

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

XIOC Signal Modules



Powering Business Worldwide

All brand and product names are trademarks or registered trademarks of the owner concerned.

Emergency On Call Service
Please call your local representative:
http://www.eaton.com/moeller/aftersales
or
Hotline After Sales Service:
+49 (0) 180 5 223822 (de, en)
AfterSalesEGBonn@eaton.com

Original Operating Instructions
The German-language edition of this document is the original operating manual.

Translation of the original operating manual
All editions of this document other than those in German language are translations of the original German manual.

1st published 2002, edition date 05/02
2nd edition 10/2002
3rd edition 04/2003
4th edition 10/2003
5th edition 12/2003
6th edition 07/2004
7th edition 09/2004
8th edition 02/2005
9th edition 11/2006
10th edition 04/2008
11th edition 10/2010
See revision protocol in the "About this manual" chapter

© Eaton Industries GmbH, 53105 Bonn

Authors: Peter Roersch
Editor: Thomas Kracht
Translator: Patrick Chadwick, David Long

All rights reserved, including those of the translation.

No part of this manual may be reproduced in any form (printed, photocopy, microfilm or any other process) or processed, duplicated or distributed by means of electronic systems without written permission of Eaton Industries GmbH, Bonn.

Subject to alteration without notice.



Danger!
Dangerous electrical voltage!

Before commencing the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally restarted.
- Verify isolation from the supply.
- Earth and short circuit.
- Cover or enclose neighbouring units that are live.
- Follow the engineering instructions (IL/AWA) of the device concerned.
- Only suitably qualified personnel in accordance with EN 50110-1/-2 (VDE 0105 Part 100) may work on this device/system.
- Before installation and before touching the device ensure that you are free of electrostatic charge.
- The functional earth (FE) must be connected to the protective earth (PE) or to the potential equalisation. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference does not impair the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that a line or wire breakage on the signal side does not result in undefined states in the automation devices.
- Ensure a reliable electrical isolation of the low voltage for the 24 volt supply. Only use power supply units complying with IEC 60364-4-41 (VDE 0100 Part 410) or HD 384.4.41 S2.
- Deviations of the mains voltage from the rated value must not exceed the tolerance limits given in the specifications, otherwise this may cause malfunction and dangerous operation.
- Emergency stop devices complying with IEC/EN 60204-1 must be effective in all operating modes of the automation devices. Unlatching the emergency-stop devices must not cause restart.
- Devices that are designed for mounting in housings or control cabinets must only be operated and controlled after they have been installed with the housing closed. Desktop or portable units must only be operated and controlled in enclosed housings.
- Measures should be taken to ensure the proper restart of programs interrupted after a voltage dip or failure. This should not cause dangerous operating states even for a short time. If necessary, emergency-stop devices should be implemented.
- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks etc.).

Contents

About this manual 7

List of revisions 7
Additional manuals 8
Target group 8
Abbreviations and symbols 8

1 Signal modules 11

Overview of the signal modules for XC-CPU100/200 11
Accessories 12
Assembly 12
PLC connection 12
Engineering notes 12
– Arrangement of the modules according to current consumption 12
– Arrangement of the modules with increased ambient temperature 12
Slot assignment in the backplanes 13
Mounting the backplane 14
– Mounting on the top hat rail 15
– Mounting on the mounting plate 15
Detaching the backplane 15
Mounting the signal modules 17
Detaching the signal modules 17
Fixing the terminal block 17
Wiring up the I/O signals 18
– Wiring up the screw terminal block 18
– Wiring up the spring-loaded terminal block 18
– Terminal capacities of the terminal blocks 18
Wiring the digital input module (24 V DC) 18
Wiring up the digital output module (24 V DC) 19
– Wiring up the relay output module 19
  RC peak-suppression filter 19
  Fuse 19
  Supply voltage for relay operation 19
– Wiring up the transistor output module 19
  Freewheel diode 19
  S and C terminals 19
Wiring of the XIOC-32DI input module and the XIOC-32DO output module 20
Wiring of the analog modules 21
– Signal selector with the analog modules 21
Connecting signal cables 22
Expansion of the XI/OC bus in the easySoft-CoDeSys 23
Dimensions 24
– Signal modules 24
– Backplane 24

2 Temperature acquisition modules 25

XIOC-4T-PT 25
– Features 25
– Wiring 26
– Data evaluation 27
1. Range: –50 to +400 °C (Pt100/Pt1000) 27
Example 1 27
Example 2 27
2. Range: –20 to +40 °C (Pt100) 28
Example 1 28
Example 2 28
– Conversion tables 28
– Fault retrieval 30
Faults that affect a single channel 30
Faults that affect more than one channel 30
XIOC-4AI-T 31
– Features 31
– Connection 31
– Configuration and Parameterization 31
Defining Measurement Parameters 31
Measurement range 31
– Diagnostics 32

3 Counter modules XIOC-…CNT-100kHz 33

Assembly 33
– RESET button on the module 33
– LED display 33
Programming 33
– Mode/operating mode switch 34
Connecting an incremental encoder to the counter input 35
– Two incremental encoders 35
Cable with attached connector for the counter module 37
– Incremental encoder with differential output 38
– Incremental encoder with NPN transistor output 38
– Incremental encoder with NPN transistor output (open-collector)38
Incremental encoder with PNP transistor output (open-collector) 38
Connecting devices to the Y outputs 38
Function summary 39
– Linear counter 39
Parameterizing the comparison value, setting module outputs 39
Overflow flag 39
Change actual value 39
Use of the reference input 39
Example of a linear counter, with the functions: 40
– Ring counter 40
Parameterizing the comparison value, setting module outputs 40
Change actual value 41
Example of a ring counter, with the functions: 41
– Additional functions for linear and ring counters 41
Counter RUN/STOP when CPU has STOP state 41
Polarity of the reference input 41
Configure counter features 42

Processing of commands 43
Set start value 43
Set end value 43
Set comparison value 43
Assign module outputs to the comparison value 1 or 2 43
Enable module output 43
Set setpoint value 44
Enable reference input 44
Enable counter input 44
Set new actual value 44
Reset Latch output and Equal flag (EQ) 44
Read out start value 44
Read out end value 44
Read out comparison value 44
Read out setpoint value 44
Read actual (= current) values 44
Read out flags 45
Clear Overflow flag 45
Clear Underflow flag 45
Read out flags 46
State display in the controller configuration 47
– FLAG summary 47
– Functional sequence for pulse processing (example) 48
Linear counter 48
Ring counter 48

4 Counter analog module XIOC-2CNT-2AO-INC 49
Features 49
LEDs 50
Programming and configuration 50
– Information exchange via the input/output image 50
Input map 50
Output image 52
Configuration of the base parameters 53
Edge evaluation of the count impulse, 1x, 2x or 4x 53
Number of reference verifications (once, permanent) 53
Output of the analog value 54
Behavior of the module with CPU RUN/STOP 54

5 Serial interface module XIOC-SER 55
Features 55
LED display 56
Design of the RS422-/RS485 interface 56
Select the module in the configurator of the easySoft-CoDeSys 56
Configuration of the interface 57
– "Transparent mode" operating mode 57
– "Suconet-K mode (slave)" operating mode 57
Master connection ↔ XIOC-SER 58
Setting the bus termination resistors 58
Configuration in the Sucosoft S40 58
Diagnostics on the master 58
Diagnostics on the slave 58
Access to the receive and send data 58

6 Telecontrol module XIOC-TC1 59
Features 59
LED display 60
Design of the RS422-/RS485 interface 60
Select the module in the configurator of the easySoft-CoDeSys 60
Configuration of the interface 61
– "Transparent mode" operating mode 61
Access to the receive and send data 61
Communications library for DNP3 protocol V1.1 61
– Prerequisites 61
– DNP3 communication and data model 61
– Function summary 63
Function DNP3_Create 66
Function DNP3_Destroy 66
Function DNP3_Execute 66
FUNCTION DNP3_OpenCom : DNP3RESULT 67
Function DNP3_CloseCom 67
Function DNP3_SetBI 67
Function DNP3_SetAI 68
Function DNP3_SetCI 68
Function DNP3_SetBIwEvent 68
Function DNP3_SetAIwEvent 69
Function DNP3_SetCIwEvent 69
Function DNP3_GetBI 69
Function DNP3_GetAI 69
Function DNP3_GetCI 70
Function DNP3_GetBO 70
Function DNP3_GetAO 70
Function DNP3_SetDbgLevel 70
– Programming 71
– FLAGs definition in DNP3 71
Binary data types flag definition 71
Flag definition for non-binary data types 71
Function code according to DNP3 level 2 72

7 Suconet K module (master) XIOC-NET-SK-M 73
Features 73
LED display 73
Design of the Suconet K (RS 485-)interface 73
Select the module in the configurator of the easySoft-CoDeSys 74
Configuration of the interface 74
Setting the bus termination resistors 74
Access to the receive and send data 74

8 PROFIBUS-DP modules XIOC-NET-DP-M / XIOC-NET-DP-S

	75
Hardware and software prerequisites	75
Features	76
– PROFIBUS-DP interface	76
– Switches for bus termination resistors	76
– Status and diagnostics display (LEDs)	76
DP module operation	77
– Download behavior	77
– Behavior after switch on of the supply voltage	77
– Behavior after RUN → STOP transition	77
– Behavior after interruption of the DP line	77
Process analysis	77
Configuration XIOC-NET-DP-S/M	77
Data exchange	78
– PROFIBUS-DP module (master) ↔ slaves	78
– PROFIBUS-DP master ↔ DP-S module	78
– XC100/XC200 ↔ DP-M module	78
XC100: cyclic data exchange	78
XC200: Periodic data exchange (monotasking)	79
	79
Determination of the bus cycle time:	79
Task control in online operation	80
Response time on PROFIBUS-DP	80
XC200: multitasking mode	80
XC100: status indication of the PROFIBUS-DP slave	81
Example: Data transfer XC200 (master) ↔ XC100 (slave)	81
Diagnostics of the PROFIBUS-DP slaves	83
– Implement diagnostics	83
– Diagnostics data evaluation	84
Monitoring data exchange	84
– Coarse diagnostics with variable from GETBUSSTATE type	84
Create variables of the GETBUSSTATE type	84
– Detailed diagnostics with DIAGGETSTATE function block85	
Inputs/outputs of the DIAGGETSTATE function block	86
Diagnostics in the slave control	88
– Query master and connection status	88
– Diagnostic module "xDPS_SendDiag"	88
Meanings of the operands	88
Description	88
Application example for sending diagnostics data (with the xDPS_SendDiag function block)	89
Program example for diagnostics in the master control	91
– Create configuration	91
Configuration of the XIOC-NET-DP-M	91
Configure XION station	92
Configuration of the EM4/LE4 module	92
– Structure of the program example with a master	92
– Function of the program example	93
– Function of the diagnostics program	93
– Function of the data exchange (monitoring)	93
– Program example for diagnostics with a master	94
Global variable declaration	94
PROGRAM PLC_PRG	94
PROGRAMM DIAG_DP	94
Parametric programming of the LE4 with analog inputs/outputs	96

9 Technical data 97

XControl 97
Digital input modules 98
Digital output modules 100
– Transistor output modules 100
– Relay output module 101
Digital input/output modules 102
– Configuration and programming of the digital inputs/outputs 102
Analog input modules 104
Analog output module 105
Analog input/output modules 107
Temperature acquisition module XIOC-4T-PT 109
Temperature acquisition module XIOC-4AI-T 110
Counter module 111
Counter analog module 112
Serial interface module/Telecontrol module 113
Suconet-K module (master) 114
PROFIBUS-DP module 114

Index 115

About this manual

List of revisions

The following significant amendments have been introduced since previous issues AWB2725-1452G:

Publication date	Page	Key word	New	Modification	Deleted
10/02	33	Counter modules XIOC-…CNT-100kHz		✓	✓
	102	Digital input/output modules	✓		
04/03	18	Terminal capacities of the terminal blocks		✓	✓
	97, 98, 100, 101, 102	Technical data		✓	✓
	102	Configuration and programming of the digital inputs/outputs	✓		
10/03	11, 107	Analog input/output modules	✓		
	99	XIOC-16DI-110VAC			
	107	Note			
12/03	11, 12, 18, 19, 20, 24, 98, 100, 107	XIOC-32DI/ XIOC-32DO	✓		
	13, 14	XIOC-BP-EXT	✓		
	49, 112	XIOC-2CNT-2AO-INC			
	55, 113	XIOC-SER	✓		
04/04	11, 21, 108	XIOC-2AI-1AO-U1-I1 XIOC-4AI-2AO-U1-I1	✓		
	33	Programming	✓		
	50	Programming and configuration	✓		
	57	Gap-Time	✓		
07/04	75, 113	PROFIBUS-DP modules XIOC-NET-DP-M / XIOC-NET-DP-S	✓		
09/04	55	XIOC-SER module Suconet-K mode (Slave)	✓		
02/05	20	Wiring XIOC-32DI/DO, conductor colour	✓		
	73, 113	Suconet K module (master) XIOC-NET-SK-M	✓		
	75, 113	XIOC-NET-DP-S	✓		
11/06	31, 110	XIOC-4AI-T	✓		
11/06, unchanged editing date	31	Note	✓		
	32	Assignment of the diagnostics information			
	110	Technical data		✓	✓
04/08	57	"Suconet-K mode (slave)" operating mode, Parameterization		✓	✓
	58	Configuration in the Sucosoft S40			
07/10	General	XIOC-16DO-S deleted		✓	✓
	11, 12, 18, 97, 98	XIOC-16DI/XIOC-8DI		✓	
	11, 12, 19, 93, 100	XIOC-16DO/XIOC-8DO		✓	
	59	XIOC-TC1	✓		
10/10	General	Changeover to Eaton document numbers		✓	

This manual describes the XIOC signal module for the XC-CPU100/200 expandable PLC types.
In Chapter 1 you will find information on mounting and wiring, which is applicable to all the signal modules.
Chapter 9 provides comprehensive technical data.
This chapter also starts with a general section.

Specific features are then dealt with separately or where it proves to be more useful, combined in groups. The other chapters contain product specific information which applies to the modules.

Additional manuals

The PLC types used in conjunction with the signal modules are described in the following manuals:

PLC type	Manual No.
XC-CPU100	MN05003004Z-EN (previously AWB2724-1453GB)
XC-CPU200	MN05003001Z-EN (previously AWB2724-1491GB)
XC-CPU600	AWB2700-1428GB

The manuals are also available online as PDF files at: http://www.eaton.com/moeller → **Support**

Enter the above mentioned manual number in order to find it quickly.

Target group

Read this manual carefully, before you install the signal module and start using it. We assume that you are familiar with basic physical concepts and are experienced in reading technical drawings and dealing with electrical equipment.

Abbreviations and symbols

The abbreviations and symbols used in this manual have the following meanings:

I/O	Input/Output
PLC	Programmable Logic Controller
I_o	Input current
I_1	Output current
U_o	Input voltage
U_1	Output voltage

In Chapter 3 Counter modules XIOC-…CNT-100kHz there is an "n" in the designation for several function block inputs and outputs. This "n" is a wildcard. For example, the designation "CounternEnable" for the inputs "Counter1Enable" and "Counter2Enable" of the "CounterControl" function block.

All dimensions are in millimeters, unless otherwise specified.

▶ Indicates instructions on what to do

→ Draws your attention to interesting tips and supplementary information

Caution!
warns of the risk of material damage.

Warning!
Indicates the risk of major damage to property, or slight injury.

Danger!
Indicates the risk of major damage to property, or serious or fatal injury.

For greater clarity, the name of the current chapter is shown in the header of the left-hand page and the name of the current section in the header of the right-hand page. Exceptions are the first page of each chapter, and empty pages at the end.

1 Signal modules

Overview of the signal modules for XC-CPU100/200

Designation	Type	Technical data
Backplane	XIOC-BP-XC	For CPU with power supply unit
	XIOC-BP-XC1	For CPU with power supply unit, 1 signal module
	XIOC-BP-2	For 2 signal modules
	XIOC-BP-3	For 3 signal modules
	XIOC-BP-EXT	I/O module for expansion
Digital input module	XIOC-8DI/-16DI/-32DI	8 channels/16 channels, 32 channels 24 V DC
	XIOC-16DI-110VAC	16 channels, 110 to 120 V AC
	XIOC-16DI-AC	16 channels, 200 to 240 V AC
Digital output module	XIOC-8DO/16DO	8 channels/16 channels, transistor output 24 V DC (source type)
	XIOC-32DO	32 channels, transistor output 24 V DC (source type)
	XIOC-12DO-R	12 channels, relay output
Digital input/output module	XIOC--16DX	16 input channels, 24 V DC 12 output channels, transistor output 24 V DC (source type)
Analog input module	XIOC-8AI-I2	Current input (channels 0 to 7) 4 to 20 mA, 12 bit
	XIOC-8AI-U1	Voltage input (channels 0 to 7) 0 to 10 V DC,12 bit
	XIOC-8AI-U2	Voltage input (channels 0 to 7) –10 to +10 V DC,12 bit
	XIOC-4T-PT	PT100/1000 input (channels 0 to 3) 15 bit, signed
	XIOC-4AI-T	4 analog inputs for thermocouples (channels 0 to 3) 15 bit, signed
Analog output module	XIOC-2AO-U1-2AO-I2	Voltage output (channels 0 to 1) 0 to 10 V DC, Current output (channels 2 to 3) 4 to 20 mA, 12 bit
	XIOC-2AO-U2	Voltage output (channel 0 + 1) –10 to 10 V DC
	XIOC-4AO-U2	Voltage output (channels 0 to 3) –10 to 10 V DC
	XIOC-4AO-U1	Voltage output (channels 0 to 3) 0 to 10 V DC
Analog input/output module	XIOC-4AI-2AO-U1	Voltage input (channels 0 to 3) 0 to 10 V DC, 14 bit Voltage output (channels 0 to 1) 0 to 10 V DC, 12 bit
	XIOC-2AI-1AO-U1	Voltage input (channels 0 to 1) 0 to 10 V DC, 14 bit Voltage output (channel 0) 0 to 10 V DC, 12 bit
	XIOC-4AI-2AO-U1-I1	Voltage input (channels 0 to 3) 0 to 10 V DC, 14 bit or current input (channels 0 to 3) 0 to 20 mA, 14 bit Voltage output (channels 0 to 1) 0 to 10 V DC, 12 bit or current output (channels 0 to 1) 0 to 20 mA, 12 bit
	XIOC-2AI-1AO-U1-I1	Voltage input (channels 0 to 1) 0 to 10 V DC, 14 bit or current input (channels 0 to 1) 0 to 20 mA, 14 bit Voltage output (channel 0) 0 to 10 V DC, 12 bit or current output (channel 0) 0 to 20 mA, 12 bit
Counter module	XIOC-1CNT-100kHz	1 channel, Input for fast counter, maximum frequency 100 kHz, switchable 1/2-phase, 2 open-collector outputs
	XIOC-2CNT-100 kHz	2 channels, Input for fast counter, maximum frequency 100 kHz, switchable 1/2-phase, 2 open-collector outputs per channel
Counter analog module	XIOC-2CNT-2AO-INC	Input for fast counters, maximum frequency of 400 kHz; 2 channels, output –10 to +10 V
Serial interface module	XIOC-SER	Serial interface, selectable: RS 232, RS 422, RS 485, SUCONET K mode (slave)
Telecontrol module	XIOC-TC1	Transparent, MODBUS, Master/Slave, SUCOM-A, DNP3

Accessories

Designation	Type	Comments
Spring-cage terminals	XIOC-TERM-18T	For digital and analog I/O modules
Screw terminals	XIOC-TERM-18S	
Plug/cable	XIOC-TERM32	For 32-pole digital input/output modules

Assembly

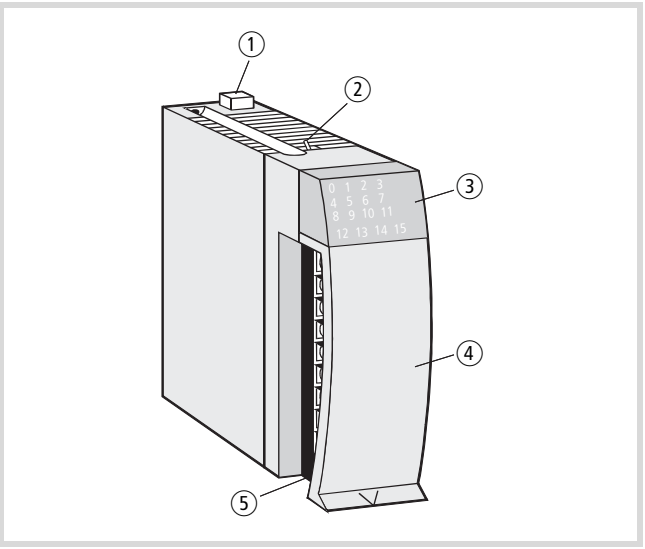


Figure 1: Assembly of a signal module

① Interlock

② LED changeover switch for XIOC-32DI/XIOC-32DO; the modules are equipped with 16 LEDs for displaying the input/output (I/O) display state. Depending on the position of the changeover switch, the LEDs indicate the I/O's 0 – 15 (switch at front) or 16 – 31 (switch at rear). The LED designated with "+" lights up when I/O 16 – 31 are displayed.

③ LED display

④ I/O cover

⑤ Terminal block

PLC connection

The XI/OC modules are the I/O modules for the XC-CPU100/200 PLC types. The following diagrams show the assembly of XI/OC modules which are connected to a PLC.

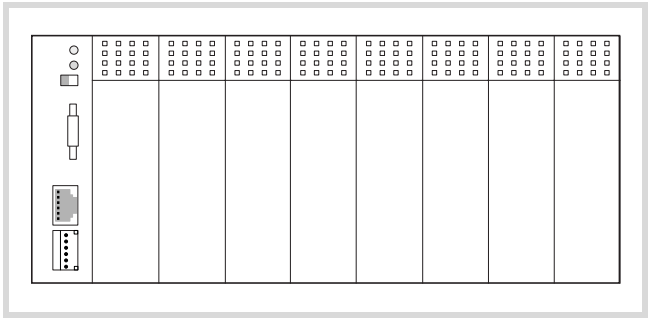


Figure 2: XC-CPU100/200 with XI/OC signal modules

Engineering notes

Arrangement of the modules according to current consumption

The CPU supplies other XI/OC modules from its integrated power supply unit. Generally, these modules should be arranged so that the modules with the higher internal current consumption (e.g. XIOC 2CNT-…) are connected first to the CPU. The modules with a lower current consumption should then follow.

Arrangement of the modules with increased ambient temperature

If the modules are used in ambient air temperature > 40° C or with limited convection (e.g. enclosed CI enclosure), measures should be implemented to prevent excessive rises in heat dissipation. This can be achieved by derating certain modules.

Module type	Technical features	Limit value at …	
		< 40 °C	> 40 °C
XIOC-16DI-AC	Simultaneity factor	1	0.75
XIOC-16DO	Rated operational current per common potential terminal	8 A	8A
XIOC--16DX	Simultaneity factor	1	0.5
	Module arrangement	any	1)

1) Locate not directly beside CPU and not directly beside further XIOC-16DX

Further details concerning engineering can be found in the manuals:

- XC-CPU100: MN05003004Z-EN (previously AWB2724-1453GB)
- XC-CPU200: MN05003001Z-EN (previously AWB2724-1491GB)

Slot assignment in the backplanes

The XI/OC modules are plugged onto backplanes that provide the connection to the PLC. The modules are also interconnected through the backplane.

The integrated bus system ensures interference-free transmission between the individual slots on the bus. In addition, the bus system supplies the individual modules with the voltage that is required for internal signal processing.

The supply voltage for the I/O electronics is applied directly to the corresponding I/O modules.

Five different backplanes are available: Four different backplanes are available:

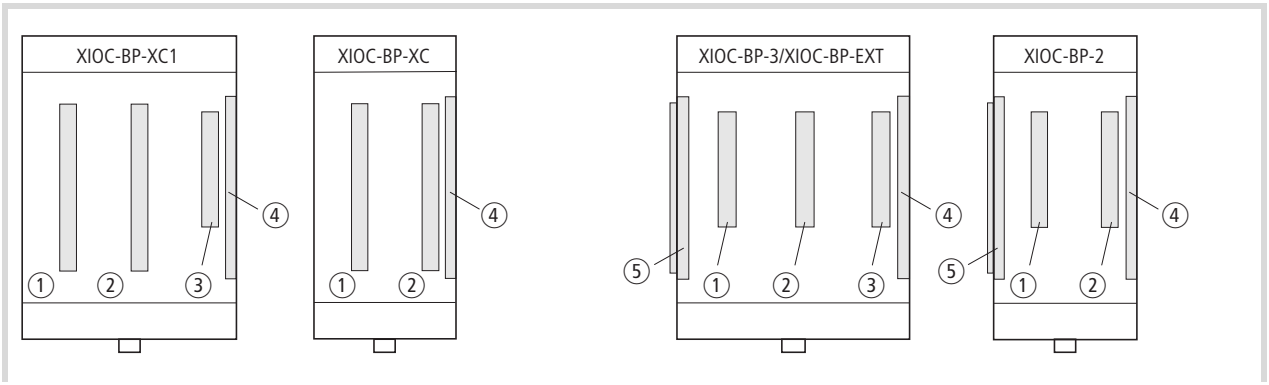
As a rule, the first backplane, which is used to take the XC-CPU100/200 CPU type is a basic backplane. You can add on several expansion backplanes to the right side. The backplanes must be arranged so that one CPU module for basic expansion and a maximum of seven XI/OC signal modules can be planned (→ fig. 4).

Through the use of bus expansion, you can add further backplanes consisting of CPU and 5, 6 or 7 I/O modules to the basic expansion. The bus expansion has the same design and the same dimensions as the XIOC-BP-3 expansion backplane. However, it is equipped with additional components for amplification of the bus signals. The arrangement of the bus expansion with the basic expansion is fixed (→ fig. 4). The maximum expansion stage can accept 15 XIOC I/O modules.

→
- If you wish to expand existing basic expansion with 6 or 7 I/O modules, you will need to replace an existing rack (backplane) (XIOC-BP-2/XIOC-BP-3) by a bus expansion (XIOC-BP-EXT). The bus expansion may only be positioned at the position indicated in Figure4.
- In the PLC Configuration, the 7th element "EXTENSION-SLOT[SLOT]" with the "Replace element" function is to be replaced by the "EXTENSION-SLOT" element. A total of up to 15 slots are indicated.

Table 1: Slot assignment in the backplanes

Backplane	Slots		
	1	2	3
XIOC-BP-XC (Basic backplane)	CPU with power supply unit		–
XIOC-BP-XC1 (Basic backplane)	CPU with power supply unit		I/O module
XIOC-BP-2 (ExpansionRack)	I/O module		–
XIOC-BP-3 (ExpansionRack)	I/O module		
XIOC-BP-EXT (bus expansion)	I/O module for expansion		



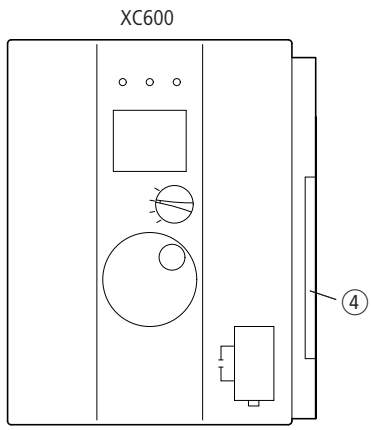


Figure 3: Top left: expandable backplane
Top right: expandable backplane

① Slot 1
② Slot 2
③ Slot 3
④ Bus expansion connector (socket)
⑤ Bus expansion connector (plug)

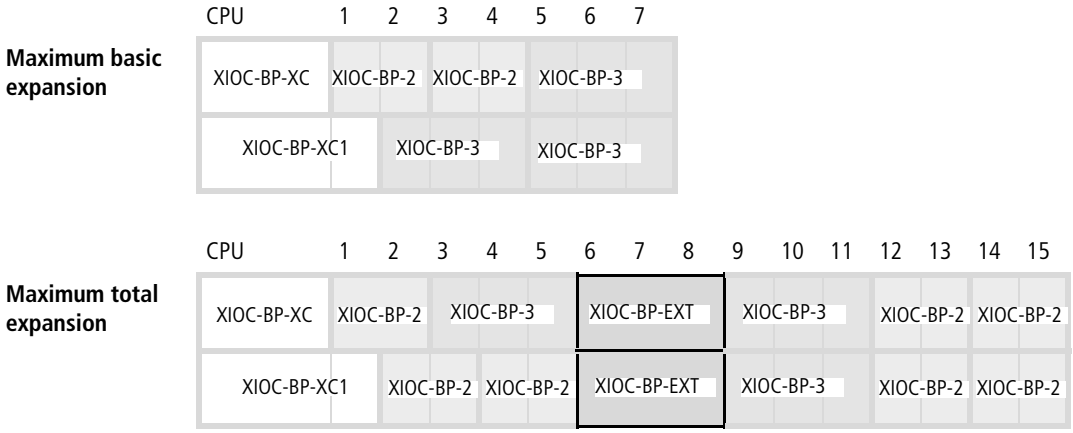


Figure 4: Maximum expansion of the I/O modules without and with XI/OC bus expansion

How to implement the software bus expansion in the PLC configurator of the easySoft-CoDeSys is described from Page 23.

Mounting the backplane

The backplane can either be snapped onto a top hat (DIN) rail, or screwed directly onto the mounting plate.

Warning!
The expansion module rack must only be plugged in or pulled out when the power is switched off. First detach the CPU or I/O modules that were plugged into the module rack. Discharge yourself from any electrostatic charge before touching electronic modules. Voltage peaks on the bus connector may cause malfunction or damage to the modules.

→ Mounting of the controls is described in:

- MN05003004Z-EN (previously AWB2724-1453GB) for XC-CPU100
- MN05003001Z-EN (previously AWB2724-1491GB) for XC-CPU200

Mounting on the top hat rail

- ▶ Use a screwdriver to pull out the locking bar until the catch snaps into position. The locking bar is then held in this position 1 .
- ▶ Place the backplane on the top hat rail so that the top edge of the rail fits into the slot, and then slide the backplane into the correct position 2.
- ▶ Press down the catch of the locking bar. The bar snaps in behind the edge of the top-hat rail. Check that the backplane is firmly seated 3.
- ▶ If you want to fit an expansion backplane: push it to the left, until the bus connector of the expansion backplane can be plugged into the bus connector socket of the basic rack or expansion backplane. Take care that the bus connectors of the backplanes are completely engaged, in order to ensure reliable electrical contact.

Mounting on the mounting plate

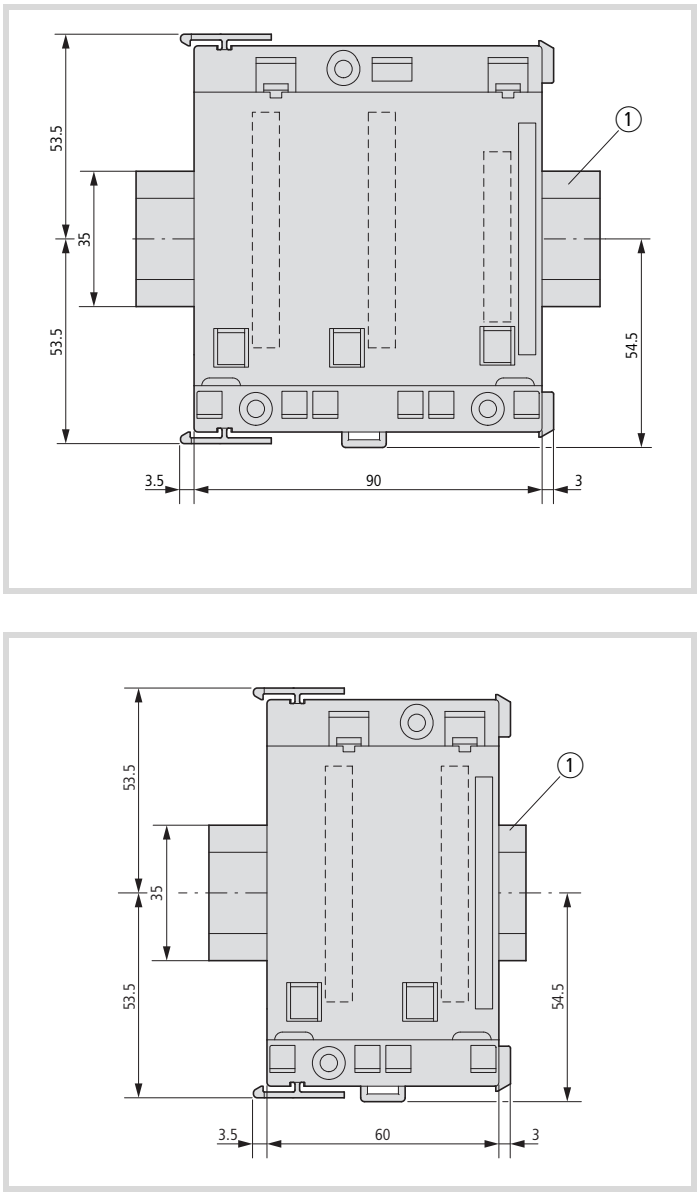
The spring contacts that protrude from the back of the backplane are intended to provide a ground for the modules. They must have a reliable electrical contact with the mounting plate.

Take care that the contact areas are protected from corrosion and – if you are using painted mounting plates – that the paint layer is removed from the contact areas.

- ▶ Plug the bus connector of the expansion backplane into the bus connector of the basic rack or expansion backplane. Take care that the bus connectors of the backplanes are completely engaged, in order to ensure reliable electrical contact.

Detaching the backplane

- ▶ Use a screwdriver to pull out the locking bar until the catch snaps into position.
  The locking bar is then held in this position 1 .
- ▶ Only with expansion backplanes: Slide the expansion backplane along the top hat rail to the right until the bus connectors are disengaged.
- ▶ Take the backplane off the rail.



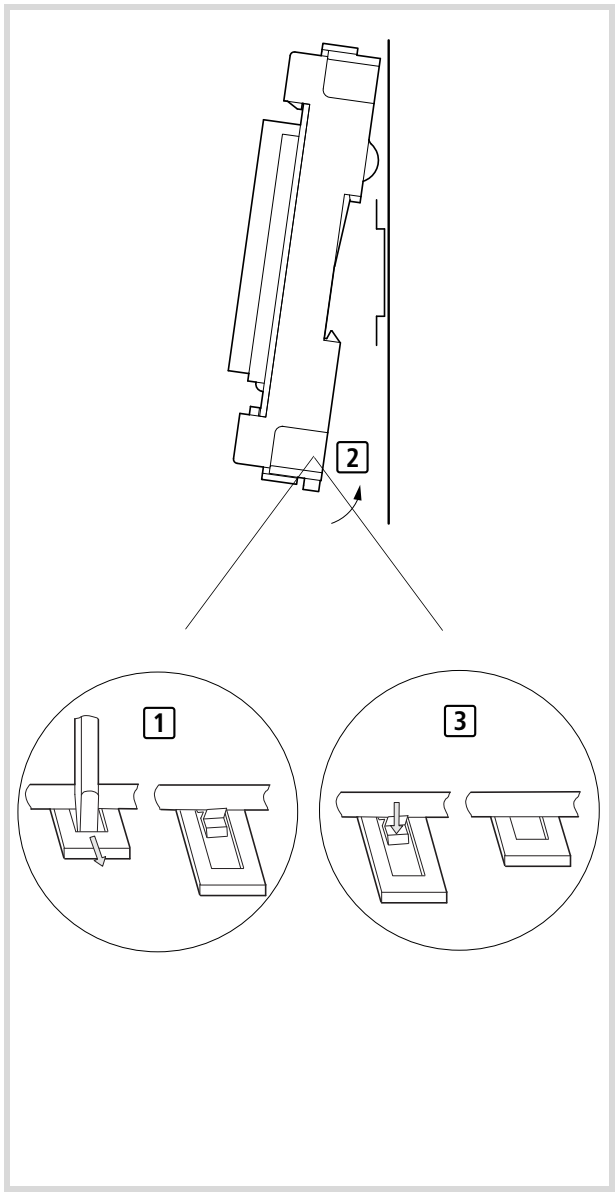


Figure 5: Mounting on a 35 mm top hat (DIN) rail,
top left: XIOC-BP-XC1, (XIOC-BP-3)
bottom left: XIOC-BP-XC, (XIOC-BP-2)

① 35 mm top hat rail

See also dimensions on Page 24.

Mounting the signal modules

- ▶ Insert the loop on the bottom of the module into the hole in the backplane [1].
- ▶ Press the top of the module onto the backplane, until you hear it click into position [2].

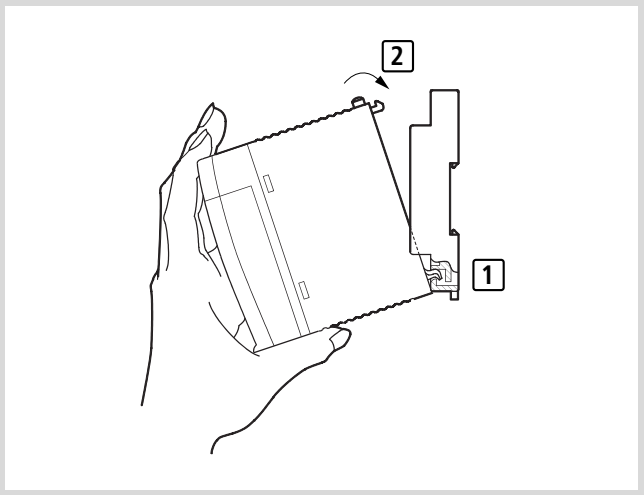


Figure 6: Mounting the signal modules

Detaching the signal modules

- ▶ Press in the catch [1].
- ▶ Keep the catch pressed in and pull the top of the module forwards [2].
- ▶ Lift up the module and remove it [3].

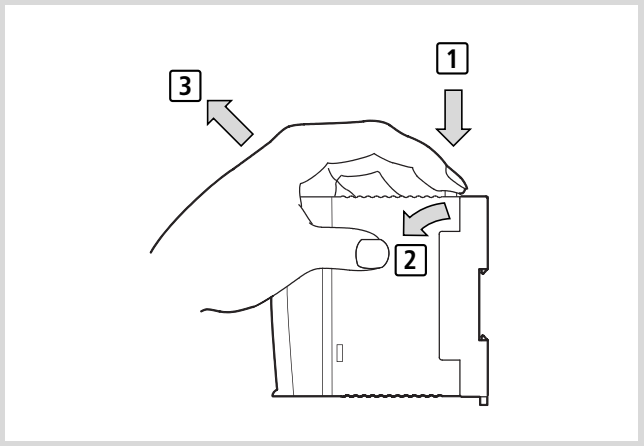


Figure 7: Detaching the modules

Fixing the terminal block

- ▶ Plug the lower end of the terminal block onto the module board. Screw in the fixing screw a short way [1].
- ▶ Push the top end of the terminal block onto the module until you hear it snap into position [2].
- ▶ Hold the top end of the terminal block firmly, and tighten up the fixing screw [3].
- ▶ Tug on the top end of the terminal block, to check that it is firmly seated and cant come loose [4].

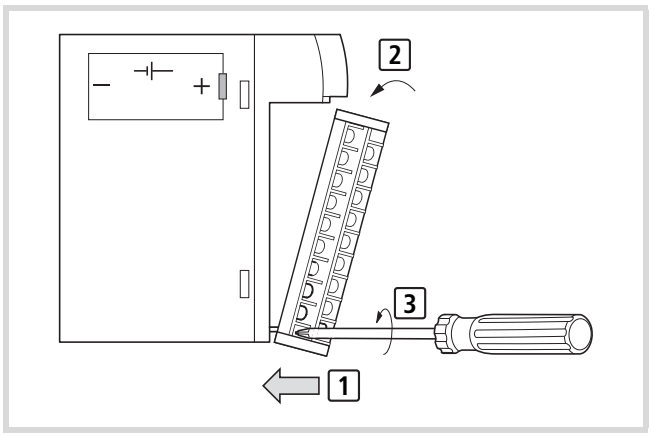


Figure 8: Fixing the terminal block

Wiring up the I/O signals

Wiring up the screw terminal block

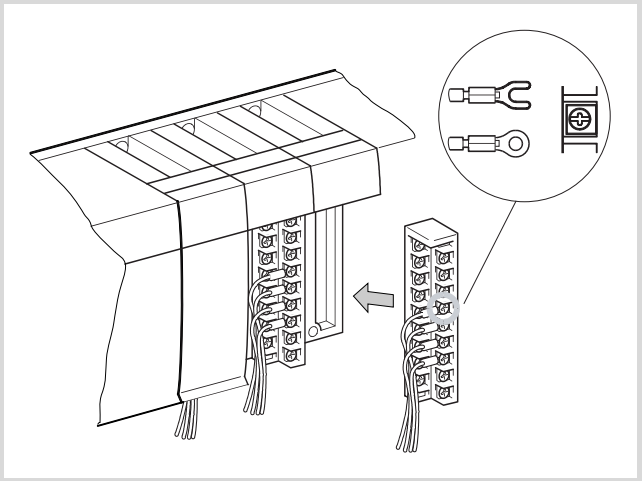


Figure 9: Wiring up the screw terminal block

→ Please observe the following notes:

- All terminals have M3 screws.
- Tighten up the screws to a torque of 0.71 to 1.02 Nm.
- If cable lugs are to be used they may have a maximum external diameter of 6 mm.
- Do not attach more than 2 cable lugs to one terminal.
- Use a cable with a maximum conductor cross-section of 0.75 mm^2 or 0.5 mm^2, if two cable lugs are going to be fixed to the same terminal.

Wiring up the spring-loaded terminal block

The spring-loaded terminal block has the same basic design as the screw terminal block. The difference lies in the way the cable is connected.

▽ **Caution!**
The cables are to be inserted into the terminals with out the use of ferrules or cable lugs.

Terminal capacities of the terminal blocks

▽ **Caution!**
For UL applications, the power supply cables to the XIOC-8DO, -16DO, -12DO-R, -16DX modules must have a cross-section of AWG16 (1.3 mm^2).

Table 2: Cable connection

Conductor	Screw connection	Spring-loaded connection
solid core	0.5 to 2.5 mm^2	0.14 to 1.0 mm^2
flexible with ferrule	0.5 to 1.5 mm^2	The cables are to be inserted into the terminals with out the use of ferrules or cable lugs.
stranded	–	0.34 to 1.0 mm^2

Wiring the digital input module (24 V DC)

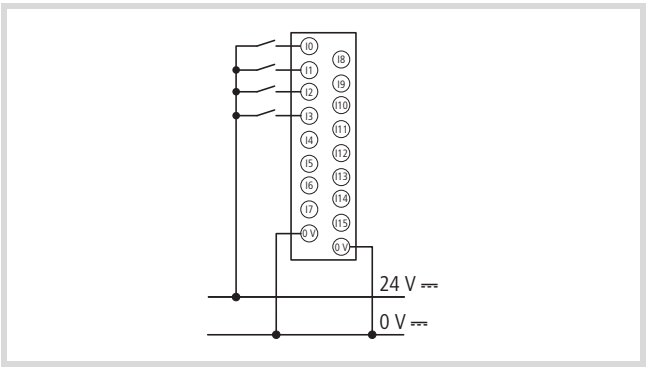


Figure 10: Example of external wiring for the DC input XIOC-8DI/16DI/32DI (here 16 DI)

- When an ON signal is applied to all input terminals, the current drawn via the input contacts is typically 4 mA.
- Sensors, such as proximity switches or photoelectric switches, can be directly attached, provided that they are current-sinking types (open-collector). Sensors that have a voltage output must be connected to the inputs via transistors.
- Use cables with a maximum length of 30 meters.

Wiring up the digital output module (24 V DC)

Wiring up the relay output module

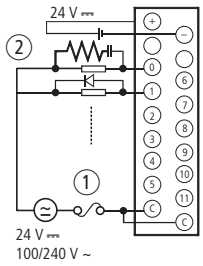


Figure 11: External wiring of the relay output XIOC-12DO-R

① Fuse

② RC peak-suppression filter or diode

RC peak-suppression filter

▶ When an inductive load is present, wire an RC peak-suppression filter (capacitor 0.1 µF and resistor about 100 Ω) parallel to the load. For DC loads, freewheel diodes must be used.

Fuse

▶ There is no fuse inside the module. Fit a 6 A fuse in the circuit (common) to protect the external wiring from being burnt out.

Supply voltage for relay operation

▶ Observe the polarity of the 24 V DC connection. Incorrect wiring can damage the internal circuitry.

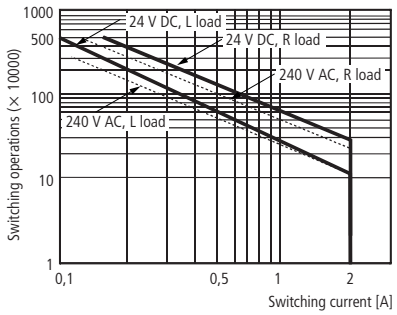


Figure 12: Operating life diagram for the relay contacts

The operating life of a contact is inversely proportional to the square of the current. Any overload currents that occur, or directly connected capacitive loads, can therefore drastically reduce the operating life of a relay.

The transistor output module is to be preferred for high-frequency switching operations.

Wiring up the transistor output module

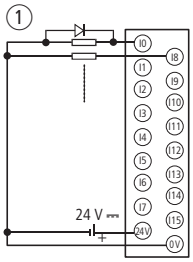


Figure 13: External wiring of the transistor output XIOC-8DO/-16DO/32DO, here: 16DO (positive logic, source type)

① Diode

Freewheel diode

▶ When using inductive loads, connect a freewheel diode in parallel.

S and C terminals

Always connect up the S and C terminals. If the module is operated without these terminals being connected, then the freewheel diodes cant carry out their function, and there is a danger that the module will not function correctly, or may even be damaged.

Wiring of the XIOC-32DI input module and the XIOC-32DO output module

The modules have a 40-pole plug connector. Connect the module with external terminals via the plug with connected cable (XIOC-TERM32). The number of the connector pin can be seen in the following diagram. Verify the assignment of conductor – connector pin (number). The cross-section of the conductors is 0.4 mm.

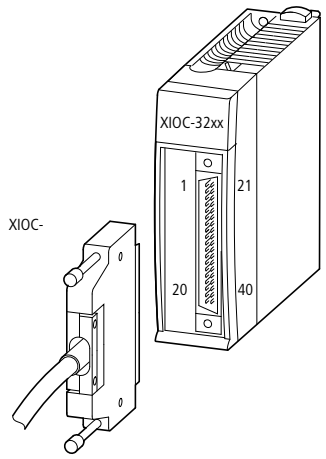


Figure 14: Cable with connector (XIOC-TERM32)

No.	Conductor colour	Signal name XIOC-32DI	Signal name XIOC-32DO
1	white	0	0
2	brown	1	1
3	green	2	2
4	yellow	3	3
5	grey	4	4
6	pink	5	5
7	blue	6	6
8	red	7	7
9	black	C	C
10	purple	8	S
11	grey/pink	9	8
12	blue/red	10	9
13	white/green	11	10
14	brown/green	12	11
15	white/yellow	13	12
16	yellow/brown	14	13
17	white/grey	15	14
18	grey/brown	C	15
19	white/pink	---	C
20	pink/brown	---	S

No.	Conductor colour	Signal name XIOC-32DI	Signal name XIOC-32DO
21	white/blue	16	16
22	brown/blue	17	17
23	white/red	18	18
24	brown/red	19	19
25	white/black	20	20
26	brown/black	21	21
27	grey/green	22	22
28	yellow/grey	23	23
29	pink/green	C	C
30	yellow/pink	24	S
31	green/blue	25	24
32	yellow/blue	26	25
33	green/red	27	26
34	yellow/red	28	27
35	green/black	29	28
36	yellow/black	30	29
37	grey/blue	31	30
38	pink/blue	C	31
39	grey/red	---	C
40	pink/red	---	S

Wiring of the analog modules

- ▶ Only use shielded cables for connection to external equipment.
- ▶ Route the cables separately from power leads or signal cables that carry differential voltages.
- ▶ Depending on the prevailing electromagnetic environment, one or both ends of the shielding should be grounded.
- ▶ Lay the AC supply power cables in separate ducts to those used for signal or data cables.
- ▶ Lay signal and data cables as close as possible to the grounded surfaces of the switchgear cabinet.

Signal selector with the analog modules

You can set the "voltage" or "current" signal types for each input and output with the XIOC-2AI-1AO-U1-I1 and XIOC-4AI-1AO-U1-I1 analog modules. The setting is implemented via the 6-pole DIP switch. In the factory default state all input and output switches are set to facilitate the processing of voltage signals. The characteristics of the inputs and outputs can be viewed in the technical data → page 108.

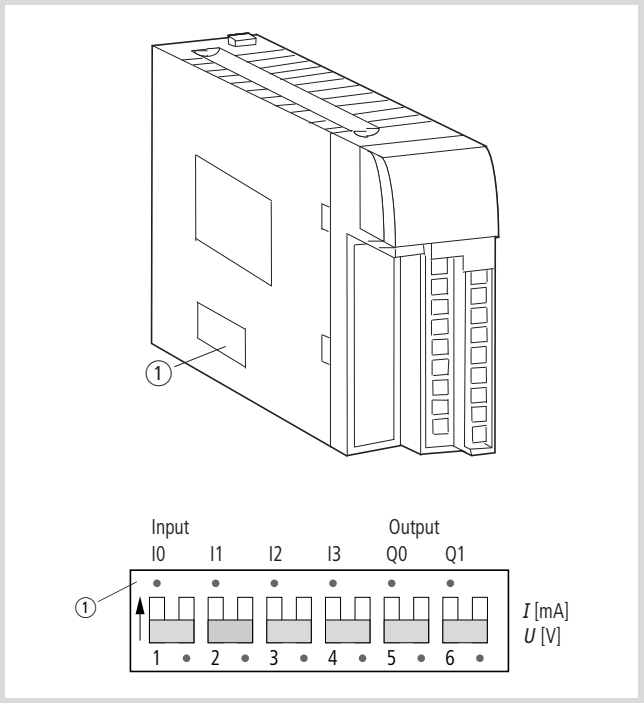
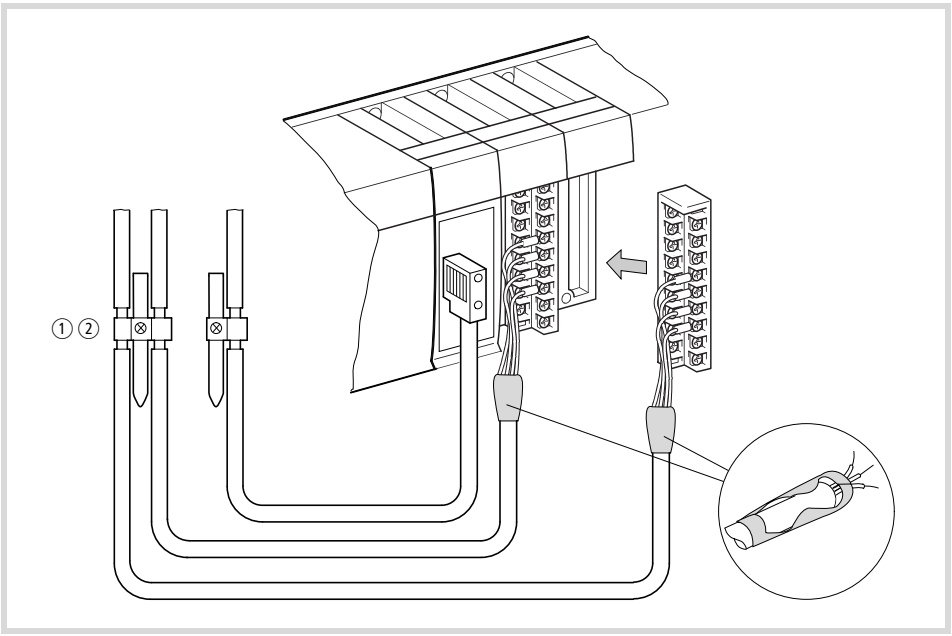


Figure 15: DIP switch for setting the "voltage" (U) or "current" (I) signal type

The "voltage" factory default state is set in the figure.

Connecting signal cables



End of the screened cables:

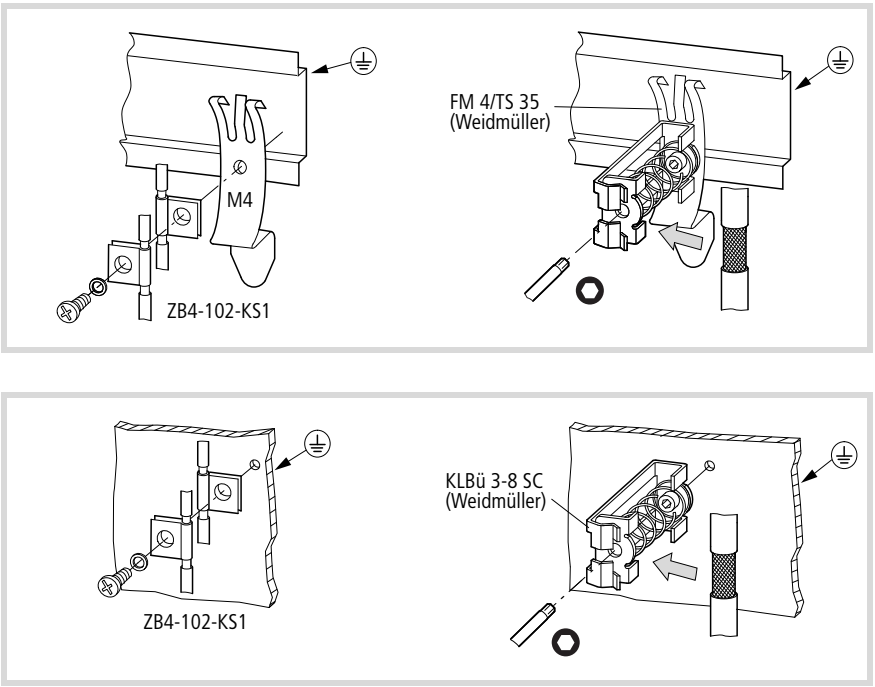
- ▶ Strip back the screen at the end of the cables and insulate it, e.g with heat shrink.

Figure 16: Shielding of signal cables, overview

① Screen earth kit for top-hat rail

② Screen earth kit for mounting plate

→ Detailed view in Figure17



- ▶ Remove the cable sheath in the contact clamp area.
- ▶ Place one contact clamp on each stripped section of the signal cables or press the stripped section into the snap fastener of the clamp strap.
- ▶ Connect the contact clamp or the clamp strap with a low-impedance connection to the top-hat rail or mounting plate.
- ▶ Attach the top-hat rail to the mounting plate.
- ▶ Ensure that all the contact areas are protected from corrosion and – if you are using painted mounting plates – that the paint layer is removed from the contact areas.
- ▶ Earth the mounting rail using as large a surface as possible.

Figure 17: Screen earth kit for top-hat rail (top) or mounting plate (bottom) with contact clamp or wire clamp, detailed view

Expansion of the XI/OC bus in the easySoft-CoDeSys

The bus expansion with the XIOC-BP-EXT backplane to a maximum of 15 slots is implemented on the software side in the PLC configuration of the easySoft-CoDeSys.

→ In total, a maximum of 15 slots are possible with an XC100/XC200 PLC → figure 4 on Page 14.

When creating a new configuration, the first 7 slots are created as EMPTY-SLOTs. Slot 7 can be replaced by an EXTENSION-SLOT. This allows the creation of a new node which enables expansion of up to 15 EMPTY-SLOTs.

The expansion backplane can be integrated as follows:

- ▶ Open the PLC Configurator
- ▶ Click with the right mouse button in the last EMPTY-SLOT.
- ▶ Select the "Replace element" command.
- ▶ Select EXTENSION-SLOT with a double-click in a new window.

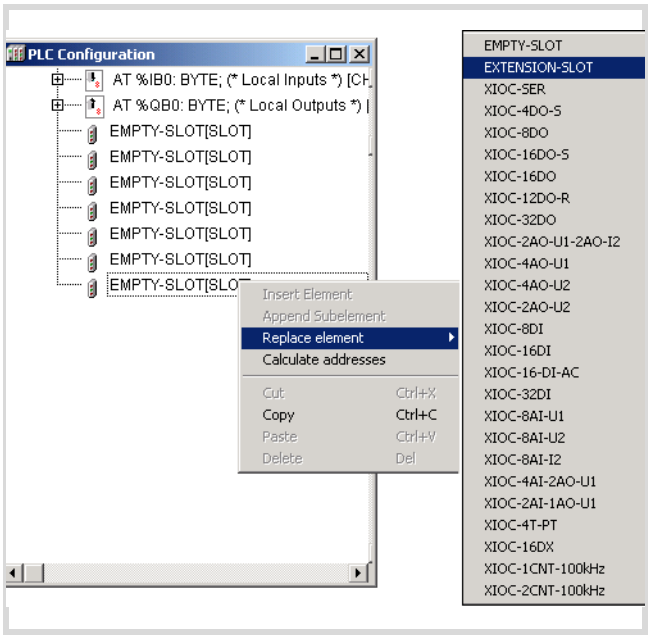


Figure 18: Expansion backplane configuration

The following illustration indicates the maximum configuration of the I/O slot.

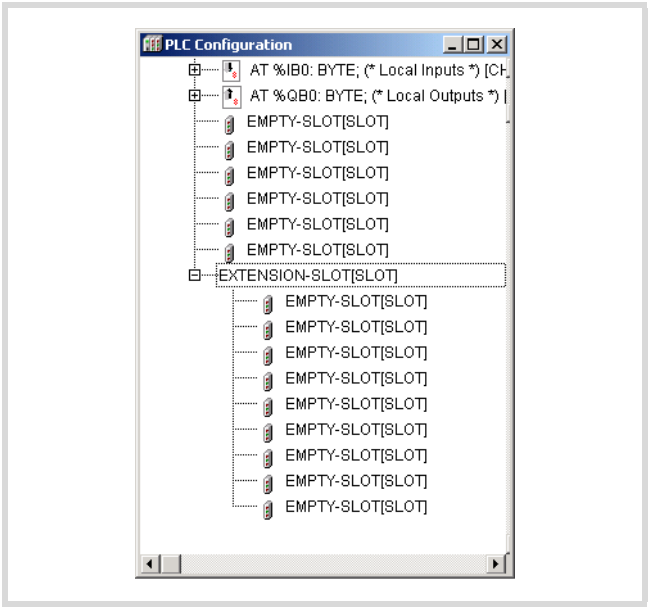


Figure 19: Maximal configuration XC100

Dimensions

Signal modules

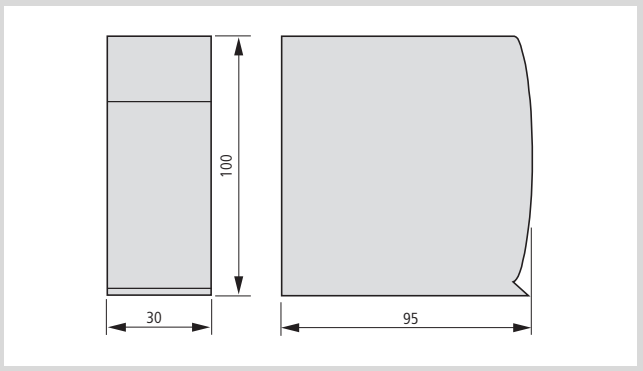


Figure 20: Signal modules

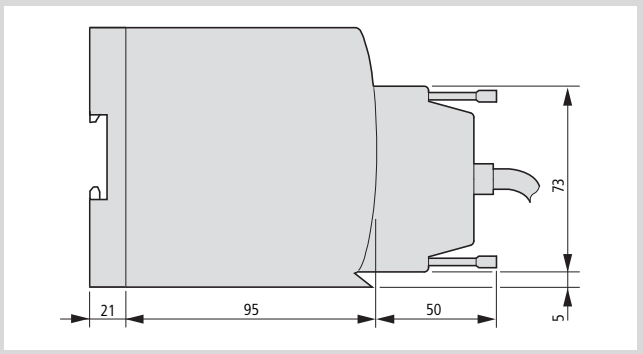


Figure 21: XIOC-32DI, XIOC-32DO with XIOC-TERM32 connector

Backplane

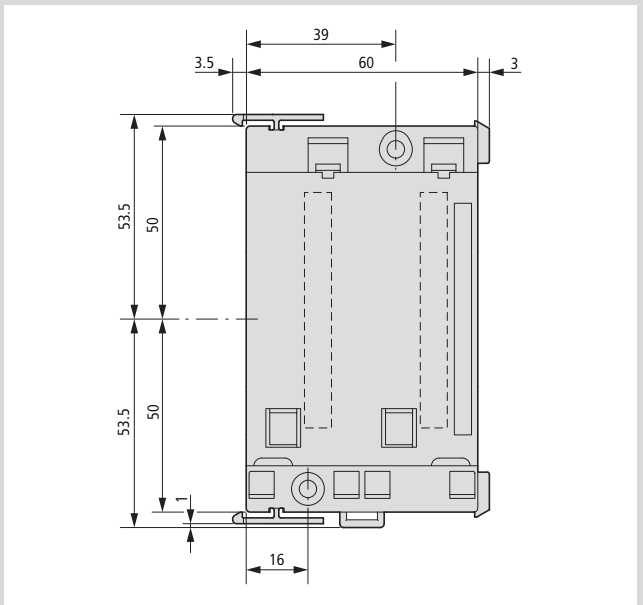


Figure 22: Dimension of the backplanes XIOC-BP-XC, XIOC-BP-2

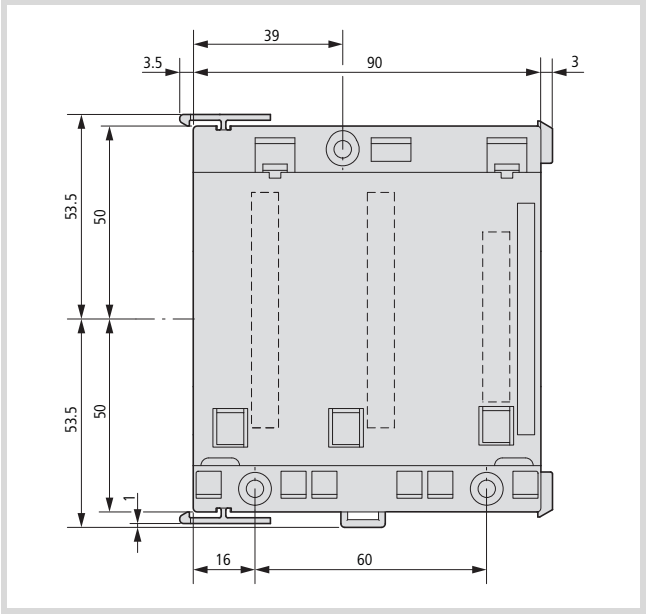


Figure 23: XIOC-BP-XC1, XIOC-BP-3, XIOC-BP-EXT backplane (rack)

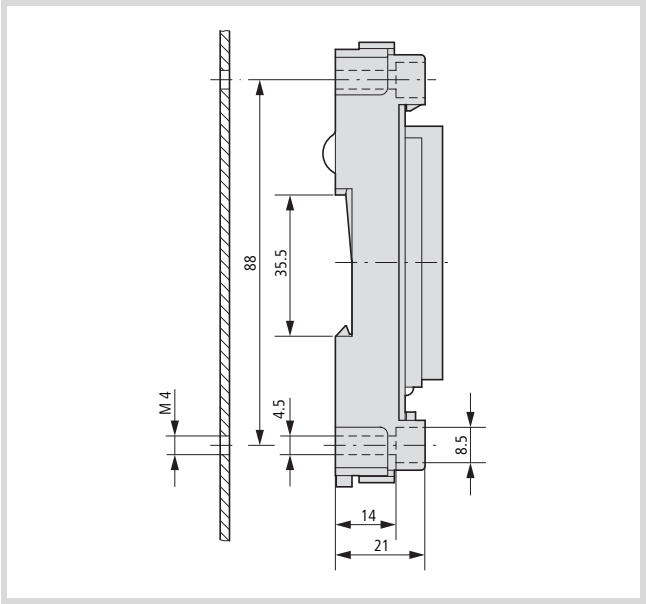


Figure 24: Dimensions of the backplanes

2 Temperature acquisition modules

XIOC-4T-PT

Features

Pt100 (IEC751) and Pt1000 resistance thermometers can be connected to the XIOC-4T-PT temperature acquisition module.

Three temperature ranges are available, that can be selected via DIP switches.

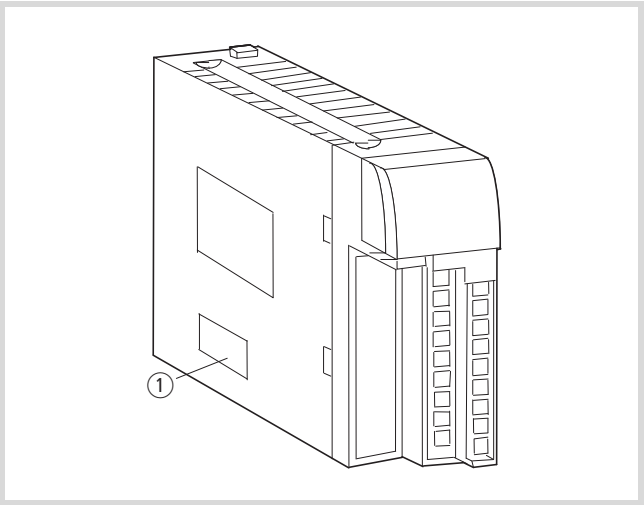
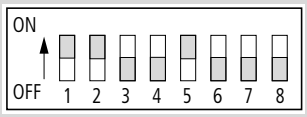
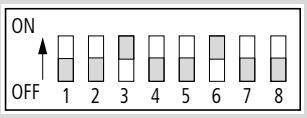
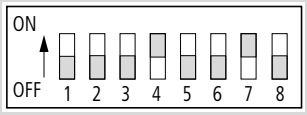


Figure 25: DIP switch position for temperature setting

① DIP switch

Table 3: Setting the temperature range

Type of resistance thermometer	Temperature measurement range (°C)	Accuracy (°C)	DIP switch
Pt100	–20 to + 40	± 0.5	 1, 2, 5 = ON
Pt100	–50 to + 400	± 3	 3, 6 = ON
Pt1000	–50 to + 400	± 6	 4, 7 = ON

Wiring

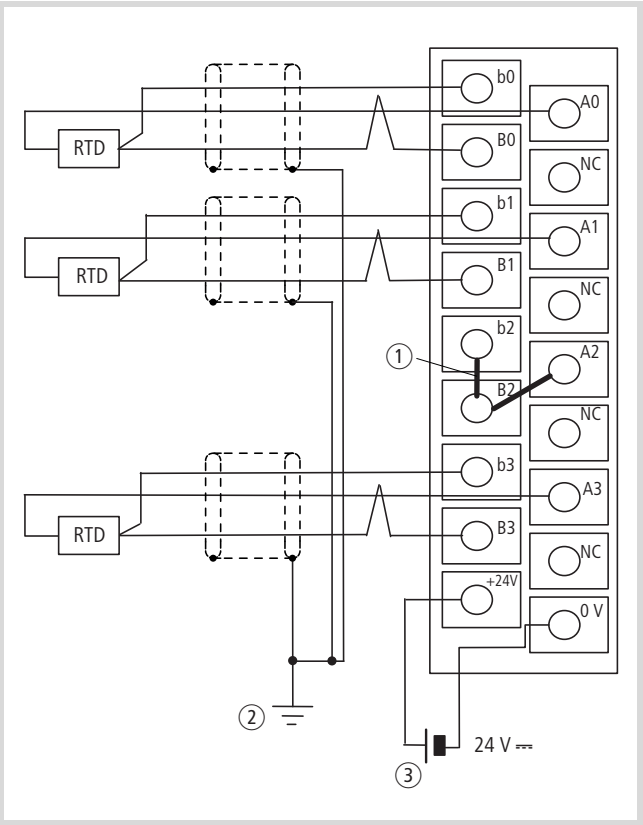


Figure 26: Wiring example

① Join the terminals of unused inputs (b2-B2-A2 in the diagram). Unused inputs have an indefinite status. The value is $7FFF_{hex}$.

② The shielding of the cable can be grounded at one or both ends, depending on the interference situation.

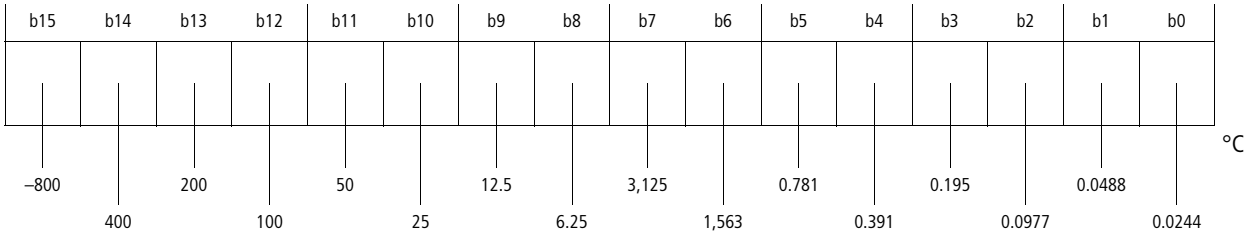
③ External supply voltage, 24 V DC

RTD = Resistance Temperature Detector

NC = Not connected/unused

Data evaluation

1. Range: –50 to +400 °C (Pt100/Pt1000)
The temperature is converted into a signed 15 bit value. The weighting of the bits can be seen in the following diagram.



Example 1

$F800_{hex}$ = 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
F_{hex} 8_{hex} 0_{hex} 0_{hex}

If you enter these bit values in the table above, the result is the following value:

(–800 + 400 + 200 + 100 + 50) °C = –50 °C

Example 2

0600_{hex} = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
0_{hex} 6_{hex} 0_{hex} 0_{hex}

(25 + 12.5) °C = 37.5 °C

If the measured value for the temperature lies outside the range (< –51 °C or > 410 °C), then the data value is displayed as $7FFF_{hex}$.

The relationship between temperature and the measured value is shown by the following equation and the diagram.

$$\text{Temperature (°C)} = \frac{\text{Decimal value, e.g. 256 } (0100_{hex})}{40.96} = 6.26\ (°C)$$

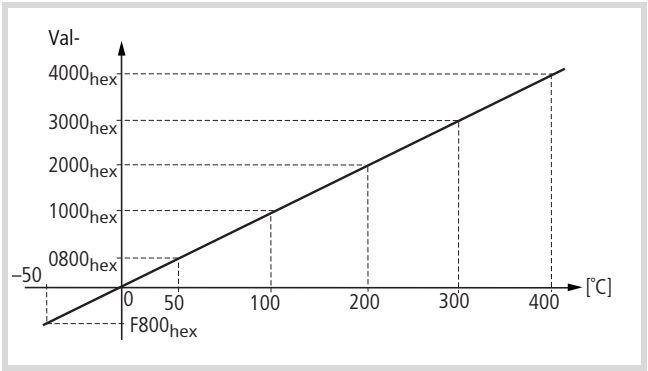
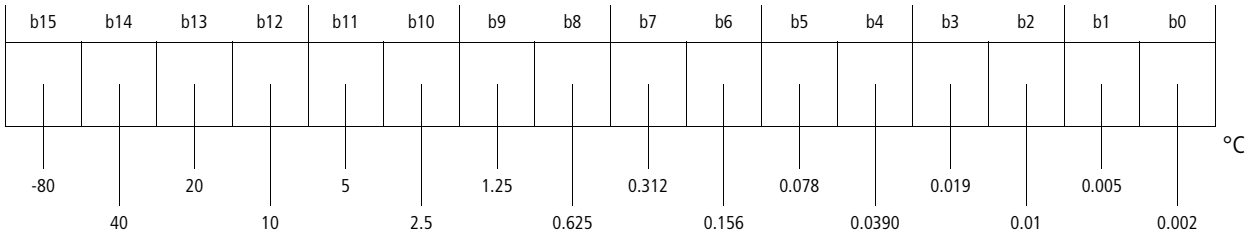


Figure 27: Temperature/measurement diagram

2. Range: –20 to +40 °C (Pt100)

The temperature is converted into a signed 15 bit value. The weighting of the bits can be seen in the following diagram.



Example 1

$E000_{hex}$ = 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

E_{hex} 0_{hex} 0_{hex} 0_{hex}

If you enter these bit values in the table above, the result is the following value:

(–80 + 40 + 20) °C = –20 °C

Example 2

0600_{hex} = 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0

0_{hex} 6_{hex} 0_{hex} 0_{hex}

(2.5 + 1.25) °C = 3.75 °C

If the measured value for the temperature lies outside the range (< –25 °C or > 45 °C), then the data value is displayed as $7FFF_{hex}$.

The relationship between temperature and the measured value is shown by the following equation and the diagram.

$$\text{Temperature (°C)} = \frac{\text{Decimal value, e.g. 256 } (0100_{hex})}{409.6} = 0.626\ (°C)$$

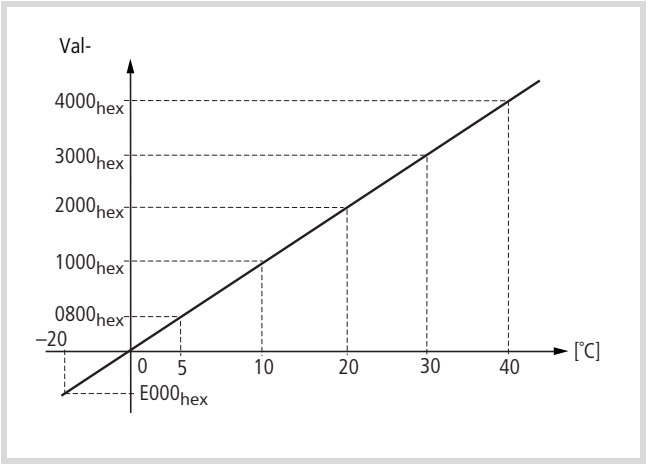


Figure 28: Temperature/measurement diagram

Conversion tables

Table 4: Conversion table for Pt100 (–20 to +40 °C)

Temperature (°C) 1)	Decimal value	Hexadecimal value	Pt100 resistance (Ω)
-25	55296	D800	90.19
-20	57344	E000	92.16
-15	59392	E800	94.12
-10	61440	F000	96.09
-5	63488	F800	98.04
0	0	0000	100.00
5	2048	0800	101.95
10	4096	1000	103.90
15	6144	1800	105.85
20	8192	2000	107.79
25	10240	2800	109.73
30	12288	3000	111.67
35	14336	3800	113.61
40	16384	4000	115.54
45	18432	4800	117.47

1) The technical data refer to the range from –20 to 40 °C.

Table 5: Conversion table for Pt100/Pt1000 (–50 to +400 °C)

Temperature (°C)[1]	Decimal value	Hexadecimal value	Pt100 resistance (Ω)[2]
-60	63078	F666	72.33
-55	63283	F733	78.32
-50	63488	F800	80.31
-45	63693	F8CC	82.29
-40	63898	F999	84.27
-35	64102	FA66	86.25
-30	64307	FB33	88.22
-25	64512	FC00	90.19
-20	64717	FCCC	92.16
-15	64922	FD99	94.12
-10	65126	FE66	96.09
-5	65331	FF33	98.04
0	0	0000	100.00
5	205	00CC	101.95
10	410	0199	103.90
15	614	0266	105.85
20	819	0333	107.79
25	1024	0400	109.73
30	1229	04CC	111.67
35	1434	0599	113.61
40	1638	0666	115.54
45	1843	0733	117.47
50	2048	0800	119.40
55	2253	08CC	121.32
60	2458	0999	123.24
65	2662	0A66	125.16
70	2867	0B33	127.07
75	3072	0C00	128.98
80	3277	0CCC	130.89
85	3482	0D99	132.80
90	3686	0E66	134.70
95	3891	0F33	136.60
100	4096	1000	138.50
110	4506	1199	142.29
120	4915	1333	146.06
130	5325	14CC	149.82
140	5734	1666	153.58
150	6144	1800	157.31
160	6554	1999	161.04
170	6963	1B33	164.76
180	7373	1CCC	168.46
190	7782	1E66	172.16
200	8192	2000	175.84
210	8602	2199	179.51
220	9011	2333	183.17
230	9421	24CC	186.82
240	9830	2666	190.45
250	10240	2800	194.07
260	10650	2999	197.69
270	11059	2B33	201.29
280	11469	2CCC	204.88
290	11878	2E66	208.45
300	12288	3000	212.02
310	12698	3199	215.57
320	13107	3333	219.12
330	13517	34CC	222.65
340	13926	3666	226.17
350	14336	3800	229.67
360	14746	3999	233.17
370	15155	3B33	236.65
380	15565	3CCC	240.13
390	15974	3E66	243.59
400	16384	4000	247.04
410	16794	4199	250.48

1) The technical data refer to the range from –50 to +400 °C for the Pt100.

2) Resistance value Pt1000 = 10 × resistance value Pt100

Fault retrieval

The following list describes some types of fault and advice on removing them.

Faults that affect a single channel

If the measurement is unstable, does not meet the specified accuracy, or indicates the value $7FFF_{hex}$:

- ▶ Check that the wiring is correct for the channel that shows the error.
- ▶ Check whether the cable from the sensor to the module runs close to mains power supply cables.
- ▶ Check that the terminal connection is firmly seated.
- ▶ Check that the data for the Pt100/1000 that is used conform to IEC751.
- ▶ Check the resistance of the external wiring (< 400 Ω).
- ▶ Check that the temperature to be measured lies within the range of the XIOC-4T-PT.

Faults that affect more than one channel

All channels indicate the value $7FFF_{hex}$:

- ▶ check that the external supply voltage is properly connected
- ▶ check whether the load capability of the external supply is adequate (≧ 1 A).

XIOC-4AI-T

Features

The temperature acquisition module XIOC-4AI-T is used for the switching on of thermocouples and for voltage measurement.

For temperature measurement the connection of thermal elements of type B, E, J, K, N, R, S, T is possible. The display is carried out in in 1/10 °C or 1/10 °F. The module recognizes when the temperature falls below or is above the range and also recognizes a wire breakage to the temperature sensor. The module has an integrated cold-junction compensation and interference voltage suppression.

Connection

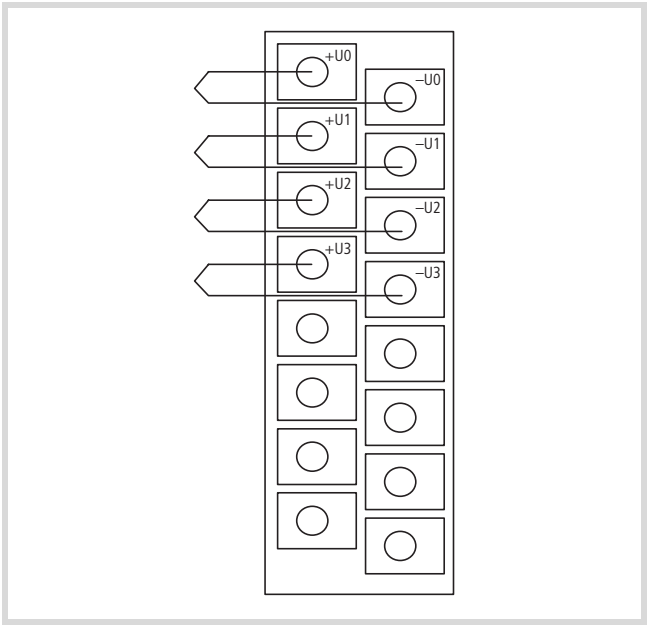


Figure 29: Connection of module

→ Terminals not identified may not be used!

Configuration and Parameterization

The configuration and parameterisation takes place, as usual in the device configuration of the programming system. After selecting the module an integer value is available for every channel that can be used in the user program. A diagnostic word which contains the display of measurement range errors is available for the assessment of diagnostic information.

Defining Measurement Parameters
For each measurement channel the following parameters can be defined:

- Thermal element type
- Scaling
- Channel active/inactive
- Interference voltage suppression 50/60 Hz

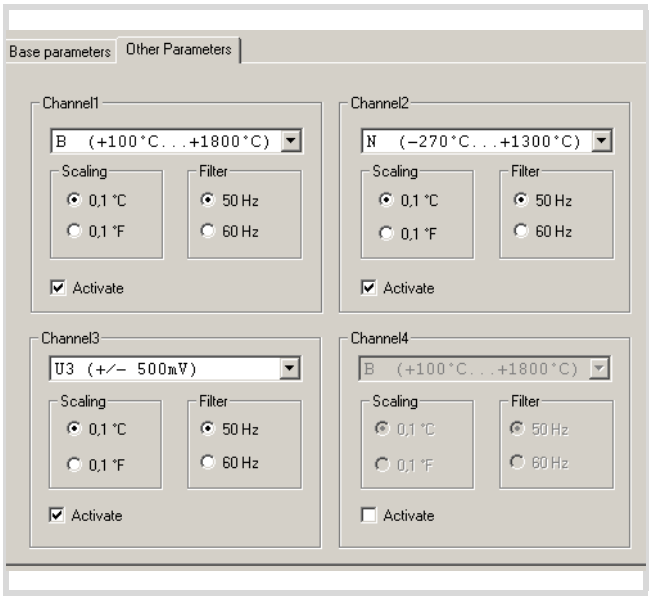


Figure 30: Parameter dialogue

→ In the operation mode "Voltage" the parameter "Scaling" has no relevance.

Measurement range
- Thermocouples

Depending on the thermocouple used various temperature ranges can be measured. The measured value display is carried out as signed integer decimal value in 1/10 Grad C or 1/10 Grad F resolution.

The decimal value 545 corresponds to 54.5 Grad at 1/10 °C setting.

Table 6: Thermocouples with temperature ranges

Element	Temperature range				
B	+100 °C	+212 °F	…	+1800 °C	+ 3272 °F
E	–270 °C	–454 °F	…	+1000 °C	+1832 °F
J	–210 °C	–346 °F	…	+1200 °C	+2192 °F
K	–270 °C	–454 °F	…	+1370 °C	+2498 °F
N	–270 °C	–454 °F	…	+1300 °C	+2372 °F
R	–50 °C	–58 °F	…	+1760 °C	+3200 °F
S	–50 °C	–58 °F	…	+1540 °C	+2804 °F
T	–200 °C	–328 °F	…	+400 °C	+752 °F

- Voltage measurement

When a voltage range (U1 = ±50 mV, U2 = ±100 mV, U3 = ±500 mV), U4 = ±1000 mV) is selected the measurement value corresponds to the signed integer value (16 Bit). The parameterization of the unit °C/°F and the measurement of the cold position remains without relevance in this measurement.

Table 7: Transformation of the voltage measurement (16 Bit signed Integer)

Measurement value [mV] with voltage range …				Transformed value	
±50 mV	±100 mV	±500 mV	±1000 mV	dec.	hex.
–50.00	–100.00	–500.00	–1000.00	–32768	0x8000
–49.998	–99.997	–499.985	–999.969	–32767	0x8001
–0.002	–0.003	–0.015	–0.031	–1	0xFFFF
0.00	0.00	0.00	0.00	0	0x0000
0.002	0.003	0.015	0.031	1	0x0001
49.998	99.997	499.985	999.969	32766	0x7FFE
50.00	100.00	500.00	1000.00	32767	0x7FFF

Table 8: Resolution for voltage measurement

Resolution[µV] with voltage range…			
±50 mV	±100 mV	±500 mV	±1000 mV
1.526 µV	3.052 µV	15.259 µV	30.519 µV

Diagnostics

The status word contains the diagnosis information for all four channels.

For every channel exceeding and shortfall of the measurement value is displayed as well as a wire breakage. With an error the corresponding ERROR-LED on the module is also lit.

Bit 15	Bit 14	Bit13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit3	Bit 2	bit1	Bit 0
Channel 3				Channel 2				Channel 1				Channel 0			
D33	D32	D31	D30	D23	D22	D21	D20	D13	D12	D11	D10	D03	D02	D01	D00

Table 9: Allocation of diagnostic information

Dx0	Range shortfall: Measurement value < Measurement start value – (1 % ±0.5 %) × Measurement range The following applies for elements with a temperature range from –270 °C: Measurement value < Measurement start value
Dx1	Out-of-range value (Measurement value > Measurement range end value + (1% ± 0.5%) × Measurement range)
Dx2	Wire breakage (only with temperature measurement)
Dx3	Reserved

x = Channel 0 … 3

3 Counter modules XIOC-…CNT-100kHz

Assembly

The counter module XIOC-1CNT-100kHz provides one channel, the module XIOC-2CNT-100kHz provides two channels, each with one input for pulse frequencies up to 100 kHz, a reference input and two digital outputs.

You can connect single-phase or two-phase incremental encoders (with/without quadruple evaluation for the two-phase).
The type of counter (linear or ring counter) is set with the aid of DIP switches.

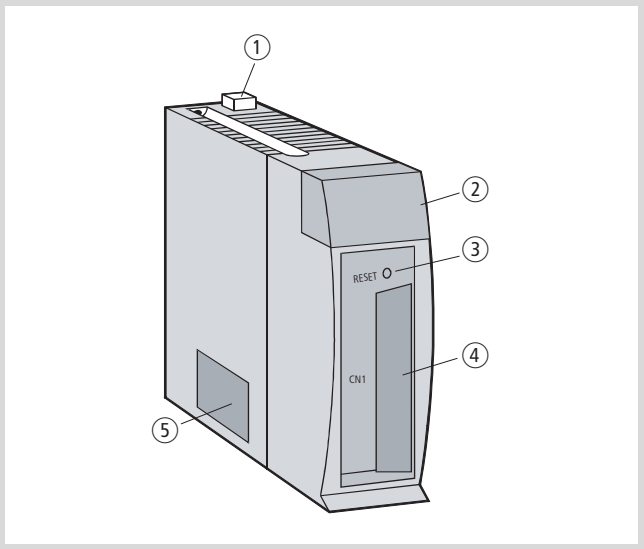


Figure 31: Assembly of the counter module

No.	Designation	Comments
①	Interlock	
②	LED display	→ page 33
③	RESET button	Sets the parameters to "0". → page 33
④	Connection for pulse generator	30 pole connection (15 pins × 2) for the XIOC-TERM30-CNT4 connector → page 36, 37
⑤	Mode switch (DIP)	This switch is used to set the operating mode → page 34

RESET button on the module

You operate the RESET button (by using a pointed object) to reset the parameters to their initial (default) setting. When the button is pushed, the ERROR-LED in the LED display lights up red.

LED display

The LEDs have the following designations:

1 A	1C	1M	PW
2B	2B	2M	ER
0	1	2	3
XIOC-2CNT-100KHZ			

1 A	1C	1M	PW
			ER
0	1		
XIOC-1CNT-100KHZ			

LED	Meaning	
1A, 1B	Encoder signal, phase A, B; channel 1	
2A, 2B	Encoder signal, phase A, B; channel 2	
1M, 2M	Encoder reference signal (marker signal); channel 1, 2 The LED lights up when a voltage is present at the input, regardless of whether the signals are inverted or not.	
PW	Indicates the power supply for the module:	
	on:	OK
	blinking:	• After incorrect parameter entry • With the counter type "Ring counter", the LED blinks if voltage has been applied to the PLC. After you have set the setpoint value (WRITEPRESETVALUE) and the comparison value (WRITESETTINGVALUE2), the LED lights up continuously.
	OFF	Hardware error
ER	Error	
	on:	• After operating the RESET button on the module • Hardware error
0, 1, 2, 3	Outputs Y	

Programming

Programming was implemented using the following function blocks:

- CounterControl,
- ReadCounter,
- WriteCounter,
- CounterControl,
- XIOC_IncEncoder.

A detailed description can be found in the "Function blocks for easySoft-CoDeSys" manual.
This manual is available as a PDF file and can be downloaded at: http://www.eaton.com/moeller → **Support.**
Use "05010002" as a search keyword to find it as quickly as possible.

The function blocks are contained in the "Counter.lib" (XC100) and "XC200_Counter.lib" library files.

Mode/operating mode switch

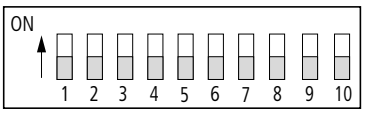


Figure 32: Mode/operating mode switch, state of delivery

→ In order to set the DIP-switches you will first have to take out the module. But switch off the supply voltage first!

	Switch	Position	Function	Channel
Type of counter input				
Mode 1	1	OFF	2-phase counter, max. 100 kHz	1 + 2
	2	OFF		
Mode 2	1	ON	1-phase counter, (pulse-change)	1 + 2
	2	OFF		
Mode 3	1	OFF	1-phase counter, (polarity reversal)	1 + 2
	2	ON		
Mode 4	1	ON	2-phase counter with 4x evaluation, max. 25 kHz	1 + 2
	2	ON		
Polarity of the reference input (marker input)				
	3/4	OFF	A voltage on the input produces a "0" signal	1/2
		ON	A voltage on the input produces a "1" signal	
CPU stop → Counter				
	5/6	OFF	CPU STOP → Counter STOP	1/2
		ON	CPU-STOP → Counter RUN	
Linear/ring counter				
	7/8	OFF	Linear counter	1/2
		ON	Ring counter	
	9/10	OFF	not used	–

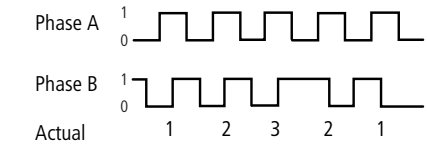


Figure 33: Mode 1 (2-phase)

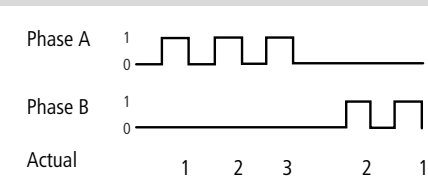


Figure 34: Mode 2 (1-phase)

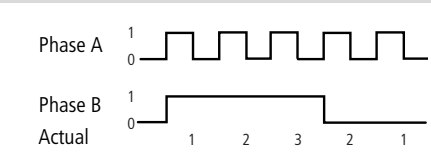


Figure 35: Mode 3 (1-phase)

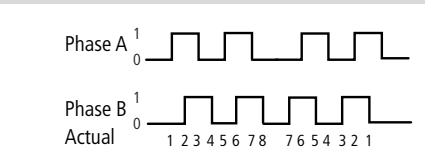


Figure 36: Mode 4 (2-phase, with quadruple evaluation)

Connecting an incremental encoder to the counter input

The counter module has an input circuit that permits the connection of various types of incremental encoder. An encoder with a differential output (+/– 5 V DC) or an open collector output (12 to 24 V DC) can be connected. The following examples illustrate the various connection options.

Two incremental encoders

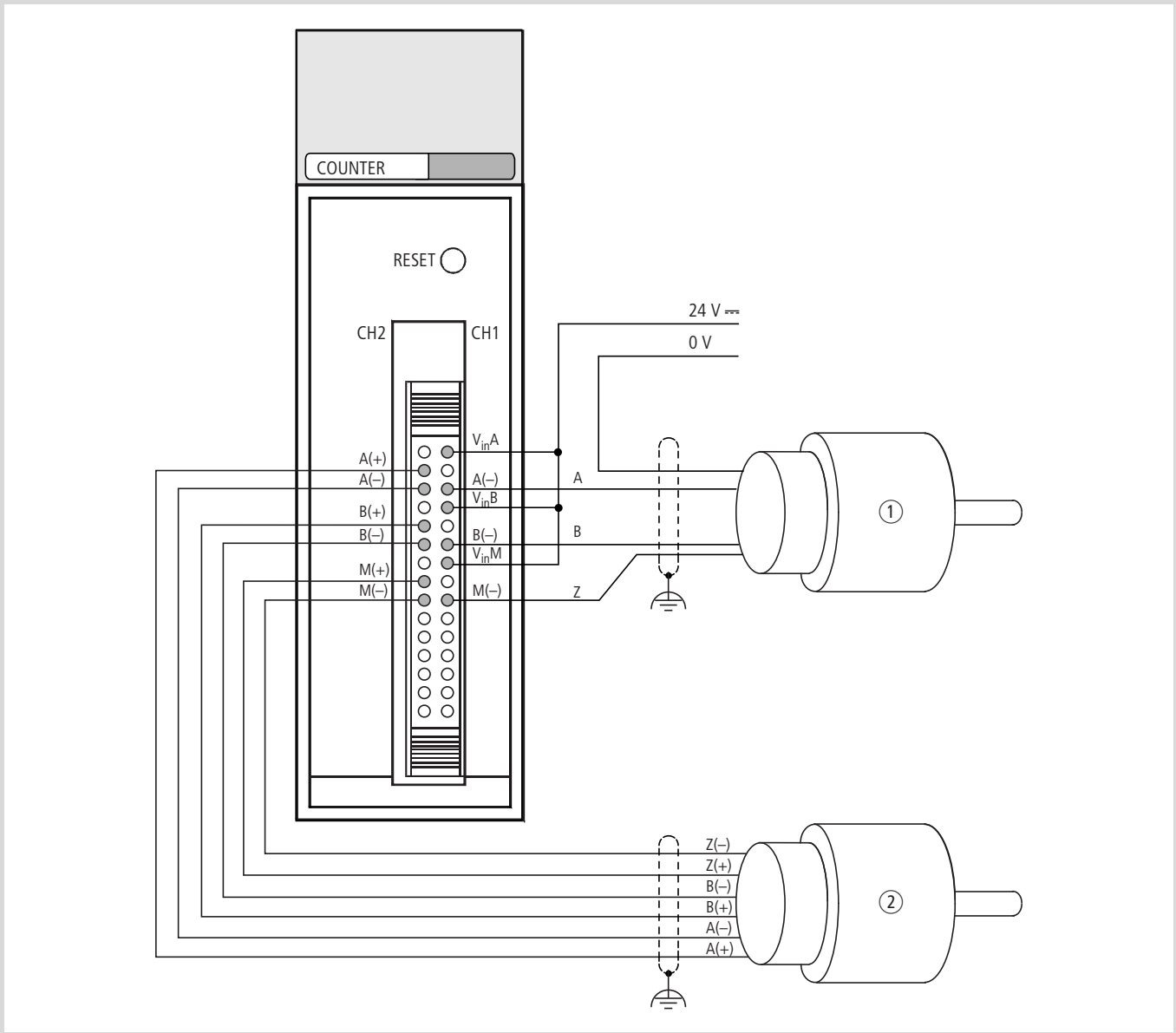
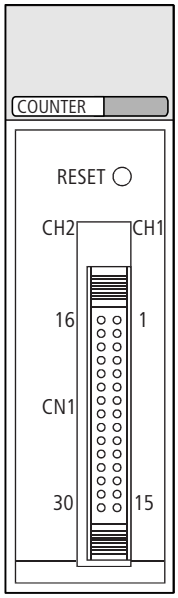


Figure 37: Connection for 2 incremental encoders (example)

① Encoder with **12 to 24 V DC** open collector outputs

② Encoder with **+/– 5 V DC** differential outputs

Terminal arrangement	No.	CH2	No.	CH1		Meaning of the signals
		XIOC-2CNT		**XIOC-2CNT/ XIOC-1CNT**		
	16	V_{IN} A	1	V_{IN} A	Phase A	If voltage input is used, connect to 12 to 24 V DC supply voltage.
	17	A (+)	2	A (+)		If the differential input is used: connect to the positive polarity.
	18	A (–)	3	A (–)		If the voltage input is used, connect to the open-collector signal. If the differential input is used, connect to the negative polarity.
	19	V_{IN} B	4	V_{IN} B	Phase B	If voltage input is used, connect to 12 to 24 V DC supply voltage.
	20	B (+)	5	B (+)		If the differential input is used: connect to the positive polarity.
	21	B (–)	6	B (–)		If the voltage input is used, connect to the open-collector signal. If the differential input is used, connect to the negative polarity.
	22	V_{IN} M	7	V_{IN} M	Marker (reference)	If voltage input is used, connect to 12 to 24 V DC supply voltage.
	23	M (+)	8	M (+)		If the differential input is used: connect to the positive polarity.
	24	M (–)	9	M (–)		If the voltage input is used, connect to the open-collector signal. If the differential input is used, connect to the negative polarity.
	25 to 27	not used	10 to 12	not used		Do not connect anything to these terminals.
	28	Y2	13	Y0	Output	Comparator output
	29	Y3	14	Y1		
	30	Com2	15	Com1		(–) reference potential for the Y outputs. The following applies for XIOC-2CNT: reference potential 1 and 2 are independent of each other.

Note: The pin numbers defined for the XIOC-1CNT-100 kHz and XIOC-2CNT-100 kHz do not match those given by the connector manufacturer.

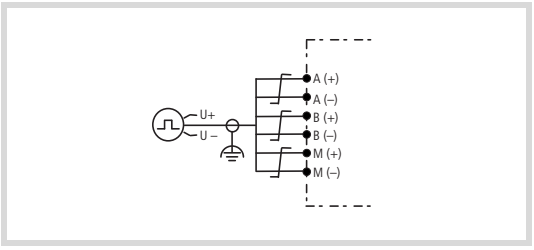


Figure 38: Encoder with differential outputs

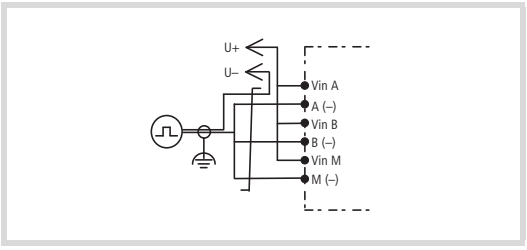


Figure 39: Encoder with voltage outputs

Cable with attached connector for the counter module

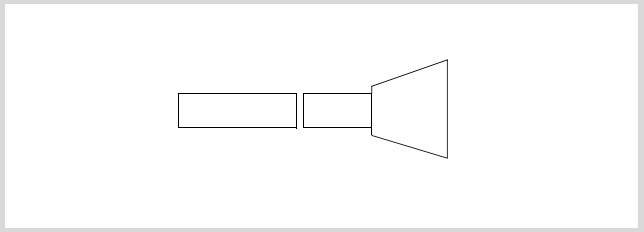


Figure 40: Cable with connector (XIOC-TERM30-CNT4)

No.	Channel 2	Colour	No.	Channel 1	Colour	Meaning of the signals	
16	VIN A	red/white	1	VIN A	black		12 to 24 V DC (open-collector)
17	A (+)	orange/black	2	A (+)	brown		(+) differential output
18	A (–)	green/white	3	A (–)	red		(–) differential-output (open-collector)
19	VIN B	blue/white	4	VIN B	orange	Phase B	12 to 24 V DC (open-collector)
20	B (+)	yellow/black	5	B (+)	yellow		(+) differential output
21	B (–)	violet/white	6	B (–)	green		(–) differential-output (open-collector)
22	VIN M	grey/black	7	VIN M	blue	reference (marker)	12 to 24 V DC (open-collector)
23	M (+)	pink/black	8	M (+)	violet		(+) differential output
24	M (–)	blue/black	9	M (–)	grey		(–) differential-output (open-collector)
25	–	green/black	10	–	white		–
26	–	pink/red	11	–	pink		
27	–	pink/blue	12	–	blue		
28	Y2	pink/green	13	Y2	light green	Output	open-collector
29	Y3	red/black	14	Y3	black/white		open-collector
30	Com2	orange/white	15	Com2	brown/white		0 V (open-collector)

Incremental encoder with differential output

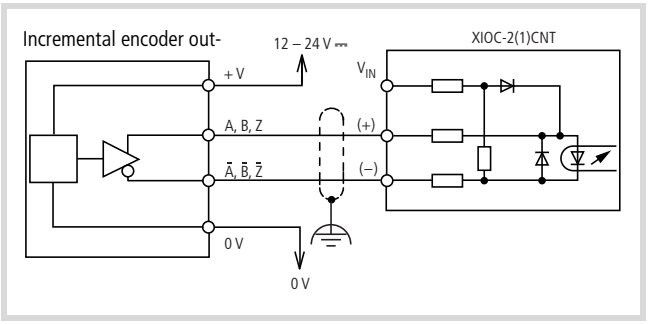


Figure 41: Connection for an incremental encoder with a differential output (example)

Incremental encoder with NPN transistor output

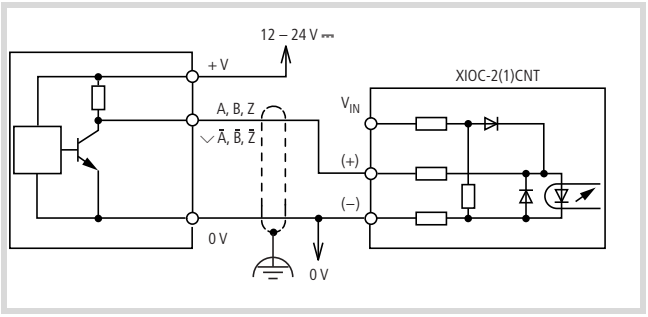


Figure 42: Connection for an incremental encoder with an NPN transistor output (example)

Incremental encoder with NPN transistor output (open-collector)

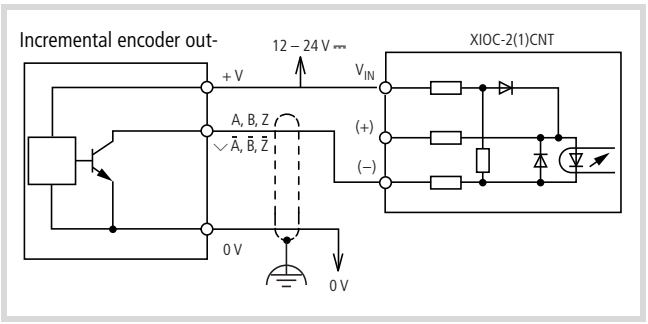


Figure 43: Connection for an incremental encoder with an open-collector NPN transistor output (example)

Incremental encoder with PNP transistor output (open-collector)

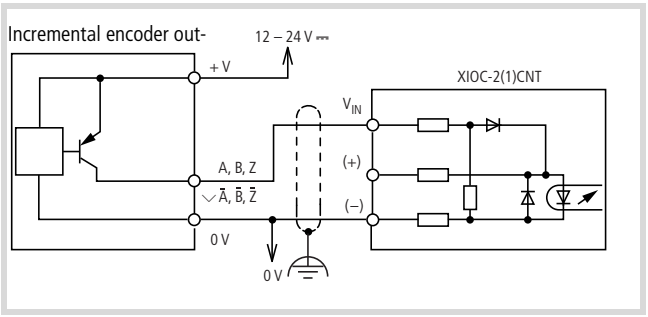


Figure 44: Connection for an incremental encoder with an open-collector PNP transistor output (example)

Connecting devices to the Y outputs

The counter module has 2 open-collector transistor outputs per channel. The diagram shows how to connect it to another device.

Caution!
Wire in an 0.5 A fuse, as shown in the diagram, to protect the internal circuitry (see figure).

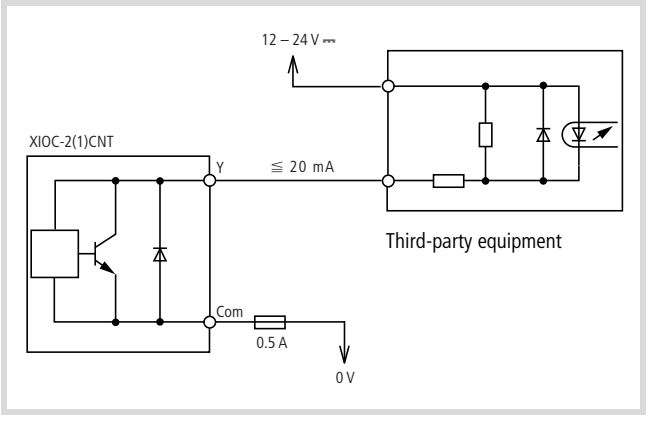


Figure 45: Connecting third-party equipment to the counter module

Function summary

A counter channel has the function of either a linear counter or a ring counter, depending on the setting of the operating mode switch on the module.

Linear counter

The counting range of the linear counter starts at the value 0 and ends at the value 4294967295 ($FFFFFFFF_{hex}$). If the counter is enabled, it starts at 0 and counts all incoming pulses up or down – depending on the count direction. If the count reaches the end value it starts again at 0.

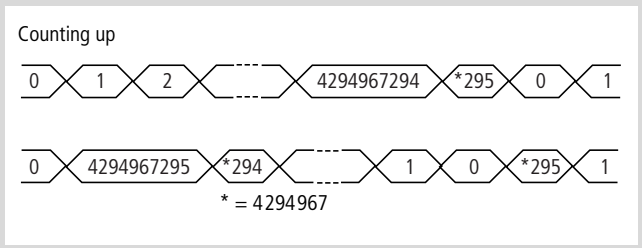


Figure 46: Counting range of the linear counter

Parameterizing the comparison value, setting module outputs

You can set a comparison value, so that an action can be performed when a defined count value has been reached. It is continuously compared with the actual value. If they are identical, two types of output can be activated. The outputs are led out directly from the module, for a fast response.

The "Latch" output (=), Equal flag:
The "Latch" output is set when equality is achieved. It is indicated by the "=" symbol. The Equal flag serves as the internal marker for the "Latch" output. The output and flag remain set until you reset them.

The "Level" output (>):
The "Level" output is set to "1" if the actual value is larger than the comparison value. If the actual value falls below the comparison value, then it is reset to "0". The "Level" output is indicated by the ">" symbol.

You can set the comparison value at the "CounternEnable" input, either at the start or during operation. This does not depend on the counter being enabled.

Example:

- Count direction: up
- Comparison value: 4294967200

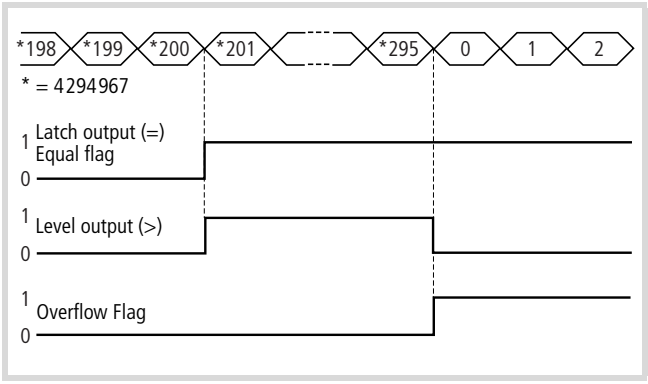


Figure 47: Setting module outputs

Overflow flag

The Overflow flag is set when the actual value changes from $FFFFFFFF_{hex}$ to 0. You can reset it by using the CLEAROVERFLOW command.

Change actual value

You can change the actual value during counting. This does not depend on the counter being enabled.

Use of the reference input

Incremental encoders send a reference marker signal once per turn. This can be used to overwrite the actual value by a setpoint value that was defined as part of the parameter settings. In order to be able to process the reference signal, the reference input must be enabled.

Example of a linear counter, with the functions:
- interrogate comparison value and reference signal
- reset outputs

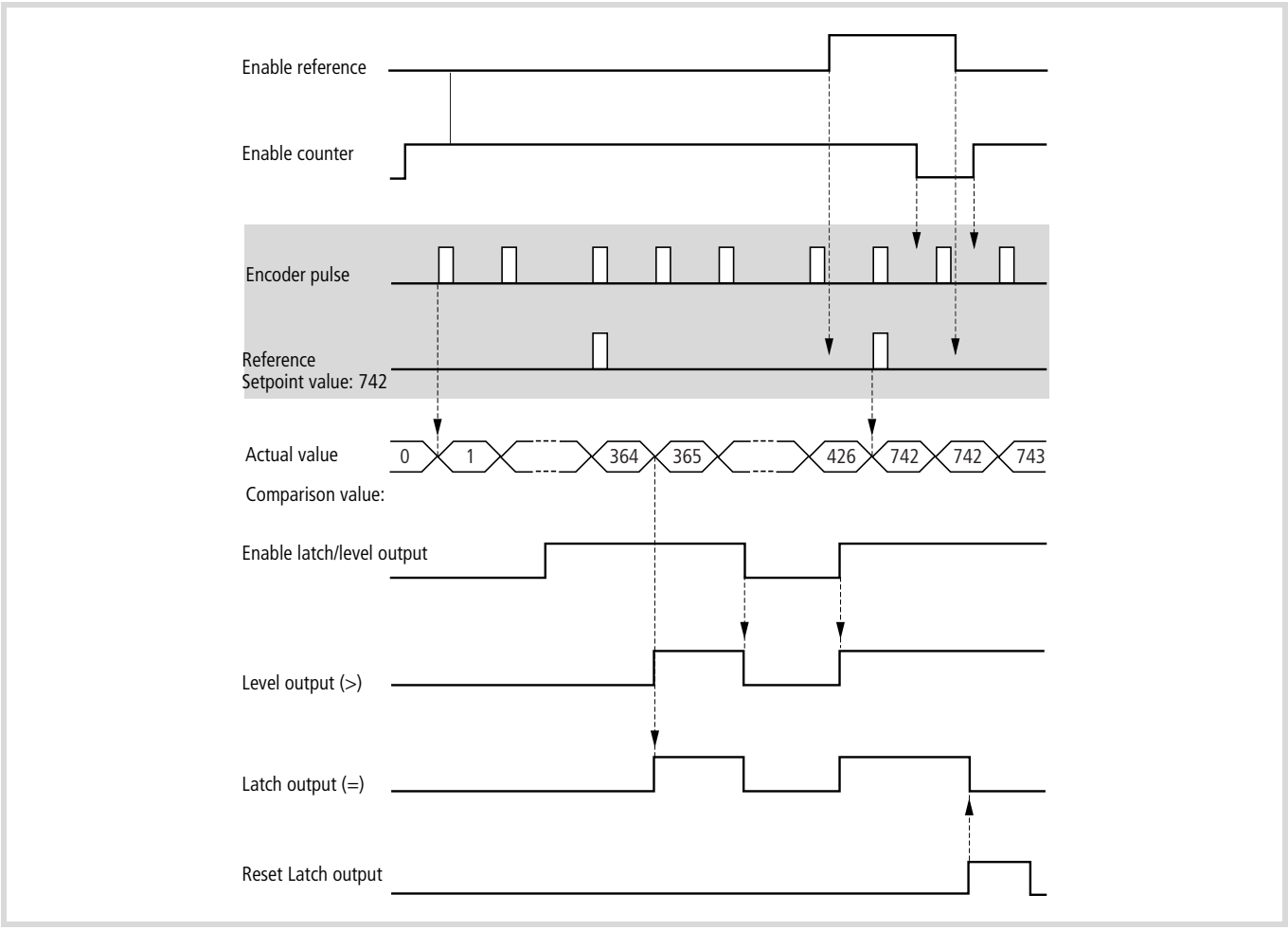


Figure 48: Example of a linear counter, with the functions "interrogate comparison value and reference signal" and "reset outputs"

Ring counter

The counting range is defined by the start and end values, whereby the start value must be lower than the end value.

As soon as the counter has been enabled, the start value is set and all incoming pulses will be counted. The following actual values will be shown, depending on the count direction (up or down).

Example:

- Start value = 10
- End value = 248

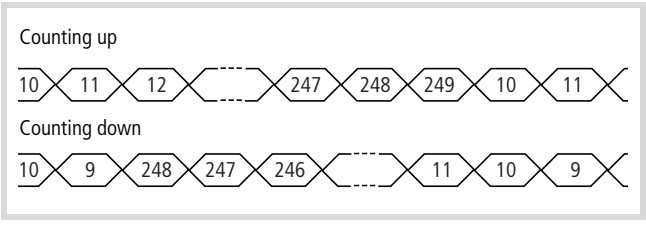


Figure 49: Counting range of the ring counter

An up counter counts up to the end value + 1, and then restarts from the start value. For a down counter, the next value is the start value – 1, carrying on to the end value.

As a rule: minimum start value = 0;
maximum end value = $FFFFFFFF_{hex}$.

Parameterizing the comparison value, setting module outputs

You can set a comparison value, so that an action can be performed when a defined count value has been reached.
The comparison value must lie between the parameter settings for start value and end value. It is continuously compared with the actual value. When equality is achieved, a "Latch" output (=) can be set.
This output is led out directly from the module, for a fast response. The Equal flag serves as the internal marker for the "Latch" output. The output and flag remain set until you reset them.

You can set the comparison value either at the start or during operation. This does not depend on the counter being enabled at the "CounternEnable" input.

Example:

- Count direction: up
- Parameters: start value: 0, end value: 294, comparison value: 200

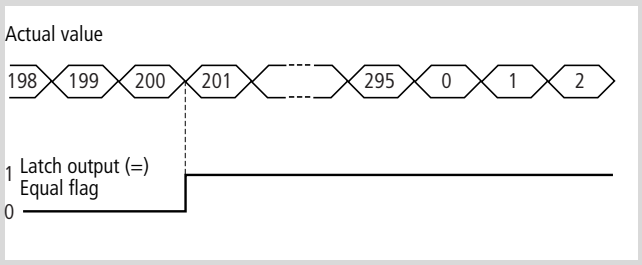


Figure 50: Set module output (Latch)

Change actual value
You can change the actual value during counting. This does not depend on the counter being enabled.

Requirement: start value ≦ actual value ≦ end value.

Example of a ring counter, with the functions:
- interrogate comparison value and reference signal
- reset outputs
- Set actual value

→ figure 51

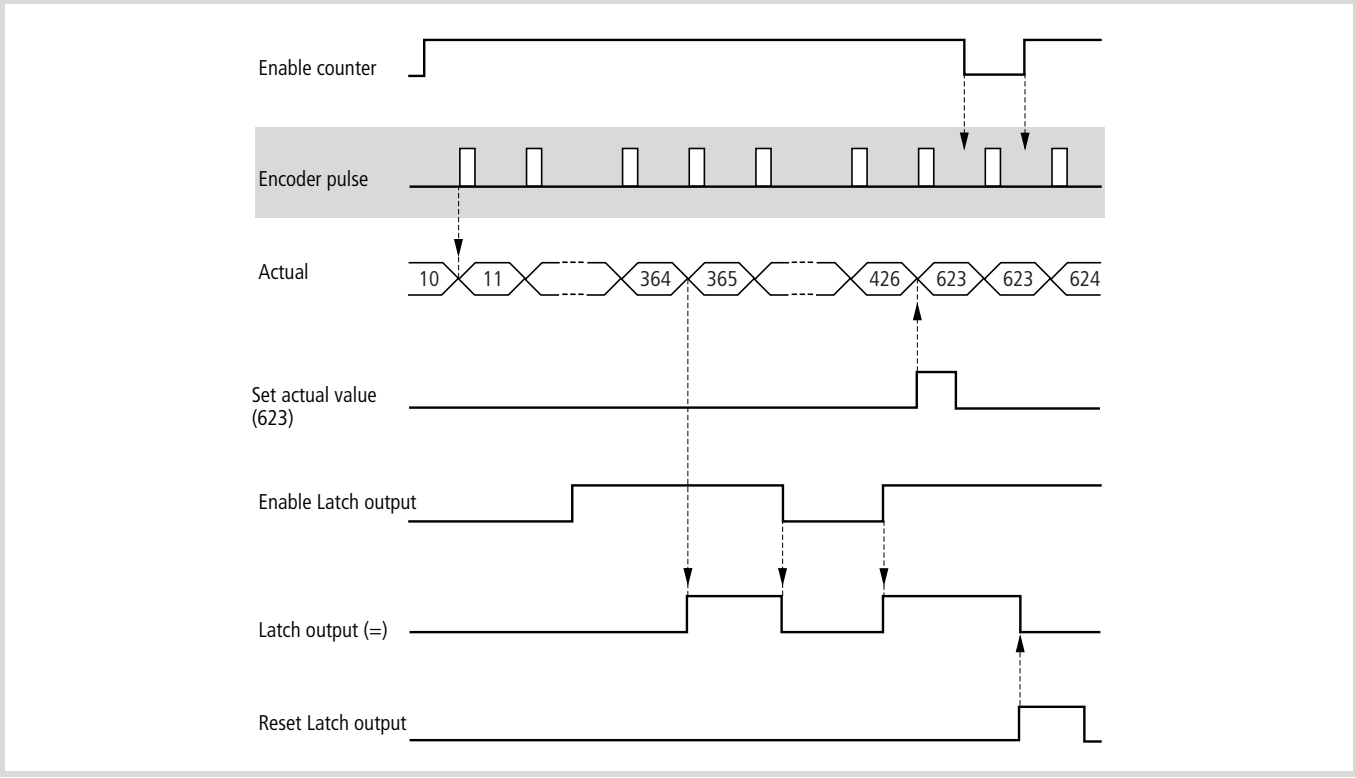


Figure 51: Example of a linear counter, with the functions "interrogate comparison value and reference signal" and "reset outputs"

Additional functions for linear and ring counters

Regardless of the type of counter input (mode 1 to 4), you can set the counter type (linear or ring counter) for each channel on the operating mode switch of the module → page 34. You can also assign other functions to the counter type, making the settings via the switch:

Counter RUN/STOP when CPU has STOP state
Counter RUN: If the CPU is in the STOP state, the encoder pulses continue to be counted.

Counter STOP: If the CPU is in the STOP state, no pulses are counted

Polarity of the reference input
This function is only activated with a linear counter.

- Switch OFF: voltage at the input produces a "0" signal.
- Switch ON: voltage at the input produces a "1" signal.

Configure counter features

Table 10: Configuration options

	Feature	Linear counter	Ring counter
Interrogation option for the counter	Start value	0	any
	End value	$FFFFFFFF_{hex}$	any
	Overflow Flag	"1" if actual value changes from FFFFFFFF → 0	0
	Underflow Flag	"1" if actual value changes from 0→ FFFFFFFF	0
	Clear Overflow flag	Set Overflow flag "0"	–
	Clear Underflow flag	Set Underflow flag "0"	–
	Enable counter	TRUE at input CounternEnable	
	Inhibit counter	FALSE at input CounternEnable	
Comparison value	Output (=)/ Equal flag	TRUE if actual value = comparison value → figure 52	
	Output (>)	TRUE if actual value > comparison value → figure 53	–
	Output (=)/ clear Equal flag	Set Output (=) and Equal flag "0"	
	Output (=) enable/inhibit	Input CompareOutputnEnable	
Reference input	Reference input = 1	Setpoint value over-writes actual value → figure 53	–
	Reference input: enable/inhibit	Input "ReferenceMark-ernEnable"	–
	Invert reference input signal	By DIP-switch	–

The diagram, shows the state of the Latch output (=) for linear and ring counters, depending on the count sequence:

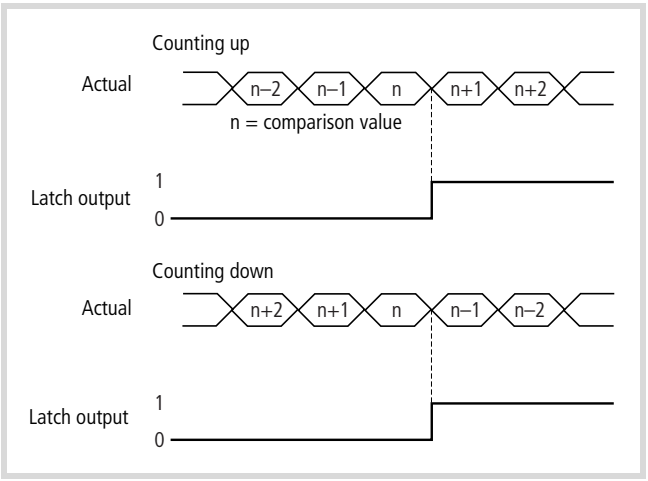


Figure 52: Interrogate comparison value

The diagram shows for the linear counter

- the state of the Level output (>), depending on the count sequence
- the acceptance of the setpoint value P, in response to the reference signal.

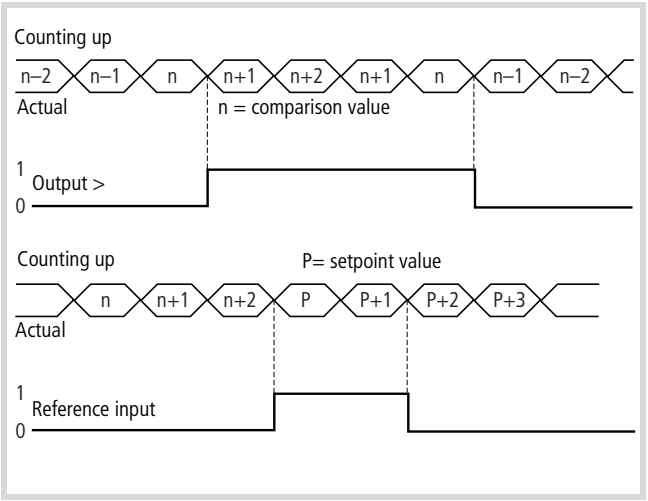


Figure 53: Interrogation of comparison and reference signals

Processing of commands

The following table describes the commands and illustrates the sequence which they are processed after the controller is switched on. You should also keep to this sequence during programming. Some of the commands may not be necessary, depending on the application. Where commands only apply to the linear counter of the ring counter, this is also mentioned. The counting range for the linear counter lies between the start value 0 and the end value "$FFFFFFFF_{hex}$".

→ The input values to the function blocks "CounterControl", "WriteCounter" and "CounterFlags" are accepted when a positive edge appears at the "Strobe" input.

Set start value

Only for ring counter:

▶ Enter the command WRITEPRESETVALUE at the "Command" input of the block "WriteCounter" and the start value at the "Data" input.

Take care that the condition "Start value < End value" is fulfilled.

Set end value

Only for ring counter:

▶ Enter the command WRITESETTINGVALUE1 at the "Command" input of the block "WriteCounter" and the end value at the "Data" input.

Set comparison value

▶ Enter the command WRITESETTINGVALUE1 (for linear counter) or WRITESETTINGVALUE2 (for ring counter) at the "Command" input of the block "WriteCounter" and the comparison value at the "Data" input.

You can access the channels individually or together.

You can set the comparison value either at the start or during operation. This does not depend on the counter being enabled at the "CounternEnable" input of the function block "CounterControl".

When the actual value matches the comparison value, the module outputs will be set. The Equal flag associated with the output is also set at the same time. You can interrogate the flag by using the command READFLAGS for the "CounterFlags" block.

The Equal flag retains its state if the state of the CPU changes from RUN → STOP or STOP → RUN.

Assign module outputs to the comparison value 1 or 2

Comparison value 1 (linear counter) or comparison value 2 (ring counter) can be assigned to several module outputs (Yn, n = 1, 2, 3, 4) and the conditions "=" and/or ">" for setting the outputs (only the "=" condition can be used with a ring counter).

▶ To achieve this, set up a bit combination (16 bits), e.g. 0021_{hex}, that is applied to the "OutputSpecification" input of the "CounterFlags" block (further information can be found in the description of the function block "CounterFlags" in the manual "Function blocks for easySoft-CoDeSys", MN05010002Z-EN; previously AWB2786-1456GB.
▶ Apply the SPECIFYOUTPUT command to the "Command" input and a "1" signal to the "Strobe" input.

The "CounterEnable" input (flag) must not be set. When the condition "Actual value = preset value" is met, the (Latch) output Y0 is set to "1" by the bit combination "0021". It will remain set until you reset it by using the "ClearEqualn" input of the "CounterControl" block.

Only for linear counters:
The (Level) output Y1 will be set to "1" if the condition "Actual value > Setpoint value" is fulfilled.
If the actual value falls below the comparison value 2, then the output is automatically reset to "0".

Enable module output

The module outputs are the "Latch" output (=) and the "Level" output (>).
The Level output is only available for the linear counter.

▶ To enable the outputs, apply a "1"signal to the "CompareOutputnEnable" of the "CounterControl" block.

An inhibit applied to the output does not affect the Equal flag.

Set setpoint value

Only for linear counters:

The command is carried out if there is a "0" signal applied to the "CounternEnable" input of the "CounterControl" function block.

▶ Enter the command WRITEPRESETVALUE at the "Command" input of the block "WriteCounter" and the setpoint value at the "Data" input.

If the encoder transmits a reference signal, the setpoint value overwrites the actual value.

Enable reference input

Only for linear counters:

▶ Apply a "1" signal to the "ReferenceMarkernEnable" (n = 1, 2) input of the "CounterControl" function block, so that the reference signal can be received from the encoder.

Enable counter input

▶ Apply a "1" signal to the "CounternEnable" input of the "CounterControl" function block, so that the signals can be received from the encoder.

When using a ring counter, the enable can only be implemented after you have set the start and end values.

Set new actual value

▶ Enter the command WRITECURRENTVALUE at the "Command" input of the "WriteCounter" block, and the actual value at the "Data" input.

Reset Latch output and Equal flag (EQ)

▶ Apply a "1" signal to the "ClearEqualn" input of the "CounterControl" function block to set the output and the Equal flag to "0".

The output and flag can only be set again if you apply a "0" signal to this input.

Read out start value

Only for ring counter:

▶ Enter the command READPRESETVALUE at the "Command" input of the "ReadCounter" block.

As soon as you have entered this command, the values will be shown at the outputs: "DataLowChanneln" and "DataHighChanneln", as well as "Outputn_UDINT" and "Outputn_DINT".

The command applies to both channels.

Read out end value

Only for ring counter:

▶ Enter the command READSETTINGVALUE1 at the "Command" input of the "ReadCounter" block.

As soon as you have entered this command, the values will be shown at the outputs: "DataLowChanneln" and "DataHighChanneln", as well as "Outputn_UDINT" and "Outputn_DINT".

The command applies to both channels.

Read out comparison value

▶ Enter the command READSETTINGVALUEn at the "Command" input of the "ReadCounter" block.

As soon as you have entered this command, the values will be shown at the outputs: "DataLowChanneln" and "DataHighChanneln", as well as " Outputn_UDINT" and "Outputn_DINT".

The command applies to both channels.

Read out setpoint value

Only for linear counters:

▶ Enter the command READPRESETVALUE at the "Command" input of the "ReadCounter" block.

As soon as you have entered this command, the values will be shown at the outputs: "DataLowChanneln" and "DataHighChanneln", as well as " Outputn_UDINT" and "Outputn_DINT".

The command applies to both channels.

Read actual (= current) values

▶ Enter the command READCURRENVALUE at the "Command" input of the "ReadCounter" block.

As soon as you have entered this command, the actual value will be shown continuously at the outputs: "DataLowChanneln" and "DataHighChanneln", as well as "Outputn_UDINT" and "Outputn_DINT".

The command applies to both channels.

Read out flags This command is described in detail on Page 46!

Clear Overflow flag Only for linear counters:

▶ Enter the command CLEAROVERFLOW at the "Command" input of the "CounterFlags" function block to clear the flag.

The flag is set when the actual value changes from $FFFFFFFF_{hex}$ to 00000000_{hex}.

You can interrogate the flag state by using the command READFLAGS for the "CounterFlags" block. 16 bits are shown at the "StatusChanneln" output of the "CounterControl" block.

Bit 9 (OF) indicates the state of the Overflow flag.

Clear Underflow flag Only for linear counters:

▶ Enter the command CLEARUNDERFLOW at the "Command" input of the "CounterFlags" function block to clear the flag.

The flag is set when the actual value changes from 00000000_{hex} to $FFFFFFFF_{hex}$.

You can interrogate the flag state by using the command READFLAGS for the "CounterFlags" block. 16 bits are shown at the "StatusChanneln" output of the "CounterControl" block.

Bit 8 (UF) indicates the state of the Underflow flag.

Read out flags
Apply the command READFLAGS to the "Command" input of the "CounterFlags" block, in order to update the function block outputs: "Outputs", "StatusChanneln", "OutputsChanneln".
A positive edge must be applied to the "Strobe" input in order to execute the command.

Their states are held until another transition edge occurs.

The states of "StatusChanneln" and "OutputsChanneln" are shown for channels 1 and 2.

- Outputs: only Bits 0 to 3 of the 16 bits have a meaning:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0	0	0	0	0	0	0	0	0	0	0	Y3	Y2	Y1	Y0

Significance of the bit: Y0 to Y3:
0: output "0" signal
1: output "1" signal

- StatusChanneln

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0	0	0	0	U/D	0F	UF	0	0	0	EQ	EC	OE	ME	CE

Significance of the bit:
Apart from EC, the bit states are retained if the CPU changes state, from RUN → STOP or STOP → RUN.

CE	Counter state (default value = 0) 0: no enable 1: enabled
ME	Reference input state (default value = 0) 0: no enable 1: enabled
OE	Output Y state (default value = 0) 0: no enable 1: enabled
EC	Equal Flag clear active (default value = 0) If the "ClearEqualn" input function of the "CounterControl" block is set to TRUE, then EC = FALSE. If it is set to FALSE, then EC = TRUE.
EQ	State of Equal flag It is set of actual value = comparison value. It will remain set until a "1" signal is applied to the "ClearEqualn" input of the "CounterControl" block.
UF	State of Underflow flag It is set if the actual value changes from 0 to 4294967296 ($FFFFFFFF_{hex}$). It will remain set until the CLEARUNDERFLOW command is applied to the "Command" input of the "CounterFlags" function block. The output words "Outputs", "StatusChanneln" and "OutputsChanneln" will be set to "0".
OF	State of Overflow flag This is set if the actual value changes from 4294967296 ($FFFFFFFF_{hex}$) to "0". It will remain set until the CLEAROVERFLOW command is applied to the "Command" input of the "CounterFlags" function block. The output words "Outputs", "StatusChanneln" and "OutputsChanneln" will be set to "0".
U/D	State of Up/Down 0: if the actual value has changed from "n" to "n – 1". 1: if the actual value has changed from "n" to "n + 1".

- OutputsChanneln

The bits contained in the word indicate the conditions on which an output depends.

Meaning of the bits

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0	>	=	0	0	>	=	0	0	>	=	0	0	>	=
Output	Y3				Y2				Y1				Y0			

Example:

0021_{hex} (0000 0000 0010 0001) shows that:

- output Y1 is set if the actual value > setpoint (target) value
- output Y0 is set if the actual value = setpoint (target) value.

State display in the controller configuration

The counter module indicates its status in 5 words, within the controller configuration:

1st word: status

2nd word: input data, Low word, channel 1

3rd word: input data, High word, channel 1

4th word: input data, Low word, channel 2

5th word: input data, High word, channel 2

The status word is composed of the following bits:

Channel					Channel 2		Channel 1		Channel 2				Channel 1			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Meaning	0	0	0	0	OF2	UF2	OF1	UF1	EQ2	OE2	ME2	CE2	EQ1	OE1	ME1	CE1

Significance of the bit:
Apart from EC, the bit states are retained if the CPU changes state, from RUN → STOP or STOP → RUN.

CE	Counter state (default value = 0) 0: no enable 1: enabled
ME	Reference input state (default value = 0) 0: no enable 1: enabled
OE	Output Y state (default value = 0) 0: no enable 1: enabled
EQ	State of Equal flag 0: no action 1: if actual value = comparison value It remains set until a "0" signal is applied to the "Compare-Outputn Enable" input of the "CounterControl" block.
UF	State of Underflow flag It is set if the actual value changes from 0 to 4294967296 ($FFFFFFFF_{hex}$). It will remain set until the CLEARUNDERFLOW command is applied to the "Command" input of the "CounterFlags" function block. The output words "Outputs", "StatusChanneln" and "OutputsChanneln" will be set to "0".
OF	State of Overflow flag This is set if the actual value changes from 4294967296 ($FFFFFFFF_{hex}$) to "0". It will remain set until the CLEARUNDERFLOW command is applied to the "Command" input of the "CounterFlags" function block. The output words "Outputs", "StatusChanneln" and "OutputsChanneln" will be set to "0".

FLAG summary

All the flags and their meanings are listed below

Flag	Designation	Meaning
CE	CounterEnable	Pulse inputs are enabled (1) or inhibited (0)[1)]
ME	ReferenceMarker Enable	Reference input is enabled (1) or inhibited (0)[1)]
OE	OutputEnable	Latch output (=) input is enabled (1) or inhibited (0)[1)]
EQ	Equal Flag	The Equal flag is set if actual value = comparison value.[1)]
EC	ClearEqual	Clear Equal flag: after being set ("1" signal) it sets the Latch output (=) to a "0" signal. The EC flag must be reset ("0" signal).
UF	Underflow	It is set if the actual value changes from 0 to 4294967296 ($FFFFFFFF_{hex}$). It will remain set until the CLEAROVERFLOW command is applied to the "Counter-Flags" function block.
OF	Overflow	This is set if the actual value changes from 4294967296 ($FFFFFFFF_{hex}$) to "0". It will remain set until the CLEAROVERFLOW command is applied to the "CounterFlags" function block.

1) Default value = 0

All flags (apart from EC) retain their states if the state of the CPU changes from RUN → STOP or STOP → RUN.

Functional sequence for pulse processing (example)

The following examples illustrate the functional sequence for processing pulses.

Actions that you can perform yourself are marked by the ▶ symbol. The functions are executed by commands that you can enter at the function block inputs, or by applying "0" or "1" signals to the inputs of the "CounterControl" block. Commands are shown in capital (upper case) letters, inputs are shown in lower case letters. The values shown in brackets represent the initial state.

Linear counter

Function	Command or input
▶ Set comparison value 1	WRITESETTINGVALUE1
▶ Set the output specification (the module outputs must be assigned to the comparison value 1 in order to set the specification)	SPECIFYOUTPUT
▶ Set the setpoint value (when using referencing)	WRITEPRESETVALUE
▶ Enable counter inputs[1)]	CounternEnable (1)
▶ Enable Latch/Level outputs[1)]	CompareOutputnEnable (1)
For referencing	
▶ Enable reference inputs[1)]	ReferenceMarkernEnable (1)
Initiate referencing	
When the reference signal is received, the preset value will overwrite the actual value, e.g. actual value = 0.	
▶ Inhibit reference inputs	ReferenceMarkernEnable (0)
Start counting (pulses are counted)	
• If actual value = comparison value 1: – Latch output (=) is set to a "1" signal – Equal flag is set to a "1" signal – Stop counting • If actual value > comparison value 1: – Level output (>) is set to "1"	
▶ Reset Latch output and Equal flag – Set the ClearEqual flag (Equal flag is set to "0", Latch output (=) is set to "0")	Clear Equaln (1)
▶ Reset the ClearEqual flag	ClearEqualn (0)
▶ Set new comparison value1	WRITESETTINGVALUE1
…	
The Overflow flag is set when the count changes from $FFFFFFFF_{hex} \rightarrow 0$:	
▶ Reset Overflow flag	CLEAROVERFLOW
The Underflow flag is set when the count changes from $0 \rightarrow FFFFFFFF_{hex}$	
▶ Reset Underflow flag	CLEARUNDERFLOW

1) Can be performed simultaneously, by using a pulse at the "Strobe" input of the "CounterControl" block.

Ring counter

Function	Command or input
▶ Set start value	WRITEPRESETVALUE
▶ Set end value	WRITESETTINGVALUE1
▶ Set comparison value 2	WRITESETTINGVALUE2
▶ Set the output specification (the module outputs must be assigned to the comparison value 2 in order to set the specification)	SPECIFYOUTPUT
▶ Enable counter inputs[1)]	CounternEnable (1)
▶ Enable Latch output[1)].	CompareOutputnEnable (1)
Start counting (pulses are counted)	
• If actual value = comparison value 2: – Latch output (=) is set to a "1" signal – Equal flag is set to a "1" signal – Stop counting	
▶ Reset Latch output and Equal flag – Set the ClearEqual flag (Equal flag is set to "0", Latch output (=) is set to "0")	Clear Equaln (1)
▶ Reset the ClearEqual flag	ClearEqualn (0)
▶ Set new comparison value 2	WRITESETTINGVALUE2
…	

1) Can be performed simultaneously, by using a pulse at the "Strobe" input of the "CounterControl" block.

4 Counter analog module XIOC-2CNT-2AO-INC

Features

The counter analog module provides two channels for counting up and down, each with a reference input and an analog output (± 10 V).

The counter inputs and the reference input can process 5 V DC differential signals (RS422) of an incremental encoder.
The incremental encoder is connected via the XIOC-TERM-18T or XIOC-TERM-18S clamp terminals with the module. The encoder can receive its power supply from the module.
The power supply is provided by the power supply unit of the CPU.

→ Verify the current consumption of all modules.

The module is a standard I/O module. It can be used on all I/O slots.

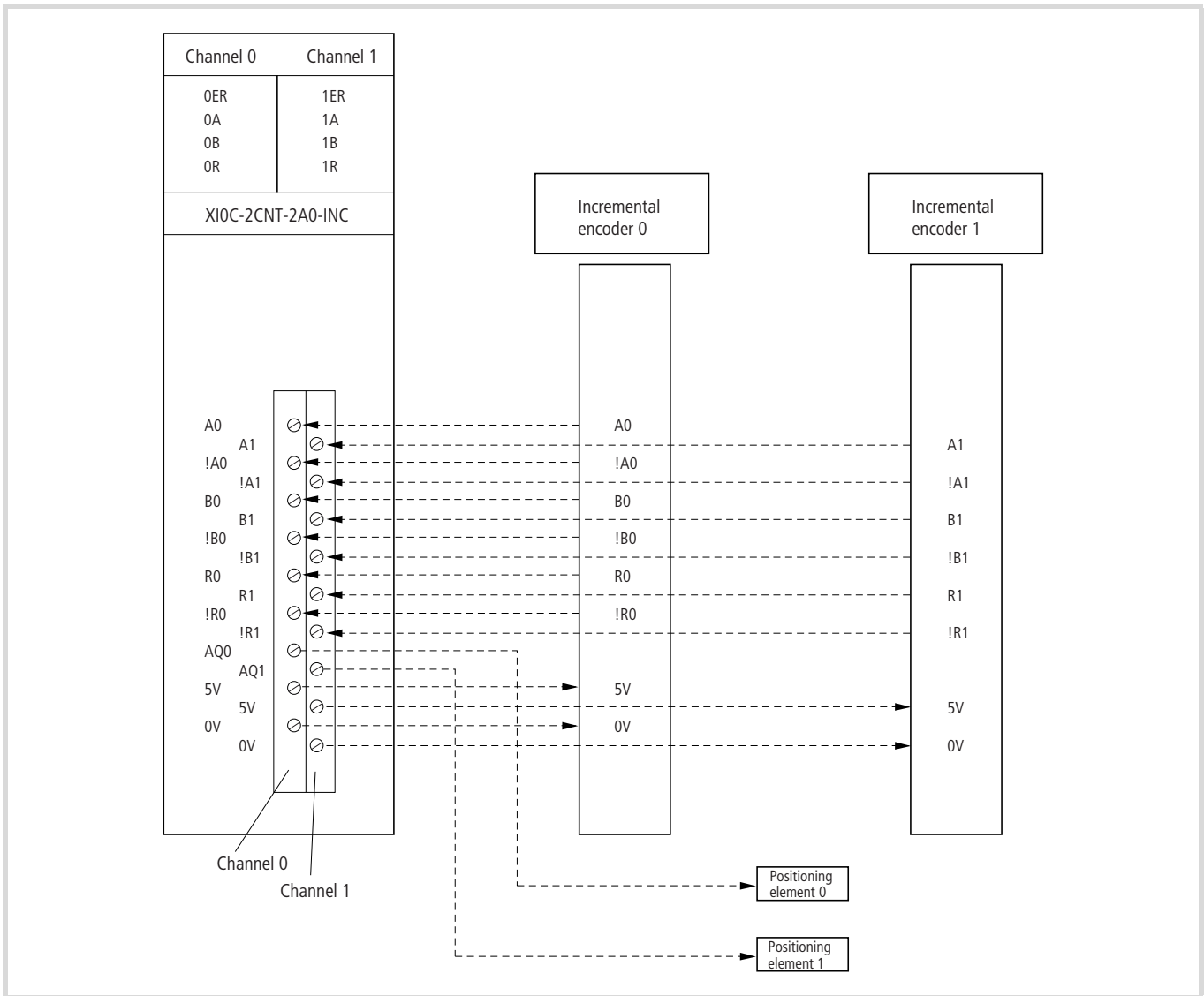


Figure 54: Connections of the counter module

LEDs

The XIOC-2CNT-2AO-INC has eight LEDs for the status display. They are assigned as follows:

Designation	Meaning	Color
ER	Error	red
A	Signal A	green
B	Signal B	green
R	Reference signal	green

The error LED lights when the edges of the A and B signals rise or fall simultaneously.

Programming and configuration

In order to access the module inputs and for actuation of the analog inputs, you can choose between:

- Direct access via the input/output image
- Access via the function blocks.

The function blocks are contained in the "Counter_Analog.lib" library file and have the following function:

XIOC_2CNT2AO_INC referencing and detecting counter values

XIOC_2CNT2AO_ANALOG setting the analog outputs

Furthermore, you must define the following parameters in the configurator of the easySoft-CoDeSys:

- Reference value
- 1, 2, 4 signal edge evaluation
- Number of reference verifications (once, permanent)

Information exchange via the input/output image

You receive the following information via the input map:

- States of signals A, B, R
- Error messages (Error)
- Reference status (Referenced)
- Zero-crossing recognition (Zero Crossing)
- Feedback "Referencing activated"
- Counter status.

You can control the following information via the output image:

- Inhibit the count impulse (Hold)
- Activation of referencing (Activate Referencing)
- Perform a reset (Reset)
- Acknowledgement of zero crossing (Zero Crossing Acknowledge)
- Acknowledge error message (Error Acknowledge)
- Write an analog value.

Input map

A channel occupies the following input bit and words which you can query:

IWn:	Signal states for channels 0 and 1 → table 11
IWn+2:	Counter value, lower Word, channel 0
IWn+4:	Counter value, higher Word, channel 0
IWn+6:	Counter value, lower Word, channel 1
IWn+8:	Counter value, higher Word, channel 1

("n" results from the configuration/slot)

Table 11: IWn: Channel 0 and 1 status signals

Channel	Channel 1								Channel 0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Meaning	tbd	RefAc1	ZC1	Ref1	Error1	R1	b1	A1	tdb	RefAc0	ZC0	Ref0	Error0	R0	B0	A0

Meaning of the bits

Bit	Designation	State	Condition
0/8	Signal A	1	A = "1" and !A = "0"
		0	A = "0" and !A = "1"
1/9	Signal B	1	B = "1" and !B = "0"
		0	B = "0" and !B0 = "1"
2/10	Signal R	1	R = "1" and !R = "0"
		0	R = "0" and !R = "1"
3/11	Error	1	Internal error (A and B edges occur simultaneously)
		0	o.k.
4/12	Ref (Referenced)	1	Referenced
		0	Not referenced
5/13	ZC (Zero Crossing)	1	Counter value = 0
		0	Counter value ≠ 0
6/14	RefAc (Referencing Activated)	1	Referencing activated (set with AcRef)
		0	Referencing not activated
7	tbd	x	Not defined

1) ZC = Zero Crossing (zero crossing bit)
The zero crossing bit is set if the counter value = 0.
If the output bit ZCA is set to "1" in the program, the ZC bit is reset.

Output image

Every channel has the following output bit and word that you can set:

QWn: Control functions, channel 0 and 1 → table 12

QWn+2: Bit 0 to 11: Analog output, channel 0

QWn+4: Bit 0 to 11: Analog output, channel 1

("n" results from the configuration/slot)

Table 12: Control functions, channel 0 and 1,

Channel	**Channel 1**								**Channel 0**							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Meaning	tbd	tbd	tbd	ErAck1	ZCA1	Reset1	AcRef1	Hold1	tbd	tbd	tbd	ErAck0	ZCA0	Reset0	AcRef0	Hold0

Table 13: Meaning of the bits

Bit	**Designation**	**State**	**Condition**
0/8	Hold	0	Enable of the input count impulse (Signals A +B)
		1	Inhibit of the input count impulse
1/9	AcRef[1)] (Activate referencing)	1	Activate referencing
		0	Do not activate referencing
02/10	Reset	0 →1	Asynchronous reset (counter value is set to the reference value) (L → H edge)
		0	–
03/11	ZCA (Zero Crossing Acknowledge)	0 →1	Reset of the zero crossover bit (L → H edge)
		0	–
04/12	ErAck (Error Acknowledge)	0 →1	Reset of the error bit (L → H edge)
		0	–
	tbd	x	Not defined

1) Activate Referencing (AcRef): Activate/deactivate referencing for the reference signal of the encoder

Configuration of the base parameters

- ▶ Open the easySoft-CoDeSys and generate the configuration with the XIOC-2CNT-2AO-INC module.
- ▶ Click on the module in the "PLC Configuration".
- ▶ Open the "Other Parameters" tab and enter the values for:
  - – Edge evaluation
  - – Number of reference checks
  - – Reference value.

Edge evaluation of the count impulse, 1x, 2x or 4x

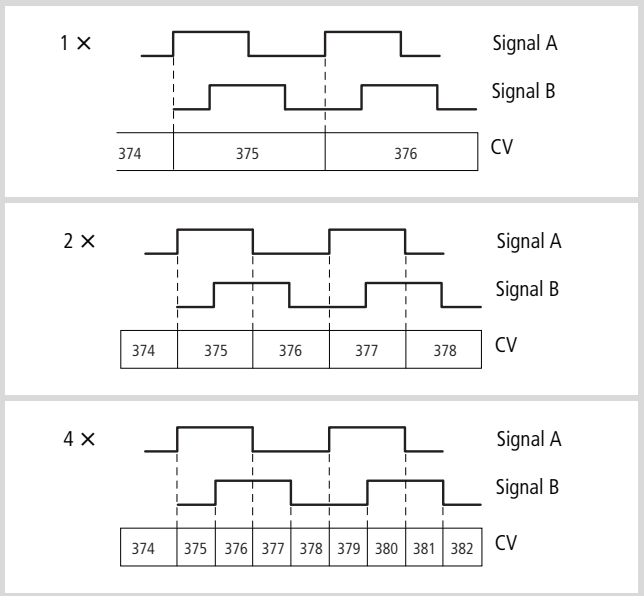


Figure 55: Edge evaluation

① CV = Counter value

② 1 × = single, 2 × = double, 4 × = quadruple

Number of reference verifications (once, permanent)

After the "Activate Referencing" module has been set, the reference pulses of the encoder will be processed by the module. If a reference pulse is detected (signal R: 0 → 1), the counter value is overwritten with the reference value. This occurs once or with every new reference pulse (permanent).

Reference value: A value from 0 to 4294967295 is possible.

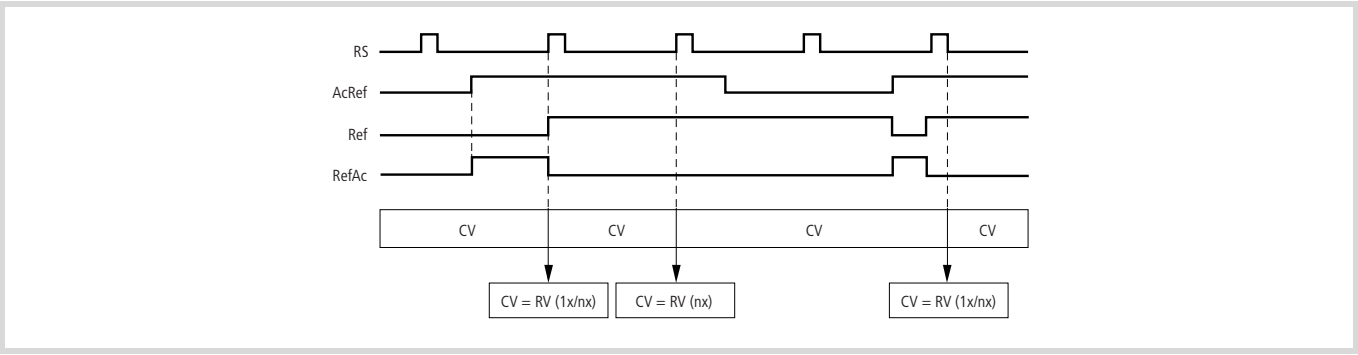


Figure 56: Referencing

Meaning of the signals → table

Table 14: Meaning of the signals

RS	Reference encoder signal	Reference signal from encoder
AcRef	Activate Referencing	Activate referencing
Ref	Referenced	Referenced
RefAc	Referencing activated	Referencing activated
CV	Counter value	Counter value
RV	Reference value	Reference value
CV=RV	The reference value overwrites the count value when setting (1x/nx): once or permanent (nx): permanent	

Explanation:

It is possible to perform referencing once or permanently. The "Activate Referencing (AcRef)" output bit should be set in order to detect the reference signal. The module reacts by setting the "Referencing Activated (RefAc)" input bit. You can query (scan) this bit. When a reference impulse is detected, the "RefAc" input bit is set to a "0" signal and the counter value is overwritten by the reference value. If a further reference impulse is detected, the counter value will be overwritten by the reference value only if you have undertaken the "permanent" setting in the PLC Configuration at ‹Number of references → Other parameters›.

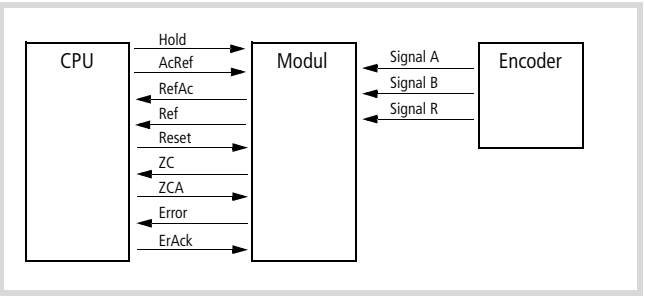


Figure 57: Signal overview

Output of the analog value
The digital value of the output word QWn (n can be seen in the configuration) is converted to an analog voltage. The value range is represented in the following illustration:

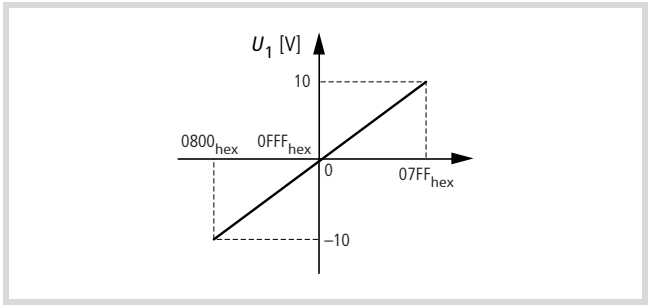


Figure 58: Value range of the analog outputs

Table 15: Value range

Digital value (hex.)	Digital value (dec.)
0	0
7FF	2047
800	2048
FFF	4095

Behavior of the module with CPU RUN/STOP
The CPU transfers the parameters with each STOP → RUN change to the module.

With a "RUN → STOP change" counters are reset to "0". Furthermore, all parameters are erased and the analog outputs are shut down (0 V DC). The module no longer counts further pulses if the CPU is in the "STOP" state.

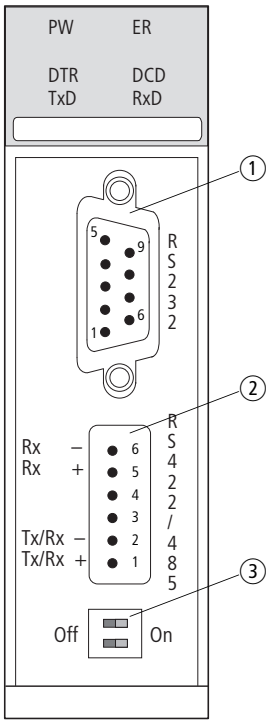
5 Serial interface module XIOC-SER

Features

The module is used in conjunction with the XC100 or XC200 CPU. It has two operating modes available:

- Transparent mode
  For communication with other devices which feature a serial interface. For this purpose an interface is made available in the RS232, RS422 and RS485 versions.
- Suconet-K mode (slave)
  As a Suconet-K slave for communication with the PS4 control system (from XIOC-SER version 02).

On an XC100 a maximum of two modules (COM interfaces) and on a XC200 a maximum of four modules (COM interfaces) can be operated. As the modules XIOC-SER and XIOC-NET-SK-M are addressed via the COM interfaces, the details of the number of modules (COM interfaces) in the PLC refers to both modules.



① **RS232**

SUB-D		
9	–	
8	CTS	Clear To Send
7	RTS	Request To Send
6	DSR	Data Set Ready
5	SGND	Signal Ground
4	DTR	Data Terminal Ready
3	TxD	Transmit Data
2	RxD	Receive Data
1	DCD	Data Carrier Detect

② **RS485**

COMBICON	
6	–
5	–
3, 4	–
2	Tx–/Rx–
1	Tx+/Rx+

② **RS422**

COMBICON	
6	Rx–
5	Rx+
3, 4	–
2	Tx–
1	Tx+

The RS485/-422 interface is galvanically isolated from the bus. The RS232 does not have galvanic isolation features.

③ Switches for bus termination resistors

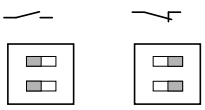


Figure 59: RS232, RS422, RS485 interfaces

LED display

LED display	LED function	Module
PW (Power)	ON	Switched on
ER (Error)	On/Off	Application specific
DTR	ON	Data Terminal Ready
DCD	ON	Data Carrier Detect
TxD	Flashing	Data is being sent
RxD	Flashing	Data is being received

Design of the RS422/RS485 interface

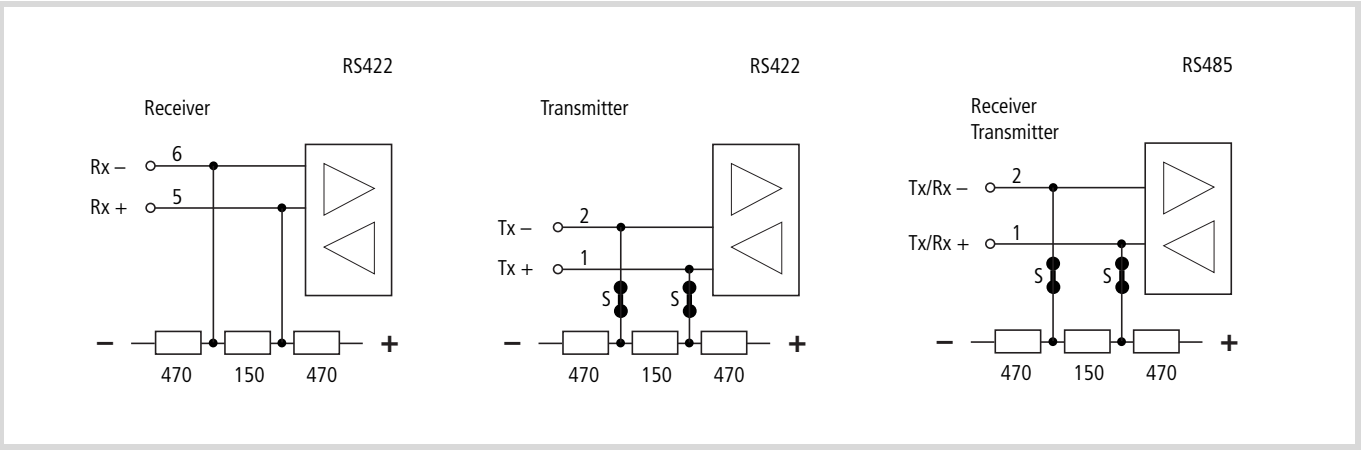


Figure 60: RS422/RS485 interface

S = switch for bus termination resistor

Select the module in the configurator of the easySoft-CoDeSys

- ▶ Open the PLC Configurator
- ▶ Click with the right mouse button on the required slot.
- ▶ Select the "Replace element" command.
- ▶ Select XIOC-SER with a double-click in a new window.

→ The assignment between the slot of the module and the COM… programming language in the configurator: Activate the "Other Parameters" tab and select COM2, 3, 4 or 5 from the "Serial interface" list field → figure 62.

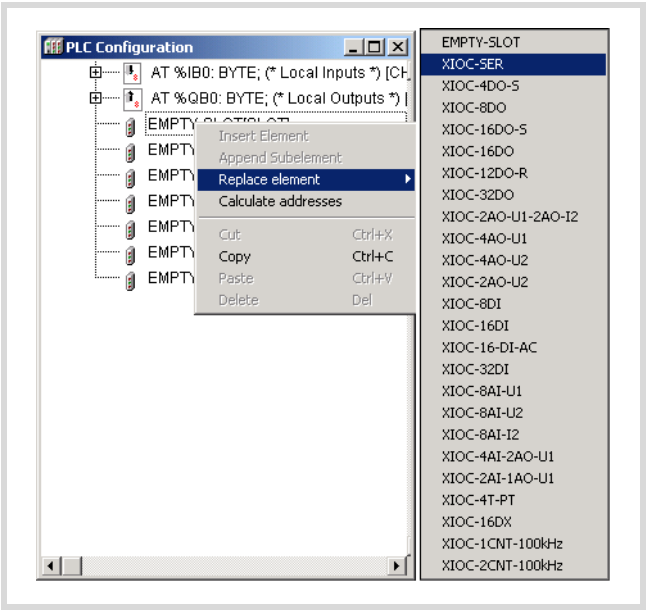


Figure 61: Integrate the module, here: XIOC-SER

Configuration of the interface

After selection of the module, "Transparent" or "Suconet K" (slave) operating mode (bus status) can be clicked in the "Other parameters" tab. The operating mode becomes active after the CPU is switched on. The power supply must be switched off and back on after a selection change.

"Transparent mode" operating mode

In this operating mode the RS232, RS485 or RS422 interface can be used for sending and receiving data.

The RS232 interface is available externally for connection via a 9-pole SUB-D plug (pins); the RS422/RS485 interface can be accessed via a 6-pole springloaded terminal block (COMBICON).

If you select the RS422 or RS485 interfaces, the position of the bus termination resistor switch is important (→ figure 60).

The resistors are integrated into the receive line (Rx-/Rx+) of the RS422 interface. They can be switched in (default setting) or out on the send line of the RS422 as well as the RS485 interface.

Both switches must be in the same setting position to guarantee perfect communication.

An example for parameter settings in transparent mode is shown in Figure 62. The parameters can be modified by a click on the arrow button.

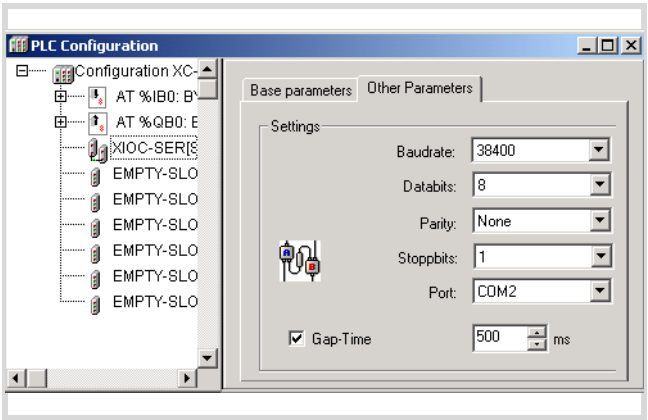


Figure 62: Default parameter in transparent mode

Serial interface:
Here you select the logical name of your interface. The serial interface module can be addressed by this name in the user program.

Setting gap time:
This function is not activated in the basic setting. The gap time is used to tolerate possible intervals when receiving telegram characters (gaps in telegrams).

"Suconet-K mode (slave)" operating mode

In this operating mode, the variable length data blocks are transferred between the XIOC-SER (Suconet K slave) module and a Suconet-K master of the PS4 system.

- ▶ Set the mode of operation (bus status) to "Suconet K" in the "Other Parameters" tab of the easySoft-CoDeSys configurator and match the parameters accordingly.
  - – Define the slave address which is displayed in the configurator of the Sucosoft S40 for the slave, in the "Suconet K address" field.
  - – Define the send and receive data count (maximum 120 bytes). The send data count of the slave (XIOC-SER) must correspond with the receive data of the master. The same applies for the send data (master) → Receive data (slave).
  - – Serial interface: Here you select the logical name of your interface. The serial interface module can be addressed by this name in the user program.
  - – Specify the Suconet-K device type. Each station on the Suconet-K rung is uniquely identified by a device type. By default, the device type for the XIOC-SER is set to SIS-TYP-A0EF, but you can change this to any other type. An XIOC-SER can therefore also be configured as a replacement for a previous Suconet-K station (for example a PS4-341-MM1). You do not have to modify the PS40 program for this purpose.

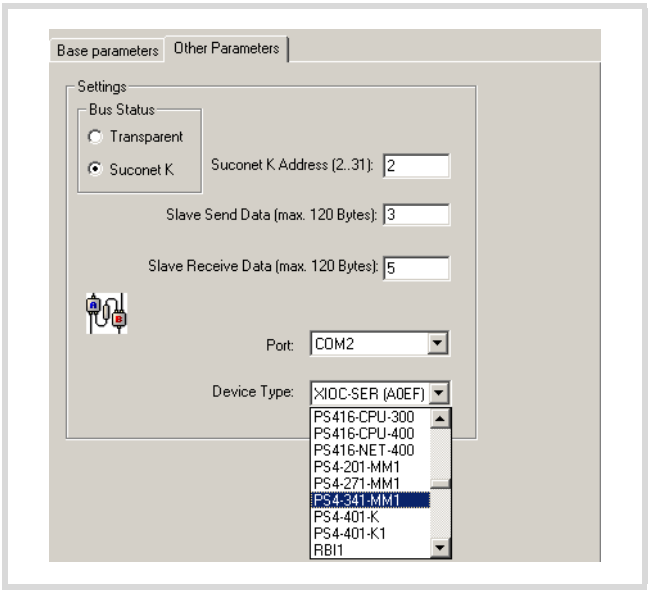


Figure 63: Communications parameters for the Suconet K operating mode

Master connection ↔ XIOC-SER

The RS485 interface is active in the Suconet K operating mode.

Master	TA/RA---------- Tx/Rx+	XIOC-SER
	TB/RB ---------- Tx/Rx-	

Setting the bus termination resistors

Set the bus termination resistors. If the module is physically the first or last module on the end of a line, set both of the S switches (→ fig. 60) to the "ON" setting (default setting). Both of the switches must be set to "OFF" at all other positions on the line. Both switches must be in the same setting position to guarantee perfect communication.

Configuration in the Sucosoft S40

In the configurator of the Sucosoft S40, extend the master with the XIOC-SER module by selecting the module from a list. Use the same device type that you have selected in the list field "Device type" in the configuration dialog of the XIOC-SER. The address is displayed in the parameter window after selection. Enter the data count in the "send data" and "receive data" fields.

Diagnostics on the master

The diagnostics byte of the slave (XIOC-SER) can be read in the master program. The method for reading the diagnostics byte can be found in the documentation of the master. The diagnostics byte of the master has the following structure:

Bit	Meaning
0	Reserved
1	0 = Station in "RUN"
	1 = Station in "Halt"
2	0 = ok
	1 = Length fault of the received data
3	Reserved
4	Reserved
5	Reserved
6	0 = ok
	1 = No connection
7	0 = ok
	1 = Incorrect device type

Diagnostics on the slave

The diagnostics is performed by the "Suconet K-Slave" function block. You can query both of the "xMasterDiscon" and "xMasterStop" outputs on the module.

You receive the following messages:

The "Suconet K-Slave" function block can be found in the "Suconet K.lib" library. It is described in the manual MN05010002Z-EN (previously AWB2786-1456GB) (Function blocks for easySoft-CoDeSys).

xMasterDiscon	0 = Master connected
	1 = Master disconnected
xMasterStop	0 = Master in RUN
	1 = Master in STOP

Access to the receive and send data

Access from the user program to the data of the XIOC-SER module is implemented in transparent mode with the aid of functions from the xSysCom100.lib library, from the SysLibCom.lib or xSysCom200.lib.

The functions are described in the manuals MN05003004Z-EN (previously AWB2724-1453GB) for XC100 and MN05003001Z-EN (previously AWB2724-1491GB) for XC200.

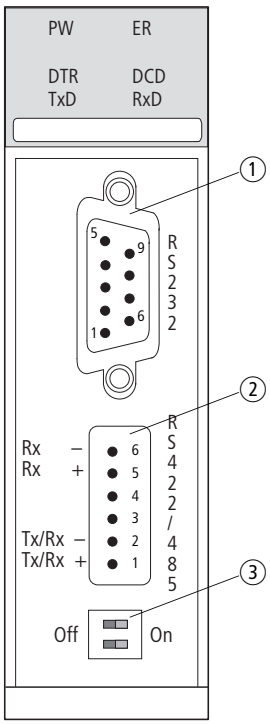
In the Suconet K operating mode you implement the "Suconet K-Slave" function block.

The "Suconet K-Slave" function block can be found in the "Suconet K.lib" library. It is described in the manual MN05010002Z-EN (previously AWB2786-1456GB) (Function blocks for easySoft-CoDeSys).

6 Telecontrol module XIOC-TC1

Features

The module is used in conjunction with the XC200 CPU. It communicates via RS232, RS422, and RS485 interfaces with other devices that have a serial interface.



① **RS232**

SUB-D

9	–	
8	CTS	Clear To Send
7	RTS	Request To Send
6	DSR	Data Set Ready
5	SGND	Signal Ground
4	DTR	Data Terminal Ready
3	TxD	Transmit Data
2	RxD	Receive Data
1	DCD	Data Carrier Detect

② **RS485**

COMBICON

6	–
5	–
3, 4	–
2	Tx–/Rx–
1	Tx+/Rx+

② **RS422**

COMBICON

6	Rx–
5	Rx+
3, 4	–
2	Tx–
1	Tx+

The RS485/422 interface is galvanically isolated from the bus. The RS232 does not have galvanic isolation features.

③ Switches for bus termination resistors

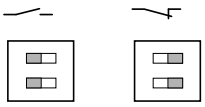


Figure 64: RS232, RS422, RS485 interfaces

LED display

LED display	LED function	Module
PW (Power)	ON	Switched on
ER (Error)	On/Off	Application specific
DTR	ON	Data Terminal Ready
DCD	ON	Data Carrier Detect
TxD	Flashing	Data is being sent
RxD	Flashing	Data is being received

Design of the RS422/RS485 interface

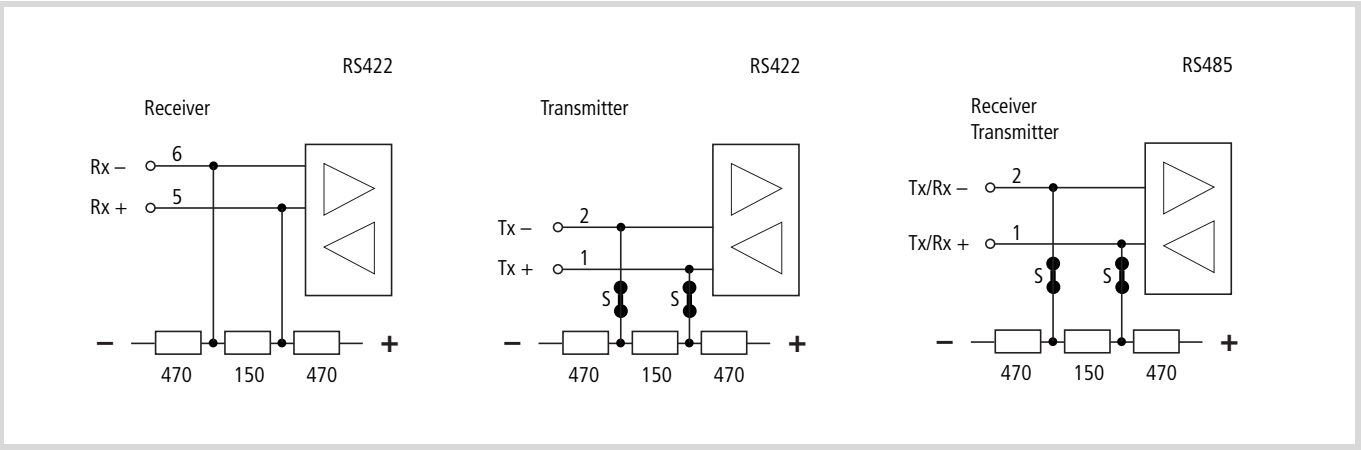


Figure 65: RS422/RS485 interface

S = switch for bus termination resistor

Select the module in the configurator of the easySoft-CoDeSys

- ▶ Open the PLC Configurator
- ▶ Click with the right mouse button on the required slot.
- ▶ Select the "Replace element" command.
- ▶ Select XIOC-TC1 with a double-click in a new window.

→ The assignment between the slot of the module and the COM… programming language in the configurator: Activate the "Other Parameters" tab and select COM2, 3, 4 or 5 from the "Serial interface" list field → figure 66.

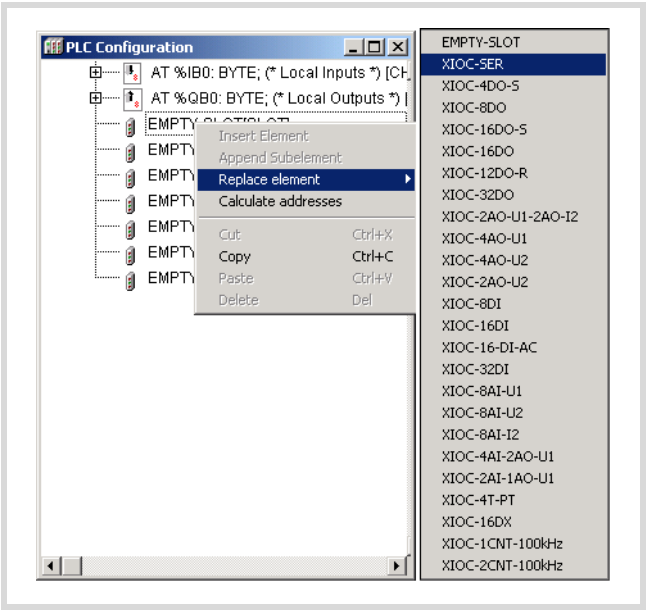


Figure 66: Integrate the module, here: XIOC-TC1

Configuration of the interface

After selection of the card, "Transparent" or "Suconet K" (slave) operating mode (bus status) can be clicked in the "Other parameters" tab. The operating mode becomes active after the CPU is switched on. The power supply must be switched off and back on after a selection change.

"Transparent mode" operating mode

In this operating mode the RS232, RS485 or RS422 interface can be used for sending and receiving data.

The RS232 interface is available externally for connection via a 9-pole SUB-D plug (pins); the RS422/RS485 interface can be accessed via a 6-pole springloaded terminal block (COMBICON).

If you select the RS422 or RS485 interfaces, the position of the bus termination resistor switch is important (→ figure 65).

The resistors are integrated into the receive line (Rx-/Rx+) of the RS422 interface. They can be switched in (default setting) or out on the send line of the RS422 as well as the RS485 interface.

Both switches must be in the same setting position to guarantee perfect communication.

An example for parameter settings in transparent mode is shown in figure 67. The parameters can be modified by a click on the arrow button.

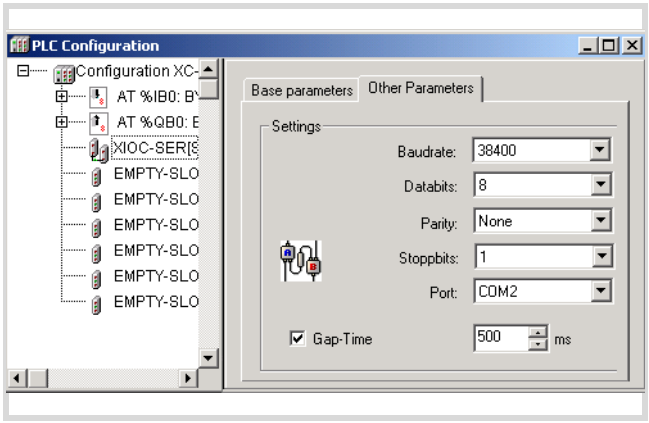


Figure 67: Default parameter in transparent mode

Serial interface:
Here you select the logical name of your interface. The serial interface module can be addressed by this name in the user program.

Setting gap time:
This function is not activated in the basic setting. The gap time is used to tolerate possible intervals when receiving telegram characters (gaps in telegrams).

Access to the receive and send data

Access from the user program to the data of the XIOC-SER module is implemented in transparent mode with the aid of functions, from the library or xSysCom200.lib. The functions are described in the manuals MN05003001Z-EN (previously AWB2724-1491GB) for XC200.

Communications library for DNP3 protocol V1.1

The DNP3 protocol (DNP= distributed network protocol) implements secure data transfer between two communication partners. The protocol was implemented for the XC200 control system in connection with the XIOC-TC1 telecontrol module. It represents an outstation from the DNP3 perspective (outstation is the DNP3 designation for 'slave') and answers the DNP3 master's corresponding data queries.

The DNP3's library functions, which were developed for the XC200 controller and CoDeSys programming system, are described below. The library implements the functionality in accord with DNP3 interoperability level 2 (DNP3-L2) pursuant to the DNP3 specification, part 8. Cited DNP3 documents reflect the status as of 15 Dec 2007.

Prerequisites

Minimum prerequisites for use are

- PLC: XC200
- Operating system version 1.05.03 or higher
- XIOC-TC1
- easySoft-CoDeSys version V2.3.9 +
- Library: DNP3.lib

DNP3 communication and data model

DNP implements a secure data connection between master and outstation. Communication is conducted here via five data objects:

- Binary Inputs
- Binary Outputs
- Analog Inputs
- Analog Outputs
- Counter

These are addressed through indices. Data is always considered here from the master's point of view:

The master reads **binary Inputs**; so the outstation writes to the master's binary input data object. The complete communication relationship is obtainable from the following figure.

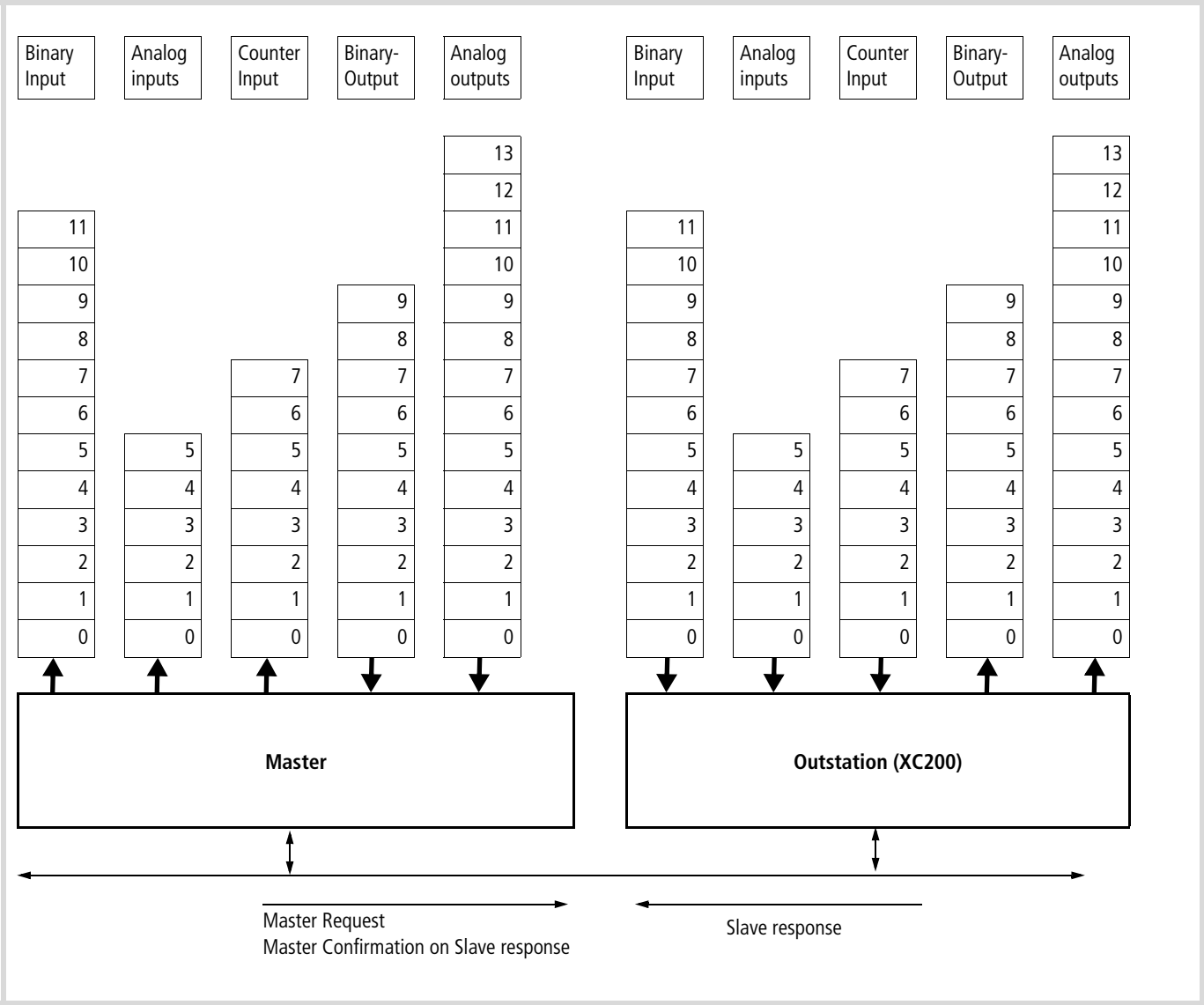


Figure 68: DNP3 master-outstation data objects and data flow

Function summary

The following functions are implemented for DNP3 protocol use:

Server functions		→ Page
DNP3_Create	Connecting the DNP3 server	66
DNP3_Destroy	Deleting the DNP3 server	66
DNP3_Execute	DNP3 state machine call	66
DNP3_OpenCom	Connection to the communication interface	67
DNP3_CloseCom	Stop the communication connection.	67
Read, write data		
DNP3_SetBI	Write (the master's) digital inputs.	67
DNP3_SetAI	Write (the master's) analog inputs.	68
DNP3_SetCI	Write the master's counter inputs.	68
DNP3_GetAO	Read the master's analog outputs.	70
DNP3_GetBO	Read the master's digital outputs.	70
DNP3_GetBI	Read the digital inputs in the outstation (read back the self-written inputs).	69
DNP3_GetAI	Read the outstation's analog inputs (read back the self-written inputs).	69
DNP3_GetCI	Read the outstation's counter inputs (read back the self-written inputs).	70
Write event-controlled data		
DNP3_Set_BIwEvent	Write the master's event digital inputs.	68
DNP3_Set_AIwEvent	Write (the master's) event analog inputs.	69
DNP3_Set_CIwEvent	Write (the master) event-counter inputs.	69
Test function		
DNP3_SetDbgLevel	Set debug level.	70

Data direction is always to be seen from the master's point of view here. So writing the digital input from the outstation's point of view means writing the digital master's inputs.

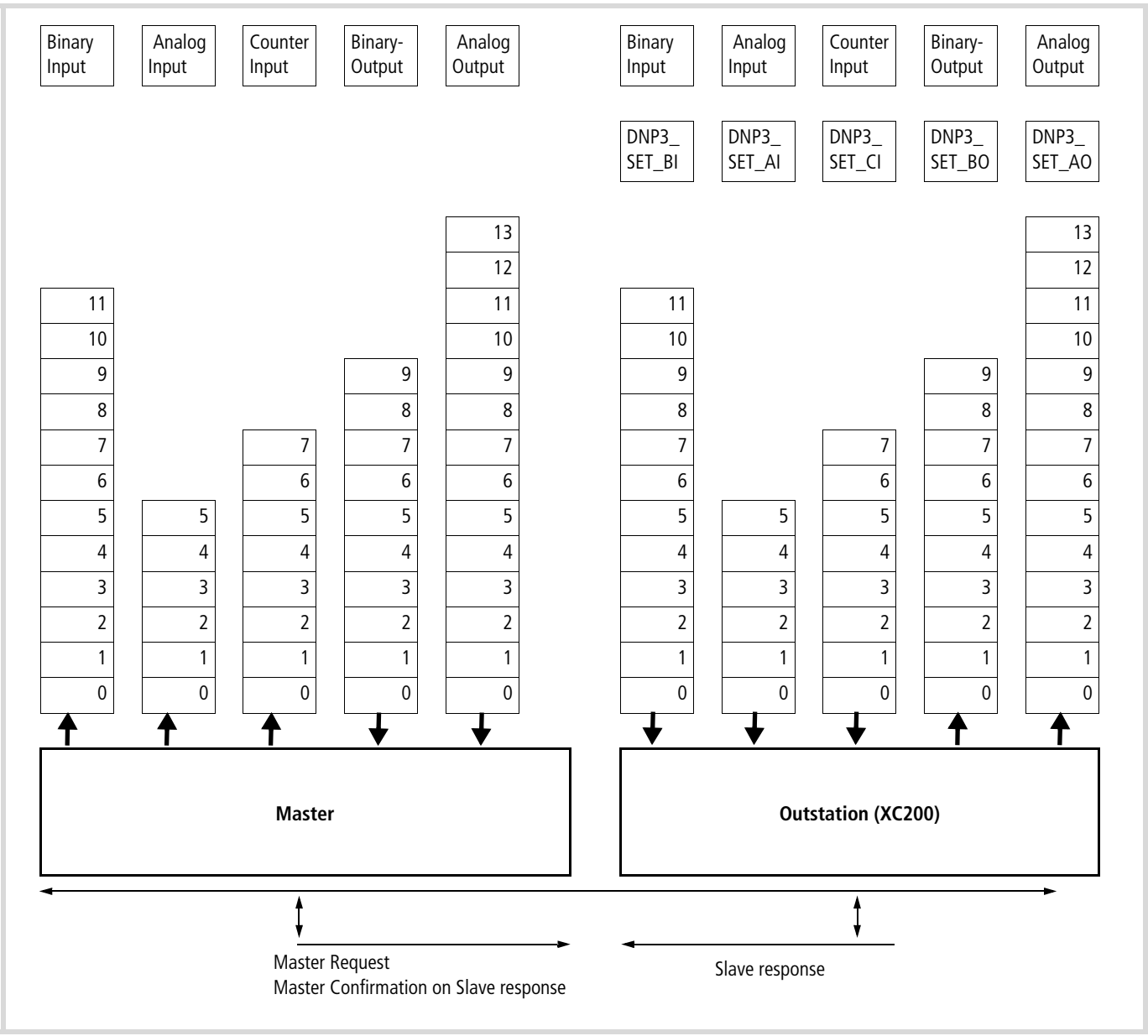


Figure 69: Assignment of functions to data objects

The functions use return values from the DNP3Result enumeration type.

Possible error causes are itemized in the following presentation. Those respectively relevant are listed in the subsequently following description of the functions.

```
TYPE DNP3RESULT :
(
     DNP3RES_OK := 0,

(* Data Link Layer *)
     DNP3DLLRES_InvalidEventForState := 20, (* internal usage *)
     DNP3DLLRES_InvalidStateCode := 21, (* internal usage *)

(* TransportFunction *)
     DNP3TFRES_SenderBusy := 40, (* internal usage *)

(* Application Layer *)
     DNP3ALRES_WrongIndex := 60, (* wIndex exeeds array bounds *)
     DNP3ALRES_InvalidFunctionCode := 61, (* internal usage *)
     DNP3ALRES_InvalidGroup := 62, (* internal usage *)
     DNP3ALRES_InvalidVariation := 63, (* internal usage *)
     DNP3ALRES_InvalidQualCode := 64, (* internal usage *)
     DNP3ALRES_InvalidRangeValue := 65, (* internal usage *)
     DNP3ALRES_InvalidTimeValue := 66, (* internal usage *)
     DNP3ALRES_CommonTimeOfOccurenceNotSet := 70, (* internal usage *)

(* PLC level *)
     DNP3PLCRES_WrongHandle := 80, (* dwDNP3Handle invalid*)
     DNP3PLCRES_CantUseSysComDll := 81, (* can´t create xSysCOM *)
     DNP3PLCRES_CantOpenComPort := 82, (* can´t open COM port *)
     DNP3PLCRES_ComPortNotOpened := 83, (* COM not opened *)
     DNP3PLCRES_CantCreateDNP3 := 84, (* allocatiobn of internal memory failed *)
     DNP3PLCRES_ArraySizeToHigh := 85, (* one or more of the array sizes is to high *)
     DNP3PLCRES_ArraySizeNotSet := 86, (* one or more of the array sizes is zero *)
     DNP3PLCRES_NotAllowedNullArg := 87, (* one of used call arguments is a NULL-Pointer *)

(* Execute events *)
     DNP3PLCRES_DataChangedByMaster := 100,(* not used *)
     DNP3RES_FORCE_DWORD:=4294967295
);
END_TYPE
```

Function DNP3_Create

```
FUNCTION DNP3_Create : DNP3RESULT
VAR_INPUT
  wAddress : WORD; (* IN: own DNP3 address *)
  pAppDataCfg : POINTER TO DNP3APPDATACFG;      (* IN: pointer to a structure filled with the sizes of application data arrays *)


  pExtCfg : POINTER TO DNP3EXTCFG;              (* IN: pointer to a structure filled with extended config information for DNP3. *)


  phDNP3 : POINTER TO DWORD;                    (* OUT: DNP3-handle *)
END_VAR
```

A DNP3 server structure is created in the XC200 controller with the DNP3_Create function. The DNP3 outstation's address and size of the areas for the data fields is transferred. These are allocated in the operating system's memory, so the need no memory space in the controller's application program memory area.

The function returns a reference to the DNP3 server in the phDNP3 variable, which is used in the further running of the other access functions.

The DNP3APPDFATACFG structure is needed to transfer the size of the data fields for communication. The number of inputs for each of the five data fields that can exchanged between the outstation and the DBP3 master data is defined here.

```
TYPE DNP3APPDATACFG :
STRUCT
  wBISize : WORD:=0; (* Size of Binary-Input array. Must be
  set to 1..1024 *)
  wAISize : WORD:=0; (* Size of Analog-Inputs array. Must be
  set to 1..1024 *)
  wCISize : WORD:=0; (* Size of Counter-Input array. Must be
  set to 1..1024 *)
  wBOSize : WORD:=0; (* Size of Binary-Output array. Must be
  set to 1..1024 *)
  wAOSize : WORD:=0; (* Size of Analog-Output array. Must be
  set to 1..1024 *)
END_STRUCT
END_TYPE
```

Further information about the DNP3 library's configuration occurs via the DNP3EXTCFG structure.

- Timeout
- Unsolicited Response

The DNP3CREATE function returns the function call's result via the general DNP3RESULT result structure. Possible errors are:

```
(* Create/Initialize DNP3 interface and allocate all arrays
  DNP3RES_OK - no errors
  DNP3PLCRES_CantAllocDNP3 - allocation of internal memory
  failed
  DNP3PLCRES_NotAllowedNullArg - one of used arguments is a
  NULL-Pointer
  DNP3PLCRES_ArraySizeToHigh - one or more of the array sizes
  is >1024
  DNP3PLCRES_ArraySizeNotSet - one or more of the array sizes
  is zero
*)
```

Function DNP3_Destroy

```
FUNCTION DNP3_Destroy:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;   DNP3 handle to DNP3 interface
END_VAR
```

The function closes a created DNP3 server and releases all allocated memory areas.

Return value:

```
  DNP3RES_OK               No errors
  DNP3PLCRES_WrongHandle   Invalid dwDNP3Handle
```

Function DNP3_Execute

```
UNCTION DNP3_Execute:DNP3RESULT
VAR_INPUT
 dwDNP3Handle : DWORD;     DNP3 handle to DNP3 interface
END_VAR
```

The function starts the DNP3 state machine. This function must be called cyclically. The function reads pending data from the XIOC-TC1 module and executes the contingent tasks.

Return value:

```
  DNP3RES_OK                    No errors
  DNP3PLCRES_WrongHandle        Invalid dwDNP3Handle
  DNP3PLCRES_ComPortNotOpened   COM not opened
  DNP3PLCRES_CantUseSysComDll   SysCom missing
Function DNP3_OpenCom
```

FUNCTION DNP3_OpenCom : DNP3RESULT

```
VAR_INPUT
  dwDNP3Handle : DWORD;   DNP3 handle to DNP3 interface
  wPortNr : WORD;         COM port number. See xSysCom200
                          library
  wBaudrate : WORD;       See xSysCom200 library
  wStopbits : WORD;       See xSysCom200 library
  wParity : WORD;         See xSysCom200 library
  wDataLength : WORD;     See xSysCom200 library
END_VAR
```

The function establishes the connection between the created DNP3 server and the XIOC-TC1 module. This logical COM number (COM2, 3, 4, 5) was assigned while defining the module's parameters in the CoDeSys control configurator. This logical number is now transferred to wPortNr.

The XsysCom200.lib library contains the definitions for defining the interface's parameters.

Example of the wPortMr port number:

```
TYPE COMPORTS :
( COM1   :=1, (* COM1     : OnBoard RS232 *)
  COM2,       (* COM2 - 5 : XIOC-SER, XIOC-TC1 *)
  COM3,
  COM4,
  COM5 ) := COM1;
END_TYPE
```

Function DNP3_CloseCom

```
FUNCTION DNP3_CloseCom:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;    DNP3 handle to DNP3 interface
END_VAR
```

The function releases the connection between the created DNP3 server and the communication module. Communication via DNP3_Execute is no longer possible.

The connection can be reactivated with DNP3_OpenComm().

Return value:

```
  DNP3RES_OK                    No errors
  DNP3PLCRES_WrongHandle        Invalid dwDNP3Handle is used
  DNP3PLCRES_ComPortNotOpened   COM not opened
  DNP3PLCRES_CantUseSysComDll   SysCom missing
```

Function DNP3_SetBI

```
FUNCTION DNP3_SetBI:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;  DNP3 handle to DNP3 interface
  wIndex : WORD;         Index of element
  bValue : BYTE;         Value that will be written to array
                         element
END_VAR
```

The function describes an element in the digital inputs range. The wIndex 0 statement describes the first element. The wBlSize variable's statement in the DNP3_Create function call defines the highest index. So here the statement is wBlsize-1.

Special DNP3 conventions are to be heeded during digital data construction in the description.

Binary values are represented by one byte. The construction thereby corresponds to the definition pursuant to DNP3 object library (DNP3 Specification, volume 6, part 2 (Binary input with flags)).

Bit	Flag meaning
0	Online (0 inactive, 1 active)
1	Restart (0, normal, 1 variable in initial status)
2	Comm_Lost (0, normal, 1 Value represents last valid data)
3	Remote_Forced (0, normal, 1 Value forced by external device)
4	Local_Forced 0, normal, 1 forced by local device e.g. HMI)
5	Chatter_Filter
6	Reserved (always 0)
7	State : 0.1 representing the state of physical or logical input

The DNP3 specification (volume 6, part 1, Basics p. 21 ff) contains the flags' exact description.

Return value:

```
  DNP3RES_OK                No errors
  DNP3ALRES_WrongIndex      When the wIndex exceed array bounds
  DNP3PLCRES_WrongHandle    Invalid dwDNP3Handle
```

Function DNP3_SetAI

```
FUNCTION DNP3_SetAI:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;   DNP3 handle to DNP3 interface
  wIndex : WORD;          Index of element
  wValue : WORD;          Value that will be written to array
                          element
  bFlags:Byte;            Flags that will be written to array
                          element
END_VAR
```

The function describes an element in the analog inputs range. The wIndex 0 statement describes the first element. The wAlSize variable statement in the DNP3_Create function call defines the highest index. So the statement here is wAlsize-1.

The flags' definition almost corresponds to that for the binary data (bit 7 is always 0 here).

Bit	Flag meaning
0	Online (0 inactive, 1 active)
1	Restart (0, normal, 1 variable in initial status)
2	Comm_Lost (0, normal, 1 Value represents last valid data)
3	Remote_Forced (0, normal, 1 Value forced by external device)
4	Local_Forced 0, normal, 1 forced by local device e.g. HMI)
5	Chatter_Filter
6	Reserved (always 0)
7	0

The flag byte's configuration and meaning

Return value:

```
  DNP3RES_OK              No errors
  DNP3ALRES_WrongIndex    wIndex exceeds array bounds
  DNP3PLCRES_WrongHandle  Invalid dwDNP3Handle
```

Function DNP3_SetCI

```
FUNCTION DNP3_SetCI:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;   DNP3 handle to DNP3 interface
  wIndex : WORD;          Index of element
  dwValue : DWORD;        Value that will be written to array
                          element
  bFlags:Byte;            Flags that will be written to array
                          element
END_VAR
```

The function describes an element in the counter range. The wIndex 0 statement describes the first element. The wClSize variable statement in the DNP3_Create function call defines the highest index. So the statement is wClsize-1 here.

See Page 71 for flag construction and definition.

Return value:

```
  DNP3RES_OK              No errors
  DNP3ALRES_WrongIndex    wIndex exceeds array bounds
  DNP3PLCRES_WrongHandle  Invalid dwDNP3Handle
```

Function DNP3_SetBIwEvent

FUNCTION DNP3_SetBIwEvent:DNP3RESULT

```
VAR_INPUT
  dwDNP3Handle : DWORD;   DNP3 handle to DNP3 interface
  wIndex : WORD;          Index of element
  bValue : BYTE;          Value that will be written to array
                          element
END_VAR
```

The function describes an element in the digital inputs range. The wIndex 0 statement describes the first element. The wBlSize variable statement in the DNP3_Create function call defines the highest index. So the statement is wBlsize-1 here.

The master can query for specific data changes in contrast to the DNP3_SetBl function. So a change to the data with the DNP3_SETBlwEvent function in the outstation is registered directly as a change with the master. Otherwise the master would always have to compare between old and new values to determine differences.

Special DNP3 conventions are to be heeded during digital data construction in the description.

Binary values are represented by one byte. The construction thereby corresponds to the definition pursuant to DNP3 object library (DNP3 Specification, volume 6, part 2 (Binary input with flags)).

See Page 71 for flag construction and definition.

Return value:

```
  DNP3RES_OK              No errors
  DNP3ALRES_WrongIndex    When the wIndex exceed array
                          bounds
  DNP3PLCRES_WrongHandle  Invalid dwDNP3Handle
```

Function DNP3_SetAIwEvent

```
FUNCTION DNP3_SetAIwEvent:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;   DNP3 handle to DNP3 interface
  wIndex : WORD;          Index of element
  wValue : WORD;          Value that will be written to array
                          element
  bFlags:Byte;            Flags that will be written to array
                          element
END_VAR
```

The function describes an element in the analog inputs range. The wIndex 0 statement describes the first element. The wAISize variable statement in the DNP3_Create function call defines the highest index. So the statement here is wAIsize-1.

The master can specifically query data changes in contrast to the DNP3_SetAI function. A data change with the DNP3_SETAIwEvent function in the outstation is thus registered directly as a change in the master. Otherwise the master must always compare between old and new values to determine differences.

See Page 71 for flag construction and definition.

Return value:

```
  DNP3RES_OK              No errors
  DNP3ALRES_WrongIndex    wIndex exceeds array bounds
  DNP3PLCRES_WrongHandle  nvalid dwDNP3Handle
```

Function DNP3_SetCIwEvent

```
FUNCTION DNP3_SetCI:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;   DNP3 handle to DNP3 interface
  wIndex : WORD;          Index of element
  dwValue : DWORD;        Value that will be written to array
                          element
  bFlags:Byte;            Flags that will be written to array
                          element
END_VAR
```

The function describes an element in the counter range.
The wIndex 0 statement describes the first element. The wCISize variable statement in the DNP3_Create function call defines the highest index. So the statement is wCIsize-1 here.

The master can query data changes specifically in contrast to the DNP3_SetCI function. A data change with the DNP3_SETCIwEvent function in the outstation is thus registered directly as a change in the master. Otherwise the master must always compare between old and new values to determine differences.

See Page 71 for flag construction and definition.

Return value:

```
  DNP3RES_OK              No errors
  DNP3ALRES_WrongIndex    wIndex exceeds array bounds
  DNP3PLCRES_WrongHandle  Invalid dwDNP3Handle
```

Function DNP3_GetBI

```
FUNCTION DNP3_GetBI:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;          DNP3 handle to DNP3 inter-
                                 face
  wIndex : WORD;                 Index of element
  pbValue : Pointer to BYTE;     Pointer to variable that
                                 will be filled with
                                 requested value
END_VAR
```

The function reads an element in the digital inputs range. Thus data written with DNP3_SetBI can be read back. The wIndex 0 statement describes the first element. The wBLSize variable statement in the DNP3_Create function call defines the highest index. So the statement is wBLsize-1 here.

The notes concerning digital data configuration are to be heeded when interpreting the values.

Return value:

```
  DNP3RES_OK              No errors
  DNP3ALRES_WrongIndex    wIndex exceeds array bounds
  DNP3PLCRES_WrongHandle  Invalid dwDNP3Handle is used
```

Function DNP3_GetAI

```
FUNCTION DNP3_GetAI:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;          DNP3 handle to DNP3 interface
  wIndex : WORD;                 index of element
  pwValue : Pointer to WORD;     Pointer to variable that will
                                 be filled with requested
                                 value
  pbFlags: Pointer to Byte;      Pointer to variable that will
                                 be filled with requested
                                 flags
END_VAR
```

The function reads an element in the analog inputs' range. This way, data written with DNP3_SetAI can be read back. The windex 0 statement describes the first element. The wAISize variable statement in the DNP3_Create function call defines the highest index; the statement is thus wAIsize-1 here. The data for values and flags are returned via two pointers. For the flags' configuration, see Page 71 for flag construction and definition.

Return value:

```
  DNP3RES_OK              No errors
  DNP3ALRES_WrongIndex    wIndex exceeds array bounds
  DNP3PLCRES_WrongHandle  Invalid dwDNP3Handle is used
```

Function DNP3_GetCI

```
FUNCTION DNP3_GetCI:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;          DNP3 handle to DNP3 interface
  wIndex : WORD;                 index of element
  pdwValue : Pointer to DWORD;   Pointer to variable that will
                                 be filled with requested
                                 value
  pbFlags: Pointer to Byte;      Pointer to variable that will
                                 be filled with requested
                                 flags
  END_VAR
```

The function reads an element in the counter range. Thus data written with DNP3_SetCI can be read back. The wIndex 0 statement describes the first element. The wCISize variable statement in the DNP3_Create function call defines the highest index. The statement here is thus wClsize-1.

Return value:

```
  DNP3RES_OK               No errors
  DNP3ALRES_WrongIndex     wIndex exceed array bounds
  DNP3PLCRES_WrongHandle   Invalid dwDNP3Handle
```

Function DNP3_GetBO

```
FUNCTION
DNP3_GetBO:DNP3RESULT
VAR_INPUT
  dwDNP3Handle : DWORD;        DNP3 handle to DNP3 interface
  wIndex : WORD;               Index of element
  pbValue : Pointer toByte;    Pointer to variable that will
                               be filled with requested value
END_VAR
```

The function reads an element in the digital output range (master's output = input for the outstation). The windex 0 statement points to the first element. The wBoSize variable's statement in the DNP3_Create function call defines the highest index. So the statement here is wBosize-1.

The notes concerning digital data configuration are to be heeded when interpreting the values.

Return value:

```
  DNP3RES_OK               No errors
  DNP3ALRES_WrongIndex     wIndex exceed array bounds
  DNP3PLCRES_WrongHandle   Invalid dwDNP3Handle is used
```

Function DNP3_GetAO

```
FUNCTION DNP3_SetAO:DNP3RESULT
VAR_INPUT
     dwDNP3Handle : DWORD;      DNP3 handle to DNP3 interface
     wIndex : WORD;             index of element
     pwValue : WORD;            Pointer to variable that will be
                                filled with requested value
     pbValue : Byte;            requested flagsvalue
END_VAR
```

The function reads an element in the analog outputs' range (the master's output = input for the outstation). The wIndex 0 statement points to the first element. The wAOSize variable statement in the DNP3_Create function call defines the highest index. The statement here is thus wAOsize-1.

Return value:

```
  DNP3RES_OK               No errors
  DNP3ALRES_WrongIndex     wIndex exceed array bounds
  DNP3PLCRES_WrongHandle   Invalid dwDNP3Handle
```

Function DNP3_SetDbgLevel

```
FUNCTION DNP3_SetDbgLevel : DNP3RESULT
VAR_INPUT
  nDbgLevel :DNP3DBGLEV;
END_VAR
```

This function logs the DNP3 library's internal states. This facilitates the investigation of communication problems between the master and outstation.

Possible values are:

```
TYPE DNP3DBGLEV :
(
DNP3DBGLEV_None:=0             No recording
DNP3DBGLEV_Error := 1,         Recording errors
DNP3DBGLEV_Warning := 2,       Recording warnings
DNP3DBGLEV_Info := 3,          Recording additional informa-
                               tion
DNP3DBGLEV_Trace := 4,         Recording function invoca-
                               tions and parameters
DNP3DBGLEV_Max := 5,           Recording of all debug outputs
DNP3DBGLEV_FORCE_DWORD:=42949  (* Internal *)
67295
):= DNP3DBGLEV_None;
END_TYPE
```

The log file is stored temporarily in the controller under \temp\dnp3plc.log and must be transferred to a host via FTP before switching off the controller. The file no longer exists after the controller is switched back on.

Programming

Programming is implemented in the following steps:

- Server creation using statement of sizes for the data fields - DNP3_Create().
- Connection to the XIOC-TC1 module - DNP3_OpenCOM()
- Cyclic call of the function to
  - Read the data (DNP3Get...)
  - Write the data (DNP3SET...)
  - DNP3_Execute() function call to execute the DNP3 state machine.
- Closing the communication connection (DNP3_CloseComm()). This occurs conveniently in the PLC program's stop event.
- Server resource destruction (DNP3_Destroy())

→ All serial communication connections are automatically destroyed independently of this when the PLC transitions to halt.

FLAGs definition in DNP3

Binary data types flag definition

Bit	Flag meaning
0	Online (0 inactive, 1 active)
1	Restart (0, normal, 1 variable in initial status)
2	Comm_Lost (0, normal, 1 Value represents last valid data)
3	Remote_Forced (0, normal, 1 Value forced by external device)
4	Local_Forced 0, normal, 1 forced by local device e.g. HMI)
5	Chatter_Filter
6	Reserved (always 0)
7	State : 0.1 representing the state of physical or logical input

Flag definition for non-binary data types

Bit	Flag meaning
0	Online (0 inactive, 1 active)
1	Restart (0, normal, 1 variable in initial status)
2	Comm_Lost (0, normal, 1 Value represents last valid data)
3	Remote_Forced (0, normal, 1 Value forced by external device)
4	Local_Forced 0, normal, 1 forced by local device e.g. HMI)
5	Chatter_Filter
6	Reserved (always 0)
7	0

Function code according to DNP3 level 2

DNP OBJECT GROUP & VARIATION			REQUEST (Master may issue and Outstation must parse)		RESPONSE (Master must parse and Outstation may issue)	
Grp	**Var**	**Description**	**Function Codes (dec)**	**Qualifier Codes (hex)**	**Function Codes (dec)**	**Qualifier Codes (hex)**
1	0	Binary Input – Any Variation	1 (read)	06 (no range,or all)		
2	0	Binary Input Event – Any Variation	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
2	1	Binary Input Event – Without time	1 (read)	06 (no range, or all) 07, 08 (limited qty)	129 (response) 130 (unsol. Resp)	17, 28 (index)
2	2	Binary Input Event – With absolute time	1 (read)	06 (no range, or all) 07, 08 (limited qty)	129 (response) 130 (unsol. Resp)	17, 28 (index)
2	3	Binary Input Event – With relative time	1 (read)	06 (no range, or all) 07, 08 (limited qty)	129 (response) 130 (unsol. Resp)	17, 28 (index)
10	0	Binary Output – Any Variation	1 (read)	06 (no range,or all)		
12	1	Binary Command – Control relay output block (CROB)	3 (select) 4 (operate) 5 (direct op) 6 (dir. op, no ack)	17, 28 (index)	129 (response)	echo of request
20	0	Counter – Any Variation	1 (read) 7 (freeze) 8 (freeze noack) 9 (freeze clear) 10 (frz. cl. noack)	06 (no range,or all)		
22	0	Counter Event – Any Variation	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
30	0	Analog Input – Any Variation	1 (read)	06 (no range,or all)		
32	0	Analog Input Event – Any Variation	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
40	0	Analog Output Status – Any Variation	1 (read)	06 (no range,or all)		
41	2	Analog Output – 16-bit	3 (select) 4 (operate) 5 (direct op) 6 (dir. op, no ack)	17, 28 (index)	129 (response)	echo of request
50	1	Time and Date – Absolute time	2 (write)	07 (limited qty = 1)		
60	1	Class Objects – Class 0 data	1 (read)	06 (no range,or all)		
60	2	Class Objects – Class 1 data	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
60	3	Class Objects – Class 2 data	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
60	4	Class Objects – Class 3 data	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
80	1	Internal Indications – Packed format	2 (write)	00 (start-stop) index=7		
No Object (function code only)			13 (cold restart)			
No Object (function code only)			23 (delay meas.)			

7 Suconet K module (master) XIOC-NET-SK-M

Features

The module is used in conjunction with the XC100 or XC200 CPU. It has the function of the master on the Suconet K line and can control up to 16 slaves. Suconet K and Suconet K1 slaves are possible.

On an XC100 a maximum of two modules (COM interfaces) and on a XC200 a maximum of four modules (COM interfaces) can be operated. As the modules XIOC-SER and XIOC-NET-SK-M are addressed via the COM interfaces, the details of the number of modules (COM interfaces) in the PLC refers to both modules.

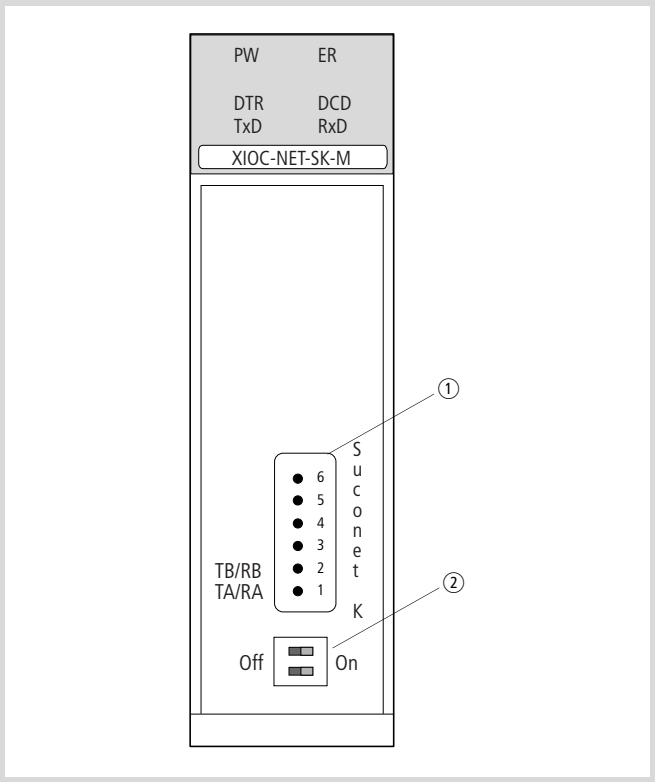
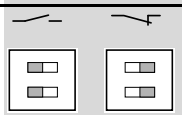


Figure 70: Suconet K interface RS485

①	RS485 (COMBICON)
6	–
5	–
4	–
3	–
2	TB/RB
1	TA/RA

②	Switches for bus termination resistors
	

The RS485 interface is galvanically isolated from the bus.

LED display

LED display	LED function	Module
PW (Power)	ON	Switched on
ER (Error)	On/Off	Application specific
DTR	ON	Ready for operation
DCD	ON	All stations connected
TxD	ON	Data is being sent
RxD	ON	Data is being received

Design of the Suconet K (RS485) interface

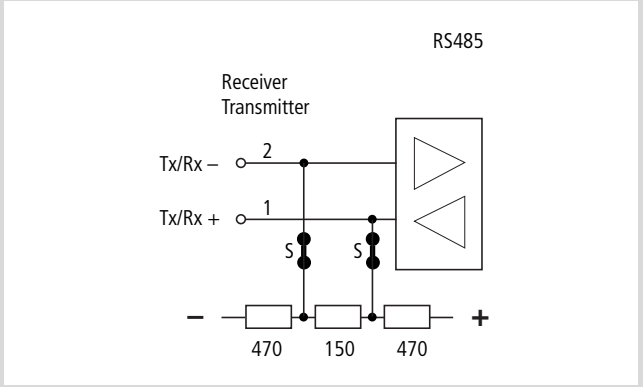


Figure 71: Suconet K interface / RS485 interface

S = switch for bus termination resistor

Select the module in the configurator of the easySoft-CoDeSys

- ▶ Open the PLC Configurator
- ▶ Click with the right mouse button on the required slot.
- ▶ Select the "Replace element" command.
- ▶ Select the module with a double-click in a new window.

→ The assignment between the slot of the module and the COM… programming language in the configurator: Activate the "Other Parameters" tab and select COM2, 3, 4 or 5 from the "Serial interface" list field → figure 73.

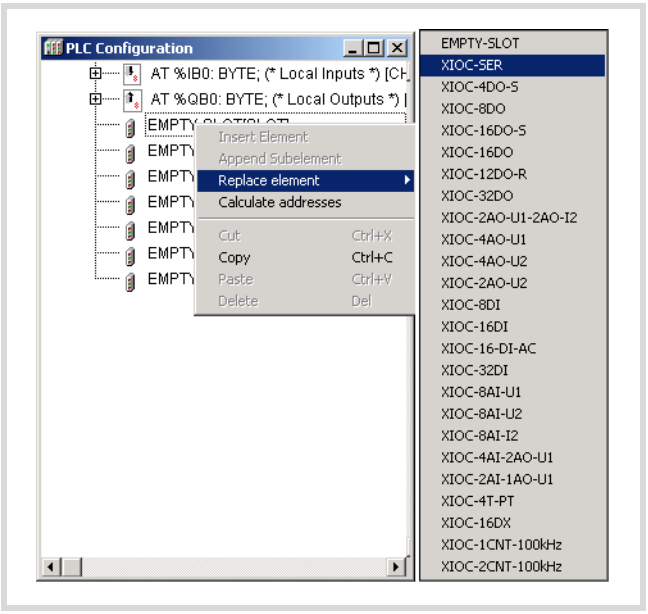


Figure 72: Integrate the module, here: XIOC-SER

Configuration of the interface

After selection of the module the baud rate and the serial interface COM2, 3, 4 or 5 can be set in the "Other Parameters" tab.

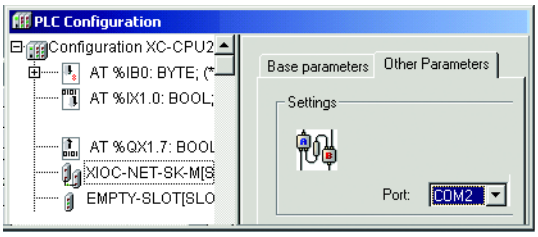


Figure 73: Parameters for Suconet K master

Setting the bus termination resistors

Set the bus termination resistors. If the module is physically the first or last module on the end of a line, set both of the S switches (→ figure 71) to the ON setting (default setting). Both of the switches must be set to "OFF" at all other positions on the line. Both switches must be in the same setting position to guarantee perfect communication.

Access to the receive and send data

Access from the user program to the data of the XIOC-NET-SK-M is implemented with the aid of the function blocks from the "SuconetK_Master.lib" library. The function blocks are described in the manual MN05010002Z-EN (previously AWB2786-1456GB) "Function blocks for easySoft-CoDeSys".

8 PROFIBUS-DP modules XIOC-NET-DP-M / XIOC-NET-DP-S

The PROFIBUS-DP modules XIOC-NET-DP-M (M = master) and XIOC-NET-DP-S (S = slave) forms the interface between the XC100-/XC200-CPU and the PROFIBUS-DP, which corresponds to the standard EN 50170 Vol. 2.

→ The master module is referred to in the following with the abbreviation DP-M module; the slave module is referred to as the DP-S module. If the description applies to both modules, they are simply referred to as the DP module.

A DP module can be inserted into one of the first three slots beside the CPU. This must also be taken into consideration with the configuration in the easySoft-CoDeSys PLC configuration.

Table 16: Maximum quantity and slots for DP modules dependant on the control type

XC	Slot	Max. quantity	Comment
XC100	1, 2 or 3	2[1)]	→ table 20
XC200	1, 2 and 3	3	No gaps between DP modules! → table 21

1) From operating system version 3.10 or higher, a DP-M and a DP-S module are possible.

The DP-M module organizes and operates the data transfer between the user program and the connected slaves. Up to 31 slaves can be addressed on one bus section. Several sections can be coupled together using repeaters, thus allowing up to 124 slaves to be connected.

The DP-S module can send and receive up to 244 bytes.

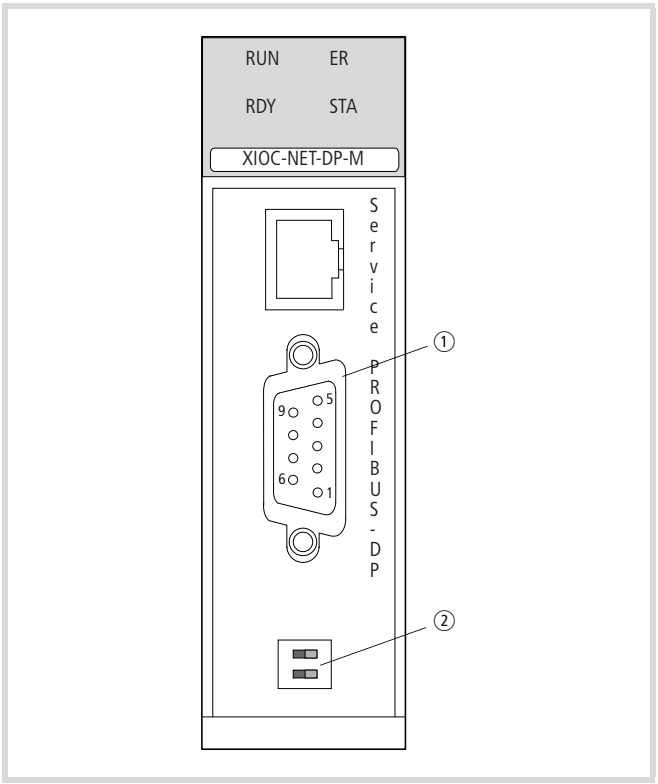


Figure 74: XIOC-NET-DP-M front view
(XIOC-NET-DP-S is identical except for the type designation)

① PROFIBUS-DP interface

② Bus termination resistors

Hardware and software prerequisites

The following prerequisites must be fulfilled for use of a DP module:

Table 17: Hardware and software prerequisites

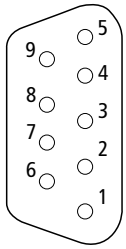
Hardware	Software DP-M	Software DP-S
XC100 ≧ V04	BTS ≧ V3.0	BTS ≧ V3.10
XC200 ≧ V04	BTS ≧ V1.02.00	BTS ≧ V1.03.02

BTS = operating system

Features

PROFIBUS-DP interface

In order to connect the PROFIBUS-DP cable to the galvanically isolated RS485 interface, you will require the special PROFIBUS-DP connector ZB4-209-DS2. It features the required wiring for malfunction free operation up to 12 Mbit/s.

PROFIBUS-DP	Pin	Meaning
	3	RxD/TxD-P
	4	CNTR-P
	5	DGND
	6	VP (+5 V DC)
	8	RxD/TxD-N

Switches for bus termination resistors

Termination resistors must be present on both ends of the cable. The DP module features switch-in bus termination resistors and can be placed at the end of a line.

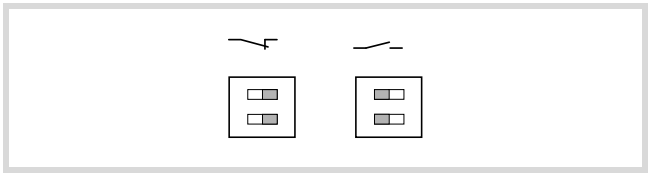


Figure 75: Bus termination resistor on the DP module (left switched on, right switched off)

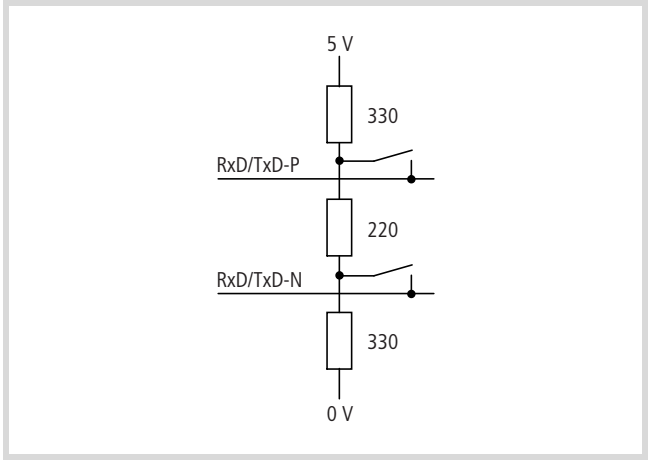


Figure 76: Bus termination resistors on the DP module

→ On modules which do not feature bus termination resistors the ZB4-209-DS2 PROFIBUS-DP connector can be used. It features a sliding switch which can be used to switch the resistors in or out.

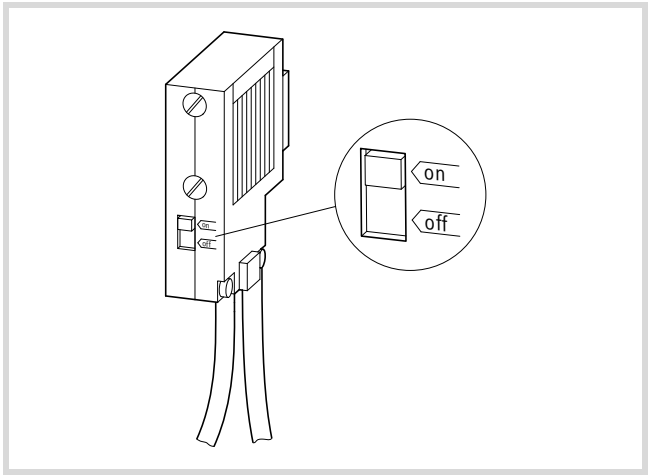


Figure 77: Bus termination resistors on PROFIBUS-DP connector

Status and diagnostics display (LEDs)

The four LEDs on the DP modules provide information concerning their status. They can occur in the following combinations:

LED-combination				Master status
RUN	○	○	ER	Communication o.k.
RDY	○	○	STA	
RUN	○	○	ER	Hardware error
RDY	⊗	○	STA	
RUN	◐	○	ER	All slaves are missing or there is no bus connector
RDY	○	○	STA	
RUN	○	○	ER	At least one slave is missing
RDY	○	○	STA	

LED-combination				Slave Status
RUN	○	○	ER	Communication o.k.
RDY	○	○	STA	
RUN	◐	○	ER	Connection to master interrupted or wrong address
RDY	○	○	STA	
RUN	⊗	○	ER	Not configured
RDY	○	○	STA	

○ ON ◐ cyclic flash ⊗ irregular flash ○ OFF

DP module operation

Download behavior

In a configuration with one or more DP modules the CPU will require a few seconds for the warm start after a project download. During this time the easySoft-CoDeSys user interface will not indicate any parameter changes or allow any data input. A "?" will appear in the configuration behind the inputs.

Behavior after switch on of the supply voltage

An error message appears when the supply voltage is applied and the CPU does not contain a user program. The following LEDs of the DP module are displayed:
ER, RDY and STA LEDs light up and the RUN LED flashes. As soon as a program is loaded, the "Error" message will disappear and the bus communication is active. As the CPU is in the STOP state, the RUN/STOP LED will flash on the CPU. A transition from STOP → RUN means the data is transferred via the bus.
The LEDs now have these states: RUN, RDY and STA LED light up and the ER LED is off.

Behavior after RUN → STOP transition

- With configuration of the XC200 with DP-M module
When the CPU switches from RUN to STOP, the master sets the content of all data to be sent to "0". The bus communication remains active. However, no application data is transferred.

In slaves without a user program, such as e.g. in an XI/ON-I/O unit, the outputs are set to "0" as a result. The slaves with a user program receive the "0" information in the receive data. A reaction to the "0" data must be programmed by the user.

- With configuration: XC200 with DP S module
After the RUN → STOP transition, the slave sets the data content which is sent to the master to "0". A reaction in the master to the "0" data must be programmed by you. The communication with the master is retained. The slave receives the current data from the master as was the case beforehand.

Behavior after interruption of the DP line

→ section "Configuration XIOC-NET-DP-S/M", "Auto Clear Mode" function

- With configuration of the XC200 with DP-M module
The master detects when the connection is interrupted to some slaves. In this case it sets the received data which the decoupled slaves send to "0".

- With configuration of the XC200 with DP-S module
  – Prerequisite: Watchdog active
    If the slave is decoupled, the slave sets the data sent by the master to "0" after the watchdog time has timed out. The data to the master continues to be updated by the slave.
  – Prerequisite: Watchdog not active
    After the slave is decoupled, the data last received from the master remains.

Process analysis

The following browser commands are available for tracing the causes of malfunctions.

geteventlist	Event list
geterrorlist	Error list
plcload	Display of the CPU loading in %. Should be under 70 %.

Configuration XIOC-NET-DP-S/M

The basic configuration is described in the manual for programming software (MN05010003Z-EN; previously AWB2700-1437GB).

In the master's configuration, you can change the "Auto Clear Mode" function in the DP Parameter tab:

- Not active (default): If a slave is disconnected from the bus, the master continues to communicate with the other slaves.
- Active: If a slave is disconnected from the bus, the master sets the outputs of all slaves on the bus to the safe state and stops all communication. To restart communication, switch the CPU power of and on again.

The "Autostart" function on the DP Parameter tab has no effect. The configuration of the XIOC-NET-DP-M can be seen in the example on Page 91.

A few peculiarities must be observed for configuration of the XIOC-NET-DP-S. The data to be transferred is packed into data blocks, which you can select in the "Inputs/Outputs" tab. There for example, you will find blocks available such as "2 Byte input con (0x91)" for inputs (data receive) as well as "2 Byte output con (0x91)" for the outputs (data send). The designation "con" stands for consistent. This means that the data, such a two bytes are consistent. This ensures that the master will process the two bytes simultaneously.

The same data blocks must be configured in the same sequence for the master PLC as well as for the slave PLC. In the configuration of the slave PLC the data direction is defined by the suffix "IEC-Input" (data receive) or "IEC-Output" (data send) (→ figure 89). The quantity of transferred data in one direction is limited to:

- Data blocks: max. 24
- Byte: max. 244

In the program, the send and receive data are accessed with the directly represented variables in the configurator.

Data exchange

PROFIBUS-DP module (master) ↔ slaves

The PROFIBUS-DP master (XIOC-NET-DP-M) supports two protocol types:

- Cyclic data exchange (DP-V0 services)
  The data exchange between the master and slaves is implemented cyclically with the PROFIBUS-DP bus. As a result the master copies the data in the input/output image of the CPU. The user program accesses this data.
- Asynchronous data exchange (DP-V1 services)
  The asynchronous data exchange serves acyclic reading and writing of data; e. g. for parametric programming of a drive. Function blocks are used for this task (see manual MN05010002Z-EN; previously AWB278-1456GB: Acyclic data access modules for PROFIBUS-DP).

PROFIBUS-DP master ↔ DP-S module

The DP master implements a cyclic data exchange (DP-V0 services) with the DP-S module. The configuration, parametric programming and programming of the PLCs is explained in section "Example: Data transfer XC200 (master) n XC100 (slave)" on Page 81.

XC100/XC200 ↔ DP-M module

The received and transmitted data of the slave are collected in the memory of the PROFIBUS-DP module (XIOC-NET-DP-M) and exchanged with the input/output image of the control. The timing of the exchange depends on the control type and the operating mode.

Table 18: Operating modes of the XC100/XC200

	Operating mode	
XC100	Without task management	Cyclic
XC200	With task management	periodic (monotasking)
		periodic (multitasking)

XC100: cyclic data exchange

On the XC100 the data exchange between the CPU and the DP-M module is determined by the program cycle.

Before the program start commences, the slave data is copied from the DP-M module into the input image of the CPU. Then the user program and the PROFIBUS-DP cycle (data exchange DP master ↔ slave) start simultaneously. At the end of the program cycle the data of the output image is copied into the DP-M module.
The bus cycle time should be less than the program cycle time. If it is longer (→ figure 78), no data exchange occurs at the end of the program cycle; the bus cycle continues. This means that the next programming cycle will be performed with the "old" data from the previous bus cycle.

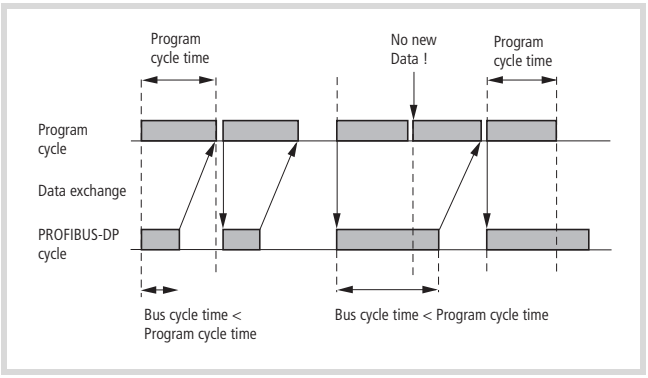


Figure 78: Data exchange between XC100 and DP-M module

XC200: Periodic data exchange (monotasking)
The XC200 always performs the user program periodically.

Without task management the default program PLC_PRG is processed with a cycle time (task interval) of 10 ms. This corresponds to a program which is managed by a single task and which is accessed with a task interval of 10 ms.

The data exchange between the CPU and the DP-M module is determined by the task interval. At the end of the task interval, the data exchange between the input/output image of the CPU and the DP module occurs.

The program start is initiated with the start of the next task interval and the DP-BUS cycle (data exchange DP-Master ↔ Slaves).

The task interval must be longer than the bus cycle time in order to guarantee a refresh of the inputs/outputs in every program cycle. If the task interval is less than the bus cycle time (→ fig. 79), data exchange will not take place at the start of the following task. The bus cycle continues and a refresh of the inputs/outputs occurs in the next cycle.

In order to derive the time required for the task interval, determine the bus cycle time in dependance on the baud rate.

Select the time for the task interval to be 5 % longer than the bus cycle time.

In general, the time for the task interval is in a range from 2 ms to 500 ms.

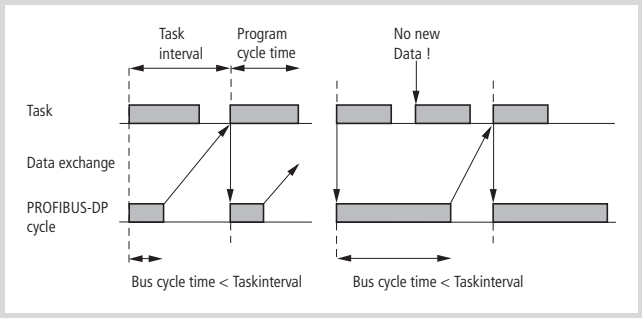


Figure 79: Data exchange with periodic operation

Determination of the bus cycle time:
In order to determine the bus cycle time you must access the Target Rotation Time (TTR) of the PROFIBUS-DP. It is a little longer than the bus cycle time.

The TTR can be taken from the bus parameters of the easySoft-CoDeSys configurator time

It is defined in "tBit" = "Bit times":

- ▶ Click on the XIOC-NET-DP-M folder in the PLC configuration.
- ▶ Open the "Bus Parameters" tab and set the baud rate.

The target rotation time is displayed in accordance with the baud rate, e. g. at a baud rate of 12 Mbit/s = 6647 tBit.

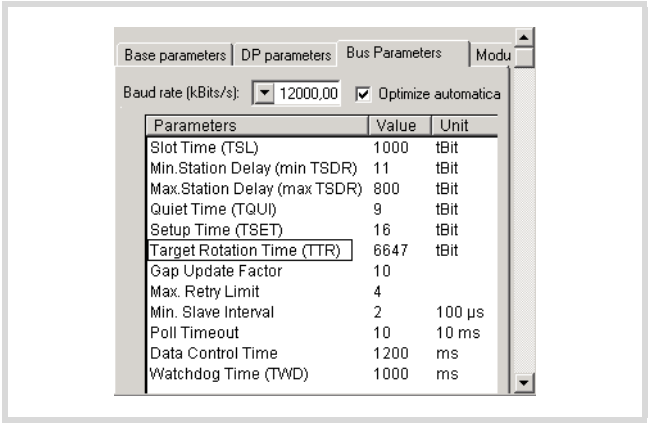


Figure 80: Setting the bus parameters

In order to ascertain the TTR in ms, determine the bit time [ns] for an individual bit using the following formula:

$$\text{Bit time [ns]} = \frac{1000000000}{\text{Baud rate [Bit/s]}}$$

Multiply the bit time with the TTR [tBit] which is defined in the configurator (→ fig. 80), you will receive a target rotation time in ms.

Example for a configuration comprised of a PROFIBUS-DP line with two stations:
The bus should be operated with a baud rate of 12000000 Bit/s.
How long is the TTR?

$$\frac{1000000000}{12000000} = 83.33 \text{ ns (time for one bit)}$$

$$83 \times 6647 \text{ (tBit config.)} = 0.55 \text{ ms (TTR)}$$

Add 5 % and you receive the time for the task interval = approx. 0.60 ms. In this case however, 2 ms should be entered as the smallest task interval is 2 ms!

If you select this configuration with two stations having different baud rates, the following TTR results:

Table 19: Target Rotation Time, dependent on the baud rate

Baud rate	1 tBit [ns]	Config. [tBit]	TTR [ms]
12 Mbit/s	83	6647	0.5539
6 MBit/s	166	5143	0.8572
3 MBit/s	333	4449	1.483
1.5 MBit/s	666	4449	2.966
500 Kbit/s	2000	3416	6.832
187.5 Kbit/s	5333	2994	15.968
93.75 Kbit/s	10666	2994	31.936
19.2 Kbit/s	52038	2994	155.9375
9.6 Kbit/s	104167	2994	311.875

A change of the station count or the transmitted data would result from another TTR!

Task control in online operation

In online mode the status of a task is defined in the configuration tree. The timing of a task can be monitored with the aid of a graphic representation. A prerequisite for this function is that the "SysTaskInfo.lib" and "SysLibTime.lib" library functions are appended into the easySoft-CoDeSys (→ MN05010003Z-EN, chapter "Resources", "Task configuration").
When "SysTaskInfo.lib" is appended, the "SysLibTime.lib" is automatically appended.

Response time on PROFIBUS-DP

Figure 81 indicates the course of an input on a PROFIBUS-DP slave from processing until a slave output is set.

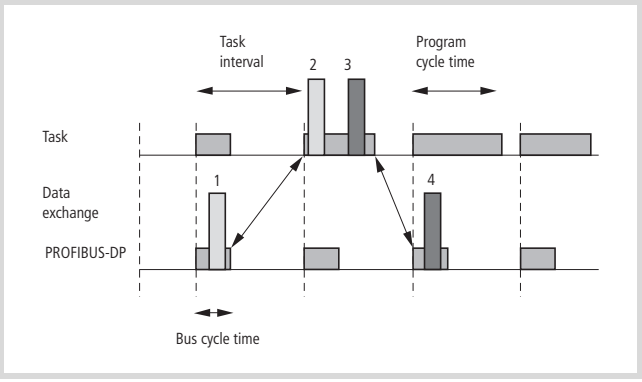


Figure 81: Response time on PROFIBUS-DP

Procedure:
Prerequisite: the bus run time is less than the task interval.

① The voltage is applied to a slave input. The "1" signal is detected during the bus cycle.

②, ③ The input data of the slave is copied into the input image of the CPU at the beginning of the following task interval. The input is processed ② and the result is presented to the output ③. The outputs are copied to the output image at the end of the task interval.

④ The output of the slave is set in the following bus cycle.

XC200: multitasking mode

The multitasking mode is described in the XC200 manual (MN05003001Z-EN; previously AWB2724-1491GB).
Here are a few notes for use of the DP module.

The data exchange between the CPU and the DP-M module is determined by the task interval. Verify that the following conditions have been fulfilled when you have assigned each configured DP-M module with a TASK:

- The tasks must have differing priorities!
- The inputs and outputs of the slave which have been coupled to a line have also been referenced!
- The set the time for a task interval is in a range from 2 ms to 500 ms.

XC100/XC200
If differing tasks operate on the inputs/outputs of a DP-line, the first configured task in which a slave output is used initiates the PROFIBUS-DP cycle.

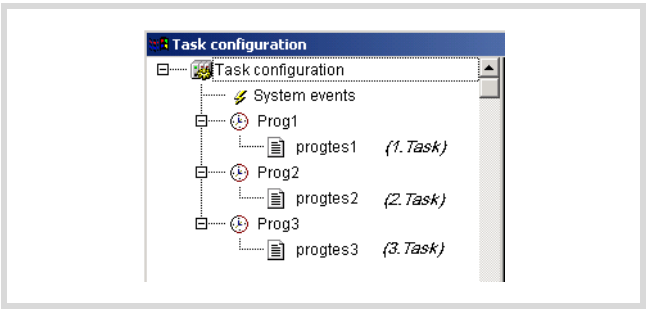


Figure 82: Configuration with three tasks

If for example, an output is not used in Task 1 but is used in Task 2 and 3, the PROFIBUS cycle will be started at the commencement of the second Task "Prog2". The data exchange occurs at the end of the task.

XC100: status indication of the PROFIBUS-DP slave

Analog and digital input and output states of the PROFIBUS-DP slave, which are connected via the DP-M module with the XC100 can be made visible in the status indication.

Prerequisites:

- A simple program (e.g.: a:=a) is loaded and the CPU is in STOP or RUN.
- The inputs/outputs are configured.
- Voltage/current is applied to the inputs.

The outputs of the PROFIBUS-DP slaves can be set in the configuration for test purposes if the following prerequisites are fulfilled:

- A simple program (e.g.: a:=a) is loaded and the CPU is in RUN.
- The inputs/outputs are configured.
- The outputs of the PLC configuration are clicked and a value is defined.

Neither a declaration or a program addressing the inputs/outputs is required.

Example: Data transfer XC200 (master) ↔ XC100 (slave)

The example shows the configuration, parametric programming and programming of the both controls. Every PLC sends 2 bytes and receives 1 byte.

The design of the controls can be seen in Figure 83.

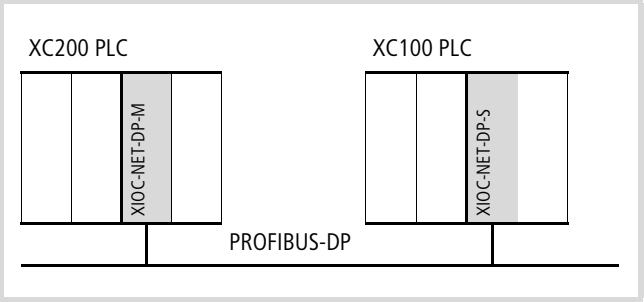


Figure 83: Design of the PLCs

▶ First of all configure the XC200 according to Figure 84.

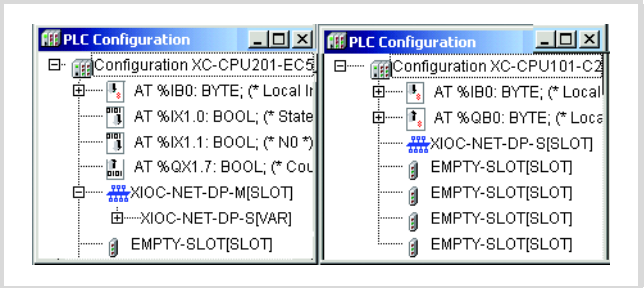


Figure 84: XC200 configuration

Define the parameters for the master in the XC200:

▶ Click on the "XIOC-NET-DP-M" and select the following settings:
  – in the DP Parameter tab: highest station address = 2
  – in the Bus Parameter tab: e.g. 1500.00

▶ Click on the "XIOC-NET-DP-S" folder.
▶ Select in the "Inputs/Outputs" tab (→ figure 85) the inputs/outputs for the slave, so that it corresponds to Figure 86.

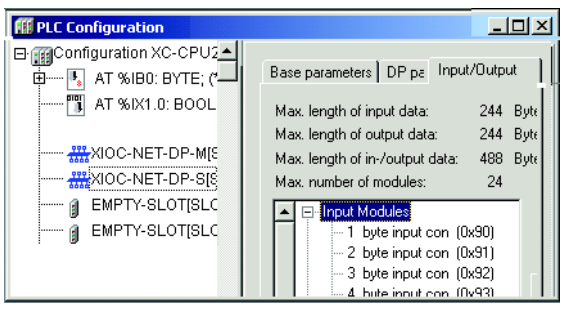


Figure 85: Selection of the inputs/outputs

→ Some input/output designations have the "con" suffix. This means that the data, such a two bytes are consistent. This ensures that the master will process the two bytes simultaneously.

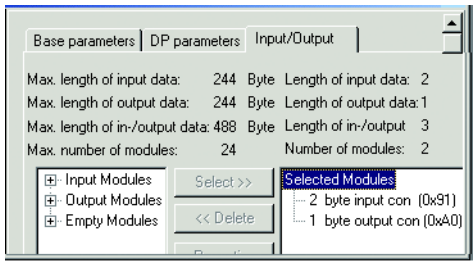


Figure 86: Parametric programming of the inputs/outputs

The direct I/O addresses are then displayed under XIOC-NET-DP-S in the XC200 control configuration → figure 89.

▶ Create the program in accordance with Figure 87.

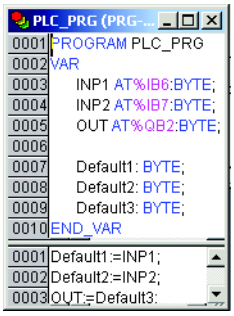


Figure 87: User program for XC200

Proceed in the same manner with the XC100 PLC.

Configure the XC100 according to Figure 88:

▶ Enter the station address "2" in the "DP Parameter" tab.
▶ Select the inputs/outputs for the slave in the "Inputs/Outputs" tab.

The selection of the modules including their identity (e.g. 0x91) and their sequence must correspond with the selection in the DP-M/DP-S module → figure 86.

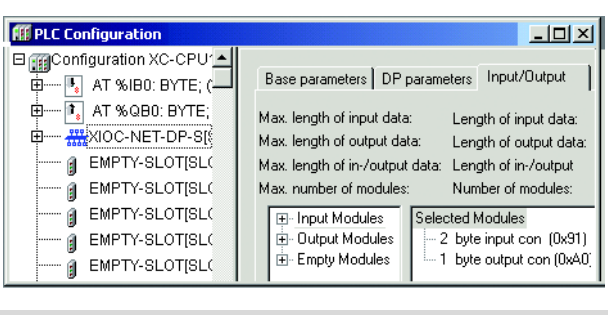


Figure 88: XC100 I/O configuration

Immediately afterwards the PLC configuration under XIOC-NET-DP-S displays the direct I/O addresses. If you compare the input/output details e.g. "2 Byte Input con" of the XC100 with the XC200, you will see that they are identical. The additional designation "IEC-Output" or "IEC-Input" provides information about the actual data direction. The details of the direct address such as IB/QB also provide the actual data direction. If for example a date in the XC100 is transferred from the QB2 (output byte) to the IB6 (input byte) of the XC200.

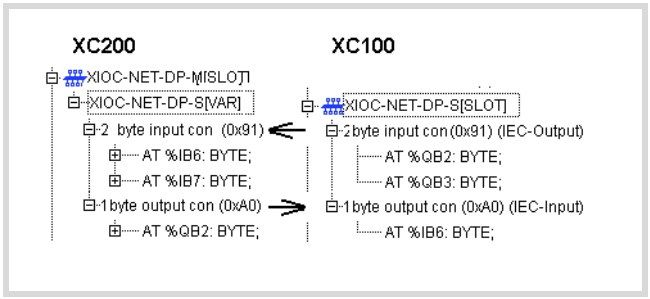


Figure 89: Display of the direct addresses and their data direction

▶ Create the program in accordance with Figure 90.

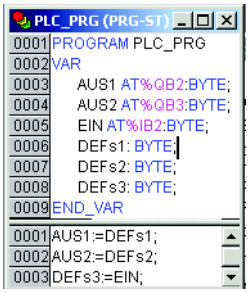


Figure 90: User program for XC100

Diagnostics of the PROFIBUS-DP slaves

The diagnostics in the PROFIBUS-DP is organized so that the master collects the diagnostics data which has been provided by the slaves.

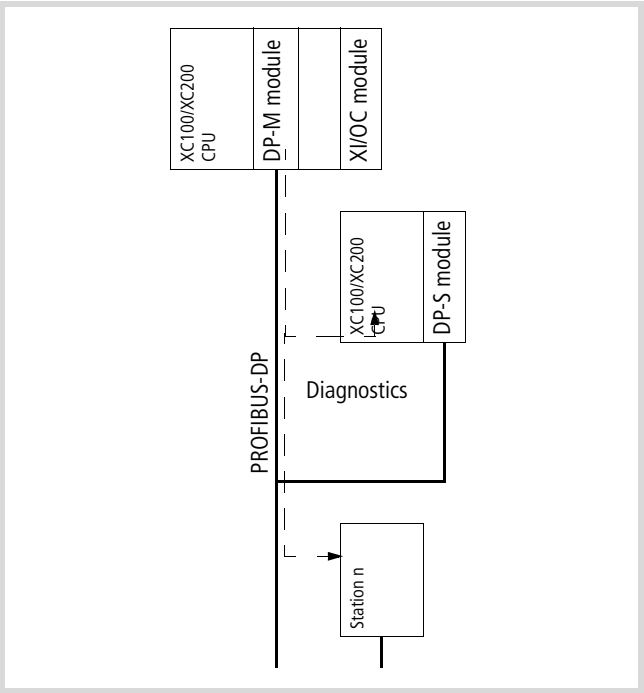


Figure 91: Diagnostics on the PROFIBUS-DP line

The evaluation of the diagnostics data can be programmed with the aid of function blocks. This can happen in two different methods. Both methods can continue to be used.

Method for existing applications	Method for new applications
With the variables of the GETBUSSTATE type and the DIAGGETSTATE function block.	With the xDiag_SystemDiag and xDiag_ModuleDiag function blocks. Software prerequisite (OS version): XC100: 3.10 XC200: 1.03.02
Library: BusDiag.lib	Library: xSysDiagLib.lib
The method is explained later	The method is described in MN05010002Z-EN (previously AWB2768-1456), chapter "Diagnostics module: xSysDiagLib".

Regardless of this, a slave can become active with the aid of the "xDPS_SendDiag" function block, e.g. in order to inform the master of a RUN → STOP or STOP → RUN transition. In this case you must program the module with the START/STOP interrupt function. The information to be sent can be placed in an array which accesses the function block → section "Diagnostics in the slave control" on Page 88.

Implement diagnostics

The "BusDiag.lib" library file provides a GETBUSSTATE structure and the DIAGGETSTATE function block for implementation of the diagnostics. In section "Program example for diagnostics in the master control" from Page 91 you will see how you can link the structure and the function block in the program with one another.

```
TYPE GETBUSSTATE;
   STRUCT
      BOLDENABLE:       BOOL;
      ENABLE:           BOOL;
      DRIVERNAME:       POINTER TO STRING;
      DEVICENUMBER:     INT;
      READY:            BYTE;
      STATE:            INT;
      EXTENDEDINFO:     ARRAY[0..129] OF BYTE;
   END_STRUCT
END_TYPE
```

The assignment between DP module and diagnostics function block is implemented with the aid of a device number, which depends additionally on the module slot → table 20 when the XC100 PLC or the → table 21 XC200 are used:

Table 20: Device number for XC100

XIOC-Slot	1	2	3
Module	DP-M	DP-S	X-module
Device No.	0	1	–
Module	DP-S	DP-M	X-module
Device No.	0	1	–
Module	DP-M/S	X-module	X-module
Device No.	0	–	–
Module	X-module	DP-M	DP-S
Device No.	–	0	1
Module	X-module	DP-S	DP-M
Device No.	–	0	1
Module	X-module	DP-M/S	X-module
Device No.	–	0	–
Module	X-module	X-module	DP-M/S
Device No.	–	–	0

Table 21: Device number for XC200

XI/OC slot	1	2	3	
Module	DP-M/S	DP-M/S	DP-M/S	
Device No.	0	1	2	
Module	DP-M/S	DP-M/S	X-module	
Device No.	0	1	–	
Module	X-module	DP-M/S	DP-M/S	
Device No.	–	0	1	
Module	DP-M/S	DP-M/S	DP-M/S	Configuration fault: Gaps are invalid![1)]
Device No.	0	–	2	
Module	X-module	X-module	DP-M	
Device No.	–	–	0	

X-module: no PROFIBUS-DP module

1) The configurator permits this design, but a fault is indicated during compilation.

Diagnostics data evaluation

You must create a variable of the GETBUSSTATE type (the procedure is described in section section "Coarse diagnostics with variable from GETBUSSTATE type") to evaluate the diagnostic data. With the EXTENDEDINFO array the variable provides each station with a (station) byte where the individual bits contain information concerning the status of the communication and the slave. The content of the byte is continually refreshed by the run time system (→ table 22 on Page 85).

Query bit 2 of this station byte for coarse diagnostics. If the slave sends a diagnostic alarm, the assigned station byte will set bit 2 to the "1" signal state. In order to reset the signal (Bit 2 → "0" signal) call up the DIAGGETSTATE function block.

Query the EXTENDEDINFO output array of the DIAGGETSTATE function block for detailed diagnostics.

→ The EXTENDEDINFO output array from the DIAGGETSTATE function block is not identical with the EXTENDEDINFO array of the variables of the GETBUSSTATE type!

Further information can be found at section "Detailed diagnostics with DIAGGETSTATE function block" on Page 85.

Monitoring data exchange

A station byte contains further information in the EXTENDEDINFO array GETBUSSTATE variable, e.g. the status of the data exchange between the master and the respective station. Query bit 1 for this purpose.

If data exchange functions bit 1 has the "1" signal state. A "0" signal indicates that the data exchange has been interrupted, e.g. by a cable break or device malfunction. In this case the slave cannot send diagnostics.

Coarse diagnostics with variable from GETBUSSTATE type

Create variables of the GETBUSSTATE type

A prerequisite for diagnostics is that the "BusDiag.LIB" file is integrated into the project. A directly addressable global variable of the GETBUSSTATE type must be created in order to access the diagnostic data. It is listed in the PLC Configuration under the "Diagnostic address" handle.

▶ Click on the "XIOC-NET-DP-M" folder in the PLC configuration.

The "Diagnostic address" is displayed on the "Base parameters" tab. The diagnostics address is called %MB4 for the XC100 and the first DP line of the XC200.

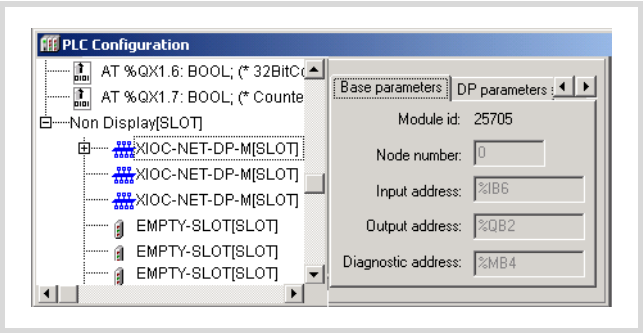


Figure 92: Diagnostic address

Declaration with XC100:

```
Var_Global
 DPSTAT AT%MB4 : GETBUSSTATE;   (* MB4 diagnostics address of
End_Var                          the DP-master *)
```

Declaration with XC200 with 3 DP lines:

```
Var_Global
  DPSTAT_1 AT%MB4 : GETBUSSTATE;    (* 1st master *)
  DPSTAT_2 AT%MBxx : GETBUSSTATE;   (* 2nd master *)
  DPSTAT_3 AT%MByz : GETBUSSTATE;   (* 3rd master *)
End_Var
```

Query variables from the GETBUSSTATE type:
The diagnostics data are written in an ARRAY OF BYTES with the EXTENDEDINFO structure names.

Evaluate the EXTENDEDINFO array:
In principle the array has the following structure:

Table 22: Station byte

Bit	7	6	5	4	3	2	1	0	Station address
Byte 0:						×	×	×	0
Byte 1						×	×	×	1
Byte 2						×	×	×	2
Byte 3						×	×	×	3
…									
Byte 125						×	×	×	125

Each byte contains diagnostics information of a station. It is continuously refreshed by the run time system. Bit 0, 1 and 2 contain the following diagnostics data. Bit 3 to bit 7 are without significance.

Table 23: Diagnostics information

Bit 0 = 1:	A configuration exists for the address.
Bit 1 = 1:	Data exchange ok Bit 1 already indicates a "1" signal when data exchange for coupling of the slave has been successful. This means: the connection is o.k. and data exchange occurs.
Bit 2 = 1:	New diagnostics data exist.

For diagnostics, monitor the station byte for fault signals commencing with address 2 up to max. address 125.
In the example it occurs with the query:

```
IF (xxx.EXTENDEDINFO[n] >=6) THEN
```

xxx = global variable of GETBUSSTATE type, e.g. DPSTAT
n = address of the station

Detailed diagnostics with DIAGGETSTATE function block

The DIAGGETSTATE function block must be accessed for each station/node (BUSMEMBERID).

```
FUNCTION_BLOCK DiagGetState
VAR_INPUT
   ENABLE:          BOOL;
   DRIVERNAME:      POINTER TO STRING ; (* XC100/XC200 = 0 *)
   DEVICENUMBER:    INT ; (*XC100: 0, 1/XC200: 0, 1, 2*)
   BUSMEMBERID:     DWORD ;
END_VAR

VAR_OUTPUT
   READY:           BOOL;
   STATE:           INT;
   EXTENDEDINFO:    ARRAY[0..99] OF BYTE ;
END_VAR
```

→ The EXTENDEDINFO output of the "DiagGetState" function block is independent of the EXTENDEDINFO output of the GETBUSSTATE structure.

The program example for diagnostics indicates a line with an XI/ON station and an EM4/LE4 input/output combination (→ from Page 92).

After the parameters have been applied to the DRIVERNAME, DEVICENUMBER and BUSMEMBERID function inputs, a "1" must be applied to the ENABLE input.

If the READY function input is a "1" and the STATE output is a "2" (compare with the defined constants "NDSTATE_DIAGINFO_AVAILABLE = 2), the EXTENDEDINFO output array can be queried.

Inputs/outputs of the DIAGGETSTATE function block

Inputs	
ENABLE	1 = activate 0 = deactivate
DRIVERNAME	= 0 **(always 0 with XC100/XC200)**
DEVICENUMBER	XC100 = 0, 1/XC200 = 0, 1, 2
BUSMEMBERID	Address of the slaves
Outputs	
READY	0 = module inactive 1 = module active
STATE	constants have been determined for the values –1, 0, 1, 2, 3: –1: NDSTATE_INVALID_INPUTPARAM 0: NDSTATE_NOTENABLED 1: NDSTATE_GETDIAG_INFO 2: NDSTATE_DIAGINFO_AVAILABLE 3: NDSTATE_DIAGINFO_ NOTAVAILABLE
EXTENDEDINFO	Further diagnostic data is present in the 100 byte.

- Data content of DIAGGETSTATE.EXTENDEDINFO
  The data content of DIAGGETSTATE.EXTENDEDINFO is subdivided into:
  – General diagnostics data (Byte 0 to 7)
  – Standard diagnostics data (Byte 8 to 13)
  – Device-specific diagnostics data (Byte 14 to 99)

The device-specific diagnostics data is described in the device documentation and in the respective GSD file.

The most important information has a grey background in the following table.

EXTENDEDINFO[0]	//with PROFIBUS-DP: slave address	
EXTENDEDINFO[1..4]	//no meaning	
EXTENDEDINFO[5]	//length byte of the device diagnostic	
EXTENDEDINFO[6&7]	//no meaning	
EXTENDEDINFO[8] (Standard byte 1)	//Status_1	
	Bit 0:	Device does not respond (no valid IO data)
	Bit 1:	Slave not ready
	Bit 2:	Divergent configuration
	Bit 3:	Further diagnostics exist
	Bit 4:	Unknown command
	Bit 5:	Invalid response
	Bit 6:	Incomplete parametric programming
	Bit 7:	Parametric programming from another master
EXTENDEDINFO[9] (Standard byte 2)	//Status_2	
	Bit 0:	Ready for new starting sequence
	Bit 1:	No parametric programming
	Bit 2:	„1“
	Bit 3:	Watchdog activated
	Bit 4:	FREEZE command active
	Bit 5:	SYNC command active
	Bit 6:	Reserved
	Bit 7:	Slave has not been engineered
EXTENDEDINFO[10] (Standard byte 3)	//no meaning	
EXTENDEDINFO[11] (Standard byte 4)	//for PROFIBUS-DP: master address	
EXTENDEDINFO[12&13] (Standard byte 5, 6)	//Own identity number	
EXTENDEDINFO[14]	//Length byte of the manufacturer-specific data	
EXTENDEDINFO[15..99]	//device-specific diagnostics.	

• Diagnostics capable XI/ON modules
If you perform diagnostics with the DIAGGETSTATE function block on a XI/ON station, the EXTENDEDINFO output displays the diagnostics data for the entire station in bytes 15 and 16. The data originate from the GSD file of the central XI/ON gateway.

Byte 17 to 99 contains the fault code for the modules with diagnostics capability. This occurs in the module sequence.
A byte will not exist for non-diagnostic capable modules.

EXTENDEDINFO[15]	//	Bit 0:	Module diagnostics present
		Bit 2:	Parametric programming incomplete
		Bit 3:	Divergent configuration
EXTENDEDINFO[16]	//	Bit 1:	–
		Bit 2:	Module bus fault
		Bit 3:	Master configuration fault
		Bit 4:	–
		Bit 5:	Station configuration fault
		Bit 6:	I/Oassistant force mode active
		Bit 7:	Module bus failure
EXTENDEDINFO [17…99]	//one or more bytes for each diagnostics capable module (→ following table; further information can be found in the "XI/ON PROFIBUS-DP" manual (AWB2700-1394G).		

The following excerpt from the "XI/ON Gateways for PROFIBUS-DP" (MN05002004Z-EN;
previously AWB2725-1529G) manual indicates the diagnostics bit of the XI/ON modules:

e.g. power supply module		
XN-BR-24VDC-D	Bit 0:	Module bus voltage warning
	Bit 2:	Field voltage missing
XN-PF-24VDC-D	Bit 2:	Field voltage missing
XN-PF-120/230VAC-D	Bit 2:	Field voltage missing
e.g. output modules		
XN-2DO-24VDC-0.5A-P XN-2DO-24VDC-2A-P XN-2DO-24VDC-0.5A-N XN-16DO-24VDC-0.5A-P	Bit 0: Bit 1:	Overcurrent channel 1 Overcurrent channel 2
e.g. analog module		
XN-1AI-I	Bit 0:	Measured value range fault
	Bit 1:	Wire breakage
XN-1AI-U	Bit 0:	Measured value range fault
XN-2AI-PT/NI-2/3	1st BYTE	
	Bit 0:	Measured value range fault (channel 1)
	Bit 1:	Wire breakage
	Bit 2:	Short-circuit
	2nd BYTE	
	Bit 0:	Measured value range fault (channel 2)
	Bit 1:	Wire breakage
	Bit 2:	Short-circuit
e.g. counter module		
XN-1CNT-24VDC (C)	Bit 0:	Short-circuit/wire breakage DO
	Bit 1:	Short-circuit 24 V DC encoder supply
	Bit 2:	Count range end false
	Bit 3:	Count range start false
	Bit 4:	Invert DI with L ret. fault
	Bit 5:	Main count direction false
	Bit 6:	Operating mode false
XN-1CNT-24VDC (M)	Bit 0:	Short-circuit/wire breakage DO
	Bit 1:	Short-circuit 24 V DC encoder supply
	Bit 2:	Encoder impulse false
	Bit 3:	Integration time false
	Bit 4:	Upper limit false
	Bit 5:	Lower limit false
	Bit 6:	Operating mode false
e.g. DOL starter module		
XS1-XBM	Bit 0:	Ident fault
	Bit 1:	PKZ short-circuit
	Bit 2:	PKZ overload
	Bit 4:	DIL1 defective
	Bit 5:	DIL2 defective

• Diagnostics byte of EM4/LE4 modules

→ Further information about the diagnostics is contained in the "EM4-204-DX1, expansion module for PROFIBUS-DP" module (AWB27-1315G).

The data content of DIAGGETSTATE.EXTENDEDINFO has the following meaning:

EXTENDEDINFO[0…13]	as previously described
EXTENDEDINFO[14]	Length byte
EXTENDEDINFO[15]	Group diagnostics byte for all modules
EXTENDEDINFO[16]	Diagnostics byte for EM4
EXTENDEDINFO[17…22]	Diagnostics byte for 1 … 6 LE

Diagnostics in the slave control

Generally the master (DP-M module) queries the slave (DP-S module) if a diagnostics fault exists. In this case the master accesses the standard diagnostics data from the slave. Evaluation of this data is described in section "Diagnostics data evaluation" on Page 84.

Furthermore, the slave can become active and send diagnostics data. Thus for example, the start/stop event can be evaluated and the master can be informed of application-specific data.

The slave activity is used to inform the master of the start/stop state as well as important user-specific data. Transfer of the data should not occur continuously as otherwise the load on the bus will be too high. The transfer is implemented with the Diagnostic module "xDPS_SendDiag" (see section below) in the slave program. You can determine the content of the user-specific data and can copy it from the area defined in the module.

If the bus connection is interrupted after the start of the function block, the send job is performed as soon as the connection is re-established.

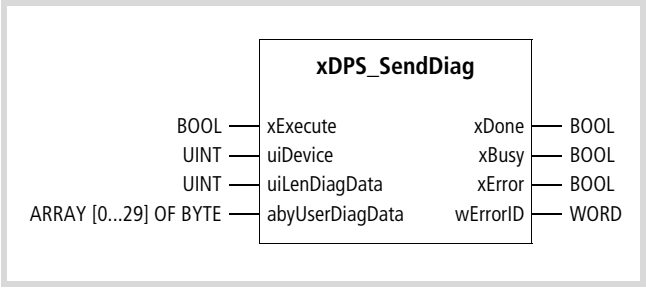
The assignment between the XIOC-NET-DP-S DP module and the diagnostics module is implemented with the aid of a device number, which is also dependent on the module slot → table 20 and Table 21.

Query master and connection status

If a query concerning the master state (RUN/STOP) or the connection state be necessary in the slave PLC, this function has to be programmed. More detailed information can be found here in the MN05010002Z-EN manual (previously AWB2786-1456GB) at "xDiag_SystemDiag" and "xDiag_ModuleDiag" function blocks.

Diagnostic module "xDPS_SendDiag"

This function block is located in the "xSysNetDPSDiag.lib" library.



Function block prototype

Meanings of the operands

xExecute	Start, Prerequisite: xBusy output = L signal xDone output = L signal The input is to be set to an L signal, after the xDone-output = H signal.
uiDevice	DP slave device number
uiLenDiagData	Length of the diagnostics data (Byte 0 to 30) The standard diagnostics data is sent with 0, → section "Data content of DIAGGET-STATE.EXTENDEDINFO The data content of DIAGGETSTATE.EXTENDEDINFO is subdivided into:" to Page 86.
abyUserDiagData	Diagnostics data of the user
xDone	H signal after the order has been processed If "xExecute" changes from a H to L signal, the "xDone" output has an L signal
xBusy	H signal, after a valid job is present
xError	The outputs should be scanned after the xDone output changes from an L signal to a H signal.
wErrorID	If the xExecute input is set to an L signal, the Error output is also set to the L signal.
	Error code 0: ok 1: incorrect device number 2: invalid length of the diagnostics data 3: no resources available 4: internal fault 5: error message of PROFIBUS-DP

Description
Access to the function block in the slave program has the effect than the master gets application-specific diagnostics data during the next access to the slave, and then exchanges the I/O data cyclically thereafter.

The CPU requires several cycles in order to process the function block!

As it can replace multiple master/slave modules, the device number must be entered on the "uiDevice" input. It represents the assignment between the function block and the module.

The following applies for the XC100: 0, 1 → table 20

The following applies for the XC200: 0, 1, 2 → table 21

Application example for sending diagnostics data (with the xDPS_SendDiag function block)

The program example has been created as a function block, which includes the xDPS_SendDiag module.

The transfer parameters are:

uiDevice:UINT;	Device number
uiLenDiagData:UINT;	Length of the diagnostics data to be sent
abyDiagData: ARRAY[0..29]OF BYTE;	Diagnostics data ByteArray

If processing of the function block is interrupted by a malfunction, the "DiagErrorWarning" variable is set.
It should be declared as a global variable.

```
FUNCTION_BLOCK DP_SendDiag_Slave

VAR_INPUT                                   (* Transfer parameter *)
      uiDevice:UINT;                        (* Device number*)
      uiLenDiagData:UINT;                   (* Length of the diagnostics data to be sent *)
      abyDiagData: ARRAY[0..29]OF BYTE;     (* Diagnostics data ByteArray *)
END_VAR
VAR_OUTPUT
      xError:BOOL;
      wErrorId:WORD;
END_VAR
VAR
      DpSndDiag : xDPS_SendDiag;
      Timer:TON;
      (*Test_Counter1: UINT;*)
      (*Test_Counter2: UINT;*)

END_VAR



Program:

IF NOT DpSndDiag.xBusy AND NOT DpSndDiag.xExecute THEN
      DpSndDiag.uiDevice:=uiDevice;
      DpSndDiag.uiLenDiagData:=uiLenDiagData;
      DpSndDiag.abyUserDiagData:=abyDiagData;
      DpSndDiag.xExecute:=TRUE;
END_IF
```

```
WHILE (NOT DpSndDiag.xDone )
DO
        Timer.PT:=T#2s;                         (* Avoid an endless loop if DpSndDiag.xDone has not been ended*)
        Timer.IN:=TRUE;
        Timer();

        IF Timer.Q =TRUE THEN
                DiagErrorWarning:=TRUE;

                EXIT;
        END_IF
        (*Test_Counter1:=Test_Counter1+1;*)
    DpSndDiag();
    xError:=DpSndDiag.xError;
    wErrorId:=DpSndDiag.wErrorId;
END_WHILE

DpSndDiag.xExecute:=FALSE;
DpSndDiag();

Timer.IN:=FALSE;
Timer();
(*Test_Counter2:=Test_Counter2+1;*)
```

Program example for diagnostics in the master control

The diagnostics will be explained using a program example which is based on the device design in figure 69. The diagnostics programs are also valid for other devices. In this example the XC100 assumes the control function.

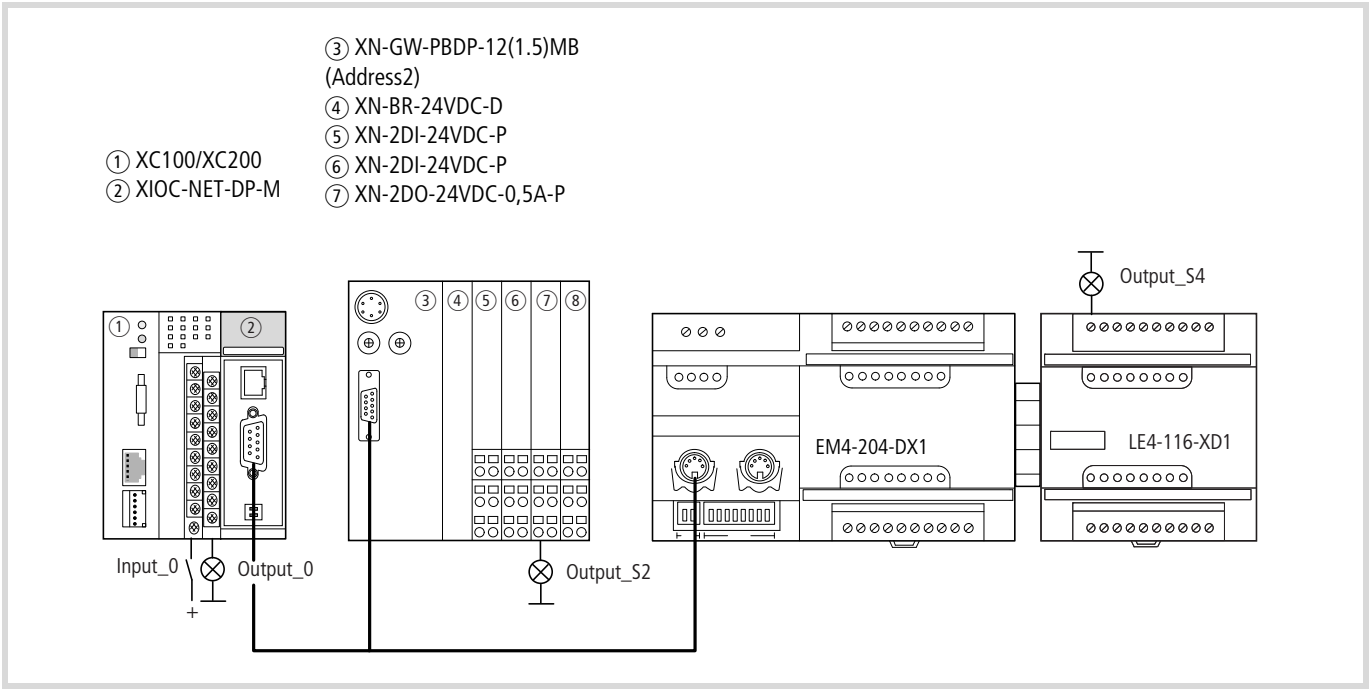


Figure 93: Configuration of the example project

Create configuration

The device configuration is implemented with the PLC Configuration of easySoft-CoDeSys (→ MN05010003Z-EN, programming software, chapter "PLC Configuration").

Create the configuration according to the following example:

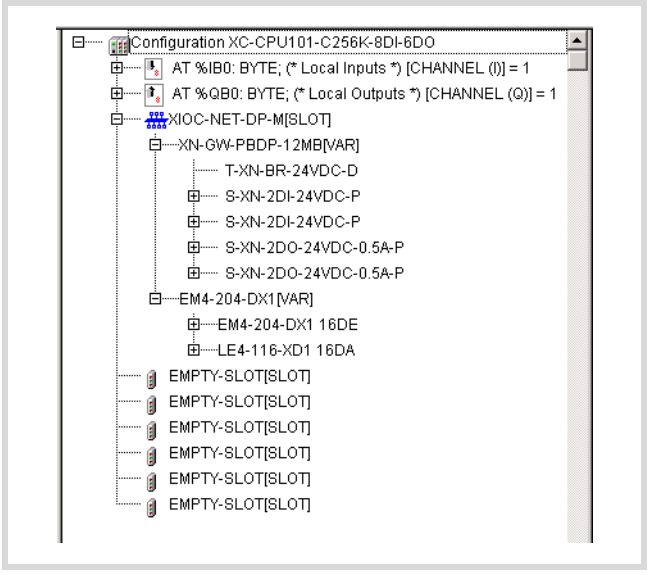


Figure 94: Device configuration in the easySoft-CoDeSys

Configuration of the XIOC-NET-DP-M

▶ Call up the "PLC Configuration" in the "Resources" tab.

The XC100 is displayed with inputs and outputs as well as several "Empty Slot" folders.

▶ Click with the right mouse button on one of the three EMPTY SLOT [Slot] folders under the QB0 output byte.

▶ Place the mouse pointer on the "replace element" and select the XIOC-NET-DP-M module from the list. It is added to the configuration and four tabs appear on the right hand window:

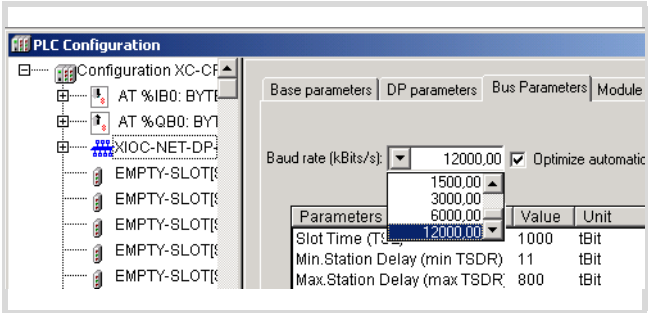


Figure 95: Configuration of the XIOC-NET-DP-M

▶ Set the baud rate in the "Bus parameters" tab and verify if the "Optimize automatic" function is active.

Configure XION station

▶ Click with the right mouse button on the XIOC-NET-DP-M[Slot] folder.
▶ Select "Append subelement" and click on a "Bus Refreshing module", e.g. XN-GW-PBDP-xxMB. It is added to the PLC configuration.
▶ Set the parameters in the various tabs for the XN-GW-PBDP:

- Enter the station address in the "DP Parameter".
- Modify the settings as follows in the "User parameters" tab (Set the cursor on the "Value" column and double click):
  – Diagnostics from modules: activate
  – Gateway diagnostics: device related diagnostics

- On the "Inputs/Outputs" tab:

Determine the I/O types of which the XION station is comprised:

▶ Select the Bus Refreshing module first in all cases:
  – Mark the T-XN-BR-24VDC-D on the left window under "empty modules".
  – Press the "Select" button in order to transfer the module to the right hand window.

▶ Proceed in the same manner with other modules. After selection of all modules, the right hand window should include all the modules:

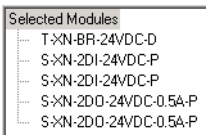


Figure 96: Configuration of the XION station

Configuration of the EM4/LE4 module

▶ Set the cursor on the XIOC-NET-DP-M[SLOT] folder and confirm with the right hand mouse button.
▶ Set the cursor on the "Append subelement" point and select the EM4-204-DX1 module from the list. The device is added to the configuration.

▶ Set the parameters in the tabs:

- Enter the station address in the "DP Parameter".
- Select your modules in the "Input/Output" tab:
  – Mark the EM4-204-DX1 module on the left window under "Input Modules" and confirm with the "Select" button. The module is selected into the right "Selected modules" window.
  – Select the "LE4-116-XD1" under "Output modules".

Both modules are displayed on the right side window and are part of the configuration. This completes the configuration.

→ If you use the LE4 with analog inputs/outputs, also read the section "Parametric programming of the LE4 with analog inputs/outputs" on Page 96.

Structure of the program example with a master

The PLC_PRG main program processes the inputs and outputs and calls the DP_DIAG subprogram which contains the diagnostics in the first section and the communications query in the second section. The communication query is implemented for two stations. If you wish to add more slaves, copy a program section and add the parameters to the declaration section.

In general, the following programming measures should be implemented:

▶ Create a GETBUSSTATE global variable type:

```
DPSTAT AT%MB4: GETBUSSTATE
```

▶ Enter the maximum bus address in the declaration section:

```
Adr_max_DP: BYTE:=124;
```

→ In this example "3" is the maximum address. If a higher address is entered, e. g. 124, without the devices actually being physically connected, the time for processing the program is extended.

Function of the program example

If a voltage is applied to input IX0.0 (Input_0 = first input on the I/O module of the CPU) the following outputs should be set:

- QX0.0 (Output_0) = first output on I/O module of the CPU,
- QX2.0 (Output_S2) first output on XION module,
- QX4.0 (Output_S4) = first output on LE4-116-XD1.

Function of the diagnostics program

Bit 2 of all station bytes must be checked for querying the diagnostics messages. This occurs with the instruction:

```
IF DPSTAT.EXTENDEDINFO[n_DWORD] >=6 THEN
  DPSTAT is an instance name of GETBUSSTATE
  N_DWORD = address of the slave
```

Sends the slave a diagnostic alarm, e.g. a short-circuit, bit 2 of the station byte is set. The DIAGGETSTATE function block is accessed and the DIAGGETSTATE.EXTENDEDINFO output array is copied in a DIAGDATA_DP dummy field. You can take the diagnostics data directly from the "DIAGSTATE.EXTENDEDINFO" output array or from the "DIAGDATA_DP" output array.

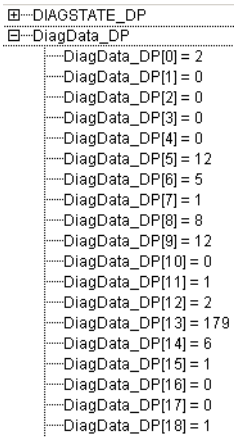
If a fault has been recognized and processed, the GETBUSSTATE.EXTENDEDINFO output array recommences the query at the first station.

If a direct query is demanded, you can set an auxiliary marker which indicates when an error message is received (⟶ note in program example) and queries the fault code contained in it.

The content of the "DiagData_DP" array corresponds with the content of the "DiagGetState.EXTENDEDINFO" array. The array is described in section "Data content of DIAGGETSTATE.EXTENDEDINFO The data content of DIAGGETSTATE.EXTENDEDINFO is subdivided into:" on Page 86.

If a short-circuit occurs on output QX2.0 (first output of the XION station) the fault is diagnosed.

In online mode the "DiagData_DP" array contains the following details:



- The "2" in the byte DiagData_DP[0] = 2 indicates the address of the slave.
- Bit 3 is set in byte 8: Extended diagnostics exist (Bit 3 = 1 signal → 00001000 binary or 8 decimal)

This indicates that further information exists for example in byte 15 and 18:

- Bit 0 is set in byte 15: module diagnostics exist
- Bit 0 is set in byte 18: overcurrent channel 1

If the short-circuit is eliminated, the slave sends the diagnostics message again which causes the bit to reset.

Function of the data exchange (monitoring)

Bit 1 of all stations should be queried to check the data exchange. This occurs with the instruction:

```
IF DPSTAT.EXTENDEDINFO[n].1 = TRUE THEN
 DPSTAT is an instance name of GETBUSSTATE
 n = address of the slave
```

With an existing connection the variables KOM2_ok or KOM3_ok are set to "1". If the connection to a slave is interrupted the variables are reset to "0".

The variables KOMx_ok can be used again in the main program.

Program example for diagnostics with a master

Global variable declaration

```
VAR_GLOBAL
   DPSTAT AT %MB4:                    GetBusState;            (*Must be generally declared*)
(*See description "Create and query variables of the GETBUSSTATE type" *)
   KOM2_ok:                           BOOL;
   KOM3_ok:                           BOOL;
   Input_0 AT %IX0.0:                 BOOL;
   Output_0 AT %QX0.0:                BOOL;
   Output_S2 AT %QX2.0:               BOOL;
   Output_S4 AT %QX4.0:               BOOL;
END_VAR
```

PROGRAM PLC_PRG

```
Declaration:
VAR
END_VAR

Program:
Output_0:=Input_0;
Output_S2:=Input_0;
Output_S4:=Input_0;

DIAG_DP;                                                    (*Diagnostics program*)

(*  IF KOM2_ok =TRUE THEN                                   Data exchange query ok?
   Data transfer: Master <-> Slave 2                        Run data exchange!
   END_IF*)

(*  IF KOM3_ok =TRUE THEN                                   Data exchange query ok?
   Data transfer: Master <-> Slave 3                        Run data exchange!
   END_IF*)
```

PROGRAMM DIAG_DP

```
Declaration:
VAR
    DIAGSTATE_DP :       DiagGetState;
    DiagData_DP:         ARRAY[0..99] OF BYTE ;
    wHelp_DP:            WORD;
    Adresse_DP:          DWORD;
    n_DWORD:             DWORD;
END_VAR
VAR CONSTANT
    Adr_max_DP:          BYTE:=124;                    (*Enter max. bus address!*)
END_VAR
```

```
Program:
(*-----------------------------------Diagnostics-----------------------------------------------*)
IF  DIAGSTATE_DP.ENABLE = FALSE THEN
    Adresse_DP:=0;
    FOR n_DWORD:=2 TO Adr_max_DP DO
        IF (DPSTAT.EXTENDEDINFO[n_DWORD] >=6) THEN
            Address_DP:=n_DWORD;
            EXIT;
        END_IF
    END_FOR

    IF  DIAGSTATE_DP.ENABLE = FALSE THEN
        DIAGSTATE_DP.DRIVERNAME:=0;                         (* always 0 *)
        DIAGSTATE_DP.DEVICENUMBER:=0;                       (* DP master is the first device with DeviceNo = 0*)
        DIAGSTATE_DP.BUSMEMBERID:=Adresse_DP;               (* Slave Address *)
        DIAGSTATE_DP.ENABLE:=TRUE;
        DIAGSTATE_DP();                                     (* Call FB *)
    END_IF
END_IF
IF DIAGSTATE_DP.ENABLE = TRUE THEN
    IF DIAGSTATE_DP.READY THEN
        IF DIAGSTATE_DP.STATE=NDSTATE_DIAGINFO_AVAILABLE THEN
        (*Diaginfo:=TRUE;*)
        (*Set auxiliary marker: If diagnostics data query =0->1, the diagnostics data is valid and can be queried.
        The marker must be reset in the user program.*)
            FOR wHelp_DP:=0 TO (DIAGSTATE_DP.EXTENDEDINFO[14]+13) BY 1 DO
            DiagData_DP[wHelp_DP]:=DIAGSTATE_DP.EXTENDEDINFO[wHelp_DP];
            END_FOR
        END_IF
        DIAGSTATE_DP.ENABLE:=FALSE;
    END_IF
    DIAGSTATE_DP();
END_IF

(* Communication ok-- Slave 2 ------------------------------------------------*)

IF DPSTAT.EXTENDEDINFO[2].1 = TRUE THEN
KOM2_ok:=FALSE;
ELSE
KOM2_ok:=TRUE;
END_IF

(* Communication ok-- Slave 3 ------------------------------------------------*)

IF DPSTAT.EXTENDEDINFO[3].1 = TRUE THEN
KOM3_ok:=FALSE;
ELSE
KOM3_ok:=TRUE;
END_IF

(* End of ProfibusDP diagnostics *)
```

Parametric programming of the LE4 with analog inputs/outputs

In this section you will discover how the LE4-206-AA1 and LE4-206-AA2 analog modules parameters are programmed with the aid of the easySoft-CoDeSys configurator:

▶ Add the EM4 -204-DX1 to the configuration and select the analog modules:

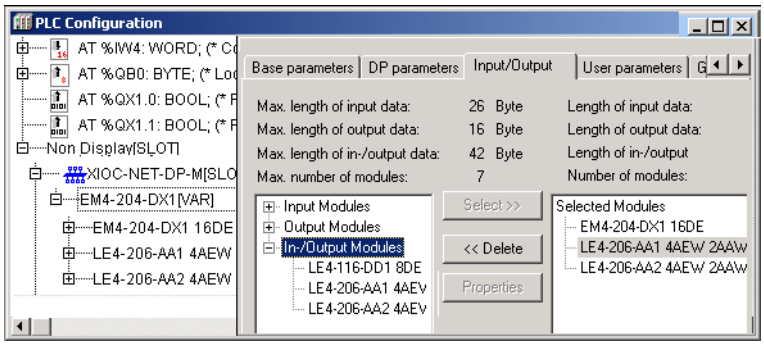


Figure 97: Adding analog modules to the configuration

▶ Mark a LE4 and click on the "Properties" button.

The "module properties" window opens.

▶ Click on the "IO count/Resolution/IOscan" text.

The following parameter setting properties are displayed:

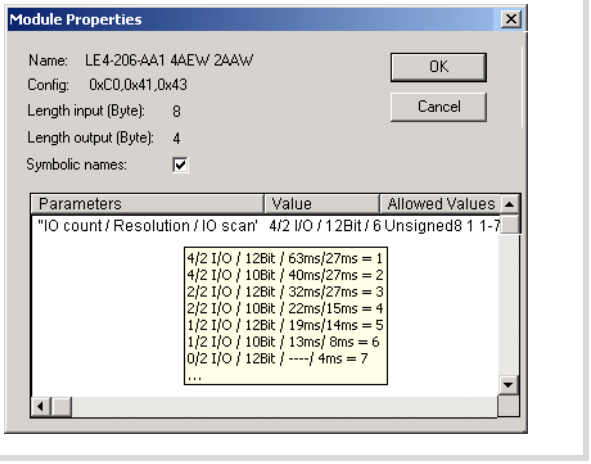


Figure 98: Analog module parameter module

The standard parameters are defined in the "value" field.
You can change the setting by clicking on the first entry.
The following value is displayed with each double click.

9 Technical data

XControl

General	
Standards and regulations	IEC/EN 61131-2, EN 50178
Ambient temperature	0 to +55 °C
Storage temperature	–25 to +70 °C
Vibration resistance	10 – 57 Hz ±0.075 mm, 57 – 150 Hz ±1.0 g
Mechanical shock resistance	15 g/11 ms
Shock resistance	500 g/∅ 50 mm ±25 g
Overvoltage category	II
Pollution degree	2
Protection class	1
Enclosure protection	IP20
Emitted interference	DIN/EN 55011/22, Class A
Electromagnetic compatibility	
Electrostatic discharge (IEC/EN 61 000-4-2)	
Contact discharge	4 kV
Radiated (IEC/EN 61 000-4-3, RFI)	
AM/PM	10 V/m
Burst (IEC/EN 61 000-4-4)	
Supply cables	2 kV
Signal cables	1 kV
Power pulses (surge) (IEC/EN 61 000-4-5)	
Supply cables, asymmetrical	0.5 kV
Radiated RFI (IEC/EN 61 000-4-6)	
AM	10 V
External supply voltage	
Rated voltage U_e	24 V DC
Permissible range	20.4 to 28.8 V DC
Input voltage ripple	< 5 %
Bridging voltage dips	
Drop-out duration	10 ms
Repeat rate	1 s

Digital input modules

Type	XIOC-8DI	XIOC-16DI	XIOC-32DI
Input type	DC input	DC input	DC input
Number of input channels	8	16	32
Number of channels with common reference potential[1)]	8	16	32, reference potential: 4 terminals
Input voltage	24 V DC	24 V DC	24 V DC
Input voltage range	20.4 to 28.8 V DC	20.4 to 28.8 V DC	20.4 to 28.8 V DC
Input resistance	Typ. 6 kΩ	Typ. 6 kΩ	Typ. 5.6 kΩ
Input current	Typ. 4.0 mA	Typ. 4.0 mA	Typ. 4.3 mA
Voltage level			
ON	≧ 15V	≧ 15V	≧ 15V
OFF	≦ 5V	≦ 5V	≦ 5V
Input signal delay			
OFF→ ON	≦ 1 ms	≦ 1 ms	5 ms
ON → OFF	≦ 1 ms	≦ 1 ms	5 ms
Electrical isolation			
between inputs and the I/O bus	Through optocouplers	Through optocouplers	Through optocouplers
Input indication	By LED (green)	By LED (green)	With LED (green)[2)]
External connection	Plug-in terminal block[3)]	Plug-in terminal block[3)]	XIOC-TERM32 (connector/cable)[3)]
Internal current consumption (5 V DC)	Typ. 6 mA	Typ. 10 mA	Typ. 100 mA
Weight	0.16 kg	0.16 kg	0.16 kg

1) The reference potential terminals are internally connected.

2) LED convertible 0 – 15, 16 – 31 (→ figure 1 on Page 12)

3) Not supplied with the module

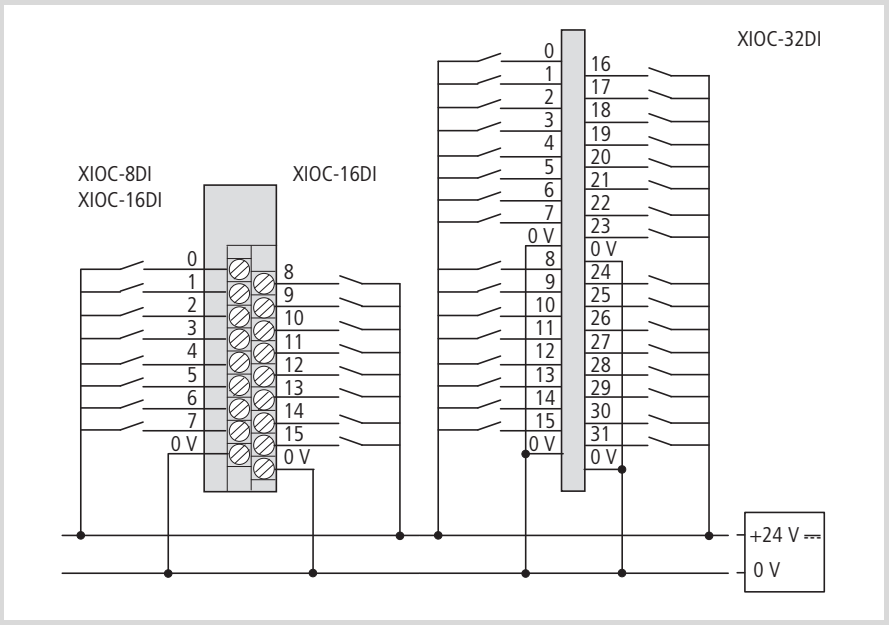


Figure 100: Terminal assignment

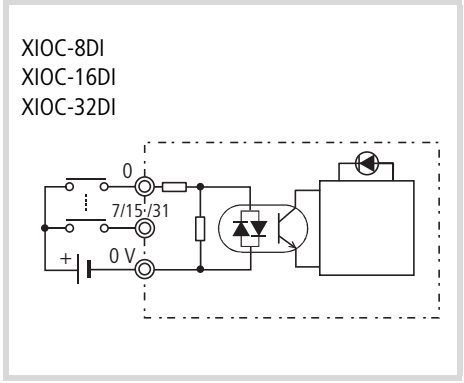


Figure 99: Connection example

Type	XIOC-16DI-110VAC	XIOC-16DI-AC
Input type	AC input	AC input
Number of input channels	16	16
Number of channels with common reference potential[1)]	16	16
Input voltage	100 to 120 V AC	200 to 240 V AC
Input voltage range	85 to 132 V DC	170 to 264 V DC
Input resistance	Typ. 16 kΩ (50 Hz) Typ. 13 kΩ (60 Hz)	Typ. 32 kΩ (50 Hz) Typ. 27 kΩ (60 Hz)
Input current	4.8 to 7.6 mA (100 V AC/50 Hz)	4.3 to 8.0 mA (200 V AC/50 Hz)
Voltage level		
ON	≧ 79 V AC	≧ 164 V AC
OFF	≦ 20 V AC	≦ 40 V AC
Input signal delay		
OFF→ ON	≦ 15 ms	≦ 15 ms
ON → OFF	≦ 25 ms	≦ 25 ms
Electrical isolation		
between inputs and the I/O bus	Through optocouplers	Through optocouplers
Input indication	By LED (green)	By LED (green)
External connection	Plug-in terminal block[2)]	Plug-in terminal block[2)]
Internal current consumption (5 V DC)	Typ. 51 mA	Typ. 51 mA
Weight	0.18 kg	0.18 kg

1) The reference potential terminals are internally connected.

2) Not supplied with the module

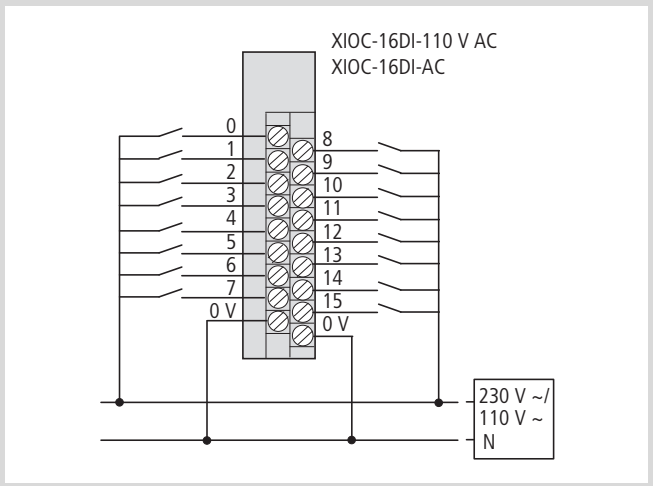


Figure 101: Terminal assignment

Digital output modules

Transistor output modules

Type	XIOC-8DO	XIOC-16DO	XIOC-32DO
Output type	Transistor output (source type)	Transistor output (source type)	Transistor output (source type)
Number of output channels	8	16	32
Number of channels with common reference potential	8	16	32
Output voltage	24 V DC	24 V DC	24 V DC
Switching current, minimum	1 mA	1 mA	1 mA
Residual current for a "0" signal	0.1 mA	0.1 mA	0.1 mA
Rated operational current			
for "1" signal	0.5 A	0.5 A	0.2 A
Per common potential terminal	4 A	8 A	3.2 A (Σ = 6.4 A)
Output signal delay			
OFF→ ON	≦ 25 µs	≦ 25 µs	≦ 0.3 ms
Overvoltage protection	Diode	Diode	Diode
Fuse[1)]	-	-	8A
Electrical isolation			
between outputs and the I/O bus	Through optocouplers	Through optocouplers	Through optocouplers
Short-circuit protection	Yes	Yes	–
Output indication	By LED (green)	By LED (green)	With 16 LEDs (green)[2)]
External connection	Plug-in terminal block[3)]	Plug-in terminal block[3)]	XIOC-TERM32 (connector and cable)[3)]
Internal current consumption (5 V DC)	Max. 80 mA	Max. 150 mA	Typ. 250 mA
External power supply[4)]	24 V DC (Page 97)	24 V DC (Page 97)	24 V DC (Page 97)
Weight	0.16 kg	0.16 kg	0.16 kg

1) A blown fuse must not be replaced by the user.

2) LED convertible: 0 – 15, 16 – 31 (→ figure 1 on Page 12)

3) Not supplied with the module

4) Important! For UL applications the power supply lines must have a cross-section of AWG16 (1.3 mm^2).

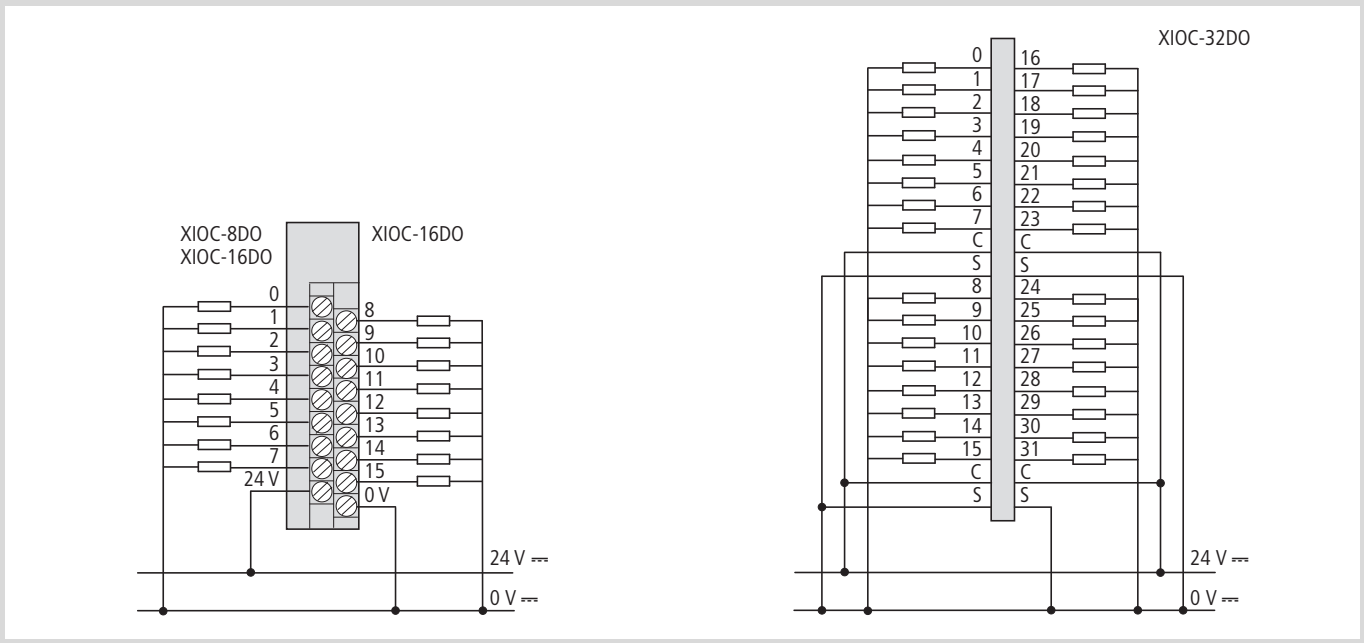


Figure 102: Assignment of the terminals and pins

Relay output module

Type	XIOC-12DO-R
Output type	Relay output
Number of output channels	12
Number of channels with common reference potential[1)]	12
Output voltage	100/240 V AC, 24 V DC
Switching current, minimum	1 mA
Rated operational current	
for "1" signal	2 A
Per common potential terminal	5 A
Output signal delay	
OFF→ ON	≦ 10 ms
ON → OFF	≦ 10 ms
Overvoltage protection	External
Fuse	External
Potential isolation between relay and the I/O bus	Through optocouplers
Output indication	By LED (green)
General	
External connection	Plug-in terminal block[2)]
Internal current consumption (5 V DC)	Typ. 40 mA
External power supply[3)]	24 V DC→ page 97
Weight	0.2 kg

→ Legends in the next column

Legend for the table:

1) The reference potential terminals are internally connected.
2) Not supplied with the module
3) An external 24 V DC voltage must applied. Caution! For UL applications the power supply lines must have a cross-section of AWG16 (1.3 mm^2).

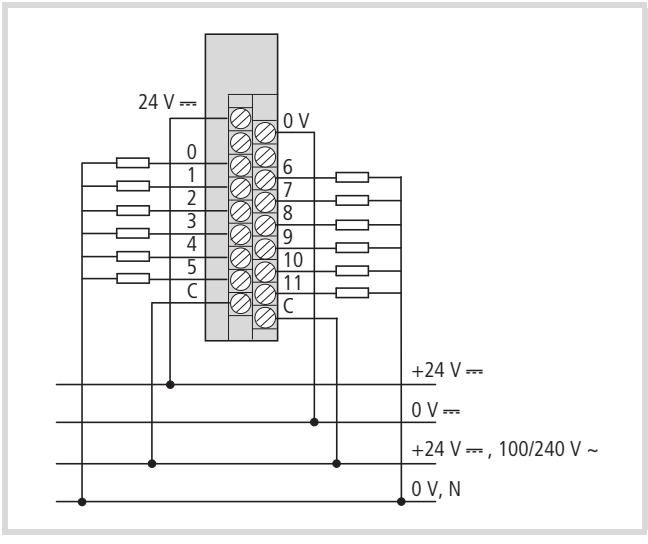


Figure 103: Terminal assignment for the XIOC-12DO-R module

Digital input/output modules

Caution!
The supply voltages for the inputs and outputs must come from the same source as those for the module.

Type	XIOC--16DX
Inputs	
Input type	DC input
Number of input channels	16 (0 to 15)
Input voltage	24 V DC
Range	20.4 to 28.8 V DC
Input resistance	5.6 kΩ
Input current	Typ. 4 mA
Voltage level	
ON	≧ 15V
OFF	≦ 5V
Input signal delay	
OFF→ ON	typically 100 µs
ON → OFF	typically 1 ms
Electrical isolation	
between inputs and the I/O bus	Through optocouplers
Input indication	By LED (green)
Outputs	
Output type	Transistor (Source)
Number of outputs	12 (0 to 11)
Output voltage	24 V DC
Residual current for a "0" signal	approx. 140 µA
Rated operational current	
for "1" signal	0.5 A DC at 24 V DC
Lamp load	4 W, without series resistor
Simultaneity factor g	1
Relative ON time (duty cycle)	100 %
Limiting of switch-off voltage	
For inductive loads	yes, –21 V (for U_N = 24 V DC)
Switching repetition rate (actions per hour)	
For time constant t ≤ 72 ms	3600 (G = 1)
Parallel wiring capability of outputs	in groups 0 to 3, 4 to 7, 8 to 11; actuation of the outputs within a group only in the same program cycle
Number of outputs	max. 3
Maximum total current	2 A per group
Minimum total current	250 mA

Type	XIOC--16DX
Output signal delay	typ. 100 µs
Overvoltage protection	Diode
Potential isolation between outputs and the I/O bus	Through optocouplers
Short-circuit protection	Yes
Short-circuit tripping current	max. 1.2 A for 3 ms per output
Output indication	By LED (green)
General	
External connection[1)]	Plug-in terminal block
Internal current sink	Typ. 50 mA
External supply voltage[2)]	24 V DC→ page 97
Weight	0.16 kg

1) Not supplied with the module

2) Important! For UL applications the power supply lines must have a cross-section of AWG16 (1.3 mm²).

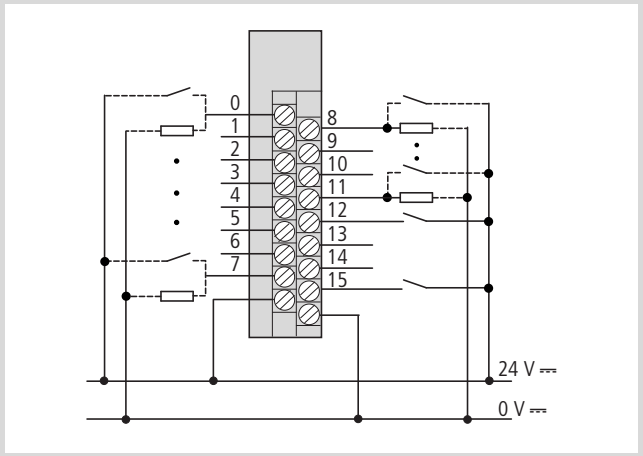


Figure 104: Terminal assignments for module XIOC-16DX

Configuration and programming of the digital inputs/outputs

The module has 16 connections. The first 12 connections (0 to 11) can be used as inputs and outputs, the connections 12 to 15 can only be used as inputs → figure 104.

The configuration of the module is undertaken in the "PLC configuration" tab. It is inserted at an "Empty slot" with "Set element". For example, the following appears:

```
---XIOC-16DXISLOT
  ---AT%IW6:WORD!(*Inputs/Outputs*) [CHANNEL (I)]
  ---AT%QW2:WORD!(*Outputs/Inputs*) [CHANNEL (I)]
```

After a double click on the input word:

```
---AT%IW6:WORD;(*Inputs/Outputs*) [CHANNEL (I)]
  ---AT%IX6.0:BOOL;(*Bit 0*)
  ---AT%IX6.1:BOOL;(*Bit 1*)
  to
  ---AT%IX6.7:BOOL;(*Bit 7*)
  ---AT%IX7.0:BOOL;(*Bit 0*)
  ---AT%IX7.1:BOOL;(*Bit 1*)
  bis
  ---AT%IX7.7:BOOL;(*Bit 7*)
```

After a double click on the output word:

```
---AT%QW2:WORD;(*Outputs/Inputs*) [CHANNEL (I)]
  ---AT%QX2.0:BOOL;(*Bit 0*)
  ---AT%QX2.1:BOOL;(*Bit 1*)
  bis
  ---AT%QX2.7:BOOL;(*Bit 7*)


  ---AT%QX3.0:BOOL;(*Bit 0*)
  ---AT%QX3.1:BOOL;(*Bit 1*)
  ---AT%QX3.2:BOOL;(*Bit 2*)
  ---AT%QX3.3:BOOL;(*Bit 3*)
  ---AT%QX3.4:BOOL;(*Bit 4*)
  ---AT%QX3.5:BOOL;(*Bit 5*)
  ---AT%QX3.6:BOOL;(*Bit 6*)
  ---AT%QX3.7:BOOL;(*Bit 7*)
```

→ The marked outputs (Bit4 … 7) can not be used!

Example
The connection "I/Q0" of the XIOC-16DX should be programmed as an input or output. The connection should be wired corresponding to the program.

- Programming the connection as an input

Declaration:

```
Start AT% IX6.0:   BOOL;
Valve:             BOOL;
```

Program (IL):

```
LD Start
ST Valve
```

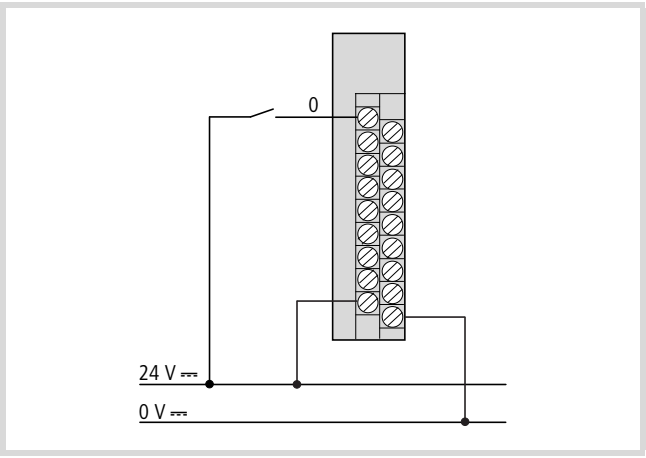


Figure 105: Wiring the connection as an input

- Programming the connection as an output

Declaration:

```
motor AT% QX2.0:   BOOL;
Start:             BOOL;
```

Program (IL):

```
LD Start
ST Motor
```

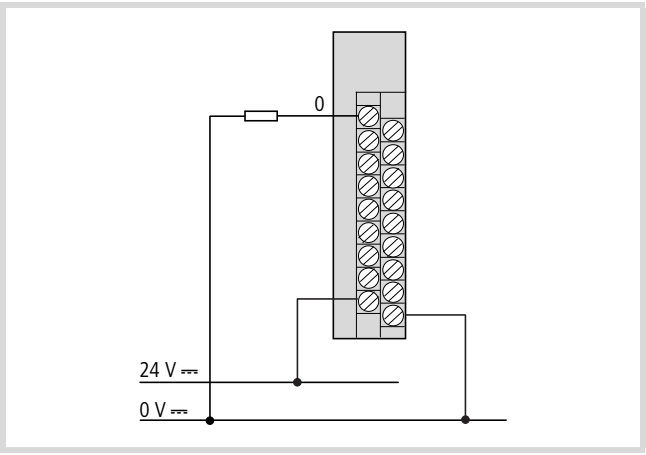


Figure 106: Wiring the connection as an output

You can proceed in the same manner with connections 1 to 11. The connections 12 to 15 can only be programmed as inputs.

Analog input modules

Type	XIOC-8AI-I2	XIOC-8AI-U1	XIOC-8AI-U2
Input current range	4 to 20 mA	–	–
Input voltage range	–	0 – 10 V DC	–10 to 10 V DC
Resolution	12 Bit	12 Bit	12 Bit
Conversion time	≦ 5 ms	≦ 5 ms	≦ 5 ms
Overall accuracy	≦ ±1 % (of end of scale)	≦ ±1 % (of end of scale)	≦ ±1 % (of end of scale)
Input resistance	–		
Voltage input	–	100 kΩ	100 kΩ
Current input	Typ. 100 Ω	–	–
Electrical isolation			
Channel to internal circuitry	Through optocouplers	Through optocouplers	Through optocouplers
Channel to channel	–	–	–
Number of channels	8	8	8
External connection	Plug-in terminal block (not supplied with the module)		
Internal current consumption (5 V DC)	100 mA	100 mA	100 mA
External supply voltage	24 V DC (+20 %, –15 %), approx. 0.15 A (approx. 0.4 A with supply switched on)		
External cabling	2-core shielded cable (≦ 20 m)		
Weight	0.18 kg	0.18 kg	0.18 kg

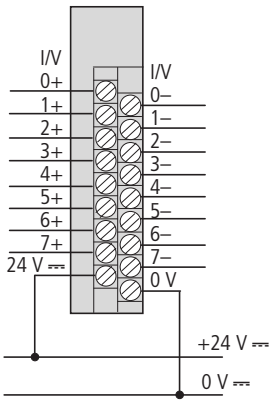
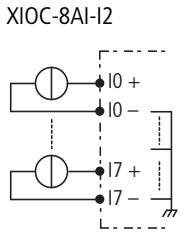


Figure 107: Terminal assignments for modules XIOC-8AI-I2 and XIOC-8AI-U1/-U2



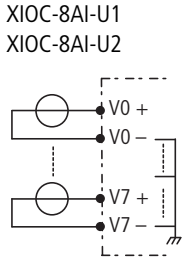


Figure 108: Module wiring

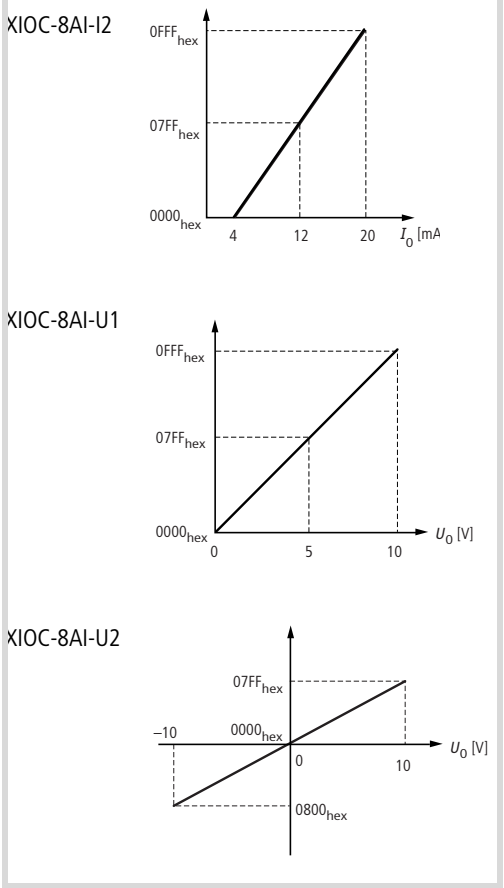


Figure 109: U/I diagram for the modules

Analog output module

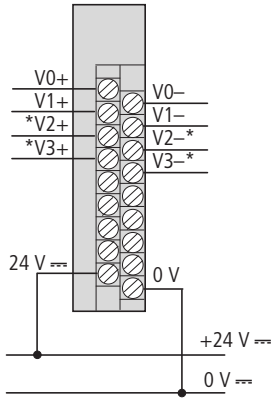
Type	XIOC-2AO-U1-2AO-I2	XIOC-2AO-U2	XIOC-4AO-U1	XIOC-4AO-U2
Output voltage range	0 – 10 V DC	–10 to 10 V DC	0 – 10 V DC	–10 to 10 V DC
Output current range	4 to 20 mA	–	–	–
Resolution	12 Bit	12 Bit	12 Bit	12 Bit
Conversion time[1)]	≦ 5 ms	≦ 5 ms	≦ 5 ms	≦ 5 ms
Overall accuracy	≦ ±1 % (of end of scale)			
External load resistance				
Voltage output	≧10k Ω	≧10k Ω	≧ 10 kΩ	≧ 10 kΩ
Current output	0 to 500 Ω	–	–	–
Electrical isolation				
Channel to internal circuitry	Through optocouplers	Through optocouplers	Through optocouplers	Through optocouplers
Channel to channel	–	–	–	–
Number of channels				
Output voltage[2)]	2 Channels (0 to 1)	2	4	4
Output current[2)]	2 channels (2 to 3)	–	–	–
External connection	Plug-in terminal block[3)]			
Internal current consumption (5 V DC)	Typ. 100 mA	Typ. 100 mA	Typ. 100 mA	Typ. 100 mA
External supply voltage	24 V DC (+20 %, –15 %), approx. 0.15 A (approx. 0.5 A with supply switched on)			
External cabling	2-core screened cable (≦ 20 m)			
Weight	0.18 kg	0.18 kg	0.18 kg	0.18 kg

1) The 5 ms refer to the conversion time of the ASIC. The nature of the output circuitry for the voltage outputs means that the settling time (to reach the final output value) varies according to the size of the voltage change. The longest time is required for a step voltage change from –10 V to +10 V:
–10 V → +10 V: 30 ms
0 V → +10 V: 5 ms
+10 V → 0 V: 14 ms
0 V → +1V: 1 ms
+1 V → 0 V: 3 ms

2) On the XIOC-2AO-U1-2AO-I2, the current and voltage outputs can be used at the same time.

3) Not supplied with the module

XIOC-2AO-U2
XIOC-4AO-U1/-U2



XIOC-2AO-U1-2AO-I2

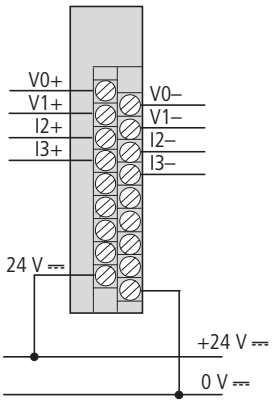
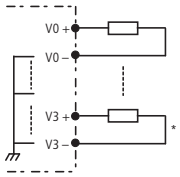


Figure 110: Terminal assignment

* not for XIOC-2AO-U2

XIOC-2AO-U2
XIOC-4AO-U1/-U2



* not for XIOC-2AO-U2

XIOC-2AO-U1-2A0-I2

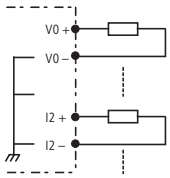
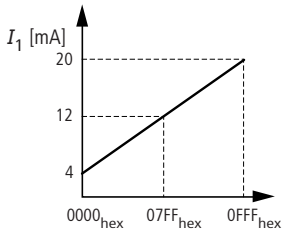
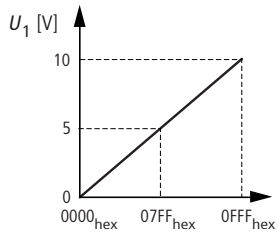


Figure 111: Module wiring

XIOC-2AO-U1-2A0-I2



XIOC-2AO-U1-2A0-I2
XIOC-4AO-U1



XIOC-2AO-U2
XIOC-4AO-U2

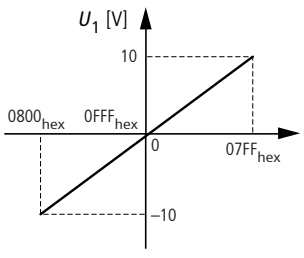


Figure 112: U/I diagram for the modules

Analog input/output modules

→ The modules can be operated with the CPUs XC-CPU101 from Version V02 and XC-CPU201.

Type	XIOC-4AI-2AO-U1	XIOC-2AI-1AO-U1
General		
External connection	Plug-in terminal block[1)]	Plug-in terminal block[1)]
Internal current consumption (5 V DC)	200 mA	200 mA
Weight	0.16 kg	0.16 kg
Inputs		
Input voltage range	0 – 10 V DC	0 – 10 V DC
Resolution	14 Bit	14 Bit
Conversion time	≦ 1 ms	≦ 1 ms
Overall accuracy	≦ 0.4 % (of end of scale)	≦ 0.4 % (of end of scale)
Input resistance	40 kΩ	40 kΩ
Electrical isolation		
Channel to internal circuitry	–	–
Channel to channel	–	–
Number of channels	4	2
Outputs		
Output voltage range	0 – 10 V DC	0 – 10 V DC
Resolution	12 Bit	12 Bit
Conversion time	≦ 1ms	≦ 1 ms
Overall accuracy	≦ 0.4 % (of end of scale)	≦ 0.4 % (of end of scale)
External load resistance	≧ 2 kΩ	≧ 2 kΩ
Electrical isolation		
Channel to internal circuitry	–	–
Channel to channel	–	–
Number of channels	2	1

1) Not supplied with the module

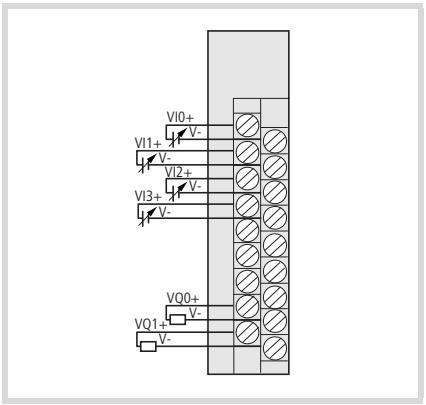
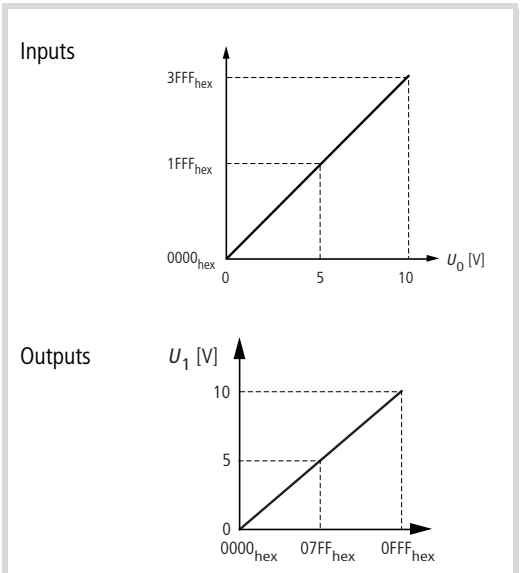


Figure 113: Terminal assignments for modules XIOC-4AI-2AO-U1 and XIOC-2AI-1AO-U1



Type		XIOC-2AI-1AO-U1-I1		XIOC-4AI-2AO-U1-I1	
For setting the "current" and "voltage" signal types → page 21					
General					
External connection		Plug-in terminal block (not supplied with the module)			
Internal current consumption (5 V DC) with signal type:					
Input	**Output**				
Voltage	Voltage	220 mA		270 mA	
Voltage	Current	280 mA		380 mA	
Current	Voltage	220 mA		270 mA	
Current	Current	280 mA		380 mA	
Electrical isolation					
Channel to internal circuitry		–		–	
Channel to channel		–		–	
Weight		0.16 kg		0.16 kg	
Inputs					
Number of channels		2		4	
Signal type		Voltage	Current	Voltage	Current
Input voltage range		0 – 10 V DC	0 to 20 mA	0 – 10 V DC	0 to 20 mA
Resolution		14 Bit		14 Bit	
Conversion time		≦ 1 ms		≦ 1 ms	
Overall accuracy		≦ 0.4 % (of end of scale)		≦ 0.4 % (of end of scale)	
Input resistance		40 kΩ	125 Ω	40 kΩ	125 Ω
Outputs					
Number of channels		1		2	
Signal type		Voltage	Current	Voltage	Current
Output voltage range		0 – 10 V DC	0 to 20 mA	0 – 10 V DC	0 to 20 mA
Resolution		12 Bit		12 Bit	
Conversion time		≦ 1ms		≦ 1 ms	
Overall accuracy		≦ 0.4 % (of end of scale)		≦ 0.4 % (of end of scale)	
External load resistance		≧ 2 kΩ	≦ 0.5 kΩ	≧ 2 kΩ	≦ 0.5 kΩ
Short-circuit proof		Yes	Yes	Yes	Yes

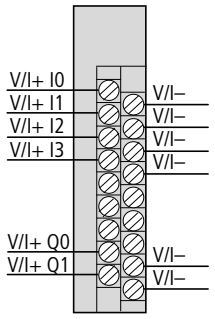
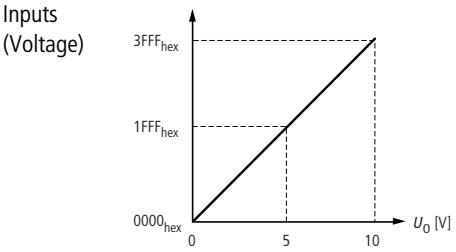
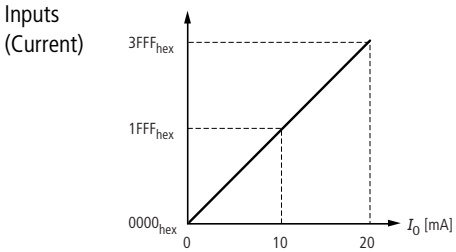
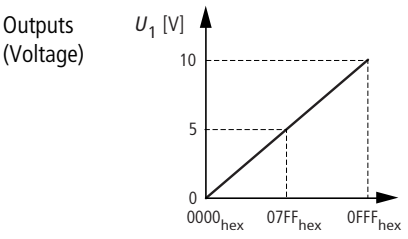
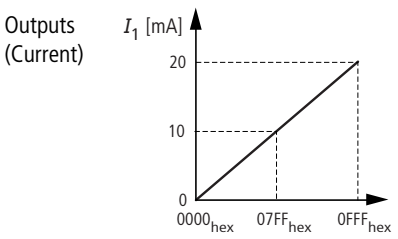


Figure 114: Terminal assignment of the XIOC-2AI-1AO-U1-I1 (I0, I1, Q0) and XIOC-4AI-2AO-U1-I1 (I0 to I3, Q0 to Q1) modules









Temperature acquisition module XIOC-4T-PT

→ More information on the temperature acquisition module can be found in chapter 2 from Page 25 onwards.

Type	XIOC-4T-PT
Platinum temperature resistance	Pt100 (IEC 751) / Pt1000
Temperature resolution	15 bit, with sign
Accuracy[1]	
–20 to 40 °C (Pt100)	±0.5 °C
–50 to 400 °C (Pt100)	±3 °C
–50 to 400 °C (Pt1000)	±6 °C
Temperature measurement range	–20 to +40 °C/–50 to +400 °C (constant current 2 mA)
Number of inputs	4
Conversion time	Typ. 1 second for 4 channels
Electrical isolation	
Between inputs and the I/O bus	Through optocoupler
Between inputs	–
External supply voltage	24 V DC
Internal current consumption	Max. 200 mA
External resistance	Max. 400 Ω/channel
External cabling	Screened cable[2]
Additional functions	Linearization
Fault detection –20 to +40 °C –50 to +400 °C	The resistance value is $7FFF_{hex}$ at: ≦ –25 °C or ≧ 45 °C ≦ –60 °C or ≧ 410 °C
Response to cable break or unused inputs	In this case, the resistance is $7FFF_{hex}$.
Weight	0.18 kg

1) The quoted accuracy applies after 10 minutes of operation. The maximum temperature deviation can be somewhat larger just after the start.
The characteristics of the RTD resistor must also be checked for correctness.

2) Not supplied with the module

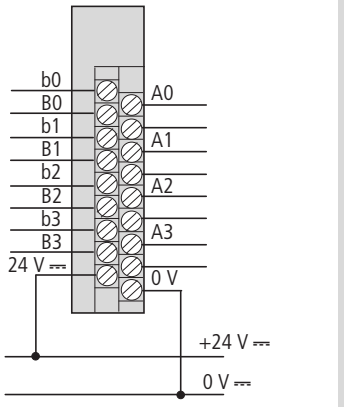


Figure 116: Terminal assignments for module XIOC-4T-PT

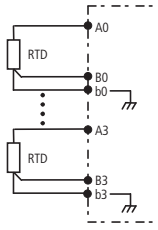


Figure 115: Module wiring

Temperature acquisition module XIOC-4AI-T

→ More information on the temperature acquisition module can be found in chapter 2 from Page 31 onwards.

Type		XIOC-4AI-T
Channels		
	Number	4
	Temperature measurement range	K type: -270 – 1370 J type: -210 – 1200 B type: 100 – 1800 N type: -270 – 1300 E type: -270 – 1000 R type: -50 – 1760 T type: -200 – 400
	Voltage measurement	– 50 mV – 50 mV –100 mV – 100 mV –500 mV – 500 mV –1000 mV – 1000 mV
	Cold-junction compensation	yes, integrated
	Interference voltage suppression	50 Hz, 60 Hz
	Unit	0.1 °C, 0.1 F
	Resolution	16 bits
	Total error	±0.5 % of range Element "E" from –270 °C to –180 °C ±2 % of measurement range
	Max. input voltage (destruction threshold)	10 V DC
	Insulation voltage	500 V_{rms} between input cables and bus backplane
	Conversion time	< 1 s
	Temperature coefficient	< 200 ppm/°C from measurement range
	Weight	0.18 kg

Counter module

→ More information on wiring up the counter module can be found in chapter 3 from Page 33.

Type	XIOC-2CNT-100 kHz	XIOC-1CNT-100kHz
Electrical isolation	250 V DC between I/O signal and bus	250 V DC between I/O signal and bus
Internal current consumption (5 V DC)	200 mA	200 mA
Ambient temperature + humidity in operation	0 to 55 °C, 20 to 90 % relative humidity (no condensation)	
Ambient temperature + humidity in storage	–10 to 75 °C, 10 to 90 % relative humidity (no condensation)	
Input		
Maximum count value	32 bit (0 to 4294967295)	32 bit (0 to 4294967295)
Maximum frequency	100 kHz (25 kHz with 4x resolution)	100 kHz (25 kHz with 4x resolution)
Number of channels	2 channels	1 channel
Input voltage	12 to 24 V DC	12 to 24 V DC
Voltage for ON	> 10 V DC	> 10 V DC
Voltage for OFF	< 4 V DC	< 4 V DC
Input current	≧ 4 mA	≧ 4 mA
Differential input voltage	+/– 5 V DC	+/– 5 V DC
Voltage for ON	2 to 5 V DC	2 to 5 V DC
Voltage for OFF	–5 to –0.8 V DC	–5 to –0.8 V DC
Differential input current	35 mA	35 mA
Electrical isolation	Through optocoupler	Through optocoupler
Number of inputs per channel	3	3
Minimum width of count pulse	ON: ≧ 4 µs, OFF: ≧ 4 µs	ON: ≧ 4 µs, OFF: ≧ 4 µs
Minimum width of marker	≧ 10 µs (during an ON transition)	≧ 10 µs (during an ON transition)
Connection for external cabling	30 pole connector XIOC-TERM30-CNT4[1)]	30 pole connector XIOC-TERM30-CNT4[1)]
External cabling	Twisted pair, screened[1)]	Twisted pair, screened[1)]
Output		
Type of output	Transistor (open collector)	Transistor (open collector)
External voltage	12/24 V DC (max. 30 V DC)	12/24 V DC (max. 30 V DC)
Minimum load current	1 mA	1 mA
Maximum load current	20 mA per output	20 mA per output
Leakage current	Max. 0.5 mA	Max. 0.5 mA
Output delay time		
ON → OFF	≦ 1 ms	≦ 1 ms
OFF→ ON	≦ 1 ms	≦ 1 ms
Voltage drop in ON state	Max. 1.5 V	Max. 1.5 V
Number of external outputs	4 outputs per module	2 outputs per module
Up/down counter	Actual (process) value ≧ setpoint value 1	Actual (process) value ≧ setpoint value 1
Ring counter	Actual (process) value = setpoint value 2	Actual (process) value = setpoint value 2
Electrical isolation	Through optocouplers	Through optocouplers

1) Not supplied with the unit

Counter analog module

→ More information on wiring up the analog counter module can be found in chapter 4 from Page 49.

Type	XIOC-2CNT-2AO-INC
General	
Channel count	2
Max. internal current consumption	450 mA
Inputs	
Counter width	32 Bit
Signals to RS422	A, !A, B, !B, R, !R
Input voltage differential	+/– 5 V DC
High	0.2 to 5 V DC
Low	–5 to –0.2 V DC
Potential isolation	
IO bus → inputs	No
Between inputs	No
Between inputs	No
Input frequency	400 kHz
Operating modes	1x, 2x, 4x signal edge evaluation
Outputs (analog)	
Resolution	12 Bit
Output voltage range	–10 to +10 V
Error	typically 0.4 %
Potential isolation	
IO bus → outputs	No
Between outputs	No
Conversion time	< 1 ms
Max. load current	10 mA
Min. load resistance	1 kΩ
Short-circuit proof	Yes
Max. output current (min. load resistance)	10 mA 1 kΩ
Power supply for encoder	
Voltage	5 V DC
Current or channel[1)]	Max. 300 mA

1) Apply an external encoder supply if the current available is insufficient.

Serial interface module/Telecontrol module

→ More information on wiring up the interface module can be found in:
Interface module → chapter 5 from Page 55.
Telecontrol module → chapter 6 from Page 59.

		XIOC-SER	XIOC-TC1
Interfaces		RS232(C), RS422, RS485	
Protocols		Tranparent-Modus, MODBUS Master/ Slave, SUCOM-A, Suconet-K-Slave	Tranparent mode, Modbus Master/ Slave, SUCOM-A, DNP3 protocol
Character formats		8E1, 8O1, 8N1, 8N2, 7E2, 7O2, 7N2, 7E1	
Control and signal cables		RTS, CTS, DTR, DSR, DCD	
Transfer rate	Kbit/s	0.3 – 57.6	0.3 – 57.6
Suconet K		187.5, 375	-
Electrical isolation			
RS232		no	no
RS422/485		yes	yes
Number of slaves		–	–
Send data	Byte	≦ 250	≦ 250
Suconet K	Byte	≦ 120	–
Receive data	Byte	≦ 250	≦ 500
Suconet K	Byte	≦ 120	–
Bus termination resistors		Switchable for RS485, RS422	
Connector type			
RS232		9-pinSUB-Dplug connector	
RS422/485		Plug-in terminal block	
Current consumption	mA	< 275 mA	< 275 mA
Weight	kg	approx. 0.2	approx. 0.2
Number of modules			
XC100		2	–
XC200		4	4
Slots		any	any

DNP3 library in connection with XIOC-TC1

General data		
Profile		DNP3 Level 2
Send data	Byte	≦ 250
Receive data	Byte	≦ 282
Can be used for		XC200 control system
Max. quantity of modules		4 (together with XIOC-SER, XIOC-NET-SK-M)
Data buffer		
Binary input		1 - 1024, byte representation (incl. flags)
Analog inputs		1 - 1024, 16 bit + 1 byte flags
Counter input		1 - 1024, 32 bit + 1 byte flags
Binary output		1 - 1024, byte representation (incl. flags)
Analog outputs		1 - 1024, byte representation (incl. flags)

Suconet-K module (master)

Type	XIOC-NET-SK-M
Number of modules (COM interface)	
XC100	2
XC200	4
Max. internal current consumption	275 mA
Connection	
RS485	6 pole cage-clamp terminal block
Electrical isolation	Yes
Suconet-K (master) mode	
Interface type	RS485
Data transfer rates	187.5 or 375 kBit/s
Telegram	Suconet K/K1
Number of slaves	16
Slave addresses	2 to 31
Number of send bytes in a block	250 Byte
Number of received bytes in a block	250 Byte

PROFIBUS-DP module

→ More information concerning the PROFIBUS-DP module can be found in chapter 8 from Page 75.

Type	XIOC-NET-DP-M/S	
EMC	→ page 97	
Function	XIOC-NET-DP-M: PROFIBUS-DP interface, Master (class 1)	XIOC-NET-DP-S: Slave
Number of slaves	Max. 124 (30 without repeater)	
Send/receive data	for every 3.5 kByte for inputs and outputs	
Inputs/outputs	XIOC-NET-DP-M: Max. 244 bytes per slave	XIOC-NET-DP-S: Max. 244 Byte
Interface	RS485	
Connector type	Sub-D, 9 pole, socket	
Electrical isolation	Yes, for internal power supply	
Current consumption	300 mA	
Baud rate/length	Kbits/s	m
	9.6	1200
	19.2	1200
	93.75	1200
	187.5	1000
	500	400
	1500	200
	3000	100
	6000	100
	12000	100
Bus termination resistors	Switch-in	
Bus diagnostics	LED	
Number of modules	XC100: 1, XC200: 3	
Slots → table 20, table 21	1, 2, 3	

Index

A
Ambient temperature, enhanced . . . 12
Analog module parametric programming . . . 96
Analog modules, overview . . . 11
Arrangement of the modules . . . 12
Assembly
Counter module . . . 33
Signal module . . . 12

B
Bus cycle time determination . . . 79
Bus expansion connector . . . 14
Bus expansion with XIOC-BP-EXT
Physical design . . . 14
Settings in the easySoft-CoDeSys . . . 23
Bus termination resistors
XIOC-NET-DP-M . . . 76
XIOC-NET-SK-M . . . 73
XIOC-SER . . . 55, 59

C
C terminals . . . 19
Cable with attached connector, for the counter module 37
Cable with plug, for the counter module . . . 20
Capacitive loads . . . 19
Change actual value . . . 39, 41
Clear Underflow flag . . . 45
Command processing for counter module . . . 43
Communication library for DNP3 protocol . . . 61
Comparison value (counter module)
Parameter setting . . . 39, 40
Read out . . . 44
Setting . . . 43
Configuration
Counter analog module . . . 53
Counter properties . . . 42
Digital inputs/outputs . . . 102
XIOC-NET-DP-M . . . 91
Configuration example, DP module . . . 91
Configuration, XIOC-NET-DP-S/M . . . 77
Connecting devices to the Y outputs (counter module) 38
Connecting signal cables . . . 22
Connecting the incremental encoder . . . 35
Connection
Connecting devices to the Y outputs of the counter module . . . 38
Incremental encoder on the counter module . . . 35
Connections, counter module . . . 49
Conversion tables, for Pt100/Pt1000 . . . 28, 29
Counter input (counter module) . . . 34
Enable . . . 44
Current consumption, module arrangement . . . 12
Cyclic data exchange, DP module . . . 78

D
Data evaluation, temperature . . . 27
Data exchange, DP module . . . 78
Data transfer, example for DP modules . . . 81
DC load . . . 19
Device number . . . 83
DIAGGETSTATE function block . . . 85
Diagnostics
DIAGGETSTATE (function block) . . . 85
EXTENDEDINFO (Array) . . . 85
GETBUSSTATE (Variable) . . . 84
Slaves in PROFIBUS-DP . . . 83
XIOC-SER . . . 58
Digital modules, overview . . . 11
Dimensions
Module rack . . . 24
Signal modules . . . 24
DNP3 communication model . . . 61
DNP3 data model . . . 61
DNP3 Protocol . . . 61

E
End value (counter module)
Read out . . . 44
Setting . . . 43
Engineering . . . 12
Example
Diagnostics in the master control (PROFIBUS-DP) . 91
Expansion backplane . . . 14
EXTENDEDINFO, Array . . . 85

F
Fault retrieval, for XIOC-4T-PT . . . 30
Filter for voltage-peak suppression . . . 19
Freewheel diodes . . . 19
Function block
xDPS_SendDiag . . . 88
Function code according to DNP3 level 2 . . . 72
Fuse, to prevent burning out the external wiring . . . 20

G
GETBUSSTATE, Variable . . . 84

I
Inductive load . . . 19
Input map, counter analog module . . . 50
Input/output status indication . . . 12
Interface
PROFIBUS-DP . . . 76
XIOC-NET-SK-M . . . 73
XIOC-SER . . . 55, 59
Interlock . . . 12

L
Latch output 39
Reset 44
LE4 96
LED changeover switch 12
LED display
Counter analog module 50
Counter module 33
XIOC-NET-SK-M 73
XIOC-SER 56, 60
Level output 39
Level-Ausgang 115
Linear counter 34, 39, 48

M
Maximum basic expansion 14
Maximum total expansion 14
Mode of operation, XIOC-SER
Suconet K (slave) 61
Transparent mode 61
Module arrangement 12
Module output (counter module)
Assign to the comparison value 1 or 2 43
Enable 43
Setting 39, 40
Module rack
Overview 11, 13
Slot assignment 13
Monotasking 79
Mounting
Module rack 14
Signal modules 17
Terminal block 17
Multitasking mode 80

O
Operating mode switch (counter module) 34
Operating mode, XIOC-SER
Suconet K (slave) 57
Transparent mode 57
Operation
DP module 77
Output map, counter analog module 52
Overflow flag (counter module) 39
Clear 45
Overload currents 19

P
Parametric programming of the LE4 with analog inputs/outputs 96
Periodic data exchange, DP module 79
Polarity 19
Preset value (counter module)
Read out 44
PROFIBUS-DP connector 76
PROFIBUS-DP module 75
Programming
Counter analog module 50
Counter module 33
Digital inputs/outputs 102
Pulse processing (example) 48

R
Read actual (current) value 44
Read out flags (counter module) 45
Receive data
XIOC-NET-SK-M 74
XIOC-SER 58, 61
Reference input (counter module) 34, 41
Enable 44
Relay contacts, operating life 19
Repeater 75
RESET button (counter module) 33
Reset Equal flag (EQ) 44
Resistance thermometer 25
Response time, PROFIBUS-DP 80
Ring counter 34, 40, 48

S
S terminals 19
Send data
XIOC-NET-SK-M 74
XIOC-SER 58, 61
Set new actual value 44
Setpoint value (counter module)
Setting 44
Shielding, signal cables 22
Signal modules
Assembly 12
Max. number per CPU 13
Overview 11
Start value (Counter module)
Read out 44
Setting 43
Station byte 84
Status display
Counter analog module 50
Status display (counter module) 47
Status indication, PROFIBUS-DP slave 81
Suconet-K mode, XIOC-SER 55
Supply voltage
for relay operation 19
I/O electronics 13
Signal modules 13
Switching operations at high frequency 19

T
Target Rotation Time 79
Task control in online operation 80
Technical data
Analog input modules 104
Analog input/output modules 107
Analog output module 105
Counter analog module 112
Counter module 111
Digital input modules 98
PROFIBUS-DP module 113
Relay output module 101
Serial interface module 113
Suconet-K module (master) 114
Temperature acquisition module 109
Transistor output modules 100
Temperature setting (XIOC-4T-PT) 25
Temperature/measurement diagram 28
Terminal block 12
Mounting 17
Terminal capacity, terminal block 18
Transparent mode, XIOC-SER 55

V
Voltage peaks (filter) 19

W
Wiring
Analog modules 21
Counter module 38
Digital input module 18
Input module XIOC-32DI, output module XIOC-32DO 20
Relay output module 19
Screw terminal block 18
Spring-loaded terminal block 18
Transistor output module 19
XIOC-4T-PT 26
With short-circuit protection 59

X
xDPS_SendDiag, function block 88

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP