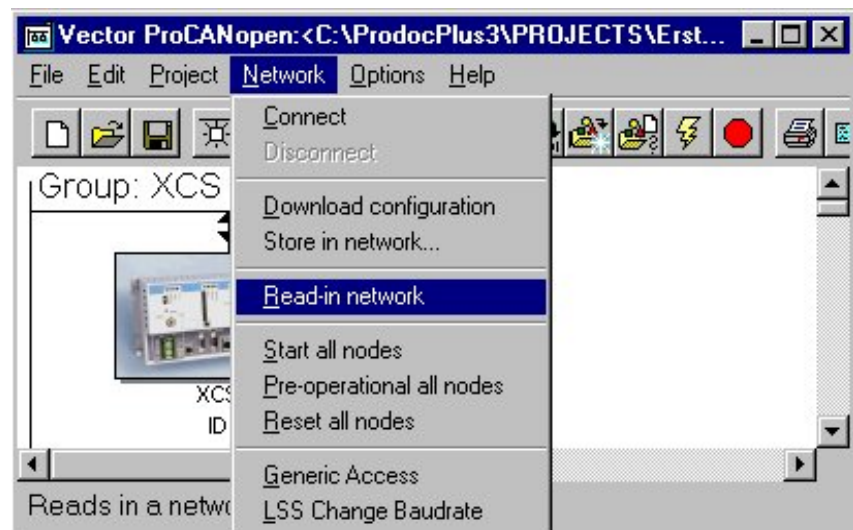
- Click on the *Configure* button. ProCANopen starts directly with the correct CANopen project.



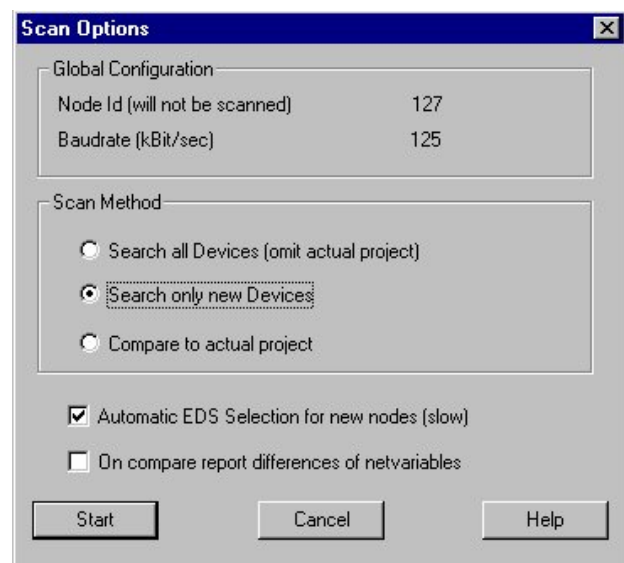


### 4.6.9 First connections with ProCANopen

First you have to read in the network.

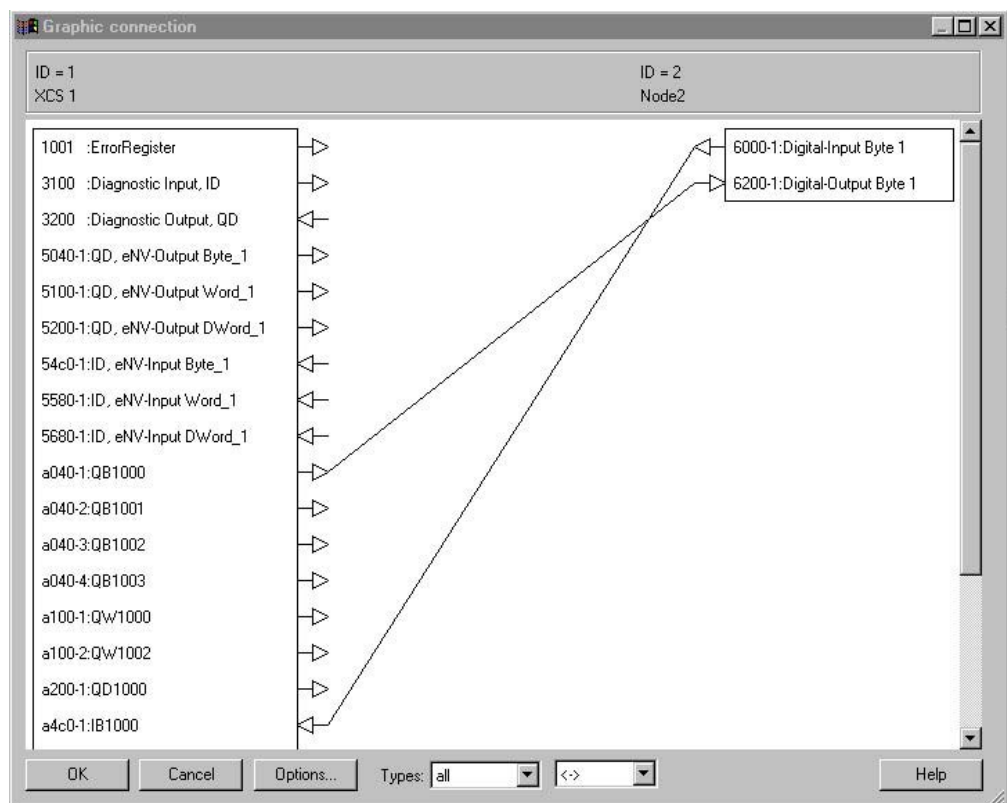
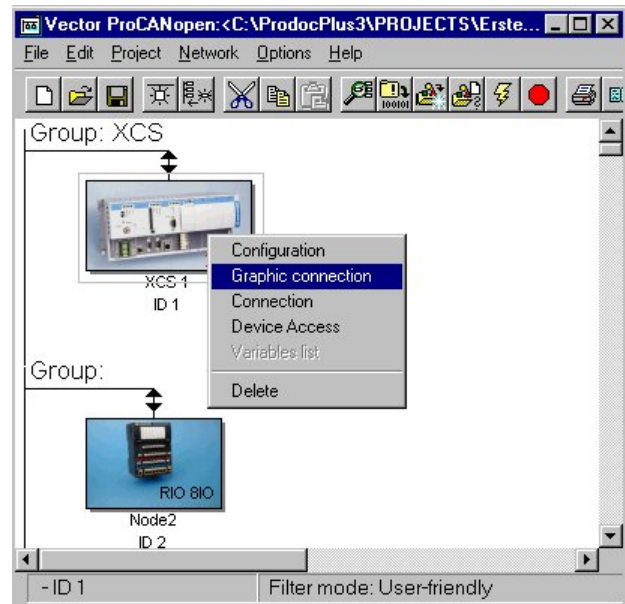


Because the network is already configured with node 1 XCx you have to read in with the *Search only new devices* scan option.



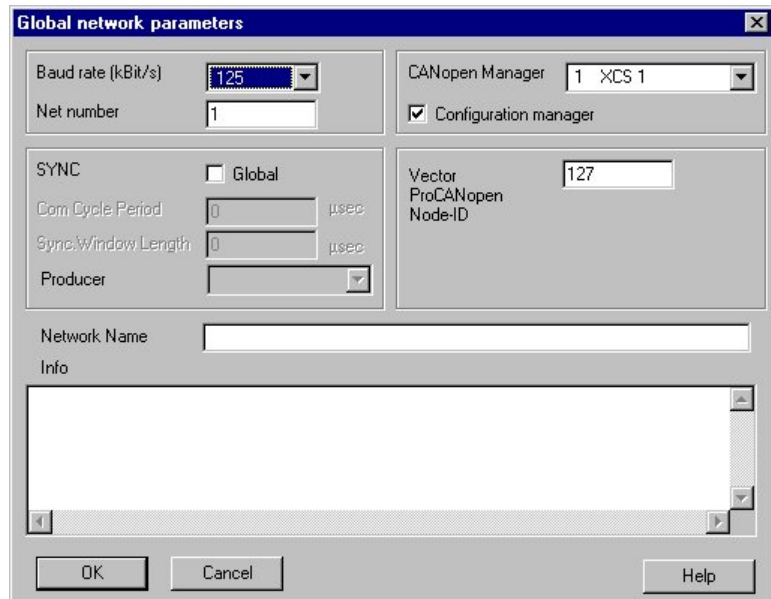
Then you can configure the network node connections.

- Click with the right mouse button on the XCx, select *Graphic connection* in the context menu of the node, then click on the node to which you wish to connect (in the example 2 RIO 8 I/O).

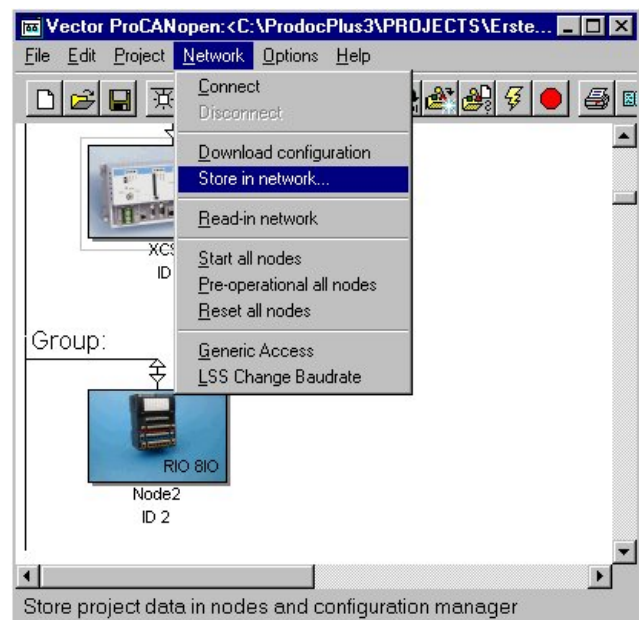




- In *Project/Global network parameters* select the CANopen manager node 1 XCx as the configuration manager.

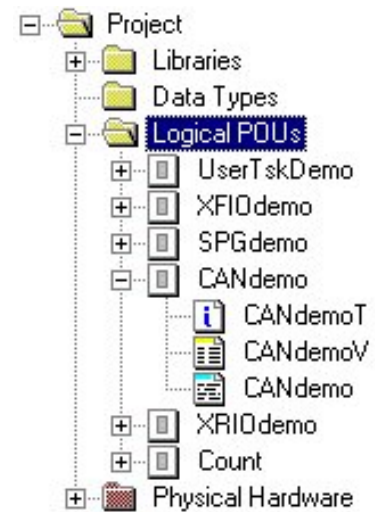


- You can save the configuration in the configuration manager using *Store in network*.  
The XCx saves the data in the compact flash. When it is switched on the network starts up.



For further information on configuring CAN please refer to the "Commissioning Field Bus Systems" manual, order no. **322 152 48**.

To test the network connection you have to create a new POU (here CANDemo) and instance it in the CanTsk.



The CANDemo POU with the associated variable worksheet:

**CANDemo: CANDemo\***

```

1  (*CAN Demo*)
2
3  CAN_1_QX7 := CAN_1_IX0;
4

```

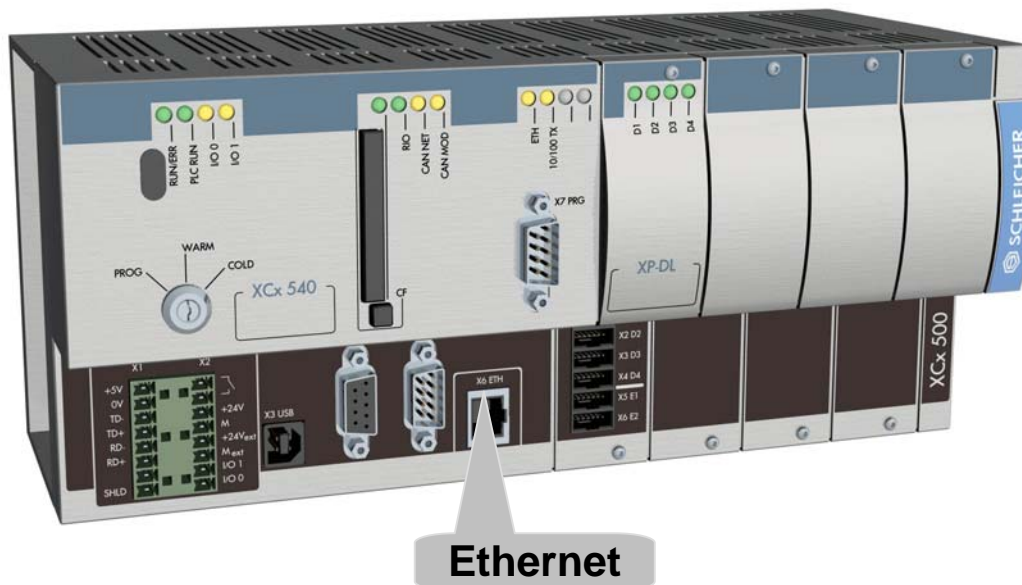
**Global\_Variables: Configuration.Resource - Configuration.Resource.Global\_Variables**

Name	Type	Usage	Description	Address	Init
<b>Global_Variables</b>					
<b>I/O_Variables</b>					
CAN_1_IX0	BOOL	VAR_GLOBAL	CANopen 1. N...	%IX 1000.0	
CAN_1_QX7	BOOL	VAR_GLOBAL	CANopen 1. N...	%QX 1000.7	
XRIO_1_IX0	BOOL	VAR_GLOBAL	XRIO Modul 1 ...	%IX 0.0	

If 24 V is connected to input 0 on the RIO 8 I/O CANopen, output 7 will be set to 1.



## 4.7 The Web-Server Functions of the XCx



### 4.7.1 General functions and concept

- The main advantage of web-server technology is that all data for operator and machine is stored in one place – the compact flash of the XCx.
- The web-server is implemented in the XCx operating system.
- Applications (web pages) are downloaded to a special area on the compact flash (with FTP or direct copying to the disk). From there the web server reads the data and sends it to the browser.
- The browser is the "thin-client" for data visualization. The application (HTML, JavaScript, Java) is loaded by the controller and runs in a shell.
- Other visualization patches have to be installed on each operator panel (Webfactory, etc.). This is known as a "fat-client" concept.

### 4.7.2 Schleicher-specific applet

Normally web technology means single-direction downloading to the browser, and the web page itself is dynamic (animation gifs or flash files). Cyclical parameter refresh is not possible.

Schleicher supplies a special Java Applet to allow bidirectional data exchange between the browser and the controller.

This applet supports functions that can be called by the HTML/ Javascript language.

These functions allow the application to write one or more PLC variable values.



#### 4.7.3 Declaring variables for visualization

Mark the variables that are to be visualized with the PDD checkbox in MULTIPROG (PDD = Process Data Directory).



Name	Type	Usage	Descri...	Address	Init	Retain	PDD	OPC
<b>Global Variables</b>								
PLCMODE_ON	BOOL	VAR_GLOBAL		%MX 1.0.0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCMODE_RUN	BOOL	VAR_GLOBAL		%MX 1.0.1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCMODE_STOP	BOOL	VAR_GLOBAL		%MX 1.0.2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCMODE_HALT	BOOL	VAR_GLOBAL		%MX 1.0.3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCDEBUC_BRESET	BOOL	VAR_GLOBAL		%MX 1.1.1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

These variables will be updated in an internal list in the XCx. The web server can read and write the variables in this list.

#### 4.7.4 Application example

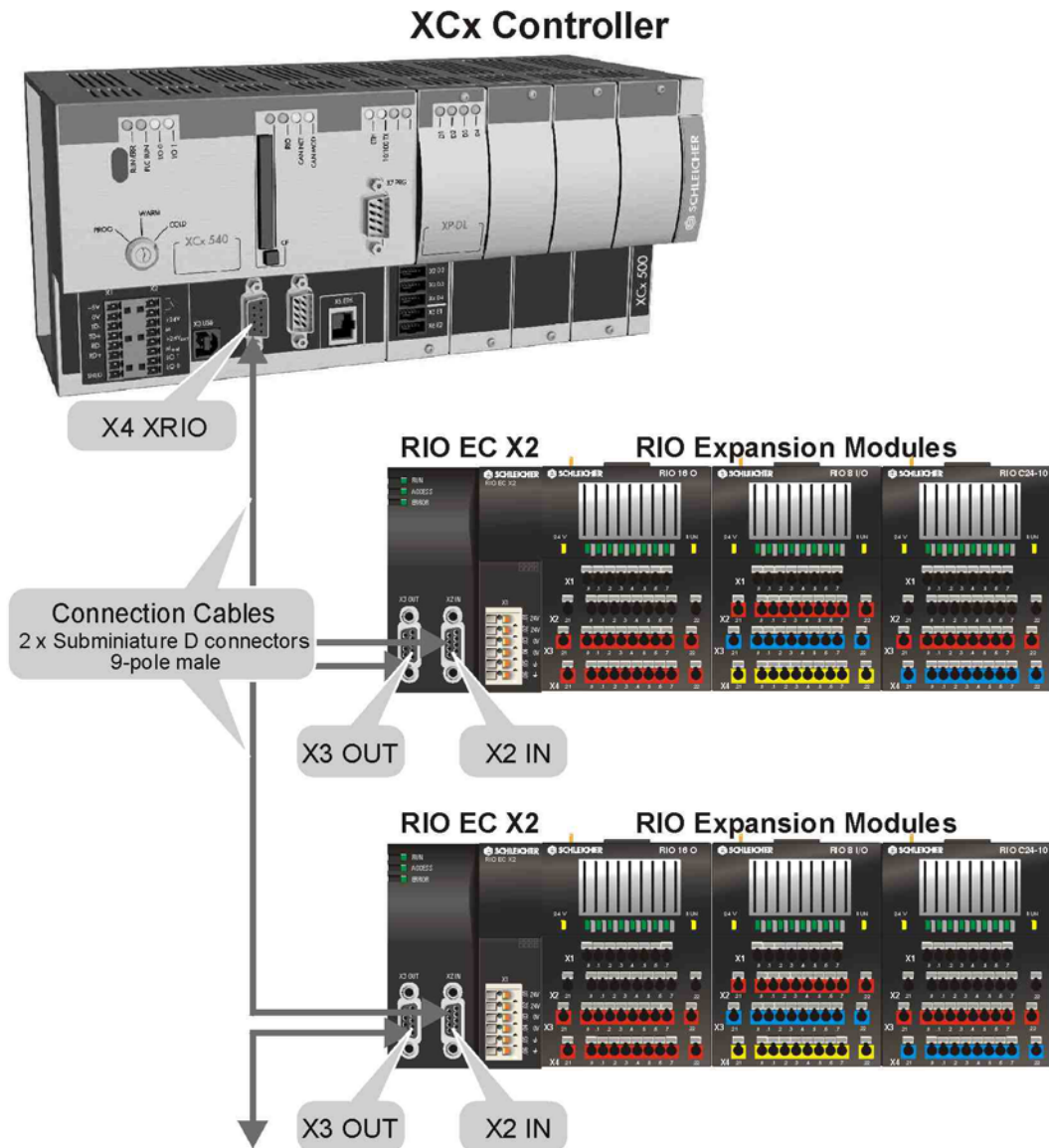
The XCx is supplied with a standard browser application.  
This application allows you to read and write PDD-marked variables.  
A status overview is also provided.

#### 4.7.5 Browser / components

- You can use any standard PC with Ethernet.
- You will need MS Explorer V 5 or higher or Netscape Navigator V 6 or higher.
- Certain terminals with Windows®-CE can be used, if the browser meets the requirements for Java Script 1.5, Java 2 , HTTP1.1.

## 5 XRIO

XRIO realizes the direct I/O level on the XCx. The RIO EC X2 bus coupler and the expansion modules from the modular RIO system are used (see also section Expansion Modules from the RIO System).



- The bus coupler X2 IN is connected directly to X4 on the XCx using a 9-pin subminiature-D connector cable.
- Max. 4 bus couplers RIO EC X2 can be connected to the XCx.
- The max cable length is 10 meters between two connection points. Use shielded cables.
- The RUN, ACCESS and ERROR LEDs indicate when the XCx operating system is accessing the bus coupler. So the RUN LED is red if the bus coupler is operating without XCx or the PLC is in stop or startup status. During normal operation with the XCx RUN and ACCESS are green.



- The I/O modules are connected on the right-hand end of the bus coupler, as with all Schleicher bus couplers (see also section Expansion Modules from the RIO System).

Furthermore information see also operating manual RIO Bus Couplers part no. 322 157 00.

Download all operating manuals free of charge from our website [www.schleicher-electronic.com](http://www.schleicher-electronic.com)

## 5.1 Recognition of XRIO Configuration by XCx Operating System

The XCx operating system automatically recognizes the XRIO configuration. If the current configuration differs from the previously saved configuration automatic PLC start will not be carried out.

- The XRIO configuration is saved on the compact flash:  
**when the XCx is first switched on** in the `lata0\OS\PLC\sysconf.txt` configuration file and in the system information `lata0\OS\logfiles\sysinfo.txt`  
**each subsequent time XCx is switched on** (as well as after reset via the MULTIPROG project control dialog and when a new project is created in MULTIPROG) only in system information `lata0\OS\logfiles\sysinfo.txt` (see The sysinfo.txt file).



The information on the current XRIO configuration from the `lata0\OS\logfiles\sysinfo.txt` file is important for the I/O driver declaration in the PLC project.

- If a difference between the configuration file and the actual XRIO configuration is found when the XCx start up, no automatic start will be carried out (XCx lockswitch set to WARM or COLD). Manual start via the MULTIPROG project control dialog is, however, possible.



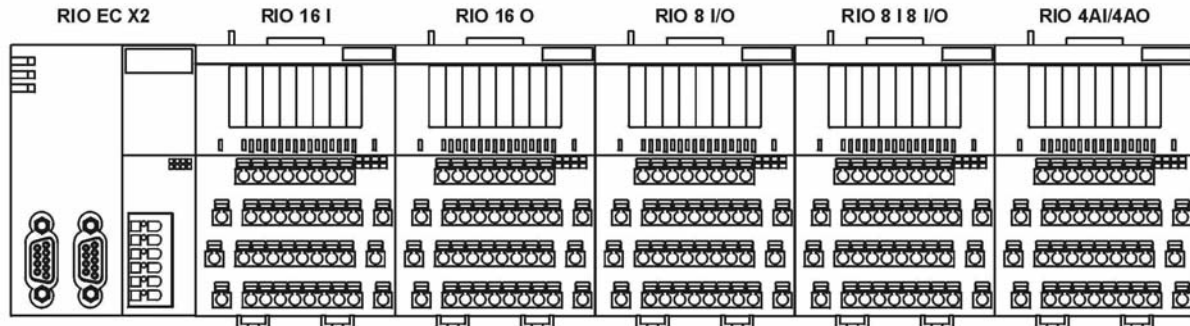
Manual start of XCx with incorrect XRIO configuration can cause unpredictable states in the I/O periphery. Precautions must be taken to prevent harm to people and equipment.

- The XCx configuration file can be updated if required:  
Enter value -1 in shared RAM location `cmpSwrdPlcRw_IWrtSysConf`. (After successful update value 100 will appear there.)  
Alternatively you can also delete the configuration file.

Then switch the XCx off and on again, or set the lock switch to PROG and initiate RESET on the XCx.

### 5.1.1 The sysinfo.txt file XRIO section

The *sysinfo.txt* file described here contains a fictitious XRIO configuration created solely for demonstration purposes. Comments have been added to the file to increase the information provided.



```
[XRIO]
Cards =1                ; One XRIO connection
Mod01 =5                ; Number of modules
#IB01 =18               ; Length of input address range in bytes
%IBS01=0                ; Start address
%IBE01=17               ; End address
#QB01 =14               ; Length of output address range in bytes
%QBS01=0                ; Start address
%QBE01=13               ; End address

[XRIO_C01M01]
Name=RIO 16I            ; Module name
ID =6                   ; Module-ID
Type=DIGITAL            ; Module class
#IB =4                  ; 4 input bytes: 2 non-debounced, 2 debounced
#QB =0                  ; 0 output bytes
%IB =0                  ; first input byte of module, byte address 0
                        ; %IB 0 not debounced
                        ; %IB 1 not debounced
                        ; %IB 2 debounced
                        ; %IB 3 debounced

[XRIO_C01M02]
Name=RIO 16O            ; Module name
ID =4                   ; Module-ID
Type=DIGITAL            ; Module class
#IB =0                  ; 0 input bytes
#QB =2                  ; 2 output bytes
%QB =0                  ; first output byte of module, byte address 0
                        ; %QB 0
                        ; %QB 1

[XRIO_C01M03]
Name=RIO 8I/O           ; Module name
ID =2                   ; Module-ID
Type=DIGITAL            ; Module class
#IB =2                  ; 2 input bytes
#QB =2                  ; 2 output bytes
%IB =4                  ; first input byte of module, byte address 4
                        ; %IB 4 not debounced
                        ; %IB 5 debounced
%QB =2                  ; first output byte of module, byte address 2
                        ; %QB 2 (dummy)
                        ; %QB 3
```



```

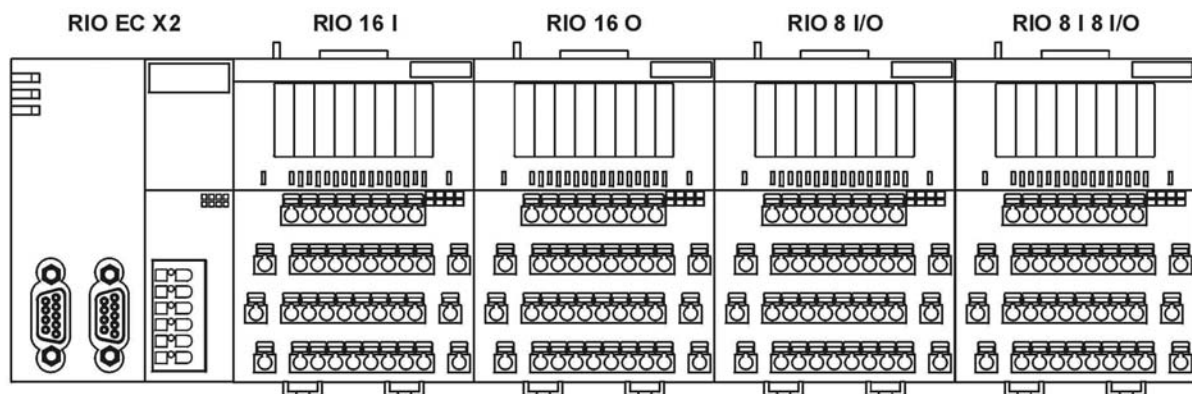
[XRIO_C01M04]
Name=RIO 8I 8I/O      ; Module name
ID  =8                ; Module-ID
Type=DIGITAL          ; Module class
#IB  =4                ; 2 input bytes
#QB  =2                ; 2 input bytes
%IB  =6                ; first input byte of module, byte address 6
                        ; %IB 6 not debounced
                        ; %IB 7 not debounced
                        ; %IB 8 debounced
                        ; %IB 9 debounced
%QB  =4                ; first output byte of module, byte address 4
                        ; %QB 4 (dummy!)
                        ; %QB 5

[XRIO_C01M05]
Name=RIO 4AI/4AO 20mA ; Module name
ID  =14               ; Module-ID
Type=ANALOG           ; Module class
#IW  =4                ; 4 input words
#QW  =4                ; 4 output words
%IW  =10               ; first input word of module, address 10
                        ; %IW 10
                        ; %IW 12
                        ; %IW 14
                        ; %IW 16
%QW  =6                ; %QW 6
                        ; %QW 8
                        ; %QW 10
                        ; %QW 12

```

### 5.2 Addressing I/O Modules

- In input addressing the debounced and non-debounced inputs are always directly after one another.  
Input debouncing at 2 ms input signal delay is realized by software in the RIO bus coupler. (The inputs on the I/O modules are additionally debounced with 01,ms).
- In output addressing, even modules with just 1 byte outputs always occupy 1 word.  
This ensures an even start address for the following modules.



Module	RIO 16I	RIO 8I/8I/O	RIO 8I/O	RIO 16O
Inputs not debounced	IB0, IB1	IB4, IB5	IB8	-
Inputs debounced	IB2, IB3	IB6, IB7	IB9	-
Outputs	-	QB1(+QB0)	QB3(+QB2)	QB4, QB5

### 5.3 XRIO Flags in Shared Memory Validity of Process Data

Use the shared memory flags

*plcMem.plcSect.flgXRIO.bXRio[]*      *ARRAY[1..4] OF BOOL*

or

*cmpSflgXRIO\_bXRio[]*      *ARRAY[1..4] OF BOOL*

to check the validity of XRIO process data

TRUE = Input image valid (PLC mode RUN, no communication error)

FALSE = Input image invalide (PLC mode STOP, not ready or communication error)

## 5.4 The XRIO Driver

The XRIO I/O driver interfaces connected RIO modules via the X4 XRIO connector, provides corresponding input and output maps, and refreshes them.

Attention!

The XRIO driver must be called exactly once per "XRIO connection". The I/O range of an XRIO connection must not be shared among different tasks as this would lead to malfunctions.

In order to obtain consistent data over the task runtime the XRIO driver has to be called in the task in which the I/Os are processed. (The input map is read at the start of the task, and the output map is written at the end of the task.)

The task interval can be between 1 and 80 ms. The watchdog for the RIO modules triggers if task interval > 80 ms.

The following table provides an overview of the user parameters currently supported by the XRIO driver.

XRIO user parameter:		
DRIVER_PARAMETER	Code	Content
DRIVER_PAR1	0 = default	<ul style="list-style-type: none"> <li>Process data (outputs) zeroed automatically on PLC-STOP</li> <li>One I/O refresh carried out per task cycle</li> </ul>
DRIVER_PAR1	Bit 0	<ul style="list-style-type: none"> <li>Not used</li> </ul>
DRIVER_PAR1	Bit 1	<ul style="list-style-type: none"> <li>Not used</li> </ul>
DRIVER_PAR1	Bit 2	<ul style="list-style-type: none"> <li>Ignore differences between configured and real number of I/O bytes.</li> </ul>
DRIVER_PAR1	Bit 3	<ul style="list-style-type: none"> <li>Additional process data refresh (configure the IN driver section)</li> </ul> <p><b>Note:</b> Function is not released at present.</p>
DRIVER_PAR1	Bit 6	<ul style="list-style-type: none"> <li>2 stop bits if set (configure the IN driver section)</li> </ul>
DRIVER_PAR1	Bit 7	<ul style="list-style-type: none"> <li>Force the XRIO normal mode (500 kBit/s) (configure the IN driver section)</li> </ul>
DRIVER_PAR2	Bit 0	<ul style="list-style-type: none"> <li>Not used</li> </ul>
DRIVER_PAR2	Bit 1:	<ul style="list-style-type: none"> <li>Prevents task change during process data update</li> </ul>
DRIVER_PAR2	Bit 2	<ul style="list-style-type: none"> <li>Not used</li> </ul>

## 5.5 Calculating Estimated Transmission Time

The transmission times on the XRIO bus are largely determined by the number of bytes to be sent.

4 Mbaud and 1 stop bit:

$((\sum \text{I/O-bytes je module} + 4) \times 2,75 \mu\text{s/byte} + (\text{no. of ECs}) \times 4 \mu\text{s/EC} + (\text{no. of IO modules}) \times 1 \mu\text{s/module} + 10 \mu\text{s})$

4 Mbaud and 2 stop bits

$((\sum \text{I/O-bytes pro module} + 4) \times 3,00 \mu\text{s/byte} + (\text{no. of ECs}) \times 4 \mu\text{s/EC} + (\text{no. of IO modules}) \times 1 \mu\text{s/module} + 10 \mu\text{s})$

500 kbaud and 1 stop bit

$((\sum \text{I/O-bytes pro module} + 3) \times 22 \mu\text{s/byte})$

500 kbaud and 2 stop bits

$((\sum \text{I/O-bytes pro module} + 3) \times 24 \mu\text{s/byte})$



Exampel (ECX2, 6 dig.IO, 2 P24, 1 A10):

4 Mbaud and 1stop bit

$$(61 + 4) \times 2,75 \mu\text{s} + 4 \times 4 \mu\text{s} + 9 \times 1\mu\text{s} + 10 \mu\text{s} = 0,214 \text{ ms} \rightarrow \text{measured } 0,20 \text{ ms}$$

4 Mbaud and 2 stop bits

$$(61 + 4) \times 3,00 \mu\text{s} + 4 \times 4 \mu\text{s} + 9 \times 1\mu\text{s} + 10 \mu\text{s} = 0,230 \text{ ms} \rightarrow \text{measured } 0,21 \text{ ms}$$

500 kbaud and 1stop bit

$$(61 + 3) \times 22,00 \mu\text{s} = 1,410 \text{ ms} \rightarrow \text{measured } 1,39 \text{ ms}$$

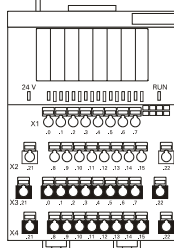
500 kbaud and 2 stop bits

$$(61 + 3) \times 24,00 \mu\text{s} = 1,540 \text{ ms} \rightarrow \text{measured } 1,51 \text{ ms}$$

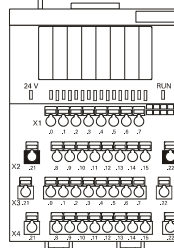
## 5.6 Expansion Modules from the RIO System

### 5.6.1 Overview

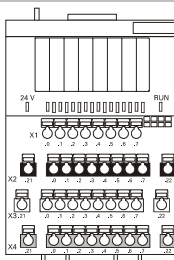
#### Digitalmodules



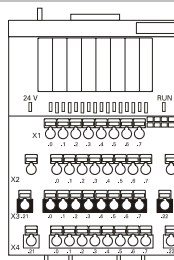
**RIO 16 I**  
16 inputs DC 24 V  
Two-wire connection system



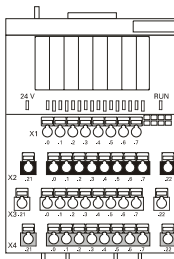
**RIO 16 O**  
16 outputs 1 A  
Two-wire connection system



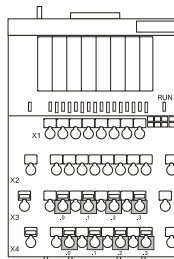
**RIO 8 I/O**  
8 combination I/Os  
All combination I/Os can be used individually as inputs DC 24 V or outputs 1 A.



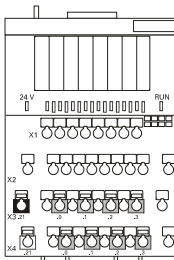
**RIO 8 I 8 I/O**  
8 inputs DC 24 V  
8 combination I/Os  
All combination I/Os can be used individually as inputs DC 24 V or outputs 1 A.  
Two-wire connection system



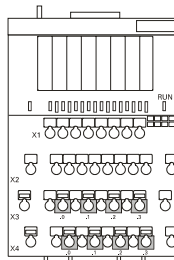
**RIO 8 O 2A**  
8 outputs 2 A  
Four-wire connection system



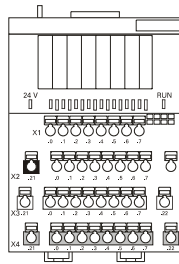
**RIO 4 I 230 VAC**  
4 inputs AC 230 V



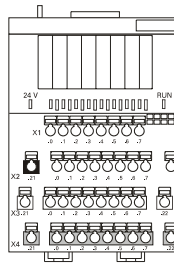
**RIO 4 O R**  
4 outputs relay



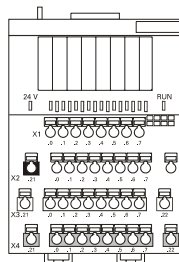
**RIO 4 I 120 VAC**  
4 inputs AC 120 V

**Analog modules**
**Voltage  $\pm 10V$** 


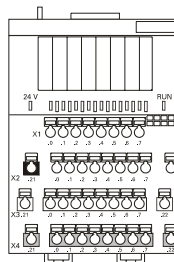
**RIO 4AI  $\pm 10V$**   
4 inputs analog  
Resolution 12 bits

**Current 20mA**


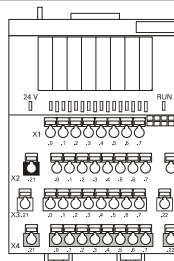
**RIO 4AI 20mA**  
4 inputs analog  
Resolution 12 bits



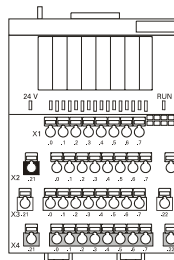
**RIO 4AI/4AO  $\pm 10V$**   
4 inputs analog  
4 outputs analog  
Resolution 12 bits



**RIO 4AI/4AO 20mA**  
4 inputs analog  
4 outputs analog  
Resolution 12 bits

**Current 4...20mA**


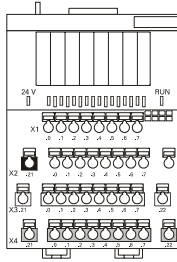
**RIO 4AI 4-20mA**  
4 inputs analog  
Resolution 12 bits



**RIO 4AI/4AO 4-20mA**  
4 inputs analog  
4 outputs analog  
Resolution 12 bits

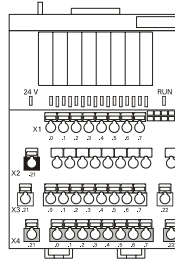


### Temperature modules



#### RIO T10-10

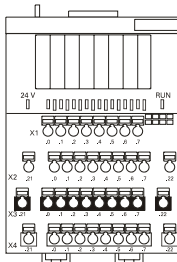
4 inputs for measuring the temperature with Pt100/Pt1000



#### RIO T20-10

4 inputs for measuring the temperature with thermo elements

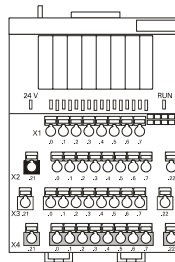
### Counter module



#### RIO C24-10

4 counters 16-bit or 2 counters 32-bit

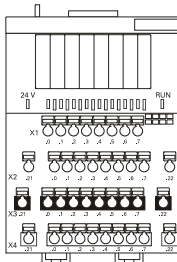
### Axis interface



#### RIO A10-10

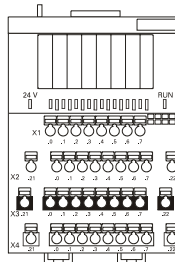
Interface of one Axis

### Positioning module



#### RIO P05-10

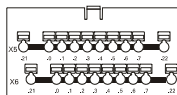
Positioning of two axes (5 V counter inputs)



#### RIO P24-10

Positioning of two axes (24 V counter inputs)

### Potential distributor (terminal expansion)



#### RIO KE 16

2 distributors each with 10 terminal connections

Only suitable for modules with clips.

Ordering information see [www.schleicher-electronic.com](http://www.schleicher-electronic.com)

For further details please refer to the "RIO Expansion Modules" Operating Manual, Part no. 322 154 15 see [www.schleicher-electronic.com](http://www.schleicher-electronic.com)



## 5.6.2 Tables of module types and module classes

### Modul types

Module- ID	HID	Name	Module Type	Input Bytes	Output Bytes	I/O- Bytes effective
0x0001	0x02	RIO 8I/O	DIGITAL	1+1	2	3
0x0002	0x06	RIO 16I	DIGITAL	2+2	0	3
0x0003	0x04	RIO 16O	DIGITAL	0	2	3
0x0004	0x08	RIO 8I 8I/O	DIGITAL	2+2	2	3
0x0005	0x0A	RIO 4AI/4AO $\pm 10V$	ANALOG	8	8	9
0x0006	0x0C	RIO 4AI $\pm 10V$	ANALOG	8	0	9
0x0007	0x0E	RIO 4AI/4AO 20mA	ANALOG	8	8	9
0x0008	0x10	RIO 4AI 20mA	ANALOG	8	0	9
0x0009	0x12	RIO RNO	DIGITAL	2+2	0	3
0x000A	0x42	RIO C24-10	COUNTER	10+2	10	11
0x000B	0x42	RIO C24-10	COUNTER	10+2	10	11
0x000C	0x62	RIO P24-10	COUNTER	10+2	10	11
0x000D	0x62	RIO P24-10	COUNTER	10+2	10	11
0x000E	0x16	RIO T10-10	ANALOG	8	0	9
0x000F	0x1E	RIO 4AI 0..10V	ANALOG	8	0	9
0x0010	0x18	RIO 4AI/4AO 4..20	ANALOG	8	8	9
0x0011	0x1A	RIO 4AI 4..20mA	ANALOG	8	0	9
0x0012	0x1C	RIO 4AI/4AO 0..10	ANALOG	8	8	9
0x0013	0x24	RIO 4OR	DIGITAL	0	2	3
0x0014	0x20	RIO T20-10	ANALOG	8	0	9
0x0015	0x00	RIO P05-10	COUNTER	10+2	10	11
0x0016	0x00	RIO P05-10	COUNTER	10+2	10	11
0x0017	0x44	RIO 8O 2Amp.	DIGITAL	0	2	3
0x0018	0x26	RIO 4I 115VAC	DIGITAL	2	0	3
0x0019	0x46	RIO 4I 230VAC	DIGITAL	2	0	3
0x001A	0x00	RIO 8I TTL	DIGITAL	2	0	3
0x001B	0x00	RIO 8O neg	DIGITAL	0	2	3
0x001C	0x70	RIO A 10-10	ANALOG	8+4	8	9
0x001C	0x70	ECX2	ANALOG	2	2	3

Input Bytes are the sum of all debounced and not debounced I/Os.

I/O-Bytes are all bytes they must be moved respectively transmitted.

### Module classes

Module type	Code
ANALOG	0
DIGITAL	1
COUNTER	2
NOMOD	3

## 5.7 Expansion Module XRIO

The module is used to expand the XCx 300 and the XCx 540. The module is equipped with following features:

- One XRIO bus interface
- Correspond to RS422
- 500 kBaud data transmission rate

### 5.7.1 Displays and Connections

Displays			
	Colour		
L1			not used
L2	green	on	Operational
	green	flashing	Preoperational
	red	on	Bus error
	yellow	on	Frame error
L3			not used
L4			not used
Connector X31			
Pin	Designation		
1	RD+	Received data plus	
2	TD+	Transmitted data plus	
3	0 V	Ground power supply	
4	nc	not connected	
5	+5 V	Power supply	
6	RD-	Received data minus	
7	TD-	Transmitted data minus	
8	nc	not connected	
9	nc	not connected	



### 5.7.2 Technical data

Basic data	
Interface	XRIO
Data transmission rate	500 kBaud
Connection system	Subminiature type D, 9-pin, socket connector



## **6 CANopen Field Bus**

### **6.1 Fundamentals**

#### **General**

CANopen is based on the CAN Application Layer for industrial CAL applications. The CANopen communications profile CiA DS-301 specifies the mechanisms for configuring and communication between devices in real-time environments.

CANopen uses the data transmission layer to ISO 11898 and CAN 2.0 A+B.

- Description of device details via an EDS (Electronic Data Sheet)
- Object-oriented communication with PDOs and SDOs
- Transmission of real-time data with 'pure' CAN as the PDO (Process Data Object) PDOs can be sent by all slaves, either event-driven or synchronized
- Complex data or low-priority services are transmitted/processed with SDOs (Service Data Objects)
- CANopen configuration managers (or masters) carry out functions such as network management during network start-up. However, they are not necessary for communication between the slaves.
- In theory up to 127 stations are possible on one bus.  
In practical terms, however, the number of bus stations is restricted by the respective bus topology, and in particular by the type of CAN transceiver modules used.  
(current limit on Schleicher CAN devices is 64)

#### **CNopen with XCx**

The XCx PLC can access network variables and parameterizing and diagnosis functions via the CANIO driver. Driver installation and network startup are described in the startup section of the XCx manual.

You can also access direct network variables, see Accessing direct network variables.

The parameterizing and diagnosis functions relate to the hardware and software of the XCx CANopen interface, which is described here as the CAN processor. See "Accessing parameterizing and diagnosis functions" and "Parameterizing and Diagnosis Functions".

### 6.1.1 Process Data Objects (PDO)

The control unit can use byte, word and double word network variables as mappable objects in PDOs.

The PLC program accesses network variables with the help of the CANIO driver via a process data map.

Direct network variables, which can be accessed without a special driver and without process map (!), are also provided.

We recommend using a CANopen configurator such as ProCANopen to configure the communication relationships in the CANopen network. All information on number, object index, data type, etc. can be taken from the supplied EDS files.

As a device conforming to CiA DSP 302 (Framework for Programmable CANopen Devices) / DSP 405 (Device Profile for IEC 1131 Programmable Devices), the XCx does not provide default mapping for PDOs.

You can find a list of mappable objects in Parameterizing and diagnosis data and Network variables.

### 6.1.2 Service Data Objects (SDO)

Special "SDO objects" are available for reading/writing up to 128 bytes of consistent user data.

If the PLC is to access these objects, corresponding arrays must first be declared in the PLC code.

Object index	Direction	Address range	Number of bytes
0x4100	Rx	Depending on IO configuration	128
0x4110	Rx		128
0x4120	Rx		32
0x4130	Rx		64
0x4200	Tx		128
0x4210	Tx		128
0x4220	Tx		32
0x4230	Tx		64

Rx means externally writable by "Write Request"

Tx means externally readable by "Read Request"

### 6.1.3 Nodeguarding

Node guarding allows a guarding master to detect a failed slave. The guarding process involves sending messages to the guarding ID (100Eh) of the slave in periodic cycles. The slave replies with a guarding message which may include a toggle bit.

If a Schleicher PLC is configured as the guarding master, a function block can be used to evaluate guarding errors in the PLC program.



### 6.1.4 Lifeguarding

While nodeguarding is used by the guarding master to detect a failed station, the monitored station uses these guarding telegrams to detect a failed master. This monitoring function of the station is called lifeguarding.

To detect a broken cable and enforce a shut-down of the outputs with CANopen, the nodeguarding and lifeguarding functions must be activated.

To activate lifeguarding the NMT manager must describe the Guard Time (100Ch) and the Life Time Factor (100Dh) objects.

If the monitoring time

$$\text{Life-Time} = \text{Life-Time-Factor} * \text{Guard-Time [ms]}$$

expires without a guarding telegram arriving at the station, a guarding error has occurred.

- A RIO CANopen bus coupler or compact module switches the NET LED to red flashing and shuts the outputs down.
- A Schleicher PLC with CANopen signals the guarding error to the PLC program using a function block.

If one of the above-mentioned objects is 0, no lifeguarding and no cable break detection is carried out.

## 6.2 CANopen-Specific PLC Addresses

### 6.2.1 Parameterizing and diagnosis data

Diagnosis data		
CANopen Object Index	PLC address	Content
0x3100 Diagnostic Input	Depending on IO configuration	Input data
0x3200 Diagnostic Output	Depending on IO configuration	Output data

### 6.2.2 Network variables

Network variables		
CANopen Object Index	PLC address	Content
0x54c0 IB Input Byte 0x5580 IW Input Word 0x5680 ID Input Dword	Depending on IO configuration	Direct NVs input data
0x5040 QB Output Byte 0x5100 QW Output Word 0x5200 QD Output DWord	Depending on IO configuration	Direct NVs output data
0xa4c0 IB Input Byte 0xa580 IW Input Word 0xa680 ID Input Dword	Depending on IO configuration	General NVs input data
0xa040 QB Output Byte 0xa100 QW Output Word 0xa200 QD Output DWord	Depending on IO configuration	General NVs output data

All network variables can also be accessed via "Service Data Object" (SDO).

Mapping to Process Data Objects (PDOs) can also be carried out via SDO. However, in the interests of diagnosis, simplification and avoiding errors we strongly recommend using the CANopen configurator, ProCANopen.



When PLC stop occurs, all network variable outputs (QW) are set to 0.



Network variables are read in and output like a process map using the CANIO driver.

Direct network variables are read in and output without process map.



### 6.2.3 Access to network variables and I/O configuration

Network variables, function codes and parameters of parameterizing and diagnosis functions are mapped to PLC addresses with the help of the I/O configuration, and provided with symbolic names in the variable declaration.

User-specific adaptation of the CANIO driver is possible via driver parameters.

CANIO user parameter:		
DRIVER_PARAMETER	Code	Content
DRIVER_PAR1	0 (default)	<ul style="list-style-type: none"> <li>Access to diagnosis data deactivated</li> <li>Process data (outputs) zeroed automatically on PLC-STOP</li> </ul>
DRIVER_PAR1	Bit 0 = 1	<ul style="list-style-type: none"> <li>Activate access to diagnosis data</li> </ul>
DRIVER_PAR1	Bit 1 = 1	<ul style="list-style-type: none"> <li>Process data (outputs) not zeroed automatically on PLC-STOP</li> </ul>
DRIVER_PAR2	0 (default)	<ul style="list-style-type: none"> <li>Process map update without use of semaphores</li> <li>Guarantees 32-bit data consistency, no consistency over the complete map.</li> <li>Fast with low overhead</li> <li>Suitable for pure I/O accesses</li> </ul>
DRIVER_PAR2	Bit 0 = 1	<ul style="list-style-type: none"> <li>Process map update using semaphores</li> <li>Data consistency over whole network variable range</li> <li>Slow and involving overhead, with waiting time for enabling semaphores</li> <li>Uses the timeout value from DRIVER_PAR3</li> <li>e.g. for superimposed protocols</li> </ul>
DRIVER_PAR2	Bit 1 = 1	<ul style="list-style-type: none"> <li>Prevents task change during process data update</li> </ul>
DRIVER_PAR2	Bit 2 = 1	<ul style="list-style-type: none"> <li>Double buffer mode, process map update using semaphores</li> <li>Data consistency over whole network variable range</li> <li>Fast but involving overhead, without waiting time for enabling semaphores</li> <li>e.g. for exchange of consistent data structures</li> </ul>
DRIVER_PAR3	Value	<ul style="list-style-type: none"> <li>Timeout in <math>\mu</math>s (when semaphores are used)</li> <li>default max. 500 <math>\mu</math>s</li> </ul>



## Example

```
(* CAN1> - Don't remove this label*)
PROGRAM netin1 : INPUT
(
    VAR_ADR      := 1000, (* CAN card / network 1 *)
    END_VAR_ADR := 1255,
    DEVICE       := DRIVER,
    DRIVER_NAME  := 'CANIO'
    (* use defaults *)
);
PROGRAM netout1 : OUTPUT
(
    VAR_ADR      := 1000,
    END_VAR_ADR := 1255,
    DEVICE       := DRIVER,
    DRIVER_NAME  := 'CANIO'
    (* use defaults *)
);
(* diagnostic interface *)
PROGRAM netin1d : INPUT
(
    VAR_ADR      := 1256, (* CAN card / network 1 *)
    END_VAR_ADR := 1259,
    DEVICE       := DRIVER,
    DRIVER_NAME  := 'CANIO',
    DRIVER_PAR1  := 1
);
PROGRAM netout1d : OUTPUT
(
    VAR_ADR      := 1256, (* CAN card / network 1 *)
    END_VAR_ADR := 1259,
    DEVICE       := DRIVER,
    DRIVER_NAME  := 'CANIO',
    DRIVER_PAR1  := 1
);
(* <CAN1 - Don't remove this label *)
```

## Declaration in PLC program:

```
VAR_GLOBAL
mNVInput1      AT %IW 1000 : INT;
mNVOutput1     AT %QW 1000 : INT;
mDiag1Input1   AT %IW 1256 : UINT;
mDiag1Input2   AT %IW 1258 : UINT;
mDiag1Output1  AT %QW 1256 : UINT;
mDiag1Output2  AT %QW 1258 : UINT;
VAR_END
```

Also note that DRIVER\_PAR1=1 must be set if the CANIO driver is to access the parameterizing and diagnosis functions. Maps for the memory locations of the parameterizing and diagnosis functions are generated here.



If double buffer mode is activated anywhere in the I/O configuration, the setting will apply to all driver accesses.

### 6.2.4 Accessing direct network variables

Direct network variables are declared as follows in the PLC program:

```
VAR_GLOBAL
(* CAN card / network 1 *)
mNV1Input  AT %MW 3.1010000 : UDINT;
mNV1Output AT %MW 3.1010512 : UDINT;
(* CAN card / network 2 *)
mNV2Input  AT %MW 3.1020000 : UDINT;
mNV2Output AT %MW 3.1020512 : UDINT;
VAR_END
```

Similarly, the corresponding memory locations of the parameterizing and diagnosis functions can also be accessed directly, i.e. without assistance from the CANIO driver.

```
VAR_GLOBAL
mDiag1Input1 AT %MW 3.1019992 : UINT;
mDiag1Input2 AT %MW 3.1019994 : UINT;
mDiag1Output1 AT %MW 3.1019996 : UINT;
mDiag1Output2 AT %MW 3.1019998 : UINT;
VAR_END
```

Here, no maps are generated for the memory locations of the parameterizing and diagnosis functions.

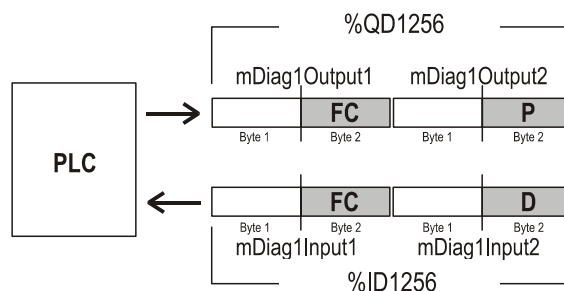
### 6.2.5 Accessing parameterizing and diagnosis functions

The PLC can initiate parameterizing and diagnosis functions in the CAN processor.

The PLC requests a function by entering the required function code in the mDiag1Output1 variable.

If a function is to be sent with parameter, first the parameter has to be entered in mDiag1Output2 and then the function code in mDiag1Output1.

After the function has been executed the function code is always displayed in mDiag1Input1, while the diagnosis data can be read in mDiag1Input2.



**FC = Function Code**  
**P = Parameter**  
**D = Data**

Direct assignment in 32-bit variables is also possible (e.g. %QD1256 and %ID1256).

## 6.3 SDO Function Blocks

### Supported functions

SDO write  
SDO read  
Read error and emergency entries

More detailed descriptions can be found in the CiA Draft-Standard 301 and the respective profiles (e.g. CiA DSP 405, 401 etc.).

### 6.4 Parameterizing and Diagnosis Functions

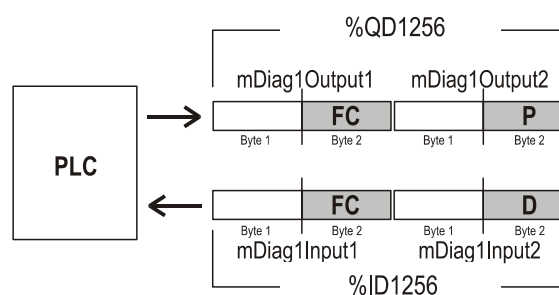
The PLC can initiate parameterizing and diagnosis functions in the CAN processor\* by transmitting a code.

Function	
0	No function
1,17	Read and set nodenumber (node ID), range: 1..127 Reboot required
2	Read and set baud rate, range: 1..8 Reboot required
3 ... 5	Reserved
6	Output CAN status code
7	Output error status
8	Output firmware version
9..15	Reserved
16	Output effective runtime of CAN stack (in 1/100 ms)
17..19	Reserved
20	Output and set startup delay, range: 1..60 s
21...101	Reserved
102	Output and set cycle time of CAN stack, range: 1..10 ms Reboot required
103...254	Reserved
255	Reset

The PLC requests a function by entering the required function code in the mDiag1Output1 variable.

If a function is to be sent with parameter, first the parameter has to be entered in mDiag1Output2 and then the function code in mDiag1Output1.

After the function has been executed the function code is always displayed in mDiag1Input1, while the diagnosis data can be read in mDiag1Input2.



**FC = Function Code**

**P = Parameter**

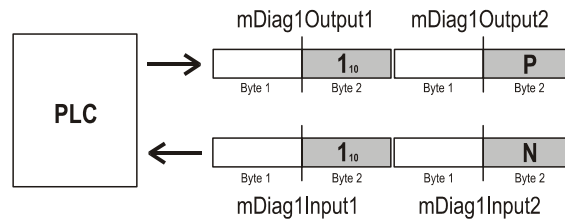
**D = Data**

Direct assignment in 32-bit variables is also possible (e.g. %QD1256 and %ID1256).

\* Here "CAN processor" refers to the hardware and software of the CANopen interface.

## 6.4.1 Function 1 Set and read CANopen node number

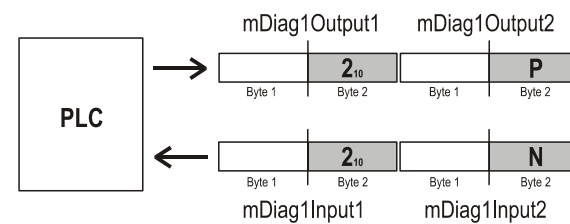
P Parameter	
0	Read node number
1	Set node number 1
.	.
127	Set node number 127



N Current set node number	
1	Node number 1
.	.
127	Node number 127

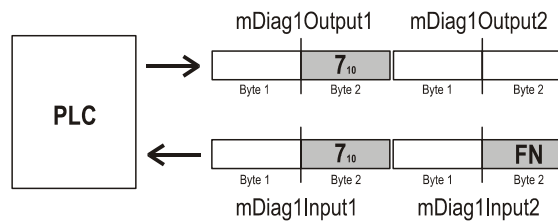
## 6.4.2 Function 2 Set and read data transmission rate

P Parameter	
0	Read current set data transmission rate
1	Set 10 kBaud
2	Set 20 kBaud
3	Set 50 kBaud
4	Set 125 kBaud
5	Set 250 kBaud
6	Set 500 kBaud
7	Set 800 kBaud
8	Set 1000 kBaud



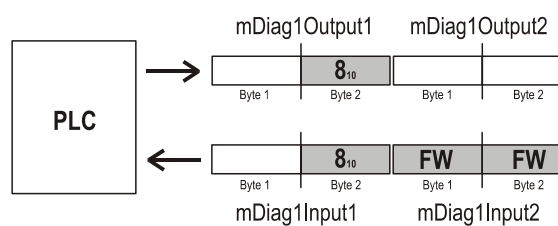
N Current set data transmission rate in kBaud	
1	10
2	20
3	50
4	125
5	250
6	500
7	800
8	1000

### 6.4.3 Function 7 Output CANopen error number



**FN** Error number - see error messages on page 83

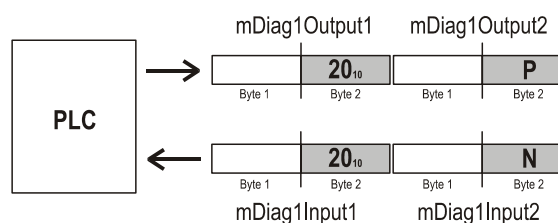
### 6.4.4 Function 8 Read CANopen firmware version



**FW** Firmware version

The firmware version is displayed in hexadecimal code.

### 6.4.5 Function 20 Set CANopen startup delay



**P** Startup delay in seconds (default)

**N** Startup delay in seconds (actual value)



When power is switched on the CAN stations require different amounts of time to be ready for CAN telegrams. The NMT manager must not send telegrams until all bus stations are ready. So a startup delay setting is only required if the XCx is configured as NMT manager.

## 6.5 Error messages

Error messages from the CAN processor are entered in the "active error buffer" or the "log book". These entries can be displayed with the NC-Dialog startup tool.

The error number can also be output by the PLC with diagnosis function 7.

The "Error Messages" section of the XCx manual contains an explanation of the error messages.

## 6.6 Expansion Module XF-CAN

The module is used to expand the XCx 300 and the XCx 540. The module is equipped with following features:

- One CANopen field bus interface
- Correspond to CAN Application Layer for industrial application CAL, data transfer layer equal ISO 11898 und CAN 2.0 A+B communication profil CiA DS-301

### 6.6.1 Displays and Connections

### 6.6.2 Technical data

Basic data	
Interface	CANopen
Connection system	Subminiature type D, 9-pin, socket

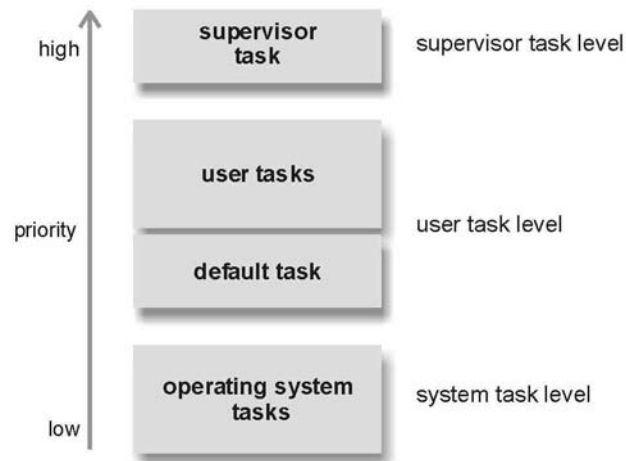


## 7 The XCx Multi-Tasking System

### 7.1 Overview

The PLC runtime system is based on a real-time operating system controlled by task priorities. There are three priority levels for tasks:

- Supervisor task level
- User task level
- System task level



In the operating system there is a specially protected level for the supervisor task. The supervisor task is a system task with maximum priority. The supervisor task detects errors such as division by zero and task time overrun, and activates the appropriate system task.

All tasks inserted by the user run at the user/default level.

Certain important firmware tasks that have to be considered when parameterizing user tasks also run on this level. See section Task Priorities.

Tasks running at the system task level are not influenced by the user; e.g. communication tasks, debug tasks, memory management tasks, system control tasks.



## 7.2 User Tasks

User tasks are all tasks that can be inserted by the application programmer.

The default task is also on the user task level. It is the user task with the lowest priority. The default task is executed when no other user task is active.



Incorrectly or inappropriately selected user task settings for type, priority, interrupt mode, etc. – especially in conjunction with longer program runtimes – can lead to controller malfunction when essential operating system tasks are displaced.

See section Task Priorities.

You can use various types of user task.

### 7.2.1 Cyclical tasks

Cyclical tasks execute the programs assigned to them within a defined interval under a user-defined priority.

In MULTIPROG you can give the individual tasks a priority between 0 and 31. Task 0 has the highest priority, task 31 the lowest. The task with the highest priority is called first. The user-task priorities are mapped to the priority levels of the real-time operating system. (See section Task Priorities)

If the watchdog time of a cyclical task is higher than the set interval time and task execution has not been completed, one or more execution cycles will be omitted.

### 7.2.2 Event tasks

The XCx operating system starts event tasks when particular events occur.

The following events are currently defined.

Internal designation	Event number	Comments
Interrupts		
PLC_EVENT_XFIO_I0	0	XFIO Interrupt (Input 0)
PLC_EVENT_XFIO_I1	1	XFIO interrupt (Input 1)
Reserved	2..3	
Synchronization		
PLC_EVENT_POS	4	Position controller task (XCN only)
PLC_EVENT_CAN	5	CANopen task
PLC_EVENT_IPO	6	CNC IPO task (XCN only)
PLC_EVENT_DECO	7	CNC DECO task (XCN only)
Reserved	8..15	

The event number is used in the MULTIPROG task setting to specify the event that starts the event task.

The specified priority is used unless a bypass option is set by the system. (Bypass cancels the normal task change so that the assigned programs are executed immediately when the event occurs.)

Up to 16 events will be put in a queue. So these events are not lost, and will be executed later. This also applies if new events occur before the assigned event task is executed.



### 7.2.3 System tasks

System tasks and system programs (SPGs) are started automatically by the operating system when an event occurs in connection with the operating system.

The SPGs which can be used are listed in the following table:

No.	Name	Event	Actions
SPG 0	WARM_START	Is executed during a warm start	<ul style="list-style-type: none"> <li>• Retentive data is not initialized</li> <li>• Non-buffered data is initialized</li> <li>• The open function of the I/O driver is executed</li> <li>• User tasks are activated</li> <li>• PLC switches to 'run' state</li> </ul>
SPG 1	COLD_START	Is executed during a cold start	<ul style="list-style-type: none"> <li>• All data is initialized</li> <li>• The open function of the I/O driver is executed</li> <li>• User tasks are activated</li> <li>• PLC switches to 'run' state</li> </ul>
SPG 2	TO_STOP	Is executed when program execution is stopped	<ul style="list-style-type: none"> <li>• User tasks are deactivated</li> <li>• All outputs are updated</li> <li>• The close function of the I/O driver is executed</li> <li>• PLC switches to 'STOP' state</li> </ul>
SPG 10	WATCHDOG	Is executed when a task has not been completed within its watchdog time	<ul style="list-style-type: none"> <li>• User tasks are deactivated</li> <li>• All outputs are updated</li> <li>• The close function of the I/O driver is executed</li> <li>• PLC switches to 'STOP' state</li> </ul>
SPG 11	ZERODIV	Is executed if division by zero occurs during program execution	<ul style="list-style-type: none"> <li>• User tasks are deactivated</li> <li>• All outputs are updated</li> <li>• The close function of the I/O driver is executed</li> <li>• PLC switches to 'STOP' state</li> </ul>
SPG 12	STACKOVER	Is executed if a stackover has occurred. Is only executed if the 'Stack-Prüfung' ['Stack check'] checkbox in the 'Ressource ... einrichten' [Resource ... Set up] dialog in MULTIPROG was activated.	<ul style="list-style-type: none"> <li>• User tasks are deactivated</li> <li>• All outputs are updated</li> <li>• The close function of the I/O driver is executed</li> <li>• PLC switches to 'STOP' state</li> </ul>
SPG 13	BADCAL	Is executed if a non-existent manufacturer-specific POU is called	<ul style="list-style-type: none"> <li>• User tasks are deactivated</li> <li>• All outputs are updated</li> <li>• The close function of the I/O driver is executed</li> <li>• PLC switches to 'STOP' state</li> </ul>
SPG 14	IOERROR	Is executed if an error occurs in the I/O driver while the process is running	<ul style="list-style-type: none"> <li>• PLC continues execution</li> </ul>
SPG 16	MATHERR	Is executed if a sliding point error occurs in an arithmetic function	<ul style="list-style-type: none"> <li>• User tasks are deactivated</li> <li>• All outputs are updated</li> <li>• The close function of the I/O driver is executed</li> <li>• PLC switches to 'STOP' state</li> </ul>
SPG 17	CPU_OVERLOAD	Is executed if a CPU overload occurs	<ul style="list-style-type: none"> <li>• User tasks are deactivated</li> <li>• All outputs are updated</li> <li>• The close function of the I/O driver is executed</li> <li>• PLC switches to 'STOP' state</li> </ul>
SPG 18	INITIODRV_ERR	Is executed if an error occurs in I/O driver initialization during a	<ul style="list-style-type: none"> <li>• PLC does not start</li> </ul>



No.	Name	Event	Actions
		cold or warm start	
SPG 19	BOUNDS_ERR	Is executed if the limits of an array or a structure are exceeded. Is only executed if the 'Index-Prüfung' [Index check] or 'Feldbegrenzungs-Prüfung' [Array limit check] checkbox in the 'Ressource ... einrichten' [Resource ... Set up] dialog in MULTIPROG was activated.	<ul style="list-style-type: none"> <li>• User tasks are deactivated</li> <li>• All outputs are updated</li> <li>• The close function of the I/O driver is executed</li> <li>• PLC switches to 'STOP' state</li> </ul>
SPG 20	BUS_ERR	Is executed if variables with a data type $\geq 2$ bytes and uneven addresses were used or an internal error has occurred in MULTIPROG. Only on Motorola platforms.	<ul style="list-style-type: none"> <li>• User tasks are deactivated</li> <li>• All outputs are updated</li> <li>• The close function of the I/O driver is executed</li> <li>• PLC switches to 'STOP' state</li> </ul>
SPG 21	STRING_ERR	Is executed if an error has occurred in a character string operation, e.g. if one character string is to be replaced by another, but cannot be found.	<ul style="list-style-type: none"> <li>• The behaviour of a character string exception has changed! In the standard setting SPG 21 is called after a character string exception has occurred. An entry with the module number and line number is also made in the error catalog. The PLC remains in 'RUN' status.</li> </ul>



System tasks are not monitored by the watchdog.



### 7.2.4 Default task

The default task runs as a background task with the lowest possible user priority and is not time-monitored. It is activated when all higher-priority user tasks have been processed. The default task is configured so that it uses some of the available residual time (max. 80 %) but is executed no more than one every 20 ms. Only one default task is permitted in each resource.



All drivers in the I/O configuration that are not explicitly assigned to a user task automatically activate the default task and are executed in the context of the default task.

---

## 7.3 User Task Information

Information is mapped to system variables for each user task. The type definitions listed below for the system variables can be found in the PLC\_Types section of the SchleicherLib library.

TYPE

```
TaskNameType : ARRAY [1..10] OF BYTE;
```

END\_TYPE

TYPE

```
TaskInfoType0 : STRUCT
```

```
MaxTask : INT; (* 00: *)
```

Max. poss. number of tasks

```
CurTask : INT; (* 02: *)
```

Current number of tasks

```
END_STRUCT (* TaskInfoType0 *);
```

END\_TYPE

TYPE

```
TaskInfoType1 : STRUCT
```

```
TaskName : TaskNameType; (* 04: *)
```

Task name

```
TaskPrio : INT; (* 14: *)
```

Task priority

```
TaskMode : INT; (* 16: *)
```

Task mode

```
TaskPeriod : INT; (* 18: [ms] *)
```

Task period in ms

```
TaskStack : INT; (* 20: *)
```

Size of used task stack

```
MainPoe : INT; (* 22: assigned PLC program *)
```

Assigned PLC program

```
TaskWatchDog : INT; (* 24: [ms] *)
```

Watchdog time in ms

```
reserve0 : DINT; (* 26: *)
```

```
MaxStack : INT; (* 30: max. used stack *)
```

Size of poss. task stack

```
CurDuration : INT; (* 32: [ticks] *)
```

Current task duration including prioritized calls

```
MinDuration : INT; (* 34: [ticks] *)
```

Minimum task duration

```
MaxDuration : INT; (* 36: [ticks] *)
```

Maximum task duration

```
AveDuration : INT; (* 38: [ticks] *)
```

Average task duration

```
CurDelay : INT; (* 40: [ticks] *)
```

Current task delay

```
MinDelay : INT; (* 42: [ticks] *)
```

Minimum task delay

```
MaxDelay : INT; (* 44: [ticks] *)
```

Maximum task delay

```
AveDelay : INT; (* 46: [ticks] *)
```

Average task delay

```
END_STRUCT (* TaskInfoType1 *);
```

END\_TYPE



The variables are declared with *TaskInfoType0* and *TaskInfoType1*.



	Name	Typ	Verwendung	Beschreibung	Adresse	Anfangsw...	Reman...
	TaskInfo0	TaskInfoType0	VAR_GLOBAL		%MD 1.1000		<input type="checkbox"/>
	TaskInfo1	TaskInfoType1	VAR_GLOBAL		%MD 1.1004		<input type="checkbox"/>
	TaskInfo2	TaskInfoType1	VAR_GLOBAL		%MD 1.1068		<input type="checkbox"/>
	TaskInfo3	TaskInfoType1	VAR_GLOBAL		%MD 1.1132		<input type="checkbox"/>
	TaskInfo4	TaskInfoType1	VAR_GLOBAL		%MD 1.1196		<input type="checkbox"/>
	TaskInfo5	TaskInfoType1	VAR_GLOBAL		%MD 1.1260		<input type="checkbox"/>

The following user task information is declared with an offset of 64 starting at 1004 ( $1004 + 64 = 1068$  etc.).

The sequence of tasks is defined by the rank of the task in the *Physikalische Hardware/Configuration/Resource/Tasks* project tree.



## 7.4 Task Priorities

The table gives an overview of recommended task priorities and their relationship to important reserved firmware tasks (tfwLAGE, tfwCANhigh, tfwIPO).

MULTIPROG priority	RTOS* priority (default)	RTOS* task name	Application
0	30	Any	E.g. user task (event 0)
1	31	Any	E.g. user task (event 1)
2	32	Any	E.g. user task (event 4)
3	33	tfwLAGE	Reserved for position controller task (XCN only)
4	34	Any	E.g. user task (event 4, 5)
5	35	tfwCANhigh	Reserved for CAN stack task (option CAN_HIGH_PRIO = 1)
6	36	Any	E.g. user task (event 5, 6)
7	37	tfwIPO	Reserved for IPO task (XCN only)
8	38	Any	E.g. user task (event 5)
9	39	tfwCANhigh	Reserved for CAN stack task (option CAN_HIGH_PRIO = 0)
10	40	Any	E.g. user task (event 5)
11..15	41..45	Any	E.g. cyclical user tasks
16..31	46	Any	E.g. other cyclical user tasks
Default	127	Default	Background task

\*Real Time Operating System



Incorrectly or inappropriately selected user task settings for type, priority, interrupt mode, etc. – especially in conjunction with longer program runtimes – can lead to controller malfunction when essential firmware tasks are displaced (tfwLAGE, tfwCANhigh, tfwIPO).



Schleicher controllers support 18 user tasks (priority levels 0..16 and the default task).  
Tasks with priority  $\geq 16$  are executed with priority 16.

### 7.5 Tasks and Watchdogs

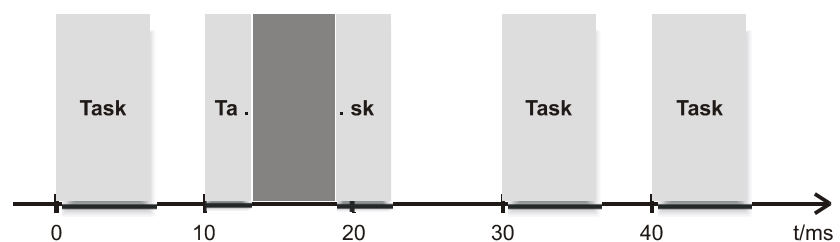
Each user-defined task has its own settable watchdog.

The watchdog checks that task execution is complete at the end of the watchdog time. If task execution is not complete at the end of this time the system task SPG 10 'WATCHDOG' is executed and the PLC switches to 'STOP' state if no other actions were programmed. An entry is also made in the error catalog. The watchdog time starts when the task is ready to execute. The watchdog interval is defined in the 'Task ... einrichten' [Task ... Setup] dialog in MULTIPROG.

If the execution time of the task and the watchdog time are roughly the same, and the CPU workload is high, the watchdog time may be exceeded during certain online operating steps.



The reason for this may be that you selected address status with powerflow when debugging in online mode.



In this example the watchdog time of the displayed task is set to 10 ms. In the figure the task exceeds its watchdog time at 20 ms. If the watchdog time of the task is set to 20 ms it will next be executed at 30 ms. In this case task execution at 20 ms will be skipped.

## 7.6 Inserting Tasks and Assigning Programs

To insert a task you have to carry out the following steps in MULTIPROG.

- Click with the right mouse button on the *Tasks* directory in the project tree, to open the context menu.



- Select *Insert/Task*, the "Insert dialog appears.
- Enter the name for the task.
- Set the required task type in the *Task Type* list.  
You can choose between default task, cyclical task, event task and system task.  
Note: If task type 'DEFAULT' is not listed, the resource already has a default task.
- Confirm the dialog with *OK*.  
The *Task Settings for ...* dialog appears. The dialog contains text and list fields, depending on the previously selected task.
- You have to enter the following parameters for the task:
 

Cyclical task	Time interval
Event task	Event number (number of interrupt)
System task	Number of a system program

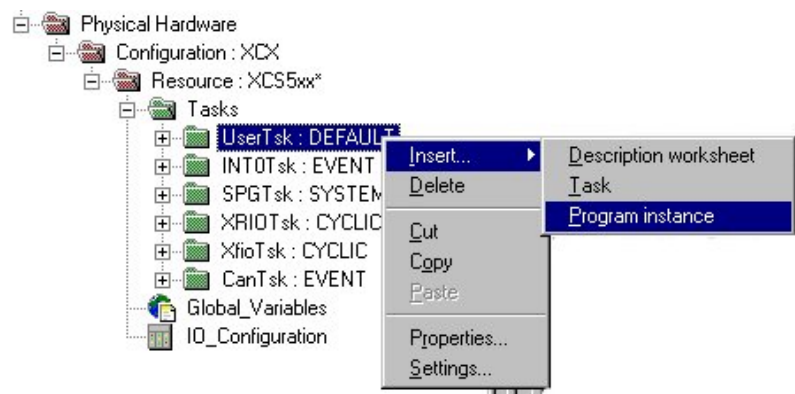
The instructions in section Task Priorities must be observed when assigning priorities.

Programs must be assigned to tasks before they can be executed. Assigning a program to a task means that an instance of the program will be executed when the task is activated. Different instances of a program can be assigned to different tasks.

Several programs can be assigned to one task. In this case the first program in the task directory will be executed first. Then the next program will be executed, and so on.

To insert programs you have to carry out the following steps in MULTIPROG.

- Click with the right mouse button on the project tree icon of the task in which the program is to be inserted.



- Select *Insert/Program instance* in the context menu.
- Enter an instance name for the program in the *Program instance* field.
- Set the required program in the *Program type* list box.
- Confirm the dialog with *OK*.  
The program symbol is inserted in the project tree.

## 8 The PLC

- Operating system: ProConOS
- Programming: MULTIPROG to IEC 61131-3
- Communication with the CNC via shared RAM

### 8.1 Programming

The XCx is programmed on a PC using MULTIPROG software (to IEC 61131-3).



The programming software is made up of the MULTIPROG software and the *AddOns* for MULTIPROG from Schleicher.

The programming system and programming instructions can be ordered as accessories. See also the section on "Accessories and Spare Parts".

The PLC is supplied with a ready-configured MULTIPROG project, which you can use as the basis for programming the PLC (see "Start-up" section).

### 8.2 PLC Operating States and Starting Behaviour

#### 8.2.1 Operating states

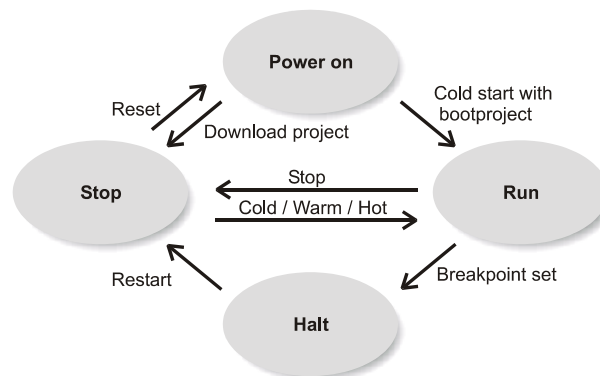
Operating state	Description
POWER ON	Power supply switched on No program loaded
STOP	Program loaded User tasks inactive Process map memory inputs are not updated Output signals are not sent to inputs and outputs
RUN	Program execution active User tasks active Process map memory inputs updated according to I/O configuration Process map memory outputs updated according to I/O configuration and program execution
HALT	Program execution will stop at a breakpoint User tasks inactive Process map memory inputs are not updated Process map memory outputs are not updated

The current state of the PLC is displayed in the MULTIPROG project control dialog in the "Status:" line.

If 'debug' is displayed behind the current state in the control dialog it means that breakpoints have been set or variables forced.

### 8.2.2 Changing operating states with MULTIPROG

You can use the graphic user interface of MULTIPROG to control when program execution on the PLC starts and stops. The buttons for changes which are not possible in the current operating state are shaded in the project control dialog.



#### Starting program execution

State change from → to	Button in control dialog	Description of what happens
Stop → Run	<b>Cold</b>	<ul style="list-style-type: none"> <li>• Cold start</li> <li>• All data is initialized</li> <li>• SPG 1 is called</li> <li>• All user tasks are activated</li> <li>• Program execution activated</li> </ul>
Stop → Run	<b>Warm</b>	<ul style="list-style-type: none"> <li>• Warm start</li> <li>• Only non-buffered data is initialized</li> <li>• SPG 0 is called</li> <li>• All user tasks are activated</li> <li>• Program execution activated</li> </ul>
Stop → Run	<b>Hot</b>	<ul style="list-style-type: none"> <li>• Hot start</li> <li>• No data is initialized</li> <li>• All user tasks are activated</li> <li>• Program execution activated</li> <li>• Not available when you start program execution for the first time after downloading</li> </ul>

#### Stopping program execution

State change from → to	Button in control dialog	Description of what happens
Run → Stop	<b>Stop</b>	<ul style="list-style-type: none"> <li>• All user tasks are deactivated when their operating cycle is complete</li> <li>• SPG 2 is called</li> <li>• Process map memory outputs are written</li> <li>• Program execution stops</li> <li>• Physical outputs are set to zero or preferred shut-off state</li> </ul>

#### General reset

State change from → to	Button in control dialog	Description of what happens
Stop → Power on	<b>Reset</b>	<ul style="list-style-type: none"> <li>• The project is deleted</li> <li>• General reset</li> </ul>

### 8.2.3 PLC starting behaviour after power supply is switched on

The PLC starting behaviour is set with the lockswitch on the XCx.

The following options are available:

- PROG PLC stop
- WARM PLC warm start to IEC 61131-3
- COLD PLC cold start to IEC 61131-3

## 8.3 System Variables

System variables provide information about the status of the system, for example about forced variables, CPU performance, etc. These variables have fixed memory addresses and can be used by the PLC program to obtain the corresponding information.

All the system variables in the following table are already declared in the Global\_Variables area of the Global\_Variables worksheet.

System variables				
Name	Data type	Log. addr. (byte)	Log. addr. (bit)	Description
PLCMODE_ON	BOOL	0	0	TRUE := current PLC state is ON
PLCMODE_RUN	BOOL	0	1	TRUE := current PLC state is RUN
PLCMODE_STOP	BOOL	0	2	TRUE := current PLC state is STOP
PLCMODE_HALT	BOOL	0	3	TRUE := current PLC state is HALT
PLCDEBUG_BPSET	BOOL	1	4	TRUE := one or more breakpoints have been set
PLCDEBUG_FORCE	BOOL	2	0	TRUE := one or more variables have been forced
PLCDEBUG_POWERFLOW	BOOL	2	3	TRUE := powerflow active
PLC_TICKS_PER_SEC	INT	44	-	Number of system ticks per second, used by the PLC as the basis for the system time. This value determines the time resolution of the PLC for time delay function blocks like TON, TOF and TP, and the shortest cycle time for the DEFAULT task and cyclical tasks.
PLC_SYS_TICK_CNT	DINT	52	-	Number of counted PLC system ticks

As well as these system variables, other variables are also defined, containing information on the system.

The type definitions of the variables can be found in the PLC\_Types section of the SchleicherLib library.



### 8.4 Libraries and Function Blocks in MULTIPROG

The available function blocks are combined in libraries.

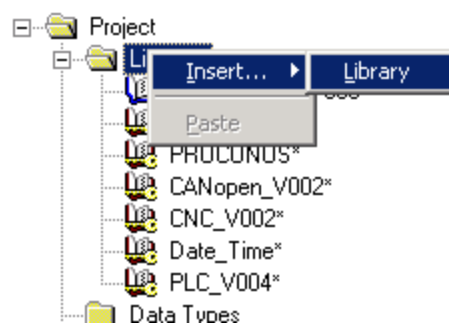
If new projects created with MULTIPROG, different libraries are inserted automatically, depending from the controller type. Other libraries can be inserted manually.

Libraries	XCN 7xx	XCS 7xx	XCN 5xx	XCS 5xx	XCN 3xx	XCS 3xx	Pro- Numeric	Pro- SyCon	MCS 20-20	MCS 20-21	Simulation
PROCONOS	+	+	+	+	+	+	+	+	+	+	+
BIT_UTIL	+	+	+	+	+	+	+	+	+	+	+
CANopen_Vxxx	+	+	+	+	o	o	+	+	-	+	-
CFB_Vxxx	o	o	o	o	o	o	o	o	-	-	-
CNC_Vxxx	+	-	+	-	+	-	+	-	-	-	-
Date_Time	+	+	+	+	+	+	+	+	+	+	-
Microline	-	-	-	-	-	-	-	-	+	+	-
MMI	o	o	o	o	o	o	o	o	o	o	-
PLC_Vxxx	+	+	+	+	+	+	+	+	-	-	-
Profibus_Vxxx	o	o	o	o	o	o	o	o	-	-	-
Serial	o	o	o	o	o	o	o	o	o	o	-
SchleicherLib_Vxxx	+	+	+	+	+	+	+	+	-	-	-
XCx7_Vxxx	+	+	-	-	-	-	-	-	-	-	-

- + inserted automatically if a new MULTIPROG project created
- o can be inserted manually
- not needed

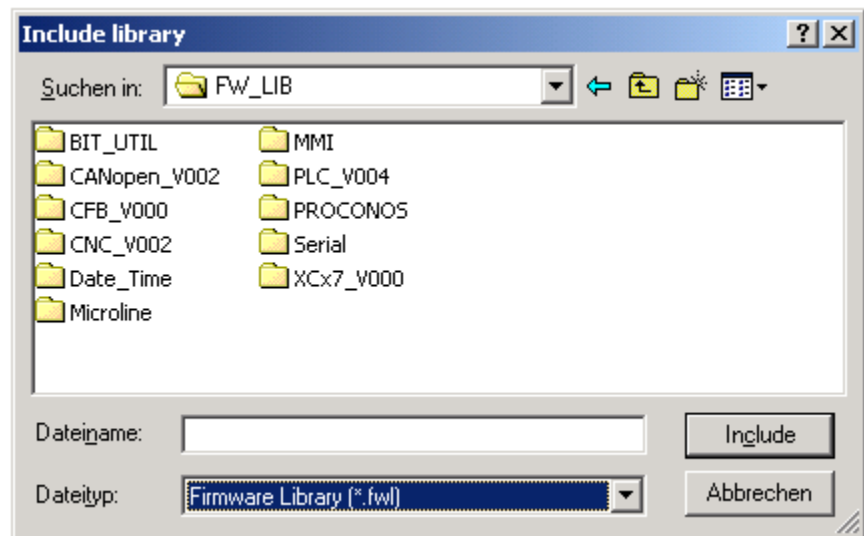
If other libraries needed, insert as follows:

- Click with the right mouse button on *Libraries* in the project tree to open the context menu *Insert/Library*.



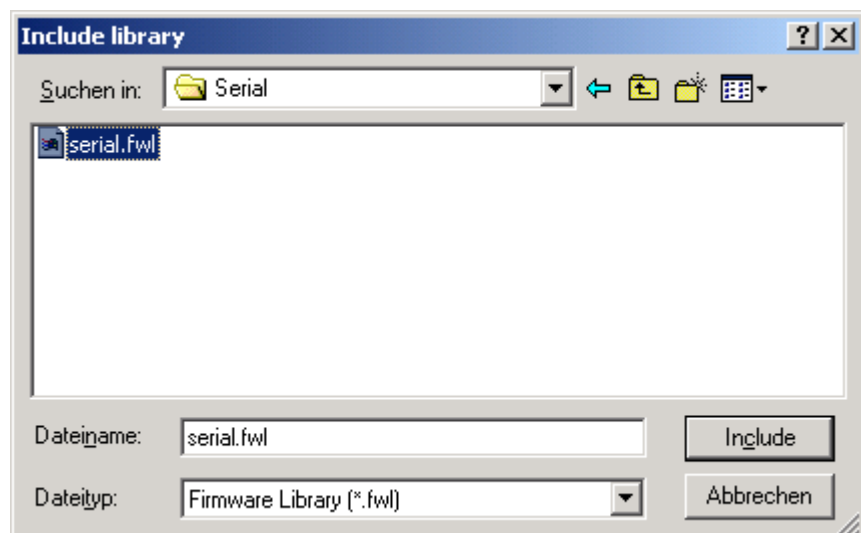
- Select the path ...KWSofMWT\PLC\FW\_LIB and the *File Type Firmware Library (\*.fwl)*.





Each library is stored in a single directory.

Open the needed directory e.g. *Serial* to include the *serial.fwl*.



An online help function providing detailed information on the selected topic is available for the libraries (except SchleicherLib). You activate online help via the context menu of the respective library (click on the library icon with the right mouse button).

**Note:** Variable declarations in function block example programs

The example programs for the function blocks contain variable declarations to IEC 61131-3 with keywords VAR and END\_VAR. If you wish to use the example programs with MULTIPROG you will have to enter the variable declarations manually, in tabular form on the variables worksheet of the respective POU.



#### 8.4.1 CANopen\_Vxxx library

The library contains function blocks for parameterizing and diagnosis the CANopen network.

Function block	No.	Description
CO_NET_SDO_WRITE	150	Sends a Service Data Object (SDO)
CO_NET_SDO_READ	151	Receives a Service Data Object (SDO)
CO_NET_GET_LOCAL_NODE_ID	152	Returns own node ID
CO_NET_GET_STATE	153	Supplies current CANopen status
CO_NET_GET_KERNEL_STATUS	154	Supplies current extended CANopen kernel status
CO_NET_NMT	155	Sets status of one or all devices in the CANopen network
CO_NET_RECV_EMY_DEV	156	Reads any emergency messages from a particular network node
CO_NET_RECV_EMY	157	Reads any emergency messages from any network node
CO_NET_RECV_ERR_DEV	160	Reads any error messages from a particular network node
CO_NET_RECV_ERR	161	Reads any error messages from any network node
CO_NET_SENDL2	162	Sends any CAN Layer 2 messages
CO_NET_PING	163	Executes a ping on a particular network node
CO_NET_RESTART_CAN	164	Restarts CANopen communication (e.g. after "bus- off")
CO_NET_RESTART_ALL	165	Restarts the complete CANopen stack
CO_NET_SHUTDOWN	166	Stops the CANopen stack
CO_NET_CAN_SYNC	170	Allows synchronization between PLC task and CANopen stack

#### 8.4.2 CFB\_Vxxx Library

Library like IEC 61131-5 with function blocks for "peer-to-peer" communication with TCP/IP.

Function blocks	No.	Description	Controller
CONNECT_V	60	Connect two members	XCx
USEND_V	61	Send data	ProNumeric
URCV_V	62	Recive data	ProSyCon

## 8.4.3 CNC\_Vxxx library

The CNC library contains function blocks for reading and writing system data, SERCOS and XRIO motion parameters.

Function block	Nr.	Description	Controller
READ_Q_PARAM_*	200 bis 207	Reads a CNC system data parameter	XCN ProNumeric
WRITE_Q_PARAM_*	208 bis 215	Writes a CNC system data parameter	
SAVE_Q_PARAM_*	221	Saves the CNC system data parameter to the hard disk	XCN
SAVE_R_PARAM_*	220	Saves the CNC arithmetic parameter to the hard disk	
READ_SERC_PARAM	302	Reads a SERCOS parameter	
WRITE_SERC_PARAM	303	Writes a SERCOS parameter	
SET_SERC_PHASE	304	Switching SERCOS communication phase	
SET_SERC_COMMAND	308	execute a SERCOS command	
MC_ANALOG	300	XRIO Motion Control block	
MC_ANALOG_1_AXIS	307	XRIO Motion Control block one axis	
MC_CAN	301	CAN MotionControl block	
MC_DP	309	PROFIBUS-DP Motion Control block	
MC_DP_1_AXIS	310	PROFIBUS-DP Motion Control block one axis	

**Note**

The function blocks MC\_ANALOG und MC\_CAN are moved from library PCL\_Vxxx to library CNC\_Vxxx (effective from CNC\_V003 / PLC\_V005).

The function blocks MC\_DP und MC\_DP\_1\_AXIS available from version CNC\_V004.

## 8.4.4 Date\_Time library

The real time clock, included in XCx, can be read/set with the following function blocks.

Function block	No.	Description	Controller
GET_TIME	130	Reads time	all
GET_DATE	128	Reads date	
SET_TIME	131	Set time	
SET_DATE	129	Set date	



### 8.4.5 MMI Library

Communication with operating terminals COP using the serial interface of controller. The used protocol is PNet.

Function block	No.	Description	Controller
PPF_COP_COMM	140	Communication with COP	XCx MCS xx-xx

### 8.4.6 PLC library

This library provides controller-specific firmware function blocks over and above the standard IEC/ProConOS function blocks.

Function block	No.	Description	Controller
PUT_ERROR	400	Generates a user-defined error message (not for new design, instead of this use 401)	all
PUT_ERROR2	401	Generates a user-defined error message	
CLEAR_ERROR	402	Deletes an error message sent with a lock flag	
READ_FILE	405	File access read	
WRITE_FILE	406	File access write	
SEND_MAIL	410	Sends an E-MAIL (SMTP client)	
XFIO_CONFIG	420	XFIO interrupt configuration	
XRIO_STATE	422	XRIO status information	
GET_MTS	430	Supplies current time value in $\mu$ s ticks	

Note (up to CNC\_V003 / PLC\_V005):  
The function blocks MC\_ANALOG and MC\_CAN moved from library PCL\_Vxxx to library CNC\_Vxxx.

### 8.4.7 Profibus\_Vxxx Library

The library provides function blocks for PROFIBUS communication

Function block	No.	Description	Controller
DP_NET_GET_STATE	190	Get status of the PROFIBUS card	XCx
DP_NET_PUT_MSG	191	Put message to Message Interface of the Hilscher card	ProNumeric
DP_NET_GET_MSG	192	Get message from Message Interface of the Hilscher card	ProSyCon

#### 8.4.8 Serial Library

Function blocks for controllers serial communication.

Function block	No.	Description	Controller
PORT_OPEN	135	Open a serial interface	all
PORT_CLOSE	136	Close a serial interface	
PORT_READ	137	Write character to serial interface	
PORT_WRITE	138	Read characters from serial interface	
PORT_STATE	139	Read status of a serial interface	

#### 8.4.9 SchleicherLib\_Vxxx library

XCx firmware data type definitions for MULTIPROG.  
The library has no function blocks.

#### 8.4.10 XCx7\_Vxxx Library

Function blocks only for XCx controllers.

Function block	No.	Description	Controller
UZH_VR	250	Working with UZH 2VR modules	XCx 700
UBA_ERR_CTRL	251	UBA modules error handling	
READ_AXIS_PAGE	305	Read parameter from axis specific remote page	
WRITE_AXIS_PAGE	306	Write parameter to axis specific remote page	



## 8.5 PLC Operating System ProConOS

### 8.5.1 ProConOS.INI initialization file

You can use the ProConOS.INI file to make application-specific changes to advanced settings (e.g. communication drivers, system tasks, CANopen stacks).

If ProConOS.INI does not exist or has been deleted, a file called "initial" containing default values will be generated when the controller software starts up.

The file is saved on the compact flash at  
*/ata0/OS/PLC/ProConOS.INI.*

#### Description of section and key entries

```
[PLC]

; start higher-priority PLC user tasks
USR_HIGH_PRIO = 1      ; yes = 1 (default), no = 0

; use the ProConOS socket communication driver
PC_SOCKET_DRV = 1      ; yes = 1 (default), no = 0
; max. number of ProConOS clients
; for simultaneous access to controller
PC_SOCKET_BLOG = 4     ; default

; use the ProConOS serial 0/1 communication driver
PC_SERIAL0_DRV = 0     ; yes = 1, no = 0 (default)
PC_SERIAL0_BR = 19200  ; Baud rate = 19200 (default)
PC_SERIAL1_DRV = 0     ; yes = 1, no = 0 (default)
PC_SERIAL1_BR = 19200  ; Baud rate = 19200 (default)

[CNC]

; start the higher-priority CNC IPO task
IPO_HIGH_PRIO = 0 ; yes=1, no=0 (default)

[CAN]

; start the higher-priority CANopen task
CAN_HIGH_PRIO = 0      ; yes = 1, no=0 (default)

; Restart the CANopen process after PLC STOP (NMT master!)
RESTART_CAN = 0 ; yes=1, no=0 (default)

; PLC STOP after CAN heartbeat error
HBE_STOP_PLC = 1 ; yes = 1 (default), no = 0
; PLC STOP after CAN Bus Off
CBO_STOP_PLC = 0 ; yes = 1 (default), no = 0

[PATHS]

; path for saving CNC online logging
ONLCONFIGPATH = "/ata0/OS/ONL"
; path for saving NC files
NCFILESPATH = "/ata0/OS/CNC"
```

## 9 The CNC

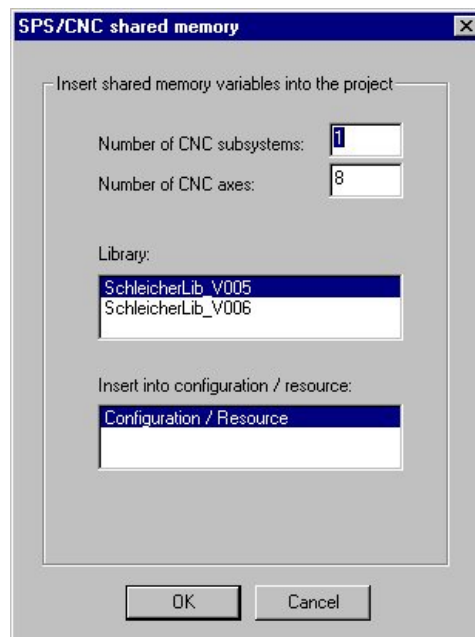
### 9.1 Configuration of the shared RAM PLC-CNC

The used shared RAM variables must be included in the PLC project.

- Choose in MULTIPROG *Extras/PLC/CNC shared memory*.



- Set the number of CNC subsystems and the number of CNC axes are used.  
The version of the used library must be correct. See chapter PLC section Libraries.



- The right variables will be inserted in Global\_Variables section CNC\_Common. The variables are marked as OPC.

Global_Variables: Configuration.Resource									
	Name	Type	Usage	Description	Address	Init	Retain	PDD	OPC
+	Global_Variables								
+	I/O_Variables								
+	Network_Variables								
+	PLC_Common								
-	CNC_Common								
	cmcSflgP2N_bCalc...	BOOL	VAR_GLOBAL		%MX 3.00007 ...		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	cmcSflgP2N_bEnR...	BOOL	VAR_GLOBAL		%MX 3.00007 ...		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	cmcSflgP2N_bBlkF...	BOOL	VAR_GLOBAL		%MX 3.00007 ...		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	cmcSflgP2N_bSupl...	BOOL	VAR_GLOBAL		%MX 3.00007 ...		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	cmcSflgN2P_bNcRdy	BOOL	VAR_GLOBAL		%MX 3.00007 ...		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	cmcSflgN2P_bLink...	BOOL	VAR_GLOBAL		%MX 3.00007 ...		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	cmcSflgN2P_bIpoFail	BOOL	VAR_GLOBAL		%MX 3.00007 ...		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	cmcSflgN2P_bIpoS...	BOOL	VAR_GLOBAL		%MX 3.00007 ...		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

## 9.2 Important Notes on the Multi-Tasking System and CNC



If the CNC is to operate safely it is important that you know the task hierarchies and take them into consideration in the individual user tasks.

For the CNC:

- The total runtime of the user tasks with priority 0 to 2 must not exceed half the runtime of the position controller task (priority 3).
- The user task running synchronously with the IPO level (event 6) must have a priority between 7 (IPO task) and 3 (position controller task).
- The CAN process is called by the system at the same time as the IPO task if the CAN cycle time is the same as the IPO cycle or position controller cycle.  
The user task associated with event 5 runs after the CAN process has been executed.



## 10 SLM Drive Module XP-SLM

### 10.1 General

The XP-SLM drive module is a high-speed serial interface with SLM protocol (Speed Loop Motor protocol) for data transfer to and from SLM drives and actuators made by Control Techniques Drives Ltd. Via this module the XCN can execute positioning and position control on up to 4 independent NC axes.


It also provides 2 incremental encoder inputs and 2 interrupt sensing inputs.



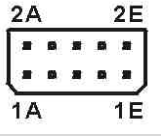
- Up to 4 drives
- RS 485 interface
- SLM protocol



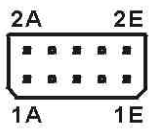
### 10.1.1 Displays and Connections

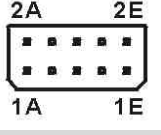
Displays			
LED		Colour	
	L1	red/green	Communication channel 1 permanent green: Communication ok permanent red: Communication error
	L2	red/green	Communication channel 2 permanent green: Communication ok permanent red: Communication error
	L3	red/green	Communication channel 3 permanent green: Communication ok permanent red: Communication error
	L4	red/green	Communication channel 4 permanent green: Communication ok permanent red: Communication error
Connections			
X11	D1	Drives	Type: har-link®, socket, 10-pin Manufacturer HARTING
X12	D2		
X13	D3		
X14	D4		
X15	E1	Encoder / interrupt	For assignment see below
X16	E2		

## 10.1.2 Assignment of Sockets on Module (Top View)

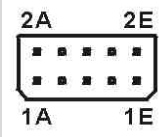
X11 D1 drives			
	Pin	Designation	Comment
	1A	COM1+	SLM protocol (RS485), drive 1
	2A	COM1-	SLM protocol (RS485), drive 1
	1B	COM2+	SLM protocol (RS485), drive 2
	2B	COM2-	SLM protocol (RS485), drive 2
	1C	COM3+	SLM protocol (RS485), drive 3
	2C	COM3-	SLM protocol (RS485), drive 3
	1D	HWEN1	Hardware Enable
	2D	nc	not connected
	1E	+24V-EXT	External power supply
	2E	GND-EXT	External power supply

As well as the COM1 connections, there are also COM2 and COM3 connections for two more drives. These are provided specially for MultiAx drives made by Control Techniques. But note that there is only one hardware enable for all axes.

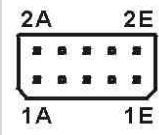
X12 D2 drive			
	Pin	Designation	Comment
	1A	COM2+	SLM protocol (RS485), drive 2
	2A	COM2-	SLM protocol (RS485), drive 2
	1B	nc	not connected
	2B	nc	not connected
	1C	nc	not connected
	2C	nc	not connected
	1D	HWEN2	Hardware Enable
	2D	nc	not connected
	1E	+24V-EXT	External power supply
	2E	GND-EXT	External power supply

X13 D3 drive			
	Pin	Designation	Comment
	1A	COM3+	SLM protocol (RS485), drive 3
	2A	COM3-	SLM protocol (RS485), drive 3
	1B	nc	not connected
	2B	nc	not connected
	1C	nc	not connected
	2C	nc	not connected
	1D	HWEN3	Hardware Enable
	2D	nc	not connected
	1E	+24V-EXT	External power supply
	2E	GND-EXT	External power supply

**X14 D4 drive**

	Pin	Designation	Comment
	1A	COM4+	SLM protocol (RS485), drive 4
	2A	COM4-	SLM protocol (RS485), drive 4
	1B	nc	not connected
	2B	nc	not connected
	1C	nc	not connected
	2C	nc	not connected
	1D	HWEN4	Hardware Enable
	2D	nc	not connected
	1E	+24V-EXT	External power supply
	2E	GND-EXT	External power supply

**X15 E1 and X16 E2 incremental encoders / interrupt sensing inputs**

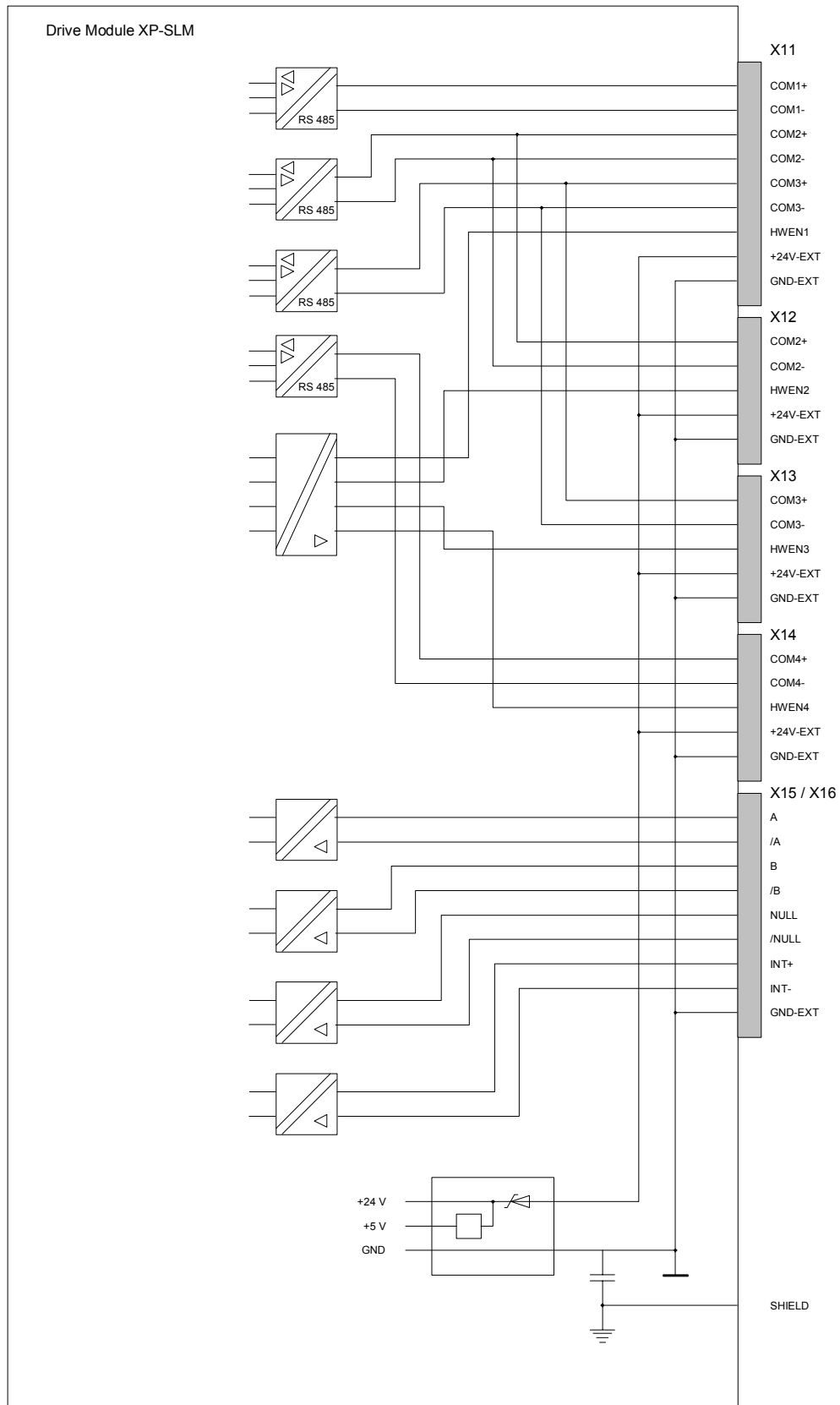
	Pin	Designation	Comment
	1A	A	Encoder input A
	2A	/A	Encoder input /A
	1B	B	Encoder input B
	2B	/B	Encoder input /B
	1C	NULL	Encoder input NULL
	2C	/NULL	Encoder input /NULL
	1D	nc	not connected
	2D	GND-EXT	Ground, external
	1E	INT+	Interrupt sensing input +
	2E	INT-	Interrupt sensing input -

## 10.1.3 Technical data for SLM drive module XP-SLM

<b>Basic data</b>	
Article No.	503 383 32-M
Number of controllable NC axes	up to 4 axes
<b>Electrical data</b>	
Power consumption of internal power supply	typ. 2 W
Isolation	yes, by optocouplers
Supply voltage external	+24V-EXT, GND-EXT
Voltage	24V DC $\pm$ 20% max. 5% residual ripple
Power consumption	1,3 W
Serial SLM interface	COM1+, COM1-; COM2+, COM2-; COM3+, COM3-; COM4+, COM4-;
Number	4
Physics	RS 485; 2.5 Mbaud
Protocol	Control Techniques ASIC CT2239-003
Hardware Enable	HWEN1; HWEN2; HWEN3; HWEN4
Number	4
Switching level	H level $\geq$ +24V-EXT - 0.5 V L level $\leq$ 1 V
Output current	0.5 A
Signal delay	<300 $\mu$ s (hardware)
Encoder inputs	A, /A, B, /B, NULL, /NULL
Number	2
Physics	RS422
Max. input frequency	2 MHz
Interrupt inputs	INT+, INT-
Number	2
Switching level	H level = +11 V to +30 V L level = -30 V to +5 V
Input current	min. H level (+11 V) $\geq$ 2 mA max. L level (+5 V) $\leq$ 2 mA typ. (+24 V) 8 mA max. (+30 V) $\leq$ 15 mA
Signal delay	<100 ns (hardware)
Triggering	flank triggering
<b>Accessories</b>	
XP-SLM-K1-3,5	3.5m connecting cable from the XP-SLM module to the drive (10-strand, Harting connector at one end) Order no.: 506 383 85



### 10.1.4 Interfaces



#### 10.1.4.1 External power supply

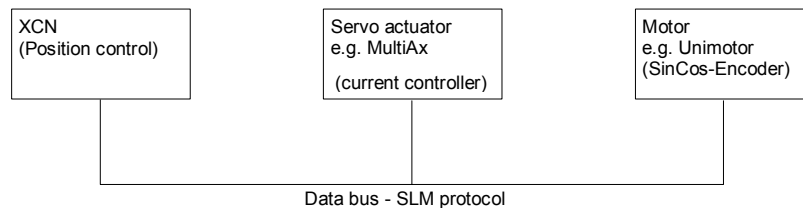
The external power supply can be fed through a 24V power pack or via the drive (e.g. MultiAx from Control Techniques). See also Wiring Example with MultiAx.

#### Serial SLM interface

An **SLM** interface always connects 3 stations:

- Motion controller (XCN with XP-SLM)
- Converter (e.g. MultiAx from Control Techniques)

#### 10.1.4.2 - Motor (e.g. Unimotor from Control Techniques)



#### Hardware enable

##### Encoder inputs

Two handwheels can be connected optionally to the encoder inputs.

(The inputs are currently only available as hardware, a software interface is in preparation.)

##### Interrupt inputs

Tracers can be connected to the interrupt inputs.

There are 2 interrupt inputs on the card.

(The software currently only processes the interrupt input of connector X15.)

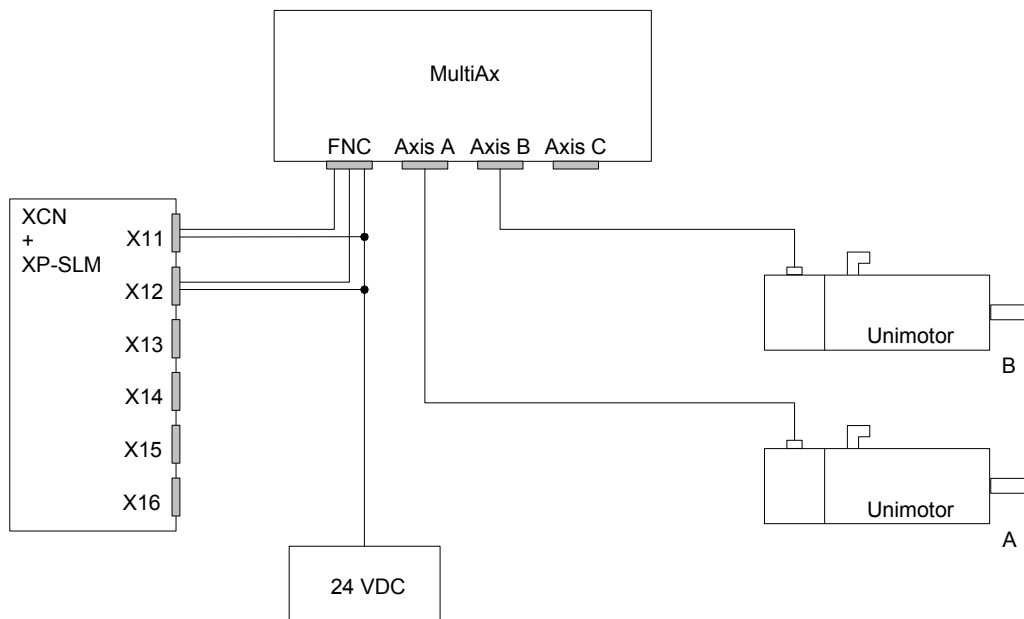
Note the following \$ functions and shared RAM variables for measuring functions:

\$ functions		
\$53	Abort motion with following error compensation through interrupt signal	
\$54	Delete remaining distance through interrupt signal	
Shared RAM		
cncMem.axSect[n].flgN2P.bRapAxStpAct	Rapid Axis Stop Active	\$53 measuring cycle active
cncMem.axSect[n].flgN2P.bRapMeasAct	Rapid Measurement Active	\$54 measuring cycle active
cncMem.axSect[n].flgN2P.bMeasValOk		Measured value is valid
cncMem.axSect[n].wrnN2P.lCurMeasPos	Current Measurement Position	Current position of drive at time of measuring (\$53 / \$54)

### 10.1.5 Application example

A MultiAx actuator and two Unimotor motors from Control Techniques are to be connected to an XCN with the SLM drive module XP-SLM. The drives are connected to the controller via connectors X1 and X2.

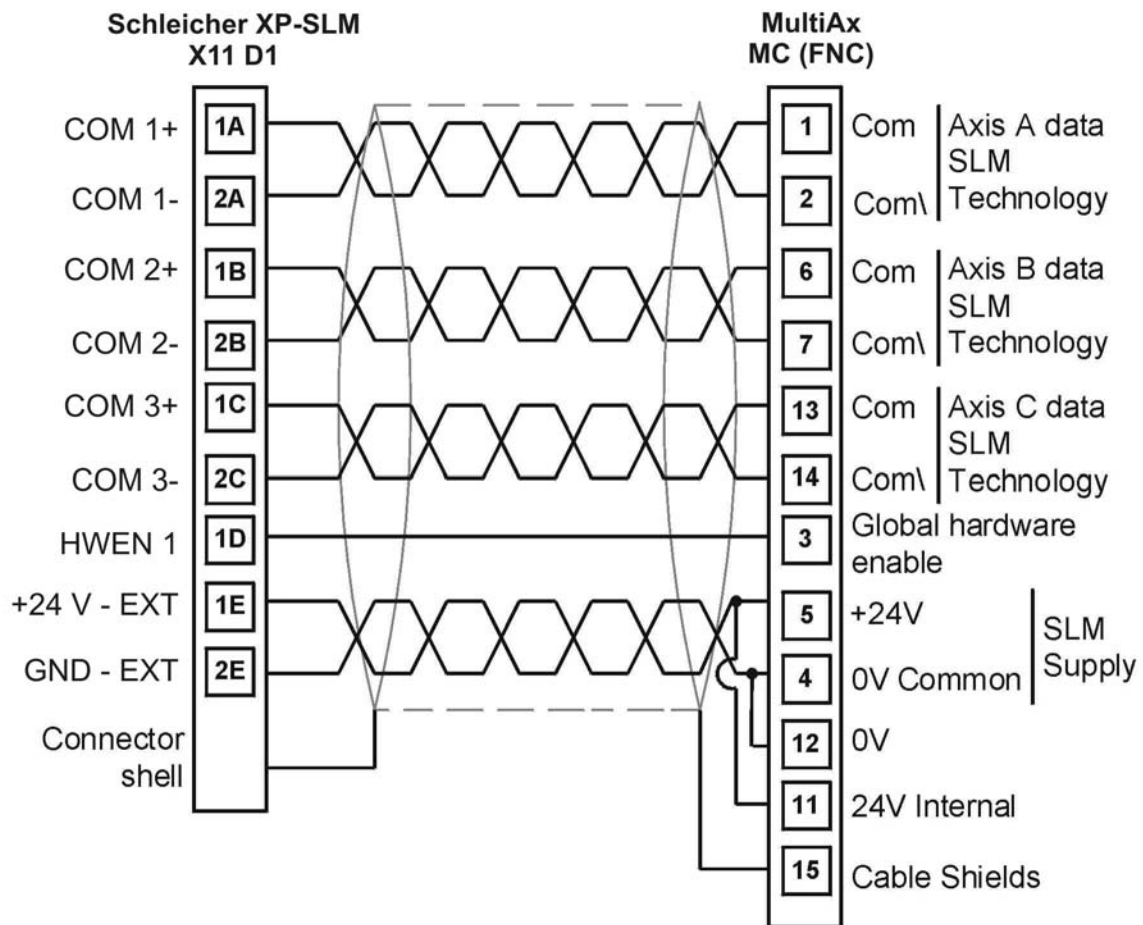
The diagram shows the basic wiring of the drives (without power section).





## 10.1.6 Wiring Example with MultiAx

## 10.1.6.1 SLM drive module XP-SLM on MultiAx (15-pin high density submin)



This type of wiring is recommended only if the modification with the electronic fuse is implemented in MultiAx.

Without this protective measure the wiring is defective and could damage the MultiAx power supply.

If this wiring is used without the modification the drive guarantee is invalidated.

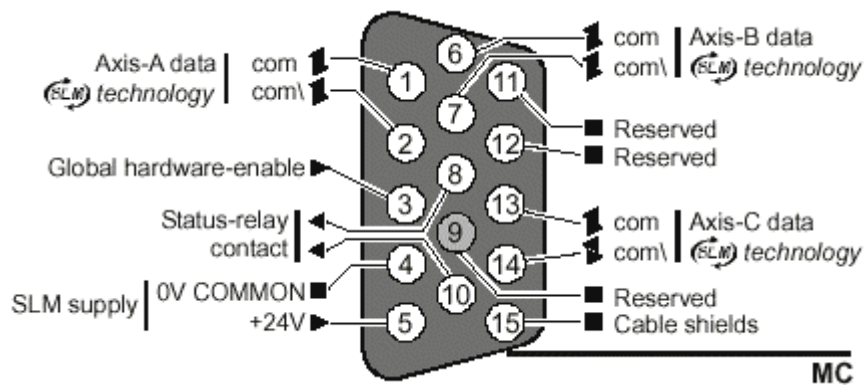


### 10.1.6.2 Notes on MultiAx



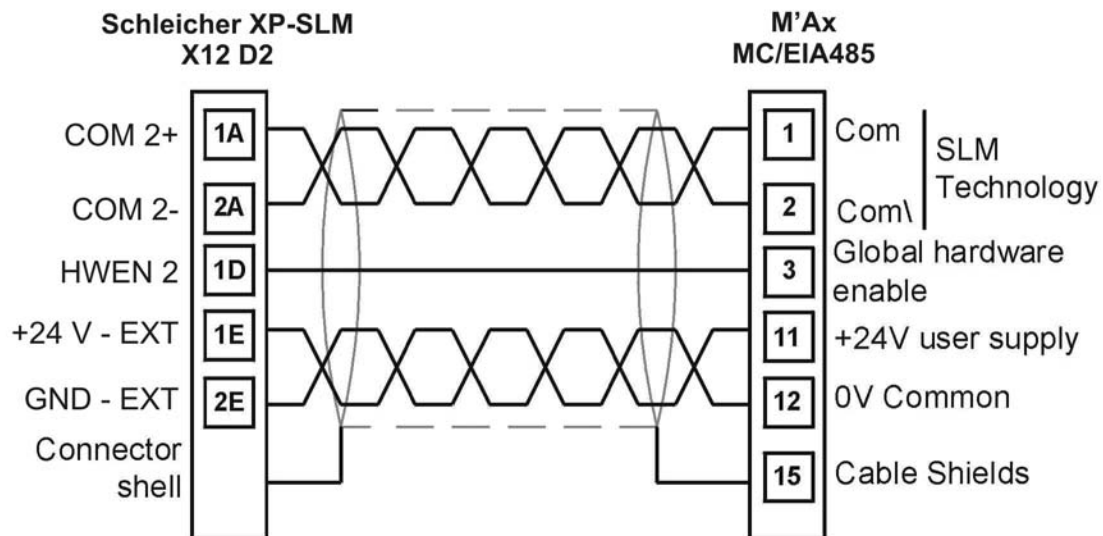
If position data is required when the drive power supply (AC) is switched off, you will have to connect an external 24V power supply to MultiAx (MC (FNC) Pin 5 and 4).

Pin assignment of the MC (FNC) connections:

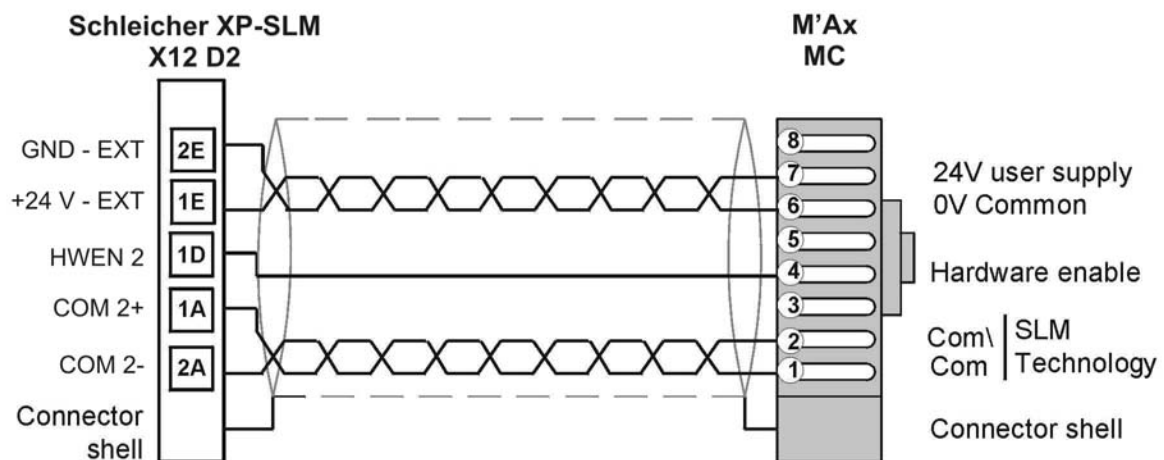


## 10.1.7 Wiring example with M'Ax

## 10.1.7.1 SLM Drive Module XP-SLM on M'Ax (15-pin high density submin)



## 10.1.7.2 SLM Drive Module XP-SLM on M'Ax (RJ45)

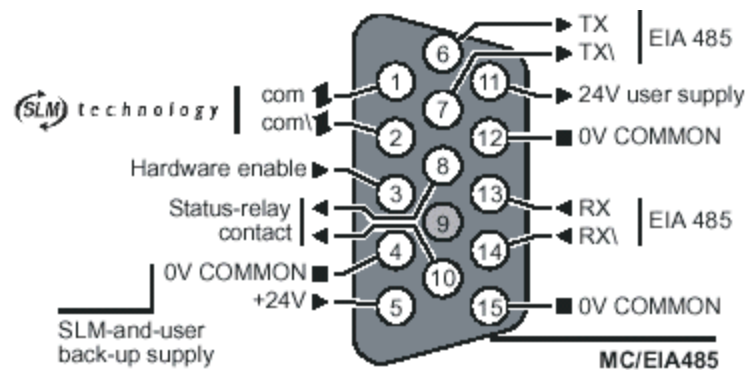


### 10.1.7.3 Notes on M'Ax

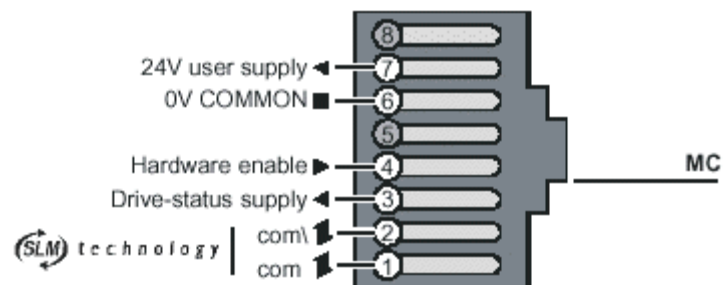


If position data is required when the drive power supply (AC) is switched off, you will have to connect an external 24V power supply to M'Ax (MC/EIA485 Pin 5 and 4).

Pin assignment of the MC/EIA485 connections:



Pin assignment of the MC/EIA485 connections:



## 11 SERCOS Module XP-SRC


The SERCOS XP-RC drive module provides a SERCOS ring on which up to 8 independent SERCOS drives can be operated via optical fiber.



- SERCOS master acc. to IEC1491
- 8 axes
- Optical fiber connection to IEC 874-2



### 11.1 Displays and Connections

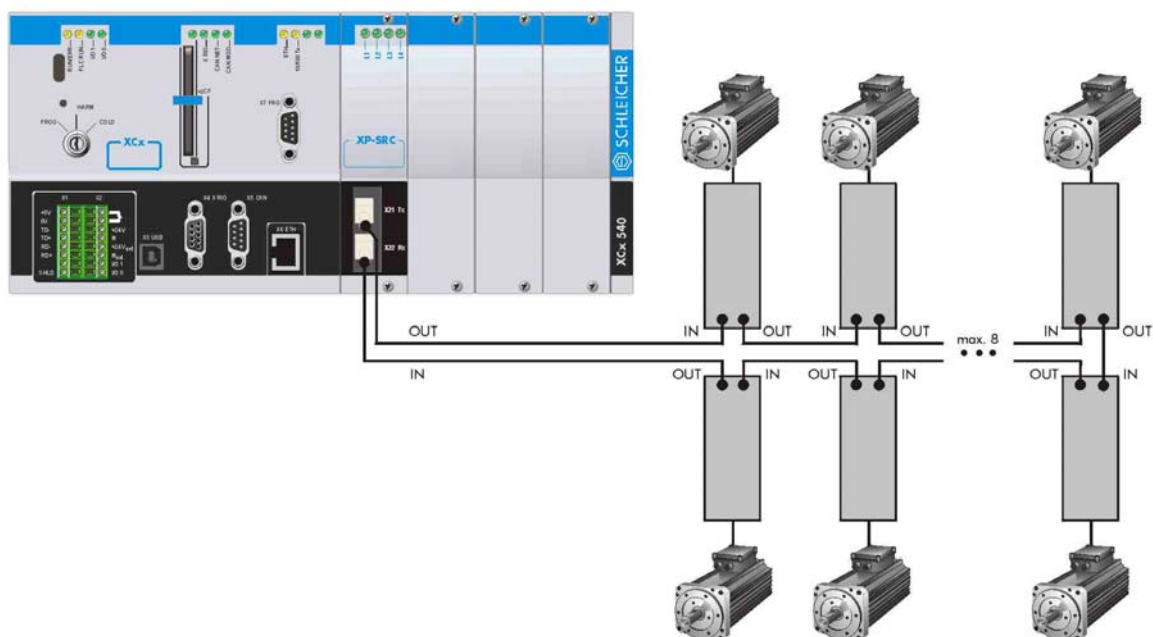
	Displays		
	Designation	Colour	
	L1 SERCOS PHASE 1	green/red	SERCOS startup phase: green on - Phase 1 complete red on - Phase 1 not complete
	L2 SERCOS PHASE 2	green/red	SERCOS startup phase: green on - Phase 2 complete red on - Phase 2 not complete
	L3 SERCOS PHASE 3	green/red	SERCOS startup phase: green on - Phase 3 complete red on - Phase 3 not complete
	L4 SERCOS PHASE 4	green/red	SERCOS startup phase: green on - Phase 4 complete red on - Phase 4 not complete
	All LEDs red: Module in Phase 0		
	All LEDs off: No axis assigned		
	Connections		
	X21 Tx	SERCOS ring output	F-SMA screw connection for optical fiber
	X22 Rx	SERCOS ring input	

## 11.2 Technical Data

Basic data	
Article No.	503 383 31-M
Number of controllable NC axes	Up to 8 NC axes (1 SERCOS ring)
SERCOS interface	
Interface	SERCOS Master to IEC 1491
Number	1 SERCOS ring
Physics	Plastic optical fiber, 2 / 4 / 8 / 16 MBaud
Controller	SERCON 816
Cycle time	1 ms ...30 ms
Connection system	
SERCOS optical fibre connection IN / OUT	F-SMA screw connections acc. to IEC 874-2

## 11.3 Application Example

You can connect up to 8 SERCOS drives to an XCN with the SERCOS drive module XP-SRC. The XP-SRC module is the master and the drives are the slaves. The optical fiber is connected as a ring, i.e. each output is connected to an input.





### Establishing Communication via the SERCOS Ring

Communication on the SERCOS ring is established in 5 phases (phases 0..4).

Establishment of communication is always restarted when

- the controller is switched on
- the reset button (on the XCN) is operated,
- the drive parameters are changed, or
- an error (e.g. cable break) is found on the SERCOS ring.

Several seconds may pass before communication phase 4 is achieved. You can follow the establishment of communication using the LED displays.



## 12 PROFIBUS-DP Modules


### 12.1 PROFIBUS-DP Master XF-DPM



- One PROFIBUS-DP master field bus interface
- Correspond to IEC 61158 Type 3
- up to 12 MBaud data transmission rate



## 12.2 Displays and Connections XF-DPM



Display				
	LED	Des.	Colour Condition	Description
	L1	BF	red	No bus connection (bus fail)
			green	Bus connection active
			yellow flashing	Bus connection active, however no data exchange.
	L2			Not used
	L3	SYS	yellow flicking	Hardware error
			green flashing	Ready connecting to bus, however no (not as yet) bus connection active.
			green flicking	Bus connection error
			green	Bus connection active
	L4	COM	yellow	Cyclic data exchange active
			red	Bus connection error
Connector X61				
	Pin	Designation		
	1	nc	not connected	
	2	nc	not connected	
	3	RxD/TxD-P	Receive/Transmit Data P	
	4	CNTR-P	Repeater Control Signal	
	5	DGND	Data Ground (0 V)	
	6	VP	Voltage Plus (+5 V)	
	7	nc	not connected	
	8	RxD/TxD-N	Receive/Transmit Data N	
	9	nc	not connected	

## 12.3 Technical Data XF-DPM


Basic data	
Interface	PROFIBUS-DP Master
Data transmission rate	up to 12 MBaud
Connection system	Subminiature type D, 9-pin, socket connector
Ambient temperature working range	0°C to 45°C

## 12.4 PROFIBUS-DP Slave XF-DPS



- One PROFIBUS-DP slave field bus interface
- Correspond to IEC 61158 Type 3
- up to 12 MBaud data transmission rate

## 12.5 Displays and Connections XF-DPS

	Display			
	LED	Des.	Colour Condition	Description
	L1	BF	red	No bus connection (bus fail) Field bus cable break or the master does not operate the bus (any more).
			green	Bus connection active
			yellow flashing	Bus connection active, however no data exchange.
	L2	DIA	yellow flashing	Static diagnosis active
	L3	SYS	yellow flicking	Hardware error
			green flashing	Ready connecting to bus, however no (not as yet) bus connection active.
			green flicking	Bus connection error
			green	Bus connection active
	L4	COM	yellow	Cyclic data exchange active
	Connector X51			
	Pin	Designation		
	1	nc	not connected	
	2	nc	not connected	
	3	RxD/TxD-P	Receive/Transmit Data P	
	4	CNTR-P	Repeater Control Signal	
	5	DGND	Data Ground (0 V)	
	6	VP	Voltage Plus (+5 V)	
	7	nc	not connected	
	8	RxD/TxD-N	Receive/Transmit Data N	
	9	nc	not connected	

## 12.6 Technical Data XF-DPS

Basic data	
Interface	PROFIBUS-DP Slave
Data transmission rate	up to 12 Mbaud
Connection system	Subminiature type D, 9-pin, socket connector
Ambient temperature working range	0°C to 45°C

## 12.7 PROFIBUS-DP Configuration Data

### File ProConOS.INI section DPxx

```
; DP parameter Definitionen:
[DP00]                ; e.g. card# 0
; any description
DESCRIPTION           = Profibus DPS #0 (XCx5 series)
BUS_ADDR              = 2      ; 0..126
MASTER_FCONF         = 1      ; the master force slaves configuration
                        ; 1 = yes (default)/ 0 = no
```

#### MASTER\_FCONF parameter:

- 0: the master configuration data will be compared with the configuration stored in the file DPSconf0x.ini
- 1: the master forces the slave configuration (the master configuration data are accepted)

#### Note

The actual XDP configuration is stored in the file  
/ram0/OS/LOG/Sysinfo.txt.

### Standard Module Configuration

#### DP Slave: IO area: 0..72

No.	Offset	Number	Byte /Word	Input / Output	Consistency
0	0	4	B	I	-
1	0	4	B	O	-
2	4	4	B	I	-
3	4	4	B	O	-
4	8	16	B	I	c
5	8	16	B	O	c
6	24	16	B	I	c
7	24	16	B	O	c
8	40	16	W	I	c
9	40	16	W	O	c



### DP Slave Configuration module structure

modType	modLen
0 - input byte without consistency	0 - 1 Byte/Word
1 - input word without consistency	1 - 2 Bytes/Words
2 - output byte without consistency	2 - 3 Bytes/Words
3 - output word without consistency	3 - 4 Bytes/Words
4 - input byte with consistency	4 - 8 Bytes/Words
5 - input word with consistency	5 - 12 Bytes/Words
6 - output byte with consistency	6 - 16 Bytes/Words
7 - output word with consistency	7 - 20 Bytes/Words
8 - dummy place	8 - 32 Bytes/Words
	9 - 64 Bytes/Words

### DPSconf0x.ini (default)

```
[DP00]
NumberOfModules = 10
[mod00]
; 4 bytes input
modType = 0
modLen = 3
[mod01]
; 4 bytes output
modType = 2
modLen = 3
[mod02]
; 4 bytes input
modType = 0
modLen = 3
[mod03]
; 4 bytes output
modType = 2
modLen = 3
[mod04]
; 16 bytes input with consistence
modType = 4
modLen = 6
[mod05]
; 16 bytes output with consistence
modType = 6
modLen = 6
[mod06]
; 16 bytes input with consistence
modType = 4
modLen = 6
[mod07]
; 16 bytes output with consistence
modType = 6
modLen = 6
[mod08]
; 16 words input with consistence
modType = 5
modLen = 6
[mod09]
; 16 words output with consistence
modType = 7
modLen = 6
```

## 13 Error messages

### 13.1 Structure of Error Messages

Error messages in the active error buffer and the error log book have error numbers and additional information.

#### Structure of Error Messages

Error messages are made up of a hexadecimal error number, an error text and up to three optional parameters, which can be shown in the error text.

Error No.(hexadecimal) / Par. 1 / Par. 2 / Par. 3. / Error text

Errors to which no error text is assigned are shown as follows:

Fehler %ErrX%, Parameter %ParX% (Par1=%Par1%, Par2=%Par2%, Par3=%Par3%)  
 [Error %ErrX%, Parameter %ParX% (Par1=%Par1%, Par2=%Par2%, Par3=%Par3%)]

With this nomenclature the following numerical values can be shown in the error message.

%ErrX%	Error number (32 bit hexadecimal)
%ParX%	Parameter value (32 Bit, hexadecimal), contains parameters 1 to 3: D31...D24 -- Parameter 1 D23...D16 -- Parameter 2 D15...D0 -- Parameter 3
%Par%	Parameter value (32 Bit, decimal), contains parameters 1 to 3: D31...D24 -- Parameter 1 D23...D16 -- Parameter 2 D15...D0 -- Parameter 3
%Par1%	Parameter value 1 (8-bit, decimal, signed)
%Par2%	Parameter value 2 (8-bit, decimal, signed)
%Par3%	Parameter value 3 (16-bit, decimal, signed)

#### Example (representation in error logbook)

Date	Time	Error text	Error number	Error parameter	Status
04.03.03	08:04:06	Fehler 0x04100006, Parameter 0x00640001 (Par1=0, Par2=100, Par3=1) [Error 0x04100006, Parameter 0x00640001 (Par1=0, Par2=100, Par3=1)]	0x04100006	0x00640001	O
04.03.03	08:04:04	Power-On	0x01100006	0x00000000	O
04.03.03	08:04:01	NC nicht betriebsbereit [NC not ready]	0x02000000	0x00000000	A

The status column in the error logbook and the active error buffer can have the following values:

A	(Active) Active error message in active error buffer
L	(Locked) Locked error message in active error buffer
I	(In) Error occurred and error message entered in active error buffer
O	(Out) Error rectified and removed from active error buffer
N	(New) Transitional status: error message entered in active error buffer but not yet in error logbook
N A	(Not Active) Transitional status: error message shortly before status "Out"



### Structure of error numbers

The error number is a 32-bit number. The high word contains the grouping, the low word the consecutive error number.

### Grouping of error numbers

0x0100nnnn	PLC runtime error
0x0110nnnn	General PLC error messages
0x0120nnnn	XRIO error
0x0200nnnn	General CNC error messages
0x0210nnnn	Subsystem-specific CNC error
0x0211nnnn	Decoder error
0x0212nnnn	Record insertion error
0x0213nnnn	Error in tool nose radius compensation
0x0214nnnn	Error in interpolation preparation
0x0220nnnn	Axis-specific error
0x0221nnnn	SERCOS diagnosis status (only on ProNumeric / PN-MIC)
0x0222nnnn	SERCOS status class 1 (only on ProNumeric / PN-MIC)
0x0223nnnn	Manufacturer-specific SERCOS error (only on ProNumeric / PN-MIC)
0x0224nnnn	Manufacturer-specific SERCOS error text (only on ProNumeric / PN-MIC)
0x0230nnnn	Drive-specific error (only on XCN)
0x0300nnnn	CAN-specific operating system error
0x0301nnnn	CAN driver error
0x0400nnnn to 0x04FFnnnn	User-defined errors, can be entered in the error logbook by the PUT_ERROR2 function block

Documentation note: Parts of the table are also in the following sections.



## 13.2 PLC Error Messages

0x0100nnnn	PLC runtime error
0x0110nnnn	General PLC error messages
0x0120nnnn	XRIO error

### 13.2.1 0x0100 PLC runtime error

Errors with this classification are ProConOs runtime errors that are merely passed on by the ProNumeric runtime system. An error number consists of a part describing the error type – error, information, warning – and the actual error number itself.

0000: Error

4000: Flag for optional information

8000: Flag for a warning

C000 : Flag for optional information + warning

Example: 0x0100 4051: Internal error: Root task error number n

This error is also marked as information.

#### 13.2.1.1 Group 0001 Task communication

Errors in this error group can occur at any time when ProConOs is running.

Error no.	Error text
0x01004021	Internal error : Memory task error number n
0x01004031	Internal error : Communication task error number n
0x01004041	Internal error : Debug task error number n
0x01004051	Internal error : Root task error number n
0x01000061	Internal error : Remote control disabled

#### 13.2.1.2 Group 0002 Code generation

Errors in this error group can occur when projects, programs or parts of programs are sent to the controller.

Additional parameters such as %u and n are replaced when the error is output.

Error no.	Error text
0x01004012	Internal Error : No code for module n
0x01004022	Memory Error! WN %u: Not enough memory in PLC! -> Shrink project
0x01004032	Error! WN %u: Instruction not implemented or incorrect data type! ->Don't use this instruction Don't use this instruction in conjunction with this datatype
0x01004042	Error! WN %u: Code exceeds 64K limit! -> shrink POU
0x01004052	Error WN %u: Missing Label -> insert missing label
0x01004062	Internal Error! WN %u: Error passing on parameter! -> check parameter passing
0x01004072	Error! "WN %u: Nesting level of parenthesis exceeded -> use less brackets
0x01004082	Error! WN %u: Nesting error! -> check brackets
0x01004092	Internal Error! WN %u: Undefined parameter! -> tried to access a parameter that wasn't passed
0x010040A2	Error! Automatic datatype conversion of operand not possible! ->use explicit type conversion
0x010040B2	Error! Automatic datatype conversion of accumulator not possible ! ->use explicit type conversion



0x010040C2	Error! WN %u: Maximum number of errors reached, code generation stopped ->clear previously occurred errors
0x010040D2	Error! Error(s) occurred in POU %u during code generation !
0x010040E2	Internal Error! WN %u: Error passing on parameter! ->check parameter passing
0x010040F2	Internal Error! WN %u: Operand not implemented or area exceeded! -> check used operand operand number
0x01004102	Error! WN %u: Illegal data type of accumulator (expecting ANYNUM)
0x01004112	Error! WN %u: Illegal data type of accumulator (expecting ANYBIT)
0x01004122	Internal Error : WN %u: Data type of accumulator not valid
0x01004132	Internal Error : WN %u: Selected index register not valid
0x01004142	Internal Error! WN %u: Illegal data type of accumulator
0x01004152	Internal Error : WN n: Not all accumulator values that were pushed are popped
0x01004162	Internal Error : WN n: Maximum number of accumulator pushes reached
0x01004172	Internal Error : WN n: Tried to pop an accumulator which was not pushed
0x01000182	Error! Versions of used MSFC doesn't match -> ask distributor for matching versions
0x01000192	Error! No MSFC data to run project! PLC resetted!

### 13.2.1.3 Group 0003 Memory management

Errors in this error group can occur when projects, programs or parts of programs are sent to the controller.

Error no.	Error text
0x01004013	Internal Error: POU '%s' invalid
0x01004023	Internal Error: POU '%s' has unknown POU type
0x01004033	Memory Error! Not enough memory on PLC for POU '%s' during insertion! -> Shrink project
0x01004043	Memory Error! POU '%s' exceeds 64K module size during insertion! -> shrink POU
0x0100C053	Warning : POU %s uses more than 80 percent of POU memory
0x01008063	Error ! Cannot insert a POU because there is no project known
0x01008073	Warning : Project uses more than 80 percent of program memory
0x01004083	Error ! Tried to insert the POU that does not belong to this project on PLC
0x01000093	Internal Error : Error in memory management
0x0100C0A3	Warning : Actual project size : %u kByte
0x010000B3	Internal Error : inserting POU
0x010040E3	Internal Error : Invalid POU type for POU '%s'
0x010000F3	Internal Error : Memory reorganization not possible, PLC stopped!
0x0100C103	Warning : Actual POU size : %u byte
0x0100C113	Internal Error : PG '%u' is defined more than once!
0x0100C123	Internal Error : SPG '%u' is defined more than once!
0x01004133	Error : Memory error for initialized data of POU '%s'!
0x01000143	Error : Acception of retain CRC failed! possible reasons: actual project hasn't any retain data actual project is 'old style' without retain CRC PLC isn't in STOP mode

0x0100C153	Internal Error : FB '%u' is defined more than once!
0x010001B3	Internal Error : Not all POU sent!
0x010001C3	Internal Error : No program memory defined!
0x010041D3	Internal Error : Invalid FB number %u!
0x010041E3	Internal Error : Invalid PG number %u!
0x010041F3	Internal Error : Invalid SPG number %u!

#### 13.2.1.4 Group 0004 PLC user error

Errors in this error group can occur when a user program is executed.

Error no.	Error text
0x01004014	Internal Error : Invalid function or function block n!
0x01004024	Error : Invalid firmware function or function block n!
0x01004034	Internal Error : Invalid program n!
0x01008044	Internal Error : Invalid change of mode!
0x01000054	Internal Error : Unknown system mode! PLC stopped!
0x01000074	Error : Division by 0! PLC stopped!
0x010080A4	Warning : Function PLC_STOP executed, PLC stopped!
0x010000B4	Error : Bus error! PLC stopped! -> This error appears on systems with memory management units (MMU) when it was tried to access memory which isn't available
0x010000C4	Error : Stack overflow! PLC stopped! -> increase stack size
0x010000D4	Internal Error : Not implemented CPU instruction! PLC stopped! -> Appears on M68-systems.
0x010000E4	Internal Error : Non initialized interrupt! PLC stopped! -> Appears on M68-systems.
0x010000F4	Internal Error : Wrong interrupt, PLC stopped! -> Appears on M68-systems.
0x01004104	Error : System Error in modul xx (belongs to previous) system error)
0x01004114	Error : System Error at line xx (belongs to previous) modul no)
0x01000124	Error : Error during indirect variable access!
0x01004134	Watchdog exceeded in task %02u
0x01000144	Internal Error : Error in task configuration
0x01000154	Error : Error in I/O configuration
0x01008164	Warning : Warmstart not possible! Coldstart performed.
0x01008174	Warning : No retain area to hold startup information.
0x01004184	Error : CPU is occupied with xx percent, PLC stopped!
0x01000194	Error : Initializing I/O driver failed!
0x010001A4	Internal error : Breakpoint unexpected, PLC stopped!
0x010001B4	Error : Unknown I/O driver
0x010041C4	Internal error : Watchdog exceeded in system task ID = %02u
0x010001D4	Internal Error : Error in data configuration!
0x010001E4	Internal Error : Error in retain data configuration!
0x01000204	Internal Error : Critical error in floating point unit!
0x01000214	Internal Error : Fatal error!
0x01000224	Error : String error! PLC stopped! -> Appears on wrong string accessing
0x01000234	Error : String error!
0x01000244	Error : String error! Output string is too short



0x01000254	Error : String error! Input string is too short
0x01000264	Error : String error! Invalid input parameter 'p' (position in string)
0x01000274	Error : String error! The second string is identical to the output string
0x01000284	Error : String error! Invalid string comparison
0x01000294	Error : String error! Datatype not supported for the string conversion
0x010002A4	Error : String error! Error in the format string
0x010002B4	Error : String error! Invalid input value for this format string -> e. g.: value = 0 + format = %c
0x010002C4	Error : String error! Error during string conversion

### 13.2.1.5 Group 0005 Debug

Errors in this error group can occur while a user program is being debugged.

Error no.	Error text
0x01000025	Error : Operand not implemented or area exceeded!
0x01000035	Error : wrong or missing trigger condition
0x01000045	Error : no memory available
0x01000055	Error : datatype not supported
0x01004065	Internal Error : Setting breakpoint at function or function block n not possible! -> Could only be possible during single stepping
0x0100C075	Error : Cannot set breakpoint at this line (WN n)! -> set breakpoint one line before or after
0x01008095	Error : write access is not allowed for this variable
0x010000A5	Error : error in recorded logic analyzer data
0x010000B5	Error : Too many powerflow addresses!
0x010000C5	Error : Powerflow not possible! -> previous error(s) shows the reason
0x010080D5	Internal Error : Error in powerflow list (invalid entries)!
0x010000F5	Internal Error : Invalid type of list!
0x01004115	Error : No debug information for POU n
0x01004125	Error : No debug information for WN n
0x01000135	Error : This debug function is not available when the program is in (E)EPROM
0x01004145	Internal Error : Missing code for POU n!
0x010041A5	Internal Error : Breakpoint in PG n not possible!
0x010041B5	Internal Error : Breakpoint in SPG n not possible!
0x010041D5	Internal Error : No force list available!
0x01000205	Internal Error : Debug task not installed! -> chosen function isn't possible with this PLC-system

### 13.2.1.6 Group 0006 Exchange of system data

Errors in this error group can occur when system data is being exchanged (PLC tasks, I/O configuration, boot project, PDD, ...).

Exception: 0x010000B6: Memory Error : Out of memory (dynamic memory)!

This error can occur at any time when ProConOs is running.

Error no.	Error text
0x01008016	Internal Error : Invalid type of module!
0x01000026	Error : Accessing Initialization file failed
0x01000036	Internal Error : Too many data requested!
0x01000046	Internal Error : Less data send!
0x01000056	Internal Error : Unexpected online service order!
0x01004066	Error : Installation of IO-Driver %u failed
0x01004076	Error : Instantiation of IO-Driver %u failed
0x01000086	Internal Error : Error during accessing file!
0x01004096	Error : Instantiation of Instance @%u failed
0x010000A6	Internal error : Data group exceeds limit of 64K!
0x010000B6	Memory Error : Out of memory (dynamic memory)!
0x010000C6	Internal Error : Incorrect refresh of retain data!
0x010000D6	Error : Device for holding retain data not available!
0x010000E6	Internal Error : Invalid group for retain data!
0x010040F6	Internal Error : (PDD) Definition of data type (%u) was made twice!
0x01004106	Internal Error : (PDD) Data type (subtype) definition (%u) not found!
0x01004116	Internal Error : (PDD) Unknown data type (%u)!
0x01004126	Internal Error : (PDD) Symbol not found (process variable : %u)!
0x01000136	Internal Error : (PDD) Symbol exists several times!
0x01004146	Internal Error : (PDD) Operand area exceeded (process variable : %u)!
0x01000156	Internal Error : (PDD) No memory for fast symbol access (information only)!
0x01004166	Error : IO-Driver %d not announced
0x01000176	Error : zip device is not installed
0x01000186	Error : file device is not installed

### 13.2.1.7 Group 0007 IO driver errors

Errors in this error group can occur when IO driver functions are used (init, open, close, read, write).

Error no.	Error text
0x01004017	Error : Board No. %u not instantiated
0x01004027	Error : Board No. %u not allowed
0x01004037	Error : Input Group (VAR_ADR := %u) doesn't fit
0x01004047	Error : Output Group (VAR_ADR := %u) doesn't fit
0x01004057	Error : Board No. %u not found
0x01004067	Error : Error reading inputs (Board No. %u)
0x01004077	Error : Error writing outputs (Board No. %u)
0x01004087	Internal Error : Can't create Semaphore (Board No. %u)



0x01004097	Error : Invalid memory size (Board No. %u)
0x010040A7	Error : Invalid board address (Board No. %u)
0x01000307	Error : Error occured in IO-Driver 'CIF_DRV'
0x01004317	Error : Can't switch Host State (Board No. %u)
0x01004327	Error : CIF Error Number: %u
	Folgende Fehler sind nur für CANopen relevant
0x01000407	Error : Error occured in IO-Driver 'CANOPEN'
0x01004417	Error : CAN Error Number: %d

### 13.2.2 0x0110 General PLC errors

Error no.	Error text	Cause / Remedy
0x01100000	Inkompatible Koppelspeicher-Struktur (Version PLC/CNC = x, Version MULTIPROG = y) [Incompatible shared RAM structure (Version PLC/CNC = x, Version MULTIPROG = y)]	The shared RAM structure in the PLC programming system differs from the one in the runtime system. Update runtime system or announce other firmware libraries.
0x01100001	Inkompatible Hardware-Konfiguration [Incompatible hardware configuration]	The current hardware configuration does not match the saved configuration. Hardware reset required.
0x01100002	PLC-Option PSOn nicht freigegeben [PLC option PSO<n> not enabled]	Option not enabled for this control system.
0x01100003	Not-Aus !!! [Emergency stop]	Emergency stop has been triggered and must be acknowledged.
0x01100004	PN-MIC n asynchron [PN-MIC n asynchron]	Only on ProNumeric / PN-MIC: SYNC line not OK.
0x01100005	Clear Error-Logbook	Note: error logbook deleted.
0x01100006	Power-ON	Note: power switched on.
0x01100007	Falsches Datum und falsche Uhrzeit nach Ausfall der Batterie [Incorrect time and date after battery failure]	Possible causes: - Battery discharged - CPU PCB card removed from controller  Time and date have been reset to the preparation time and date of the operating system version.
0x01100008	Datenverlust im SRAM-Speicher nach Ausfall der Batterie [Data loss in SRAM memory after battery failure]	Possible causes: - Battery discharged - CPU card previously removed from controller  SRAM memory has been completely deleted.
0x01100009	PLC Start	Note: PLC started
0x0110000A	PLC Stop	Note: PLC stopped



## 13.2.3 0x0120 XRIO errors

These error messages apply only to the XCx.

The XRIO connection number is indicated in the first parameter.

Error no.	Error text	Cause / Remedy
0x01200001	XRIO: Fehler während der Initialisierung (XRIO Verbindung n, Status=m) [XRIO: Error during initialization (XRIO connection n, status=m)]	
0x01200002	XRIO: Unbekannte Modul-ID (XRIO Verbindung n, Modul-ID=m) [XRIO: Unknown module ID (XRIO connection n, module ID=m)]	
0x01200003	XRIO: Zu viele Module werden benutzt (XRIO Verbindung n, Anzahl=m) [XRIO: Too many modules used (XRIO connection n, number=m)]	
0x01200004	XRIO: Maximale Anzahl von Bytes überschritten (XRIO Verbindung n, Anzahl=m) [XRIO: Max. number of bytes exceeded (XRIO connection n, number=m)]	
0x01200005	XRIO: Steckplatzlisten-Fehler (XRIO Verbindung n) [XRIO: Module location list error (XRIO connection n)]	
0x01200006	XRIO: Fehler während der Übertragung (XRIO Verbindung n, Status=m) [XRIO: error during transmission (XRIO connection n, status=m)]	



### 13.3 CNC Error Messages

0x0200nnnn	General CNC error message
0x0210nnnn	Subsystem-specific CNC error
0x0211nnnn	Decoder error
0x0212nnnn	Record insertion error
0x0213nnnn	Error in tool nose radius compensation
0x0214nnnn	Error in interpolation preparation
0x0220nnnn	Axis-specific error
0x0221nnnn	SERCOS diagnosis status (only on ProNumeric / PN-MIC)
0x0222nnnn	SERCOS status class 1 (only on ProNumeric / PN-MIC)
0x0223nnnn	Manufacturer-specific SERCOS error (only on ProNumeric / PN-MIC)
0x0224nnnn	Manufacturer-specific SERCOS error text (only on ProNumeric / PN-MIC)
0x0230nnnn	Drive-specific error (only on XCN)



## 13.3.1 0x0200 General CNC errors

Error no.	Error text	Cause / Remedy
0x02000000	NC nicht betriebsbereit [NC not ready]	Possible causes: - PLC not running - Emergency stop activated - Drive module not recognized or initialization failed - Measuring circuit error (on SERCOS module)
0x02000001	IPO-Zykluszeit zu klein [Interpolation cycle time too short]	Set interpolation time not sufficient. This time is set in the controller's default settings (Schleicher Dialog - Startup - Default settings)
0x02000002	CNC-Option PSO<n> nicht freigegeben [CNC option PSO<n> not enabled]	Option not enabled. Please request enabling code.
0x02000003	Laufzeitüberwachung PN-MIC (Karte n) [Runtime monitoring PN-MIC (card n)]	Only on ProNumeric / PN-MIC: Interpolation runtime too long, so axis setpoint for drive card (PN-MIC) is provided too late. If increasing the interpolation scan rate does not help there is a serious system error! Please contact Schleicher Service.
0x02000004	Kein Speicher für CNC-Systemdaten (Fehler-Nr. n) [No memory for CNC system data (Error no. n)]	A system error has occurred while generating CNC data fields. Please contact Schleicher Service.
0x02000005	CNC-System auf falscher (SPS-) Hardware [CNC system on incorrect (PLC) hardware]	This error occurs if you attempt to run a CNC operating system on a controller CPU that is only configured for a PLC operating system.
0x02000006	Systemparameter inkonsistent [System parameters inconsistent]	This error message occurs when the controller is powered up. System parameters (Q parameters) can become inconsistent if the number of system parameters has changed. This can occur, for example, if you change the number of CNC axes or if an operating system update has been carried out.  The system parameters must be updated if the CNC is to run properly.  When Schleicher Dialog starts this error is detected and a dialog window opens for you to instruct the controller to update the system parameters.
0x02000007	Kein Speicher für Remote-Page [No memory for remote page]	This error message can occur when the controller is powered up, for example if insufficient memory has been provided for the drives.  Please contact Schleicher Service.
0x02000008	MC-Zykluszeit zu klein [MC cycle time too short]	Increase MC cycle time (drive parameter <i>CYCLE_TIME</i> ).



### 13.3.2 0x0210 Subsystem-specific CNC errors

In this error group the subsystem number is indicated in the additional parameter.

Error no.	Error text	Cause / Remedy
0x02100000	Einlesefreigabe fehlt (System n) [No input enable (System n)]	This message indicates that input enable is missing (no acknowledgement of a flag indicating an NC function change: <i>cncMem.sysSect[n].flgN2P.</i> <i>bMFct1Mod,</i> <i>bMFct2Mod,</i> <i>bMFct3Mod,</i> <i>NcBlkMod,</i> <i>TFctMod,</i> <i>SFctMod</i> )
0x02100001	Arbeitsraumbegrenzung (System n) [Working area limits (System n)]	
0x02100002	Unerreichbare Position (System n) [Unreachable position (System n)]	
0x02100003	Gewindebohren aktiv (System n) [Tapping active (System n)]	
0x02100004	Falsche Transformation-Schnittstelle (System n) [Wrong transformation interface (System n)]	This error message appears if the interface version of the loaded transformation does not match the interface version of the installed operating system version.  Please contact Schleicher Service.
0x02100005	Falsche Anzahl Roboterachsen (System n, Qnnnn) [Wrong number of robot axes (System n, Qnnnn)]	The number of robot axes is defined by the number of NC axis entries (in Q.063 – kinematic axis sequence). This error message appears if that number does not match the number set in the transformation.
0x02100006	Kein Speicher für Transformation (System n) [No memory for transformation (System n)]	
0x02100007	Linkparameter unzulässig (System n, Qnnnn) [Invalid link parameter (System n, Qnnnn)]	The required entries for robot arm length are checked according to the respective transformation. The criterium here is merely whether a required parameter is not equal to zero, but not whether an arm length has the correct value.  This error message appears if a required arm length has not been entered. In that case there is an invalid value in link parameter Q.nnn, System n.

## 13.3.3 0x0211 Decoder errors

The subsystem number is indicated in the first parameter, the number of the NC record in the second parameter.

Error no.	Error text	Cause / Remedy
0X02110000	Mehr als 4 Kennbuchstaben (System n, Nnnn) [More than four identifiers (System n, Nnnn)]	CNC words begin with a maximum of 4 letters.
0X02110001	Zahl länger als 14 Ziffern (System n, Nnnn) [Number longer than 14 digits (System n, Nnnn)]	Numerical values must have more than 14 digits (including the decimal point).
0X02110002	Mehr als 3 Sonderzeichen (System n, Nnnn) [More than 3 special characters (System n, Nnnn)]	
0X02110003	Unzulässiges Zeichen (System n, Nnnn) [Invalid character (System n, Nnnn)]	Delete the CNC record and re-enter if necessary (try programming on a PC text system using lower case letters or no ASCII symbols).
0X02110004	LF oder ")" fehlt (System n, Nnnn) [LF or ")" missing (System n, Nnnn)]	
0X02110005	Unzulässige Zeichenkombination (System n, Nnnn) [Invalid character combination (System n, Nnnn)]	
0X02110006	Unzulässige G-Funktion (System n, Nnnn) [Invalid G function (System n, Nnnn)]	
0X02110007	Achszahl zu klein für \$-Funktion (System n, Nnnn) [Number of axes too low for \$ function (System n, Nnnn)]	E.g. \$ 30 (synchronous operation): At least 2 axes have to be programmed.
0X02110008	Keine Rundachse o. Spindel im System (System n, Nnnn) [No rotary axis or spindle in the system (System n, Nnnn)]	see Q.054.
0X02110009	Achse im System unbekannt (System n, Nnnn) [Axis not known in system (System n, Nnnn)]	Axis letter is set in Q.021. Pay attention to system group Q.020.
0X0211000A	Achse keine Rundachse o. Spindel (System n, Nnnn) [Axis not rotary axis or spindle (System n, Nnnn)]	see Q.054.
0X0211000B	Winkelgeschwindigkeit = 0 (System n, Nnnn) [Angular velocity = 0 (System n, Nnnn)]	With G95 a velocity greater than 0 must be programmed.



0X0211000C	Syntaxfehler in \$-Funktion (System n, Nnnn) [Syntax error in \$ function (System n, Nnnn)]	
0X0211000D	Zu viele Befehle (System n, Nnnn) [Too many instructions (System n, Nnnn)]	
0X0211000E	Unzulässige \$-Funktion (System n, Nnnn) [Invalid \$ function (System n, Nnnn)]	
0X0211000F	Befehl unbekannt (System n, Nnnn) [Instruction unknown (System n, Nnnn)]	
0X02110010	Zu viele SE,RS,WA,M,T,D,H Funktionen (System n, Nnnn) [Too many SE, RS, WA, M, T, D, H functions (System n, Nnnn)]	For each CNC record you can program 8 SE, RS, WA functions, 3 M functions and 2 H functions.
0X02110011	Unzulässiger Wert (System n, Nnnn) [Invalid value (System n, Nnnn)]	E.g. B%R0 with R0 <= 0
0X02110012	Unzulässiger R-Parameter (System n, Nnnn) [Invalid R parameter (System n, Nnnn)]	Parameter does not exist or must not be changed (R300...R310).
0X02110013	Unzulässige Verknüpfung (System n, Nnnn) [Invalid operation (System n, Nnnn)]	Error in parameter calculation
0X02110014	Funktion noch nicht vorhanden (System n, Nnnn) [Function not yet available (System n, Nnnn)]	
0X02110015	Satz nicht gefunden (System n, Nnnn) [Record not found (System n, Nnnn)]	Record does not exist in the selected program/program section.
0X02110016	Unterprogramm-Schachtelung größer als 4 (System n, Nnnn) [Subroutine nesting greater than 4 (System n, Nnnn)]	
0X02110017	Programm nicht gefunden (System n, Nnnn) [Program not found (System n, Nnnn)]	The selected program does not exist.
0X02110018	Werkzeug verschlissen (System n, Nnnn) [Tool worn (System n, Nnnn)]	

0X02110019	Keine Koordinate erlaubt (System n, Nnnn) [Coordinates not allowed (System n, Nnnn)]	E.g. axis is in follow-up operation.
0X0211001A	Werkzeug-Speicher nicht vorhanden (System n, Nnnn) [Tool memory not available (System n, Nnnn)]	The called tool compensation number T...0x is not in the memory. You can set the number of the tool memory in Q.01.
0X0211001B	M17/M30 fehlt (System n, Nnnn) [M17/M30 missing (System n, Nnnn)]	Program end identifier missing. M17/M30 does not have to be programmed if Q28 bit 5 is set.
0X0211001C	Zu viele Nullpunkt-Verschiebungen (System n, Nnnn) [Too many zero point offsets (System n, Nnnn)]	E.g. N10 G54 G55 in one CNC record.
0X0211001D	G02/03-Satz falsch programmiert (System n, Nnnn) [G02/03 record incorrectly programmed (System n, Nnnn)]	Less than two axes or no mid-point coordinate or the plane selection was programmed incorrectly. The error message is also displayed if the assignment to the coordinate system is missing in Q.54.
0X0211001E	Radius = 0 (System n, Nnnn)	
0X0211001F	Kreisendpunkt-Fehler (System n, Nnnn) [Circle end position error (System n, Nnnn)]	The difference between the programmed start and end radius is larger than the circle end position control (can be set with Q.06).
0X02110020	Zu viele Achsen programmiert (System n, Nnnn) [Too many axes programmed (System n, Nnnn)]	
0X02110021	G33 - Achsanzahl > 2 (System n, Nnnn) [G33 - number of axes > 2 (System n, Nnnn)]	You are only allowed to program 2 axes.
0X02110022	Division durch 0 (System n, Nnnn) [Division by 0 (System n, Nnnn)]	Parameter calculation results in a division by 0.
0X02110023	Modulozahl Q.037 = 0 (System n, Nnnn) [Modulo number Q.037 = 0 (System n, Nnnn)]	If you use rotary axes you must enter the pulses/rotation in Q.037.
0X02110024	Q-Parameterzugriff nicht erlaubt (System n, Nnnn) [Q parameter access not allowed (System n, Nnnn)]	The CNC program can only change alterable Q parameters if Q37 bit 6 is set.
0X02110025	G97 (S in 1/min) nicht erlaubt (System n, Nnnn) [G97 (S in 1/min) not allowed (System n, Nnnn)]	
0X02110026	Radiusachse nicht definiert (System n, Nnnn) [Radius axis not defined (System n, Nnnn)]	The radius axis assignment in \$ 34 for programming cutting speeds is incorrect. The selected axis is the main spindle.



0X02110027	Keine Leitachse für \$30/\$31 (System n, Nnnn) [No lead axis for \$30/\$31 (System n, Nnnn)]	
0X02110028	Nur ein Unterprogramm-Aufruf erlaubt (System n, Nnnn) [Only one subroutine call permitted (System n, Nnnn)]	
0X02110029	SRK-Ebene undefiniert (System n, Nnnn) [SRK level undefined (System n, Nnnn)]	The coordinate assignment in Q.054 is incomplete. The plane assignment with G17 to G19 or instruction \$47 is missing.
0X0211002A	Syntaxfehler in G76/G77 (System n, Nnnn) [Syntax error in G76/G77 (System n, Nnnn)]	
0X0211002B	\$33 nicht programmiert (System n, Nnnn) [\$33 not programmed (System n, Nnnn)]	
0X0211002C	G76 Schnittanzahl <= 0 (System n, Nnnn) [G76 number of cuts <= 0 (System n, Nnnn)]	
0X0211002D	G76/G77 Steigung <= 0 (System n, Nnnn) [G76/G77 lead <= 0 (System n, Nnnn)]	
0X0211002E	G76/G77 Gewinde nicht möglich (System n, Nnnn) [G76/G77 thread not possible (System n, Nnnn)]	
0X0211002F	G76/G77 Winkel zu groß (System n, Nnnn) [G76/G77 angle too large (System n, Nnnn)]	
0X02110030	G76/G77 keine Hauptspindel (System n, Nnnn) [G76/G77 no main spindle (System n, Nnnn)]	
0X02110031	F/S-Wert < 0 [F/S value < 0]	
0X02110032	Option nicht freigegeben (System n, Nnnn)n [Option not enabled (System n, Nnnn)n]	
0X02110033	Illegaler Befehl OCI (System n, Nnnn) [Invalid OCI instruction (System n, Nnnn)]	
0X02110034	SRK mit G25 nicht erlaubt (System n, Nnnn) [SRK with G25 not permitted (System n, Nnnn)]	



0X02110035	Keine Achse oder nur \$38 programmiert (System n, Nnnn) [No axis programmed or only \$38 (System n, Nnnn)]	
0X02110036	Zusätzliche OCI-Achse programmiert (System n, Nnnn) [Additional OCI axis programmed (System n, Nnnn)]	
0X02110037	Illegaler Befehl, Vorgängersatz OCI (System n, Nnnn) [Invalid instruction, previous OCI record (System n, Nnnn)]	
0X02110038	Falscher FFTP-Wert (System n, Nnnn) [Incorrect FFTP value (System n, Nnnn)]	FFTP value > 100 or < 0, the FFTP value must be programmed in % of rapid feed
0X02110039	Keine Transformation aktiv (System n, Nnnn) [No transformation active (System n, Nnnn)]	
0X0211003A	Precompilieren nicht möglich (System n, Nnnn) [Precompiling not possible (System n, Nnnn)]	
0X0211003B	Vorschub zu groß (System n, Nnnn) [Feedrate is too high (System n, Nnnn)]	
0X0211003C	Verfahrstrecke zu groß (System n, Nnnn) [Travel too large (System n, Nnnn)]	For a rotary/continuous axis if the travel distance programmed in the NC record exceeds the internal resolution
0X0211003D	Unerreichbare Position (System n, Nnnn) [Unreachable position (System n, Nnnn)]	
0X0211003E	Kein Programm aktiv (System n, Nnnn) [No program active (System n, Nnnn)]	
0X0211003F	G77 Beschleunigungsfehler (System n, Nnnn) [G77 Beschleunigungsfehler (G77 acceleration error)]	
0X02110040	G77 Verzögerungsfehler (System n, Nnnn) [G77 deceleration error (System n, Nnnn)]	
0X02110041	Gelenkkonfiguration nur in PTP erlaubt (System n, Nnnn) [Joint configuration allowed only in PTP (System n, Nnnn)]	



0X02110042	G39 verboten bei Spiegeln / Drehen (System n, Nnnn) [G39 prohibited with mirroring/rotation (System n, Nnnn)]
0X02110043	Unzulässiger Rampenwert (ACC) (System n, Nnnn) [Invalid ramp value (ACC) (System n, Nnnn)]
0X02110044	Falsche Transformation (System n, Nnnn) [Incorrect transformation (System n, Nnnn)]
0X02110045	G72 .. G75 fehlt (System n, Nnnn) [G72 .. G75 missing (System n, Nnnn)]

### 13.3.4 0x0212 Error in automatic record insertion

The subsystem number is indicated in the first parameter, the number of the NC record in the second parameter.

Error no.	Error text	Cause / Remedy
0x02120000	Radius / Fase zu groß (System n, Nnnn) [Radius/chamfer too large (System n, Nnnn)]	In a CNC record you have programmed a transition radius (RA..) or a transition chamfer (RB..) whose starting position is outside the programmed coordinates of the CNC record in which the radius or the chamfer is programmed or whose target position would go beyond the coordinates of the next CNC record.
0x02120001	Übergangsfase nicht erlaubt (System n, Nnnn) [Transition chamfer not allowed (System n, Nnnn)]	In a CNC record you have programmed a transition chamfer (RB..) that is not between two G1 records.
0x02120002	Unerreichbare Position (System n, Nnnn) [Unreachable position (System n, Nnnn)]	
0x02120003	G02/03-Satz falsch programmiert (System n, Nnnn) [G02/03 record incorrectly programmed (System n, Nnnn)]	
0x02120004	Kreisendpunktfehler (System n, Nnnn) [Circle end position error (System n, Nnnn)]	



### 13.3.5 0x0213 Error in tool nose radius compensation

The subsystem number is indicated in the first parameter, the number of the NC record in the second parameter.

Error no.	Error text	Cause / Remedy
0x02130000	NC-Satz ohne Verfahrbewegung (System n, Nnnn) [NC record without motion (System n, Nnnn)]	
0x02130001	Konturradius <= Schneidenradius (System n, Nnnn) [Contour radius <= tool radius (System n, Nnnn)]	
0x02130002	Konturradius = 0 (System n, Nnnn) [Contour radius = 0 (System n, Nnnn)]	
0x02130003	Winkel zu spitz (System n, Nnnn) [Angle too sharp (System n, Nnnn)]	The selected tool cannot move to the programmed position.
0x02130004	Werkzeugradius zu groß (System n, Nnnn) [Tool radius too large (System n, Nnnn)]	The selected tool cannot move to the programmed position.
0x02130005	Startpunkt = Endpunkt (System n, Nnnn) [Start position = end position (System n, Nnnn)]	The selected tool cannot move to the programmed position.
0x02130006	SRK mit G50 nicht möglich (System n, Nnnn) [SRK with G50 not possible (System n, Nnnn)]	

### 13.3.6 0x0214 Error in interpolation preparation

The subsystem number is indicated in the first parameter, the number of the NC record in the second parameter.

Error no.	Error text	Cause / Remedy
0x02140000	Speicher voll (System n, Nnnn) [Memory full (System n, Nnnn)]	
0x02140001	Unerreichbare Position Satzanfang (System n, Nnnn) [Unreachable position record start (System n, Nnnn)]	
0x02140002	Unerreichbare Position Satzende (System n, Nnnn) [Unreachable position record end (System n, Nnnn)]	
0x02140003	Unerreichbare Position Satzmitte (System n, Nnnn) [Unreachable position record middle (System n, Nnnn)]	



0x02140004	Kreisendpunktfehler (System n, Nnnn) [Circle end position error (System n, Nnnn)]
0x02140005	Keine Gewindeleitachse (System n, Nnnn) [No thread lead axis (System n, Nnnn)]
0x02140006	Kein Spindelwert (System n, Nnnn) [No spindle actual value (System n, Nnnn)]
0x02140007	M03 / M04 / M05 falsch (System n, Nnnn) [M03 / M04 / M05 incorrect (System n, Nnnn)]
0x02140008	\$61 Stichlänge=0 (System n, Nnnn) [\$61 Stich length=0 (System n, Nnnn)]
0x02140009	Programmierter Weg zu lang (System n, Nnnn) [Programmed path too long (System n, Nnnn)]
0x0214000A	Wechsel der Gelenkkonfiguration nicht möglich (System n, Nnnn) [Joint configuration change not possible (System n, Nnnn)]

## 13.3.7 0x0220 Axis errors

The axis number is indicated in the parameter.

Error no.	Error text	Cause / Remedy
0x02200000	Messkreisfehler (Achse n) [Measuring circuit error (Axis n)]	Cyclical data exchange between the PN-MIC xx and the drive has been interrupted.
0x02200001	Stillstandsüberwachung (Achse n) [Zero-speed control (Axis n)]	An axis has left the zero-speed control window Q.046 without a travel assignment or did not reach the set control window quickly enough. The control deletes the controller enable after the delay time Q.047 (ms). Possible causes: <ul style="list-style-type: none"> <li>• Control time too short (Q.047)</li> <li>• Control window too small (Q.046)</li> <li>• Controller amplification factor too small</li> <li>• Drift compensation set incorrectly</li> </ul>
0x02200002	Schleppabstand zu groß (Achse n) [Following error too large (Axis n)]	The drive cannot follow the values set in the CNC. Following error control causes feed stop with ramp. Possible cause: Either the maximum velocity Q.023 is too large, the maximum following error Q.042 too small, the KV factor Q.022 too small or the drive defective.
0x02200003	Reglerfreigabe fehlt (Achse n) [no controller enable (Axis n)]	The position controller was opened from the outside by taking away the controller enable while the axis was moving.
0x02200004	Softwareendschalter + (Achse n) [Software limit switch + (Axis n)]	The boundary set in Q.035 has been reached (only effective after homing).
0x02200005	Softwareendschalter - (Achse n) [Software limit switch - (Axis n)]	The boundary set in Q.036 has been reached (only effective after homing).
0x02200006	Vorschubfreigabe fehlt (Achse n) [No feed enable (Axis n)]	The travel assignment of an axis cannot be executed due to a missing feed enable (PLC program).
0x02200007	Hardwareendschalter + (Achse n) [Hardware limit switch + (Axis n)]	The axis is on the + hardware limit switch. It can be moved manually in "-" direction.
0x02200008	Hardwareendschalter - (Achse n) [Hardware limit switch - (Axis n)]	The axis is on the - hardware limit switch. It can be moved manually in "+" direction.



0x02200009	Achse nicht referenziert (Achse n) [Axis not referenced (Axis n)]	A program can only be started after all its axes have been synchronized. If axes do not need to be homed, e.g. rotary axes/spindles set Q.52 bit 3 = 1 to tell the program that the axis has been synchronized.
0x0220000A	Repositionieren (Achse n) [Repositioning (Axis n)]	
0x0220000B	SERCOS-Zykluszeit zu klein (Achse n) [SERCOS cycle time too short (Axis n)]	The set cycle time for SERCOS is too short.
0x0220000C	Fehler beim Nachführbetrieb (Achse n) [Error in follow-up operation (Axis n)]	A non-programmed follow-up operation was executed during the CNC program.
0x0220000D	Sollwertfehler (Achse n) [Set value error (Axis n)]	
0x0220000E	Gewindefehler (Achse n) [Thread error (Axis n)]	
0x0220000F	Watchdog Antriebsmodul (Achse n) [Drive module watchdog (Axis n)]	
0x02200010	PCI-Basisadresse ungültig (Achse n) [PCI base address invalid (Axis n)]	Only on ProNumeric / PN-MIC
0x02200011	Falsche Modul-Nummer (Achse n) [Incorrect module number (Axis n)]	
0x02200012	Falsche Kanal-Nummer (Achse n) [Incorrect channel number (Axis n)]	
0x02200013	Kartenkennung ungültig (Achse n) [Card ID invalid (Axis n)]	
0x02200014	Drehzahlsollwert zu groß (Achse n) [Desired rotary speed too high (Axis n)]	



## 13.3.8 0x0221 SERCOS diagnosis status

These error messages apply only to ProNumeric / PN-MIC.  
The axis number is normally indicated in the first parameter.

Error no.	Error text	Cause / Remedy
0x02210000	Falsche Adresse des Antriebs (Achse n) [Incorrect drive address (Axis n)]	
0x02210001	Handshake-Timeout SERCOS (Achse n) [SERCOS handshake timeout (Axis n)]	
0x02210002	Antriebstelegramm-Ausfall SERCOS (Achse n) [SERCOS drive telegram failure (Axis n)]	
0x02210003	Antriebsfehler (Achse n) [Drive error (Axis n)]	
0x02210004	Fehler beim Umschalten Phase 2 -> 3 (Achse n) [Error while switching phase 2 -> 3 (Axis n)]	
0x02210005	Fehler beim Umschalten Phase 3 -> 4 (Achse n) [Error while switching phase 3 -> 4 (Axis n)]	
0x02210006	Kommandofehler Antrieb (Achse n) [Drive instruction error (Axis n)]	
0x02210007	SERCOS-PLL asynchron (Ring n, Antrieb n) [SERCOS PLL asynchronous (Ring n, Axis n)]	

**13.3.9 0x0222 SERCOS status class 1**

These error messages apply only to ProNumeric / PN-MIC.  
The axis number is indicated in the first parameter.

Error no.	Error text	Cause / Remedy
0x02220000	Überlast Antrieb (Achse n) [Overload drive (Axis n)]	
0x02220001	Übertemperatur Verstärker (Achse n) [Amplifier overheated (Axis n)]	
0x02220002	Übertemperatur Motor (Achse n) [Motor overheated (Axis n)]	
0x02220003	Kühlung Antrieb (Achse n) [Drive cooling (Axis n)]	
0x02220004	Steuerspannung Antrieb (Achse n) [Control voltage drive (Axis n)]	
0x02220005	Messkreisfehler Antrieb (Achse n) [Drive measuring circuit error (Axis n)]	
0x02220006	Kommutierung Antrieb (Achse n) [Drive commutation (Axis n)]	
0x02220007	Überstrom Antrieb (Achse n) [Drive overcurrent (Axis n)]	
0x02220008	Überspannung Antrieb (Achse n) [Drive overvoltage (Axis n)]	
0x02220009	Unterspannung Antrieb (Achse n) [Drive undervoltage (Axis n)]	
0x0222000A	Phasenfehler Antrieb (Achse n) [Drive phase error (Axis n)]	
0x0222000B	Regelfehler Antrieb (Achse n) [Drive control error (Axis n)]	

0x0222000C	Kommunikation Antrieb (Achse n) [Drive communication (Axis n)]
0x0222000D	Lagegrenzwert Antrieb (Achse n) [Drive position limit (Axis n)]
0x0222000E	Reserviert Antrieb (Achse n) [Drive reserved (Axis n)]
0x0222000F	Fehler Antrieb (KLasse 1) (Achse n) [Drive error (class 1) (Axis n)]

### 13.3.10 0x0223 Manufacturer-specific SERCOS errors

These error messages apply only to ProNumeric / PN-MIC.

The drive's manufacturer-specific error numbers are displayed.

### 13.3.11 0x0224 Manufacturer-specific SERCOS error texts

These error messages apply only to ProNumeric / PN-MIC.

The drive's manufacturer-specific error texts are displayed.



### 13.3.12 0x0230 Drive errors

These error messages apply only to the XCN.

The CNC axis number is normally indicated in the first parameter.

In the following error messages MC stands for MotionControl.

Error no.	Error text	Cause / Remedy
0x02300001	SLM: Sende/Empfangs (Tx,Rx) Fehler [SLM: Send/receive error (Tx, Rx)]	
0x02300002	SLM: SLM nicht vorhanden (Achse n, Ort=m) [SLM: SLM not found (Axis n, Location=m)]	Possible cause: - Access to the SLM-ASIC defective
0x02300003	SLM: Unbekannte SLM-ID (Achse n, Ort=m) [SLM: Unknown SLM ID (Axis n, Location=m)]	SLM-ID not recognized. Possible causes: - SLM drive without power - Communication with SLM drive not possible
0x02300004	SLM: Checksum Fehler (Encoder-Object) (Achse n, Ort=m) [SLM: Checksum error (encoder object) (Axis n, Location=m)]	Possible cause: - The encoder data saved in the SLM module (on the motor) is defective.
0x02300005	SLM: Checksum Fehler (Motor-Object) (Achse n, Ort=m) [SLM: Checksum error (motor object) (Axis n, Location=m)]	Possible cause: - The motor data saved in the SLM module (on the motor) is defective.
0x02300006	SLM: Parameter konnte nicht gelesen werden (Achse n, Ort=m) [SLM: Parameter could not be read (Axis n, Location=m)]	Possible causes: - Read access currently not possible - Parameter does not exist
0x02300007	SLM: Parameter konnte nicht geschrieben werden (Achse n, Ort=m) [SLM: Parameter could not be written (Axis n, Location=m)]	Possible causes: - Write access currently not possible - Parameter does not exist
0x0230000C	SLM: Kommunikations Fehler (Achse n, Ort=m) [SLM: Communication error (Axis n, Location=m)]	Possible cause: - SLM cable malfunction due to external influences
0x0230000D	SLM: CRC-Fehler (Achse n, Ort=m) [SLM: CRC error (Axis n, Location=m)]	Possible cause: - SLM cable malfunction due to external influences
0x0230000E	SLM: Fehler Service-Kanal (Achse n, Ort=m) [SLM: Service channel error (Axis n, Location=m)]	Possible cause: - Time limit overrun while reading or writing



0x02300014	MC: Zykluszeit MotionControl falsch [MC: Incorrect MotionControl cycle time]	Possible cause: The cycle time (position controller scan rate) is set incorrectly: The <i>CYCLE_TIME</i> parameter must be set in 1000 µs steps (Schleicher Dialog, <i>Startup/Drive Configuration/Drive Parameters</i> menu).
0x02300015	MC: Zykluszeit IPO falsch [MC: Incorrect interpolation cycle time]	Possible cause: - The interpolation cycle time (interpolation scan rate) is set incorrectly: The number entered for the <i>Interpolation scan rate</i> must be a whole-number multiple of the position control cycle (Schleicher Dialog, <i>Startup/Default Settings</i> menu).
0x0230001E	MC: CAN-Interface ist nicht bereit [MC: CAN interface not ready]	Possible cause: - No ready message from PLC in MC_CAN() function block status word
0x0230001F	MC: XRIO (ANALOG) -Interface ist nicht bereit [MC: XRIO (ANALOG) interface not ready]	Possible cause: - No ready message from PLC in MC_ANALOG() function block status word
0x0230005A	MC: Aufruf ohne Initialisierung [MC: Call without initialization]	Possible cause: - The cyclical MotionControl task has detected a defective initialization (see logbook for further error messages)
0x02300060	MC: zu wenig Speicher (Ort=m) [MC: Insufficient memory (Location=m)]	The memory required for MotionControl could not be generated.
0x02300061	MC: Funktion z.Z. nicht verfügbar (Ort=m) [MC: Function currently not available (Location=m)]	
0x02300062	MC: unbekannte Antriebsschnittstelle (Ort=m) [MC: Unknown drive interface (Location=m)]	
0x02300063	MC: fataler Fehler (Ort=m) [MC: Fatal error (Location=m)]	
0x02300064	MC: Datei (drive_*.ini) kann nicht generiert werden (File_ID=m) [MC: File (drive_*.ini) cannot be generated (File_ID=m)]	
0x02300065	MC: Datei (drive_*.ini) ist nicht vorhanden (File_ID=m) [MC: File (drive_*.ini) does not exist (File_ID=m)]	



0x02300066	MC: Datei (drive_*.ini) kann nicht geschrieben werden (File_ID=m) [MC: File (drive_*.ini) cannot be written (File_ID=m)]	The INI, OLD or CRC file is write-protected
0x02300067	MC: fehlende/fehlerhafte Parameter in Datei (drive_*.ini) (File_ID=m) [MC: Missing or defective parameter in file (drive_*.ini) (File_ID=m)]	No current value was found in the INI file or the parameter does not exist.
0x02300068	MC: Datei (drive_*.ini) ist fehlerhaft (File_ID=m) [MC: File (drive_*.ini) is defective (File_ID=m)]	The INI file contains invalid or defective data.
0x02300069	Warnung: MC: neue Achse (mit Default-Einstellungen) eingefügt (File_ID=m) [Warning: MC: New axis inserted (with default settings (File_ID=m)]	Note, warning: One or more axes have been inserted in the XCN.
0x0230006A	Warnung: MC: nicht zugeordnete Achse(n) in Datei (drive_*.ini) (File_ID=m) [Warning: MC: File (drive_*.ini) contains non-assigned axis/axes (File_ID=m)]	Note, warning: One or more axes have been deleted in the XCN.
0x0230006B	Warnung: MC: Zuordnung der Achsen wurde geändert in Datei (drive_*.ini) (File_ID=m) [Warning: MC: Axis assignment changed in file (drive_*.ini) (File_ID=m)]	Note, warning: The assignment of one or more axes has been changed in the XCN.
0x0230006C	MC: Datei (drive_*.old) kann nicht generiert werden (File_ID=m) [MC: File (drive_*.old) cannot be generated (File_ID=m)]	The OLD file could not be generated.
0x0230006D	MC: Datei (drive_*.crc) kann nicht generiert werden (File_ID=m) [MC: File (drive_*.crc) cannot be generated (File_ID=m)]	The CRC file could not be generated.
0x0230006E	MC: temporäre Datei (drive_*.ini) ist nicht vorhanden (File_ID=m) [MC: Temporary file (drive_*.ini) does not exist (File_ID=m)]	The drive_*.ini file on the RAM disk in the XCN – with the drive parameters that have been edited online – does not exist after the accept request.

0x02300095	MC: TaskSynchronisation TIMEOUT	<p>The time limit for initial MotionControl task synchronization with the interpolation task has been exceeded. Check the interpolation cycle time and the MotionControl cycle time. Dividing the interpolation cycle time by the MotionControl cycle time must produce a whole number.</p> <p>(Interpolation cycle time / MotionControl cycle time) = whole number</p>
0x02300096	MC: Task asynchron [MC: Asynchronous task]	<p>The MotionControl task is not synchronized with the interpolation task. Check the interpolation cycle time and the MotionControl cycle time. Dividing the interpolation cycle time by the MotionControl cycle time must produce a whole number.</p> <p>(Interpolation cycle time / MotionControl cycle time) = whole number</p>
0x023000A0	MC: CAN-Antriebsfehler (Achse n, Ort=m) [MC: CAN drive error (Axis n, Location=m)]	This error message is generated as soon as the PLC program signals a drive error via STATUS_WORD (bit 3) in the MC_CAN function block.
0x023000A1	MC: Analog-Antriebsfehler (Achse n, Ort=m) [MC: Analog drive error (Axis n, Location=m)]	This error message is generated as soon as the PLC program signals a drive error via STATUS_WORD (bit 3) in the MC_ANALOG function block.
0x02300FA0	SERCOS: Allgemeiner Fehler [SERCOS: General error]	
0x02300FA1	SERCOS: Fataler Fehler [SERCOS: Fatal error]	
0x02300FA2	SERCOS: Fehler bei der Phasenumschaltung (Achse n, IDN=x) [SERCOS: Error during phase switching (Axis n, IDN=x)]	<p>The SERCOS communication phase cannot be changed. Possible cause: - A SERCOS parameter was rejected by the drive. IDN displays this parameter number.</p>
0x02300FA3	SERCOS: Zustandsklasse 1 Fehler (Achse n, IDN 11=x) [SERCOS: Status class 1 error (Axis n, IDN 11=x)]	A SERCOS error has occurred (status class 1). IDN 00011 contains the rest of the error information:
		Bit 0:     Overload shutdown
		Bit 1:     Amplifier overheating shutdown
		Bit 2:     Motor overheating shutdown
		Bit 3:     Cooling error shutdown
		Bit 4:     Control voltage error
		Bit 5:     Feedback error
		Bit 6:     Error in commutation system
		Bit 7:     Overcurrent
		Bit 8:     Overvoltage
		Bit 9:     Undervoltage error
		Bit 10:    Phase error in power supply
Bit 11:    Excessive deviation		



		Bit 12: Communication error
		Bit 13: Position limit exceeded
		Bit 14: (reserved)
		Bit 15: Manufacturer-specific error
0x02300FA4	SERCOS: Herstellerspezifischer Fehler (Achse n, Fehler-Nr.=x) [SERCOS: Manufacturer-specific error (Axis n, Error no.=x)]	Shows the drive manufacturer's error number.
0x02300FA5	SERCOS: Kommunikationsfehler / Ring nicht geschlossen [SERCOS: Communication error / ring not closed]	
0x02300FA6	SERCOS: 2. Ring/Modul ist nicht erlaubt [SERCOS: 2nd ring/module not permitted]	
0x02300FC0	Warnung: Altes Betriebssystem, keine PLC-SERCOS Schnittstelle [Warning: Old operating system, no SPS-SERCOS interface]	Only on ProNumeric / PN-MIC: The CNC operating system is incompatible with the SERCOS interface on the PN-MIC card.

The parameter that defines the Location or File\_ID is defined as follows:

Parameter value	Error location	Comment
100	CALLING_PARAMETERS	
200	ALLOCATE_MEMORY	
300	READ_FILE	
301	SLM_INI_FILE_ID	Error while generating / modifying the DRIVE_SLM.INI file
302	CAN_INI_FILE_ID	Error while generating / modifying the DRIVE_CAN.INI file
303	ANALOG_INI_FILE_ID	Error while generating / modifying the DRIVE_ANA.INI file
304	SERCOS_INI_FILE_ID	Error while generating / modifying the DRIVE_SRC.INI file
400	CALCULATE_PARAMETERS	
500	INIT_CNC_INTERFACE	
600	INIT_DRIVE_INTERFACE	
601	IDENTIFYSLM	
602	READENCODER	
603	READMOTOR	
604	INITSLMPHASE1	
605	INITSLMPHASE2	
900	CYCLIC_TASK	

## 13.4 CAN Error Messages

0x0300nnnn	CAN-specific operating system error
0x0301nnnn	CAN driver error

### 13.4.1 0x0300 CAN-specific operating system errors

The first parameter indicates the card number of the active CAN card (on ProNumeric this is a PN-MIC card).

Error no.	Error text	Cause / Remedy
0x03000001	CAN: Ungültige Karte oder falsche Shared Memory Revision (Karte n) [CAN: Invalid card or incorrect shared memory revision (Card n)]	Only on ProNumeric / PN-MIC: Update PN-MIC firmware.
0x03000002	CAN: Karte nicht verfügbar (Heartbeat-Fehler) (Karte n) [CAN: Card not available (heartbeat error) (Card n)]	Only on ProNumeric / PN-MIC: PN-MIC card possibly defective. Switch computer off and on again.
0x03000003	CAN: Synchronisationsfehler (Treiberzugriff gesperrt) (Karte n) [CAN: Synchronization error (driver access blocked) (Card n)]	Warning that I/O map has been delayed.
0x03000011	CAN: Fehler Speicherzuweisung für Ereignisliste (Karte n) [CAN: Memory allocation error for event list (Card n)]	Max. memory for event list exceeded. Collect events (see SDO function blocks).
0x03000012	CAN: Überlauf Fehler-Ereignisliste (Karte n) [CAN: Error event list overrun (Card n)]	Max. number of error events exceeded.
0x03000013	CAN: Überlauf Emergency-Ereignisliste (Karte n) [CAN: Emergency event list overrun (Card n)]	Max. number of emergency events exceeded.
0x03000014	CAN: Überlauf Ereignisliste (Karte n) [CAN: Event list overrun (Card n)]	Max. number of general events exceeded.



### 13.4.2 0x0301 CAN driver errors

The first parameter indicates the card number of the active CAN card (on ProNumeric this is a PN-MIC card).

Error no.	Error text	Cause / Remedy
0x03010006	CAN: Feldbusfehler (Karte n) [CAN: Field bus error (Card n)]	Check CAN network installation.
0x03010096	CAN: SPS- (Betriebsystem-) Fehler (Karte n) [CAN: PLC (operating system) error (Card n)]	Check I/O configuration in PLC programming system. Reset controller if necessary.
0x030100A0	CAN: Konfiguration konnte nicht gelesen werden oder SPS-Timeout (Karte n) [CAN: Configuration could not be read or PLC timeout (Card n)]	Reset controller if necessary.
0x030100AA	CAN: Ungültige Konfiguration (Karte n) [CAN: Invalid configuration (Card n)]	Check CAN bus configuration.
0x03010328	CAN: Unbekannter CAN-Fehler (Karte n) [CAN: Unknown CAN error (Card n)]	Error on card <n>. Check CAN network.

## 13.5 User-defined Error Messages

0x0400nnnn to 0x04FFnnnn	User-defined errors, can be entered in the error logbook by the PUT_ERROR function block
--------------------------------	--

### 13.5.1 0x0400 User-defined error messages

User-defined errors (in groups 0x0400 to 0x04FF), which can be entered in the error logbook by the PUT\_ERROR function block

A user-defined error text can be assigned to each of these error numbers (including display parameters which can be transferred to the function block).

Error message examples:

Error no.	Error text	Cause / Remedy
0x04000000	Anwender-Fehler [Gruppe 0, Nummer 0] (Par1=x, Par2=y, Par3=z) [User error [Group 0, Number 0] (Par1=x, Par2=y, Par3=z)]	
0x04000001	Anwender-Fehler [Gruppe 0, Nummer 1] (Par1=x, Par2=y, Par3=z) [User error [Group 0, Number 1] (Par1=x, Par2=y, Par3=z)]	
0x04000002	Anwender-Fehler [Gruppe 0, Nummer 2] (Par1=x, Par2=y, Par3=z) [User error [Group 0, Number 2] (Par1=x, Par2=y, Par3=z)]	



### 13.6 NC Memory Access Errors

These error messages are not entered in the active error buffer.

These are dialog errors that can occur when the CNC memory is accessed, if Schleicher Dialog, via the shared RAM, makes requests that have caused an error (e.g. accept or delete NC programs or accept configuration data). They are returned in the request cell.

Accepting NC programs:

*cncMem.comSect.wrdPNRw.lRecvNcPrg*

Accepting the configuration file:

*cncMem.comSect.wrdPNRw.lRecvConfig*

Accepting the arithmetic parameter file:

*cncMem.comSect.wrdPNRw.lRecvRVa1*

Deleting NC programs:

*cncMem.comSect.wrdPNRw.lRqDelNcPrg*

Error no.	Error text	Cause / Remedy
-2	Datei-Fehler: Die Datei kann nicht geöffnet werden [File error: File cannot be opened]	Access to PC file system not possible
-3	NC-Speicherfehler: Die max. Anzahl der NC-Programme ist überschritten [NC memory error: Max. number of NC programs exceeded]	Increase number of NC programs or delete programs
-4	NC-Speicherfehler: Der NC-Programmspeicher ist voll [NC memory error: Der NC-Programmspeicher ist voll]	Enlarge NC program memory or delete programs
-5	NC-Speicherfehler: Der NC-Satz ist zu lang (max. 120 Zeichen) [NC memory error: NC record too long (max. 120 characters)]	
-6	NC-Speicherfehler: Das NC-Programm läuft gerade [NC memory error: NC program running]	An active program cannot be overwritten or deleted
-7	NC-Speicherfehler: Das NC-Programm befindet sich gerade im Lesezugriff [NC memory error: NC program in read access]	A program in read access cannot be deleted or overwritten
-8	NC-Speicherfehler: Das NC-Programm befindet sich gerade im Schreibzugriff [NC memory error: NC program in write access]	A program in write access cannot be deleted or overwritten
-9	NC-Speicherfehler: Das NC-Programm ist nicht vorhanden [NC memory error: Das NC-Programm ist nicht vorhanden]	Selected program number for deleting does not exist



-10	NC-Speicherfehler: Falsche NC-Programmnummer [NC memory error: Incorrect NC program number]	Selected program number for deleting is incorrect
-11	NC-Speicherfehler: Das NC-Programm ist aktiv [NC memory error: The NC program is active]	An active program cannot be deleted or overwritten
-12	Falsche Anfangskennung (Q- oder R-Parameter-Datei) (Zeile n) [Incorrect start identifier (Q or R parameter file) (Line n)]	An error has occurred in line <n> due to an incorrect start identifier.
-13	Falsche Parameternummer (Q- oder R-Parameter-Datei) (Zeile n) [Incorrect parameter number (Q or R parameter file) (Line n)]	An error has occurred in line <n> due to an incorrect parameter number.
-14	Falscher Parameterwert (Q- oder R-Parameter-Datei) (Zeile n) [Incorrect parameter value (Q or R parameter file) (Line n)]	An error has occurred in line <n> due to an incorrect parameter value.
-15	Parameter nicht vorhanden (Q- oder R-Parameter-Datei) (Zeile n) [Parameter does not exist (Q or R parameter file) (Line n)]	An error has occurred in line <n>. The parameter does not exist.
-16	Falsche Systemnummer (Q- oder R-Parameter-Datei) (Zeile n) [Incorrect system number (Q or R parameter file) (Line n)]	An error has occurred in line <n>. The selected system does not exist.
-17	Falsche Achsnummer (Q- oder R-Parameter-Datei) (Zeile n) [Incorrect axis number (Q or R parameter file) (Line n)]	An error has occurred in line <n>. The selected axis does not exist.
-18	Steuerung befindet sich nicht im Resetzustand (Q- Parameter-Datei) [Controller not in reset status (Q parameter file)]	Configuration data can only be transferred if the control system is in reset status: Carry out NC reset.
-19	Es konnten nicht alle NC-Programme gelöscht werden [Not possible to delete all NC programs]	Not possible to delete running programs.
-20	MDI-Satz nicht möglich [MDI record not possible]	NC must be in an automatic mode. If an NC program is running the NC must be in NC stop status. There are also other particular causes for this error message, e.g. - Tool radius compensation active - Threading cycle active - RA, RB, RD record - OCI record active
-21	Fehler beim Teachen [Error while teaching]	Possible causes are: - Maximum record length exceeded (120 characters) - An axis in the system has not been referenced



## 14 Technical data

Power supply		
Rated voltage	DC 24 V (19,2 V .. 30 V inclusive 5% residual ripple)	
Current consumption	max. 4.5 A (incl. all digital I/O)	
Isolation (to internal electronic)	X1 (RS422)	yes
	X2 (digital I/O)	yes
	X3 USB	no
	X4 XRIO	yes
	X5 CAN	yes
	X6 ETH (Ethernet)	yes
	X7 PRG (RS232)	no
Serial interfaces		
RS 232	Programming and diagnosis X7 PRG	
RS 422	Operating devices X1	
More interfaces		
USB	Programming, diagnosis, operating interface X3	
ETH (Ethernet)	Programming, diagnosis, operating interface X6	
CAN	CANopen bus port	
XRIO	Port for the special Schleicher XRIO bus to connect the RIO EC X2 bus coupler	
Direct Inputs / Outputs (on X2 Connector)		
Outputs	DC 24 V, max. 500 mA, General Purpose	
Number	2 (I/O 0, I/O 1)	
Switching level	H level $\geq +24$ V-EXT - 0,5 V L level $\leq 1$ V	
Signal delay	<300 $\mu$ s (Hardware)	
Inputs		
Number	2 (I/O 0, I/O 1)	
Switching level	H level = +11 V bis +30 V L level = -30 V bis +5 V	
Input current	min. H level (+11 V) $\geq 2$ mA max. L level (+5 V) $\leq 2$ mA typ. (+24 V) 8 mA max. (+30 V) $\leq 15$ mA	
Signal delay	<100 ns (Hardware)	
Trigger	Edge triggering	
BUSY contact		
DC 24 V, max. 2 A, General Purpose		
Number	1	
Type	potential-free relay contact, NO	

Other technical data	
CPU	XCx500: CPU Intel StrongARM SA1110, 32 Bit, 206 MHz XCx300: CPU Intel PXA210, 32 Bit, 200 MHz
Memory	SDRAM 32 MB SRAM 1 MB FLASH (internal) 4 MB max.: 16384 kB data, 4096 kB programs PLC flags: 2048 kB flags not residual, 256 kB flags residual
Real time clock	Resolution 1s, battery-backed, calendar and leap-year
Buffer	min. 3 months (With fully charged buffer unit. The buffer unit is full charged after 4 h uninterrupted operation time of the controller.)
Buffer unit	Vanadium-Pentoxid-Lithium cell 3 V / 50 mAh + SuperCAP
Processing time pro 1000 commands	Bit 50 µs Byte / Word 30 µs

Climatic ambient conditions	
Ambient operating temperature	0 ... +55°C (class KV acc. DIN 40040), free circulation limited to 0 ... +45°C if XF-PDM or XF-PDS modules are used.
Storage temperature	-25 ... +70°C (class HS acc. DIN 40040)
Relative humidity	10 ... 95% (class F acc. DIN 40040), no condensation
Air pressure in operation	860 ... 1060 hPa

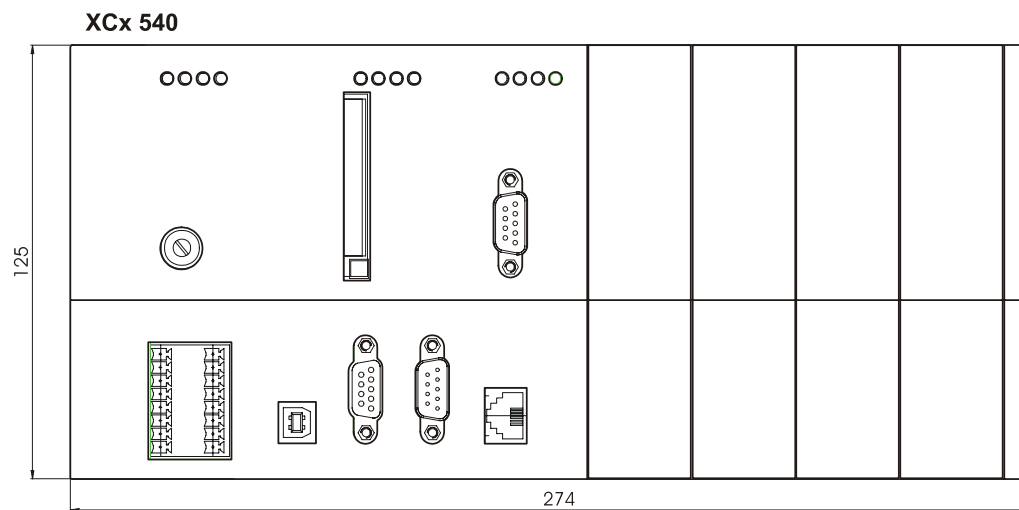
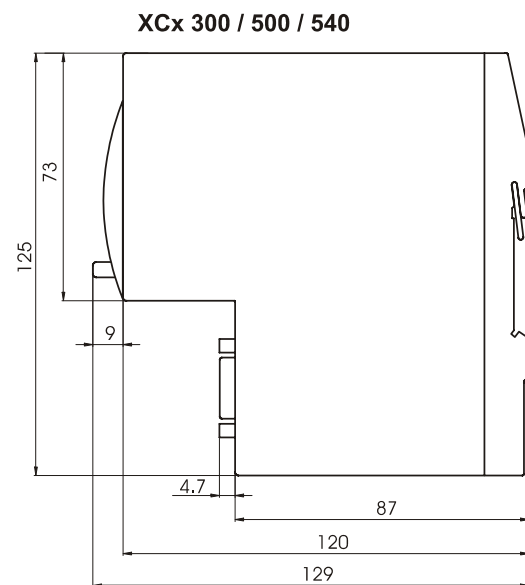
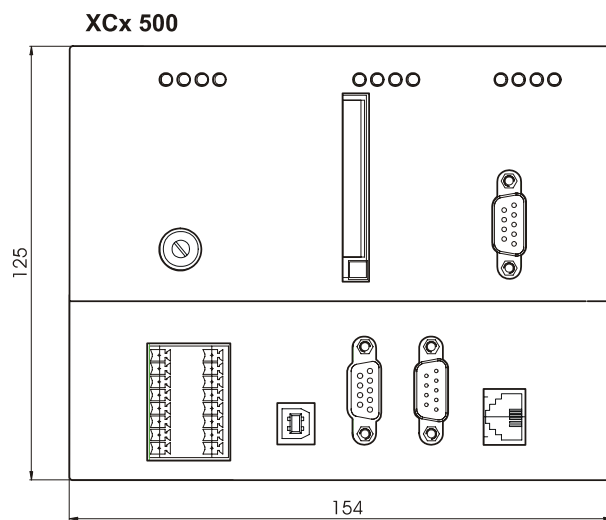
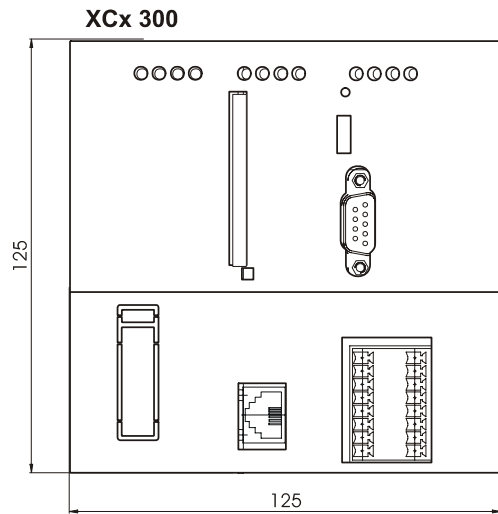
Electrical safety	
Protection type	IP 20 acc. EN 60529
Clearance / creepage distance	acc. DIN EN 61131-2 between current circuits, parts and isolated current circuits according over voltage category II, pollution severity 2
Test voltage	AC 350 V / 50Hz for rated device voltage DC 24 V

Electromagnetic compatibility (EMC)	
Electrostatic discharge	acc. EN 61000-4-2: 8 kV air, 4 kV contact
Electromagnetic fields	acc. EN 61000-4-3: field intensity 10 V / m, 80 ... 1000 MHz
Burst	acc. EN 61000-4-4: 2 kV on DC power supply lines, 1 kV on I/O signal lines
Surge	acc. EN 61000-4-5: 1 kV CM and 0,5 kV DM DC power supply lines 0.5 kV CM and 0.5 kV DM on DC I/O signal lines
Emitted interference (EMB)	acc. EN 55011: limiting value class A, group 1

Mechanical strength	
Vibration	acc. DIN EN 60068-2-6 10 ... 57 Hz constant amplitude 0,075mm 57 ... 150 Hz constant acceleration 1 g
Shock test	acc. DIN EN 60068-2-27, half sinus wave 15g / 11ms
Free fall	acc. DIN EN 60068-2-32: fall height 1m (in original package)



### 15 Dimensions



## 16 Appendix

### 16.1 Trademarks

- WINDOWS is a registered trademark of Microsoft Corporation.
- CANopen is a registered trademark of CAN in Automation e.V,
- ProCANopen is a registered trademark of Vector Informatik GmbH
- CANalyzer is a registered trademark of Vector Informatik GmbH
- Pentium is a registered trademark of Intel Corp.
- VxWorks is a registered trademark of Wind River Systems Inc.

All other trademarks or product names are registered trademarks of their respective owners.



## 17 Safety-related Information

The term automation system as used in this manual includes control units, their components (modules), other parts (such as racks, cables), operator panels, and the software used for programming, commissioning and operating the control units. This operating manual can only describe a fraction of the automation system (e.g. modules).

The technical design of SCHLEICHER automation systems is based on the EN 61131-2 (IEC 61131-2) product norm. The systems and devices have CE marking according to the EMC directive 89/336/EEC and, if applicable, the low-voltage directive 73/23/EEC.

The machinery directive 89/392/EEC is not applicable, because the safety objectives of the directive are covered by the low-voltage and EMC directives.

When SCHLEICHER automation systems are part of the electrical equipment of a machine, the manufacturer must include them in the conformity evaluation process. In this case the DIN EN 60204-1 norm must be observed (safety of machines, general requirements for electrical equipment of machines).

When an automation system is properly maintained and used for its intended purpose it will not normally cause damage to property or present health hazards. However, improper configuration, installation, maintenance or operation of the system or machine, ignoring the instructions in this manual, or intervention by insufficiently qualified personnel may result in connected actuators (such as motors, hydraulic units, etc.) becoming a source of danger.

### 17.1 Correct Use of the System

SCHLEICHER automation systems are state-of-the-art products and manufactured to recognised safety requirements. All the same, their use can cause danger to the health and safety of operators and others, or damage machines, systems or other property.

The automation system must only be used in perfect technical condition for its intended purpose, with attention given to safety and danger, and observing the operating manual. Correct transport, storage, installation, operation and maintenance of the system are all prerequisites for smooth and safe operation of the control system. Malfunctions, in particular those which may affect safety, must be immediately resolved.

Automation systems are designed exclusively to control machines and systems. Automation systems are not intended for any other use than the above. The manufacturer will therefore accept no liability for any damages resulting from the incorrect use of the systems.

When using automation systems, all instructions given in this manual regarding mechanical and electrical setup, commissioning and operation must be observed.

### 17.2 Selection and Qualification of Personnel



All configuring, programming, installation, commissioning, operation and maintenance work on the automation system must be carried out by trained personnel such as electricians or electrical engineers.

Personnel responsible for configuring and programming the system must be familiar with all safety-related issues in automation technology.

System operators must be instructed on the operation of the control system and be familiar with the relevant operating instructions.

All personnel responsible for installing, commissioning and maintaining the system must have had appropriate training qualifying them to work on automation systems.

### 17.3 Configuring, Programming, Installation, Commissioning and Operation

The automation system will in most cases be a part of a larger system in which machines are controlled. When configuring, installing and commissioning automation systems to control machines the machine manufacturer and the user must observe the safety regulations as defined in the machinery directive 89/392/EWG. For specific applications national accident prevention regulations such as VBG 4.0 will apply.

Safety-related components on the controlled machine must be designed such that they operate independently from the control system. Emergency stop components must be operational in all control modes. In an emergency stop the power supply to all switching elements controlled by the control system must be cut off. The power supply can be cut off using a safety relay such as SCHLEICHER type SNO 2002-17.

Measures must be taken for restarting an interrupted control program following voltage dips or power failures. Operating conditions should never cause danger, not even for a short time. In the event of danger the emergency stop must be immediately triggered.

In order to prevent an open-circuit in the signal circuit causing non-controllable conditions in the control system, the relevant hardware and software safety precautions must be taken for I/O interfacing. Control elements and their assigned control panel elements must be installed in a place where they are sufficiently protected against inadvertent use.

### 17.4 Maintenance

Measuring and testing on active devices must be carried out in accordance with the regulations and instructions of the VBG 4.0 accident prevention regulation. The appropriate power tools must be used.

Repairs on control components must be carried out at repair shops authorised by SCHLEICHER. Opening the components and repairs by unauthorised personnel may lead to personal injury or damage to property.

Always be disconnected the device from the mains before opening it (either disconnect the mains plug or use the cut-out switch).

Control modules may only be replaced when the power is switched off. Disassembly and assembly must be carried out according to the directives for mechanical assembly.

Fuses may only be replaced with those types specified in Technical Data.

Batteries may only be replaced with those types specified in Technical Data. Batteries must always be disposed as hazardous waste.

### 17.5 High Voltages



When the cabinet is opened or casing is removed from system components certain parts of the automation system are exposed. These parts may be subject to dangerous high voltages.

The user must prevent any unauthorised and incorrect access to the system (for example, by ensuring that the cabinet is locked).

Personnel must be familiar with all sources of danger and measures for commissioning and maintaining the system in line with the instructions given in this manual.



### **17.6 Dealing With Used Batteries**

When the batteries in the automation system are dead they must be disposed of in a battery return system or through public waste disposal facilities.

Batteries should be fully discharged before disposal. A battery is discharged when the function of the device is impaired due to insufficient battery capacity.

When batteries for disposal are not fully discharged precautions must be taken to prevent short circuits. For example by sticking tape over the poles of the battery.



## 18 Pictures and Index

Figure 1: XCx 300 Overview .....	9
Figure 2: XCx 500 Overview .....	10
Figure 3: XCx 540 Overview .....	11
Figure 4: Installation clearances .....	12
Figure 5: Fixing the XCx on DIN rail .....	13
Figure 6: Disassembly the XCx .....	13
Figure 7: Installation RIO expansion modules with XCx 300 .....	14
Figure 8: Dimensions of holes for fixing with screws .....	15
Figure 9: Connecting the voltage supply XCx 300 .....	16
Figure 10: Connecting the voltage supply XCx 500 and XCx 540 .....	17
Figure 11: End clips with XCx 300 .....	18
Figure 12: Slide contacts RIO with XCx 300 .....	18
Figure 13: Connecting the service PC to the XCx .....	28
Figure 14: XRIO connection to XCx .....	41

## Index

### A

Addressing	
XRIO I/O modules .....	64
Application example	
Web-server .....	59
Assigning	
program to task .....	95
Axis errors .....	151

### B

Browser for visualization with web server .....	59
Bus coupler RIO EC X2 .....	60

### C

CAN driver errors .....	162
CAN error messages .....	161
CAN HANDLER ERROR .....	83
CANopen .....	47
access to direct network variables .....	78
access to network variables .....	76
configuring with ProCANopen .....	52
declaring the I/O driver .....	49
error messages .....	83
Expansion Module XF-CAN .....	84
field bus card .....	52
lifeguarding .....	74
network variables .....	75
nodeguarding .....	73
parameterizing and diagnosis functions .....	80
PDO .....	73
PLC addresses .....	75
SDO .....	73
SDO function blocks .....	79
specifications .....	47
status variable .....	75
<b>CAN-specific operating system errors</b> .....	161
CF see Compact Flash .....	24
CNC	
multi-tasking system .....	108
Overview .....	107
CNC error messages .....	140
Compact Flash .....	24

Configuration .....	129
XRIO .....	61
Configuring	
CANopen with ProCANopen .....	52
Connection	
CANopen .....	25
EtherNet .....	24
programming device .....	24
supply voltage .....	26
USB .....	25
X RIO .....	25
X2 see Connection supply voltage .....	26
X3 see USB connection .....	25
X4 see X RIO connection .....	25
X5 see CANopen field bus connection .....	25
X6 ETH see EtherNet connection .....	24
X7 see Programming device interface .....	24
Connections .....	21
PROFIBUS-DP module XF-DPM .....	126
PROFIBUS-DP module XF-DPS .....	128
SERCOS module XP-SRC .....	122
SL module .....	110
Controls .....	21
Cyclical task .....	86

### D

Data transmission rate	
setting CANopen .....	81
Declaring	
I/O driver for CANopen .....	49
network variables .....	51
variables for visualization with web server .....	59
Decoder errors .....	143
Default task .....	90
Diagnosis CANopen	
Code 1 node number .....	81
Code 2 data transmission rate .....	81
code 20 set CANopen startup delay .....	82
Code 7 CANopen error number .....	82
code 8 CANopen firmware version .....	82
Dimensions	
see Installation dimensions .....	12
DIN rail installation XCx .....	13
DIN rail XCx .....	13
Displays, LEDs on front .....	22
Drive errors .....	156



Driver	
XRIO .....	65

## E

Error in interpolation preparation .....	149
Error in tool nose radius compensation .....	149
Error messages .....	131
CAN .....	161
CNC .....	140
PLC .....	133
user-defined .....	163
Error messages .....	22, 23
Error messages CANopen .....	83
Ethernet	
connection for web .....	58
TCP/IP connection for programming .....	28
Event task .....	87
Expansion modules	
XRIO .....	67
Expansion Modules	
XF-CAN .....	84
XRIO .....	71

## F

Firmware function blocks see Libraries .....	100
Firmware task priorities .....	93
Flags	
XRIO .....	64
Flash memory see Compact Flash .....	24
Following error .....	151
Function block	
DP_NET_GET_MSG .....	104
DP_NET_PUT_MSG .....	104
Function block	
CLEAR_ERROR .....	104
CO_NET_CAN_SYNC .....	102
CO_NET_GET_KERNEL_STATUS .....	102
CO_NET_GET_LOCAL_NODE_ID .....	102
CO_NET_GET_STATE .....	102
CO_NET_NMT .....	102
CO_NET_PING .....	102
CO_NET_RECV_EMU .....	102
CO_NET_RECV_EMU_DEV .....	102
CO_NET_RECV_ERR .....	102
CO_NET_RECV_ERR_DEV .....	102
CO_NET_RESTART_ALL .....	102
CO_NET_RESTART_CAN .....	102
CO_NET_SDO_READ .....	102
CO_NET_SDO_WRITE .....	102
CO_NET_SENDR .....	102
CO_NET_SHUTDOWN .....	102
CONNECT_V .....	102
DP_NET_GET_STATE .....	104
for diagnosis XRIO .....	46
GET_DATE .....	103
GET_MTS .....	104
GET_TIME .....	103
MC_ANALOG .....	103
MC_ANALOG_1_AXIS .....	103
MC_CAN .....	103
PORT_CLOSE .....	105
PORT_OPEN .....	105
PORT_READ .....	105
PORT_STATE .....	105
PORT_WRITE .....	105

PPF_COP_COMM .....	104
PUT_ERROR .....	104
PUT_ERROR2 .....	104
READ_AXIS_PAGE .....	105
READ_FILE .....	104
READ_Q_PARAM_* .....	103
READ_SERC_PARAM .....	103
SAVE_Q_PARAM_* .....	103
SAVE_R_PARAM_* .....	103
SEND_MAIL .....	104
SET_DATE .....	103
SET_SERC_PHASE .....	103
SET_TIME .....	103
UBA_ERR_CTRL .....	105
URCV_V .....	102
USEND_V .....	102
UZF_VR .....	105
WRITE_AXIS_PAGE .....	105
WRITE_FILE .....	104
WRITE_Q_PARAM_* .....	103
WRITE_SERC_PARAM .....	103
XFIO_CONFIG .....	104
XRIO_STATE .....	104
Function blocks and libraries .....	100

## G

General CNC errors .....	141
General PLC errors .....	138

## I

Installation	
XCx .....	12
Installation clearances .....	12
Installation dimensions .....	12
Installation position .....	12
IP address	
checking with ping .....	31
set/output on PC .....	30
set/output on XCx .....	30

## K

Key-operated switch see Mode switch .....	26
---	----

## L

LED displays .....	22
Library	
CANopen .....	102
CFB_Vxxx .....	102
CNC .....	103
Date_Time .....	103
MMI .....	104
PLC .....	104
Profibus_VxxxSerial .....	104
SchleicherLib .....	105
Serial .....	105
XCx7_Vxxx .....	105
Lifeguarding CANopen .....	74

## M

Mechanical installation .....	12
Mode switch .....	26

Modules	
XF-CAN Expansion Module CANopen .....	84
XRIO Expansion Module .....	71
Mounting clearances see Installation clearances .....	12
Mounting position see Installation position .....	12
Multi-tasking system	
notes on CNC .....	108

## N

Network variables	
CANopen .....	75
Node ID see node number .....	81
Node number	
setting CANopen .....	81
Nodeguarding CANopen .....	73

## O

Operating state HALT .....	97
Operating state POWER ON .....	97
Operating state STOP .....	97
Operating states on the PLC .....	97
Overview	
XCx family .....	8

## P

Parameterizing and diagnosis functions CANopen .....	80
Parameterizing CANopen	
Code 1 node number .....	81
data transmission rate CANopen .....	81
PDD Process Data Directory .....	59
PDO CANopen .....	73
ping .....	31
PLC	
basics .....	97
changing operating states .....	98
operating states .....	97
ProConOS operating system .....	106
programming .....	97
starting behaviour .....	99
system variables .....	99
PLC addresses	
CANopen .....	75
PLC addresses for CANopen .....	75
PLC error messages .....	133
PLC starting behaviour .....	99
ProCANopen	
first connection .....	54
installing .....	52
integrating in MULTIPROG .....	53
Process Data Directory .....	59
ProConOS .....	106
ProConOS.ini .....	106
ProConOS.ini initialization file .....	106
PROFIBUS-DP module XF-DPM .....	125
connections .....	126
LED displays .....	126
technical data .....	126
PROFIBUS-DP module XF-DPS .....	127
configuration data .....	129
connections .....	128
LED displays .....	128
technical data .....	128
Program	
assigning to task .....	95
Programming device interface .....	24

Programming the PLC .....	97
---------------------------	----

## R

Real time clock	
read/set function blocks .....	103
Record insertion error .....	148
RIO EC X2 for XRIO .....	60
RS232	
interface for startup .....	28

## S

Safety information	
warning signs .....	7
Safety-related information	
accident prevention regulation .....	171
commissioning .....	171
configuring .....	171
correct use of the system .....	170
emergency stop component .....	171
installation .....	171
maintenance .....	171
programming .....	171
selection of personnel .....	170
Safety-related Information .....	170
SDO CANopen .....	73
SDO function blocks .....	79
SERCOS diagnosis status .....	153
SERCOS error text, manufacturer-specific .....	155
SERCOS error, manufacturer-specific .....	155
SERCOS module XP-SRC .....	121
connections .....	122
LED displays .....	122
technical data .....	123
SERCOS status class 1 .....	154
Shared RAM	
Configuration .....	107
SL modul	
connections .....	110
SL module	
LED displays .....	110
SLM Drive Module XP-SLM	
technical data .....	113
SLM module .....	109
SPG, see system programs .....	88
Status variable	
CANopen .....	75
Subsystem-specific CNC errors .....	142
Supervisor task .....	85
sysinfo.txt configuration XRIO .....	62
System programs (SPGs) .....	88
System task .....	88
System variables on PLC .....	99

## T

Task	
creating .....	95
cyclical task .....	86
default task .....	90
event task .....	87
firmware task priorities .....	93
priorities .....	93
priority level overview .....	85
supervisor task .....	85
system task .....	88
task information .....	91



user task .....	86
watchdog .....	94
Task information.....	91
Task priorities .....	93
TCP/IP connection.....	28
Technical data.....	166
PROFIBUS-DP module XF-DPM .....	126
PROFIBUS-DP module XF-DPS .....	128
SERCOS module XP-SRC .....	123
Tool nose radius compensation error .....	149
Trademarks .....	169
Transformation errors.....	142
Transmission time .....	
XRIO .....	65

## U

USB (Universal Serial Bus).....	25
User libraries, see Libraries .....	100
User task .....	86
User-defined error messages.....	163

## V

Variable declarations .....	
in function block example programs .....	101
Variables .....	
CANopen access to direct network variables .....	78
CANopen access to network variables.....	76
declaring network variables .....	51
for visualization with web server .....	59

## W

Watchdog .....	94
Watchdog task execution .....	94

Web-server .....	
application example .....	59
declaring variables for visualization .....	59
required browser.....	59
Schleicher applet .....	58
Web-Server .....	
functions and concept.....	58

## X

X2 see Connection supply voltage .....	26
X3 see USB connection .....	25
X4 see X RIO connection .....	25
X5 see CANopen field bus connection.....	25
X6 ETH see EtherNet connection.....	24
X7 see Programming device interface .....	24
XF-DPM PROFIBUS-DP module .....	125
XF-DPS PROFIBUS-DP module.....	127
XP-SLM see SLM Module.....	109
XP-SRC SERCOS module .....	121
XRIO .....	
access to I/O level .....	60
addressing .....	64
bus coupler RIO EC X2 .....	60
Configuration, recognizing the.....	61
Driver .....	65
Expansion Modules .....	67
Expansion Module .....	71
Flags .....	64
sysinfo.txt.....	62
Transmission time.....	65
XRIO errors .....	139