Manual

02/18 MN050005EN

XC300
Modular control

Powering Business Worldwide

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

For customers in US/Canada contact:

EatonCare Customer Support Center

Call the EatonCare Support Center if you need assistance with placing an order, stock availability or proof of shipment, expediting an existing order, emergency shipments, product price information, returns other than warranty returns, and information on local distributors or sales offices.

Voice: 877-ETN-CARE (386-2273) (8:00 a.m. – 6:00 p.m. EST)
After-Hours Emergency: 800-543-7038 (6:00 p.m. – 8:00 a.m. EST)

Drives Technical Resource Center

Voice: 877-ETN-CARE (386-2273) option 2, option 6
(8:00 a.m. – 5:00 p.m. Central Time U.S. [UTC-6])
email: TRCDrives@Eaton.com
www.eaton.com/drives

Original operating manual
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 operating manual.

1. Edition 2018, publication date 02/18

Author: Thomas Hettwer
Redaction: Antje Panten-Nonnen

Subject to alteration.

Danger!
Dangerous electrical voltage!

Before commencing the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally retriggered.
- Verify isolation from the supply.
- Ground and short-circuit.
- Cover or enclose neighbouring units that are live.
- Follow the engineering instructions (IL) 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 equalizing. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference do 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 connection so that a cable or wire breakage on the signal side does not result in undefined states in the automation device.
- Ensure a reliable electrical isolation of the low voltage for the 24 V supply. Only use power supply units complying with IEC 60364-4-41 or HD 384.4.41 S2 (VDE 0100 part 410).
- Deviations of the mains voltage from the nominal value must not exceed the tolerance limits given in the technical data, 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 switching off devices must not cause restart.
- Built-in devices for enclosures or cabinets must only be run and operated in an installed state, desk-top devices or portable devices only when the housing is closed.
- 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 switching off 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.).
- During operation, and depending on their degree of protection, variable frequency drives may have live, uninsulated, moving, and/or rotating parts, as well as hot surfaces.
- The impermissible removal of the required cover, improper installation or incorrect operation of the motor or variable frequency drive can cause the failure of the device and serious injury and/or material damage.
- Comply with all applicable national accident prevention regulations (e.g. BGV A3) when working with energized variable frequency drives.
- The electrical installation must be carried out in accordance with the relevant regulations (e.g. with regard to cable cross sections, fuses, PE).
- All transport, installation, commissioning and maintenance work must only be carried out by trained personnel (observe IEC 60364, HD 384 or DIN VDE 0100 and national accident prevention regulations).
- If applicable, systems in which variable frequency drives are installed must be equipped with additional monitoring and protective devices in accordance with the applicable safety regulations, e.g., the German Equipment and Product Safety Act, accident prevention regulations, etc. Making changes to the variable frequency drives by using the operating software is allowed.
- Keep all covers and doors closed during operation.
- When designing the machine, the user must incorporate mechanisms and measures that limit the consequences of a drive controller malfunction or failure (an increase in motor speed or the motor’s sudden stop) so as to prevent hazards to people and property, e.g.:
  - – Additional stand-alone devices for monitoring parameters that are relevant to safety (speed, travel, end positions, etc.)
  - – Electrical and non-electrical safety devices (interlocks or mechanical locks) for mechanisms that protect the entire system
  - – Due to the possibility of there being capacitors that are still holding a charge, do not touch live device parts or terminals immediately after disconnecting the variable frequency drives from the supply voltage. Heed the corresponding labels on the variable frequency drives

Content

0 About This Manual

This manual contains information about the XC300 Modular control, types:

- XC-303-C32-002
- XC-303-C21-001
- XC-303-C11-000

→ **Shorter name XC300**
Where this manual refers to all types, the shorter name XC300 is used.
If only one type is meant specifically, this is explicitly stated.

Work with the up-to-date documentation for the XC300 Modular control. The current edition of this documentation and additional literature can be found online, → section 0.3, "Additional documentation", page 8.

→ For up-to-date information about the XC300 Modular control refer to the product pages:
www.eaton.eu/xc300
and the EATON online catalog.
http://www.eaton.eu/ecat

Entering "XC300" in the search field will take you directly to this product group.

0.1 Target group

This manual is intended for electricians and automation engineers, as well as for the individuals who are in charge of performing the electrical installation and those who will be using the XC300 Modular control as an integrated operating and control device in their own applications.

Specialist knowledge of electrical engineering and physics is required for commissioning, as is well-founded knowledge of the communication interfaces and procedures used.

CAUTION

Installation requires qualified electrician

0.2 Change protocol

The following table lists major amendments.

Publication date	Page	Keyword	new	modified	deleted
02/18		First edition	–	–	–

0.3 Additional documentation

More information on the topic can be found

in the manual:

- "XN300 slice modules"– MN050002
- "CANopen Gateway XN-312-GW-CAN"– MN050003
- Manufacturer's documentation on the XSOFT-CODESYS-3, online help for the individual libraries
- Documentation on the CAN subscribers used
- PLC programming with CODESYS manual, Chapter "Online functions."

in the installation leaflet:

- IL050018ZU, XC-303-...

→ The documents mentioned in this manual, as well as this manual itself, can also be downloaded free of charge from the internet in PDF format at:
http://www.eaton.eu/doc
www.eaton.eu/DE/→ Customer Support →
Download Center – Documentation

Enter the document number ("05005," for example) or "XC300" into the **Quick Search** text field.

0.4 Legal Disclaimer

All the information in this manual has been prepared to the best of our knowledge and in accordance with the state of the art. However, this does not exclude the possibility of there being errors or inaccuracies. We assume no liability for the correctness and completeness of this information.
This information in particular does not contain any guarantee for the provision of specific properties.

Do not use the XC300 Modular control before you have read and understood this manual.

It is assumed that the user of this manual is thoroughly familiar with the information found in the manuals for incorporating the XC300 into automation processes.

Hazards posed by the XC300 cannot be eliminated if the safety instructions are not observed – especially if the XC300 Modular control is commissioned and maintained by insufficiently qualified personnel and/or the XC300 is used improperly. Eaton assumes no liability for any damages resulting from cases such as these.

0.5 I/O system XN300

The ultra-compact XN300 slice card modular I/O system, which features a plug-in connection system, complements the XC300 Modular control with application-oriented functions to achieve an optimized system solution.

There are XN300 slice card modules available for power distribution, power supply, digital and analog input/output modules, and technology modules.

→ For up-to-date information about the I/O system XN300, refer to the product pages:
www.eaton.eu/xn300
and for the available product range, refer to the EATON online catalog.
http://www.eaton.eu/ecat

Entering "XN300" in the search field will take you directly to this product group.
All modules are described in the "XN300 slice modules" manual.

0.6 Writing Conventions

Symbols with the following meaning are used in this manual:

▶ indicates actions to be taken.

Hazard warnings of material damages

ATTENTION
Warns about the possibility of material damage.

Hazard warnings of personal injury

⚠ **CAUTION**
Warns of the possibility of hazardous situations that may possibly cause slight injury.

⚠ **WARNING**
Warns of the possibility of hazardous situations that could result in serious injury or even death.

⚠ **DANGER**
Warns of hazardous situations that result in serious injury or death.

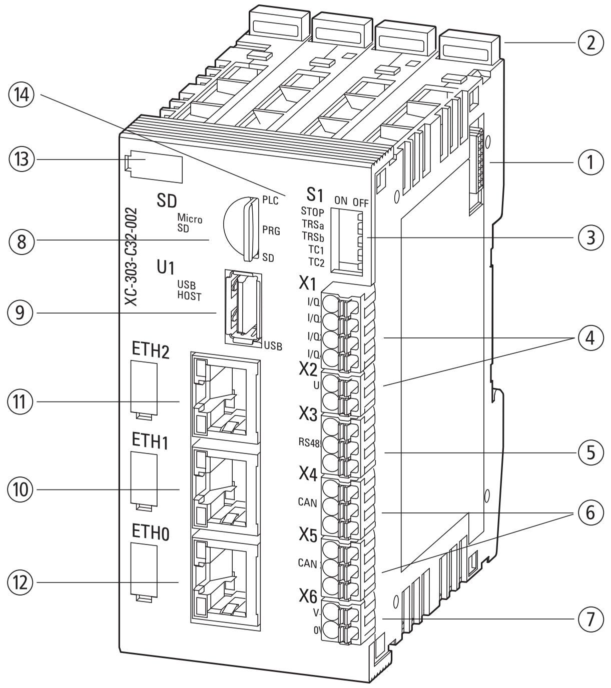
Notes

→ Indicates useful tips.

▶ Indicates actions to be taken.

1 Description

1.1 System overview



Assignment based on device type

XC-303-S1		C32	C21	C1
STOP	PLC	√	√	√
TRSa, TRSb	RS458	√	√	–
TC1	CAN1	√	√	–
TC2	CAN2	√	–	√

Equipment based on device type

XC-303-		C32	C21	C1
X1, X2	I/Q	√	–	–
X3	RS485	√	√	–
X4	CAN1	√	√	–
X5	CAN2	√	–	√

XC-303-		C32	C21	C1
U1	USBHost	√	√	–
ETH2		√	√	–
ETH1		√	√	√
ETH0		√	–	–

Figure 1: Structure of the XC300

① Bus connector to the I/O system XN300
for expanding the local I/O level with up to 32 XN300 I/O slice cards
② Locking tabs for attachment to the DIN rail
③ S1 (assignment based on device type)
DIP switch for switching on the operating status and termination resistors

External, push-in connection level (equipment based on device type)
④ X1 four interruptible input/output channels with
X2 related supply voltage I/Q1, …, I/Q4
⑤ X3 RS485 interface
⑥ X4, X5 two CAN interfaces for connecting standard CAN components
⑦ X6 supply voltage

External connections
⑧ SD slot for MicroSD memory card
⑨ U1 USB-Host for saving process data and carrying out program updates

Ethernet interface (equipment based on device type)
⑩ ETH2: 10/100/1000 MBit, e.g., SCADA
⑪ ETH1: 10/100 MBit, e.g., machine network
⑫ ETH0: 10/100 MBit, e.g., cell network, I/O network

Displays
⑬ Marker areas for individual labels
⑭ Status indicator LEDs under the front panel

The XC300 Modular control is Eaton's powerful and flexible control system, allowing machine and plant builders to achieve a slim and modern automation concept in combination with the compact I/O system XN300 and the innovative Multi-touch display XV300.

Thanks to the large number of interfaces, the PLC is suitable for use as a universal and flexible data node for a wide range of applications.
The integrated OPC server allows for standardized data transfer in machine-to-machine (M2M) communications and the web server supports visualization in HTML5 format.

Up to three Ethernet interfaces with individual IP addresses facilitate network segmentation, whereby access is protected in accordance with current standards in communication.

The XC300 can therefore be seamlessly integrated into modern automation architectures in line with Industry 4.0.

The local functional level of the XC300 can be flexibly scaled to include all modules of the I/O system XN300. The result: control solutions in the most compact of spaces with a high number of connections.

The PLC consists of the:

- Controller with power supply adapter and local inputs/outputs → page 14
- I/O system XN300 → Separate manual "XN300 slice modules".

→ To program the XC300, the programming software XSOFT-CODESYS-3 version 3.5.9 or higher is required.

1.2 Communication and Interfaces – Decentralized Expansion

The XC300 Modular control supports decentralized expansion via fieldbus interfaces. The following protocols are supported for communication on the respective interface:

Protocol		Interfaces
CANopen	Master / Device	CAN1, CAN2
easyNet	Master / Device	CAN1, CAN2
Modbus RTU	Master / Device	RS485
EtherCAT	Master (1x)	ETH0, ETH1, ETH2 (fieldbus)
Modbus TCP	Master / Device	ETH0, ETH1, ETH2 (fieldbus)
Ethernet/IP	Master	ETH0, ETH1, ETH2 (fieldbus)
OPC-UA/SCADA	Server	ETH0, ETH1, ETH2 (SSL/TLS/IPSec)
WEB - VISU	HTML5 server	ETH0, ETH1, ETH2 (SSL/TLS/IPSec)
Ethernet	HTTP, HTTPS, DHCP (client), DNS, SFTP, SSL, TLS	ETH0, ETH1, ETH2

1.3 Intended Use

The XC300 Modular control is designed for use in machine and plant building as a compact, fanless PLC with extensive functions and interfaces that enable the implementation of modern, flexible automation concepts.

The XC300 is only approved for use in automation technology to control machines and systems in closed spaces.

The XC300 Modular control must not be used to implement safety functions (in the sense of protecting people and machinery).
Observe the labels on the housing of the XC300 and the approval information and technical data.

Any other use must be discussed and agreed upon with the manufacturer in advance.

→ section 19.2, "Approvals and National Approvals for Devices XC300", page 144

→ section 19.3, "Technical Data", page 145

The XC300 Modular control may only be operated if it has been correctly mounted and connected by qualified electrical specialists. The installation must conform to the regulations governing electromagnetic compatibility (EMC).

⚠ **DANGER**
DUE TO AUTOMATED START-UP

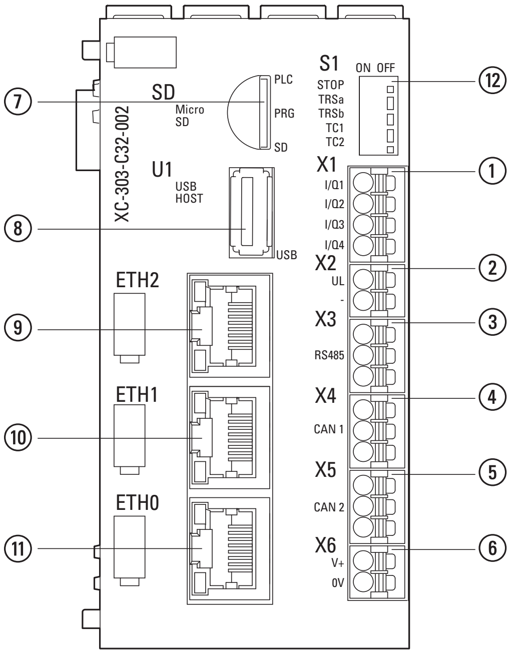
When the XC300 is switched on, the equipment being started up must not pose any hazards, such as the inadvertent start-up of motors or unexpected switch-on of voltage.

1.4 Device Overview

The following device types are available:

- XC-303-C32-002
- XC-303-C21-001
- XC-303-C11-000

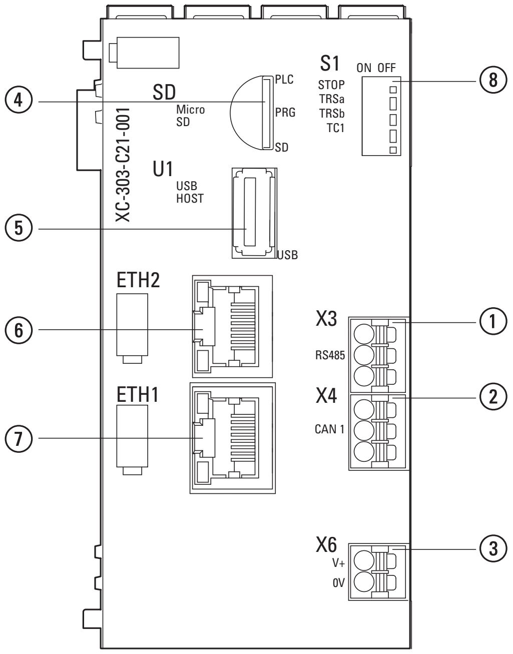
1.4.1 XC-303-C32-002



① Local inputs/outputs
② 24 V DC supply voltage for local inputs/outputs
③ RS485 with data cable A, data cable B, GND
④ CAN1 with CAN_H, CAN_L, GND
⑤ CAN2 with CAN_H, CAN_L, GND
⑥ 24 V DC POW supply voltage
⑦ Slot for MicroSD
⑧ USB-Host 2.0 host, type A
⑨ ETH2, 10/100/1000 MBit/s
⑩ ETH1, 10/100 MBit/s
⑪ ETH0, 10/100 MBit/s
⑫ DIP switch for switching on the operating status and termination resistors

Figure 2: XC-303-C32-002

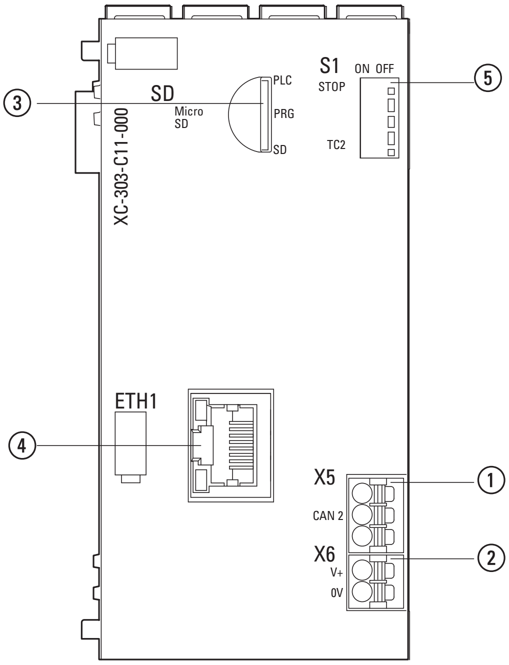
1.4.2 XC-303-C21-001



① RS485 with data cable A, data cable B, GND
② CAN1 with CAN_H, CAN_L, GND
③ 24 V DC POW supply voltage
④ Slot for MicroSD
⑤ USB-Host 2.0 host, type A
⑥ ETH2, 10/100/1000 MBit/s
⑦ ETH1, 10/100 MBit/s
⑧ DIP switch for switching on the operating status and termination resistors

Figure 3: XC-303-C21-001

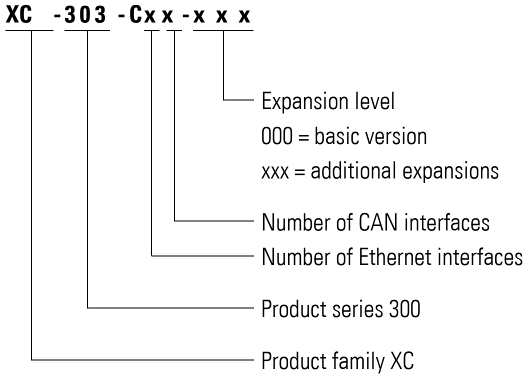
1.4.3 XC-303-C11-000



① CAN2 with CAN_H, CAN_L, GND
② 24 V DC POW supply voltage
③ Slot for MicroSD
④ ETH1, 10/100 MBit/s
⑤ DIP switch for switching on the operating status and termination resistors

Figure 4: XC-303-C11-000

1.5 Model code XC300



1.5.1 Device Version Number

The device version number is shown on the left side of the housing of each XC300 Modular control. These are the first two digits of the device number.

The device version number provides useful service information about the hardware version and the version of the operating system.

1.6 Functionality

1.6.1 Supply Voltage

A voltage supply is provided to supply power to the processor unit.

Terminal X6 (labeled V+/0V) has a 24 V connection for the processor unit and all of the slice card modules connected to the system bus.

If there is a voltage dip of the 24-V supply voltage (switching threshold is approx. 10 V) then a power-down logic switches off the 5-V-supply to the signal modules (central I/O).

1.6.2 Local Inputs/Outputs

Equipment based on device type for the XC-303-C32-002

The four digital X1 inputs/outputs can be used either as inputs or outputs and connected with sensors or actuators.

When used as outputs Q1 to Q4, they are designed for 24 V signals and have another separate shared supply voltage at X2 (labeled UL/–), which is galvanically isolated all the way to the bus.

Outputs Q1 to Q4 can be loaded with 500 mA, a duty factor (DF) of 100 %, and a utilization factor (g) of 100 %.

The outputs are short-circuit proof.

1.6.3 Processor Unit

Belonging to the processor unit are:

- Real-time clock (RTC) → page 18
- Memory → page 18
- CPU Drives → page 18
- Programming Interfaces → page 19

Real-time clock (RTC)

The XC300 is equipped with a real-time clock, which can be operated from the user program.

Possible functions are:

- Display mode for hours (12/24 hour display),
- Reading and setting of the real-time clock.

A description of the functions can be found in the manufacturer's documentation for XSOFT-CODESYS-3 on the SysLibRTC.lib library.

You can also set or query the real-time clock using the following PLC shell commands:

- setrtc (set the real-time clock)
- getrtc (query the real-time clock)

The real-time clock continues to run for up to 10 days even if the supply voltage fails, thanks to buffering with a memory backup capacitor.

Memory

The XC300 Modular control has a 128 MB program memory (FLASH), 512 MB working memory, and a 128 kB memory for storing non-volatile application data (NV-RAM).

CPU Drives

The XC300 has the following drives available:

- internal
  - System memory \home
- external (optional)
  - MicroSD memory card \home\disk_sd
  - USB-Device \home\disk_usb.

The boot application and the operating system are stored in the transaction-safe program memory in compressed format so they are safe even if a power failure occurs.
The boot application and relevant parts of the operating system are "unzipped" and copied to the working memory during operation.
The retentive data is stored in the NV-RAM.
Only the real-time clock (RTC) is buffered with a capacitor. This allows it to continue running without power for 10 days.

→ "Transaction-safe" means that if there is a voltage dip when a file is being processed, the file system and the opened file are generally not destroyed.
It is possible however, that data which you have written into the file last opened may be lost.

Figure 5 shows the interaction between the different memory systems/ drives of the XC300.

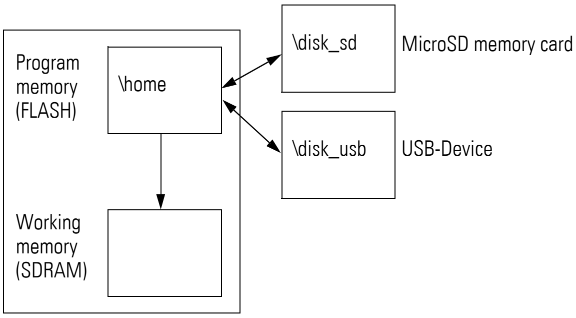


Figure 5: XC300 memory organization

1.6.4 Programming Interfaces

CAUTION
INTERFERENCE

The values specified in the technical data, as well as the device's electromagnetic compatibility (EMC), cannot be guaranteed if the following are used: unsuitable cables, improperly assembled and terminated cables, and/or wiring that does not conform to the applicable standards. Only use cables assembled and terminated by professionals.

The cables being used must be assembled and terminated as required by the port/interface description in this document.
The wiring instructions for the interface in question must be followed when wiring the device.
Generally applicable guidelines and standards must be observed.

CAUTION
NON-GALVANICALLY ISOLATED INTERFACES

The device may be damaged by potential differences.

▶ The GND terminals of all bus modules must be connected.

▶ Do not connect the interface to the device or disconnect it without first de-energizing the system.

1.6.4.1 MicroSD memory card, USB-Device

MicroSD memory card and USB-Device are used as external mass storage. You can load the recipe data, general data and the user program onto them. The operating system (OS) supports memory types with the FAT32 file system.

You can transfer the operating system to the MicroSD or the USB-Device in order to load it onto other controllers when updating the operating system → page 47.

ATTENTION

The file system of the MicroSD memory card and USB-Device is not transaction-safe.
Make sure that all the files of the program are closed before you insert or remove a MicroSD and/or USB-Device or turn off the voltage.

MicroSD memory card

The SD LED lights up green when a MicroSD is inserted.

→ A MicroSD memory card that is suitable for industrial data storage is recommended for use, e.g., article 191087, MicroSD memory card 2 GB with adapter.

USB-Device

Equipment based on device type for the XC-303-C32-002 and XC-303-C21-001

The XC300 Modular control provides the option through the USB-Host to connect USB peripheral devices supported by the hardware and operating system, e.g., a USB stick as an external memory.
The USB interface can also be used as a programming interface, for loading the user program or carrying out a system update.

CAUTION

When using commercially available peripheral devices with the USB-Host, it is important to keep in mind that their EMC interference immunity parameters may render them unsuitable for use in industrial environments.

CAUTION
NON-GALVANICALLY ISOLATED INTERFACES

The device may be damaged by potential differences.

▶ Do not connect the interface to the device or disconnect it without first de-energizing the system.

→ Note that the USB interface has a maximum load of 100 mA.

USB-Host

The USB interface is a USB 2.0 host interface.

Version: USB 2.0 host type A, downstream plug

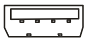


Figure 6: USB-Host

USB cable

- Only use standard USB 2.0 cables with a shield.
- Max. cable length: 5 m.

See also:
- Interface assignment → page 43
- Technical Data → page 145

1.6.4.2 Ethernet

Communication between the XC300 Modular control and the programming device is possible using any Ethernet interface ETH0, ETH1, or ETH2.

The Ethernet interfaces of the device provide different transfer rates:

Interface	Transfer rate MBit/s	XC-303-C32-002	XC-303-C21-001	XC-303-C11-000
ETH0	10, 100	✓	–	–
ETH1	10, 100	✓	✓	✓
ETH2	10, 100, 1.000	✓	✓	–

Use the Ethernet interfaces for programming, debugging, and for updating the controller operating system.

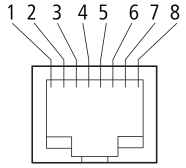
All Ethernet interfaces provided by the respective device can be used for programming in the same way.

Recommendation: Connect a PC via Ethernet

Note the following:

- Each Ethernet interface must be correctly configured.
- Two Ethernet interfaces cannot connect on the same subnet.

Version: RJ45 socket, 8-pin



Ethernet (Patch) Cable

Decisive factors when selecting the connection cable are the environmental conditions (interference, flexibility, data transfer rate) at the location where they are used.

- Only use standard Ethernet (patch) cables with RJ45 plugs, category Cat-5 or higher (IEC/EN 61131-2, EN 50178).
- Maximum cable length: 100 m
  If the network extends beyond that range, suitable infrastructure components must be used.
- Observe the installation guidelines for (Ethernet) cabling set out in ISO/IEC 11801 and EN 50173.

CAUTION
FORCES ON THE ETHERNET INTERFACE

Communications may be affected, and the connection’s mechanical components may be damaged, if the Ethernet interface is subjected to strong vibrations or the RJ45 plug-in connection is subjected to pulling.

- Protect the RJ45 plug-in connection from strong vibrations.
- Protect the RJ45 plug-in connection from tensile forces at the socket.

To establish communication between the XC300 Modular control and the programming device, follow the description for your connected device.

Default Addresses

The Ethernet interfaces are delivered with the following default IP addresses:

ETH2: 192.168.138.248
ETH1: 192.168.137.248
ETH0: 192.168.119.248

Subnet mask: 255.255.255.0

See also:
- Interface assignment → page 43
- Technical Data → page 145

1.6.4.3 RS485

Equipment based on device type for the XC-303-C32-002 and XC-303-C21-001

The serial RS485 interface (X3) can be used to connect an external display unit, for example.

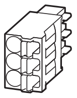
The RS485 port can connect to another controller as a Modbus RTU master or Modbus RTU device.

The RS485 interface is isolated from the system supply.

Max. possible cable lengths 100 m:

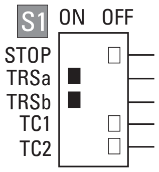
Configurable baud rate kBaud	Modbus RTU	Serial interface
1.2	x	–
2.4	x	–
4.8	x	x
9.6	x	x
19.2	x	x
38.4	x	x
57.6	x	x
115.2	x	x

Version: X3 plug connector, 3-pin, push-in spring-type terminal block



Termination Resistors

To connect the termination resistors for the RS485 cable, set the two DIP switches S1 TRSa and TRSb to the ON position.



RS485 Cable

- Only use the specified connection cable pre-assembled by an electrician.
- Max. cable length: 100 m.

See also:

- Interface assignment → page 43
- Technical Data → page 145

1.6.4.4 CAN/easyNet

Equipment based on device type

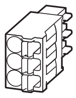
- XC-303-C32-002 CAN1, XC-303-C32-002 CAN2
- XC-303-C21-001 CAN1
- XC-303-C11-000 CAN2

Both CAN interfaces can be operated individually as CAN masters or CAN devices.

The CAN1 interface (X4) is galvanically isolated from the U_S (X6) supply, while the CAN2 interface (X5) is not.

The connections of the interfaces are the same. The CPU can be operated on the CAN bus both as a network (NMT) master and as an NMT device.
The CANopen and easyNet protocol can be run in parallel by the XC300.

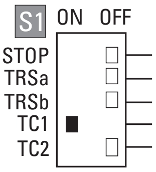
Version: X4 and/or X5 plug connector, 3-pin, push-in spring-type terminal block



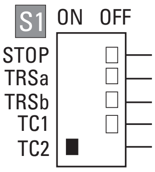
Bus Termination Resistors

Bus termination resistors must be used at the first and last subscriber on the CAN bus. A termination resistor of 120 Ω can be connected via the DIP switch for the purpose.

To connect the termination resistor for CAN1, set DIP switch S1 TC1 to the ON position.



For CAN2, set DIP switch TC2 to the ON position.



CAN/easyNet baud rate	Max. cable length
50 kBaud	1000 m[1),2)]
100 kBaud	600 m[1)]
125 kBaud	500 m[1)]
250 kBaud	250 m[1)]
500 kBaud	100 m
800 kBaud	50 m
1000 kBaud	25 m

1) From a cable length of 200 m, it is a good idea to use optocouplers in the CAN subscribers
2) From a bus length of approx. 1000 m, it may be necessary to use repeaters.

If an application attempts to activate a baud rate that is not supported, the CAN controller switches to passive mode.

Reaction of the subscriber on the CAN bus

Monitoring subscribers/bus: CAN telegrams are sent and received directly from the user program. An interruption on the CAN bus will only be detected if the respective CAN subscriber is monitored by the PLC (node-guarding function).

Start/Stop behavior: can be configured through the application.

- Reset to default value (usually 0).
- Current value is retained (user-specific)

Switch on voltage:

The order in which you switch on the power supply to the individual CAN subscribers has no effect on how the CAN-bus functions. Depending on the parameters set, the controller “waits” for subscribers that are not present or starts them when the subscriber is connected to the CAN net.

Communication with CAN subscribers:
Communication with the CAN subscribers and their configuration are described in the following documents:

- Manufacturer’s documentation XSOFT-CODESYS-3, online help library description: CANUser.lib/CANUser_Master.lib
- Fieldbus support -> CAN-based fieldbuses
- CANopen -> CANopen Manager (master)
- Documentation on the CAN subscribers to be used

Data exchange on the CAN bus is indicated by a green flashing LED next to the assigned X4 or X5 plug connectors. → figure 33, page 54

CAN Connection Cable

- Only use a connection cable that meets the requirements set out in ISO 11898.
- The maximum cable length is determined by the baud rate.

See also:
- Interface assignment → page 43
- Technical Data → page 145

2 Installation

CAUTION

Installation requires qualified electrician

DANGER OF ELECTRIC SHOCK!

All (un)installation work must be carried out with the entire system in a de-energized state.
Observe the safety rules set out in DIN VDE 0105:

- De-energize and isolate the system.
- Secure devices from inadvertently being switched on again.
- Ensure the installation is de-energized at all poles.
- Grounding and short-circuiting.
- Cover or enclose neighboring units that are live.

2.1 Control panel layout

The layout of the components inside the control panel is a major factor for achieving interference-free functioning of the plant or machinery.
Ensure that power and control components are separated in the layout during the planning, design, and implementation phases.
Power components include:

- Contactors
- Coupling/interfacing components,
- Transformers,
- Variable frequency drives,
- Current converters.

In order to effectively exclude any electromagnetic interference, it is advisable to divide the system into sections, according to their power and interference levels. In small control panels it is often enough to provide a sheet steel dividing wall, to reduce interference factors.

2.1.1 Ventilation

In the project engineering phase, ensure that there is sufficient ventilation for the XC300 Modular control.
In order to ensure that the maximum operating ambient temperature will not be exceeded, make sure that there is enough clearance between the system block's vents and any neighboring components, as well as between the vents and the control panel’s back plate.

Maintain a minimum clearance of 50 mm from passive components.
If the adjacent components are active elements (e.g., power supplies, transformers) a minimum clearance of 75 mm must be observed.

2.2 Mounting XC300

The XC300 is only mounted horizontally on a DIN rail (TS 35) in accordance with EN/IEC 60715.

Observe the minimum clearances for ventilation.

Expand the XC300 Modular control by the required XN300 slice modules into one system block before installation.

Ensure the following when installing the device in a control cabinet, installation distribution unit, or housing:

- The connections for the voltage supply and terminals are protected from direct contact during operation.
- The DIN rail establishes a conductive connection to the control panel’s back plate.
- The entire system block is mounted on the rail and secured by closing the locking tabs.

→ Detailed information on mounting the XN300 slice modules can be found in the “XN300 slice modules” manual.

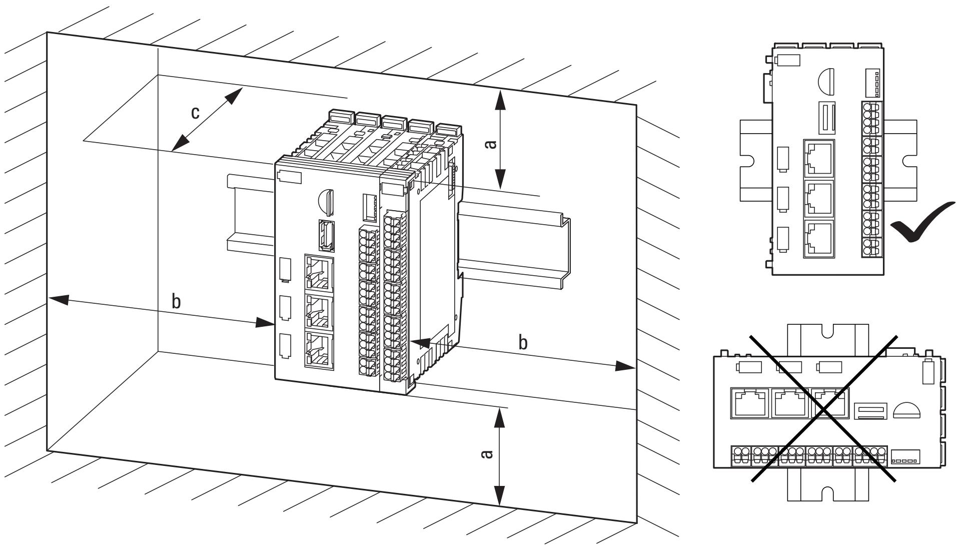


Figure 7: Mounting the XC300 as a system block

a	b	c	ϑ
30 mm (1.18”)	**30 mm** (1.18”)	**100 mm** (3.94”)	**≦ 55°C** (≦ 131°F)

To mount the system block, follow the steps below:

- ▶ The XC300 Modular control takes the first position on the left in the system block.
- ▶ Disengage the locking tabs on the sides of the XN300 slice modules by pulling on the front cover (blue).
  Ensure that all locking tabs (blue) are on the front so they catch the inserted slice card. The locking function of the front cover helps with this.

→ The front cover of the XC300 Modular control is secure and cannot be loosened.

- ▶ Insert one XN300 slice module each from the right in such a way that the locking tabs engage the guide.

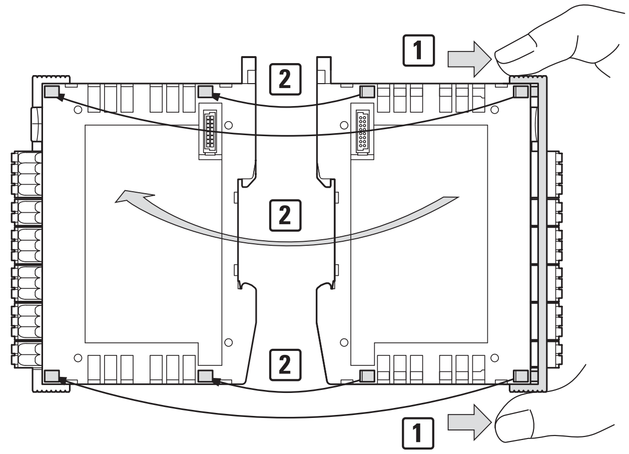


Figure 8: Assembling the XC300 and XN300 modules into one system block

- ▶ Grab the front cover from the top and bottom and push it back toward the XN300 slice module so that the modules lock solidly into place with each other.

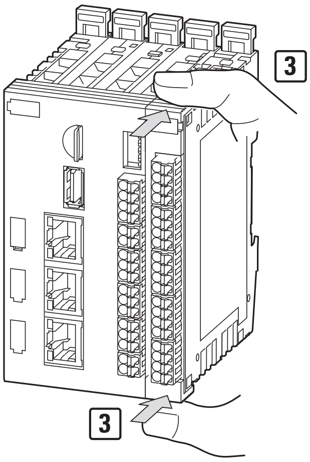


Figure 9: Locking the system block in place

- ▶ Repeat these steps until all XN300 slice modules are combined with the controller to form one system block.
- ▶ Pull the locking tabs up on the back of the XC300 Modular control and all XN300 slice moduless. Use a screwdriver as an aid.

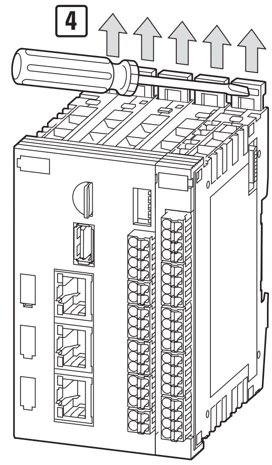


Figure 10: Securing the system block on the DIN-rail

- ▶ Tilt the system block forward and place it against the DIN-rail’s bottom edge in an inclined position.

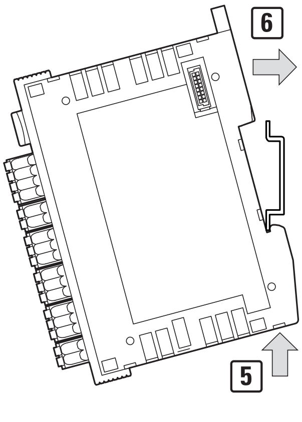


Figure 11: Placing the system block against the bottom edge of the DIN-rail

- ▶ Push the system block over the DIN-rail’s top edge.
- ▶ Secure the system block.
  Press down the locking tabs on the back of the XC300 Modular control and all XN300 slice moduless. Use a screwdriver as an aid.

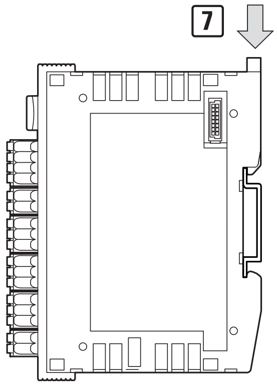


Figure 12: Locking the system block into place on the DIN-rail

- ▶ Check to make sure that the system block is solidly mounted.

2.3 Uninstalling and Disassembling the XC300

- ▶ De-energize the installation.
  Observe the safety rules set out in DIN VDE 0105:
- ▶ Use a screwdriver, for example, to press the release mechanism in order to disengage all contacts.
- ▶ Remove all **plug connectors from the** XC300 Modular control and the XN300 slice modules.

To disassemble the system block, proceed as follows:

- ▶ Pull the locking tabs up on the back of the XC300 and the XN300 slice modules. Use a screwdriver as an aid.

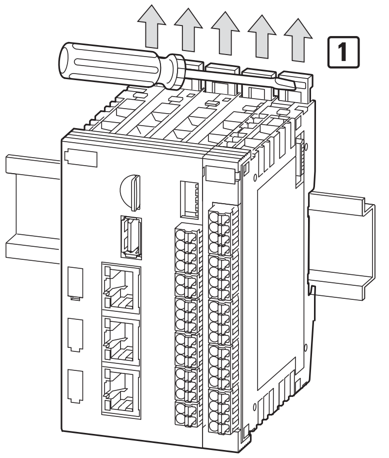


Figure 13: Loosening the system block fastening

- ▶ Tilt the system block forward, then pull the block, from its bottom edge, away from the DIN-rail.

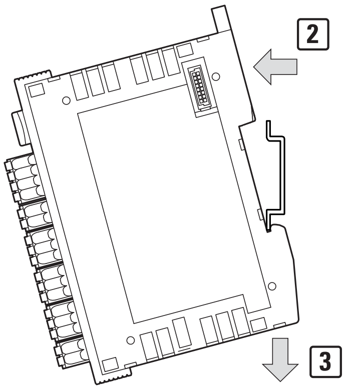


Figure 14: Placing the system block against the bottom edge of the DIN-rail

- ▶ Unlock the locking tabs between the modules by pulling on the front cover (blue).

→ The front cover’s stay-put function will indicate that the locking tabs have been unlocked. The front cover of the XC300 Modular control is secure and cannot be loosened.

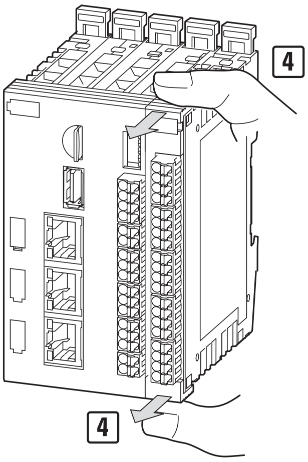


Figure 15: Unlocking the front cover

- ▶ When the locking tabs are unlocked, you can separate the modules from each other.

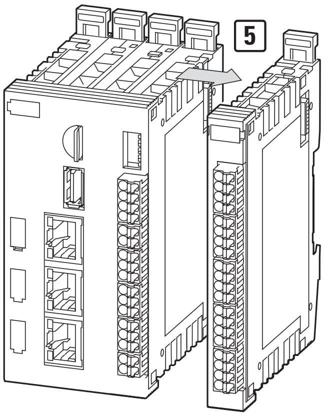


Figure 16: Disassembling the system block

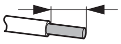
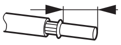
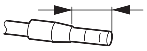
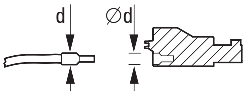
2.4 Terminals

Plug connector

X1 – Xn: The plug connectors with push-in spring-type terminals are supplied with the XC300 Modular control.
The cable, which is fitted with ferrules, is pushed into the corresponding contact and locked into place.

In order to release the conductor, simply press on the release mechanism, e.g., with a screwdriver, to pull out the cable from the corresponding contact.

Table 1: Connection specifications

Cable cross-sectional areas			**XC300**
10 mm (0.39") 	Solid	mm^2	0.2 - 1.5
10 mm (0.39") 	Flexible with uninsulated ferrule	mm^2	0.2 - 1.5
10 mm (0.39") 	Flexible with insulated ferrule	mm^2	0.2 - 0.75
	Ferrule d	mm	≦ 2.8
	AWG		24 – 16
	Stripping length	mm	10

2.5 Potential conditions between the components

The XC300 Modular control establishes a connection between the protective ground and the DIN rail via a contact point. All ground connections for the supply voltages are connected to the protective ground. One exception to this is the supply voltage for the local inputs and outputs U_L.

Common

- 0V
- ⏚

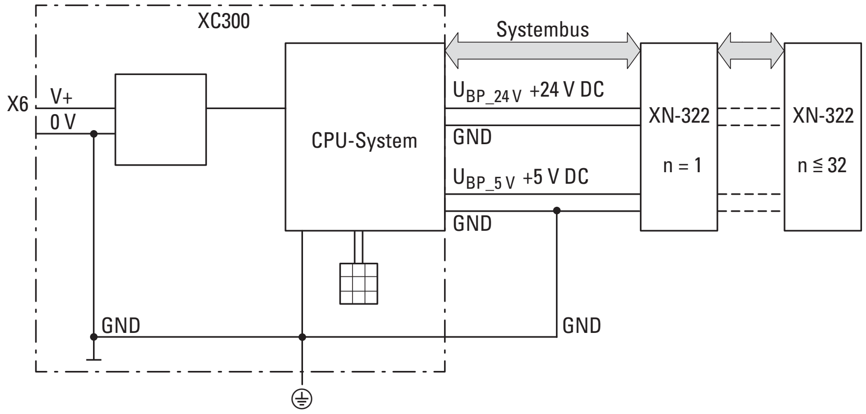


Figure 17: Potential relationships XC300 Modular control with I/O system XN300

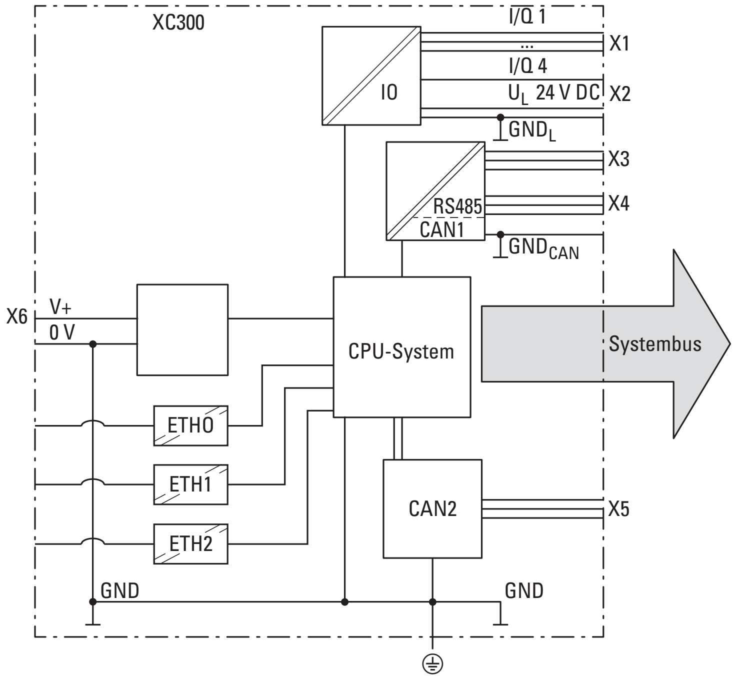


Figure 18: Potential Relationships XC300

The galvanic isolation of the Ethernet interfaces is listed in the appendix, see Chapter “Appendix”, → section “Galvanic isolation ETH0, ETH1, ETH2 from”, page 146.

2.6 Wiring in accordance with EMC requirements

Undesired faults can occur on the field bus and the analog inputs due to electromagnetic interference. This can be minimized beforehand by the implementation of suitable EMC-measures.
These include:

- EMC-conformant system configuration,
- Routing all analog input and field bus cables in a way that meets EMC requirements,
- Measures designed to reduce potential differences,
- The correct installation of the fieldbus system (cable, connection of the bus connectors, etc.),
- Using shielding

For the DIN rail

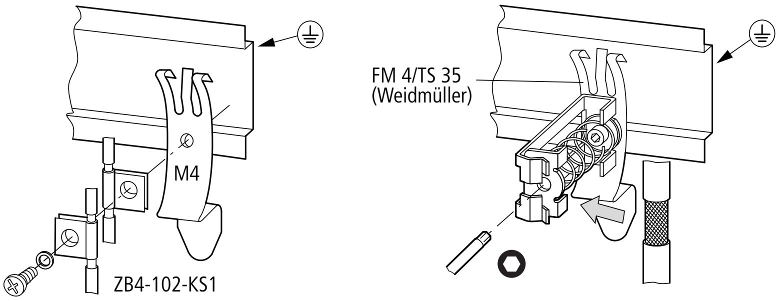


Figure 19: Field bus shielded by using a shield

The XC300 Modular control has a protective ground on the back.

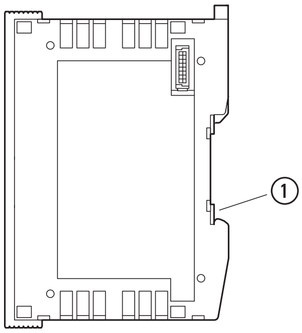


Figure 20: Side view XC300

① Functional earth

2.6.1 Cable routing and wiring

Cables are divided into the following categories:

- Power cables (e.g., cables that carry high currents, or cables to converters, contactors or solenoid valves),
- Control and signal cables (e.g., digital input cables),
- Measurement and signal cables (e.g., fieldbus cables),

→ Always route power cables, control cables, and signal cables as far apart as possible. This avoids capacitive and inductive coupling.
If separate cable routing is not possible, then the first priority must be to shield the cable responsible for the interference.

Take care to implement proper cable routing both inside and outside the control panel, to keep interference as low as possible:

▶ Avoid parallel routing of sections of cable in different power categories.
▶ As a basis rule, keep AC cable separated from DC cables.
▶ Keep to the following minimum clearance:
  - at least 10 cm between power cables and signal cables;
  - at least 30 cm between power cables and data or analog cables.

▶ When routing cables, make sure that the outgoing and return leads of a circuit pair are routed together. The opposing currents on this cable pair cause the sum of all currents to equal zero. The generated electromagnetic fields cancel each other out.

2.6.2 Suppressor circuit for interference sources

▶ Connect all suppressor circuits as close to the source of interference (contactors, relays, solenoids) as possible.

→ Switched inductors should always have suppressor circuitry fitted.

2.6.3 Shielding

▶ Use shielded cables for the connections to the data interfaces.

The general rule is:
the lower the coupling impedance, the better the shielding effect.

2.7 Lightning Protection

2.7.1 External lightning protection

All cables between buildings must be shielded. Metal conduits are recommended for use here. Fit signal cables with overvoltage protection, such as varistors or other surge arresters. Implement these measures ideally where the cable enters the building and at least at the control panel.

2.7.2 Internal lightning protection

Internal lightning protection covers all measures to reduce the effects of a lightning strike and the resulting electrical and magnetic fields on metallic installations and electrical plants. These measures are:

- Potential equalizing/grounding,
- Shielding,
- Using surge protective devices.

2.8 Connecting the power supply

DANGER

The supply voltage of the XC300 Modular control must be implemented as a SELV power adapter in safety-relevant applications.

The system bus is supplied via the XC300 Modular control with 5 V DC/1.6 A and 24 V DC/1.6 A. The supply voltage for system bus communication with the XN300 slice modules is provided via the 5 V on the system bus of the XC300 Modular control.

The system bus also provides a 24 V DC supply voltage for internal power to the XN300 slice modules. Modules with high energy requirements have an additional power supply. Some XN300 slice modules need an external supply of 24 V DC.

→ Information on this can be found in the "XN300 slice modules" manual.

The supply voltage is reverse polarity protected.

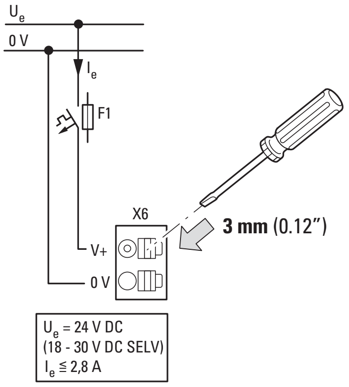


Figure 21: Connecting the power supply

2.9 Connecting Local Inputs/Outputs

The digital inputs/outputs at X1 are configured as inputs by default. This configuration can be changed in the corresponding dialog of the XSOFT-CODESYS-3, see → section 6.1 “Configuring Local Inputs/Outputs of the Device”, page 81.

2.9.1 Connecting the Supply Voltage for Local Outputs

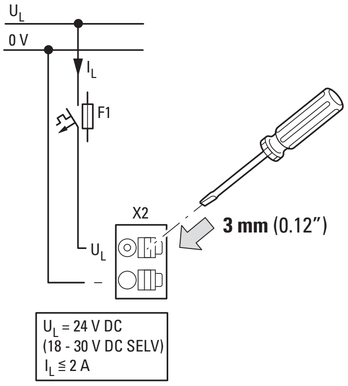


Figure 22: Supply voltage for outputs Q1 – Q4

The cable cross-section of the +24 V supply voltage for the inputs/outputs must be designed for the maximum total current of all outputs.

2.9.2 Connecting Outputs

Connect the supply voltage at X2 for outputs Q1 – Q4.

A maximum current of 0.5 A is permitted per output. The maximum total current for all four outputs is 2 A. The outputs are short-circuit-proof for 24 V DC and GND.

If a short circuit or overload occurs at one of the digital outputs Q1 – Q4, the outputs switch off. If the short circuit or overload is corrected, the output automatically switches back on.

One LED is assigned to each output. A switched (active) output is indicated by the LED lighting up green.

Connect the outputs to a connection cable measuring max. 30 m long without a shield.

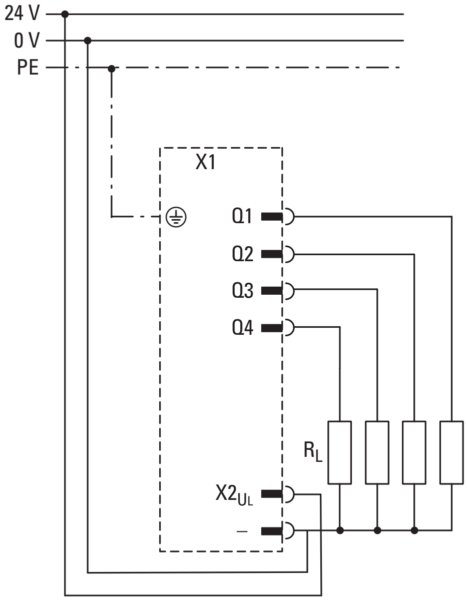


Figure 23: Connecting local outputs on the controller

→ Wiring examples can be found in the "XN300 Slice Modules" manual.

2.9.3 Suppressor circuit for inductive loads

High induced voltages may be produced when inductive loads are switched off.

In order to prevent system malfunctions caused by voltage peaks (e.g., coupling on analog cables), it is recommended to use a suppressor circuit (RC suppressors or flyback diodes) directly on inductive loads.

2.9.4 Connecting Inputs

XC300 has four inputs with a +24 V level for reading in signal statuses “0” and “1”. The modules feature input filters designed to suppress glitches on the corresponding signal cables.

Inputs I1 to I4 can be configured as interrupt inputs.

Connect the inputs to a shielded connection cable measuring max. 30 m long.

One LED is assigned to each input for the purposes of indicating the current status of the input in question.

The logical state of the process image does not affect the status indication. The status is indicated even if the controller is in RESET mode.

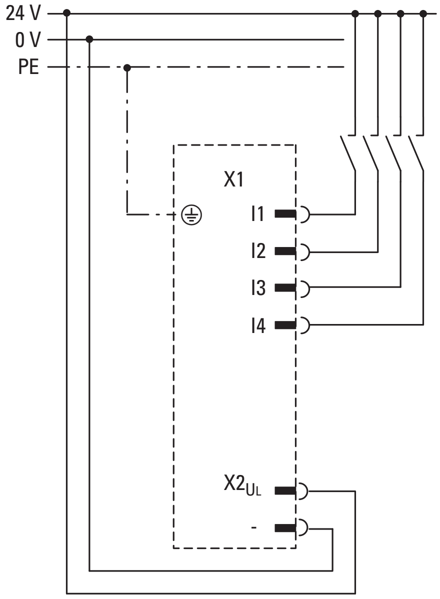


Figure 24: Connecting local digital inputs

2.9.5 Connecting Interrupt Inputs

Inputs I1 to I4 can be used as interrupt inputs. The interrupts can be triggered with a rising or falling edge or with an edge changeover.
The inputs must be configured for this in the parameter dialog of the XSOFT-CODESYS-3.

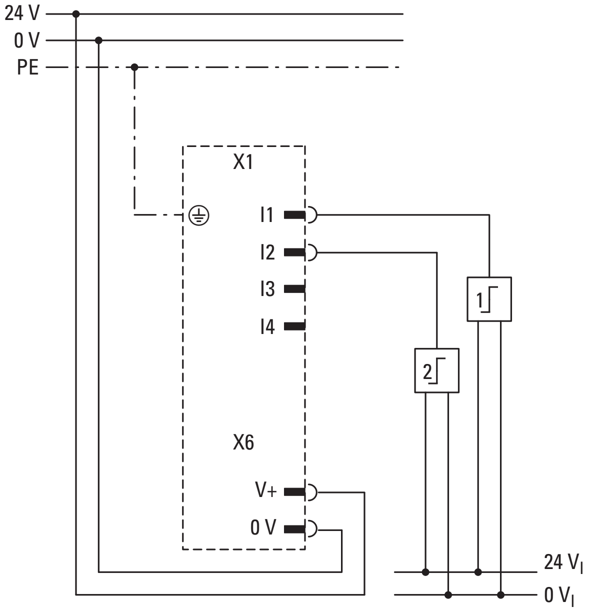


Figure 25: Interrupt input connections

The statuses at the terminals are registered in the controller with an input delay of < 300 µs.

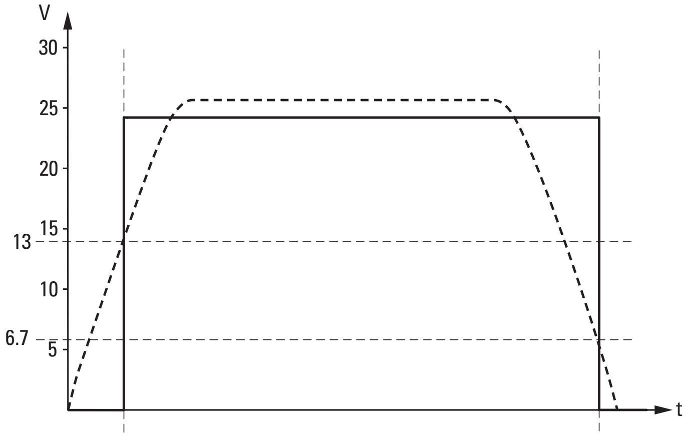


Figure 26: Switching thresholds for interrupts

See also:
- Time requirement at the interrupt inputs: Technical Data → page 145

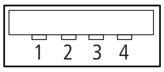
2.10 Interface assignment

2.10.1 USB host 2.0

→ Note that the USB interface has a maximum load of 100 mA.

The output voltage is protected against overload.

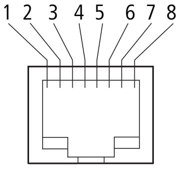
Table 2: USB interface assignment

USB host type A	PIN	Signal
	1	+5 V ⎓
	2	USB–
	3	USB+
	4	GND

See also:
- Functionality → page 17
- Technical Data → page 145

2.10.2 Ethernet

Table 3: Ethernet interface assignment

RJ45 socket	PIN	Ethernet 10/100 MBit	Ethernet 1000 MBit
	8	–	Bl–_D4
	7	–	Bl+_D4
	6	Rx–	Rx–_D2
	5	–	Bl–_D3
	4	–	Bl+_D3
	3	Rx+	Rx+_D2
	2	Tx–	Tx–_D1
	1	Tx+	Tx+_D1

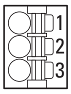
See also:
- Functionality → page 17
- Technical Data → page 145

2.10.3 RS485

Equipment based on device type for:

- XC-303-C32-002
- XC-303-C21-001

Table 4: RS485 interface assignment

Plug connector	X3	RS485
	1	Data cable A
	2	Data cable B
	3	GND RS485

Connecting the Termination Resistor

To connect the termination resistors for the RS485 cable, set the two DIP switches S1 TRSa and TRSb to the ON position.

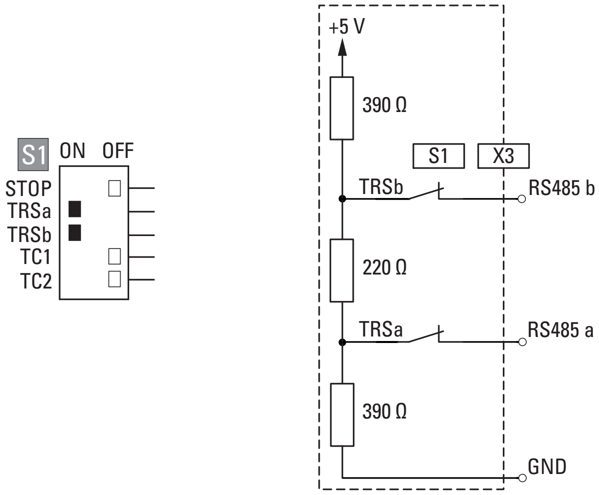


Figure 27: Connecting the termination resistors for the RS485 interface

See also:

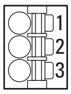
- Functionality → page 24
- Technical Data → page 145

2.10.4 CAN/easyNet

Equipment based on device type

- XC-303-C32-002 CAN1, XC-303-C32-002 CAN2
- XC-303-C21-001 CAN1
- XC-303-C11-000 CAN2

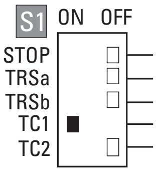
Table 5: CAN/easyNet interface assignment

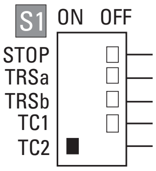
Plug connector	X4, X5	CAN	easyNet
	1	CAN_H	ECAN_H
	2	CAN_L	ECAN_L
	3	GND CAN	GND CAN

Connecting the Bus Termination Resistors

Bus termination resistors must be used at the first and last subscriber on the CAN bus. A termination resistor of 120 Ω can be connected via the DIP switch for the purpose.

To connect the termination resistor for CAN1, set DIP switch S1 TC1 to the ON position.





For CAN2, set DIP switch TC2 to the ON position.

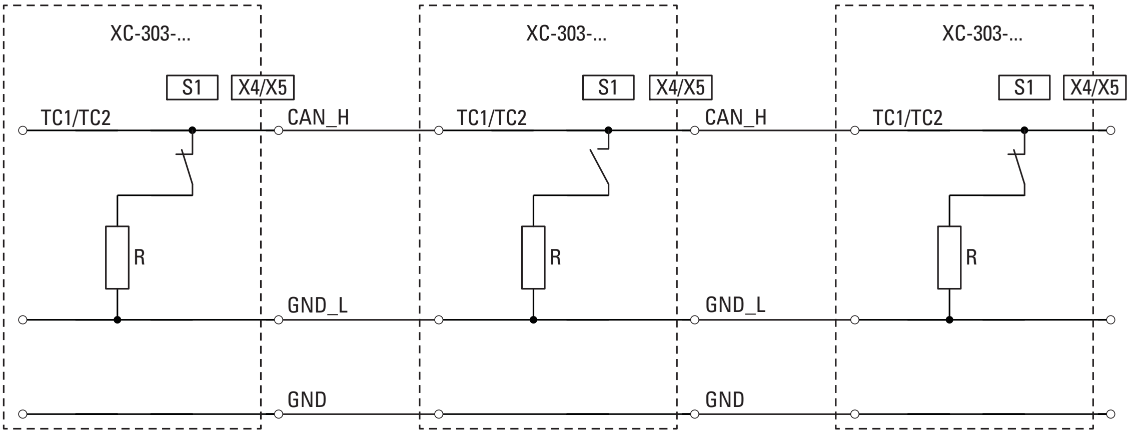


Figure 28: Connecting the termination resistors for the CAN interfaces (CAN1 or CAN2)

See also:

- Functionality → page 25
- Technical Data → page 145

3 Commissioning

The XC300 Modular control is commissioned in the system block together with the XN300 slice moduless.

→ Further information can be found in the “XN300 slice modules” manual.

3.1 General commissioning instructions

The signals received by analog modules are very small in comparison to digital signals. In order to ensure that these modules work properly, it is absolutely necessary to route all cables carefully:

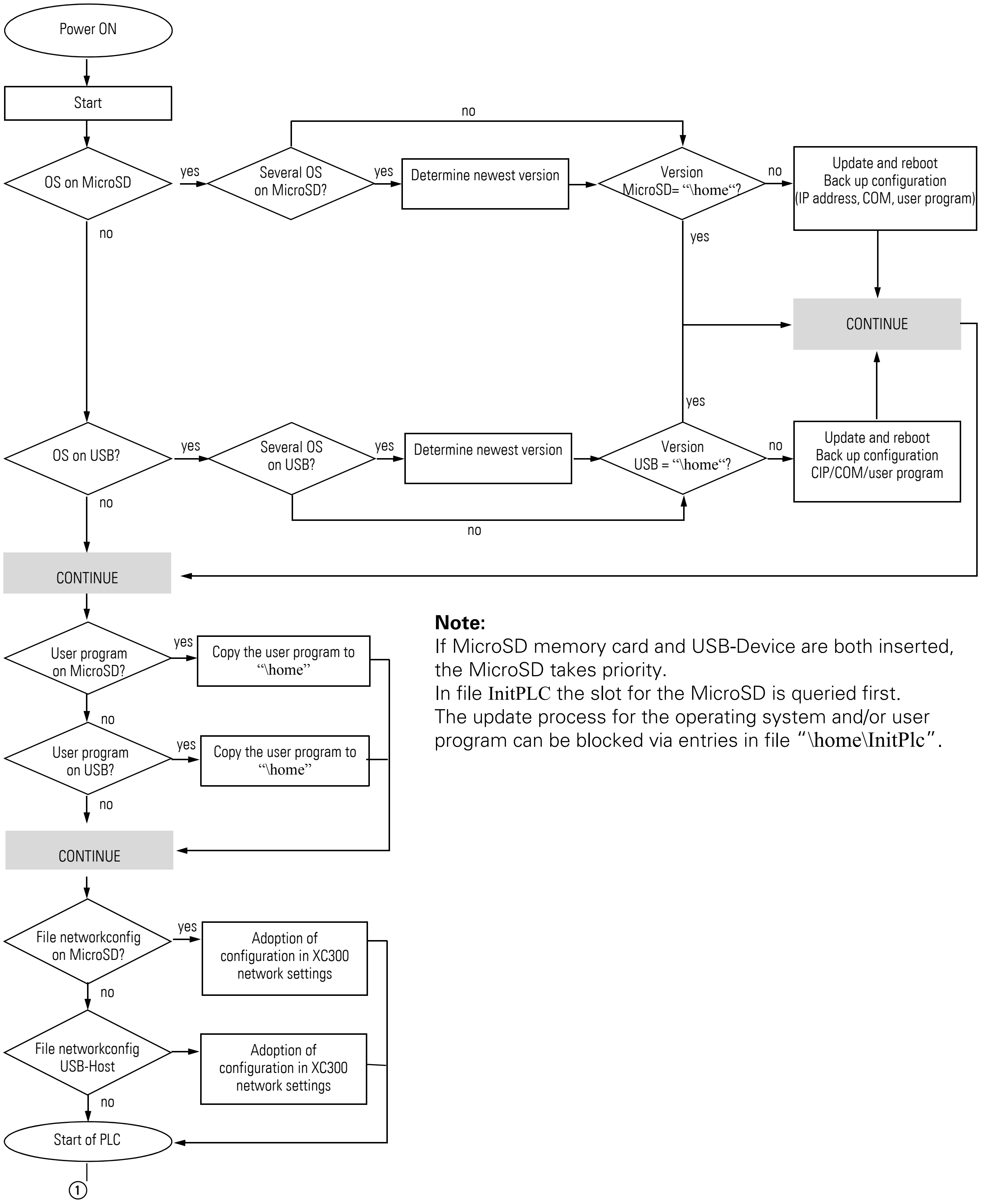
- The mounting rail (DIN rail) must have a proper ground connection.
- The cables used to connect to the analog signal sources must be as short as possible and must not be routed parallel to digital signal cables.
- Analog signal cables must be shielded.
- The shielding must be terminated at a shielding bus.
- Do not route the input cables parallel to load circuits.
- Suppressor circuit for all contactor coils (RC suppressors or flyback diodes)

3.2 Startup Behavior XC300

To update the operating system/user program, the drives are checked for the most up-to-date versions and then copied into the internal memory.

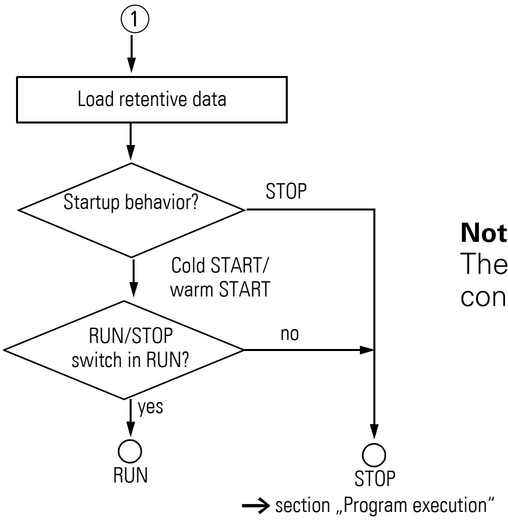
The following flow diagrams show which program is used. The diagrams also show the update process for the operating system (OS) using the MicroSD memory card or USB-Host.

Once the power is restored, only the boot application saved in the XC300 Modular control will be started.


Power ON
Start
OS on MicroSD
yes
no
Several OS on MicroSD?
yes
no
Determine newest version
Version MicroSD= “\home“?
no
yes
Update and reboot Back up configuration (IP address, COM, user program)
CONTINUE
yes
OS on USB?
yes
no
Several OS on USB?
yes
no
Determine newest version
Version USB = “\home“?
no
Update and reboot Back up configuration CIP/COM/user program
CONTINUE
User program on MicroSD?
yes
Copy the user program to “\home”
no
User program on USB?
yes
Copy the user program to “\home”
no
CONTINUE
File networkconfig on MicroSD?
yes
Adoption of configuration in XC300 network settings
no
File networkconfig USB-Host
Adoption of configuration in XC300 network settings
no
Start of PLC
1

Note:
If MicroSD memory card and USB-Device are both inserted, the MicroSD takes priority.
In file InitPLC the slot for the MicroSD is queried first.
The update process for the operating system and/or user program can be blocked via entries in file “\home\InitPlc”.

Figure 29: Startup sequence XC300 with MicroSD memory card and USB-Host



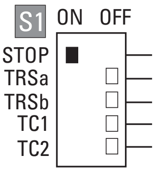
Note:
The startup behavior of the XC300 is configured in file “\home\InitPlc.”

3.3 Program execution

The program execution is influenced by the position of DIP switch 1.

3.3.1 DIP Switch S1

DIP switch STOP switches the CPU operating status between ON and OFF.



S1	ON	OFF
STOP	PLC stops	PLC runs
Termination		
TASa,TASb	X3 (RS485)	–
TC1	X4 (CAN1)	–
TC2	X5 (CAN2)	–

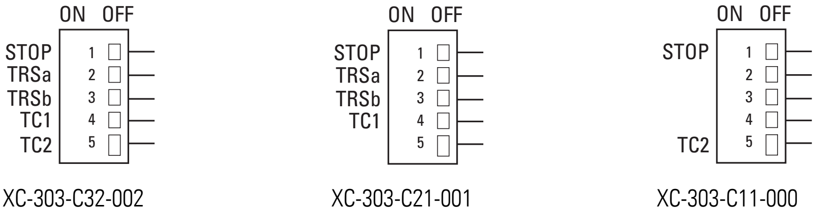


Figure 30: DIP switch S1 version (based on device type)

3.3.2 Program Start (STOP —> RUN)

You have the following possibilities to start the program:

	Program exists in main memory	Program should be loaded
Requirement	• CPU in STOP • STOP switch to ON	• CPU in STOP • STOP switch to OFF
Action	• Set STOP switch to OFF position	• Load program • In online operation, issue the "Start" command.
Result for all variables	CPU in RUN Values are retained at the start	CPU in RUN Initial values are activated.

3.3.3 Program Stop (RUN —> STOP)

A change of the STOP DIP switch to the ON position leads the CPU to the STOP state after completion of the program cycle (ending of all active tasks).

What should happen when the CPU is in the STOP state is defined in the XSOFT-CODESYS-3 Device tab, PLC settings.

Defaults:

- In the "Always update variables" field: "Disabled (only update when used in a task)"
- In the "Behavior of outputs in STOP" field: "Retain current values"

With these settings, the output values used in the I/O tasks are retained when the tasks are completed.

For test and simulation purposes, the "Always update variables" field should be set to "Enabled 1 (use bus cycle when not used in any task)."
This is the only setting that ensures that the display of all inputs and outputs is up to date during the simulation.

You can stop the program in one of two ways:

- In online operation, issue the STOP command.
- Set the STOP DIP switch to the ON position.

3.4 Power Off/Interruption of the Supply Voltage

When the program is running, powering off or interrupting the (CPU) power supply will cause the program cycle or all tasks to be aborted immediately. The data may no longer be consistent in some cases.

All outputs used in the I/O tasks are set to 0 or switched off. The behavior of retentive variables in shown in can be seen in → table 7.

The remaining program cycle will not be completed when power is reconnected!

If the consistency of the data is absolutely necessary for an application, other measures are required, such as the use of a uninterruptible power supply with battery backup. The PLC is started as shown in → figure 34 and → figure 29.

Table 6: Behavior of the variables after a program start

	Behavior of the variables after …	
Variable type	**Cold start**	**Warm start**
Non-retentive	Activation of initial values	Activation of initial values
Retain Persistent	Values remain in memory	Values remain in memory

CPU operating state display

The operating status of the CPU is indicated by the LEDs under the front PLC and PRG:

CPU status	**PLC LED**	
	Green	**Red**
RUN	On	Off
STOP	Flashing	Off
NOT READY	Off	On

If the runtime system starts up incorrectly, the status NOT READY is signaled by the PLC LEDs. The CPU remains in STOP state. The CPU can be restarted once the fault is corrected; see also → section 3.5, “LED Indicators”, page 54.

3.4.1 Testing and Commissioning (Debugging)

The XC300 Modular control supports the following testing and commissioning features:

- Breakpoint/Single step mode,
- Single cycle mode,
- Forcing,
- Online modification, → PLC programming with CODESYS manual, chapter “Online Functions.”

Breakpoint/single-step mode

Breakpoints can be set within the application program. If an instruction has a breakpoint attached, then the program will halt at this point. The following instructions can be executed in single-step mode. Task monitoring is deactivated.

ATTENTION

At this point, any set outputs will remain set.

Single cycle mode

In single-cycle operation, one program cycle is performed in real time.
The outputs are enabled during the cycle. At the end of the cycle, the output images are cancelled and the outputs are switched off. Task monitoring is active.

3.4.2 Forcing

All variables of the user program can be preset.
A local output is only forced if the corresponding variable is forced and the CPU is in the RUN state.

Status Indication

The inputs/outputs are to be referenced in order to visualize the states of the configured inputs/outputs in an interval controlled task in the PLC configurator. The following syntax is sufficient in the ST programming language in order to be able to display individual I/O bits.

Example:

```
%IB0;    (referencing of inputs I0.0 – I0.7)
%QB0;    (referencing of outputs Q0.0 – Q0.7)
```

in the application list (AWL):

```
LD       %IB0
ST       Default byte
LD       Default byte
ST       %QB0
```

3.4.3 Reset

There are three different types of Reset commands:

- Reset-Warm,
- Reset-Cold,
- Reset-Origin.

→ table 7: The commands also affect the state of the CPU: shows the commands to use for initializing a retentive variable range. The commands also affect the state of the CPU.

Table 7: Behavior of the program and variables after a reset

	Reset command		
Variable type	**Reset-Warm**	**Reset-Cold**	**Reset-Origin**[1]
Non-retentive	Activation of initial values	Activation of initial values	Activation of initial values
Retain Persistent	Values remain in memory	Values remain in memory	

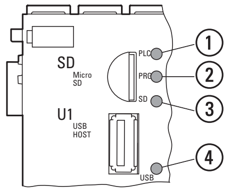
Program	paused;
	then restart
		paused;
		then restart
			deleted; controller in NOT READY status;
			PRG LED goes out;
			then reload program on the device and restart

[1] After “Reset-Origin” the program must be reloaded on the XC300. In online operation, the “Start” command can now be issued.

→ More information can be found in the XSOFT-CODESYS-3 online help using the search term “RETAIN.”

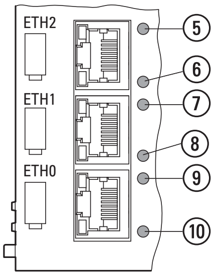
3.5 LED Indicators

The status indicator LEDs under the front panel indicate the signal status of the XC300 Modular control, the interfaces, and the inputs and outputs.



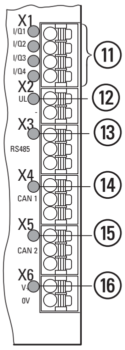
① LED PLC RuntimeSystem loaded (green)/
communication with programming system active (green flashing)
RUN (green)/STOP (green flashing)/ERROR (red)
② LED PRG (red)/No Error(green)
③ LED MicroSD found (green/–)
④ LED USB active (green/–)

Figure 31: Indicators in example XC-303-C32-002



⑤ LED Ethernet2 Activity (green)
⑥ LED Ethernet2 Activity ETH2 [10/100] Mbps (orange) /1000 Mbps (green)
⑦ LED Ethernet1 Activity (green)
⑧ LED Ethernet1 Activity ETH1 [10/100] Mbps (orange)
⑨ LED Ethernet0 Activity (green)
⑩ LED Ethernet0 Activity ETH0 [10/100] Mbps (orange)

Figure 32: Indicators in example XC-303-C32-002



⑪ LED Q1–Q4 (green)
⑫ LED 24 V found U_L (green)
⑬ LED RS485 connected (green/red)
⑭ LED CAN1 connected (green/red) flashes when data is being exchanged (Rx + Tx)
⑮ LED CAN2 connected (green/red) flashes when data is being exchanged (Rx + Tx)
⑯ LED 24 V DC found V+ (green)

Figure 33: Indicators in example XC-303-C32-002

Table 8: Status LED indicators

	Status LED		Description
	red	**green**	
PLC	–	–	RuntimeSystem not started • No power supply at X6
	–	flashing	Communication
	–	ON	PLC RuntimeSystem started
	ON	–	Error in RuntimeSystem • Flash error • Error in checksum • SSL connection could not be started Remedy: Inspect log file
	ON	ON	RuntimeSystem is started and has an error.
PRG	–	–	No application found on device
	–	ON	Application running
		flashing	Communication (green flashing)
	ON	flashing	Application is stopping, program has error e.g., because • Faulty access to memory • Task interval time > watchdog time
	–	flashing	Application is stopping
SD		–	No MicroSD memory card found
		ON	MicroSD memory card found
USB	–	–	No USB-Host found
	–	ON	USB-Host inserted
RS485	–	–	
	–	ON	Communication channel opened
	–	flashing	Bus activity
CAN1 CAN2	–	–	No bus activity
	–	flashing	Bus activity OK (CAN), FLASH for each telegram
	ON	–	Bus error
	flashing	flashing	WARNING
	orange	**green**	
ETH0, ETH1, ETH2	–	–	No active connection found
	–	ON	Active connection found
	flashing	ON	Active connection and data is being received or sent
Ethernet Activity	–	–	Ethernet communication/link status
	–	ON	
	ON	–	
	ON	ON	

	Status LED		Description
	–	Green	
Q1 – Q4	–	–	Input has LOW signal The logical state of the process image does not affect the status indication; the status is indicated even if the controller is in RESET mode.
	–	ON	Input has HIGH signal
POW		–	No power supply at X6
		ON	24 V DC power supply found for device at X6
U_L at Q1 – Q4		–	No power supply
		ON	Power supply present

1) flashing: The flashing frequency may differ based on the transfer rate on the bus.
When large amounts of data are being exchanged the flashing LED may look like it is constantly on.

3.6 Operating statuses when starting the XC300

The operating statuses are indicated by PLC and PRG.

3.7 Programs and project

3.7.1 Loading Programs on the Device

You must log on in order to load recently created or modified programs on the device. The question "Load the new program?" will appear. The load operation will start once this prompt has been confirmed.

→ Please note that the "Retain" variables are initialized during the load process, but the "PERSISTENT" variables retain their value.

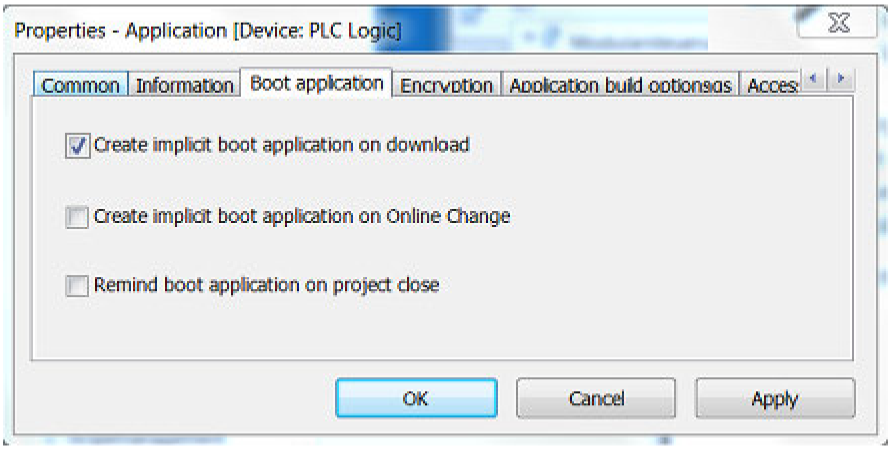
In order to safely store the program, a boot application must be generated by the user program. With the "Create boot application" command the program is loaded from the PC into the system memory and saved as a zero-voltage protected boot application.

The following steps are necessary in order to create a boot application:

▶ Change over to the "Online" folder.
▶ Select the "Login" command.
▶ Select the "Create boot application" command.

Alternatively, XSOFT-CODESYS-3 can be configured so that a boot application is automatically created when the user program is downloaded. To create a boot application automatically during download, proceed as follows:

▶ In the browser, select "Application."
▶ In the context menu, select the "Properties" command.
▶ Select the "Boot application" tab.
▶ Enable the option "Create implicit boot application on download" by activating the checkbox.


Properties - Application [Device: PLC Logic]

☑ Create implicit boot application on download

□ Create implicit boot application on Online Change

□ Remind boot application on project close

3.7.2 Storing and Deleting a Boot Application

Storing a Boot Application on MicroSD memory card

- ▶ Ensure that there is communication with the XC300 device.
- ▶ Ensure that the boot application has been converted without errors.
- ▶ Ensure that the MicroSD card is inserted into the device.
- ▶ Double-click on the device in the Explorer, e.g., "Device (XC-303)," go to the "Device" tab, and select "PLC Shell."
- ▶ Enter the command copyprojtosd in the command line.

A directory "XC303" with subfolder "project" is created for the boot application. The boot application is stored in this subfolder under the name <Application name>.app. For example: "Application.app"<Application name> refers to the application in the Explorer. A file <Application name>.crc is also created.

The boot application is stored on the MicroSD memory card in the specified folder structure.

You can use the PLC shell command copyprojtosd to copy the boot application (e.g., for a backup copy) or change the name of the file. In the XSOFT-CODESYS-3 software, however, only the boot application with the name <Application name>.app is active.

Storing a Boot Application on USB-Device

- ▶ Ensure that there is communication with the XC300 device.
- ▶ Ensure that the boot application has been converted without errors.
- ▶ Ensure that a USB-Device is inserted into the USB-Host.
- ▶ Double-click on the device in the configuration tree diagram, e.g., "Device (XC303)," go to the "Device" tab, and select "PLC Shell."
- ▶ Enter the command copyprojtousb in the command line.

A directory XC303 with subfolder project is created for the boot application. The boot application is stored in this subfolder under the name <Application name>.app. For example: "Application.app"<Application name> refers to the application in the Explorer. A file <Application name>.crc is also created.

The boot application is stored on the USB-Device in the specified folder structure.

You can use the PLC shell command copyprojtousb to copy the boot application (e.g., for a backup copy) or change the name of the file. In the XSOFT-CODESYS-3 software, however, only the boot application with the name <Application name>.app is active.

Deleting a Boot Application on MicroSD

Click on folder "Device → PLC Shell" and for the XC300, enter command remprojfromsd .

Deleting a Boot Application on USB-Device

Click on folder "Device → PLC Shell" and for the XC300, enter command remprojfromusb .

3.8 Updating the operating system

With the XC300 it is possible to replace the operating system with the latest version. Eaton provides the most up-to-date operating system version online: http://www.eaton.eu/XC300 → **Software**

→ If you transfer a current operating system to an older hardware version, it is possible that not all functions of the operating system will be supported by the hardware.

→ When an operating system is imported into the controller, the existing operating system will be overwritten.

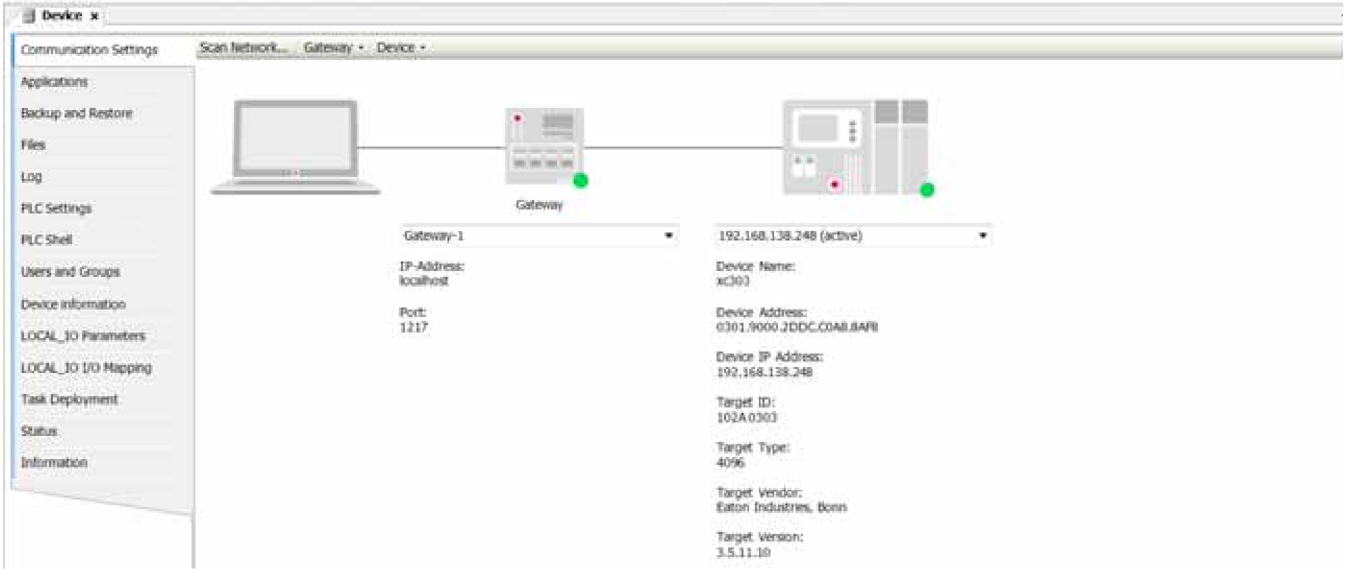
→ Note the information in file
XC303 Firmware_Release_Notes_DE.txt.

The operating system of the XC300 is updated within the XSOFT-CODESYS-3 programming system.

Procedure:

▶ Establish a connection via the Ethernet interface of the PC with the XC300.
→ chapter 5, "Establishing a Connection with a PC – XC300", page 77.

▶ In the XSOFT-CODESYS-3, go to the "Device" window and activate the "Firmware" tab.



Gateway-1

IP-Address:
localhost

Port:
1217

192.168.138.248 (active)

Device Name:
xc303

Device Address:
0301.9000.2DDC.C0A8.8AF8

Device IP Address:
192.168.138.248

Target ID:
102A 0303

Target Type:
4096

Target Vendor:
Eaton Industries, Bonn

Target Version:
3.5.11.10

Figure 34: Device" window

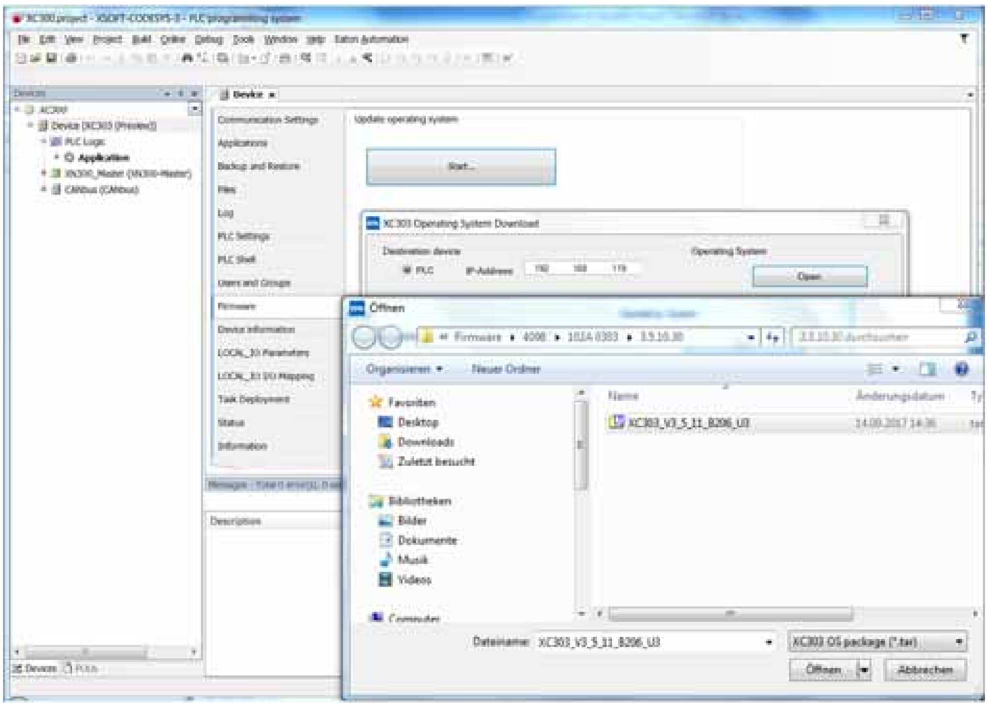


Figure 35: “FTP installation” option

- ▶ Enter the PLC’s IP address and click on “Next. ”
- ▶ Select the “XC303” device type and click on the “Next” button.
- ▶ Select the “PLC operating system” component and click on “Next.”
- ▶ Stop the PLC.
  Set DIP switch S1 Stop to the ON position.
- ▶ Click on the “Download” button to start downloading the firmware.
- ▶ Wait until the “Download successful” window appears.
  Then close the window with “RETURN. ”
- ▶ Click on the “Close” button to exit the Setup wizard.
- ▶ Start the PLC. Set DIP switch S1 Stop to the OFF position.
- ▶ Perform a restart for the XC300 Modular control.

The update process when using removable media or a local directory (→ figure 35, page 60) is similar to that for FTP installation.

Procedure:

- ▶ Insert the USB-Device or MicroSD memory card into the PC.
- ▶ Select „Removeable Drive“ in the window „XC303 OperatingSystem Download“ (→ figure 35).
- ▶ Select the removable media directory and click on “Next.”
- ▶ Select the “XC303” device type and click on the “Next” button.
- ▶ Select the “PLC operating system” component and click on “Next.”
- ▶ Click on “Install” to start downloading the firmware.
- ▶ Click on “Done” to exit the Setup Wizard.

The firmware files are now located on the removable media in the directory **XC303/Firmware**.

▶ Insert the USB-Device or the MicroSD memory card into the XC300.

The system will check whether there is a different operating system on the removable medium. If one is found, the system will start updating the firmware. You will recognize that this is happening when the status indicator LEDs under the front panel, PLC and PRG (green), start to flash in alternating sequence along with SF (red).

→ The new firmware version will not be enabled until the PLC is restarted.

3.9 Programming in Series Production via MicroSD or USB

The operating system and application can be programmed via MicroSD memory card or USB-Device.

→ section 3.2, “Startup Behavior XC300”, page 47

4 Configuring the Controller

4.1 Task Configuration

Project processing can be controlled via tasks. Each task can be assigned a range of programs which should be run during execution of the task.

The task is defined by a name, a priority and a type which defines under which conditions a task starts. Task condition and priority determine the sequence in which the tasks are to be processed.

You can set “Cyclic” or “Event-triggered” as the task condition. A cyclical task is restarted after the set interval time has elapsed. An event-triggered task is only started when the event occurs. You also have the option of connecting system events, such as “Start,” “Stop,” or “Reset,” to the execution of a program.

The task priorities can be configured with a value from 0 to 31 where 0 is the highest priority and 31 is the lowest priority.

In principle the output image is written onto the physical outputs before every task is called and the input image is read (updating of the input/output map). The task is executed thereafter. In addition, all system activities are carried out before or after the task call. This includes, for example, communication with the XSOFT-CODESYS-3 or online changes.

Updating of the input/output image by multiple tasks is described in the → section 4.3, “Multitasking”, page 67.

All IEC tasks, including those with the highest priority can be interrupted by an interrupt or an event controlled task.

Time monitoring (Watchdog) can be activated for each task.

→ For a detailed task configuration description, please refer to the manual for programming PLCs with XSOFT-CODESYS-3

At the end the control specific settings are explained on the basis of an example.

Creating a Task (Example)

First create the cyclic task “Basic” with the assigned program “Basic_prog”. Then you can add the event controlled task “Param” with the program “Param_prog”. In the program “Basic_prog” an event is programmed which invokes the “Param” task.

The following steps are necessary in order to create a task:

- Add a task
- Define the program call
- Create the program

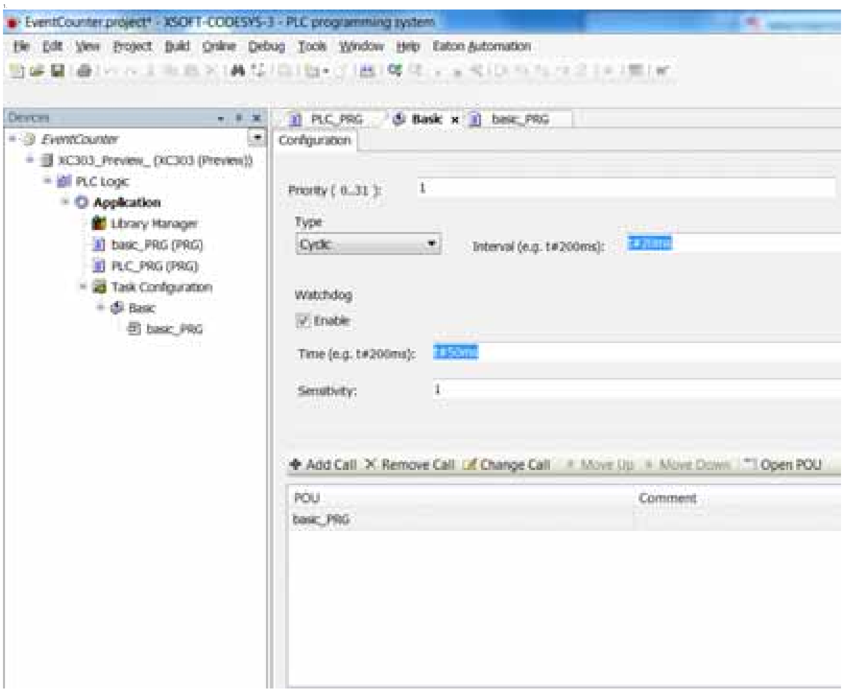
4.1.1 Creating a Cyclic Task

- ▶ In the configuration tree diagram for the project, select the device “Device (XC303)” -> “PLC Logic” -> “Application.”
- ▶ In the configuration tree diagram, select “Task Configuration.”
- ▶ In the context menu, select the command “Add Object” and “Task.”
- ▶ Enter in the Name field a name such as “Basic.”
- ▶ Set the task in the dialog as in figure 36.

Define the program call

With the program call you define which program is to be called with the “Basic” task.

- ▶ Right-click on the icon for the previously created “Basic” task and in the context menu, select the command “Add Object” -> “Program Call.”
- ▶ Enter the name “basic_PRG” in the “Add Call” window.
- ▶ Confirm the program name by selecting “Add.”



EventCounter.project* - XSOFT-CODESYS-3 - PLC programming system

Priority (0..31): 1

Type: Cyclic

Interval (e.g. t#200ms): t#20ms

Watchdog

☑ Enable

Time (e.g. t#200ms): t#50ms

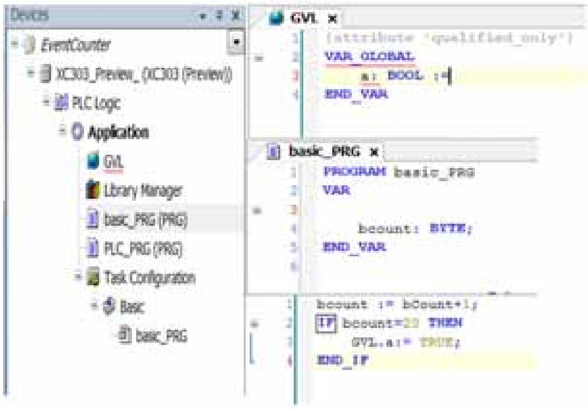
Sensitivity: 1

POU	Comment
basic_PRG	

Figure 36: Configuring the cyclic task with program call

Writing a program

- ▶ Open the Editor for the “basic_PRG” program element by double-clicking it.
- ▶ You can now enter a program. In the program example, the “bCount” variable is incremented.
  When the counter reaches 20 the global variable a = TRUE.



```
{attribute 'qualified_only'}
VAR_GLOBAL
    a: BOOL :=
END_VAR
```

```
PROGRAM basic_PRG
VAR

    bcount: BYTE;
END_VAR
```

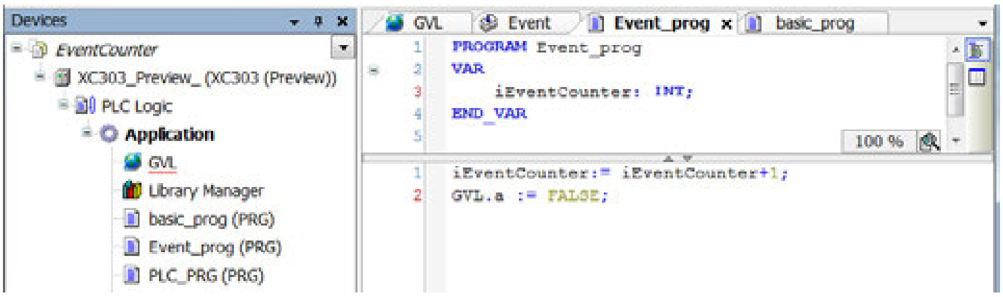
```
bcount := bCount+1;
IF bcount=20 THEN
   GVL.a:= TRUE;
END_IF
```

Figure 37: Program element for cyclic task

4.1.2 Create Event-type task and program start set

The procedure corresponds to the creation of a cyclic task.

- ▶ Create “Event”-type task with the name “Event” as shown in figure 40.
- ▶ Enter the global Boolean variable “a” as the event by clicking the button, double-clicking the “GVL” folder in the “Input Help” window, selecting the variable “a,” and confirming with “OK.”
- ▶ Enter the program call “Event_prog.”



```
PROGRAM Event_prog
VAR
    iEventCounter: INT;
END_VAR
```

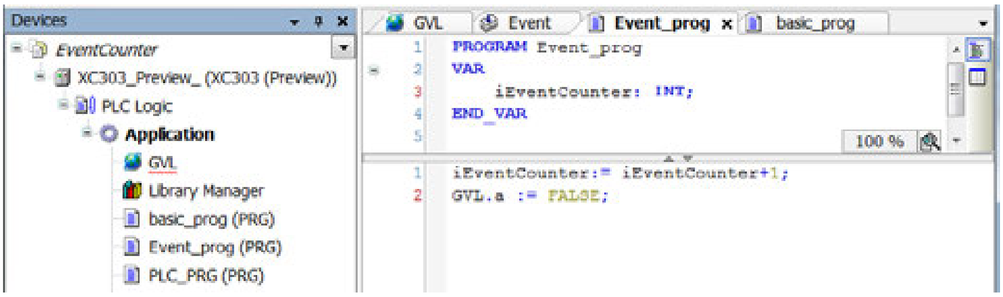
```
iEventCounter:= iEventCounter+1;
GVL.a := FALSE;
```

Figure 38: Create Event-type task

Writing a program

▶ Select “Application” and in the context menu, select the command “Add Object” and choose (POU) as the object with the name “Param_prog.”

▶ You can now enter a program. The program example Param_prog (figure 39) increments the variable “value” by the value 1.
The “Param_prog” program is processed if variable a = TRUE.



```
PROGRAM Event_prog
VAR
    iEventCounter: INT;
END_VAR

iEventCounter:= iEventCounter+1;
GVL.a := FALSE;
```

Figure 39: Programmed element for event controlled task

Events can only be triggered by Boolean variables.

4.2 System events

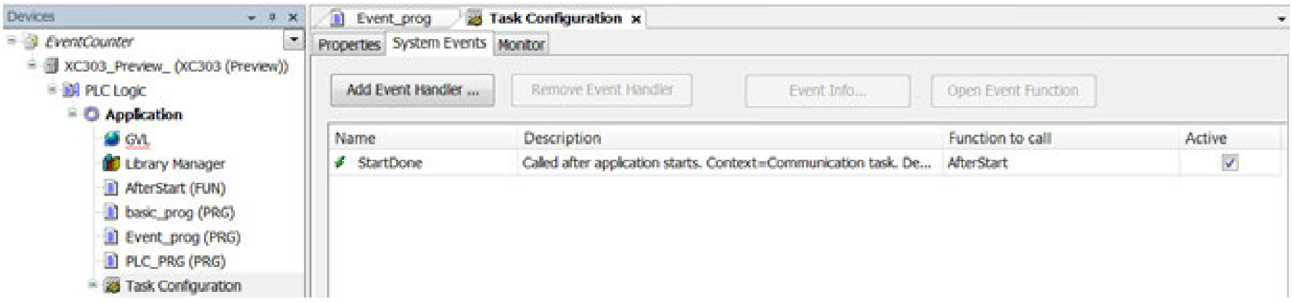
A function can be activated using a system event. It can be used when the PLC is started to initialize modules with new parameters, for example.
The system events are independent of the task!

Assigning a Function to a System Event

▶ In the task configuration, add an event handler in the “System Events” tab.

▶ Select one of the prescribed events, such as “StartDone,” and enter a function name in the “Function to be Called” field, such as “AfterStart.”

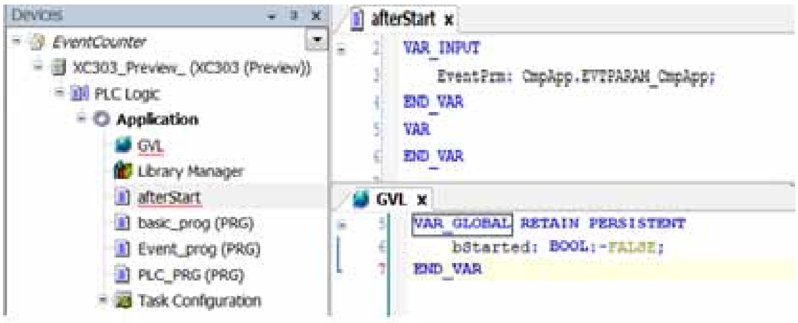
▶ “System Events” – the event, e.g., “Start,” and enter the name of the POU (e.g., “Power_prog”) to be processed.



Name	Description	Function to call	Active
StartDone	Called after application starts. Context=Communication task. De...	AfterStart	☑

Figure 40: Assigning a function to the system event

▶ Double-click a function name to open the Editor and configure what is to be processed after the start.



```
VAR_INPUT
    EventPrm: CmpApp.EVTPARAM_CmpApp;
END_VAR
VAR
END_VAR
```

```
VAR_GLOBAL RETAIN PERSISTENT
    bStarted: BOOL:=FALSE;
END_VAR
```

Figure 41: Programming a function

→ Further information concerning the system events can be found in the online help of the programming system.

4.3 Multitasking

The XC300 runtime system is a multitasking system. This means that multiple tasks can be performed at the same time (in parallel).

Updating the Input/Output Image

If the local and central inputs/outputs are programmed in multiple tasks, the input/output image in XSOFT-CODESYS-3 is updated according to special rules:

By default, only the inputs are read before each task and the outputs to be used in the task in question are set.
This means that the CPU does not work with a constant input/output image within a cycle; instead, the input/output image changes during the cycle depending on the task.

After starting, the system updates the inputs used in the first task, for example. The name of the first task “Prog1” is in the figure 43. If the system detects an input connected to an XN300 slice module by the configuration, such as XN322-16DI, only this input of the module is updated in the image. If further inputs exist in this task and they are assigned to other slice modules, the inputs of these modules are also updated.

The same process occurs for the outputs. For example, if outputs QX6.0 and QX7.1 of output module “1” are used by different tasks, the outputs of this module are not only updated before the first task. The first task could set output QX6.0 and then the second task may reset the output due to the changed input image.

→ Do not access the same input or output from different tasks.

Examples

The examples are based on the following configuration:

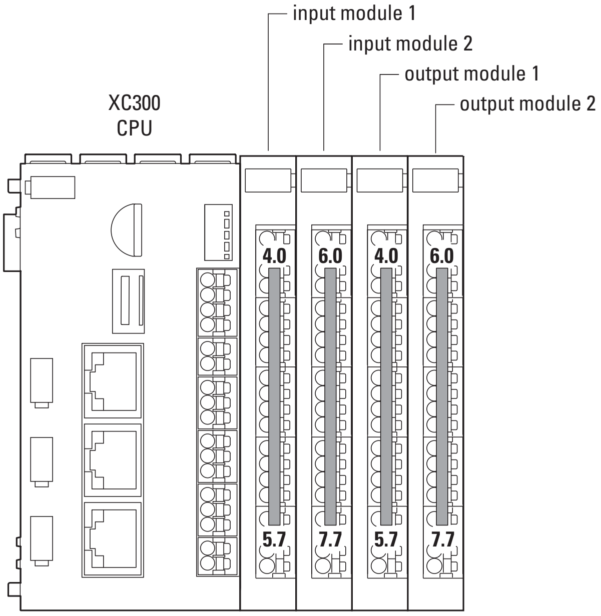
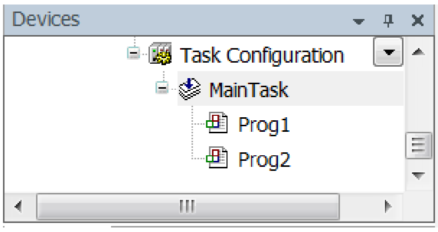


Figure 42: XC300 configuration with local inputs/outputs and I/O system XN300 inputs/outputs as central expansion

The task configuration appears as follows:



Devices
- Task Configuration
  - MainTask
    - Prog1
    - Prog2

Figure 43: Task configuration for examples

Example 1

Table 9: Task details for example 1

Task name	Priority	Cycle time
Prog 1	2	50 ms
Prog 2	1	10 ms

In the first task “Prog1,” input IX4.0 of input module 1 and input IX6.0 of input module 2 are used in the program “progtes(1).” Only these inputs are updated before the first task “Prog1” starts.

In the second task “Prog2,” input IX4.1 of input module 1 is programmed in the program “progtes(2).” Before the second task “Prog2” starts, this input of this input module is updated.

Example 2

Table 10: Task details for example 2

Task name	Priority	Cycle time
Prog 1	2	50 ms
Prog 2	1	20 ms

In example 2, input IX4.0 is programmed in the first task and input IX4.1 and output QX4.0 are programmed in the second task.

Input IX4.0 is updated at the start of the first task. If this input is set, output QX4.0 is also set; otherwise it is reset.

At the start of the second task, input IX4.1 of the input module is updated and output QX4.0 is set or reset.

The control sequence may therefore result in output QX4.0 being set in the first task and then reset in the second task. The output is toggled.

Creating a Task with Consistent Inputs/Outputs

Avoid access to the physical outputs from several tasks. To ensure a clear control sequence, create a task for the local inputs/outputs and the inputs/outputs of the local expansion, in which all inputs are copied to global variables and at the end of the interval, all outputs of global variables are written to the output modules (I/O update task). The I/Os are consistent (data integrity) within this task. The global variables can then be used instead of the I/Os in other tasks.

→ Do not access the same input or output from different tasks.

→ On the XC300 PLC a maximum of 50 tasks are possible.
"Free-running"-type tasks cannot be configured.
Note with parametric programming of the watchdog time that the POU called with the interrupt service routines, extends the task run times accordingly.

4.4 Task Monitoring with a Watchdog

The processing time of a task can be monitored using a watchdog.
The following applies when defining the monitoring time:

Processing time < Task interval time < Watchdog (time)

If the processing time of the task exceeds the interval time, the end of the second interval time is awaited until the task is restarted.
→ Watchdog deactivated.

The watchdog interrupts the program processing if the processing time of the task exceeds the watchdog time.

Furthermore, the frequency (sensitivity) can be set, which allows a certain number of times the watchdog can be exceeded. In this case the outputs of the PLC are switched off and the user program is set to the “Halt” state. Afterwards, the user program must be reset with RESET.

→ If the watchdog is deactivated, tasks are not monitored.

CAUTION
If you want to parameterize a task without a Watchdog or want to deactivate the Watchdog at a later time, all the outputs which have been accessed up to this time can continue to remain active. This is the case for example, when the task can't be ended due to a continuous loop (programming error) and/or missing end condition (stepping condition). The outputs continue to retain their “High” potential until the operating mode is changed from RUN to STOP or until the control voltage for the outputs is switched off.

Configuring a Watchdog

You can select the following settings in the task configuration:

- Watchdog on/off
- Watchdog time
- Watchdog sensitivity.

These settings apply for time controlled and event controlled tasks.

Watchdog active

The watchdog is started at the commencement of every processing cycle and reset again at the end of the task.

→ The following rule applies when defining the watchdog time for multiple tasks: Each watchdog time must be greater than the sum of the task interval times.

If the processing time is longer than the watchdog time (sensitivity = 1) – e.g. with a continuous loop in a program – the watchdog becomes active. If the processing cycle is shorter than the watchdog time, the watchdog is not activated.

The triggering of the watchdog continues to be dependant on the watchdog sensitivity. The watchdog sensitivity determines when the watchdog will be triggered, after the watchdog time has been exceeded by a determined number of consecutive occasions.

The watchdog is triggered:

- immediately when the watchdog time is exceeded with a watchdog sensitivity of “1”,
- immediately after the “x”th consecutive time that the watchdog time is exceeded with a watchdog sensitivity of “x”.

For example, a task with a watchdog time of “10 ms” and a watchdog sensitivity of “5” will end at the latest after 5 × 10 ms = 50 ms.

Example: Watchdog active

The interaction of interval time (IZ), task run time (TZ), watchdog time (WT) and watchdog sensitivity are illustrated by the following configuration example:

- Watchdog on
- Watchdog time (WT) = 15 ms
- Watchdog sensitivity = 2

The interval time (IZ) of the task is 10 ms.

Variant ①: The watchdog is not triggered as the task time always remains below the defined watchdog time.

Variant ②: The watchdog is triggered 15 ms after commencement of the second task ↯ , as both task times are longer than the defined watchdog time and occur consecutively.

Variant ③: The watchdog is triggered 15 ms after commencement of the second consecutive task, which is longer than the defined watchdog time.

Variant ④ – infinite loop: The watchdog is triggered ↯ , as the task time is longer than the watchdog time multiplied by the watchdog sensitivity (2 x 15 ms = 30 ms).

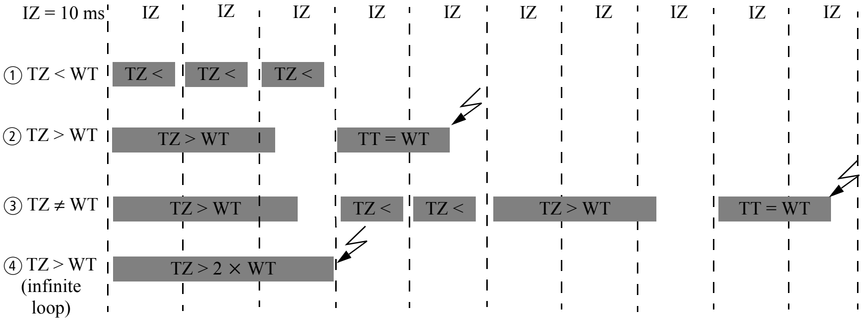


Figure 44: Watchdog active, multiple tasks with differing priority

Watchdog deactivated

The cycle time of a task is not monitored when the watchdog is deactivated. The task ends when it has been processed and only starts again when the next interval time starts.

Example: Watchdog deactivated

The interval time (IZ) is 10 ms.

Variant ①: The interval time (IZ) of a task was set to 10 ms.
The actual task run time (TZ) is 15 ms. The task was started during the initial call but only ended during the second cycle. Therefore, the task is not started again in the second cycle. The task can only be restarted in the third cycle – after 20 ms. This means that the task does not run every 10 ms but rather only within a time interval of 2 × 10 ms.

Variant ②: The running cycle is not ended.

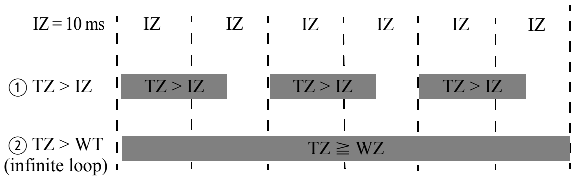


Figure 45: Watchdog deactivated

Multiple tasks with the same priority

You can assign several tasks with the same priority. The tasks are split according to the “Time Slice” principle and are practically executed simultaneously as part intervals (Round Robin).

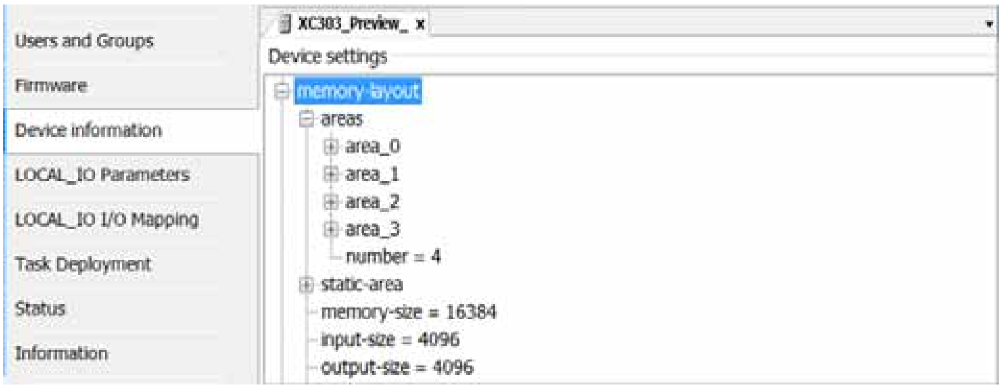
4.5 Limit Values for Memory Usage

The data memory of the XC300 Modular control is divided into memory segments. The segment sizes are shown in figure 46.

The global data utilizes multiple segments. The required amount can be specified to suit the size of the loaded program.

You can see the segment size under ‹Device (XC303) → Device information → memory-layout›:

Example



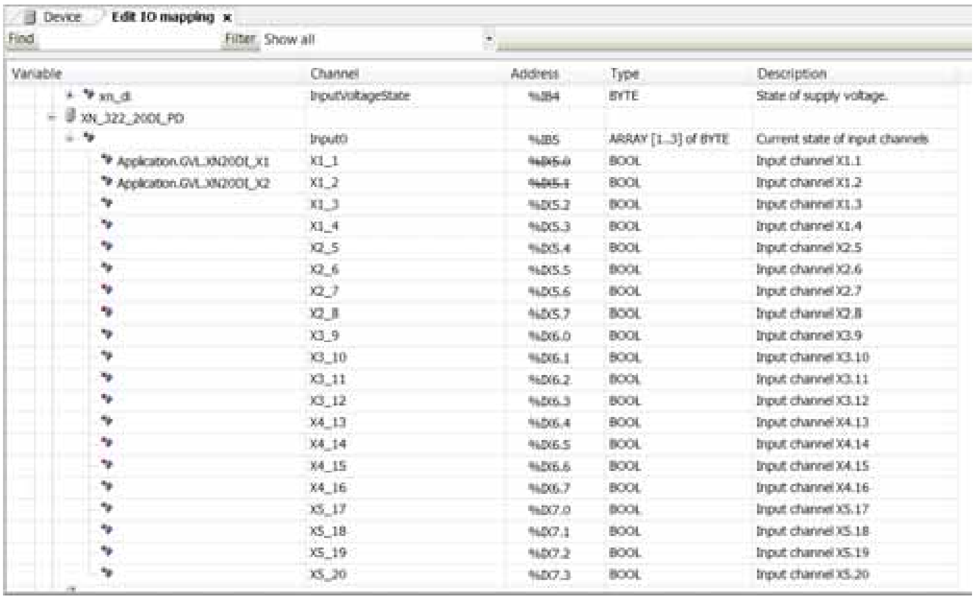
```
memory-layout
  areas
    area_0
    area_1
    area_2
    area_3
    number = 4
  static-area
  memory-size = 16384
  input-size = 4096
  output-size = 4096
```

Figure 46: Memory segment size Device (XC303)

The maximum segment size for inputs, outputs, and markers is 1,280 bytes each and for retain variables, 98 kB.

4.6 Addressing inputs/outputs and marker

If you open the PLC configuration of a new project, you will receive the current view of the default settings of the addressing. In this setting the addresses are automatically assigned and address conflicts (overlaps) are reported.



Device | Edit IO mapping

Find | Filter: Show all

Variable	Channel	Address	Type	Description
xn_di	InputVoltageState	%IB4	BYTE	State of supply voltage.
XN_322_20DI_PD				
	Input0	%IB5	ARRAY [1..3] of BYTE	Current state of input channels
Application.GVL_XN20DI_X1	X1_1	~~%IX5.0~~	BOOL	Input channel X1.1
Application.GVL_XN20DI_X2	X1_2	~~%IX5.1~~	BOOL	Input channel X1.2
	X1_3	%IX5.2	BOOL	Input channel X1.3
	X1_4	%IX5.3	BOOL	Input channel X1.4
	X2_5	%IX5.4	BOOL	Input channel X2.5
	X2_6	%IX5.5	BOOL	Input channel X2.6
	X2_7	%IX5.6	BOOL	Input channel X2.7
	X2_8	%IX5.7	BOOL	Input channel X2.8
	X3_9	%IX6.0	BOOL	Input channel X3.9
	X3_10	%IX6.1	BOOL	Input channel X3.10
	X3_11	%IX6.2	BOOL	Input channel X3.11
	X3_12	%IX6.3	BOOL	Input channel X3.12
	X4_13	%IX6.4	BOOL	Input channel X4.13
	X4_14	%IX6.5	BOOL	Input channel X4.14
	X4_15	%IX6.6	BOOL	Input channel X4.15
	X4_16	%IX6.7	BOOL	Input channel X4.16
	X5_17	%IX7.0	BOOL	Input channel X5.17
	X5_18	%IX7.1	BOOL	Input channel X5.18
	X5_19	%IX7.2	BOOL	Input channel X5.19
	X5_20	%IX7.3	BOOL	Input channel X5.20

Figure 47: Default setting of the addressing

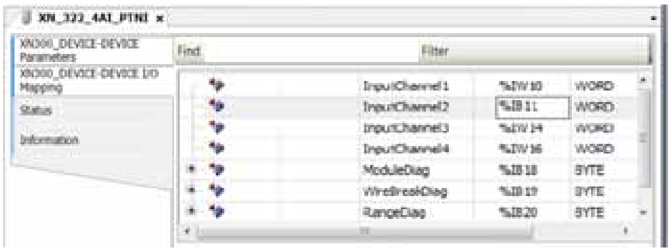
If you add a module to the PLC in the configurator, the configurator will assign this module with an address. Further modules are assigned with the next addresses in ascending order. You can also assign the addresses freely. However, if you access the “Automatic calculation of addresses” function later, the addresses are shown in reassigned ascending order.

Check for overlapping addresses

Duplicate addresses are identified during conversion and an error message is generated.

Addressing by Data Type

If you assign a word-addressable variable of the module, such as I/O system XN300 InputChannel2 of the XN322-4AI-PTNI, with an address that does not match the data type (e.g., IB11) in the “Input Address” field, the word address following the previous variables automatically appears in the controller configurator (e.g., IW12).



XN_322_4AI_PTNI
XN300_DEVICE-DEVICE Parameters
XN300_DEVICE-DEVICE I/O Mapping
Status
Information
Find
Filter

InputChannel1	%IW10	WORD
InputChannel2	%IB11	WORD
InputChannel3	%IW14	WORD
InputChannel4	%IW16	WORD
ModuleDiag	%IB18	BYTE
WireBreakDiag	%IB19	BYTE
RangeDiag	%IB20	BYTE

Figure 48: Addressing by Data Type

4.7 Diagnostics

The following diagnostics options are available:

- LED indicators
- Error messages when converting the project
- I/O system XN300 master diagnostic options
- Shell commands for diagnostics

Subscriber found / Subscriber not found

Runtime system started/not started

PLC LED Indicator

If the program is experiencing a fault, the status indicator LED under the front PLC flashes red and green because the program is being stopped automatically, → figure 33, page 54.

	Status LED		**Description**
	red	**green**	
PLC	–	–	RuntimeSystem not started • No power supply at X6
	–	flashing	Communication
	–	ON	PLC RuntimeSystem started
	ON	–	Error in RuntimeSystem • Flash error • Error in checksum • SSL connection could not be started Remedy: Inspect log file
	ON	ON	RuntimeSystem is started and has an error.

1) flashing: The flashing frequency may differ based on the transfer rate on the bus.

A program error can be caused by the following:

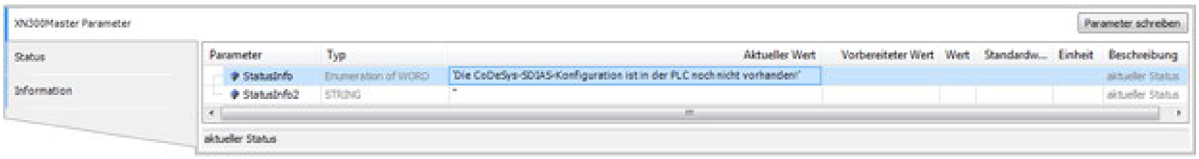
- Watchdog: e.g., interval time of a task > watchdog time
- Faulty access to memory: e.g., an attempt was made to access a protected area

I/O system XN300 master diagnostic options

For errors that occur in the system block (local expansion XC303 with modules with I/O system XN300), the status of the XN300 master can be called up for diagnostic purposes.
To do so, follow the steps below:

- ▶ Open XSOFT-CODESYS-3 and go to the device online.
- ▶ In the configuration tree diagram, select the device “Device (XC303)” and “XN300_Master.”
- ▶ Double-click “XN300_Master” and select the “XN300Master Parameter” tab.

- ▶ In the event of an error, the “Current Value” column contains an error number and a related description, such as that shown in the following example with error number “47” and the description “The CoDeSys-SDIAS configuration is not yet available in the PLC!”

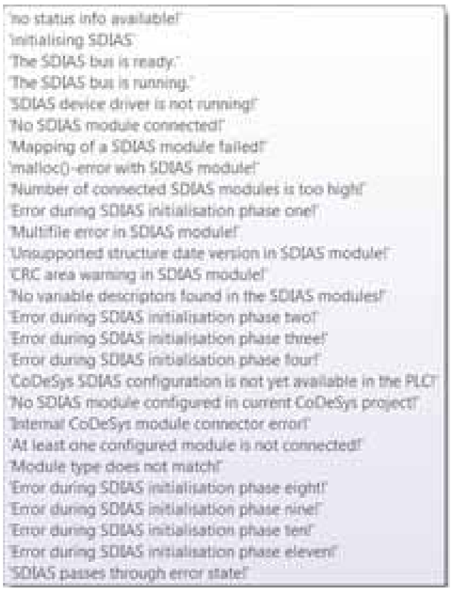


Parameter	Typ	Aktueller Wert	Vorbereiteter Wert	Wert	Standardw...	Einheit	Beschreibung
StatusInfo	Enumeration of WORD	'Die CoDeSys-SDIAS-Konfiguration ist in der PLC noch nicht vorhanden!'					aktueller Status
StatusInfo2	STRING	'					aktueller Status

aktueller Status

Figure 49: XN300Master Parameter tab with error message
(Example German sreen with special error message)

Sample display of messages from local XN300 system bus



```
'no status info available!'
'initialising SDIAS'
'The SDIAS bus is ready.'
'The SDIAS bus is running.'
'SDIAS device driver is not running!'
'No SDIAS module connected!'
'Mapping of a SDIAS module failed!'
'malloc()-error with SDIAS module!'
'Number of connected SDIAS modules is too high!'
'Error during SDIAS initialisation phase one!'
'Multifile error in SDIAS module!'
'Unsupported structure date version in SDIAS module!'
'CRC area warning in SDIAS module!'
'No variable descriptors found in the SDIAS modules!'
'Error during SDIAS initialisation phase two!'
'Error during SDIAS initialisation phase three!'
'Error during SDIAS initialisation phase four!'
'CoDeSys SDIAS configuration is not yet available in the PLC!'
'No SDIAS module configured in current CoDeSys project!'
'Internal CoDeSys module connector error!'
'At least one configured module is not connected!'
'Module type does not match!'
'Error during SDIAS initialisation phase eight!'
'Error during SDIAS initialisation phase nine!'
'Error during SDIAS initialisation phase ten!'
'Error during SDIAS initialisation phase eleven!'
'SDIAS passes through error state!'
```

→ More information on diagnostics can be found in the manual for the I/O system XN300,
“XN300 slice modules”, or “CANopen Gateway XN-312-GW-CAN”

Shell Commands for Diagnostics for Modules from the I/O system XN300

- ▶ Double-click on the device in the Explorer, e.g., “Device (XC303),” go to the “Device” tab, and select “PLC Shell.”
- ▶ Enter the command “?” in the command line.

5 Establishing a Connection with a PC – XC300

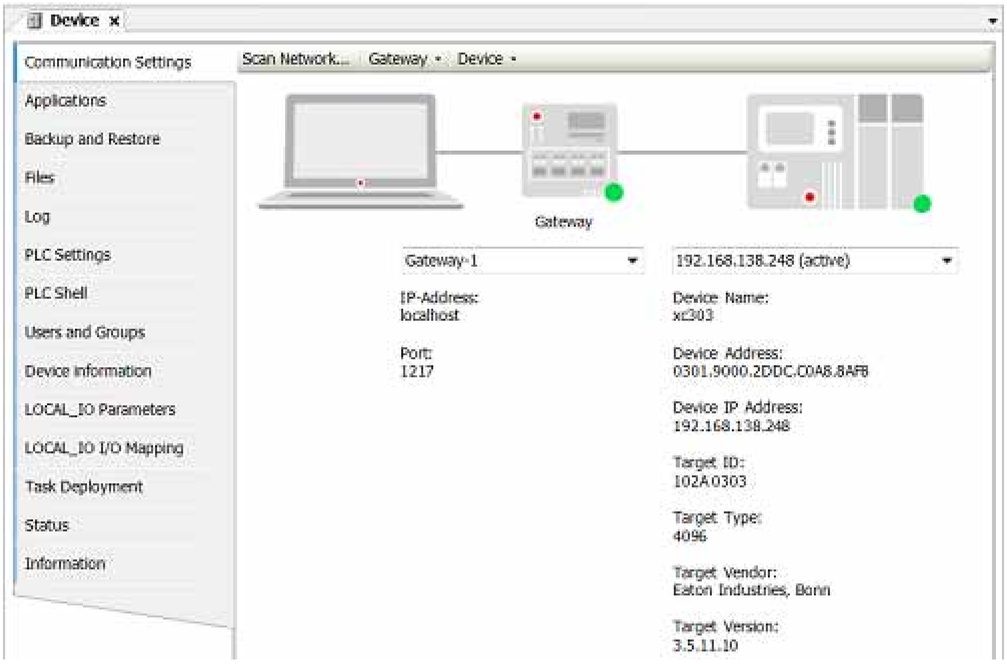
The connection between the PC and CPU can be established via one of the Ethernet interfaces ETH0 to ETH2.

The transfer rate of the Ethernet connection is controlled with the autosensing function. Components in (detect) mode automatically determine whether the connection has a rate of 10- or 100-MBits per second.

This chapter describes what settings must be defined here in the XSOFT-CODESYS-3 software. A new project is created in the following example.

5.1 Searching the Network with Automatic Controller Detection

The PC attempts to connect to the controller and reports the successful connection by displaying green marker points.



Gateway-1

IP-Address:
localhost

Port:
1217

192.168.138.248 (active)

Device Name:
xc303

Device Address:
0301.9000.2DDC.C0A8.8AF8

Device IP Address:
192.168.138.248

Target ID:
102A 0303

Target Type:
4096

Target Vendor:
Eaton Industries, Bonn

Target Version:
3.5.11.10

Figure 50: Successful connection between PC and controller XC300

5.2 Creating a new project

- ▶ First ensure that you have connected the PC to the Ethernet interface of the device by cable using the Ether interfaces or USB/Ethernet adapter.
- ▶ Ensure that the IP addresses of the PC and device come from the same number range.
- ▶ Start the XSOFT-CODESYS-3.
- ▶ Select “File,” “New Project,” “Standard Project” and enter a project name, e.g., “Test.”
- ▶ In the window that follows, select “XC-303 (Eaton Automation)” as the device and choose your desired programming method, such as “Structured Text (ST).”
- ▶ Confirm the displayed information by clicking on “OK.”

The device will be added to the project. The standard tree diagram for the project is created as the configuration tree diagram. The application contains the library manager with all required libraries, and the main task “PLC_PRG” is also created.

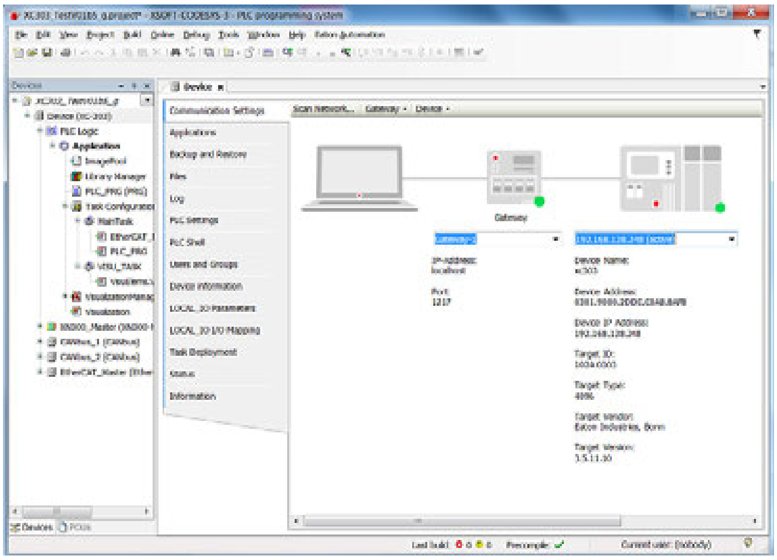
Establishing Communication

- ▶ The device name “Device (XC-303),” the application “Application,” and the main program “PLC_PRG” can be edited and renamed.
- ▶ Double-click the device “Device (XC-303)” and select the “Communication” tab.
- ▶ Click on the “Search Network” button or enter the IP address of the controller.

All devices that can be accessed are listed.

- ▶ You can use the “Flash” button to display a temporary flashing pattern on the marked device.

→ Use this to uniquely identify the controller.



IP-Address:
localhost

Port:
1217

Device Name:
xc303

Device Address:
[illegible]

Device IP Address:
192.168.128.248

Target ID:
[illegible]

Target Type:
4096

Target Vendor:
Eaton Industries, Bonn

Target Version:
3.5.11.30

Figure 51: Adding a device using its IP address

5.3 Changing the IP Address on the Controller

When changing the IP addresses, ensure that the IP address of the PC and of the XC300 Modular control belong to the same network/subnet. Example:

IP address of the **PC**: 192.168.138.100
Subnet mask: 255.255.255.0

IP address of the **XC300**: 192.168.138.248
Subnet mask: 255.255.255.0

There are three different ways to change the IP address of the controller:

1. PLC shell commands getipconfig; setip... ...addr/...netmask/...gateway
2. setipconfig and getipconfig function in the EA_Network library
3. networkconfig.templ on MicroSD memory card or USB-Device

PLC Shell Command

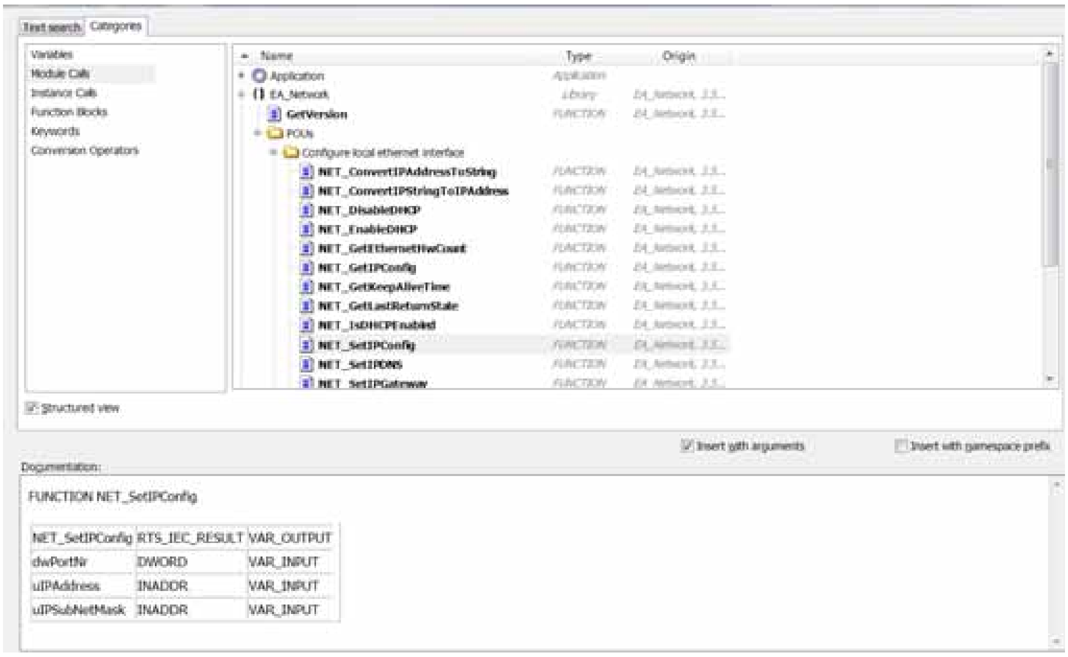
- Open the PLC browser, see → section 16.1, “Accessing PLC Shell Commands”, page 133.
- In the command line, enter setipconfig <IP-Adresse> On
- Confirm with <Eingabetaste>.
- Restart the device once you have changed the IP address.

setipconfig Function

Ensure that the EA_Network library is added to the library manager.

Use the function in your program, e.g., in “PLC_PRG.”

- Open the input tool with <F2> to do this.



Name	Type	Origin
Application	Application	
EA_Network	Library	EA_Network, 3.5...
GetVersion	FUNCTION	EA_Network, 3.5...
POUs		
Configure local ethernet interface		
NET_ConvertIPAddressToString	FUNCTION	EA_Network, 3.5...
NET_ConvertIPStringToIPAddress	FUNCTION	EA_Network, 3.5...
NET_DisableDHCP	FUNCTION	EA_Network, 3.5...
NET_EnableDHCP	FUNCTION	EA_Network, 3.5...
NET_GetEthernetHwCount	FUNCTION	EA_Network, 3.5...
NET_GetIPConfig	FUNCTION	EA_Network, 3.5...
NET_GetKeepAliveTime	FUNCTION	EA_Network, 3.5...
NET_GetLastReturnState	FUNCTION	EA_Network, 3.5...
NET_IsDHCPEnabled	FUNCTION	EA_Network, 3.5...
NET_SetIPConfig	FUNCTION	EA_Network, 3.5...
NET_SetIPDNS	FUNCTION	EA_Network, 3.5...
NET_SetIPGateway	FUNCTION	EA_Network, 3.5...

FUNCTION NET_SetIPConfig

NET_SetIPConfig	RTS_IEC_RESULT	VAR_OUTPUT
dwPortNr	DWORD	VAR_INPUT
uIPAddress	INADDR	VAR_INPUT
uIPSubNetMask	INADDR	VAR_INPUT

Figure 52: Input Tool

- ▶ In the "Category," select "Element Call."
- ▶ Select "EA_Network," "POUs," "Configure local ethernet interface," "NET_SetIPConfig."
- ▶ Enter the port, IP address, and subnet mask as parameters.

networkconfig

If you cannot establish a connection to the device, because the IP address is not known, for example, you can use the "networkconfig.templ" file with the aid of the MicroSD memory card or a USB-Device to reset the IP address of the Ethernet interfaces of the XC300.

Setting the IP address of ETHx Ethernet interface:

Requirement: Variable NW_UPDATE_FROM_USB or NW_UPDATE_FROM_SD must be set to "1."

- ▶ Save file "networkconfig.templ" to a directory on your PC. The file is located on the XC300 in directory "home".
- ▶ Edit the file.
- ▶ Assign a valid IP-V4-IP address and subnet mask e.g., ETH1_IP_ADDRESS=192.168.137.248 and ETH1_SUBNET_MASK=255.255.255.0

→ When changing the properties of an Ethernet adapter, all parameters must be entered completely in the file, even if they correspond to the default settings.

- ▶ Save file networkconfig.templ under the name networkconfig.
- ▶ Copy the file to the MicroSD memory card/USB-Device in directory XC303/update.
- ▶ Insert the MicroSD memory card into the XC300.
- ▶ Reboot the XC300 and then remove the file from the MicroSD memory card/USB-Device.

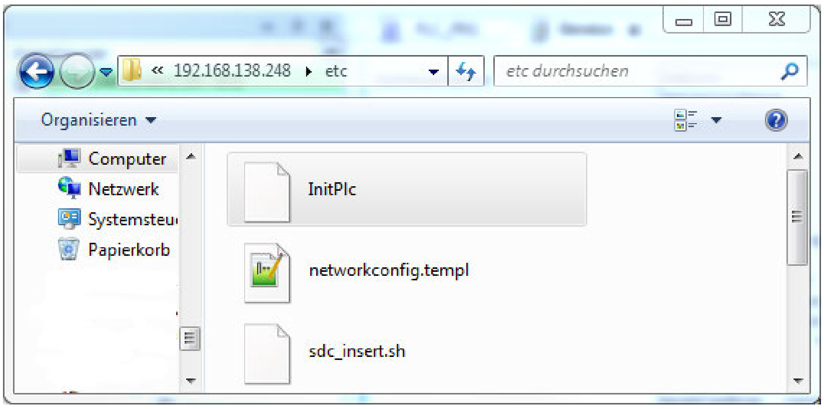


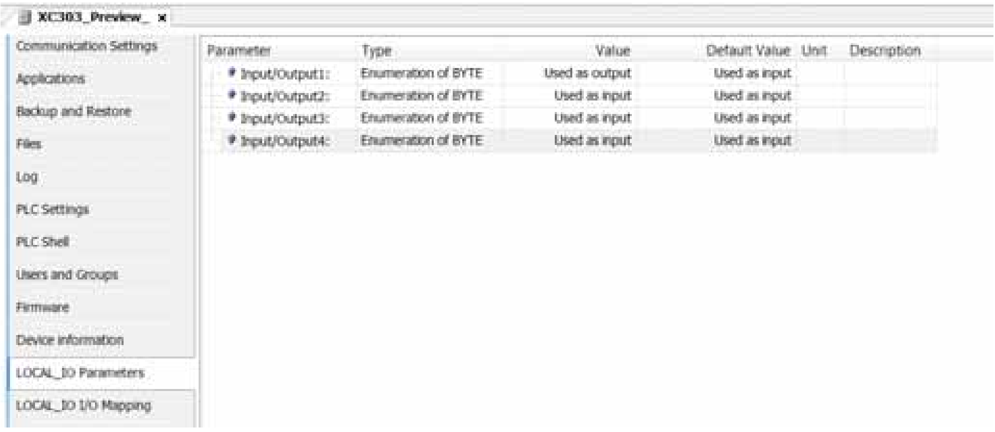
Figure 53: File networkconfig.templ on the XC300 (Example German sreen)

6 Configuring and parametizing Inputs/Outputs

6.1 Configuring Local Inputs/Outputs of the Device

The local inputs/outputs of the controller, I/Q1 to I/Q4, are configured as inputs by default. If you wish to change the configuration, proceed as follows:

- ▶ Select the device in the configuration tree diagram and open the “Device” tab by double-clicking it.
- ▶ Select “LOCAL_IO Parameter.”
- ▶ Double-click the “Value” column to get a selection field where you can define the nodes as inputs or outputs or as interrupt inputs.



Parameter	Type	Value	Default Value	Unit	Description
Input/Output1:	Enumeration of BYTE	Used as output	Used as input		
Input/Output2:	Enumeration of BYTE	Used as input	Used as input		
Input/Output3:	Enumeration of BYTE	Used as input	Used as input		
Input/Output4:	Enumeration of BYTE	Used as input	Used as input		

Figure 54: Configuring local inputs/outputs

6.2 Interrupt Processing

If an interrupt occurs, the operating system executes the program organizational unit (POU) which is linked to the interrupt source.

→ The execution of the interrupt POU is **not** time monitored. Programmed infinite loops are not terminated.

The local inputs I1 – I4 can be used as interrupt sources.

They are configured via Device -> local_IO parameter.

Available settings:

- Int falling
- Int rising
- Int falling + rising

The POU initiated by the interrupt is always run to completion and cannot be interrupted by a new interrupt. A new interrupt is only carried out after the current interrupt has ended.

> *ATTENTION*
> All the outputs controlled (H signals) up to this point remain active and can’t be switched off.

The interrupts are enabled in the RUN state of the CPU and disabled in the STOP state. Interrupt sources which are not enabled in the configuration do not initiate an interrupt. If a POU is not assigned to an enabled interrupt source, the interrupt is recognized and executed but without running a POU.

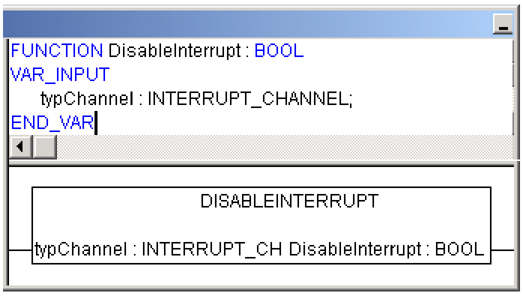
Frequent interrupts during program execution can cause the programmed task time to time out and result in a RESET being initiated by the Watchdog.

User interrupts can be inhibited and re-enabled from the program. The functions Disableinterrupt and Enableinterrupt exist for this purpose. A call parameter in the XSOFT-CODESYS-3 software determines if an individual interrupt or all interrupts are enabled or disabled. A disabled interrupt must be enabled with the same parameter used to disable it.

The two functions DisableInterrupt and EnableInterrupt belong to the library EA_XC303Events.lib. This library must – if not already done so – be integrated into the XSOFT-CODESYS-3 library manager, see → section 15.2, “Installing additional system libraries”, page 130.

DisableInterrupt

With this function, you disable (deactivate) a parameterized physical interrupt by accessing it from the user program.

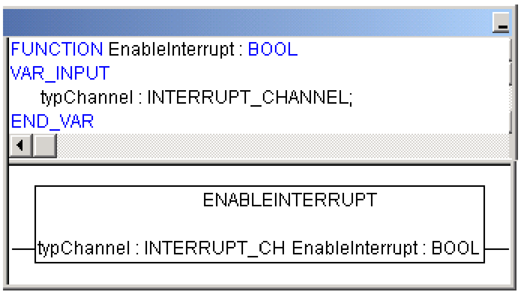


```
FUNCTION DisableInterrupt : BOOL
VAR_INPUT
    typChannel : INTERRUPT_CHANNEL;
END_VAR
```

Figure 55: “DisableInterrupt” function

EnableInterrupt

With this function, the physical interrupt which was deactivated beforehand can now be re-enabled as an active interrupt.



```
FUNCTION EnableInterrupt : BOOL
VAR_INPUT
    typChannel : INTERRUPT_CHANNEL;
END_VAR
```

Figure 56: "EnableInterrupt" function

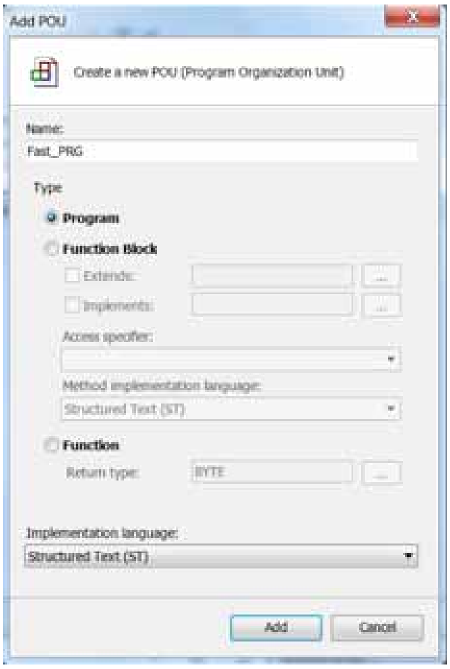
Configuration

The interrupt is configured and assigned a priority in the "PLC and Task Configuration" of the CODESYS (activate the "Resources" tab and open the "Task Configuration → System Events" folder). Each interrupt can be assigned with a POU here.

Example for Interrupt Processing

A "Basic" task contains a POU "PLC_PRG". A further POU "Fastprog" should be processed if an L → H edge at input I0.4 generates an interrupt.

▶ Create the POUs "FAST_PRG" as shown in → figure 57.



Add POU

Create a new POU (Program Organization Unit)

Name: Fast_PRG

Type

● Program

○ Function Block

□ Extends:

□ Implements:

Access specifier:

Method implementation language: Structured Text (ST)

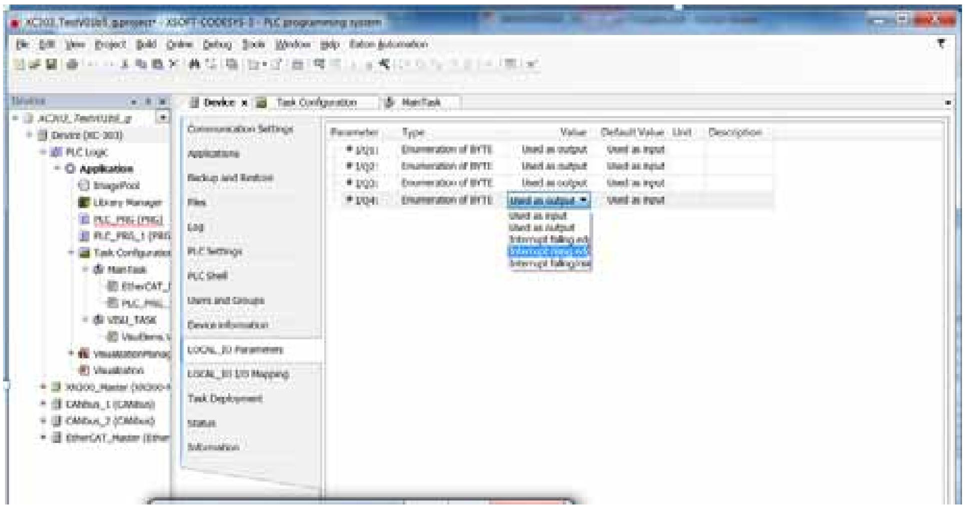
○ Function

Return type: BYTE

Implementation language: Structured Text (ST)

Figure 57: PLC and Task Configuration

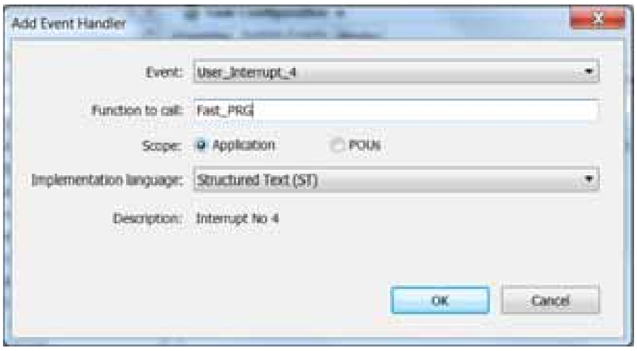
- ▶ Switch to the PLC configuration and assign input I0.4 (L → H edge) the interrupt source “Interrupt rising edge,” for example, from the drop-down menu.



Parameter	Type	Value	Default Value	Unit	Description
I/Q1:	Enumeration of BYTE	Used as output	Used as input		
I/Q2:	Enumeration of BYTE	Used as output	Used as input		
I/Q3:	Enumeration of BYTE	Used as output	Used as input		
I/Q4:	Enumeration of BYTE	Used as output	Used as input		

Figure 58: Assigning the I0.5 → interrupt source

- ▶ Change over to the Task configuration and open the System Events folder.



Add Event Handler

Event: User_Interrupt_4

Function to call: Fast_PRG

Scope: ● Application ○ POUs

Implementation language: Structured Text (ST)

Description: Interrupt No 4

Figure 59: System events

- ▶ Select the “User_Interrupt_4.”
- ▶ Mark the area of the “Called POU” column.
- ▶ Select the FAST_PRG POU and confirm with “OK.”
- ▶ Save the project. You can now test it.

With each rising edge at input 4, the program FAST_PRG opens.

6.3 Secure Communication

Information in the XSOFT-CODESYS-3 online help under:

- Security screen
- Certificate selection
- Encryption and signing with certificates

What can be protected?

- Communication with the controller
- Communication with the web server
- Boot project
- Online access to the application
- Access to the application via the programming PC

7 Assigning System Parameters

7.1 Overview

System parameters independent of the project can be set by you and saved on the memory card. They are compiled in the following files:

- netwerkconfig.templ -> networkconfig
- deleteboardapplication.templ -> deleteboardapplication

The memory card can be used in other controllers, too.
When started, the controller automatically adopts the parameters set in the files.

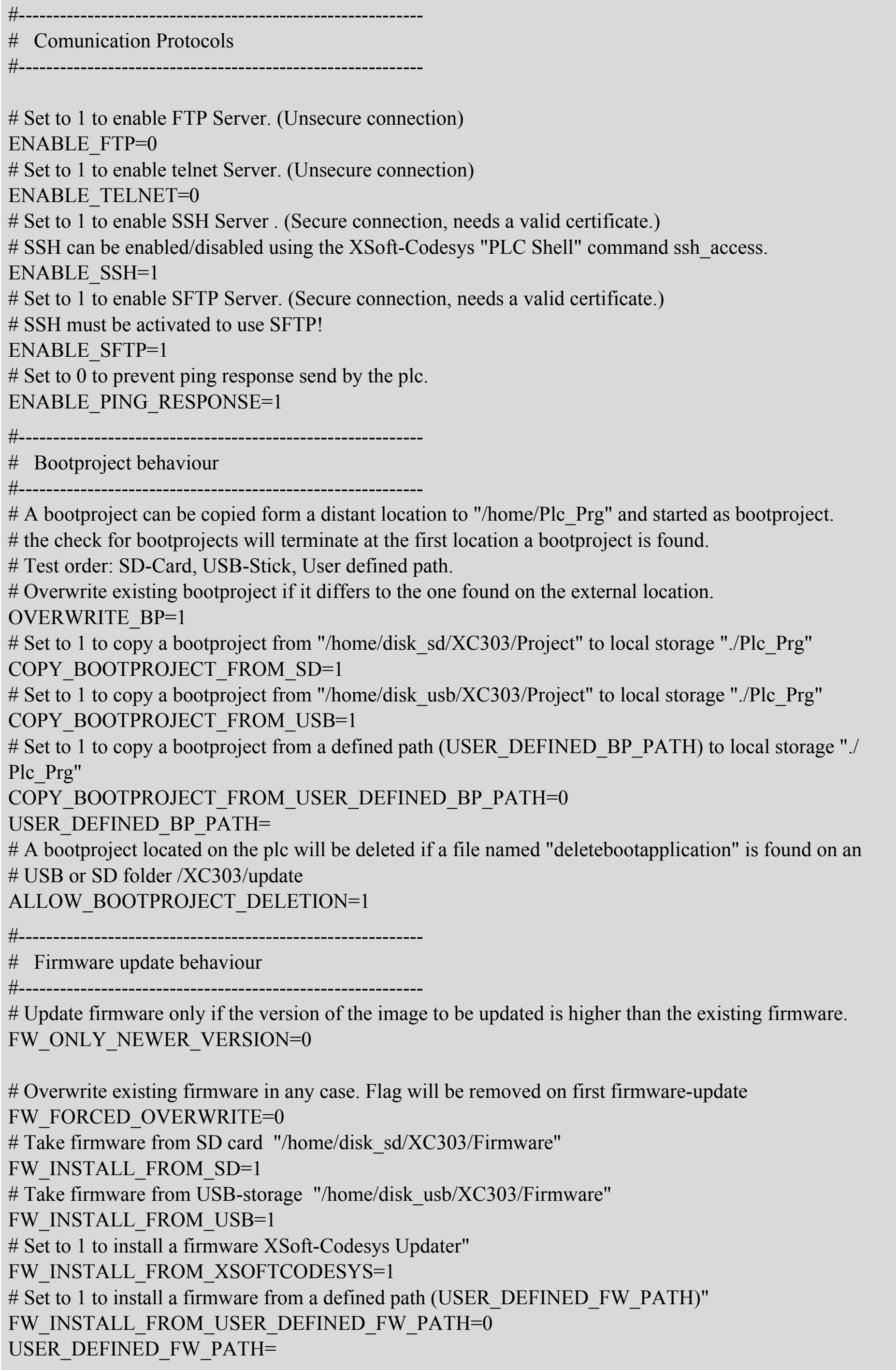
- SD/USB in folder XC-303/update
- deleteboardapplication.templ -> deleteboardapplication

Templates on the controller in folder \home.

7.2 File InitPLC with Default Parameters

The InitPLC file is always generated with all controller-specific entries.

All parameters are already predefined. The following display contains the parameters with their default values, which you can adapt as desired.



```
#-----------------------------------------------------------
#   Comunication Protocols
#-----------------------------------------------------------

# Set to 1 to enable FTP Server. (Unsecure connection)
ENABLE_FTP=0
# Set to 1 to enable telnet Server. (Unsecure connection)
ENABLE_TELNET=0
# Set to 1 to enable SSH Server . (Secure connection, needs a valid certificate.)
# SSH can be enabled/disabled using the XSoft-Codesys "PLC Shell" command ssh_access.
ENABLE_SSH=1
# Set to 1 to enable SFTP Server. (Secure connection, needs a valid certificate.)
# SSH must be activated to use SFTP!
ENABLE_SFTP=1
# Set to 0 to prevent ping response send by the plc.
ENABLE_PING_RESPONSE=1
#-----------------------------------------------------------
#   Bootproject behaviour
#-----------------------------------------------------------
# A bootproject can be copied form a distant location to "/home/Plc_Prg" and started as bootproject.
# the check for bootprojects will terminate at the first location a bootproject is found.
# Test order: SD-Card, USB-Stick, User defined path.
# Overwrite existing bootproject if it differs to the one found on the external location.
OVERWRITE_BP=1
# Set to 1 to copy a bootproject from "/home/disk_sd/XC303/Project" to local storage "./Plc_Prg"
COPY_BOOTPROJECT_FROM_SD=1
# Set to 1 to copy a bootproject from "/home/disk_usb/XC303/Project" to local storage "./Plc_Prg"
COPY_BOOTPROJECT_FROM_USB=1
# Set to 1 to copy a bootproject from a defined path (USER_DEFINED_BP_PATH) to local storage "./
Plc_Prg"
COPY_BOOTPROJECT_FROM_USER_DEFINED_BP_PATH=0
USER_DEFINED_BP_PATH=
# A bootproject located on the plc will be deleted if a file named "deletebootapplication" is found on an
# USB or SD folder /XC303/update
ALLOW_BOOTPROJECT_DELETION=1
#-----------------------------------------------------------
#   Firmware update behaviour
#-----------------------------------------------------------
# Update firmware only if the version of the image to be updated is higher than the existing firmware.
FW_ONLY_NEWER_VERSION=0

# Overwrite existing firmware in any case. Flag will be removed on first firmware-update
FW_FORCED_OVERWRITE=0
# Take firmware from SD card  "/home/disk_sd/XC303/Firmware"
FW_INSTALL_FROM_SD=1
# Take firmware from USB-storage  "/home/disk_usb/XC303/Firmware"
FW_INSTALL_FROM_USB=1
# Set to 1 to install a firmware XSoft-Codesys Updater"
FW_INSTALL_FROM_XSOFTCODESYS=1
# Set to 1 to install a firmware from a defined path (USER_DEFINED_FW_PATH)"
FW_INSTALL_FROM_USER_DEFINED_FW_PATH=0
USER_DEFINED_FW_PATH=
#------------------------------------------------------------
#   Reset network settings
#------------------------------------------------------------
# You should take care to move the networkconf file, once the reset values have been applied.
# A networkconf template file can be found on /home/etc/networkconfig.templ
# Reset network configuration by booting the plc with a config File located on /home/disk_usb/XC303/
update/networkconfig
NW_UPDATE_FROM_USB=1
# Reset network configuration by booting the plc with a config File located on /home/disk_sd/XC303/update/
networkconfig
NW_UPDATE_FROM_SD=1

#------------------------------------------------------------
#   Start behaviour
#------------------------------------------------------------
# Delay the start of the runtime system for a given number of seconds.
# (This start delay may be needed to detect USB devices.)
PLC_START_DELAY_SEC=4
# Set to 1 to skip the plc-runtime start
DISABLE_PLC_START=0
# Restrict the processor core usage of the plc-runtime. Valid entries: "CORE_0", "CORE_1", "BOTH"
PLC_CORE_USAGE=CORE_1
# Set to 1 to constantly check rts-state.
PLC_SUPERVISE_RTS=1
# Set to 1 to Reboot the PLC if the runtimesystem terminates unexpectedly.
PLC_REBOOT_ON_RTS_TERMINATION=1

#------------------------------------------------------------
#   Log Files
#------------------------------------------------------------
# Defined path used to safe log files.(Default path "/var/log" is volatile,
# log files located there are lost after a power cycle.)
LOGFILE_DIR=/var/log
# Name of log file containing messages during the start up process.
LOGFILE_STARTUP=XC303Log.txt
# Name of log file containing messages from the plc runtime system.
LOGFILE_RTS=codesyscontrol.log

#------------------------------------------------------------
#   Map network drive
#------------------------------------------------------------
# A network drive may be mapped into the view of the plc filesystem using nfs (Linux/Unix) or cifs
(Windows).
# Using nfs (network file system). A path "/home/disk_nfs" will be created.
MAP_NFS_FOLDER=0
NFS_SOURCE_PATH=
# Using cifs (common internet file system). A path "/home/disk_cifs" will be created
MAP_CIFS_FOLDER=0
CIFS_SOURCE_PATH=
CIFS_USER_NAME=
CIFS_DOMAIN_NAME=
CIFS_PASSWORD=
```

7.3 Switching On the Controller with Memory Card with File “InitPLC” Inserted

These system parameters are also active after a new program is loaded.

The start process updates the following:

- Firmware
- Boot project (user program)
- Network configuration

See also → section 3.2, “Startup Behavior XC300”, page 47.

In all checking processes, once the XC300 Modular control is started the MicroSD slot is checked first and only then is the USB-Host checked.

If a boot application exists on the MicroSD memory card or USB-Device and this application has a newer version then the version in the XC300, the boot application in the controller is updated.

The retentive data is then loaded.
If the controller is in a warm/cold start and the “STOP” DIP switch is set to the “OFF” position, the user program is started.

The DIP switch configuration is described in detail in → section 3.3.1, “DIP Switch S1”, page 49.

7.4 File “networkconfig”

The device settings for the network can be changed in this file.
Empty entries are ignored.

To adjust the parameters, proceed as follows:

Requirement: Variable NW_UPDATE_FROM_USB or NW_UPDATE_FROM_SD must be set to “1.”

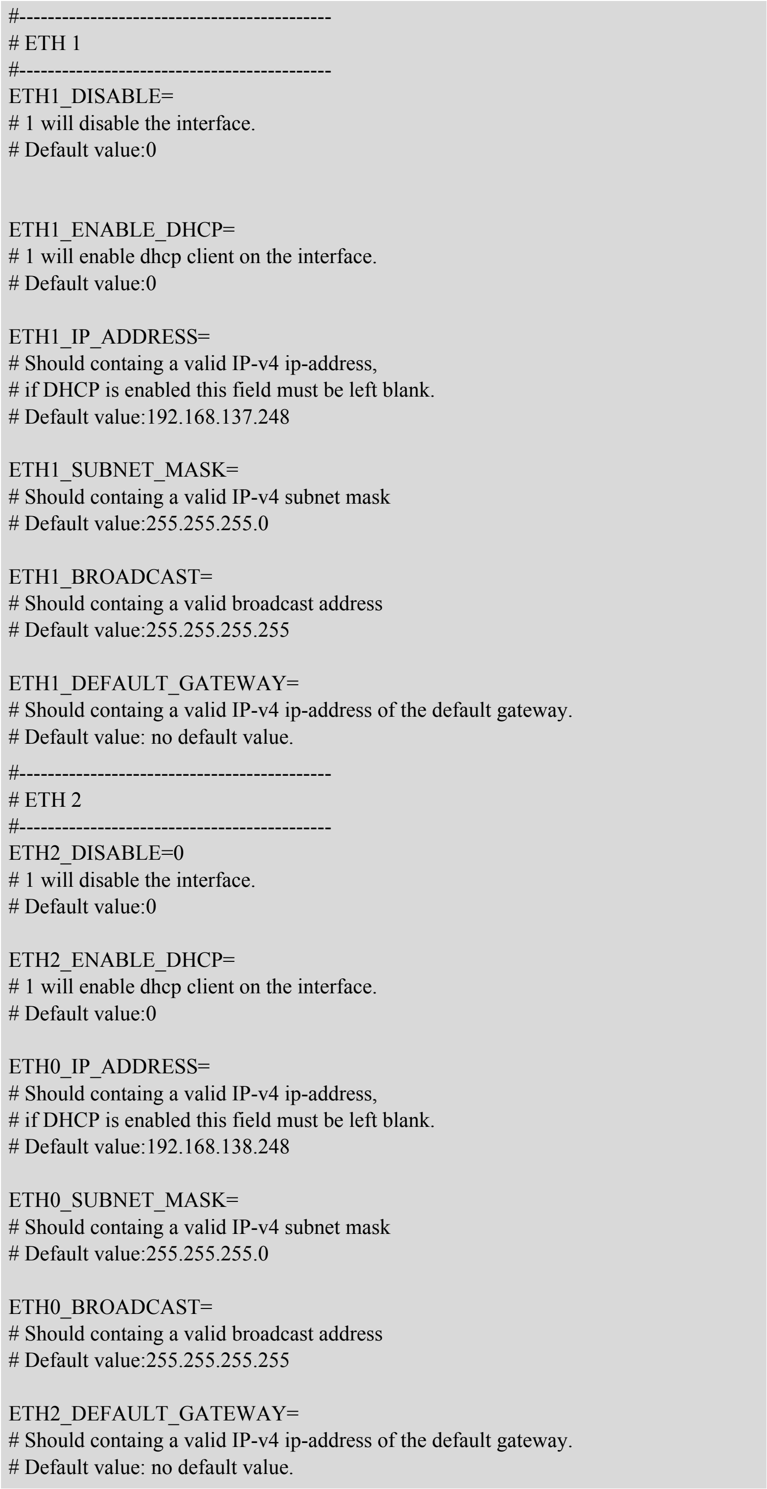
▶ Open the file networkconfig and edit the desired parameters.
▶ Save the changed file under the name “networkconfig” in the directory ./XC303/update on the MicroSD memory card or USB-Device.
▶ Insert the MicroSD memory card into the slot or connect the USB-Device.
▶ Start the XC300 Modular control.

Templates of file networkconfig and deleteBootproject can be found in the Eaton Download Center.

Table 11: Predefined default parameters in file “networkconfig”

```
#-------------------------------------------
# ETH 0
#-------------------------------------------
ETH0_DISABLE=
# 1 will disable the interface.
# Default value:0

ETH0_ENABLE_DHCP=
# 1 will enable dhcp client on the interface.
# Default value:0

ETH0_IP_ADDRESS=
# Should containg a valid IP-v4 ip-address,
# if DHCP is enabled this field must be left blank.
# Default value:192.168.119.248

ETH0_SUBNET_MASK=
# Should containg a valid IP-v4 subnet mask
# Default value:255.255.255.0

ETH0_BROADCAST=
# Should containg a valid broadcast address
# Default value:255.255.255.255

ETH0_DEFAULT_GATEWAY=
# Should containg a valid IP-v4 ip-address of the default gateway.
# Default value: no default value.
```

File “networkconfig”



```
#-------------------------------------------
# ETH 1
#-------------------------------------------
ETH1_DISABLE=
# 1 will disable the interface.
# Default value:0


ETH1_ENABLE_DHCP=
# 1 will enable dhcp client on the interface.
# Default value:0

ETH1_IP_ADDRESS=
# Should containg a valid IP-v4 ip-address,
# if DHCP is enabled this field must be left blank.
# Default value:192.168.137.248

ETH1_SUBNET_MASK=
# Should containg a valid IP-v4 subnet mask
# Default value:255.255.255.0

ETH1_BROADCAST=
# Should containg a valid broadcast address
# Default value:255.255.255.255

ETH1_DEFAULT_GATEWAY=
# Should containg a valid IP-v4 ip-address of the default gateway.
# Default value: no default value.
#-------------------------------------------
# ETH 2
#-------------------------------------------
ETH2_DISABLE=0
# 1 will disable the interface.
# Default value:0

ETH2_ENABLE_DHCP=
# 1 will enable dhcp client on the interface.
# Default value:0

ETH0_IP_ADDRESS=
# Should containg a valid IP-v4 ip-address,
# if DHCP is enabled this field must be left blank.
# Default value:192.168.138.248

ETH0_SUBNET_MASK=
# Should containg a valid IP-v4 subnet mask
# Default value:255.255.255.0

ETH0_BROADCAST=
# Should containg a valid broadcast address
# Default value:255.255.255.255

ETH2_DEFAULT_GATEWAY=
# Should containg a valid IP-v4 ip-address of the default gateway.
# Default value: no default value.
```

8 Expansion

8.1 Local expansion

The XC300 controller can be expanded centrally by up to 32 XN300 slice modules. To do so, the controller and the XN300 slice modules are connected and mounted together on a mounting rail (DIN rail). The I/O system XN300 provides a wide range of XN300 slice moduless, which offer special functions in addition to digital and analog inputs and outputs, such as the weighing module or the motor driver module.

→ For up-to-date information about the I/O system XN300, refer to the product pages:
www.eaton.eu/xn300
and for the available product range, refer to the EATON online catalog.
http://www.eaton.eu/ecat

Entering "XN300" in the search field will take you directly to this product group.
All modules are described in the "XN300 slice modules" manual.

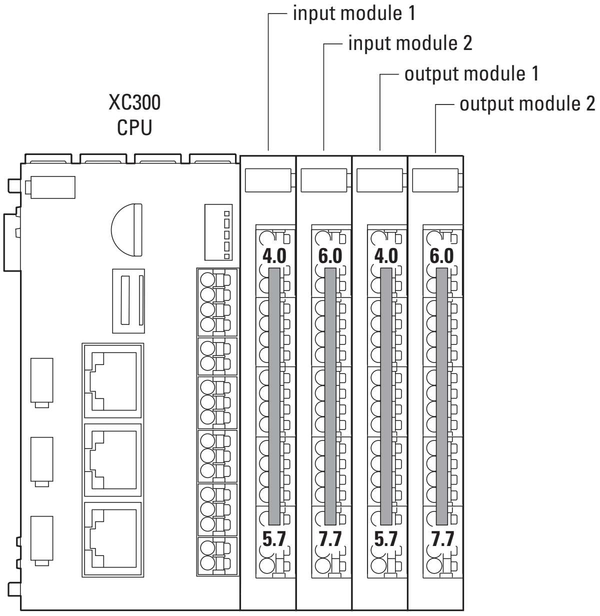


Figure 60: Central XC300 expansion with XN300 slice modules

8.1.1 Configuring Inputs/Outputs of the Central Expansion

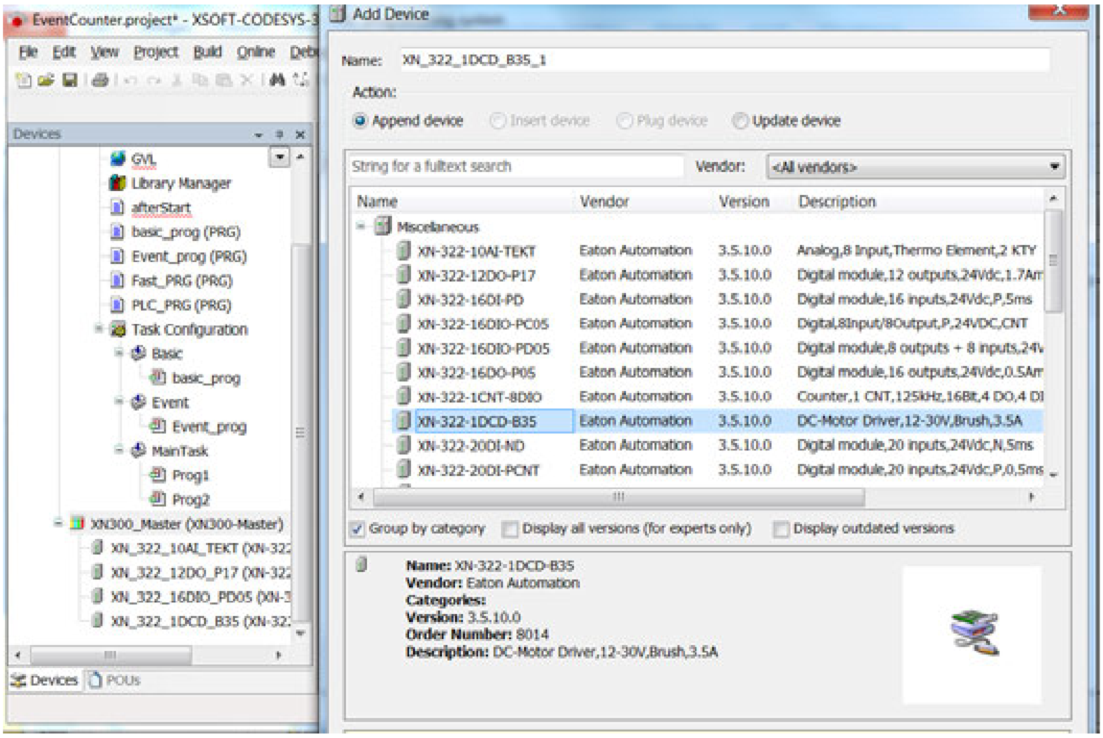
If you have expanded the XC300 controller with XN300 slice modules and mounted them as a system block on the DIN rail, you can configure it in XSOFT-CODESYS-3 as follows:

- ▶ Ensure that all XN300 slice modules are connected and supplied with power.
- ▶ Open your XSOFT-CODESYS-3 project.
- ▶ Select the device “Device (XC-303)” in the device tree diagram.
- ▶ In the context menu, select the “Add Device...” command.
- ▶ In the “Add Device” window, go to “Miscellaneous” and select “XN300 Master.”
- ▶ Confirm the selection by clicking the “Add Device” button.

The XN300 master appears in the device tree diagram.

- ▶ Select “XN300 master” in the device tree diagram.
- ▶ In the “Add Device” window, go to “Miscellaneous” and select the XN300 slice modules that you installed with XC300 as one system block. Select the XN300 slice modules in the same order as they were physically connected to XC300.

All devices are now shown in the device tree diagram.



Name: XN_322_1DCD_B35_1

Action:

● Append device ○ Insert device ○ Plug device ○ Update device

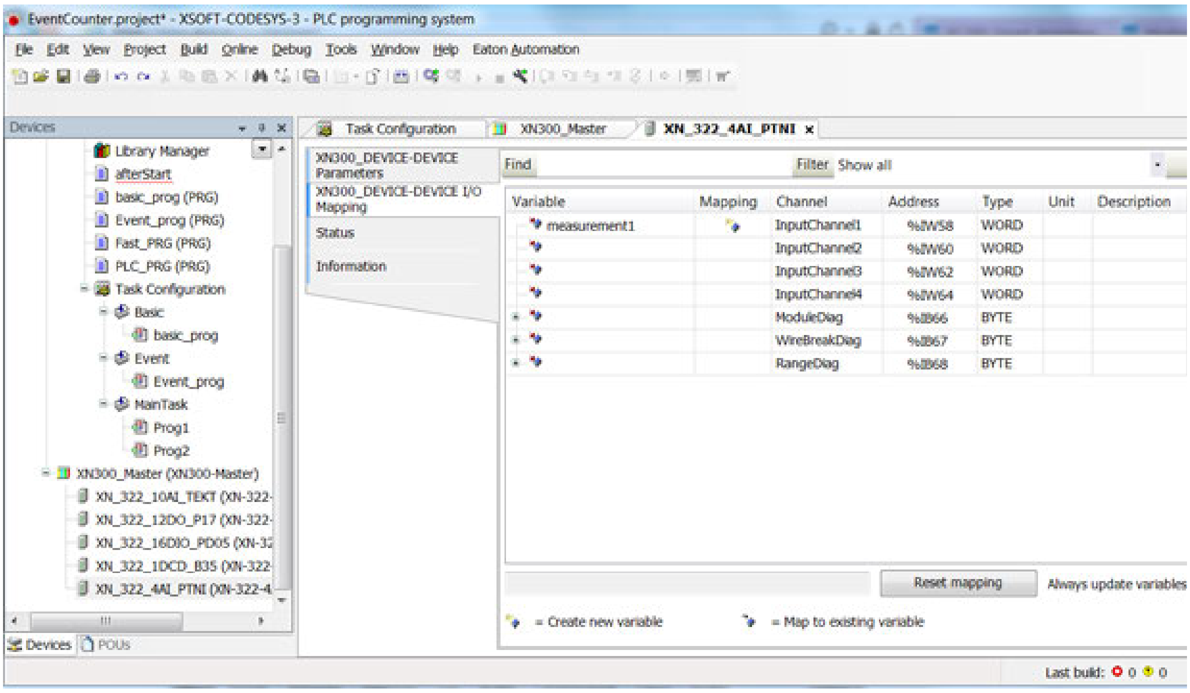
String for a fulltext search Vendor: <All vendors>

Name	Vendor	Version	Description
Miscellaneous			
XN-322-10AI-TEKT	Eaton Automation	3.5.10.0	Analog,8 Input,Thermo Element,2 KTY
XN-322-12DO-P17	Eaton Automation	3.5.10.0	Digital module,12 outputs,24Vdc,1.7Am
XN-322-16DI-PD	Eaton Automation	3.5.10.0	Digital module,16 inputs,24Vdc,P,5ms
XN-322-16DIO-PC05	Eaton Automation	3.5.10.0	Digital,8Input/8Output,P,24VDC,CNT
XN-322-16DIO-PD05	Eaton Automation	3.5.10.0	Digital module,8 outputs + 8 inputs,24V
XN-322-16DO-P05	Eaton Automation	3.5.10.0	Digital module,16 outputs,24Vdc,0.5Am
XN-322-1CNT-8DIO	Eaton Automation	3.5.10.0	Counter,1 CNT,125kHz,16Bit,4 DO,4 DI
XN-322-1DCD-B35	Eaton Automation	3.5.10.0	DC-Motor Driver,12-30V,Brush,3.5A
XN-322-20DI-ND	Eaton Automation	3.5.10.0	Digital module,20 inputs,24Vdc,N,5ms
XN-322-20DI-PCNT	Eaton Automation	3.5.10.0	Digital module,20 inputs,24Vdc,P,0,5ms

☑ Group by category □ Display all versions (for experts only) □ Display outdated versions

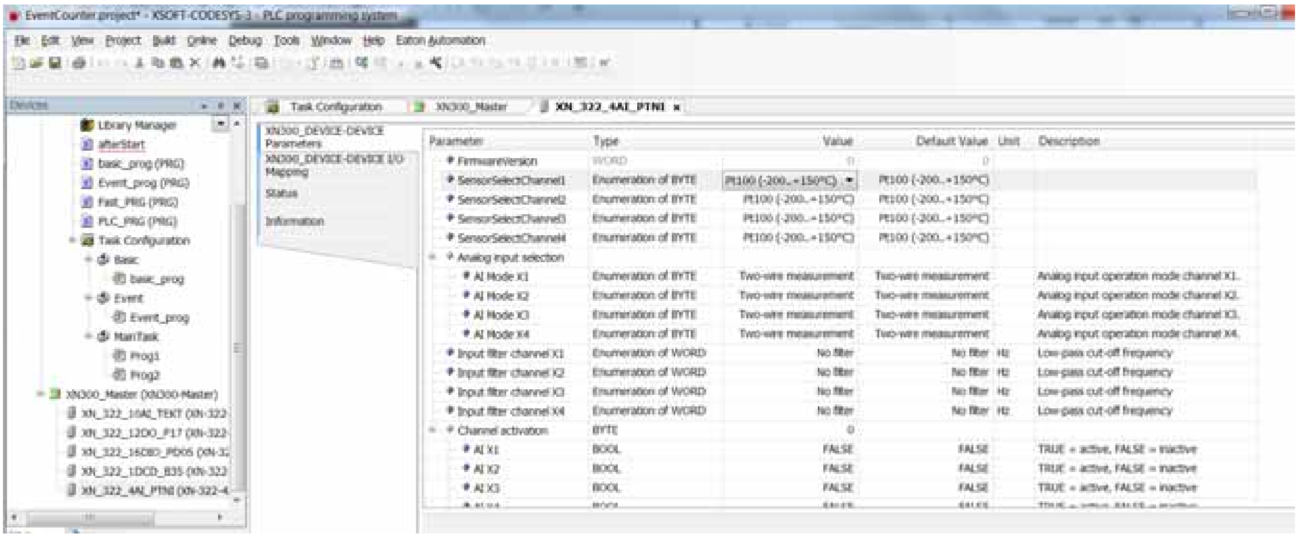
Name: XN-322-1DCD-B35
Vendor: Eaton Automation
Categories:
Version: 3.5.10.0
Order Number: 8014
Description: DC-Motor Driver,12-30V,Brush,3.5A

Figure 61: Device tree diagram with XN300 slice modules



Variable	Mapping	Channel	Address	Type	Unit	Description
measurement1		InputChannel1	%IW58	WORD		
		InputChannel2	%IW60	WORD		
		InputChannel3	%IW62	WORD		
		InputChannel4	%IW64	WORD		
		ModuleDiag	%IB66	BYTE		
		WireBreakDiag	%IB67	BYTE		
		RangeDiag	%IB68	BYTE		

Figure 62: Declaring values



Parameter	Type	Value	Default Value	Unit	Description
FirmwareVersion	WORD	0	0		
SensorSelectChannel1	Enumeration of BYTE	Pt100 (-200...+150°C)	Pt100 (-200...+150°C)		
SensorSelectChannel2	Enumeration of BYTE	Pt100 (-200...+150°C)	Pt100 (-200...+150°C)		
SensorSelectChannel3	Enumeration of BYTE	Pt100 (-200...+150°C)	Pt100 (-200...+150°C)		
SensorSelectChannel4	Enumeration of BYTE	Pt100 (-200...+150°C)	Pt100 (-200...+150°C)		
Analog input selection					
AI Mode X1	Enumeration of BYTE	Two-wire measurement	Two-wire measurement		Analog input operation mode channel X1.
AI Mode X2	Enumeration of BYTE	Two-wire measurement	Two-wire measurement		Analog input operation mode channel X2.
AI Mode X3	Enumeration of BYTE	Two-wire measurement	Two-wire measurement		Analog input operation mode channel X3.
AI Mode X4	Enumeration of BYTE	Two-wire measurement	Two-wire measurement		Analog input operation mode channel X4.
Input filter channel X1	Enumeration of WORD	No filter	No filter	Hz	Low-pass cut-off frequency
Input filter channel X2	Enumeration of WORD	No filter	No filter	Hz	Low-pass cut-off frequency
Input filter channel X3	Enumeration of WORD	No filter	No filter	Hz	Low-pass cut-off frequency
Input filter channel X4	Enumeration of WORD	No filter	No filter	Hz	Low-pass cut-off frequency
Channel activation	BYTE	0			
AI X1	BOOL	FALSE	FALSE		TRUE = active, FALSE = inactive
AI X2	BOOL	FALSE	FALSE		TRUE = active, FALSE = inactive
AI X3	BOOL	FALSE	FALSE		TRUE = active, FALSE = inactive

Figure 63: Parameterize XN300 slice modules

8.2 Remote expansion

→

→ Section “ The XC300 Modular control supports decentralized expansion via fieldbus interfaces. The following protocols are supported for communication on the respective interface:”, page 12
The following chapters describe the individual fieldbus interfaces.

9 CAN

The components of the I/O system XN300 can be networked via the Eaton CAN-Gateway XN-312-GW-CAN with the XC300 Modular control into a CAN node.
Each CAN node allows expansion through the I/O system XN300 by up to 32 XN300 slice modules.

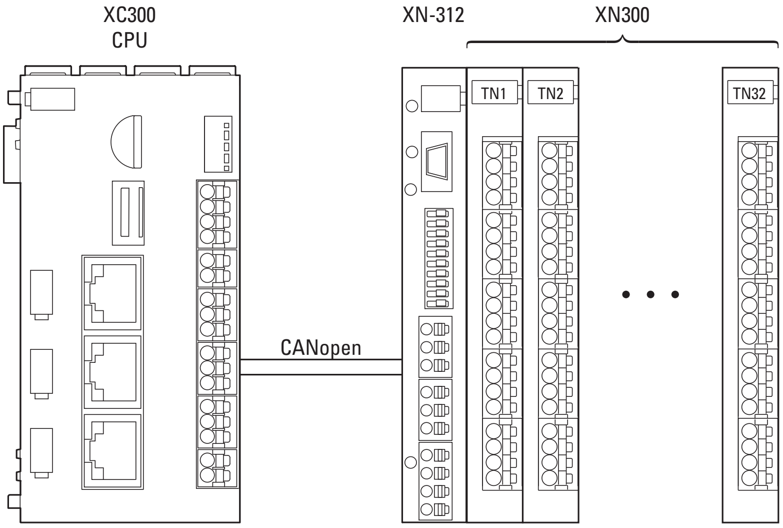


Figure 64: CAN node as decentralized expansion of the XC300 Modular control

For this purpose the system block, consisting of a gateway and XN300 slice moduless, is mounted on a DIN rail. The system block and controller can be mounted separately. The XN300 system provides a wide range of XN300 slice moduless, which offer special functions in addition to inputs and outputs, such as the weighing module or the motor driver module.

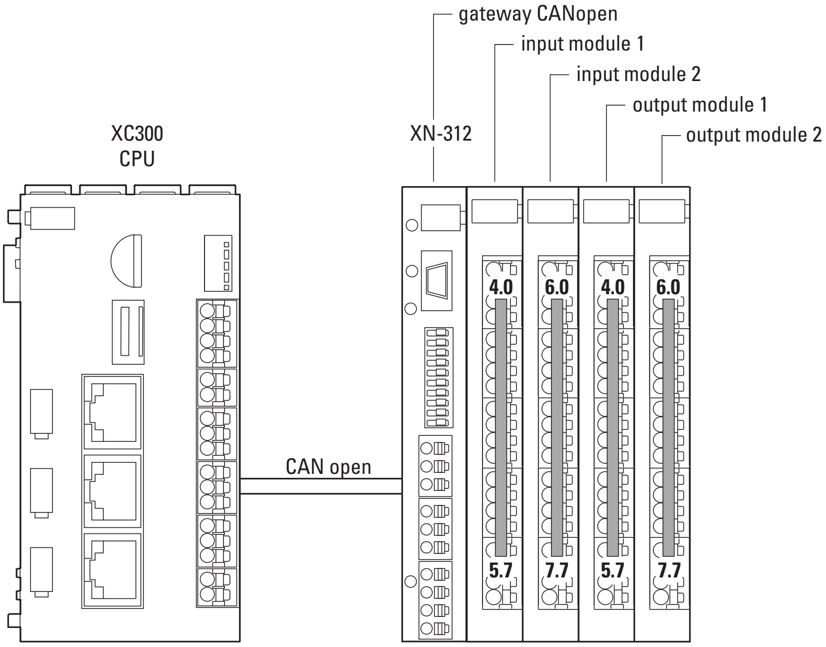
→ An overview of the XN300 slice modules can be found in the “XN300 slice modules” manual.

9.1 Configuring and commissioning CANopen

The XC300 Modular control can communicate both via CAN and CANopen. The following describes communication via CANopen.

Any CAN gateway can be connected to the CAN master via an EDS file.

The following shows an example of how the CANopen gateway XN-312-GW-CAN, including XN300 slice modules, is connected to the CANopen master.



→ If you are commissioning the fieldbus for the first time, you must first set up the relevant PLC programming environment.

Initial Commissioning

The following steps must be carried out:

- ▶ Connect the XC300 Modular control to the programming PC.
- ▶ Connect the XC300 to the CAN-Gateway XN-312-GW-CAN via a CAN bus cable.
- ▶ Set the DIP switch for the CAN.
→ section 2.10.4, “CAN/easyNet”, page 45
- ▶ Set the node ID and baud rate.
- ▶ Save the configuration of the I/O system XN300 components connected to the CAN-Gateway XN-312-GW-CAN in the “config check” in the gateway.
- ▶ Open your project in XSOFT-CODESYS-3 or create a new one,
→ section 5.2, “Creating a new project”, page 78.

9.1.1 Adding the CAN manager

- ▶ Select the device “Device (XC-303)” in the device tree diagram.
- ▶ Right-click to open the context menu and select “Add Device...”

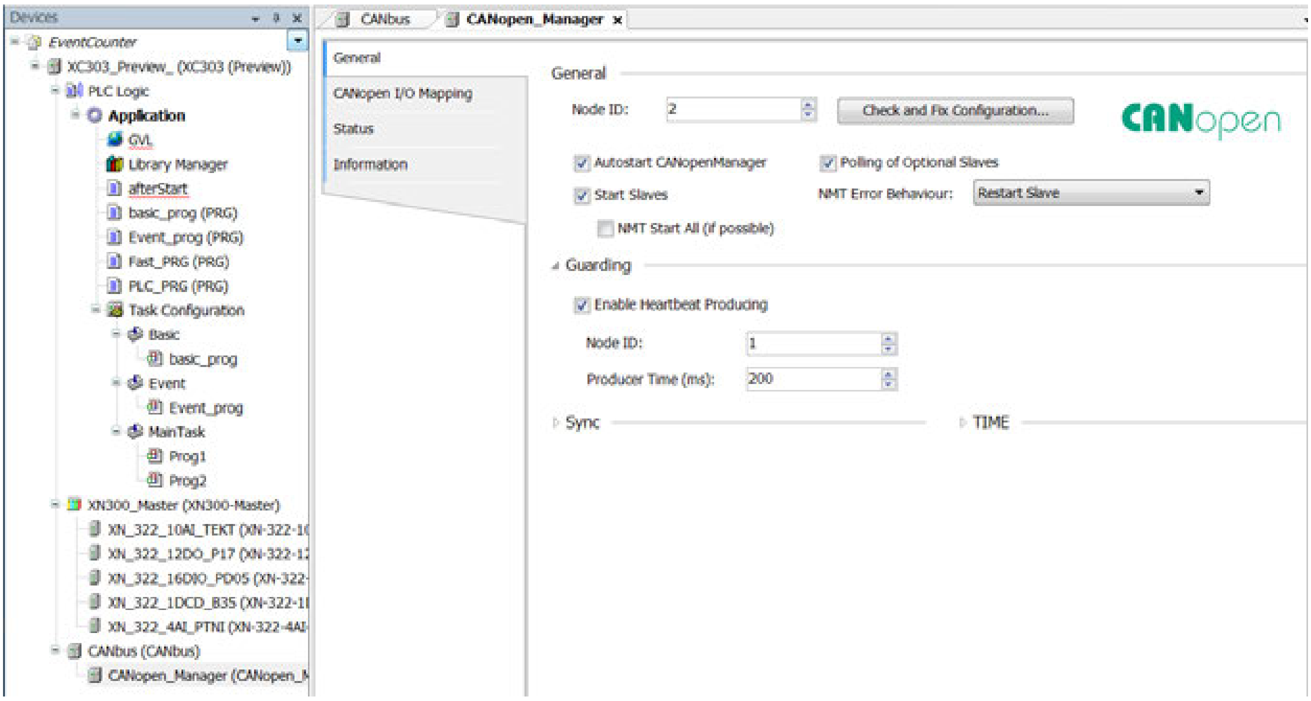
The “Add Device” dialog box will appear. You can leave the dialog box open.

- ▶ In the “Add Device” window, go to “Fieldbuses” and select the “CANbus.”
- ▶ Confirm the selection by clicking the “Add Device” button.

The CANbus appears in the configuration tree diagram.

- ▶ Select “CANbus” in the configuration tree diagram.
- ▶ Select the baud rate.
- ▶ In the “Add Device” window, go to “Fieldbuses” and select the “CANopen_Manager.”

The CANopen_Manager appears in the configuration tree diagram. EDS files for the fieldbus subscribers from EATON are automatically installed.



General

Node ID: 2

☑ Autostart CANopenManager

☑ Polling of Optional Slaves

☑ Start Slaves

NMT Error Behaviour: Restart Slave

□ NMT Start All (if possible)

Guarding

☑ Enable Heartbeat Producing

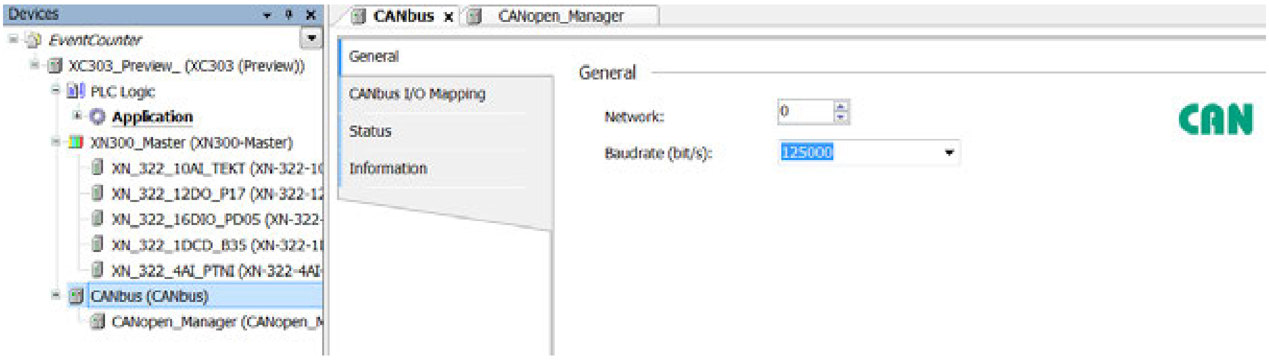
Node ID: 1

Producer Time (ms): 200

Figure 65: CANopen parameters for the CANopen_Manager

9.1.2 Setting the Baud Rate via CANbus

- ▶ In the configuration tree diagram, double-click “Devices” on <CANbus (CANbus)>.
- ▶ Open the “CANbus” tab and select the Baudrate, e.g., <125000>.


Devices
EventCounter
XC303_Preview_ (XC303 (Preview))
PLC Logic
Application
XN300_Master (XN300-Master)
XN_322_10AI_TEKT (XN-322-1(
XN_322_12DO_P17 (XN-322-12
XN_322_16DIO_PD05 (XN-322-
XN_322_1DCD_B35 (XN-322-1I
XN_322_4AI_PTNI (XN-322-4AI-
CANbus (CANbus)
CANopen_Manager (CANopen_M
CANbus
CANopen_Manager
General
CANbus I/O Mapping
Status
Information
General
CAN

Network: 0

Baudrate (bit/s): 125000

Figure 66: CANopen_manager baud rate

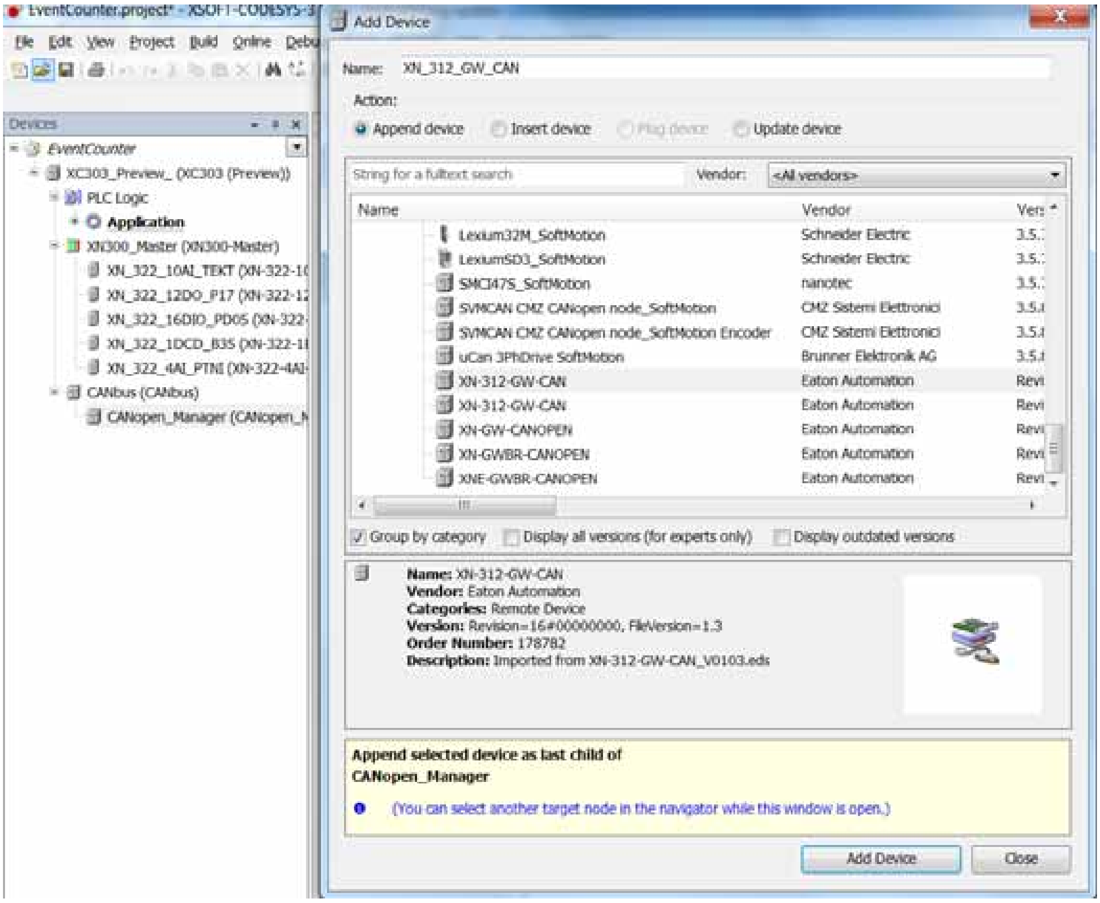
→ Detailed information can be found in the online help for XSOFT-CODESYS-3, which can be opened by pressing the <F1> function key.

- ▶ EDS files for fieldbus subscribers from other companies must be additionally installed in XSOFT-CODESYS-3, see → section 9.1.5.2, “Installing the EDS File”, page 105.

9.1.3 CAN-Gateway XN-312-GW-CAN Integrating

- ▶ To add the CANopen gateway to your configuration, right-click on “CANopen_Manager (CANopen_Manager)”:
- ▶ Then select <Append device...>.

The list will show all the devices that can be configured:


EventCounter.project* - XSOFT-CODESYS-3
File Edit View Project Build Online Debu
Devices
EventCounter
XC303_Preview_ (XC303 (Preview))
PLC Logic
Application
XN300_Master (XN300-Master)
XN_322_10AI_TEKT (XN-322-1
XN_322_12DO_P17 (XN-322-1
XN_322_16DIO_PD05 (XN-322-
XN_322_1DCD_B35 (XN-322-1
XN_322_4AI_PTNI (XN-322-4AI-
CANbus (CANbus)
CANopen_Manager (CANopen_M
Add Device
Add Device
Close

Name: XN_312_GW_CAN

Action:

● Append device ○ Insert device ○ Plug device ○ Update device

String for a fulltext search Vendor: <All vendors>

Name	Vendor	Vers
Lexium32M_SoftMotion	Schneider Electric	3.5.
LexiumSD3_SoftMotion	Schneider Electric	3.5.
SMCI47S_SoftMotion	nanotec	3.5.
SVMCAN CMZ CANopen node_SoftMotion	CMZ Sistemi Elettronici	3.5.
SVMCAN CMZ CANopen node_SoftMotion Encoder	CMZ Sistemi Elettronici	3.5.
uCan 3PhDrive SoftMotion	Brunner Elektronik AG	3.5.
XN-312-GW-CAN	Eaton Automation	Revi
XN-312-GW-CAN	Eaton Automation	Revi
XN-GW-CANOPEN	Eaton Automation	Revi
XN-GWBR-CANOPEN	Eaton Automation	Revi
XNE-GWBR-CANOPEN	Eaton Automation	Revi

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: XN-312-GW-CAN
Vendor: Eaton Automation
Categories: Remote Device
Version: Revision=16#00000000, FileVersion=1.3
Order Number: 178782
Description: Imported from XN-312-GW-CAN_V0103.eds

Append selected device as last child of
CANopen_Manager

(You can select another target node in the navigator while this window is open.)

Figure 67: Adding a CANopen device

- ▶ Select the corresponding CAN-Gateway XN-312-GW-CAN.
- ▶ Select <**Add device**> again.

The selected CAN-Gateway XN-312-GW-CAN will be added to the configuration tree diagram.

- ▶ If the desired CAN device is not found in the list, update the X-SOFT-CODESYS-3 version you are using or load the appropriate EDS file, → section 9.1.5.2, “Installing the EDS File”, page 105.
- ▶ Check the settings for the parameters for XN_312_GW_CAN by double-clicking in the configuration tree diagram on “XN_312_GW_CAN (XN-312-GW-CAN).”
- ▶ Select the “XN_312_GW_CAN/General” tab and check the “Node ID” for the device.

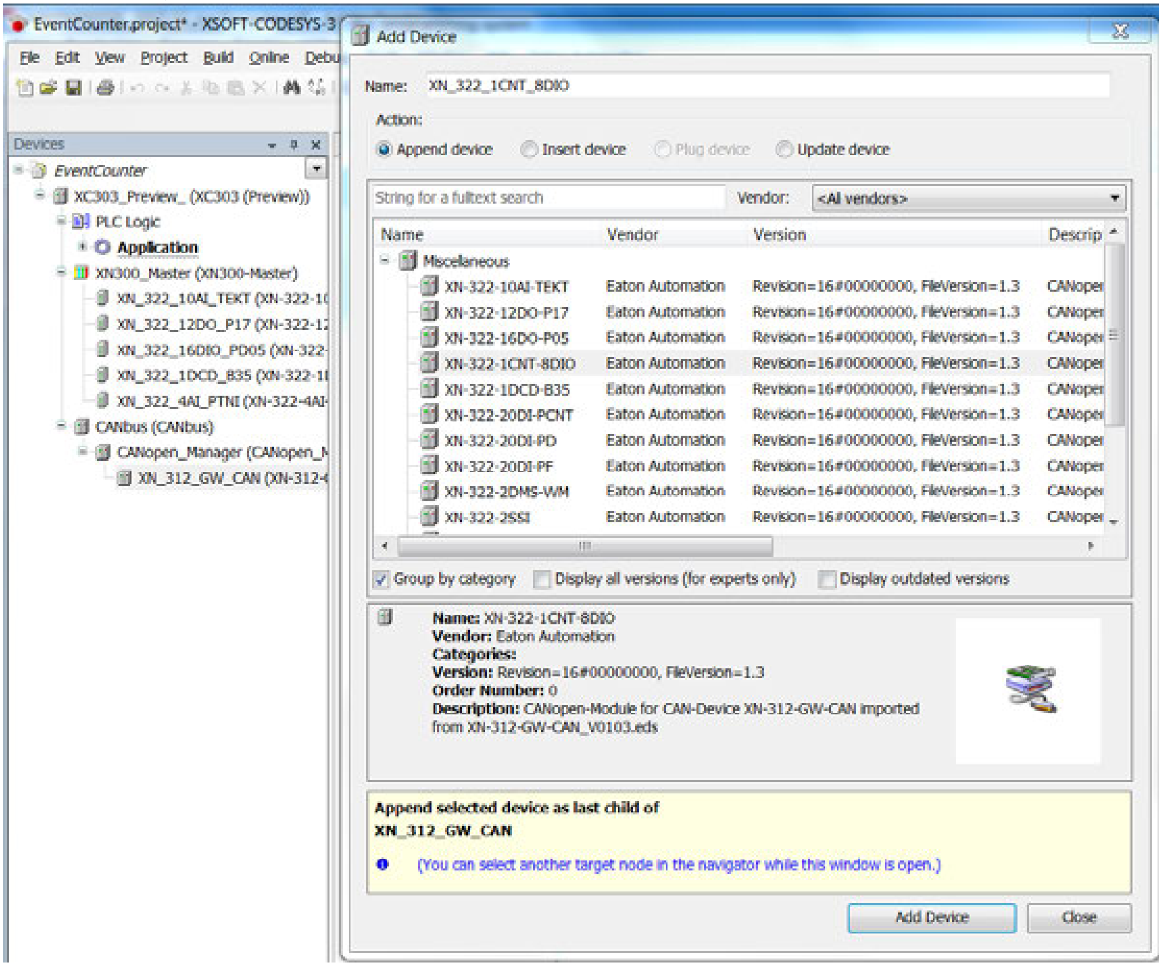
→ All other gateway settings and information on mapping can be found in the manual for the CAN-Gateway XN-312-GW-CAN.

9.1.4 Attach XN300 slice modules

▶ To configure the system block, right-click on the CANopen device, e.g., “XN_312_GW_CAN (XN-312-GW-CAN)” in the “Devices” configuration structure.

▶ Then select <Add Device>.

The “Add Device” dialog box will appear if it is not open already. The dialog box will show a list with all the XN300 slice modules that can be configured.


EventCounter.project* - XSOFT-CODESYS-3
File Edit View Project Build Online Debu
Devices
EventCounter
XC303_Preview_ (XC303 (Preview))
PLC Logic
Application
XN300_Master (XN300-Master)
XN_322_10AI_TEKT (XN-322-1(
XN_322_12DO_P17 (XN-322-12
XN_322_16DIO_PD05 (XN-322-
XN_322_1DCD_B35 (XN-322-1I
XN_322_4AI_PTNI (XN-322-4AI-
CANbus (CANbus)
CANopen_Manager (CANopen_M
XN_312_GW_CAN (XN-312-(
Add Device
Add Device
Close

Name: XN_322_1CNT_8DIO

Action:

● Append device ○ Insert device ○ Plug device ○ Update device

String for a fulltext search Vendor: <All vendors>

Name	Vendor	Version	Descrip
Miscellaneous			
XN-322-10AI-TEKT	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper
XN-322-12DO-P17	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper
XN-322-16DO-P05	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper
XN-322-1CNT-8DIO	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper
XN-322-1DCD-B35	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper
XN-322-20DI-PCNT	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper
XN-322-20DI-PD	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper
XN-322-20DI-PF	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper
XN-322-2DMS-WM	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper
XN-322-2SSI	Eaton Automation	Revision=16#00000000, FileVersion=1.3	CANoper

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: XN-322-1CNT-8DIO
Vendor: Eaton Automation
Categories:
Version: Revision=16#00000000, FileVersion=1.3
Order Number: 0
Description: CANopen-Module for CAN-Device XN-312-GW-CAN imported from XN-312-GW-CAN_V0103.eds

Append selected device as last child of
XN_312_GW_CAN

(You can select another target node in the navigator while this window is open.)

Figure 68: “Add Device” window for CAN module selection for the XN-312-GW-CAN

▶ Select the desired XN300 slice module and then select <Add Device>. The XN300 slice module will be added to the “Devices” configuration tree diagram.

→ The order of the XN300 slice modules in the configuration tree diagram can be changed as desired by dragging and dropping elements.

A maximum of 32 XN300 slice modules can be added to the CANopen device.

All devices are now shown in the device tree diagram.

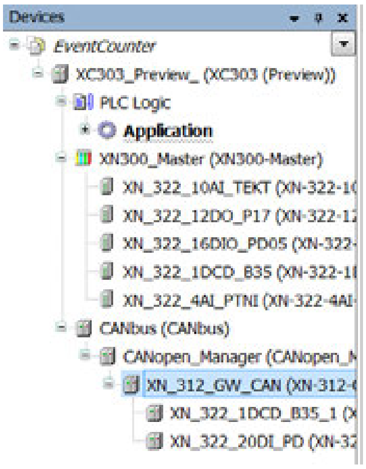

Devices
EventCounter
XC303_Preview_ (XC303 (Preview))
PLC Logic
Application
XN300_Master (XN300-Master)
XN_322_10AI_TEKT (XN-322-1(
XN_322_12DO_P17 (XN-322-1
XN_322_16DIO_PD05 (XN-322-
XN_322_1DCD_B35 (XN-322-1
XN_322_4AI_PTNI (XN-322-4AI
CANbus (CANbus)
CANopen_Manager (CANopen_M
XN_312_GW_CAN (XN-312-
XN_322_1DCD_B35_1 (X
XN_322_20DI_PD (XN-32

Figure 69: XC300 with CANopen master,
CAN-Gateway XN-312-GW-CAN and 4 XN300 slice moduless

If you are using analog modules, observe the following requirements:

- If default mapping (index 0x6401/x) is used for analog modules, PDO-based data transmission for these I/O slice modules must be enabled by setting the value of object index 0x6423, subindex 0 to TRUE.
- Regulating the bus load on the CAN fieldbus
  The following procedure is recommended in order to prevent the XN-312-GW-CAN gateway from generating a heavy load on the CAN fieldbus:
  - Use internal module filters,
    see „XN-300 slice modules“, MN050002-EN“ manual.
  - If using the default mapping (index 0x6401/x), a transmission filter can be configured for each individual payload data object; SDO objects: AI_DELTA_VALUE, index 0x6426/x).
    When this filter is configured, the gateway will update and transmit the corresponding value only if this value is different from the last transmitted value by more than the configured delta.

By setting an inhibit time or selecting a synchronous transmission type, TPDO properties can be configured in such a way as to avoid impermissible bus loads.

ATTENTION

The short data refresh cycle times in the XN300 system can result in the CAN transmission path being subjected to high loads. Accordingly, it is recommended to limit the data being transmitted by using the various PDO properties available (e.g., inhibit time), especially in the case of analog input and technology modules featuring counters.

9.1.5 Description Files for CANopen

The gateway XN-312-GW-CAN is integrated into the CANopen structure with the aid of a standardized EDS file (Electronic Data Sheet).

This EDS file lists all objects with the associated sub-indexes and the corresponding entries.
The EDS file is automatically installed in XSOFT-CODESYS-3 with the CANopen master.

9.1.5.1 Standard EDS files

Make sure that you are working with the latest version of the standard EDS files in your PLC programming environment. If subscribers are not listed, check whether there are updates for the files in the Download Center, → page 8. To find it quickly, simply enter “XN300” or “EDS” as a search term.

9.1.5.2 Installing the EDS File

If you want to install an updated EDS file or an EDS file from another device manufacturer, proceed as follows:

- Select and download the latest version of the EDS file.
- Save and unzip the *.zip file to a project folder of your choice.
- In XSOFT-CODESYS-3, select <Tools | Device Repository …>.
- Click the “Install...” button.
- In the Explorer, select the desired EDS file.
- Confirm by clicking on “Open.”
- Confirm the prompt “The device has been successfully installed” by clicking “OK.”

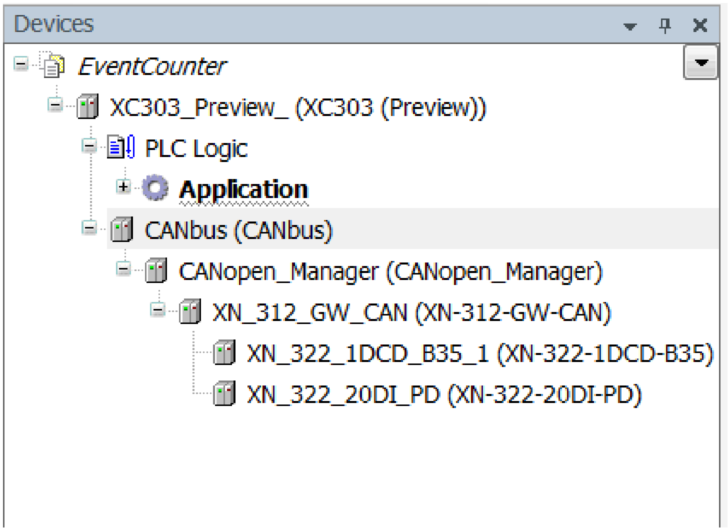

Devices
EventCounter
XC303_Preview_ (XC303 (Preview))
PLC Logic
Application
CANbus (CANbus)
CANopen_Manager (CANopen_Manager)
XN_312_GW_CAN (XN-312-GW-CAN)
XN_322_1DCD_B35_1 (XN-322-1DCD-B35)
XN_322_20DI_PD (XN-322-20DI-PD)

Figure 70: “Devices” pane with CANopen manager in the configuration structure

- Double-click on “CANopen_Manager” (CANopen manager).
- Check the settings under the “CANopen_Manager” tab.

9.1.6 Integrating libraries for CANopen communication

By default, all the libraries required for CANopen communication will be automatically added after the CAN master is added and the project is compiled for the first time.

- 3S CANopenStack
- CAA CiA405
- CANbusDevice
- EA_CAN types (when using EATON CAN devices)

9.1.7 LEDs on the device

The status indicator LEDs under the front panel report the status.
→ figure 33, page 54

	Status LED		Description
	red	green	
CAN1 CAN2	–	–	No bus activity
	–	flashing	Bus activity OK (CAN), FLASH for each telegram
	ON	–	Bus error
	flashing	flashing	WARNING

1) flashing: The flashing frequency may differ based on the transfer rate on the bus.

Each successfully sent/received CAN telegram triggers the LEDs.

10 EtherCAT

The following describes how the XC300 Modular control controller is configured as an EtherCAT master in order to operate decentralized EtherCAT subscribers.

EU5C-SWD-ETHERCAT is used here as an example of an EtherCAT subscriber. This EtherCAT gateway acts as a coordinator for other SWD subscribers and their input and output functions.

A target and project configuration must be saved in the gateway in order for the EU5C-SWD-ETHERCAT gateway to exchange data with the XC300 controller. How these configurations are applied and stored in the device is described in the “ Gateway EU5C-SWD-ETHERCAT” manual. This manual can be downloaded from the internet as a PDF from: www.hilscher.com <Products | Partner Products | SmartWire-DT | EU5C-SWD-ETHERCAT | Downloads>

Requirements:

- PC with installed PLC programming environment XSOFT-CODESYS-3 V3.5.10 or higher and Ethernet adapter.
- The Ethernet adapter for the PC is connected to the XC300 controller via Ethernet.
- XC300 is connected to the EU5C-SWD-ETHERCAT gateway.
- The gateway is connected to a SmartWire-DT network.
- The gateway contains a valid SmartWire-DT configuration (SWD LED on the device lights up green constantly).
- You have access to the device description files for the EU5C-SWD-ETHERCAT gateway.

10.1 Integrating the EtherCAT Master

▶ Select the device “Device (XC-303)” in the device tree diagram.
▶ Right-click to open the context menu and select “Add Device...”

The “Add Device” dialog box will appear. You can leave the dialog box open.

▶ In the “Add Device” window, go to “Fieldbuses” and select the “EtherCAT | Master | EtherCAT_Master.”
▶ Confirm the selection by clicking the “Add Device” button.

The EtherCAT master appears in the configuration tree diagram.

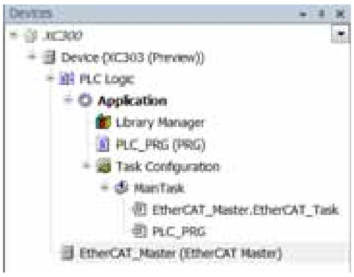
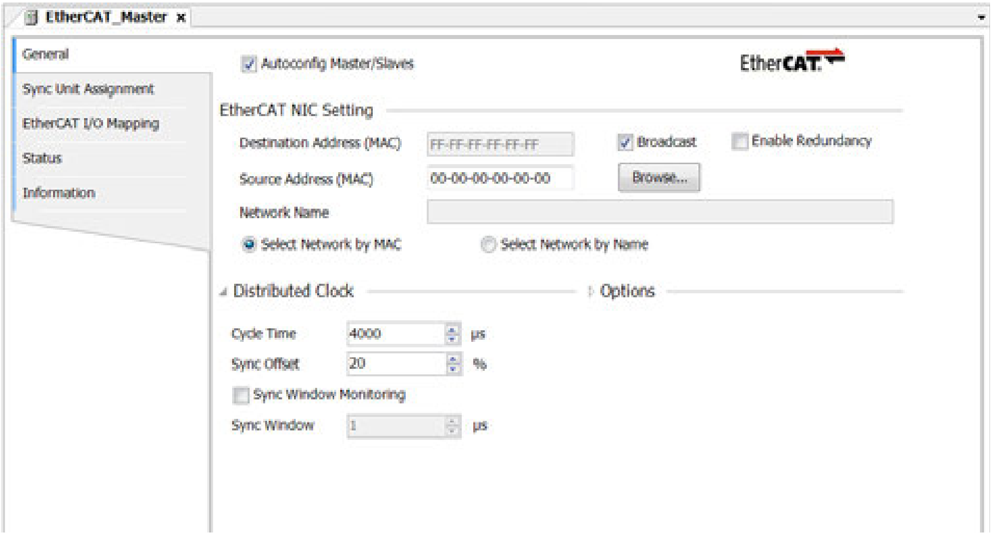


Figure 71: Device tree diagram with EtherCAT master

10.2 Configuring the EtherCAT Master

- ▶ Double-click "EtherCAT_Master (EtherCAT Master)" in the device tree diagram for the controller configuration.
- ▶ In the "General" tab, click on the <Browse> button.



☑ Autoconfig Master/Slaves

EtherCAT NIC Setting

Destination Address (MAC) FF-FF-FF-FF-FF-FF ☑ Broadcast □ Enable Redundancy

Source Address (MAC) 00-00-00-00-00-00

Network Name

● Select Network by MAC ○ Select Network by Name

Distributed Clock

Options

Cycle Time 4000 µs

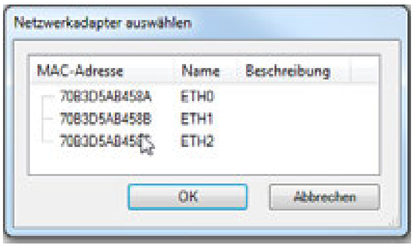
Sync Offset 20 %

□ Sync Window Monitoring

Sync Window 1 µs

Figure 72: EtherCAT master parameters

XSOFT-CODESYS-3 reports the MAC addresses and names of the existing Ethernet interfaces of the XC300 controller.

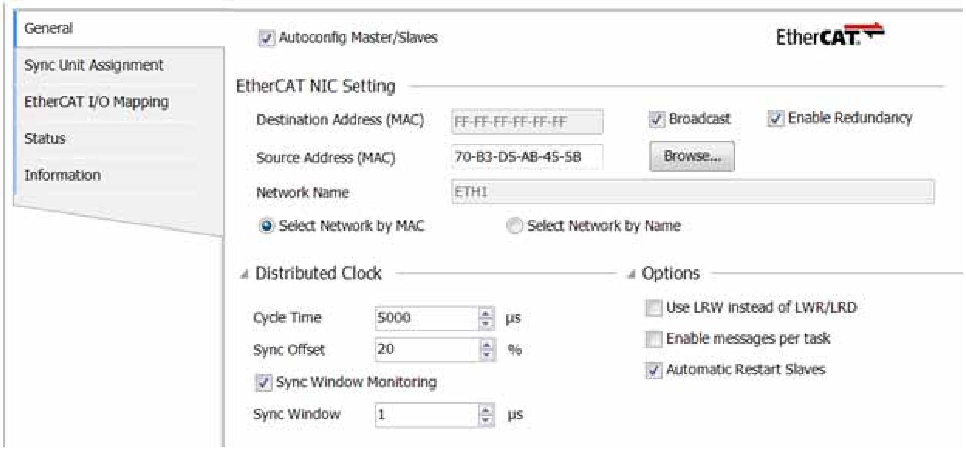


Netzwerkadapter auswählen

MAC-Adresse	Name	Beschreibung
70B3D5AB458A	ETH0	
70B3D5AB458B	ETH1	
70B3D5AB45[illegible]	ETH2	

Figure 73: List of all found Ethernet interfaces of the XC-303-C32-002 controller

- ▶ Select the Ethernet interface to which you have connected or will connect the EtherCAT gateway.
- ▶ In the "General" tab, select the setting "Choose Network by Name." This allows the controller to connect to the network using its name, such as "ETH2" regardless of the MAC address.
- ▶ Activate the "Automatic Restart Slaves" option so that communication with the subscribers starts automatically.



☑ Autoconfig Master/Slaves

EtherCAT NIC Setting

Destination Address (MAC) FF-FF-FF-FF-FF-FF ☑ Broadcast ☑ Enable Redundancy

Source Address (MAC) 70-B3-D5-AB-45-5B Browse...

Network Name ETH1

● Select Network by MAC ○ Select Network by Name

Distributed Clock

Cycle Time 5000 µs

Sync Offset 20 %

☑ Sync Window Monitoring

Sync Window 1 µs

Options

□ Use LRW instead of LWR/LRD

□ Enable messages per task

☑ Automatic Restart Slaves

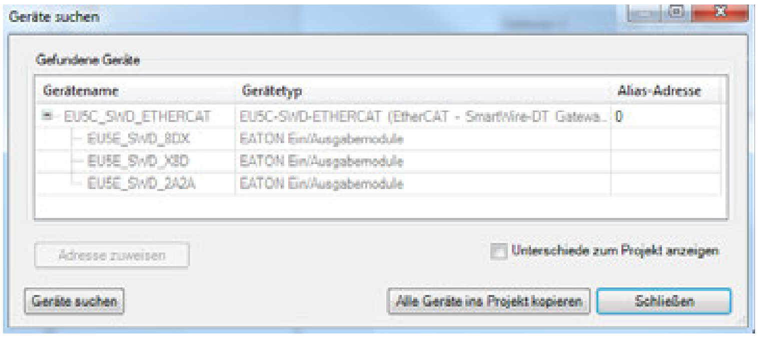
Figure 74: EtherCAT Master settings

10.3 Searching for and Integrating EtherCAT Subscribers

To connect the XC300 controller to an EtherCAT subscriber, such as to the EU5C-SWD-ETHERCAT gateway, proceed as follows:

- ▶ Log into the XC300 controller by entering the <Online|Login> command sequence in the XSOFT-CODESYS-3 software.
- ▶ Right-click on “EtherCAT_Master (...)”
- ▶ Then select <Scan Devices…>.

The PLC programming software performs an automatic search for all devices connected to the EtherCAT master XC300. The results window lists all found devices.



Gefundene Geräte

Gerätename	Gerätetyp	Alias-Adresse
EU5C_SWD_ETHERCAT	EU5C-SWD-ETHERCAT (EtherCAT - SmartWire-DT Gatewa...	0
EU5E_SWD_8DX	EATON Ein/Ausgabemodule	
EU5E_SWD_X8D	EATON Ein/Ausgabemodule	
EU5E_SWD_2A2A	EATON Ein/Ausgabemodule	

□ Unterschiede zum Projekt anzeigen

Figure 75: Results list of search for devices

- ▶ Click the <Copy All Devices to Project> button.

The devices are displayed in the device tree diagram for the controller configuration. SWD subscribers that are already connected to the EtherCAT gateway are also automatically shown in the tree diagram, e.g., EU5E_SWD_8DX.

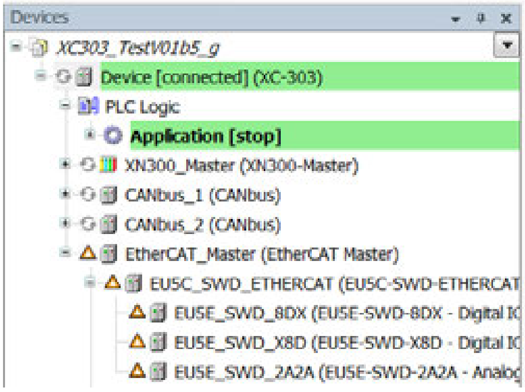

Devices
XC303_TestV01b5_g
Device [connected] (XC-303)
PLC Logic
Application [stop]
XN300_Master (XN300-Master)
CANbus_1 (CANbus)
CANbus_2 (CANbus)
EtherCAT_Master (EtherCAT Master)
EU5C_SWD_ETHERCAT (EU5C-SWD-ETHERCAT
EU5E_SWD_8DX (EU5E-SWD-8DX - Digital IC
EU5E_SWD_X8D (EU5E-SWD-X8D - Digital IC
EU5E_SWD_2A2A (EU5E-SWD-2A2A - Analog

Figure 76: Tree diagram with EU5C-SWD-ETHERCAT gateway and SWD subscribers

- ▶ If the desired EtherCAT subscriber is not found in the list, update the X-SOFT-CODESYS-3 version you are using or load the appropriate XML file, → section 10.6, “Description Files for EtherCAT”, page 112.
- ▶ Check the parameter settings for the EtherCAT gateway by double-clicking in the tree diagram on “EU5C-SWD-ETHERCAT (...)”

→ All other gateway settings and information on mapping can be found in the “Gateway EU5C-SWD-ETHERCAT” manual, see note → page 107, and in the XSOFT-CODESYS-3 online help.

10.4 Manually Integrating EtherCAT Subscribers

Alternatively to the description in → section 10.3, “Searching for and Integrating EtherCAT Subscribers”, page 109, you can also add EtherCAT subscribers to the XC300 controller manually. To do so, follow the steps below:

- ▶ Right-click on “EtherCAT_Master (...)” in the device tree diagram.
- ▶ Then select <Add Device…>.

The list will show all the devices that can be configured:

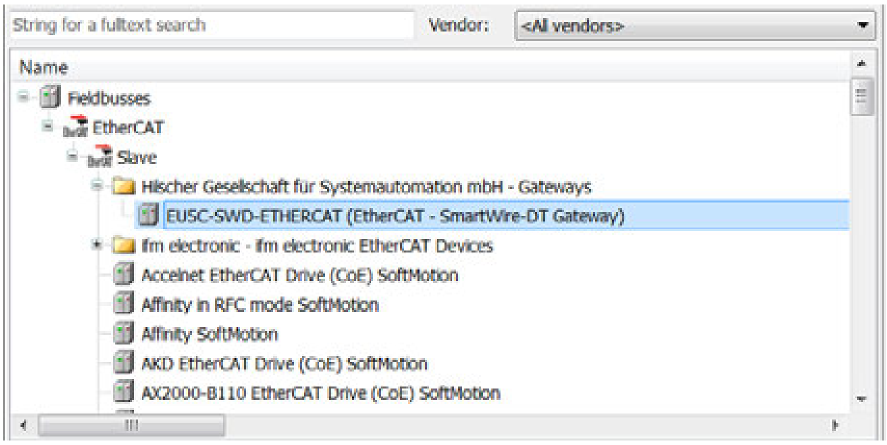

String for a fulltext search
Vendor:
<All vendors>
Name
Fieldbusses
EtherCAT
Slave
Hilscher Gesellschaft für Systemautomation mbH - Gateways
EU5C-SWD-ETHERCAT (EtherCAT - SmartWire-DT Gateway)
ifm electronic - ifm electronic EtherCAT Devices
Accelnet EtherCAT Drive (CoE) SoftMotion
Affinity in RFC mode SoftMotion
Affinity SoftMotion
AKD EtherCAT Drive (CoE) SoftMotion
AX2000-B110 EtherCAT Drive (CoE) SoftMotion

Figure 77: Adding an EtherCAT device

- ▶ Select the corresponding EtherCAT subscriber, e.g., in the “Hilscher...” directory, “EU5C-SWD-ETHERCAT.”
- ▶ In the “Add Device” window, select <Add Device> again.

The EtherCAT subscriber is added to the device tree diagram.

10.5 Adding SWD Subscribers

- ▶ Right-click on “EU5C_SWD_ETHERCAT(…)” in the device tree diagram and select <Add Device>.
- ▶ In the “Add Device” window, select the desired SWD subscribers, e.g., “DIL-SWD-32-001-…”

The SWD subscribers are displayed in the device tree diagram.

→ All other gateway settings and information on mapping can be found in the “Gateway EU5C-SWD-ETHERCAT” manual, see note → page 107, and in the XSOFT-CODESYS-3 online help.

10.6 Description Files for EtherCAT

Device description files in XML format are needed in order to integrate and configure EtherCAT subscribers in XSOFT-CODESYS-3. They contain standardized descriptions of the EtherCAT subscribers and their parameters.

The device description files are automatically installed in XSOFT-CODESYS-3 with the EtherCAT master.

10.6.1 Standard XML Files

Make sure that you are working with the latest version of the device description files in your PLC programming environment. If subscribers are not listed, check whether there are updates for the files in the Download Center for the device manufacturer in question.

Example:

As an example, the EU5C-SWD-ETHERCAT gateway will be integrated and configured as an EtherCAT subscriber. The gateway coordinates communication to and between other subscribers from SmartWire-DT.

Several device description files for the gateway and the respective SmartWire-DT subscribers are required in XSOFT-CODESYS-3 for this purpose. The current versions of the device description files are available in a zipped file in the AUTOMATION Download Center at the following link:

ftp://ftp.moeller.net/AUTOMATION/DOWNLOAD/CONFIGURATIONFILES_CFG_GSD_EDS/SEVERAL_PARTS/ETHERCAT/EU5C-SWD/

Save the CODESYS.zip file on your PC and unzip the file. It contains several device description files, such as

- Eaton_ETHERCAT_Modules.xml
- Dajo_EU5C_SWD-ETHERCAT_Modules.xml
- Hilscher_SWD_ETHERCAT_GW.xml
- Phoenix_ETHERCAT_Modules.xml
- Woehner_ETHERCAT_Modules.xml

Install the device description files in the PLC programming environment, e.g., XSOFT-CODESYS-3.

10.6.2 Installing an XML File

If you want to install an updated XML file or an XML file from another device manufacturer, proceed as follows:

- ▶ Select and download the latest version of the device description files.
- ▶ Save and unzip the *.zip file to a project folder of your choice.
- ▶ In XSOFT-CODESYS-3, select <Tools | Device Repository …>.
- ▶ Click the <Install…> button.
- ▶ In the Explorer, select the desired XML file from the project folder.
- ▶ Confirm by clicking on “Open.”
- ▶ Confirm the prompt “The device has been successfully installed” by clicking “OK.”
- ▶ Double-click “EtherCAT_Manager (EtherCAT Manager).”
- ▶ Check the settings in the “EtherCAT Manager” tab.

10.6.3 Integrating Libraries for EtherCAT Communication

By default, all the libraries required for communication will be automatically added after the EtherCAT master is added and the project is compiled for the first time.

- IODrvEtherCAT

11 Modbus

11.1 Configuring Modbus TCP

11.1.1 Integrating the Ethernet Adapter

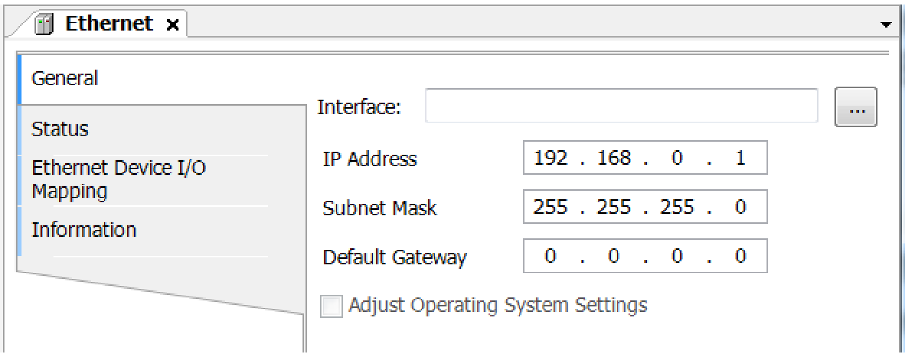
XSOFT-CODESYS-3 offers two different Ethernet connection options.
An Ethernet adapter can be used to define the protocol: EthernetIP or ModbusTCP.

- ▶ Select the device “Device (XC-303)” in the device tree diagram.
- ▶ Right-click to open the context menu and select “Add Device...”

The “Add Device” dialog box will appear. You can leave the dialog box open.

- ▶ In the “Add Device” window, go to “Fieldbuses” and select “Ethernet/IP | Ethernet Adapter | Ethernet.”
- ▶ Confirm the selection by clicking the “Add Device” button.

The Ethernet master appears in the configuration tree diagram.



Interface:

IP Address 192 . 168 . 0 . 1

Subnet Mask 255 . 255 . 255 . 0

Default Gateway 0 . 0 . 0 . 0

□ Adjust Operating System Settings

Figure 78: Ethernet master parameters

11.1.2 Integrating the ModbusTCP Master and Device

- ▶ To integrate the Modbus TCP master into your configuration, right-click “Ethernet (Ethernet).”
- ▶ Then select <Add Device…>.

The list shows the available Ethernet devices or Ethernet-based protocols such as Modbus.

- ▶ Select “Modbus | Modbus TCP Master | Modbus TCP Master.”
- ▶ Then select <Add Device…>
- ▶ Select “Modbus TCP Master” in the device tree diagram.
- ▶ Right-click to open the context menu and select “Add Device...”

- ▶ Select “Modbus TCP Slave.”
- ▶ Then select <Add Device…>

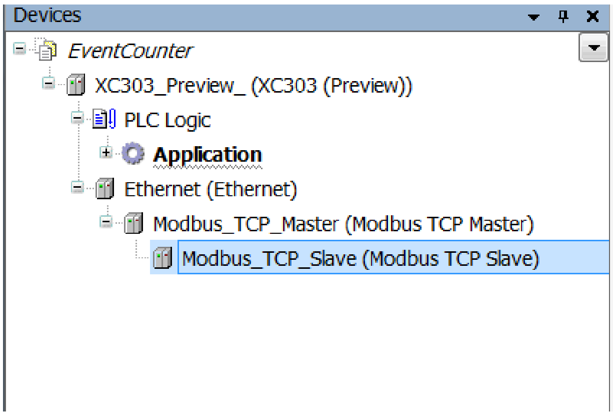

Devices
EventCounter
XC303_Preview_ (XC303 (Preview))
PLC Logic
Application
Ethernet (Ethernet)
Modbus_TCP_Master (Modbus TCP Master)
Modbus_TCP_Slave (Modbus TCP Slave)

Figure 79: Adding a Modbus TCP slave

The TCP master does not need any information as to which Modbus TCP slave is connected to the bus. It only needs the IP address and the register number of the Modbus TCP slave.

▶ Select the slave connection “Modbus_TCP_Slave” in the configuration tree diagram.
▶ Select <Insert device>.

The Modbus TCP device is added to the configuration tree diagram.

A maximum of 32 Modbus TCP masters can be added.

11.1.3 Integrating Libraries for Modbus TCP Communication

By default, all the libraries required for communication will be automatically added after the Modbus TCP master is added and the project is compiled for the first time.

- IODrvModbusTCP
- IODrvEthernet

11.2 Configuring Modbus RTU

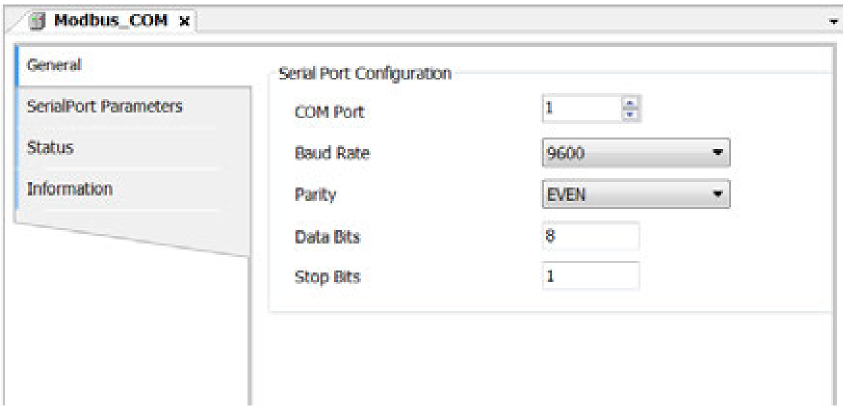
11.2.1 Integrating the Modbus Master

- ▶ Select the device “Device (XC-303)” in the device tree diagram.
- ▶ Right-click to open the context menu and select “Add Device...”

The “Add Device” dialog box will appear. You can leave the dialog box open.

- ▶ In the “Add Device” window, go to “Fieldbuses” and select “Modbus | Modbus Serial Port | Modbus COM Port.”
- ▶ Confirm the selection by clicking the “Add Device” button.

The Modbus COM port appears in the configuration tree diagram.



Serial Port Configuration

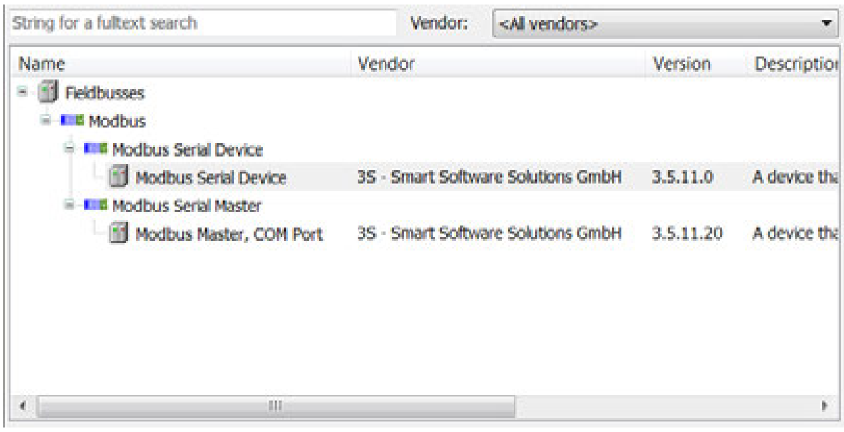
COM Port	1
Baud Rate	9600
Parity	EVEN
Data Bits	8
Stop Bits	1

Figure 80: Modbus COM port parameters

11.2.2 Integrating Modbus Subscribers

- ▶ To integrate the Modbus subscriber into your configuration, right-click “Modbus_COM_Port.”
- ▶ Then select <Add Device…>.

The list will show all the devices that can be configured:



String for a fulltext search   Vendor: <All vendors>

Name	Vendor	Version	Descriptio
Fieldbusses			
Modbus			
Modbus Serial Device			
Modbus Serial Device	3S - Smart Software Solutions GmbH	3.5.11.0	A device th
Modbus Serial Master			
Modbus Master, COM Port	3S - Smart Software Solutions GmbH	3.5.11.20	A device th

Figure 81: Adding a Modbus device

- ▶ Select the corresponding Modbus device, e.g., “Modbus Serial Device.”
- ▶ Select <Add Device> again.

The selected Modbus device will be added to the configuration tree diagram.

11.3 Integrating Libraries for Modbus RTU-Communication

By default, all the libraries required for communication will be automatically added after the Modbus master is added and the project is compiled for the first time.

- IODrvModbus

12 Remote Access to the Controller

The controller supports remote access via the following mechanisms

- SSH (Port: 22)
- SFTP (Port: 22)

These communication protocols are enabled in the controller by default. The PLC acts as a server, so it can also be operated from an external source. Corresponding entries in file /home/InitPLC can be used to disable the communication protocols individually, but keep in mind that the SFTP protocol will not work without the SSH protocol enabled.

The communication protocols use the following login information:

- User Name: XC303_User
- Password: XC303_User

The password can be changed from the XSoft-Codesys using PLC shell command “setuserpwd.”

Using SSH

The SSH protocol can be used for remote control of the PLC. It is a text-based protocol that can be used both to display and modify files and to start programs.

Using SFTP

The SFTP protocol offers a mostly graphical interface for displaying and modifying files. (e.g., /home/InitPLC file)

To connect to and operate the client interface, refer to the corresponding documents for the client in question.

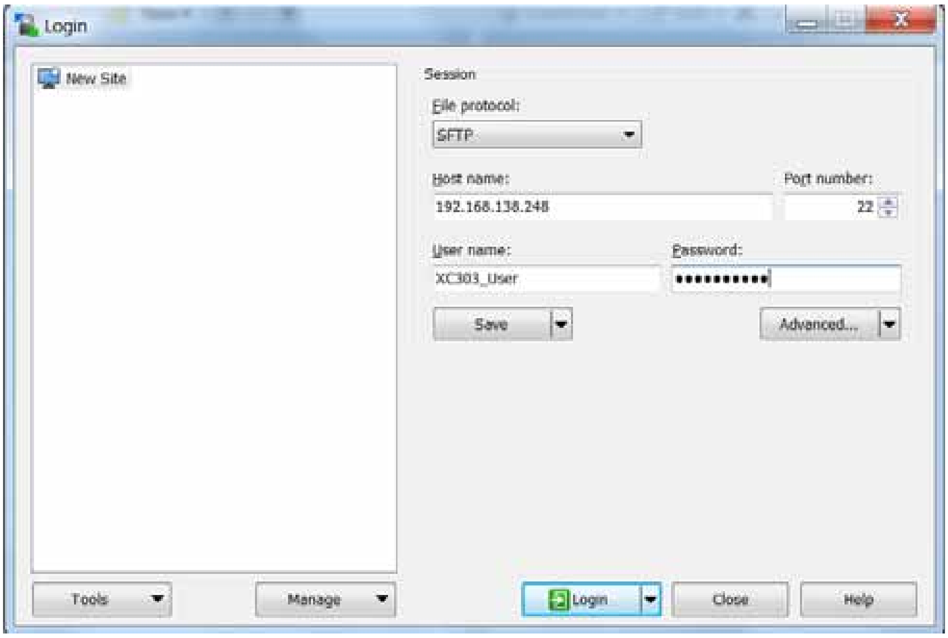
12.1 SFTP Server

The SFTP server is started on the device by default. Once started, file InitPlc is read and the system parameters are set.
By default the setting in InitPlc is ENABLE_SFTP = 1.

12.1.1 Accessing the Device using the SFTP Server

To access the device via WinSCP using the SFTP server, proceed as follows:

- ▶ Ensure that there is communication with the device.
- ▶ Open WinSCP and enter the IP address for the Ethernet connection.
- ▶ In the “Log in as” window, enter the standard user name “XC303_User” and the standard password “SC303_User”.



File protocol: SFTP

Host name: 192.168.138.248

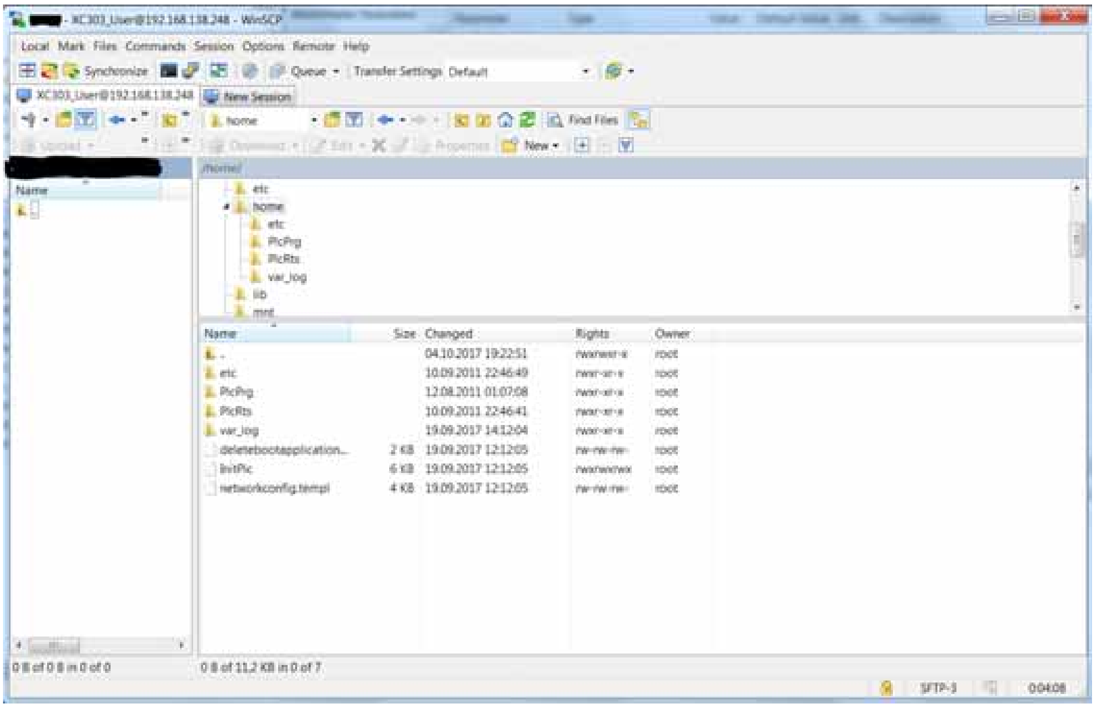
Port number: 22

User name: XC303_User

Password: •••••••••••

Figure 82: Logging in as a standard user

You then have access to the directory /home on the device.



```
[redacted] - XC303_User@192.168.138.248 - WinSCP
[redacted]
/home/
  etc
  home
    etc
    PlcPrg
    PlcRts
    var_log
  lib
  mnt

Name                      Size  Changed              Rights      Owner
..                              04.10.2017 19:22:51  rwxrwxr-x   root
etc                             10.09.2011 22:46:49  rwxr-xr-x   root
PlcPrg                          12.08.2011 01:07:08  rwxr-xr-x   root
PlcRts                          10.09.2011 22:46:41  rwxr-xr-x   root
var_log                         19.09.2017 14:12:04  rwxr-xr-x   root
deletebootapplication...  2 KB  19.09.2017 12:12:05  rw-rw-rw-   root
InitPlc                   6 KB  19.09.2017 12:12:05  rwxrwxrwx   root
networkconfig.templ       4 KB  19.09.2017 12:12:05  rw-rw-rw-   root

0 B of 0 B in 0 of 0
0 B of 11,2 KB in 0 of 7
SFTP-3   0:04:08
```

Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
XC303_User@192.168.138.248
New Session
home
Find Files
Upload
Download
Edit
Properties
New
Name

Figure 83: /home directory on the XC300 Modular control

The /home directory is the only directory on the device that standard users are allowed to write to.

Directory	Description
home/etc	Write-protected flash area; system-specific files such as InitPlc
home/PlcPrg	Contains boot project
home/PlcRts	Runtime system settings
home/disk_usb	Optional: only available if a USB stick is inserted

12.1.2 Changing the STFP Server Password

The password for the SFTP server can be changed using the browser command SSH -> setuserpwd.

See → section 16, “PLC Shell Commands”, page 133.

12.1.3 Switching Off the SFTP Server

To switch off the SFTP server, proceed as follows:

- ▶ Ensure that there is communication with the device.
- ▶ Log in to the device as an administrator.
- ▶ Edit file InitPlc on the device.
- ▶ Set the system parameter to “ENABLE_SFTP=0.”
- ▶ Save the file.
- ▶ Restart the device.

The SFTP server is now switched off.

13 Web-Server

The XC300 controller has a web server for sharing the visualization created by XSOFT-CODESYS-3.
The handling of the web visualization is explained in the XSOFT-CODESYS-3 help.

Keywords: CODESYS Visualization / Object “Web Visu”

13.1 Web visualization

A description of the web visualization interface can be found in section 7.4, “Web Visualization,” of the manual for programming PLCs with CODESYS.

The visualization can be accessed via any of the Ethernet interfaces in your XC300 Modular control.

If the device is being operated via Ethernet interface ETH2, for example, with the default IP address, the XC300-specific access to the web visualization would be:

http:\\192.168.138.248:8080/webvisu.htm
https:\\192.168.138.248/webvisu.htm

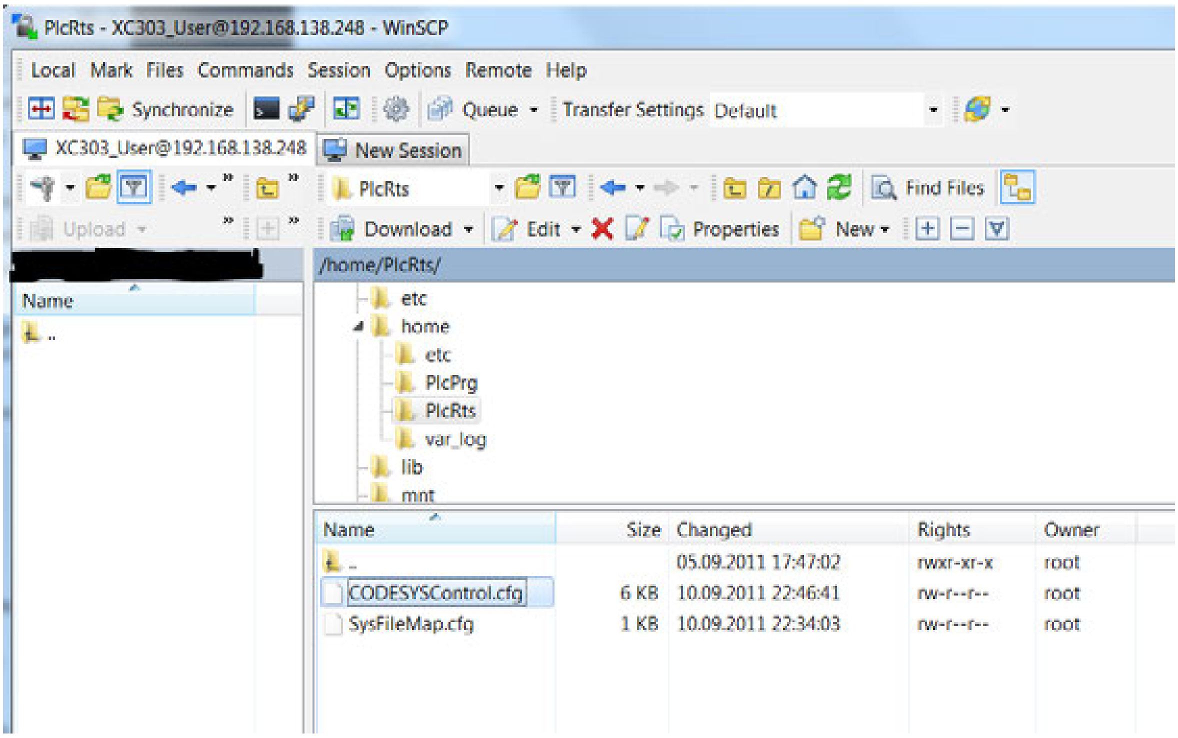
If you have changed the IP address, replace the IP address in the “http:\\...” with the address you have selected.

ATTENTION
The number of clients with simultaneous access to the XC300 is limited to 10.

13.2 Web Server for XSOFT-CODESYS-3

Access to the web server for XSOFT-CODESYS-3 projects can be established via either http or https protocol.
The protocol that is available is defined in file CODESYSControl.cfg by 3S.

This file can be found on the device in directory \home\PlcRts.


PlcRts - XC303_User@192.168.138.248 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
XC303_User@192.168.138.248
New Session
PlcRts
Find Files
Upload
Download
Edit
Properties
New
[redacted]
Name
/home/PlcRts/
etc
home
etc
PlcPrg
PlcRts
var_log
lib
mnt

Name	Size	Changed	Rights	Owner
..		05.09.2011 17:47:02	rwxr-xr-x	root
CODESYSControl.cfg	6 KB	10.09.2011 22:46:41	rw-r--r--	root
SysFileMap.cfg	1 KB	10.09.2011 22:34:03	rw-r--r--	root

Figure 84: Directory “PlcRts” on XC300

To change the protocol for the web server, proceed as follows:

Open the file and search for the entry for [CmpWebServer].

Depending on the parameter connection type, the protocols are permitted as follows:

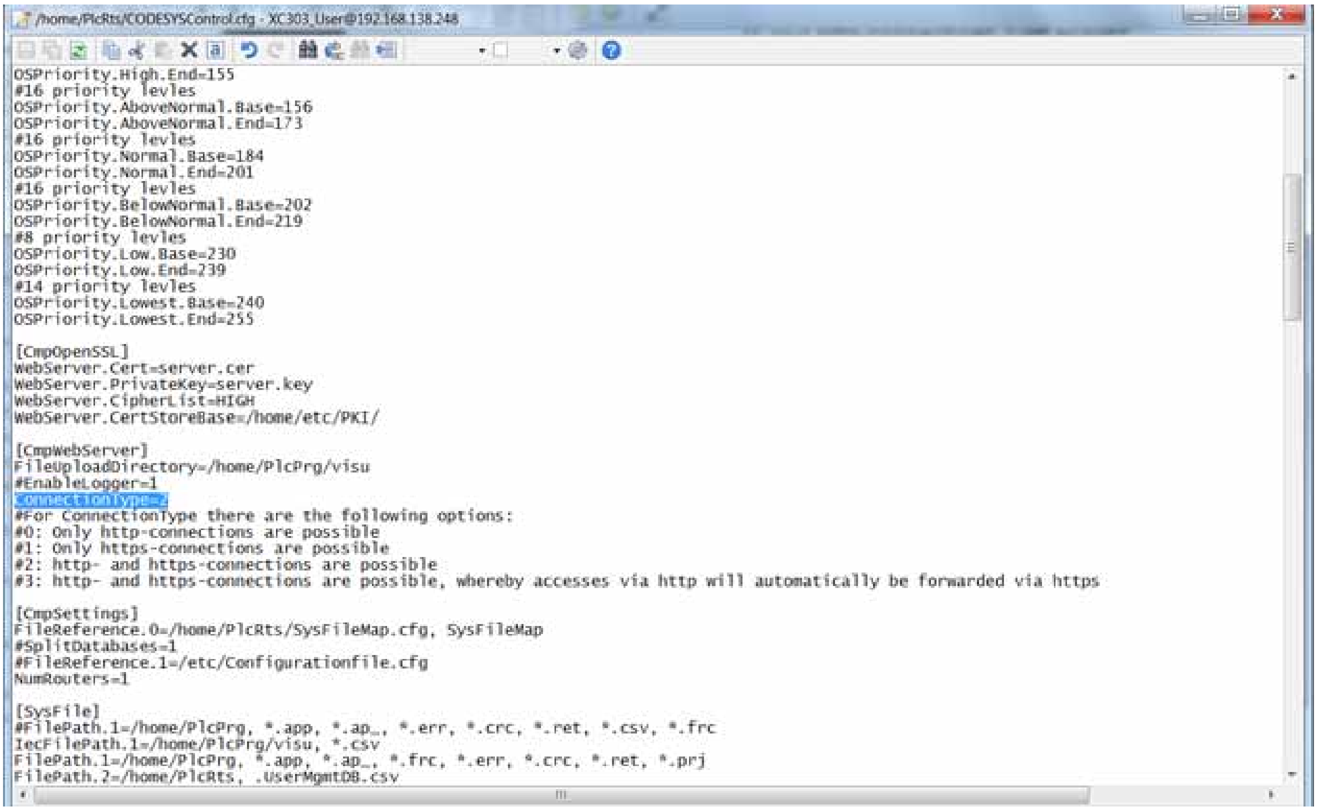
ConnectionType =

0: Only http connections allowed

1: Only https connections allowed

2: http and https connections allowed

3: http and https connections allowed, but access via http is automatically redirected to https.



/home/PlcRts/CODESYSControl.cfg - XC303_User@192.168.138.248

```
OSPriority.High.End=155
#16 priority levles
OSPriority.AboveNormal.Base=156
OSPriority.AboveNormal.End=173
#16 priority levles
OSPriority.Normal.Base=184
OSPriority.Normal.End=201
#16 priority levles
OSPriority.BelowNormal.Base=202
OSPriority.BelowNormal.End=219
#8 priority levles
OSPriority.Low.Base=230
OSPriority.Low.End=239
#14 priority levles
OSPriority.Lowest.Base=240
OSPriority.Lowest.End=255

[CmpOpenSSL]
WebServer.Cert=server.cer
WebServer.PrivateKey=server.key
WebServer.CipherList=HIGH
WebServer.CertStoreBase=/home/etc/PKI/

[CmpWebServer]
FileUploadDirectory=/home/PlcPrg/visu
#EnableLogger=1
ConnectionType=2
#For ConnectionType there are the following options:
#0: Only http-connections are possible
#1: Only https-connections are possible
#2: http- and https-connections are possible
#3: http- and https-connections are possible, whereby accesses via http will automatically be forwarded via https

[CmpSettings]
FileReference.0=/home/PlcRts/SysFileMap.cfg, SysFileMap
#SplitDatabases=1
#FileReference.1=/etc/Configurationfile.cfg
NumRouters=1

[SysFile]
#FilePath.1=/home/PlcPrg, *.app, *.ap_, *.err, *.crc, *.ret, *.csv, *.frc
IecFilePath.1=/home/PlcPrg/visu, *.csv
FilePath.1=/home/PlcPrg, *.app, *.ap_, *.frc, *.err, *.crc, *.ret, *.prj
FilePath.2=/home/PlcRts, .UserMgmtDB.csv
```

Figure 85: ConnectionType

14 OPC UA

OPC Unified Architecture (OPC UA) is a standardized communication protocol for data exchange between machines and facilitates universal use of all devices.

OPC UA is supported by CODESYS version V3.5.9 or higher.

The OPC UA server is integrated into the controller. When the controller starts, the OPC UA server automatically starts, too.

The scope of function of OPC UA is the same as OPC DA. The URL of the OPC UA server must be entered in the OPC UA client:

“opc.tcp://<IP-Address>:4840”

As with OPC DA, the symbol list in X-SOFT-CODESYS 3 must be set up.

Additional information must be loaded on the controller in order to operate the OPC UA server. The symbol configuration is generated during download for this purpose.

- Basic types of derived function blocks
- Contents of attributes allocated via compiler pragmas
- Variable type (INPUT / OUTPUT / IN-OUT)

To generate the symbol configuration, proceed as follows:

▶ Select the application and open the context menu.
▶ Select “Add Object” and “Symbol Configuration.”
▶ Specify a file name.
▶ Activate the checkbox option “Support OPC UA Functions.”
▶ Confirm with “OK.”

The symbol configuration editor opens.
Here you can define all variables that should be transferred.

The symbol configuration will be updated with the next download or online change.

15 Libraries, function blocks, and functions

The libraries contain IEC function blocks and functions that you can use, for example, for the following tasks:

- Data exchange through the CAN bus
- Controlling the real-time clock
- Determining bus load of the CAN bus
- Triggering interrupts
- Sending/receiving data through the interfaces

> CoDeSys supports the user with different functions and function blocks that are managed in libraries.
> Available libraries provide support for:
> - Fieldbus communication (e.g., CAN)
> - Real-time clock functions
> - Communication via serial interfaces
> - File handling functions

The following libraries are available:

- EA_Network library for Ethernet
- IODrvEtherCAT for EtherCAT
- EA_CANopen library for CANopen
- IODrvModbus for ModbusRTU
- IODrvModbusTCP for ModbusTCP
- IODrvEthernet for ModbusTXC300 Modular controlCP
- EA_CAN types (when using EATON CAN devices)
- EA_CAN library with function blocks for CAN
- EA_XC303Events library for interrupt processing

> Examples of libraries available to the user:
> - EA_CANopen
> - IODrvModbus
> - SysCom
> - SysFileAsync

15.1 Using libraries

Libraries with frequently used functions are automatically added to a project when a standard project is created. These include Standard.lib, for example, with well-known elements such as the counter, timer, trigger elements, etc. Other libraries that are automatically added are the IoStandard.lib and 3SLicense.lib.

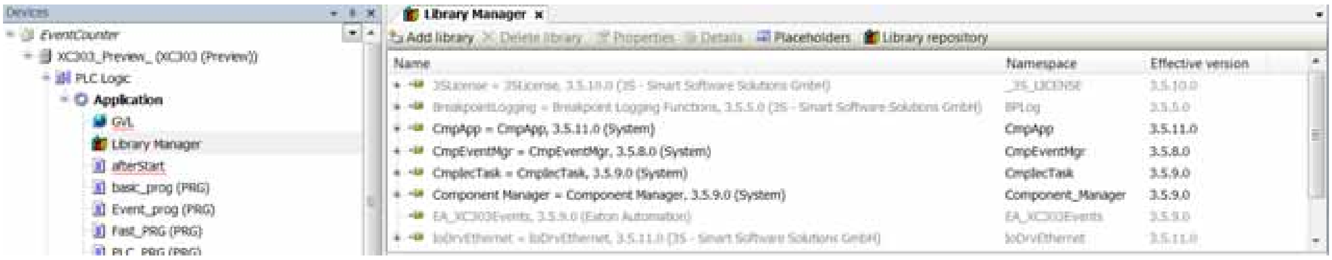
When a project is opened the following libraries are copied to the library manager.

- 3SLicense (system library)
- Standard.lib
- IoStandard
- BreakPointLogging

If more libraries are required for the application, you will need to install these in the library manager.
Once the project is saved, these libraries are assigned to the project and will be available the next time the project is opened.

15.2 Installing additional system libraries

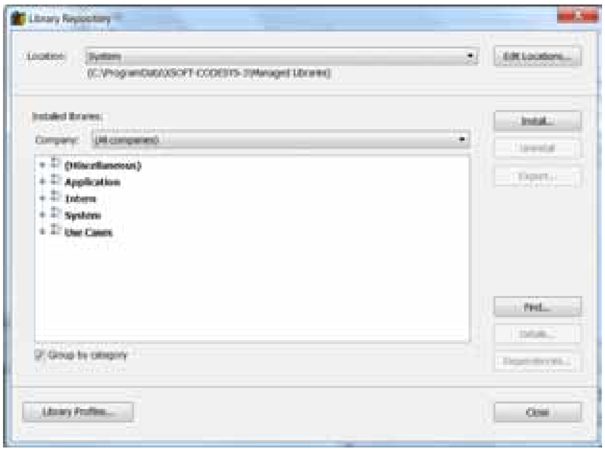
More libraries can be installed as needed.
The following steps are necessary for this:



Name	Namespace	Effective version
3SLicense = 3SLicense, 3.5.10.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.10.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
CmpApp = CmpApp, 3.5.11.0 (System)	CmpApp	3.5.11.0
CmpEventMgr = CmpEventMgr, 3.5.8.0 (System)	CmpEventMgr	3.5.8.0
CmpIecTask = CmpIecTask, 3.5.9.0 (System)	CmpIecTask	3.5.9.0
Component Manager = Component Manager, 3.5.9.0 (System)	Component_Manager	3.5.9.0
EA_XC303Events, 3.5.9.0 (Eaton Automation)	EA_XC303Events	3.5.9.0
IoDrvEthernet = IoDrvEthernet, 3.5.11.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	3.5.11.0

Figure 86: Manually installing libraries

- ▶ Ensure that the library to be installed is saved in a folder on your PC.
- ▶ Double-click “Library Manager” in the configuration tree diagram.
- ▶ Click on the “Add Library” option.



Location: System
(C:\ProgramData\XSOFT-CODESYS-3\Managed Libraries)

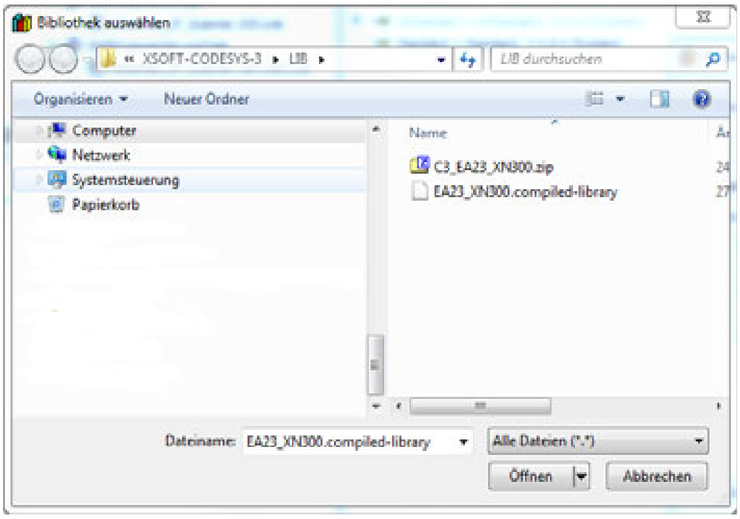
Installed libraries:
Company: (All companies)

- (Miscellaneous)
- Application
- Intern
- System
- Use Cases

☑ Group by category

Figure 87: Adding a library to the library manager

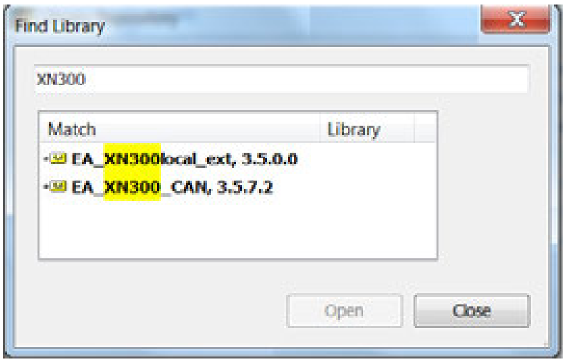
- ▶ Click the “Install” button.
- ▶ In the Explorer, select the library in the folder where it is saved.



C3_EA23_XN300.zip
EA23_XN300.compiled-library

Dateiname: EA23_XN300.compiled-library

- ▶ Confirm by clicking on “Open.”
  The new window will show the libraries available, depending on the target system.
- ▶ Select “Add Library.”

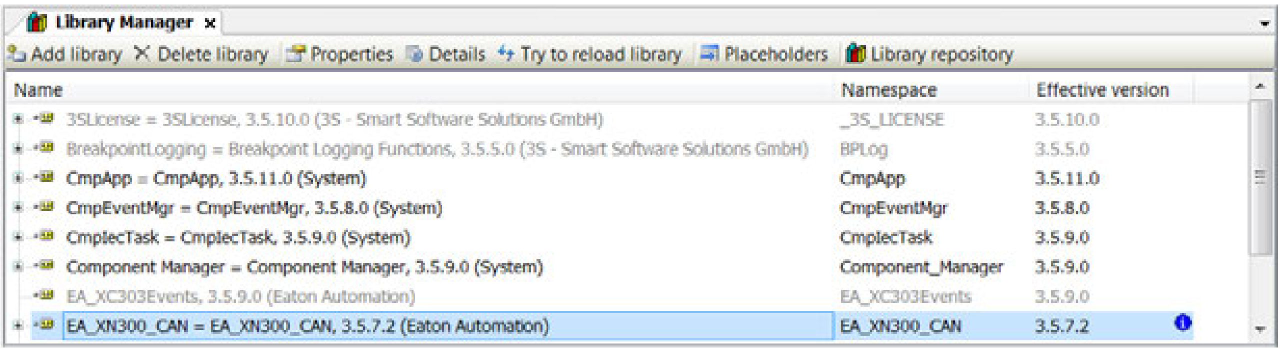


XN300

EA_XN300local_ext, 3.5.0.0
EA_XN300_CAN, 3.5.7.2

Figure 88: Searching for a library in the library manager

- ▶ Enter a search string, such as “XN300,” so the installed library can be found quickly.
- ▶ Select the library to be installed and click “OK.”
  The library now appears in the Library Manager.



Name	Namespace	Effective version
3SLicense = 3SLicense, 3.5.10.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.10.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
CmpApp = CmpApp, 3.5.11.0 (System)	CmpApp	3.5.11.0
CmpEventMgr = CmpEventMgr, 3.5.8.0 (System)	CmpEventMgr	3.5.8.0
CmpIecTask = CmpIecTask, 3.5.9.0 (System)	CmpIecTask	3.5.9.0
Component Manager = Component Manager, 3.5.9.0 (System)	Component_Manager	3.5.9.0
EA_XC303Events, 3.5.9.0 (Eaton Automation)	EA_XC303Events	3.5.9.0
EA_XN300_CAN = EA_XN300_CAN, 3.5.7.2 (Eaton Automation)	EA_XN300_CAN	3.5.7.2

Figure 89: Searching for a library in the library manager

→ The user has the option of creating their own libraries, in order to save recurring tasks as functions or blocks in custom libraries.

15.3 “EA_Network.lib” library

The EA_Network.lib library provides functions for all Eaton controllers with Ethernet interfaces, in order to configure these Ethernet interfaces.

→ Manufacturer’s documentation on XSOFT-CODESYS-3,
Online help for EA-NEtwork.lib library

16 PLC Shell Commands

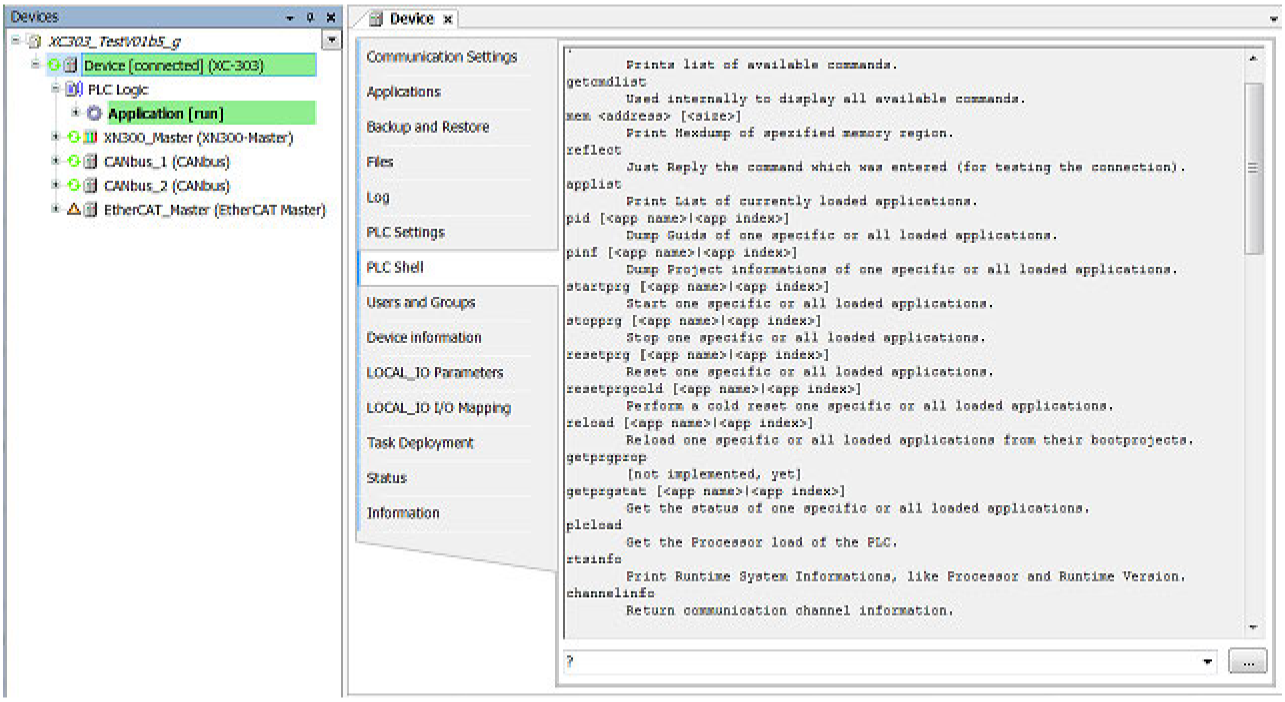
The PLC shell is a text-based control monitor. PLC shell commands for scanning particular information from the PLC are entered in an entry line and sent to the PLC as a string. The response string is displayed in a results window in the PLC shell. This functionality can be used for diagnostic and debugging purposes.

16.1 Accessing PLC Shell Commands

To enter a PLC shell command, proceed as follows:

- ▶ Establish a connection to the device.
- ▶ In the CoDeSys software, double-click on the device “Device (XC303)” in the configuration tree diagram.
- ▶ Select the “PLC Shell” tab.
- ▶ In the entry line below, enter the PLC shell command or click on the “...” button and select the desired PLC shell command by double-clicking it.
- ▶ Add other information to the command, if necessary, such as the baud rate for setcomconfig.
- ▶ Press the Return button.

The result will be displayed.



```
?
        Prints list of available commands.
getcmdlist
        Used internally to display all available commands.
mem <address> [<size>]
        Print Hexdump of spezified memory region.
reflect
        Just Reply the command which was entered (for testing the connection).
applist
        Print List of currently loaded applications.
pid [<app name>|<app index>]
        Dump Guids of one specific or all loaded applications.
pinf [<app name>|<app index>]
        Dump Project informations of one specific or all loaded applications.
startprg [<app name>|<app index>]
        Start one specific or all loaded applications.
stopprg [<app name>|<app index>]
        Stop one specific or all loaded applications.
resetprg [<app name>|<app index>]
        Reset one specific or all loaded applications.
resetprgcold [<app name>|<app index>]
        Perform a cold reset one specific or all loaded applications.
reload [<app name>|<app index>]
        Reload one specific or all loaded applications from their bootprojects.
getprgprop
        [not implemented, yet]
getprgstat [<app name>|<app index>]
        Get the status of one specific or all loaded applications.
plcload
        Get the Processor load of the PLC.
rtsinfo
        Print Runtime System Informations, like Processor and Runtime Version.
channelinfo
        Return communication channel information.
```

```
?
```

Figure 90: PLC shell tab with result of the browser command “?”

16.2 Overview of PLC Shell Commands

The following PLC shell commands are available.

Table 12: PLC shell commands in alphabetical order

Command	Description
?	Get a list of implemented commands
applist	Print list of currently loaded applications.
caninfo	Display CAN controller information
canload <arg>	Display of the loading of the CAN fieldbus <arg> <none> show CAN-Bus load. <reset> reset CAN-Bus load statistics.
cert-getapplist	Display all registered applied certificates (Component ID and application)
cert-genselfsigned [<number retrieved by "cert-getapplist"> <expdays=>]	Generate all self-signed certificates
cert-gendhparams [len in bits]	Generate parameters needed for Diffie-Hellman key exchange. Attention: may run for several minutes!
channelinfo	Return communication channel information.
copyprojtosd	Copy the (boot) project onto a Multi Media Card (incl. directory structure/project directory)
copyprojtousb[1)]	Copy the (boot) project onto the USB drive (incl. directory structure/project directory)
enabledhcp <arg>	Enable DHCP function for controller; <arg>= address of adapter number
getbattery	Display battery status
getcmdlist	Used internally to display all available commands.
gethwinfo	Show hardware information
getipconfig	Display Ethernet address
getipdns	Display current DNS address
getipgateway	Display Gateway address
getprgprop	[not implemented yet]
getprgstat [<app name>|<app index>]	Get the status of one specific or all loaded applications.
getrtc	Display data and time [YY:MM:DD] [HH:MM:SS]
gettimezone	Show actual time zone information (offset to UTC)
getversion	Show device versions list
isdhcpenabled <arg>	Query whether DHCP is enabled; <arg>= address of adapter number
mem <address> [<size]>	Show free memory on disk_sys. Print hex dump of specified memory region
pid [<app name>|<app index>]	Output project ID
pinf [<app name>|<app index>]	Output project information
plcload[1)]	Show system performance: CPU utilization
reboot	Accept changes (registry save) and restart PLC
reflect[1)]	Mirror current command line for test purposes.
reload [<app name>|<app index>]	Reload boot application
remprojfromsd	Delete backup project from MicroSD card
remprojfromusb	Delete backup project from USB stick
resetprg [<app name>|<app index>]	Reset user program

Command	Description
resetprgcold [<app name>|<app index>]	User program cold reset
rtsinfo	runtime system information (version, IO drivers)
setipdns	Set DNS address [setipdns adr1.adr2.adr3.adr4]
setipaddr <arg>)	Set IP address (<arg> = AdapterNo Adr1.Adr2.Adr3.Adr4)
setipbroadcast <arg>	Set IP of broadcast address (<arg> = AdapterNo Adr1.Adr2.Adr3.Adr4)
setipnetmask <arg>	Set IP subnet mask (<arg> = AdapterNo Adr1.Adr2.Adr3.Adr4)
setipgateway <arg>	Set IP default gateway (<arg> = AdapterNo Adr1.Adr2.Adr3.Adr4)
setrtc1)	Set date and time [YY:MM:DD] [HH:MM:SS]; example: setrtc 03:07:24 10:46:33
settimezone <arg>	Set new time zone information (<arg> = offset to UTC)
startprg	Start user program
stopprg	Stop user program
xngetversion	Show version information, show current XN driver version
xnmodules	Show currently configured and connected SDIAS modules
xnvariables	Show XN module variables
xnconfig	Show XN module configuration variables
xnstatistics	Show statistical XN data

1) You can access help in the CoDeSys software with advanced information for this PLC shell command.
Enter a question mark followed by a space before the command in the command line of the PLC browser, e.g. ? plcload

16.3 Shell Commands for CPU Utilization

plcload

The PLC shell command plcload provides information on the current system utilization of the CPU.

This information can also be accessed via the “SysGetPlcLoad” function in the 3S standard library SysPlcCtrl23.

A utilization of more than 95 percent can cause a failure of the serial and Ethernet communication and/or an impairment of the real-time response.

canload

The PLC shell command canload shows the utilization of the CAN bus. caninfo shows the baud rates and statuses of the CAN buses.

The information on the CAN bus can also be accessed using the CAN_GetBusLoad and CAN_GetBusState functions in the EA_CAN library.

17 Malfunctions

This section provides troubleshooting information for your XC300 Modular control in case it does not behave as expected.

Malfunction	Cause	Remedy
XC300 will not boot up	There is no 24 V DC supply voltage	Check the input wiring.XC300 Modular control

18 Maintenance

18.1 Cleaning and maintenance

The devices are maintenance-free.
The following tasks may be necessary:

▶ Clean the device if it is dirty.

> ⚠ **CAUTION**
> **POINTED, SHARP OBJECTS, OR CORROSIVE LIQUIDS**
>
> Do not use pointed or sharp objects (such as knives) to clean the device.
> Do not use aggressive or abrasive cleaning agents or solvents.
>
> Make sure that no liquids get into the unit (short-circuit hazard) and that the unit is not damaged in any way.

▶ Clean the device carefully with a clean, soft, moist cloth.
If dirt is stubborn, first spray a small amount of cleaning agent on the moist cloth.

18.2 Repairs

For repairs, please contact your distributor or Eaton’s Technical Support.

> ⚠ **CAUTION**
> **DESTRUCTION OF THE DEVICE**
>
> The device should only be opened by the manufacturer or by an authorized repair center. Operate the device only with the housing fully closed and sealed.

For repairs, please contact your distributor or Eaton’s Technical Support.

Use the original packaging to ship the device.

18.3 Storage and transport

> ⚠ **CAUTION**
> **UV LIGHT**
>
> Plastics will become brittle when exposed to UV light.
> This form of artificial aging will reduce the unit’s lifespan.
> Protect the device from direct sunlight or other sources of UV radiation.

CAUTION
SHORT-CIRCUIT HAZARD

If the device is or has been exposed to environmental fluctuations (ambient temperature, air humidity), condensation may form on or inside it. As long as this condensation is present, there will be a short-circuit hazard.
Do not switch on the device when it has condensation in or on it.
If the device has condensation in or on it, or if the device has been exposed to environmental fluctuations, let the device settle to the existing ambient temperature before switching it on.
Do not expose the device to direct thermal radiation from heating appliances.

The specified ambient conditions must be met when transporting and storing the equipment.

The max. ambient temperature for storage/transport must not exceed the specified value.

→ Before commissioning
If storing/transporting the device in cold weather conditions or in such a way that it will be exposed to extreme differences in temperature, make sure that no condensation forms on or inside the device.
If there is condensation in or on the device, do not switch on the device until it is completely dry.

▶ Use the original packaging to ship the device.

The device is sturdily built, but the components inside it are sensitive to excessively strong vibrations and/or mechanical shock.

Accordingly, make sure to protect the device from mechanical loads that exceed the scope of the device’s intended use.

The device should only be transported in its original packaging after being packed properly.

18.4 Disposal

CAUTION

Installation requires qualified electrician

INSTRUCTION

Dispose of recyclables as required by your local recycling regulations.

DANGER

RISK OF EXPLOSION OF LITHIUM BATTERY

The lithium battery inside the device may explode if handled incorrectly.

▶ Dispose of the device properly.

Devices no longer being used must be professionally disposed of as per local regulations or returned to the manufacturer or the relevant sales office.

19 Appendix

19.1 Dimensions and Weights

		XC-303-…
Dimensions (H × D × W)	mm	80.3 x 72 x 50
	inch	3.16 x 2.83 x 1.97
Space units (SU) width		
Weight	kg	0.3
	lb	0.66
Mounting		DIN-rail IEC EN 60715, 35 mm
Mounting position		vertical, upright

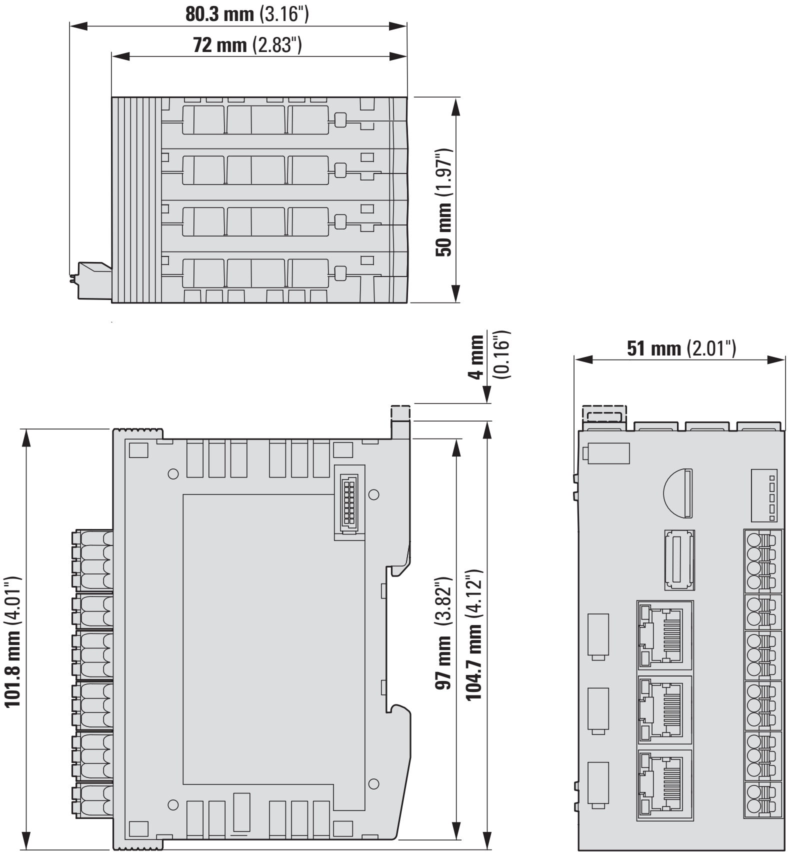


Figure 91: Dimensions of the XC303-…

19.2 Approvals and National Approvals for Devices XC300

XC300 and devices of the XN300 system are approved for use in several countries and regions.

Product standards	IEC/EN 61131-2 CE mark
NA Certification	cULus (in progress)
Protection rating	IEC: IP20

19.3 Technical Data

19.3.1 General ambient conditions

		Unit	XC-300
Ambient climatic conditions			
Standards			IEC/EN/DIN EN 61131-2
Ambient temperature		°C	-25 - +55
Storage temperature		°C	-40 - +80
Mounting position			horizontal, based on DIN rail
Relative humidity, no condensation (IEC/EN 60068-2-30)		%	5 - 95
Air pressure (in operation)		m	max. 2000
Ambient mechanical conditions			
Protection rating			IEC: IP20
Vibration resistance			
Constant amplitude (5 - 8.4 Hz)	mm		±3.5 mm
Constant acceleration (8.4 - 150 Hz)	g		±1.0
Shock resistance (half sinewave, 15 g/11 ms)		Shocks	9
Dropped fall height (EN 60068-2-31)		mm	50
Free fall, packaged (EN 60068-2-32)		m	0.3
Overvoltage category			II
Pollution level			2
Insulation resistance (UL 61010-2)		V_{eff}	500
Terminals			Plug-in terminal block
Terminal cross-sections			
Spring-type terminals			
Solid		mm^2	0.2 - 1.5
Flexible with uninsulated ferrule		mm^2	0.25 - 1.5
Flexible with insulated ferrule		mm^2	0.25 - 0.75
Collar	d	mm	≦ 2.8
AWG			24–16
Stripping length		mm	10
Electromagnetic compatibility (EMC)			→ page 147
PLC			
PLC microprocessor XC300			Freescale CORTEX A7 Dual core with 960 MHz
Memory			
Program memory		MB	QSPI-FLASH: 128 MB
Working memory		MB	DDR-RAM: 512 MB
Data memory		kByte	NV-RAM: 128
Retentive data		kByte	96

	Unit	XC-300
Operating System		Embedded Linux Version 3.12 or higher
Watchdog		yes
RTC (Real-Time Clock)		yes
External storage media		
Micro Secure Digital Card		yes, optional
USB stick		yes, optional
PLC status indicator		LED, one color, green LED, one color, red
System bus		
Maximum number of bus subscribers		32
Update rate I/O image	ms	< 1
Ethernet interfaces		
Connection technology		RJ45
Data transfer rate		
ETH0, ETH1	MBit/s	10/100
ETH2	MBit/s	10/100/1.000
Galvanic isolation ETH0, ETH1, ETH2 from		
each other		yes
System bus		yes
U_L supply voltage I/Q1 – I/Q4		yes
RS485		yes
CAN1		yes
CAN2		yes
Local inputs		yes
Local outputs		yes
Connection technology		RJ45
USB interface		
Type of interface		USB 2.0 host type A, downstream plug
Power supply for connected devices:		
Rated voltage	V DC	5 (-5 % +10 %)
max. current	mA	100
Galvanic isolation from		
System bus		no
POW		no
U_L supply voltage I/Q1…I/Q4		yes
RS485		yes
CAN1		yes
CAN2		no
Local inputs		yes
Local outputs		yes

Electromagnetic compatibility			
Interference immunity			
ESD (IEC/EN 61000-4-2)	Contact discharge		4 kV
	Air discharge		8 kV
Electromagnetic fields (RFI): criterion A			
0.8 - 1.0 GHz		V/m	10
1.4 - 2.0 GHz		V/m	3
2.0 - 2.7 GHz		V/m	1
Burst (EN 61131-2, Level 3) criterion B	POW	kV	2
	ETH 0…2	kV	1
	Local inputs/outputs	kV	1
Surge, (EN 61131-2, Level 1), criterion B	POW, symmetric, asymmetric	kV	0.5
	ETH0…2/CAN/RS485, symmetric	kV	1
	Local inputs/outputs, symmetric	kV	1
Cable-controlled interference, induced by high-frequency fields EN61131-2		V	10

19.3.2 Supply voltage to device and system bus

POW supply voltage to CPU		Unit	
Mains failure bridging			
Duration of dip		ms	20
Rated operating voltage			
Nominal value	U_e	V DC	24
Permissible range		V DC	18 - 30
Residual ripple		%	≦ 5
Protection against polarity reversal			yes
Input current I_e		A	max. 2.8A
Maximum current consumption at 24 V			8 W
Galvanic isolation of POW from			
system bus)			no
U_L supply voltage I/Q1 – I/Q4			yes
RS485			yes
CAN1			yes
CAN2			no
Local inputs			yes
Local outputs			yes
ETH0, ETH1, ETH2			yes
POW status indicator			LED, one color, green
U_{BP_24} output voltage for the XN300 slice modules			
Output rated voltage		V DC	24
Output current		A	1.6
Short-circuit rating			yes
Galvanic isolation from POW			no
U_{BP_5} output voltage for the XN300 slice modules			
Output rated voltage		V DC	5
Output current		A	1.6
Short-circuit rating			yes
Galvanic isolation from POW			no

19.3.3 Power Supply to Local Inputs/Outputs

Power supply to local inputs/outputs		Unit	
Rated operational voltage U_L			
Nominal value	U_e	V DC	24
Permissible range		V DC	18 - 30
Residual ripple		%	≦ 5
Protection against polarity reversal			yes
Input current I_L		A	max. 2
Galvanic isolation U_L from			
System bus			yes
POW			yes
RS485			yes
CAN1			yes
CAN2			yes
Local inputs			no
Local outputs			no
ETH0, ETH1, ETH2			yes
Status Indication			LED, one color, green

19.3.4 Local inputs

Each of the transistor inputs I1 – I4 can alternatively be operated as an output Q1 – Q4.

Digital inputs (EN 61131-2, type 1)	**Unit**	
Inputs	Number	4
Input rated voltage	V DC	24, observe polarity
Voltage range	V DC	24 - 30
Input current per channel at rated operating voltage of 24 V	mA	typ. 2
Switching levels as per EN 61131-2		
Limit values type “1”	V DC	low < 5, high > 15
Debounce setting		
Off → On	ms	< 0.2
On → Off	ms	< 0.2
Can be used as interrupt inputs		I1 – I4
Galvanic isolation from		
each other		no
Local outputs		no
System bus		yes
POW		yes
RS485		yes
CAN1		yes
CAN2		yes
ETH0, ETH1, ETH2		yes
Status Indication		LED, one color, green
Cable length (unscreened)	m	< 30

19.3.5 Local outputs

Each of the transistor outputs Q1 – Q4 can alternatively be operated as inputs I1 – I4.

The outputs are switched off when the controller is RESET.

Digital outputs			
Channels		Number	4
Short-circuit proof			yes
Insulation voltage		VAC	500
Heat dissipation per channel			
Q1 – Q4		W	< 0.1
Load circuits			
Q1 to Q4		A	0.5
Output delay			
Off → On			typ 0,1 ms
On → Off			typ 0,1 ms
Duty factor DF		%	100
Simultaneity factor g			1
Galvanic isolation from			
each other			no
from the inputs			no
System bus			yes
POW			yes
RS485			yes
CAN1			yes
CAN2			yes
ETH0…ETH2			yes
Status Indication			LED, one color, green
Cable length (unscreened)		m	< 30

19.3.6 CAN1 and CAN2 Interface

CAN(open)/easyNet		**Unit**	
Data transfer rate CAN1, CAN2		Kbits/s	50/100/125/250/500/1000
Device profile			As per DS 301 V4 ???
29 Bit Identifier			yes
PDO type			asyn., cyc., acyc.
Connection			Plug connector, 3-pin, push-in
Bus termination resistors for the first and last CAN subscriber			120 Ω switchable for CAN1 with DIP switch TC1 CAN2 with DIP switch TC2
CANopen/easyNet subscribers		Number	max. 126/8
Galvanic isolation of CAN1 from			
U_{POW} supply voltage			yes
System bus			yes
U_L supply voltage I/Q1 – I/Q4			yes
RS485			no
CAN2			yes
Local inputs			yes
Local outputs			yes
ETH0, ETH1, ETH2			yes
Galvanic isolation of CAN2 from			
U_{POW} supply voltage			no
System bus			no
U_L supply voltage I/Q1 – I/Q4			yes
RS485			yes
CAN1			yes
Local inputs			yes
Local outputs			yes
ETH0…ETH2			yes
Status Indication			LED, multicolor, green, red
Connection cable			A shielded cable must be used for the interface. The shield must be placed externally on two sides.

19.3.7 RS485 interface

RS485 serial interface (without handshaking)		
Connection technology		Plug connector, 3-pin, push-in
Data transfer rate as		
Modbus RTU	Bit/s	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Serial interface	Bit/s	4800, 9600, 19200, 38400, 57600, 115200
Bus termination resistors for the first and last subscriber		220 Ω switchable; For connection both DIP switches TRSa and TRSb must be set to ON position
Galvanic isolation of RS485 from		
U_{POW} supply voltage		yes
System bus		yes
U_L supply voltage I/Q1 – I/Q4		yes
CAN1		no
CAN2		yes
Local inputs		yes
Local outputs		yes
ETH0, ETH1, ETH2		yes
Status Indication		LED, multicolor, green, red
Connection cable		A shielded cable must be used for the interface. The shield must be placed externally on two sides.

19.4 Glossary

AWL	Application list
Boot project	Project loaded from the non-volatile memory of the CPU
Debounce setting	The debounce setting defines the delay of a pulse edge from the input terminal I1 – I4 to the GPIO of the CPU.
Forcing	Presetting of changed operators/variables to a fixed value
/home	Top file directory in the controller XC300 XC300 Modular control which a user with status "XC-303_user" can access
MicroSD	Micro Secure Digital Card
OS	Operating System
PLC	Memory Programmable Logic Controller
POU	Programming Organizational Unit A POU type object is a programming organizational unit in a CoDeSys project. Source code for your PLC program is written in POUs.
PRG	Program
RC	Remote Control
RTS	Runtime System
Task	Process, task carried out by the operating system

Index

At Eaton, we're energized by the challenge of powering a world that demands more. With over 100 years experience in electrical power management, we have the expertise to see beyond today.
From groundbreaking products to turnkey design and engineering services, critical industries around the globe count on Eaton.

We power businesses with reliable, efficient and safe electrical power management solutions. Combined with our personal service, support and bold thinking, we are answering tomorrow's needs today.
Follow the charge with Eaton.

Visit: www.eaton.com/electrical

Eaton addresses worldwide:
www.eaton.com

E-Mail: info-bonn@eaton.com
Internet: www.eaton.eu/xc300

Eaton Industries GmbH
Hein-Moeller-Str. 7 – 11
D-53115 Bonn

MN050005EN ETN/ICPD MOC 02/18

EATON
Powering Business Worldwide