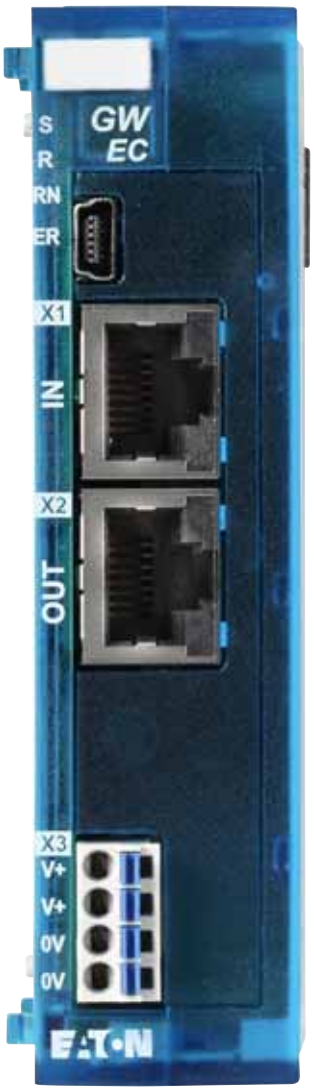
EtherCAT gateway
XN-312-GW-EC



EtherCAT®

EATON
Powering Business Worldwide

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

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.

1st edition 2020, publication date 07/20

Authors: Thomas Hettwer, Andreas Lüngen, Klaus-Dieter Moeller
Editor: Bettina Ewoti

Danger!
Dangerous electrical voltage!

Before starting with the installation

- Disconnect the power supply of the device.
- Secure against retriggering
- Verify isolation from the supply
- Ground and short-circuit
- Cover or enclose neighbouring units that are live.
- Follow the mounting instructions (AWA/IL) for the device.
- Only suitably qualified personnel in accordance with EN 50 110-1/-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 equipotential bonding. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed in such a way that inductive and capacitive interference will not have a negative impact on the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that cable or wire breakage on the signal side will not result in undefined states in the automation devices.
- 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 specifications, otherwise this may result in malfunction and hazardous states.
- Emergency stop devices complying with IEC/EN 60204-1 must remain functional in all of the automation devices' operating modes. Unlatching the emergency stop devices must not result in an automatic restart.
- Built-in devices for enclosures or cabinets must only be run and operated in an installed state; desktop devices and portable devices only when the housing is closed.
- Measures should be taken to ensure the proper restarting of programs interrupted after a voltage dip or outage. This should not result in dangerous operating states even for a short time. If necessary, emergency stop devices should be implemented.
- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks, etc.).

Contents

0 About this manual

This manual describes the installation, commissioning, and programming of the XN-312-GW-EC gateway.

The gateway is an integral part of the XN300 system, as are the slice modules with designation XN-322.

Support center

The latest version of this manual can be found in other languages on the Internet by visiting our Support Center at:

http://www.eaton.eu/documentation

By entering the search keyword "Gateway" or "XN300" into the quick search or by entering the document designation, e.g. "MN050010".

0.1 List of revisions

This is the first version of this document, meaning there are no changes.

0.2 Target group

This manual is intended for automation technicians and engineers.

Extensive knowledge of how to work with the field bus being used will make it easier to understand the contents of this manual.

A specialist knowledge of electrical engineering is needed for commissioning and programming.

0.3 Legal disclaimer

All information in this operator manual was provided by us to the best of our knowledge and belief and in accordance with the current state-of-the-art. However, this does not exclude the possibility of inaccuracies so that we cannot accept any liability for the accuracy and completeness of the information. In particular, this information does not guarantee any particular properties.

The devices described here must only be set up and operated as specified in this manual and in the installation instructions provided with the device. Installation, commissioning, operation, maintenance and refitting of the devices must only be carried out by qualified persons. The devices must only be used in the areas recommended and only in conjunction with third-party devices and components that have been approved by us. Only use is technically faultless condition is permitted. Fault-free and safe operation of the system requires proper transport, storage, installation and commissioning as well as careful operation and maintenance. If the following safety instructions are not observed, particularly with regard to commissioning and mainte-

nance of the devices by insufficiently qualified personnel and/or in the event of improper use of the devices, any hazards caused by the devices cannot be excluded. We assume no liability for any injury or damages incurred.

0.4 Device designations and abbreviations

- CoE - CAN application layer over EtherCAT
- EoE - Ethernet over Ether-CAT
- ESI - EtherCAT Slave Information can be found in XML description files
- FoE - File Access over EtherCAT
- COB-ID - Communication OBject IDentifier
- DC - Distributed Clock
- DIP - Dual Inline Package
- EC - EtherCAT
- MDP - Modular Device Profile.
- PDO - Process Data Objects
- RPDO - Receive Process Data Objects
- SDO - Service Data Objects
- SM - SyncManager
- SoE - Servo Profile over EtherCAT
- SSI - Synchronous Serial Interface
- TPDO - Transmit Process Data Objects
- XN300 - Device series, including the XN-312 gateway and XN-322 slice modules
- XML - EXtensible Markup Language; description file for representing hierarchically structured data in text file format

0.5 Writing conventions

Symbols used in this manual have the following meanings:

DANGER

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

CAUTION

Warns of the possibility of hazardous situations that could result in slight injury or even death.

NOTICE

Warns about the possibility of material damage.

Indicates useful tips.

▶ Indicates instructions to be followed.

For greater clarity, the name of the current chapter and the name of the current section are shown at the top of each page.

1 EtherCAT – General information

This section goes over the various special characteristics behind the EtherCAT protocol. For more detailed information, please visit the EtherCAT Technology Group website at:

http://www.ethercat.org/

1.1 How EtherCAT works

Ethernet for Control Automation Technology, or EtherCAT for short, is an Ethernet-based field bus system with special properties that make it ideal for the needs of automation applications.

In an EtherCAT system, data packets are sent by a controller and go through every EtherCAT node one after the other. Once a data packet reaches the last node, this node will detect that its EtherCAT OUT port is not connected to anything and will then send the data back through the line. Before this, however, every node will process the data intended for it on the fly as the packet keeps being sent forward, with the result being that the delay in transmitting the data packet to the next downstream node is negligible and has a magnitude of only a few nanoseconds. From an Ethernet perspective, this means that an EtherCAT bus segment is a large Ethernet node and that data is transferred between this node and the EtherCAT master. Within a segment itself, the EtherCAT nodes are connected to each other via an EtherCAT IN line and an EtherCAT Out line, which makes it possible for an Ethernet data packet to reach every single node in both the transmission and reception directions so that these nodes can process the corresponding data. This makes it possible to make full use of the full-duplex properties of 100BASE-TX, achieving a payload data rate of over 90%.

A pure EtherCAT system is created when direct communications are established without a switch.

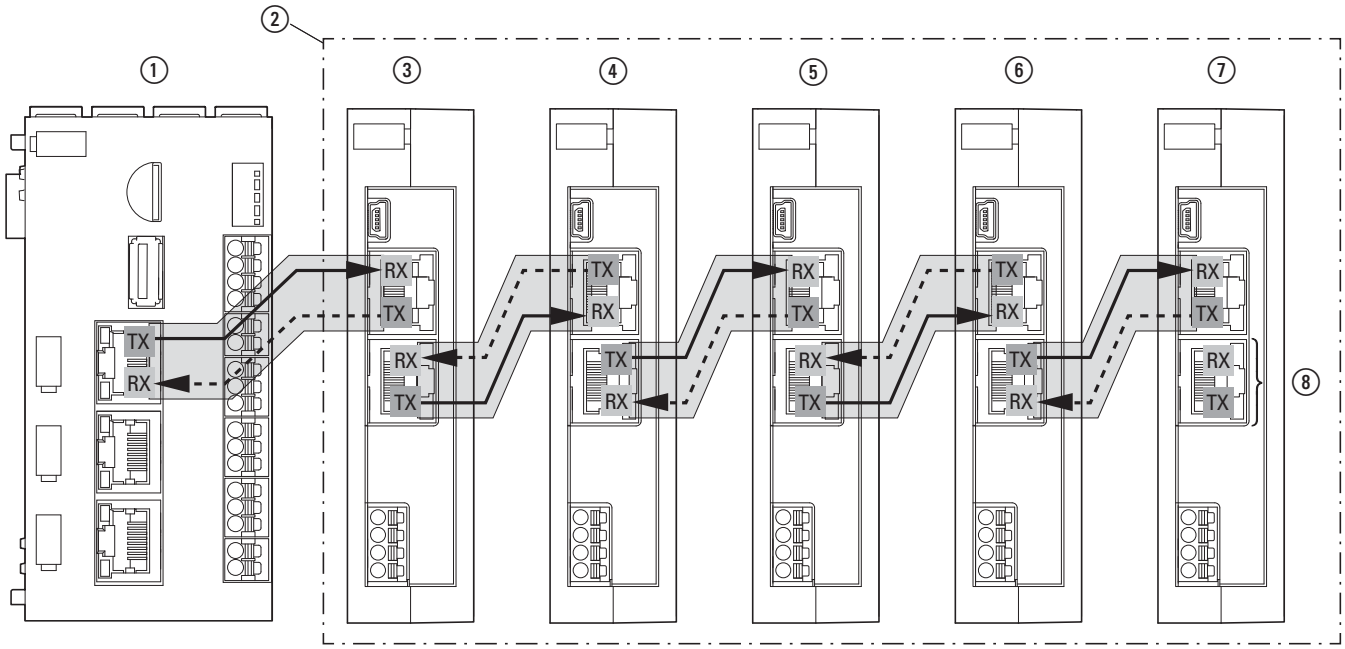


Figure 1: EtherCAT master and nodes

① Controller with EtherCAT master
② EtherCAT segment
③ EtherCAT node 1
④ EtherCAT node 2
⑤ EtherCAT node 3
⑥ EtherCAT node 4
⑦ EtherCAT node 5
⑧ If the RJ45 X2 port (EtherCAT OUT) on the last EtherCAT node is not connected to anything, the node will internally connect the incoming Rx signal to Tx and send it back to the EtherCAT master this way.

1.2 Protocol properties

The EtherCAT protocol, which is optimized for process data, is encapsulated within a standard Ethernet frame with the use of a specific EtherType (0x88A4). Other Ethernet protocols are tunneled fully transparently when using EtherCAT.

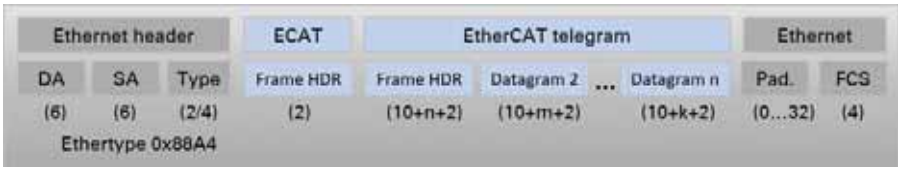


Figure 2: EtherCAT: Standard Ethernet frame as defined in IEEE 802.3

An EtherCAT frame contains one or more datagrams and can be made up of multiple subframes so as to transmit large amounts of data.

Within this context, the data sequence is independent of the physical order of the nodes on the network and specific addresses can be targeted. The datagram header specifies the type of access that the master wants on the network:

- Read, write, or read & write
- Access to a specific node with direct addressing or access to multiple slaves with logical addressing (implicit addressing)

The configuration and the process data mapping are configured by the master in the slaves during startup.

Logical addressing is used for cyclical process data transfers. When using this method, each datagram addresses a specific section of the process image table on the EtherCAT segment. Moreover, when the network starts up, each slave is assigned one or more addresses in the corresponding address space. In addition to cyclical data, other datagrams can be inserted as well in order to use asynchronous or event-driven communication.

In addition to logical addressing, the master can address a node based on its position on the network. This option is used to determine a network's topology when starting up and to then compare it to the expected configuration.

Once the configuration has been checked, the master can assign each node a configured node address and reach the nodes using these fixed node addresses. This means that accessing specific devices will still be possible even if the topology changes.

1.3 Modular Device Profile

Within the context of EtherCAT, a modular device is a device with modular and functional expansion options.

The XN-312-GW-EC EtherCAT gateway conforms to the Modular Device Profile (MDP) as defined in the EtherCAT specification (ETG 5001) based on CoE (CANopen over EtherCAT). Meanwhile, the XN-322 modules are functional expansions within this context. The modules' properties are defined in the ESI (EtherCAT slave information) device descriptions, which in turn are grouped together in a single XML file.

Using the module configuration as a basis, a dynamic object dictionary is built for each gateway based on the modules actually connected to the gateway. This object dictionary is stored in the device's RAM when restarting.

Within this context, every XN-322 module has entries for input, output, configuration, and information data that are assigned directly to the module.

The object dictionary, process data objects (PDO), and service data objects (SDO) are represented on the gateway, but are transmitted in an EtherCAT data frame and are not subject to the usual limitations in CANopen.

The process data objects (PDO) are used to transfer real-time data quickly and efficiently (e.g., I/O data, setpoint and process values). No objects are addressed in the EtherCAT frame. Instead, the contents of the process data for previously mapped parameters are sent directly.

The service data objects (SDOs) constitute the communication channel for transmitting device parameters (e.g., programming encoder resolutions). Since these parameters are transmitted acyclically (e.g., only once when the network starts up), SDO objects have a lower priority.

As per the MDP, there are two defined areas in the object dictionary:

- Communication area
- Device parameter area

File Access over EtherCAT (FoE), Servo Profile over EtherCAT (SoE), and Ethernet over EtherCAT (EoE) are not supported as of this writing.

2 XN-312-GW-EC gateway

2.1 Proper use

XN-312-GW-EC gateways are part of the XN300 system. They act as EtherCAT devices and make it possible to connect an XN300 I/O group to a controller's EtherCAT master using the EtherCAT field bus system. In fact, the XN-312-GW-EC gateway supports groups of up to 32 XN300 slice modules.

In addition to these gateways, the XN300 system also includes various XN300 slice modules that include both digital and analog I/O modules, as well as specialty modules with counter, weighing, and motor driver functionalities, that can be connected to the gateway as a functional unit. These modules can be joined together without the use of tools in order to form a system block. All XN300 system slice modules communicate with the gateway through the system bus.

The system bus is not designed for transmitting safety-relevant signals and must not be used as a replacement for controllers such as burner, crane, and two-hand safety controllers.

The following diagram shows the connection between an XN-312-GW-EC gateway and its associated components. The gateway is designed in such a way that it is compatible and can communicate with any EtherCAT master that meets the specification requirements.

→ XN-312-GW-EC gateways support XN300 slice modules of a specific version or higher. XN300 slice modules of a lower version are not compatible with XN-312-GW-EC gateways; please refer to the following as well → Section “5.2 XN-322 slice modules supported by the XN-312-GW-EC”, page 44.

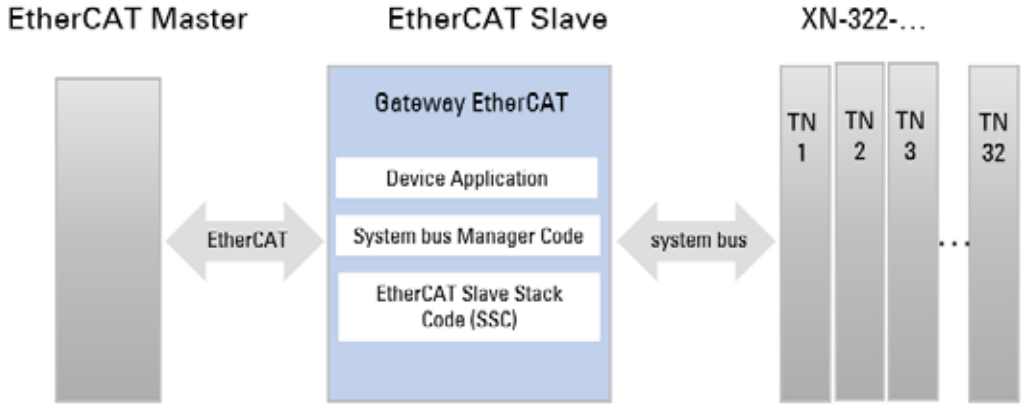


Figure 3: Connection between XN-312-GW-EC gateway and components

2.2 Overview of functions

XN-312-GW-EC gateways can be used to connect a system bus to EtherCAT, and make it possible to access the data of up to 32 I/O modules using EtherCAT. Accordingly, these gateways can be integrated as modular field bus modules into control systems that use this type of communication, making it possible to access the data of every single individual system bus module from a PLC. The gateways will automatically detect the I/O slice modules present on the system bus and send the I/O slice module configuration to the EtherCAT master.

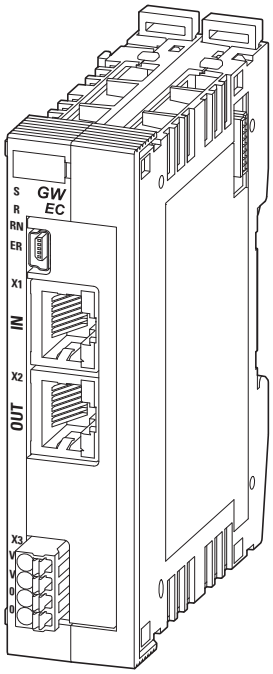


Figure 4: The XN-312-GW-EC gateway can be used to establish a connection to an EtherCAT field bus.

The RJ45 X1 (EtherCAT IN) and X2 (EtherCAT OUT) ports are used to establish a connection to the network. EtherCAT IN refers to the connection direction towards the controller, while EtherCAT OUT is used to connect to the next downstream node or set up redundant communications.

The gateway's 24 V DC POW power supply needs to be connected to X3. A system bus is used for data communication and for supplying power within the system block. The POW power supply is used to produce the power for the system bus, i.e., 5-V power for electronics and 24-V power for analog modules and specialty modules.

The USB diagnostic interface on XN-312-GW-EC gateways can be used to connect them to a computer in order to use the XN300-Assist planning and commissioning program. In addition to various online and offline functions, XN300-Assist provides the option of updating the XN-312-GW-EC gateway's operating system.

2.3 Versions

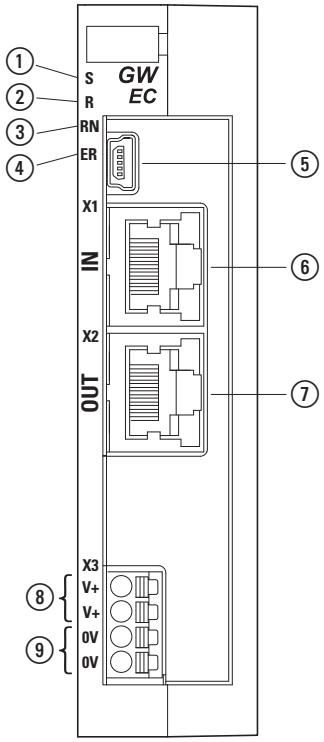


Figure 5: Front view of XN-312-GW-EC

① LED S, Sync status
② R LED, Reset status
③ LED RN, EtherCAT Run status
④ EC LED, EtherCAT error status
⑤ Mini-USB diagnostic interface
⑥ EtherCAT IN
⑦ EtherCAT OUT
⑧ Power supply, 24 VDC POW – the two terminals are internally connected to each other
⑨ 0 V, the two terminals are internally connected to each other

2.4 Field bus connection

Both RJ45 X1 (EtherCAT IN) and X2 (EtherCAT OUT) ports are used to establish a connection to the network. Both ports feature an LED that gives information on the port's connection status. The data transfer rate for EtherCAT is based on Ethernet with a fixed full-duplex rate of 100 Mbit/s and a maximum transmission distance of 100 m.

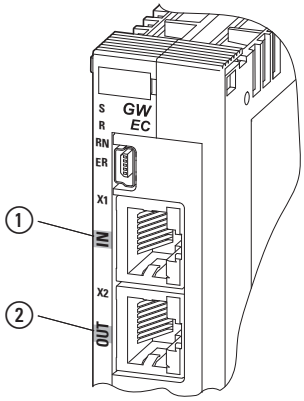
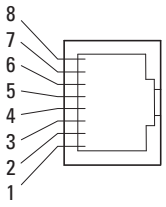


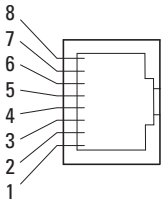
Figure 6: The configuration of the terminal sockets at the gateway
① ?X1 EtherCAT IN
② X2 EtherCAT OUT

① LED IN yellow
② LED OUT yellow

EtherCAT IN

X1	Pin	Function
	1	Tx+/Rx+
	2	Tx-/Rx-
	3	Rx+/Tx+
	4, 5	–
	6	RX-/Tx-
	7, 8	–

EtherCAT OUT

X2	Pin	Function
	1	Tx+
	2	Tx-
	3	Rx+
	4-5	–
	6	RX-
	7, 8	–

2.5 Diagnostics interface

The diagnostic interface on the XN-312-GW-EC can be used to establish a connection to a PC in order to download firmware or connect XN300-Assist to the EtherCAT gateway.

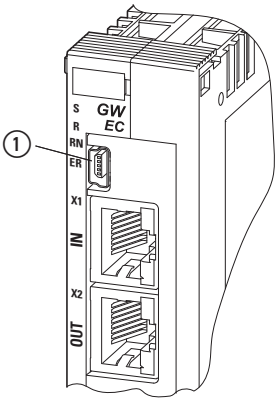
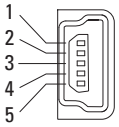


Figure 7: Mini USB 2.0 type B diagnostic port

Mini USB 2.0 type B dagnostic port	Pin	Function
	1	+5 V
	2	D-
	3	D+
	4	–
	5	GND

2.6 Supply voltage connection

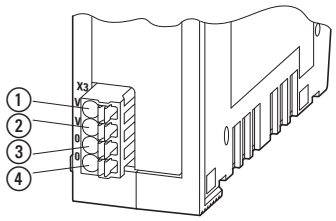
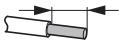
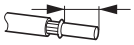
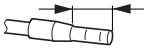


Figure 8: Specifications for connection to supply voltage

① +24V V+
② +24V V+
③ GND 0V
④ GND 0V

The two V+ terminals are internally connected to each other (X4: ① and ②), as are the two 0 V terminals (X4: ③ and ④). Only one V+ and one 0 V terminal need to be connected in order to power the gateway The internally connected terminals can be used to extend the 24 V and 0 V supply voltage connections. However, it is necessary to make sure that a total current of 6 A is not exceeded per terminal.

The plug connector with push-in spring-cage terminals is included in the scope of supply.

XN-312-…	**XN-312-GW-EC**
I_e [A]	≦ 2.0
X3	
10 mm (0.39“) 	0.2 - 1.5 mm^2
10 mm (0.39“) 	0.2 - 1.5 mm^2
10 mm (0.39“) 	0.25 - 0.75 mm^2
AWG	AWG 24 - 16

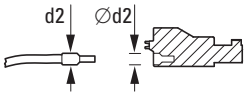


Figure 9: Ferrule with d2 = Max. 2.8 mm

→ The spring-cage terminals can be used to connect ultrasonically spliced (ultrasonically welded) wires.

2.7 LED status indicators

The device's status LEDs are located underneath the front cover and can be seen underneath the corresponding labels when they light up.

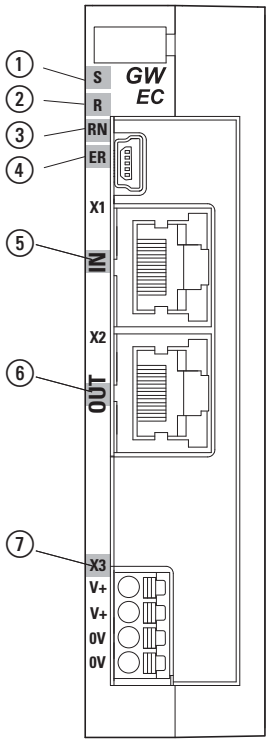


Figure 10: LEDs on the device

① S Sync (green)
② R Reset (red)
③ RN EtherCAT Run
④ ER EtherCAT Error
⑤ IN EtherCAT Link/Active (LA)
⑥ OUT EtherCAT Link/Active (LA
⑦ X3 DC power OK (green)

Table 1: What the LEDs mean

LED	Color	EtherCAT status	Description
S (Sync)			
	green	Continuous light	System bus running synchronously
	AUS	–	System bus not running synchronously
R (Reset)			
	red		System bus RESET in progress
	AUS	–	System bus currently not being reset

LED	Color	EtherCAT status	Description
RN (EtherCAT Run)			
	green	Continuous light	OPERATIONAL
		Fast flashing	INITIALIZATION or BOOTSTRAP
		Single flash	SAFE OPERATIONAL
		Fast flickering	PRE-OPERATIONAL
	AUS	–	INIT
ER (EtherCAT Error)			
	red	Continuous light	PDI watchdog timeout The field bus has been turned off due to an error, e.g., • Wiring faults • Module with wrong baud rate on bus
		Fast flashing	Boot error
		Double flash	Watchdog timeout
		Single flash	General runtime error
		Fast flickering	Configuration not OK
	AUS	–	No errors or faults
IN (EtherCAT)			
	yellow	Continuous light	Connection to upstream EtherCAT node established
		Flashing	Data communication with upstream EtherCAT node established
	AUS	–	No connection to upstream EtherCAT node
OUT (EtherCAT)			
yellow	yellow	Continuous light	Connection to downstream EtherCAT node established
		Flashing	Data communication with downstream EtherCAT node established
	AUS	–	No connection to downstream EtherCAT node
X3 (DC power OK)			
	green	Continuous light	+24 VDC power OK
	AUS	–	+24 VDC power faulty

2.8 Potential relationship between the components

The entire XN300 system operates with a common supply voltage. All XN300 slice modules feature a contact point on the back that is used to establish a functional earth connection to the DIN-rail. Moreover, the protective earth and the functional earth have the same potential in XN300 systems. Finally, the EtherCAT field bus and the XN300 system are galvanically isolated from each other.

Common

- 0V
- ⏚

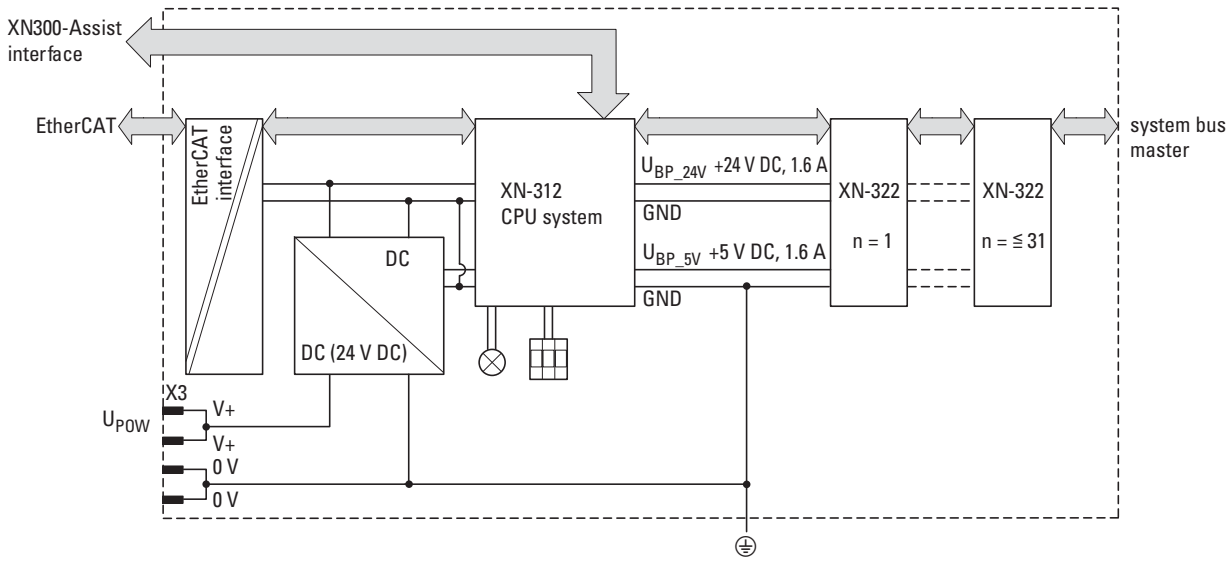


Figure 11: Function principle of XN300 system

3 The EtherCAT finite state machine

The are various states and state transitions that are defined in the EtherCAT protocol. XN-312-GW-EC gateways implement these EtherCAT states using the system bus for communication with the XN300 slice modules.

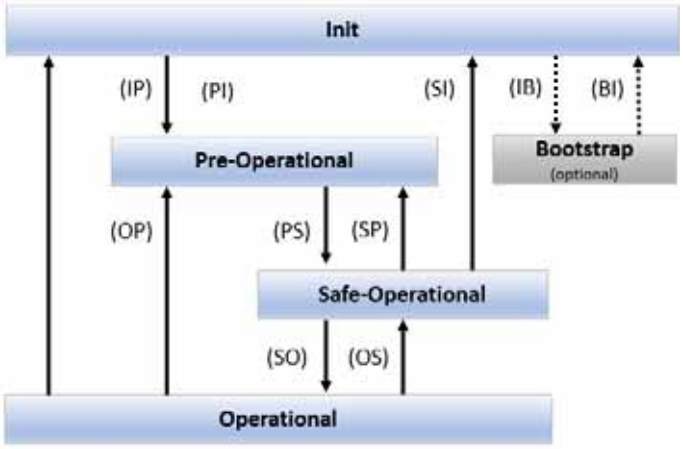


Figure 12: EtherCAT states and state transitions

State	Description	Notes
INIT	Device booting. The EtherCAT master will search for nodes on the system bus and generate an object list. The SDO/PDO nodes configuration is being generated and stored as a CoE object dictionary. The Mailbox SyncMaster is being configured.	The module status LED on the station's nodes will flash green.
INIT -> PRE-OPERATIONAL	• Mailbox SyncManager (SM) will be configured. • Distributed Clock (DC) will be configured.	
PRE-OPERATIONAL	The device has been started and has checked in on the EtherCAT bus The PDO transfer is being initialized but has not yet started. Input and output data is not being transmitted yet.	
PRE-OPERATIONAL -> SAFE OPERATIONAL	• The module list will be read by the EtherCAT master • PDO SMs will be configured • FMMUs will be configured • PDO mapping will be written by the master • Configuration of XN300 slice modules via CoE. • The system bus is starting with the PDOs for the inputs	

State	Description	Notes
SAFE OPERATIONAL	The device has been initialized and is active. PDOs are being transmitted to the master. Inputs are being read. Outputs will remain in a safe state of "0".	The module status LED on the station's nodes will show a solid green light.
SAFE OPERATIONAL -> OPERATIONAL	• The system bus is starting with the PDOs for the outputs.	
OPERATIONAL	The device has been initialized and is active. PDOs are being written to the master. Inputs are being read. Outputs will be written.	

4 XN-312-GW-EC gateway object dictionary

For more detailed information on the object dictionary, the basic structure of the individual objects, and the modular device profile used, please refer to the corresponding EtherCAT specifications:

- Part 1: General MDP Device Model (ETG.5001.1 S)
- Part 3: MDP Fieldbus Gate-way Profile Specifications“(ETG.5001.3 S)

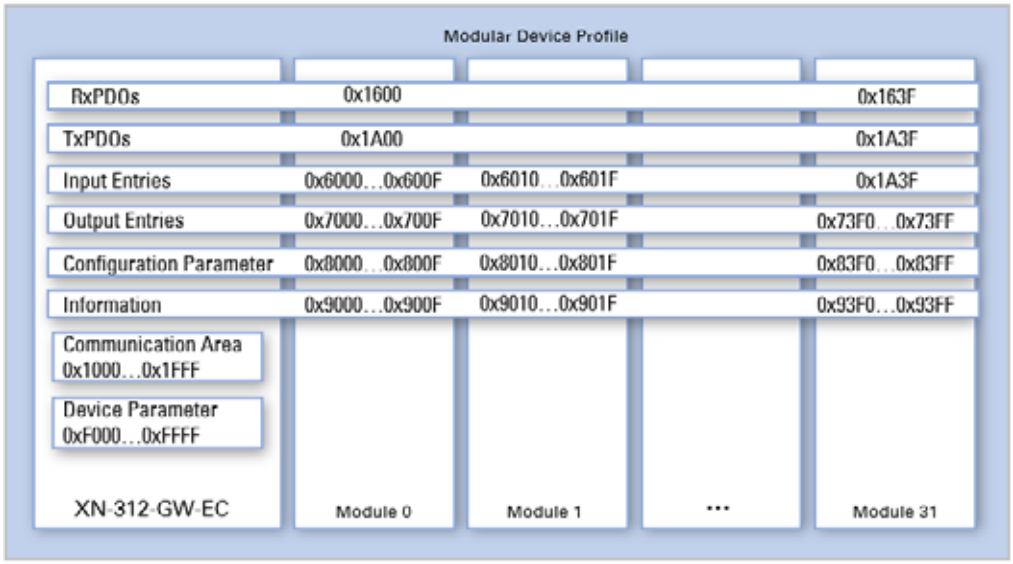


Figure 13: Modular device profile

4.1 Communication area

All communication area mandatory objects are supported. Certain communication area mandatory objects are conditional objects and will only appear in the object dictionary if the modules connected to the gateway require them. RxPDO mapping objects, for example, will only be integrated if output modules are plugged into the station.

The entries in the EtherCAT object dictionary area specific to these objects are created statically. They are found in indexes 0x1000 to 0x1FFF and can be accessed with SDOs as soon as the gateway is communicating on the EtherCAT bus.

The following table provides an overview of all the objects supported by the XN-312-GW-EC gateway.

Table 2: Object overview I

Index	Use M - Mandatory O - Optional C - Conditional	Name
0x1000	M	Device type (0×00005001)
0x1008	M	Device name (XN-312-GW-EC)
0x1009	M	Hardware version
0x100A	M	Software version
0x1018	M	Identity (device identification)
0×1018/#01	O	Vendor ID (455)
0×1018/#02	O	Product code (67108868)
0×1018/#03	O	Revision
0×1018/#04	O	Serial Number
0×10F1	O	Error register
0×10F1/#01	O	Local error reaction
0×10F1/#02	O	Sync error counter limit (200)
0×1600…0×17FF	C	RxPDO mapping Mandatory object for output modules
0×1A00 - 0×1BFF	C	TxPDO mapping Mandatory object for input modules
0×1C00	C	Sync manager type
0×1C00/#01	O	Sync manager 0
0×1C00/#0	O	Sync manager 1
0×1C00/#03	O	Sync manager 2
0×1C00/#04	O	Sync manager 3
0×1C12	C	RxPDO Assign PDO Assignment is a mandatory object for output modules
0×1C13	C	TxPDO Assign PDO Assignment is a mandatory object for input modules
0×1C32	C	SM output parameter Synchronization Manager output parameter
0×1C32/#01	C	Sync mode (2)
0×1C32/#02	C	Get cycle time
0×1C33	C	SM input parameter Synchronization Manager input parameter
0×1C33/#01		Sync mode (2)
0×1C33/#02		Get cycle time
A description of the following objects can be found in the user manuals for the corresponding XN-322 slice modules		

4.1.1 Device type (0x1000)

Object 0x1000 contains the type and function of the EtherCAT device.

A value of 0000 1389_{hex} indicates that the device is a device with a modular device profile (MDP).

Index (hex)	Object	Name	M/O
0x1000	VAR	device type	M

Sub-index (hex)	Description	Default	Catalog number	Access
0x00	device type	0x00001389	UDINT	const

4.1.2 Device name (0x1008)

The object contains the vendor-specific device name.

Index (hex)	Object	Name	M/O
0x1008	VAR	manufacturer device name	M

Sub-index (hex)	Description	Default	Catalog number	Access
0x00	manufacturer device name	XN-312-GW-EC	Visible string	const

4.1.3 Hardware version (0×1009)

Object 0x1009 contains the designation for the hardware version.

Index (hex)	Object	Name	M/O
0x1009	VAR	manufacturer hardware version	M

Sub-index (hex)	Description	Default	Catalog number	Access
#00	manufacturer hardware version	–	Visible string	const

4.1.4 Software version (0×100A)

The object contains the EtherCAT gateway's software version.

Index (hex)	Object	Name	M/O
0x100A	VAR	manufacturer software version	M

Sub-index (hex)	Description	Default	Catalog number	Access
#00	manufacturer software version	–	Visible string	const

4.1.5 Identity object (0×1018)

Object 0x1018 contains manufacturer-specific information regarding the XN-312 gateway.

The vendor ID (sub-index #01) is a unique ID that is used to clearly identify the manufacturer. Meanwhile, the vendor-specific product code (sub-index #02) is used to identify the specific device version in question.

The vendor-specific revision number (sub-index #03) consists of a major revision number and a minor revision number.

Index (hex)	Object	Name	M/O
0x1018	RECORD	Identity object	M

Sub-index (hex)	Description	Default	Catalog number	Access
#00	number of entries	–	–	–
#01	vendor ID	0x000088FA	UDINT	ro
#02	product code	0x04000004	UDINT	ro
#03	revision	0	UDINT	ro
#04	serial number	0	UDINT	ro

4.1.6 RxPDO mapping object (0×1600…0×17FF)

The PDO mapping objects are used to define the structure of the PDOs as per ETG 1000.6 (Application Layer Protocol Specification). The 0x16xx objects contain the corresponding process data mapping assignments for the connected XN300 slice modules. The content depends on the module configuration.

The numbering of the PDOs depends on the module's position in the XN300 system and is generated as follows:

- Index = 0×1600 + (position number - 1) for output modules

Within this context, an object consists of one or more sub-objects, while the sub-object represents the module channel.

For each module, special RxPDO objects that are named based on the module name and type are created.

Example: "XN-322-8DO-P05 RxPDO mapping"

4.1.7 TxPDO mapping object (0×1A00…0×1BFF)

The PDO mapping objects are used to define the structure of the PDOs as per ETG 1000.6 (Application Layer Protocol Specification). The 0x1Axx objects contain the corresponding process data mapping assignments for the connected XN300 slice modules. The content depends on the module configuration.

The numbering of the PDOs depends on the module's position in the XN300 system and is generated as follows:

- Index = 0×1A00 + (position number - 1) for input modules

Within this context, an object consists of one or more sub-objects, while the sub-object represents the module channel.

For each module, special TxPDO objects that are named based on the module name and type are created.

Example: "XN-322-8DO-P05 TxPDO mapping"

4.1.8 Sync manager type (0×1C00)

This object describes the use of the Sync Manager channels and contains the type of the individual EtherCAT Sync Managers.

Index (hex)	Object	Name	M/O
0x1C00	ARRAY	Sync manager type	O

Sub-index (hex)	Description	Default	Catalog number	Access
#00	number of entries	–	USINT	ro
#01	SUB-index #01	02hex (Mailbox IN)	USINT	ro
#02	SUB-index #21	01hex (Mailbox OUT)	USINT	ro
#03	SUB-index #03	04hex (Process Data IN)	USINT	ro
#04	SUB-index #04	03hex (Process Data OUT)	USINT	ro

4.1.9 RxPDO assign object (0x1C12)

Object 0x1C12 references the RxPDO mapping objects and defines which PDOs should be transmitted with the EtherCAT output data. (Access: RO.)

4.1.10 TxPDO assign object (0x1C13)

Object 0x1C13 references the TxPDO mapping objects and defines which PDOs should be transmitted with the EtherCAT input data. (Access: RO.)

4.1.11 Sync Manager input/output parameter (object 0x1C32, object 0x1C33)

These objects contain information regarding the synchronization behavior as specified in ETG1020 (EtherCAT Protocol Enhancements).

Object 0x1C32 contains the sync manager output parameters and object 0x1C33 contains the sync manager input parameters for all configured XN300 slice modules in an XN300 system block.

Index (hex)	Object	Name	M/O
0x1C32	ARRAY	Sync manager output parameter	O
0x1C33		Sync manager input parameter	O

Sub-index (hex)	Name	Default	Catalog number	Access
Sub-index #00	number of entries	–	USINT	ro
Sub-index #01	sync mode	–	UINT	rw
Sub-index #02	cycle time	–	UDINT	ro
Sub-index #04	sync modes suppored	–	UINT	ro
Sub-index #05	min cycle time	–	UDINT	ro
Sub-index #06	calc and copy time	–	UDINT	ro
Sub-index #08	get cycle time	–	UINT	rw
Sub-index #09	delay time	–	UDINT	ro
Sub-index #0A	sync0 cycle time	–	UDINT	ro
Sub-index #0B	cycle time small	–	UINT	ro
Sub-index #0C	sync manager event missed counter	–	UINT	ro
Sub-index #20	Sync error	–	Bool	ro

4.2 Module object area (0×6000 - 0×AFFF)

Object area	Index range	Modular device
Input data object area	0x6xxx	Conditional
Output data object area	0x7xxx	Conditional
Configuration data object area	0x8xxx	Optional
Information data object area	0x9xxx	Optional
Diagnosis data object area	0xAxxx	Optional

4.2.1 Input data object area (0×6xxx)

An input data object with multiple sub-objects is assigned to every module with input data. The number of sub-objects within this context depends on the module's number of channels.

Input data objects are mapped to TxPDOs that are read cyclically by the EtherCAT master. The input data objects' index depends on the module's position in the XN300 system:

- Index = 0×6000 + (position number - 1) x 0×0010

Input data objects are automatically mapped in the system configuration and updated in the cyclical process data transfer.

Index TxPDO	Functional group
0x6xx0	Digital input
0x6xx1	Analog input
0x6xx2	Analog range
0x6xx3	Status word
0x6xx4	Counter/Encoder
0x6xx5	PWM period
0x6xx6	Timestamp
0x6xx7	–
0x6xx8	–
0x6xx9	Measurement value
0x6xxA	Status
0x6xxB	Miscellaneous
0x6xxC	–
0x6xxD	Error bits
0x6xxE	Reference voltage OK
0x6xxF	Voltage OK

4.2.2 Output data object area (0×7xxx)

An output data object with multiple sub-objects is assigned to every module with output data. The number of sub-objects within this context depends on the module's number of channels. Output data objects are mapped to RxPDOs that are written cyclically by the EtherCAT master.

The output data objects' index depends on the module's position in the XN300 system:

- Index = 0×7000 + (position number - 1) x 0×0010

Output data objects are automatically mapped in the system configuration and updated in the cyclical process data transfer.

Index RxPDO	Functional group
0x7xx0	Digital output
0x7xx1	Analog output
0x7xx2	–
0x7xx3	Control word
0x7xx4	PWM
0x7xx5	Time offset
0x7xx6	–
0x7xx7	–
0x7xx8	DriveCMD
0x7xx9	Command
0x7xxA	Config
0x7xxB	Miscellaneous
0x7xxC	–
0x7xxD	–
0x7xxE	–
0x7xxF	–

4.2.3 Configuration data object area (0×8xxx)

A configuration data object with multiple sub-objects is assigned to every module with output data. The number of sub-objects within this context depends on the module's number of channels. The EtherCAT master writes this object when transitioning from the PRE-OP state to the SAFE-OP state.

The configuration data objects' index depends on the module's position in the XN300 system:

- Index = 0×8000 + (position number - 1) x 0×0010

Configuration data objects are automatically mapped in the system configuration and transmitted to the XN300 station during the initialization phase for configuration. This value can also be changed at runtime with SDO communication.

Since the 0x8xx1 LED control object is not required for configuration purposes, it can be mapped manually or set at runtime using SDO communication.

Index (RxPDO)	Functional group
0x8xx0	–
0x8xx1	LED control
0x8xx2	–
0x8xx3	–
0x8xx4	PWM
0x8xx5	Time offset
0x8xx6	–
0x8xx7	–
0x8xx8	DriveCMD
0x8xx9	Command
0x8xxA	Config
0x8xxB	Miscellaneous
0x8xxC	–
0x8xxD	–
0x8xxE	–
0x8xxF	–

4.2.4 Module information (0×9xxx)

An information data object is assigned to every XN300 slice module integrated into the system. The information data objects' index depends on the module's position in the XN300 system:

- Index = 0×9000 + (position number - 1) x 0×0010

Information data objects are automatically mapped in the system configuration and updated in the cyclical process data transfer. Since the information is information from the station's XN300 slice modules, the data corresponding to the READ-ONLY objects can be viewed in the CoE online view or read with SDO communications at runtime.

Index (hex)	Object	Name	M/O
0x9000… 0x900F	RECORD	Module information	Conditional

Sub-index (hex)	Descrip-tion	Default	Catalog number	Access
#00	number of entries	–	USINT	ro
#01	module status	- (0 = Module OK)	UDINT	ro
#02	device ID	00000000_{hex}	UDINT	ro
#03	FPGA version	00000000_{hex}	UDINT	ro
#04	HW version	00000000_{hex}	UDINT	ro
#05	Serial number	0	Visible string	ro
#06	Firmware version	00000000_{hex}	UDINT	ro

4.3 Device parameter area

The device parameter area contains all the parameters that belong to the EtherCAT device (gateway). Object information is shown in XSOFT-CODE-SYS-3 in online mode.

Index Sub-index	Description / Value
0xF000	Modular device profile The modular device profile object contains all information required in order to interpret the object area's module objects.
Sub-index #01	Module index distance (16) Maximum number of objects per module.
Sub-index #02	Maximum number of modules (32) Max. 32 modules on the gateway
Sub-index #03	General configuration Available sub-indices in the 0×8xx0 general configuration objects
Sub-index #04	General information Available sub-indices in the 0×9xx0 general information objects
Sub-index #05	Module PDO group of devices
0xF030	Configured module ident list, ID list of the configured XN-322 modules
0×F030: Sub-index#01...#20	SubIndex 001 -SubIndex 020 Module ID number as per the module's position in the XN300 configuration (1 to 32)
0xF050	Detected module ident list, ID list of mounted XN-322 modules
0×F050: Sub-index#01...#20	SubIndex 001 -SubIndex 020 Module ID number as per the module's position in the XN300 system (1 to 32)
0xF100	System bus version
Sub-index #01	Systembus manager protocol version

Index Sub-index	Description / Value
Sub-index #02	Systembus manager option bits
Sub-index #03	Systembus manager FPGA version
0xF101	XN-312-GW-EC info
Sub-index #01	Serial Gateway serial number
0xF110	System bus diagnostics
Sub-index #01	Retry counter
0xF111	Error info
0×F111: Sub-index#01	Error counter
0×F111:Sub-index#02	New error counter
0×F111: Sub-index#03	Last error
0×F111: Sub-index#04	Second last error
0×F111: Sub-index#05	Third last error
0×F111: Sub-index#06	Fourth last error
0×F111: Sub-index#07	Fifth last error
0xF120	System bus statistics
Sub-index #01	IsoWriteTaskDuration
Sub-index #02	IsoWriteTaskMaxDuration
Sub-index #03	IsoReadTaskDuration
Sub-index #04	IsoReadTaskMaxDuration
0xFB00	Device control
0×FB00: Sub-index#01	Software restart (0)
0xFB10	System bus configuration
0xFB10: Sub-index#01	Free run cycle time (4000); [in µs]
0xFB10:Sub-index #02	IsoStartPoint (50)

4.3.1 Modular device profile Object 0xF000

Object 0xF000 contains basic information regarding the modular device profile used.

Index (hex)	Object	Name	M/O
0xF000	VAR	Device type	M

Sub-index (hex)	Description	Default	Catalog number	Access
#00	number of entries	–	USINT	ro
#01	Module index distance	$0x10_{hex}$	UINT	ro
#02	Maximum number of modules	$0x20_{hex}$	UINT	ro
#03	General configuration	$0x00000001_{hex}$	UDINT	ro
#04	General information	$0x00000000_{hex}$	UDINT	ro
#05	Module PDO device group	$0x0000_{hex}$	UINT	ro

4.3.2 Configured module ident list ((0×F030)

Object 0xF030 contains a list of the configured XN300 slice modules with the corresponding module ID numbers.

Consecutive sub-indexes #01 ≦ nh ≦ #1F (31_{dec}) describe the corresponding XN300 slice modules in the order in which they are configured in the controller configuration. Moreover, each entry contains a number identifying the corresponding module, see also → Section "5.1 Module ID number", page 43.

Index (hex)	Object	Name	M/O
0xF030	ARRAY	Configured module ident list	O

Sub-index (hex)	Description	Default	Catalog number	Access
#00	number of entries	–	USINT	rw
#01…#20	module indent number	–	USINT	rw

4.3.3 Detected module ident list (Object 0xF050)

Object 0xF050 contains a list of the detected XN300 slice modules with the corresponding module ID numbers.

Object 0xF050 contains the module IDs for all the XN300 slice modules physically present in an XN300 system block.))

Meanwhile, consecutive sub-indexes #01 ≦ nh ≦ $1F_{hex}$ (31_{dec}) are used to describe the XN300 slice modules in the order in which they are installed in the XN300 station, with each sub-index corresponding to one individual slice module. Moreover, each entry contains a number identifying the corresponding module, see also → Section "5.1 Module ID number", page 43.

Index (hex)	Object	Name	M/O
0xF050	ARRAY	Detected module ident list	O

Sub-index (hex)	Description	Default	Catalog number	Access
#00	number of entries	–	USINT	ro
#01…#20	module ident number	–	UINT	ro

4.3.4 System bus version (object 0xF100)

Object 0xF100 contains information about the system bus.

Index (hex)	Object	Name	M/O
0xF100	RECORD	Systembus versions	O

Sub-index (hex)	Description	Default	Catalog number	Access
#00	number of entries	–	USINT	ro
#01	System bus manager protocol version	0	UINT	ro
#02	System bus option bits	0	UINT	ro
#03	System bus FPGA version	0	USINT	ro

4.3.5 System bus diagnostics Object 0xF110

Object 0xF110 contains diagnostic information regarding the system bus.

Index (hex)	Object	Name	M/O
0xF110	RECORD	System bus diagnostics	O

Sub-index (hex)	Description	Default	Catalog number	Access
#00	number of entries	–	USINT	ro
#01	retry counter	0x00	UDINT	ro

4.3.6 Error info Object 0xF111

Object 0xF111 contains information regarding errors that have occurred in the system.

Index (hex)	Object	Name	M/O
0xF111	RECORD	ErrorInfo	O

Sub-index (hex)	Description	Default (hex)	Catalog number	Access
#00	number of entries	–	USINT	ro
#01	Error counter	0x00000000	UDINT	ro
#02	New error counter	0x00000000	UDINT	ro
#03	Last error	0x00000000	UDINT	ro
#04	Second last error	0x00000000	UDINT	ro
#05	Third last error	0x00000000	UDINT	ro
#06	Fourth last error	0x00000000	UDINT	ro
#07	Fifth last error	0x00000000	UDINT	ro

Design of the data bytes:

The structure of the error information for the 32-bit value is as follows:

Sub-index 1≦n≦254	Byte 1								Byte 0							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	Reason code															

Sub-index 1≦n≦254	Byte 3								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Device number								Error code							

Device number	Error source
0x00…0x1F	XN300 slice module node number on the system bus generating the error
0xFF	XN-312-GW-EC

4.3.6.1 Error codes

The error codes are shown under the CoE tab when using online communication; please refer to → Section "12.7.4 ONLINE CoE XN-312-GW-EC gateway", page 92 and → Section "12.6 Diagnostics messages", page 87.

Error code	Error information
1	The I/O mapping file stored in the system bus slave FPGA could not be read
2	The I/O mapping file stored in the system bus slave FPGA could not be parsed
3	The system bus slave could not be detected
4	A standard data object could not be created
5	The mailbox handler could not be started
6	Error while attempting to determine the cycle time
7	Error while attempting to set the cycle time
8	The input handler could not be started
9	Error while attempting to stop the input handler
10	The output handler could not be started
11	Error while attempting to stop the output handler
12	Error while attempting to create the PDO mapping
13	Attempt to copy the input PDOs failed
14	Attempt to copy the output PDOs failed
15	Error during SDO access to the system bus slave
16	PDO data reading could not be started

Error code	Error information
17	PDO data reading had to be terminated
18	PDO data writing could not be started
19	PDO data writing had to be terminated
20	Attempt to call a function upon a state transition failed
21	Attempt to parse an SDO in the stored I/O mapping file failed
22	CAN handler must be restarted
23	Function failed
24	Error in ISO Read task state
25	Error in ISO Write task state
26	System bus task error
27	Error while attempting to set up info file container
28	Error while attempting to write info file container
29	Attempt to read nameplate file failed

4.3.6.2 Reason codes

Reason code	Error information
1	Cause unknown
2	No free memory space available
3	File is unusable
4	System bus is not ready
5	System bus has not confirmed the state transition
6	The module configuration on the system bus could not be checked
7	Attempt to request state transition on the system bus side failed
8	Master has attempted to change the PDO mapping data objects
9	Inconsistent configuration
10	Error while attempting to request mutex
11	Attempt to start SDO communication failed
12	Timeout
13	Entry not found
14	Module-specific function failed
15	Basic task handshake
16	PLL
17	Task status
18	SPI error
19	Slave function failed during state transition
20	Incorrect entry in system bus slave internal mapping file
21	No copying entries available anymore

Reason code	Error information
22	The switch from ISO Full to the Read-Only task state on the system bus failed
23	Excessively long filename
24	Bad checksum

4.3.7 System bus statistics (0×F120)

Object 0xF120 contains statistical data regarding the system bus.

Index (hex)	Object	Name	M/O
0xF120	RECORD	SDIAS statistics	O

Sub-index (hex)	Description	Default (hex)	Catalog number	Access
#00	number of entries	–	USINT	ro
#01	ISO write task duration	0x00000000	UDINT	ro
#02	maximum ISO write task duration	0x00000000	UDINT	ro
#03	ISO read task duration	0x00000000	UDINT	rw
#04	maximum ISO read task duration	0x00000000	UDINT	rw

4.3.8 Device control (0×FB00)

Object 0xFB00 contains gateway monitoring options.

Index (hex)	Object	Name	M/O
0xFB00	RECORD	Device control	O

Sub-index (hex)	Description	Default (hex)	Catalog number	Access
#00	number of entries	–	USINT	ro
#01	software restart	0x00	USINT	rw

4.3.9 System bus configuration (0×FB10)

Object 0xFB10 contains configuration options for the system bus.

Index (hex)	Object	Name	M/O
0xFB10	RECORD	SDIAS configuration	O

Sub-index (hex)	Descrip-tion	Default	Catalog number	Access
#00	number of entries	–	USINT	ro
#01	Free run cycle time	4000 [µs]	UINT	rw
#02	ISO start time as a %	50 [%]	USINT	rw

In free run mode, this object can be used to define the cycle time on the system bus with sub-index 0x01. The default value is 4 ms. In order for it to be possible to set the cycle time, a startup command must be set for the state transition from Init to PreOp or PreOp to SafeOp. The standard cycle times for the system bus are accepted.

Valid free run cycle time [µs]
250
500
1000…32000 (in increments of 1000 µs)

If an invalid value is entered, the gateway will ignore it and the default value of 4 ms will be used.

The ISO start time can be defined within a range of 10 to 90%.

5 XN300 I/O module support

5.1 Module ID number

Each XN300 slice module has a specific identification number. There is an individual identification number for each type of XN-322 slice module.

XN300 slice module	Identification number Default value <default> decimal
XN-322-20DI-PD	8001_{dec}
XN-322-20DI-PCNT	8002_{dec}
XN-322-20DI-PF	8003_{dec}
XN-322-12DO-P17	8004_{dec}
XN-322-16DO-P05	8005_{dec}
XN-322-2DMS-WM	8006_{dec}
XN-322-4AI-PTNI	8007_{dec}
XN-322--7AI-U2PT	8008_{dec}
XN-322-8AI-I	8009_{dec}
XN-322-10AI-TEKT	8010_{dec}
XN-322-8AIO-U2	8011_{dec}
XN-322-8AIO-I	8012_{dec}
XN-322-8AO-U2	8013_{hex}
XN-322-1DCD-B35	8014_{dec}
XN-322-1CNT-8DIO:	8015_{dec}
XN-322-2SSI	8016_{dec}
XN-322-4DO-RNO	8018_{dec}
XN-322-20DI-ND	8019_{dec}
XN-322-16DI-PD	8020_{dec}
XN-322-8DI-PD	8021_{dec}
XN-322-16DIO-PD05	8022_{dec}
XN-322-16DIO-PC05	8023_{dec}
XN-322-8DIO-PD05	8024_{dec}
XN-322-8DO-P05	8025_{dec}
XN-322-4AIO-U2	8026_{dec}
XN-322-4AIO-I	8027_{dec}

5.2 XN-322 slice modules supported by the XN-312-GW-EC

The XN-312-GW-EC supports the following XN300 slice modules. The version is printed on the right side of every XN300 slice module (e.g., "Version: 3.02").

Table 3: List of supported XN-322- … slice modules

Catalog number	Description	Supported starting from version
Power supply modules	XN-322-4PS-20	3.00
	XN-322-18PD-M	3.00
	XN-322-18PD-P	3.00
Digital I/O modules	XN-322-8DI-PD	3.01
	XN-322-16DI-PD	3.01
	XN-322-20DI-PD	3.00
	XN-322-20DI-PF	3.00
	XN-322-20DI-PCNT	3.00
	XN-322-20DI-ND	3.00
	XN-322-8DO-P05	3.00
	XN-322-12DO-P17	3.00
	XN-322-16DO-P05	3.00
	XN-322-8DIO-PD05	3.01
	XN-322-16DIO-PD05	3.00
	XN-322-16DIO-PC05	3.00
Analog I/O modules	XN-322-4AI-PTNI	3.01
	XN-322-7AI-U2PT	4.05
	XN-322-8AI-I	4.03
	XN-322-10AI-TEKT	3.04
	XN-322-8AO-U2	3.00
	XN-322-4AIO-U2	4.04
	XN-322-8AIO-U2	4.06
	XN-322-4AIO-I	4.03
	XN-322-8AIO-I	4.05
Technology modules	XN-322-2DMS-WM	3.04
	XN-322-1DCD-B35	3.00
	XN-322-1CNT-8DIO	4.03
	XN-322-2SSI	3.00
	XN-322-4DO-RNO	3.01

For up-to-date information on the firmware and the XML files, please visit the Download center, → Page 121.

6 Operating modes

The XN-312-GW-EC supports free run mode and DC synchronous mode.

6.1 Free run

In "free run" mode, input values are applied and output values are output cyclically, with the cycle being based on a timer in the node that functions as a trigger. The corresponding cycle time can be changed by the master (optional).

In "free run" mode, the local cycle works independently of the communication cycle and/or the master cycle.

6.2 Distributed clocks (DC synchronous)

"Distributed clocks" refers to a logical network of distributed clocks that are run locally in the various network nodes and that synchronize themselves with each other continuously in order to keep the same time. To this end, the EtherCAT line has a selected EtherCAT slave that provides the reference clock used to synchronize the slave clocks (S) of the other nodes and the controller. This reference clock node is usually the first node on the line with this function, as this ensures that all downstream nodes will receive the time entered by this first node into the protocol within the same protocol cycle. This, in turn, ensures that they will be able to correctly determine offset times that need to be taken into account in the protocol's transmission. When the EtherCAT system starts, the EtherCAT master takes the time from a master clock (e.g., a hardware-based BIOS clock in the PC) and sends it to the selected reference clock when EtherCAT starts.

The nodes use the synchronized clock as a reference when applying their input signals and setting output signals, and accordingly transmit the system image corresponding to a defined point in time as the protocol is transmitted.

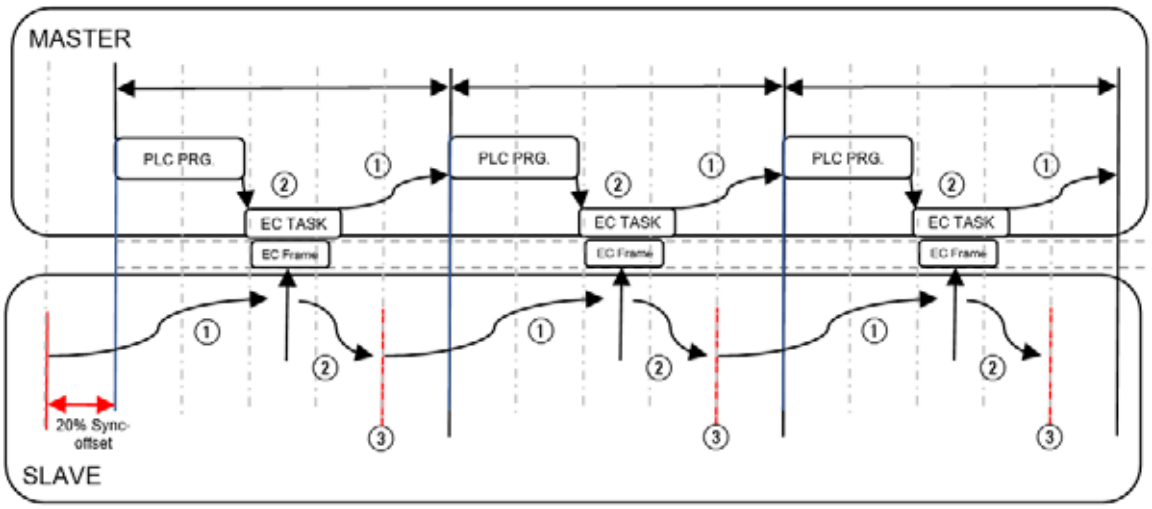


Figure 14: General distributed clocks diagram

① Input values
② Output values
③ Output values are set and input values are read

Please note that exact synchronization is always particularly important when spatially distributed processes require simultaneous actions.

7 Device description files for EtherCAT (XML/ESI)

EtherCAT uses an .xml file that contains EtherCAT slave information (ESI) in order to describe a device and its functions. This file can be imported into the EtherCAT control program.

This XML file lists all objects with the associated sub-indexes and the corresponding entries.

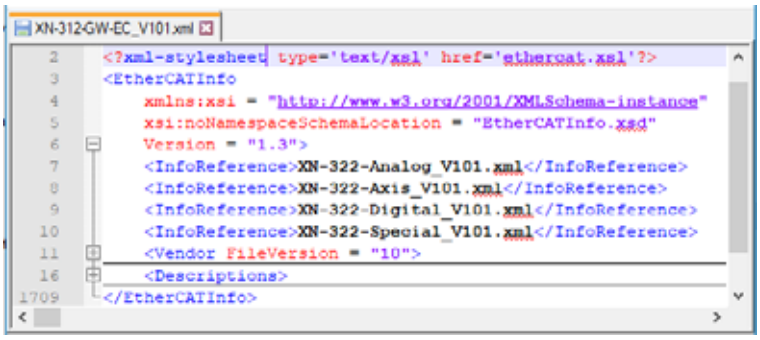


Figure 15: Header of a XML file for XN-312-GW-EC

In order to be able to run an XN300 I/O system through the gateway on an EtherCAT field bus, the gateway's and I/O modules' properties must be stored in a description file.

This file is standardized for the relevant field bus. and can be loaded into the PLC programming environment for the field bus master being used.

Check whether the gateway and the XML file support the XN300 modules you need.

→ It might be necessary to use a newer version of the XML file in order to be able to use new XN300 slice modules.

Make sure that you are working with the latest version of the device description files in your PLC programming environment. If applicable, check whether there are available updates for the files at our Download Center
→ Page 121.

Installing an XML file with XSOFT-CODESYS-3

By default, the required XML files will already come pre-installed on the programming system. If you nevertheless end up having to import XML files later on, follow the steps below:

- Select and download the latest version of the XML file.
- Save and unzip the "*.zip" file to a project folder of your choice.
- Click on the <Tools | Device Repository…> menu option in XSOFT-CODESYS-3.
- Select the <EtherCAT> field bus in the <Field buses> directory.
- Click on the "Install..." button.
- Select the XML files in the project folder and then click on the <Open> button.
- The EtherCAT gateway, with all the XN300 slice module parameters, will be installed. Close the dialog box by clicking on "Close."

After installing the new XML file, you can select the new device version from the device selection. If you update an existing project with a new XML version, you will need to update all the previously installed devices after installing the XML file.

8 Installation

DANGER OF ELECTRIC SHOCK!

All installation work must be carried out with the entire installation in a de-energized state.
Always follow the safety rules:
- De-energize and isolate the system.
- Verify isolation from the supply.
- Secure against restart.
- Short-circuit and ground.
- Cover adjacent live parts.

Gateways must only be installed and wired up by qualified electricians or other persons familiar with electrical engineering. The field bus address will be automatically assigned and cannot be set on the device. The data transfer rate is full-duplex 100 Mbit/s.

The gateway is installed in the following order:

- Join the gateway and all I/O slice modules to form a system block.
- Mount the system block on the DIN-rail.
- Connect the power supply.
- Connect the field bus.

8.1 Mounting

8.1.1 Installation prerequisites

Install the XN-312-GW-EC gateway in a control panel, a distribution board or an enclosure so that the power supply and terminal capacities cannot be touched accidentally during operation. Snap the device onto an IEC/EN 60715 DIN-rail.

The DIN-rail must establish a conductive connection to the control panel's back plate. The device must be installed in a horizontal position (device designation on top).

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.

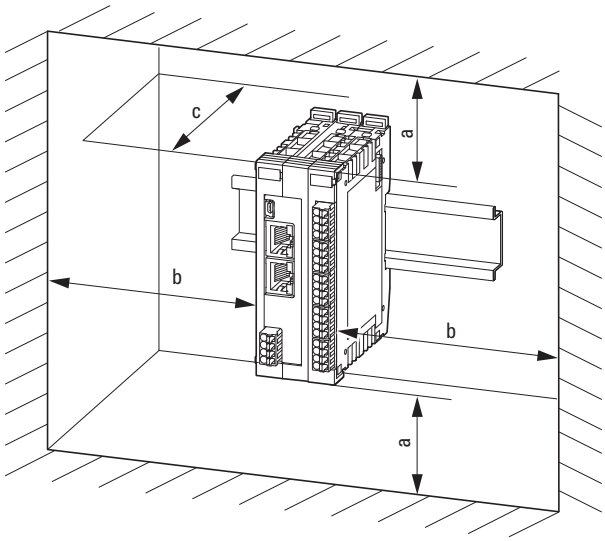


Figure 16: Horizontal installation only!

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

8.1.2 Mounting the system block on the DIN-rail

To mount the system on the DIN-rail, join the XN 300 slice modules and the gateway to form a system block and then snap the entire system block onto the DIN-rail.

To mount the system block, follow the steps below:

▶ The gateway must be the first element on the left in the system block.

- Disengage the side locking tabs on the XN300 slice modules by pulling on the front cover (blue). Make sure that all locking tabs (blue) are in the front so that they will engage the new slice (the front cover stay-put function is intended to make the process easier).

→ The gateway's front cover is non-detachable and cannot be removed.

- Attach an XN300 slice module from the right in such a way that the locking tabs engage the guide.

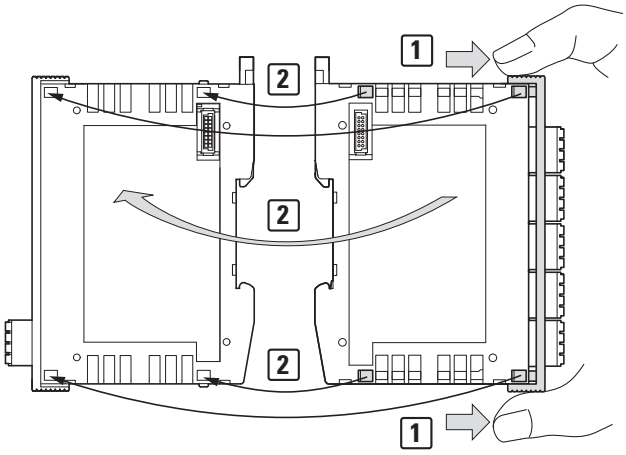


Figure 17: Joining the gateway and an XN300 slice module to form a system block

- Grab the front cover from the top and bottom and push it back towards the XN300 slice module so that the slice modules lock solidly into place.

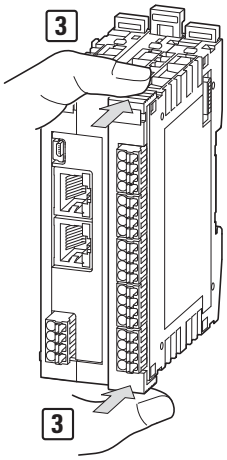


Figure 18: Locking the system block in place

- Repeat these steps until you have added all the XN300 modules you need to the system block.
- Pull the locking elements at the back of the gateway and the XN300 slice modules upwards. You can use a screwdriver to do this.

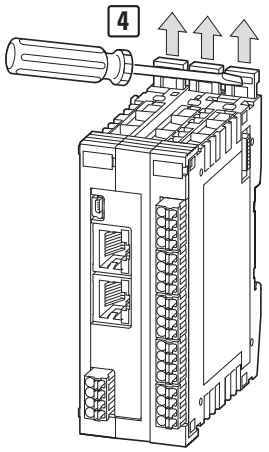


Figure 19: Securing the system block on the DIN-rail

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

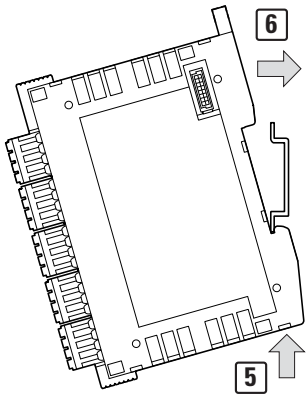


Figure 20: Placing the system block against the bottom edge of the DIN-rail

▶ Push the system block over the DIN-rail's top edge.
▶ Push the locking elements on the back of all XN300 slice modules downwards in order to secure the modules. You can use a screwdriver to do this.

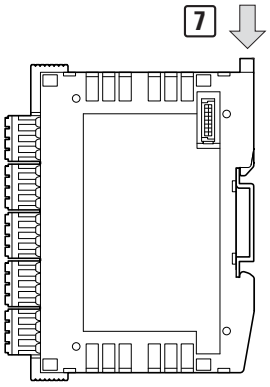


Figure 21: Locking the system block into place on the DIN-rail

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

8.2 Dismantling

To remove the gateway and the XN300 slice modules, follow the steps below:

▶ Slide the locking elements on the back of all XN300 slice modules upwards. You can use a screwdriver to do this.

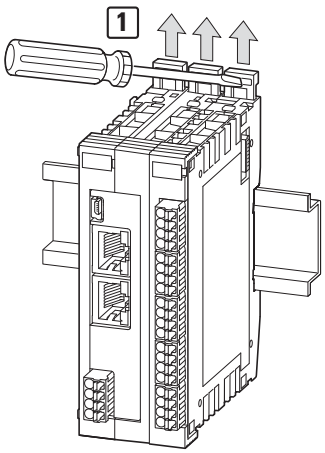


Figure 22: Disengaging the system block

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

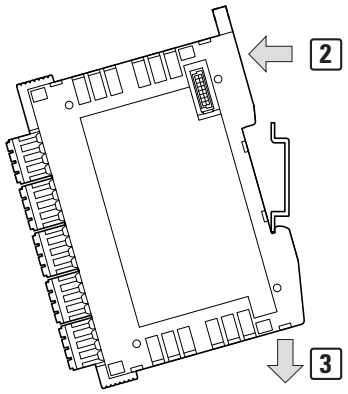


Figure 23: Placing the system block against the bottom edge of the DIN-rail

▶ Disengage the locking tabs between the slice modules by pulling on the front cover (blue). The front cover's stay-put function will indicate that the locking tabs have been disengaged.

→ The gateway's front cover is non-detachable and cannot be removed.

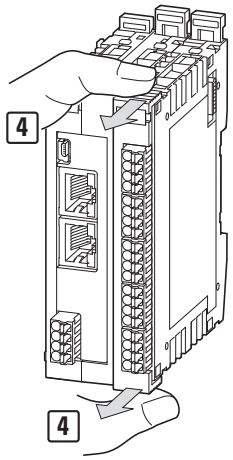


Figure 24: Disengaging the front cover

▶ Once the locking tabs have been disengaged, you can separate the slice modules and the gateway from each other.

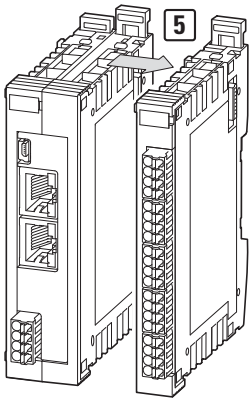


Figure 25: Separating the gateway and the XN300 slice modules from the system block

8.3 Connecting the power supply

DANGER

In safety-relevant applications the power supply providing power to the XN300 system must feature a PELV power supply unit (protective extra low voltage).

DANGER

The gateway has protection against polarity reversal for the 24 V DC POW supply. If, however, the gateway is connected to a grounded device, e.g., a computer, via the diagnostic interface, the gateway may be destroyed if the polarity of the power supply is reversed!

The gateway and the modules are powered using the X3 terminals. The gateway uses the voltage at X3 to produce the 5-V supply voltage for the modules on the system bus, with the maximum output current being 1.6 A.

Moreover, the gateway uses the voltage at X3 to produce the 24-V supply voltage for the following system bus modules, with the maximum output current being 1.6 A:

- Analog module
- Technology modules

In order for XN300 slice modules with terminals for an external power supply to be able to perform their specified functions, they must be supplied with an external 24 VDC voltage.

Take into account the total power consumption and the voltage drop on your system bus and, if necessary, include additional power supply modules in your design. You can use the XN300-Assist planning and commissioning program for the corresponding calculations.

The two V+ terminals are internally connected to each other, as are the two 0 V terminals. The internally connected terminals can be used to extend the 24 V and 0 V supply voltage connections. However, it is necessary to make sure that a total current of 6 A is not exceeded per terminal.

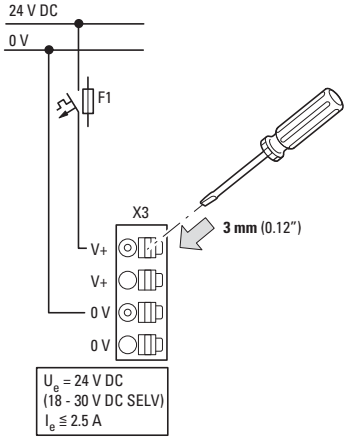


Figure 26: Connection of power supply

▶ Connect the 24 V DC voltage to the connection terminals X3 on the front side of the gateway.

Miniature circuit-breaker F1 for POW

- Cable protection in accordance with DIN VDE 0641 Part 11, IEC/ EN 60898:
  - Miniature circuit-breaker 24 V DC rated operational current 3 A; trip type C or
  - Fuse 3 A, utilization category gL/gG
- Cable protection for cable AWG 24 in accordance with UL 508 and CSA-22.2 no. 14:
  - Miniature circuit-breaker 24 V DC rated operational current 3 A; trip type C or
  - Fuse 3 A

Connection example

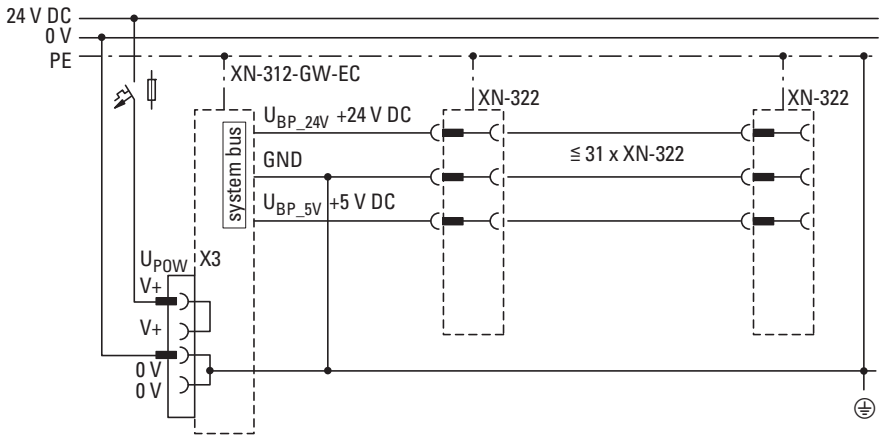


Figure 27: Connecting example for XN-312-GW-EC gateway in XN300 system

8.4 Connect field bus

EtherCAT

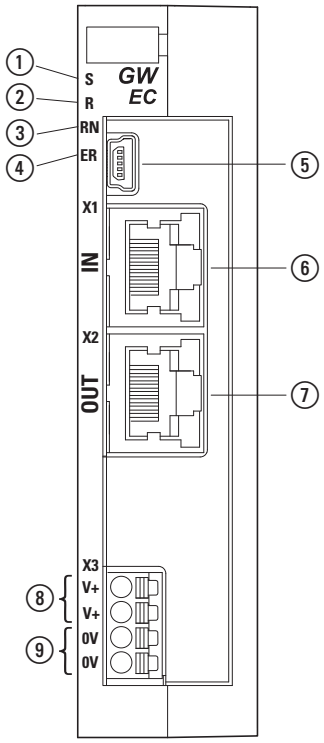


Figure 28: Connecting EtherCAT

To connect the EtherCAT field bus, follow the steps below:

- Plug the RJ45 connector into port X1.
- You can use the X2 field bus interface on the gateway to connect additional field bus modules or set up communication redundancy.

→ Secure the field bus cable in close proximity to the device in order to reduce mechanical strain on the connection.

8.4.1 Maximum cable length

The maximum cable length for an EtherCAT cable with full-duplex 100 Mbit/s is 100 m.

It may be necessary to use repeaters when using cable lengths of 100 m or more.

8.5 Connecting the diagnostics interface

The gateway features a Mini-USB port on the front. By using a programming cable, you can connect it to a computer and do the following with the corresponding software:

- Use the XN300-Assist planning and commissioning software to run real-time diagnostics on the system bus
- Updating the operating system with XN300-Assist.

You can use any of the following programming cables:

USB 2.0 cable: Mini-B connector <-> Type A connector

8.6 Wiring in accordance with EMC requirements

Undesired faults can occur due to electromagnetic interference. This can be minimized beforehand by the implementation of suitable EMC measures. These include:

- EMC-conformant system configuration,
- EMC compliant cable routing,
- Measures designed to reduce potential differences
- the correct installation of the field bus system (cable, connection of the bus connectors, etc.),
- Using shielding

for DIN-rail

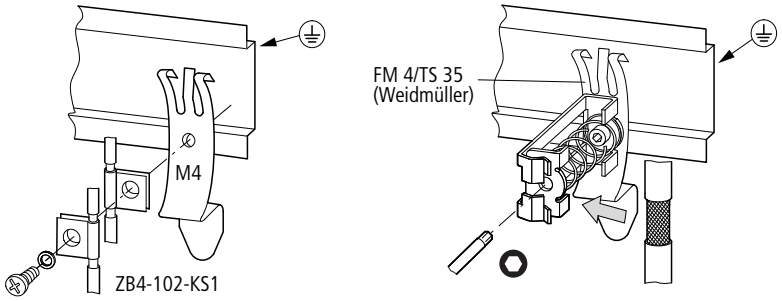


Figure 29: Field bus shielded by using a shield

The gateway features a functional earth contact point on the back.

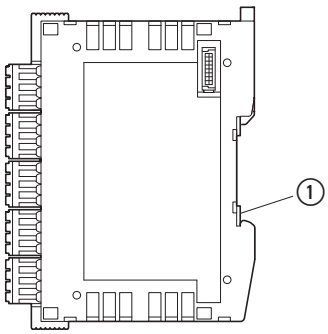


Figure 30: XN-312-GW-EC side view

① Functional earth

9 Commissioning

Before switching on, check whether the power supply for the gateway is connected correctly. The configuration and installation of the system bus must also have been carried out correctly with all modules connected.

DANGER

If you have already integrated devices and equipment into a system, block off any parts of the system that are connected to the working area in order to prevent access. This will ensure that no one is injured if the system behaves unexpectedly, e.g., if motors start up unexpectedly.

Startup takes place in several steps:

1. Putting the system bus into operation
2. Load and start the PLC program.

DANGER

Switch off the power supply whenever replacing XN300 slice modules.
The XN300 slice modules may be ruined otherwise!

A configuration check will run automatically when the system is turned on. For this purpose, the EtherCAT gateway will get the project configuration from the EtherCAT master and compare it to the current actual configuration on the system bus. If the actual configuration matches the project configuration or at least contains the XN300 slice modules in the project configuration, the EtherCAT master will switch to the pre-operational state.

In order for the system to start running, the actual configuration (i.e., the configuration that is physically present) must match the planned configuration in the controller.

9.1 Establishing EtherCAT field bus communications

For the EtherCAT master, go to the "General" tab and select the Ethernet port that should be used to connect to the EtherCAT gateway.

▶ To do so, click on the "Browse..." button.

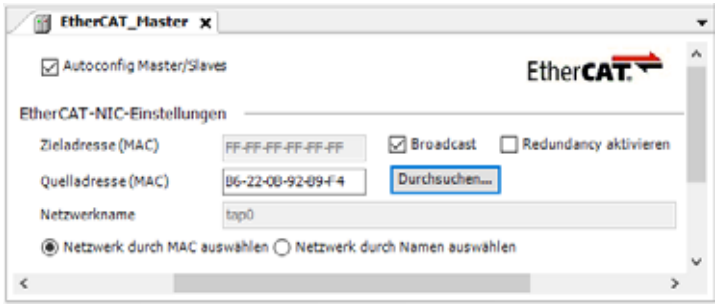


Figure 31: EtherCAT Master "General" tab

The "Select Network Adapter" dialog box will appear.

▶ In the "Select network adapter" dialog box, select the controller Ethernet port that should be used to connect to the EtherCAT gateway ("ETH2," for example).

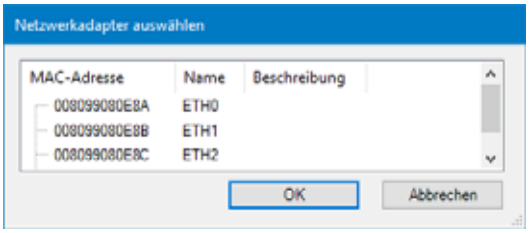


Figure 32: "Select network adapter" dialog box

The selected port will be automatically entered into the "Network name" field.

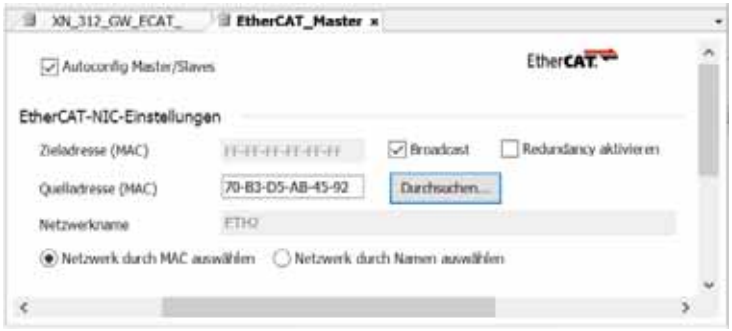


Figure 33: Configuring the Ethernet connection under the "General" tab for the EtherCAT master

The green RN LED on the gateway will show a green continuous light if the gateway and the system bus are successfully communicating. This requirement must be met in order for process data to be transferred between the gateway and the PLC via the field bus.

→ If the field bus address on the gateway changes, the change will not take effect for the communication on the field bus until after the gateway is turned off and then back on.

9.2 Status indicators when field bus communications are successfully established

LED	Status	Description
S	Continuous light green	System bus running synchronously
R	–	No power on system bus
RN	Continuous light green	Device ready for communication
EC	OFF	No errors or faults

In operation

If an XN300 slice module is removed from, added to, or replaced on the system bus, you will need to switch off the power supply.

If a configuration change is made during ongoing operation, this will always result in the gateway being reset.

10 XN300-Assist

The XN300-Assist planning, ordering and commissioning program can be used to perform the following functions, among others:

- Selecting modules, as well as a gateway
- Reading device parameters for the gateway and the system bus' modules
- Status indication of the inputs/outputs
- Wiring test
- Importing actual configurations
- XN-312-GW-EC firmware update
- Display of cyclical and acyclical diagnostics messages

The Online Help function provides a detailed description of how to use XN300-Assist. In order to open it, click on the "?" icon on the XN300-Assist menu bar or press the <F1> key.

10.1 Firmware update with XN300-Assist

Eaton may make new operating systems available in order to provide new functionalities; Download Center → Page 121.

For more information on how to update the operating system for the XN-312-GW-EC gateway with the help of XN300-Assist, please refer to the online help for XN300-Assist. To open this online help, follow the steps below:

▶ Connect the PC to the XN-312-GW-EC diagnostic interface.
▶ Start XN300-Assist.
▶ Open the online help for XN300-Assist by clicking on the "?/Help" menu option.
▶ Open the "Search" tab and enter "Update operating system" into the search box.

The corresponding information will guide you on how to carry out a firmware update.

10 XN300-Assist

10.1 Firmware update with XN300-Assist

11 Configuring an EtherCAT system with XSOFT-CODESYS

The following sections provide examples showing how to connect a gateway, including the corresponding I/O slice modules, to an EtherCAN master.

Before commissioning an EATON PLC (e.g. XC303, XV300) with XSOFT-CODESYS-3 programming software, the PLC needs to be connected to a programming computer. To do this, the EATON PLC first needs to be connected to the XN-312-GW-EC EtherCAT gateway using an Ethernet cable.

11.1 General instructions for working with XSOFT-CODESYS-3

Please note that the table in XSOFT-CODESYS-3 will often hide whole columns.
If you are missing any data, make sure that all the columns in the table are being shown!

The following sample table includes columns that are not being shown even though they exist.

To show the columns, you would need to move the cursor over the column header border between two columns and then double-click or expand the next hidden column by dragging the border and dropping it.

Please note that only one hidden column will be shown after this. If there are multiple hidden columns, you will need to repeat this step multiple times.

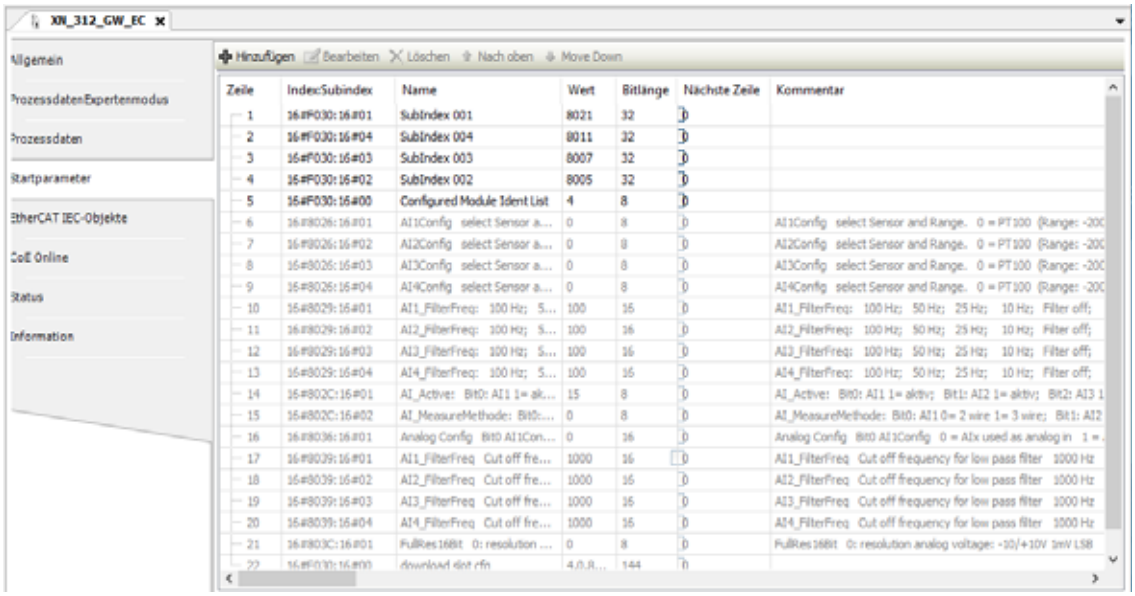


Figure 34: Table in XSOFT-CODESYS-3 with two hidden columns

- In the table above, you would move the cursor over the header border between the "Bit length" and "Next line" columns and click and drag the border to the right.
- You would then move the cursor between "Bit length" and "Jump to line in case of error" and drag the border between the two to the right.
- Finally, you would move the cursor between "Bit length" and "Abort in case of error" and drag the border between the two to the right.

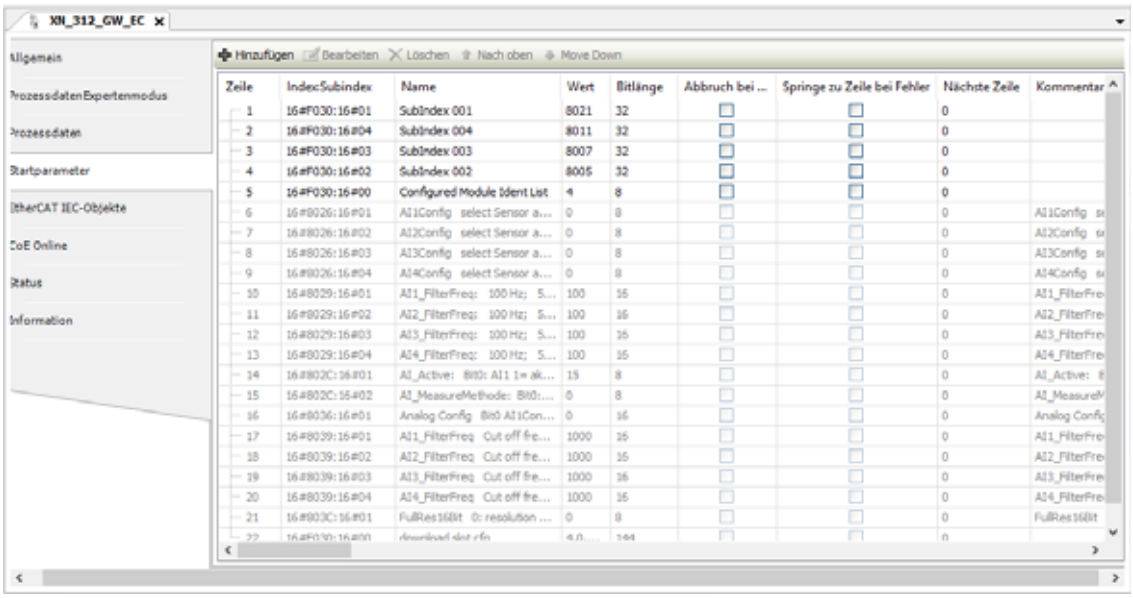


Figure 35: Table in in XSOFT-CODESYS-3 showing all the columns

11.2 Configuring the system with XSOFT-CODESYS-3

To set up a configuration, you will need XSOFT-CODESYS-3 Version 3.5.15 or higher. The information below uses standard mode. Moreover, the example below uses the following devices:

- XC-303 EtherCAT master
- EtherCAT device XN-312-GW-EC
- XN300 slice modules:
  - XN-322-8DI-PD
  - XN-322-16DO-P05
  - XN-322-4AI-PTNI
  - XN-322-8AIO-U2

11.2.1 Starting XSOFT-CODESYS-3 and creating a new project

The following description refers to a project created in "standard" mode.

Creating Project

▶ After starting XSOFT-CODESYS-3, open a new project by clicking on < File | New Project...>.
▶ The "Standard project" dialog box will appear. Select the "Standard project" option in the "Templates" and click on OK.
▶ The "Standard project" dialog box will appear. Use the "Device" drop-down menu to select the PLC model you are using in your application, e.g., "XC303".
▶ Now use the "PLC_PRG in" drop-down menu to select the programming language you want, e.g., "Structured Text (ST)".

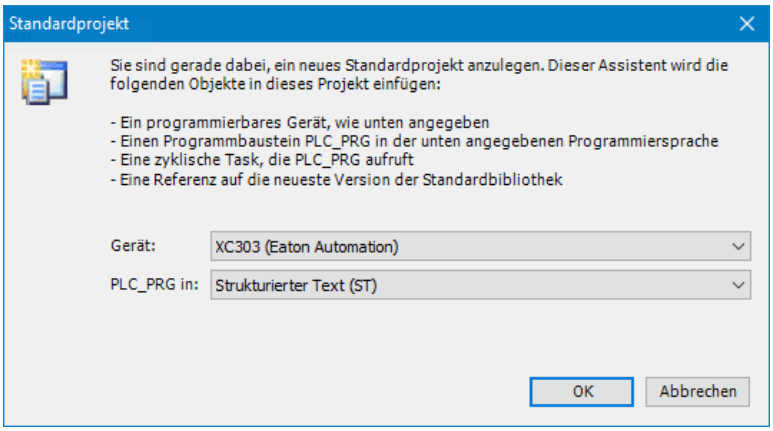


Figure 36: Configuring a standard project

▶ Confirm the PLC information being displayed by clicking on "OK."

The device will be added to the project.

11.2.2 Configuring the EtherCAT master

Follow the steps below to enable communication via EtherCAT:

- ▶ In the "Devices" pane on the left, select the device you added previously, e.g., "Device (XC303)".
- ▶ Right-click on it to open the context menu and select the <Add device...> option.

The "Add device" dialog box will appear. You can leave the dialog box open.

If the dialog box is not showing a device, make sure that <All manufacturers> is being shown in the "Manufacturer" drop-down menu.

- ▶ Click on the <EtherCAT | EtherCAT master | Add device> option.

The field bus will be added to the configuration structure.

All the required libraries will be automatically integrated with the EtherCAT master; for more information, please refer to → Section "11.2.6 Adding EtherCAT communication libraries", page 77.

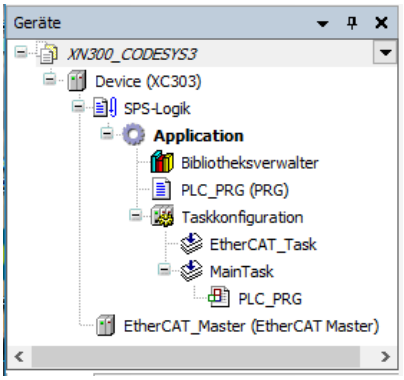


Figure 37: Configuration structure with bus system selected

- ▶ Click on the "EtherCAT_Master (EtherCAT Master)" node and select the <Add Device...> option in the context menu.

The "Add device" dialog box will appear if it is not open already.

- ▶ Select <Field buses | EtherCAT | Slave | Eaton Industries GmbH | Gateways | XN-312-GW-EC | Add Device>.

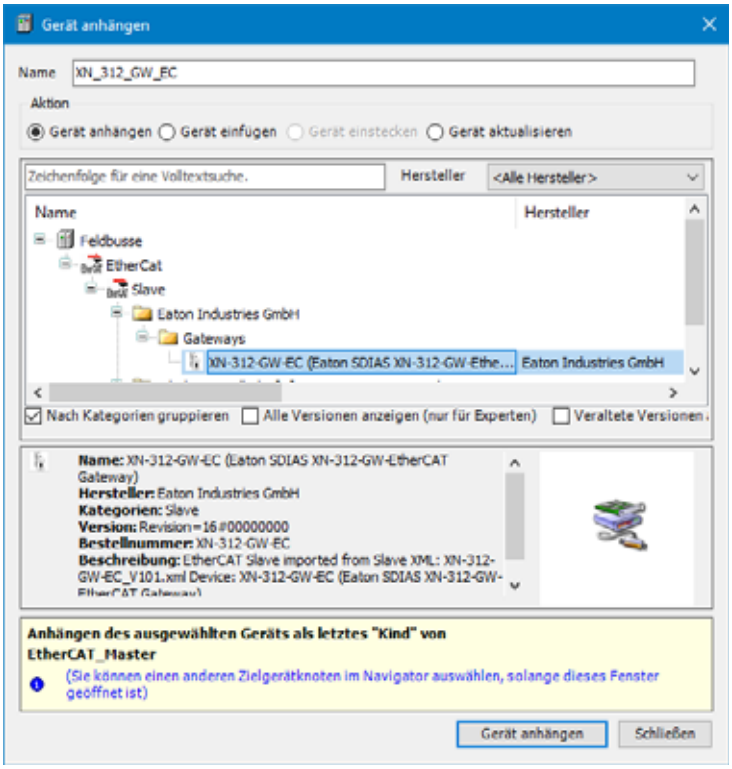


Figure 38: "Add Device" window

The device will be configured as an EtherCAT master.

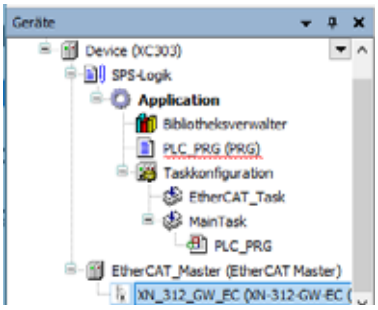


Figure 39: "Devices" dialog box with EtherCAT master and EtherCAT gateway in configuration tree

▶ Double-click on EtherCAT_Master(EtherCAT Master).

Defining an Ethernet port

Define the port through with the EtherCAT master should communicate with the EtherCAT device.

▶ To do so, go to the "EtherCAT Master/General" tab and click on the "Browse..." button.

The "Select Network Adapter" dialog box will appear.

▶ Select the EtherCAT master Ethernet port you want ("ETH1," for example).
▶ Confirm your selection by clicking on "OK".

▶ Open the "EtherCAT Master/General" tab and check the settings there.

– "Automatic slave restart" should be enabled under the options.
– Check whether you want to enable redundancy and/or use distributed clocks; please refer to the following as well: → Section "13.5.2 Redundancy", page 118, → Section "13.5.1 Enabling distributed clocks", page 118.

→ For more detailed information, please refer to the online help for CODESYS, which can be opened by pressing the <F1> function key.

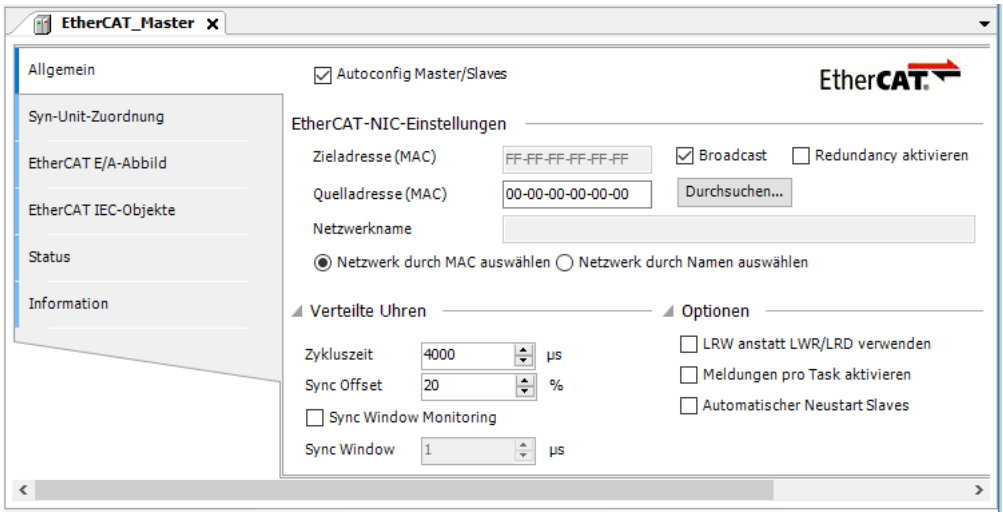


Figure 40: EtherCAT master parameters

Baud rate of the EtherCAT master

The baud rate for the EtherCAT master (XC303 in this case) and the EtherCAT device (XN-312-GW-EC in this case) is fixed at full-duplex 100 Mbit/s.

11.2.3 Configuring EtherCAT devices

▶ To add the XN-312-GW-EC EtherCAT device to your configuration, right-click on "EtherCAT Master (EtherCAT Master)."
▶ Click on the <Add device…> option.

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

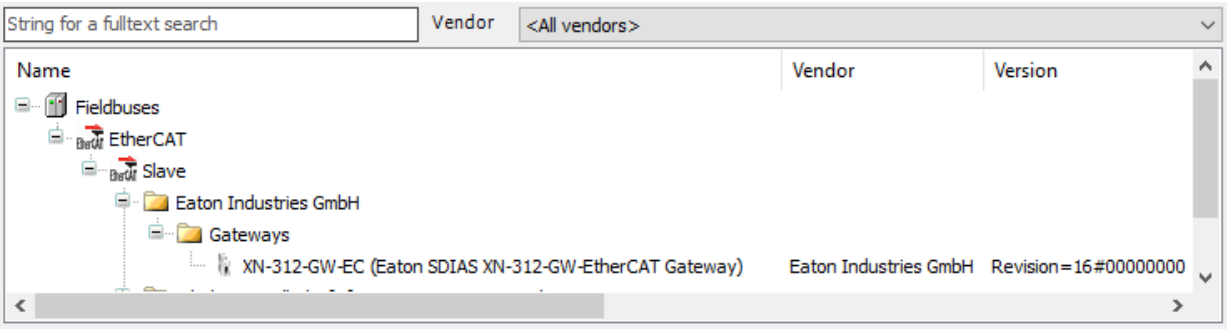


Figure 41: Adding an EtherCAT device

- ▶ Select the EtherCAT device you want (XN-312-GW-EC, for example).
- ▶ Click on the <Add device> button.

The selected EtherCAT device will be added to the configuration structure in the "Devices" pane.

- ▶ If the EtherCAT devices you want are not found in the list, update the XSOFT-CODESYS-3 version you are using or load the appropriate XML file; → Section " Installing an XML file with XSOFT-CODESYS-3", page 48.
  You can find updates and XML files by visiting the Download Center on our website → Page 121.

11.2.4 Configuring the XN312 gateway

Check the settings for the EtherCAT device's parameters.

- ▶ Double-click on the selected EtherCAT device in the "Devices" configuration tree ("XN_312_GW_EC (XN-312-GW-EC)," for example).
- ▶ Open the "XN-312-GW-EC/General" tab and check the settings there.
  - – Select the expert settings by enabling the "Enable expert settings" option.
  - – In the "Distributed clocks" section, enable the "Custom DE settings" option in the "Select DC" drop-down menu by enabling the checkbox underneath.
  - – In the "Sync0" section, enable the "Enable Sync 0" option. Select "x1" in the "Sync Unit Cycle" drop-down menu.

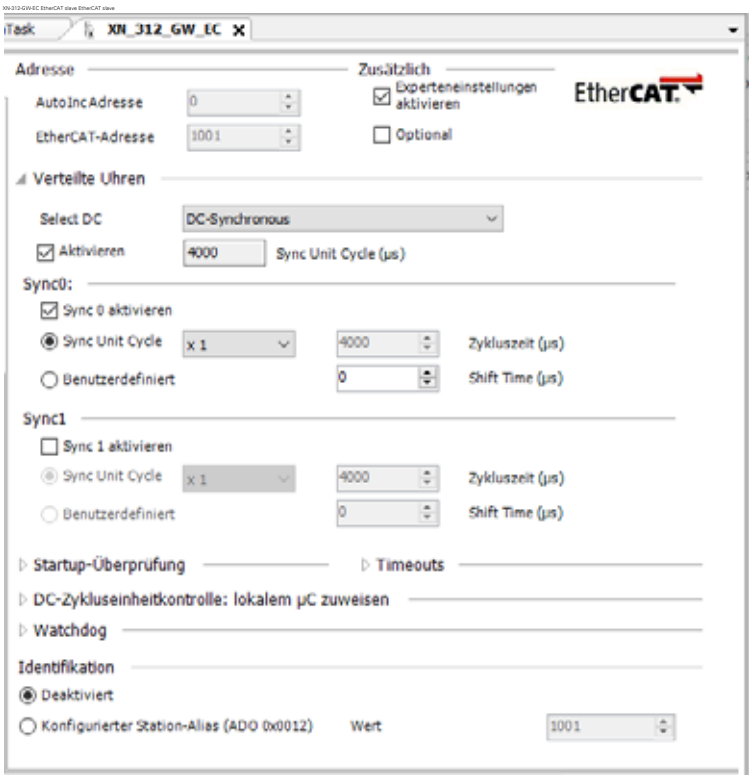


Figure 42: XN-312-GW-EC EtherCAT gateway settings

11.2.4.1 Configuring XN-322 slice modules

- ▶ To configure the system block, right-click on the EtherCAT device, e.g., "XN_312_GW_EC (XN-312-GW-EC...)" in the "Devices" configuration structure.
- ▶ Then click on the <Add device...> option.

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.

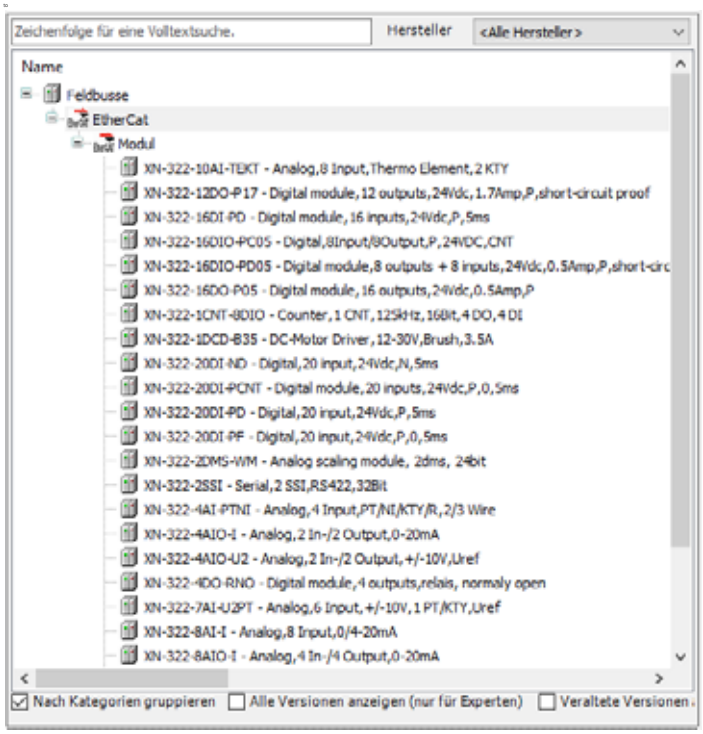


Figure 43: "Add Device" dialog box for selecting the XN300 slice modules for the EtherCAT device
(XN-312-GW-EC in this case)

- ▶ Select the I/O slice module you want, e.g., "7.XN-322-8DI-PD".
- ▶ Then click on the <Add device...> option.
- ▶ Repeat these steps until you have added all the I/O slice modules from the example: XN-322-8DI-PD, XN-322-16DO-P05, XN-322-4AI-PTNI, XN-322-8AIO-U2.

The I/O slice modules will be added to the "Devices" configuration tree.

→ You can change the order of the I/O slice modules in the configuration structure by dragging and dropping them accordingly.

A maximum of 32 I/O slice modules can be added to the EtherCAT device.

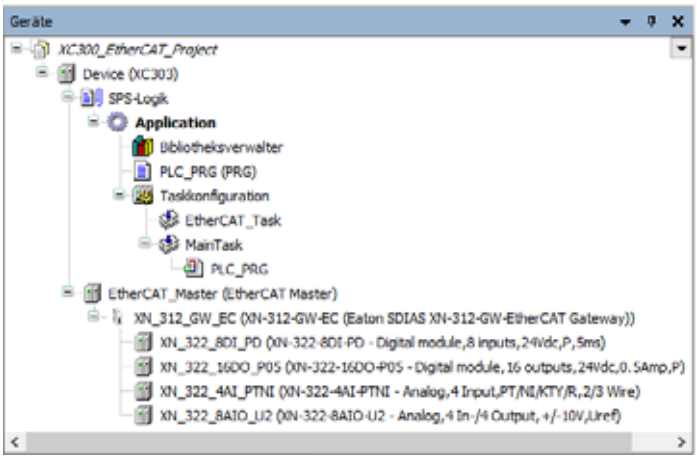


Figure 44: "Devices" configuration structure

11.2.4.2 Automatic mapping

The communication objects will be automatically mapped for every I/O slice module. For information on which data is provided in the objects corresponding to the various XN300 slice modules, please refer to the appendix
→ Section “13.4 XN300 slice module objects”, page 104.

In the relevant "Supported objects" section.

▶ Open the "Process data" tab and check to make sure that all of the I/O slice modules' process data is included in the automatic mapping.

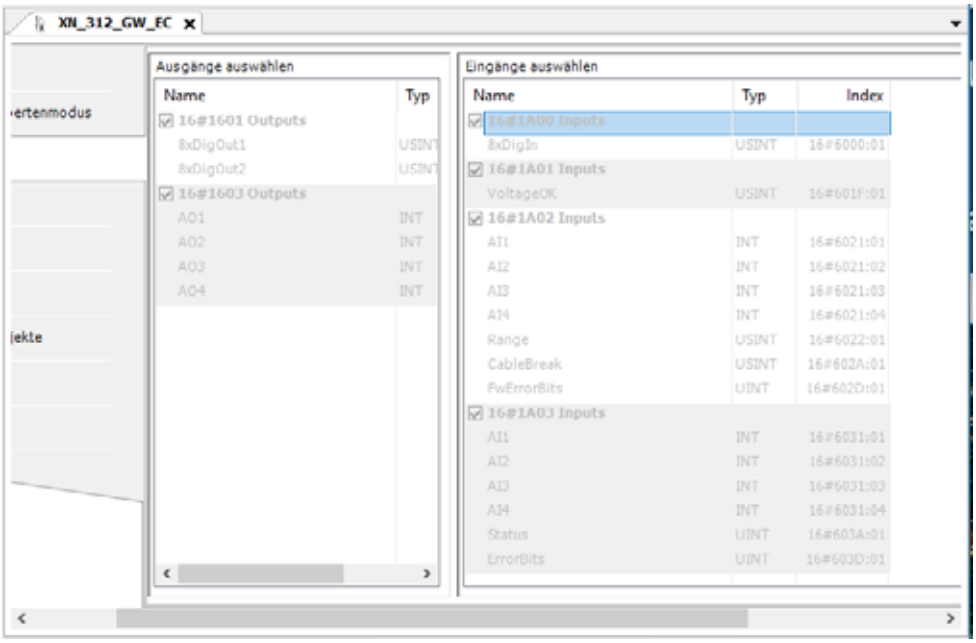


Figure 45: Mapping table in "PDO Mapping" tab

11.2.5 Configuring device initialization parameters

The "Service data object" tab shows the SDO objects that are written to the gateway when the system starts. In the case of certain XN300 slice modules, additional parameter values can be initialized. The following example shows how to enable an analog input in I/O slice module XN_322_10AI_TEKT.

Follow the steps below to modify the parameter values used for initialization:

- ▶ Double-click on the EtherCAT device, e.g., "XN_312_GW_EC (XN-312-GW-EC)."
- ▶ Open the "SDOs (Service Data Object)" tab.
- ▶ Click on the "New..." button.

A list of all available SDO objects will be shown.

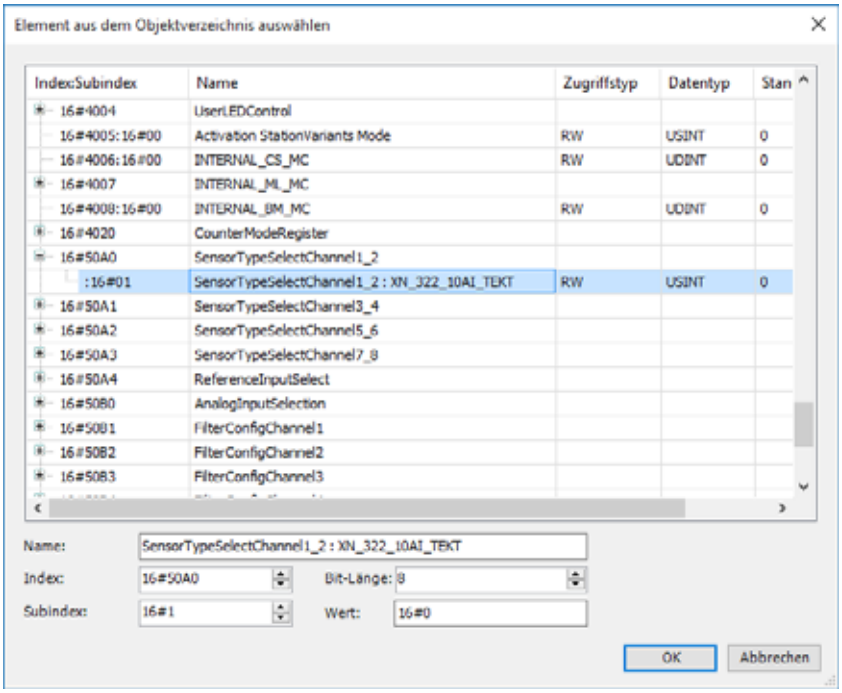


Figure 46: List of all available SDO objects

- ▶ Select the additional SDO object that should be transmitted when the device is initialized, e.g., "SensorTypeSelectChannel1_2 : XN_322_10AI_TEKT."
- ▶ Enter the default value you want into the "Value" field, e.g., "1".

→ The "SDOs (Service Data Object)" tab will show default device initialization values with a gray font and modified device initialization values with a black font.

11.2.6 Adding EtherCAT communication libraries

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

→ For more detailed information, please consult the documentation for the PLC you are using.

12 Sample project

The following sections list the various product-specific and manufacturer-specific object dictionary entries. There is a detailed description of the objects for each XN300 slice module, as well as the representation of the objects in the XN-312-GW-EC gateway CoE.

This sample project uses four XN300 slice modules running on the EtherCAT gateway. The goal is to show which XN300 slice module data is transferred, how, and with which data type. The following XN300 slice modules are used in the sample project:

- XN-322-8DI-PD
- XN-322-16DO-P05
- XN-322-4AI-PTNI
- XN-322-8AIO-U2

12.1 Establishing communications with the PLC

To establish communications with the PLC from the PC, follow the steps below:

▶ Use an Ethernet cable to connect your PC's Ethernet port to one of the Ethernet ports on the PLC (ETH0 on XC300, for example). The default setting for the IP address for ETH0 on XC300 is 192.168.119.248.
▶ Make sure that both Ethernet ports (the one on the PC and the one on the PLC) are in the same number range. With a subnet mask of 255.255.255.0 and the default setting for the PLC Ethernet address, the PC's IP address should be 192.168.119.... .
▶ Open the XSOFT-CODESYS-3 project. You may want to use the project you already created in section → Section “11.2.1 Starting XSOFT-CODESYS-3 and creating a new project”, page 70.
▶ Double-click on the "XC303" PLC in the "Devices" dialog box.
▶ Open the "Device/Communication tab" and click on the "Scan network..." button.

The "Select Device" dialog box will appear and the PLC will be shown.
If the dialog box stays blank, this means that there is no Ethernet connection.

▶ Select the PLC and confirm with "OK."

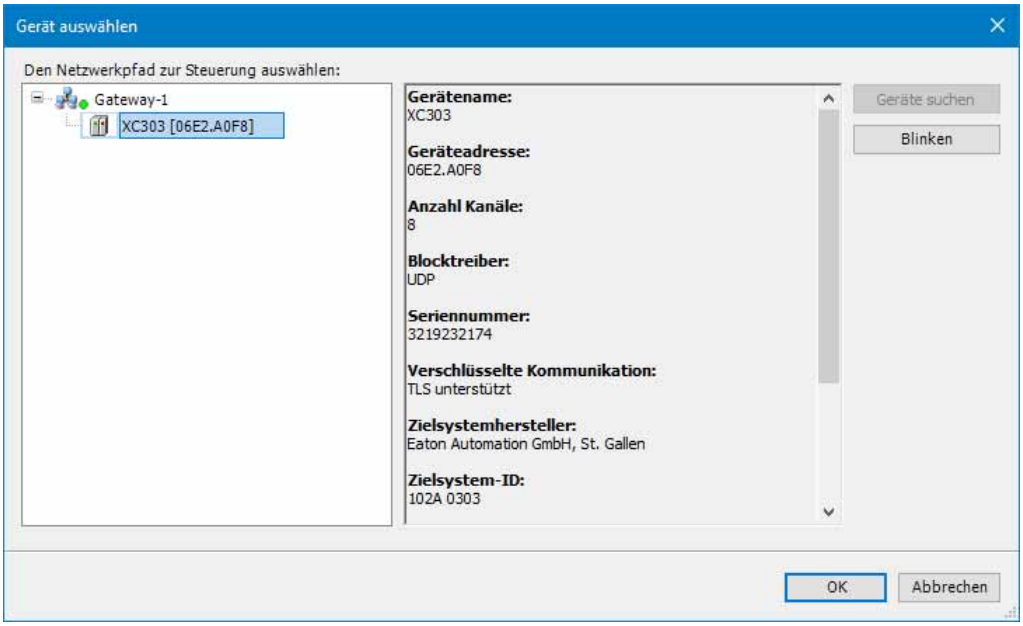


Figure 47: "Select Device" dialog box

Communications between the PC and the PLC will be established, and this will be shown with "... (active)" in the drop-down menu underneath the PLC.

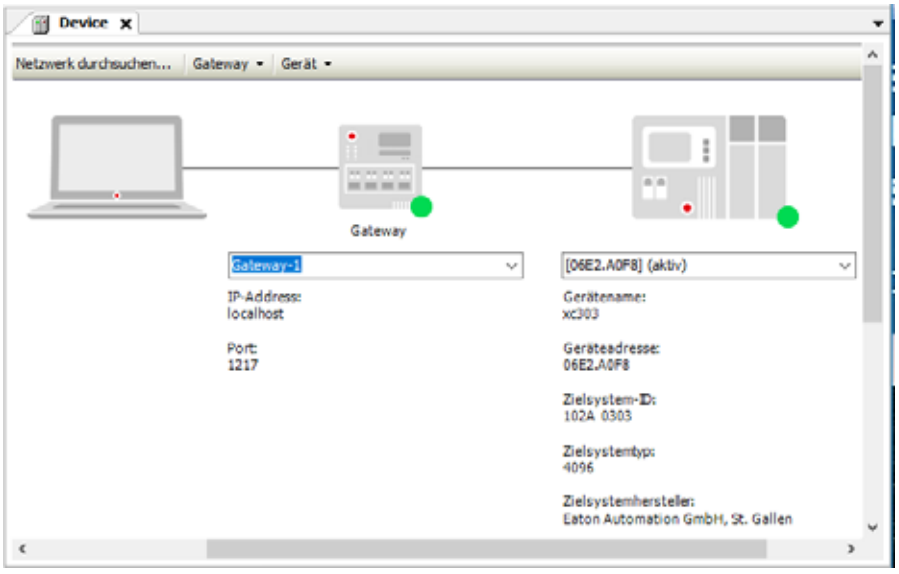


Figure 48: Device(XC303)/Communication tab

12.2 Updating the operating system for the XC303 PLC

If the following error message appears, the PLC firmware needs to be updated:

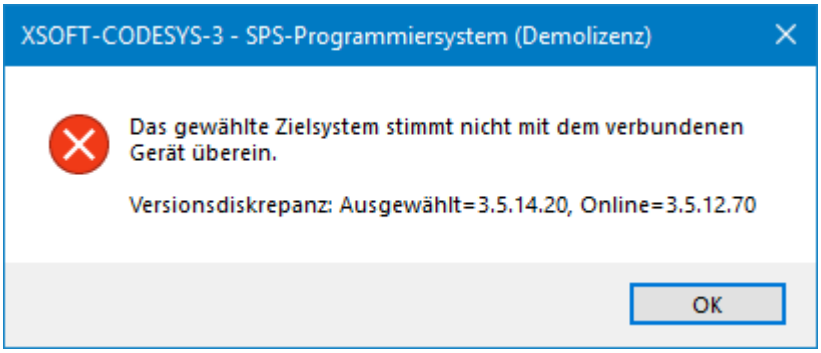


Figure 49: Fault message

To update the PLC's operating system, follow the steps below:

- Select the XC303 PLC in the configuration tree.
- Double-click on XC303.
- Go to the "Device" tab and select "Firmware."
- Click on the "Start..." button in the "Update operating system" section.
- The XC303 Operating System Transfer dialog box will appear.

The Firmware Setup Wizard will appear.

- As the installation type, select "Installation through network (FTP, SFTP, SCP)" and confirm by clicking on "Next."
- Select device type "XC-303" and confirm by clicking on "Next."
- Select the IP address for the PLC's Ethernet port (192.168.119.248, for example).
- Select the components that should be installed and confirm by clicking on "Next."
- Click on "Install."
- Exit the wizard by clicking on "Done."

12.3 Configuring parameters

In order to configure the parameters for each XN300 slice module, the connection to the PLC must be offline. Parameters generally need to be configured under the "Start parameters" tab for the corresponding XN300 slice module. Please note that not all XN300 slice modules have start parameters.

In our sample project, XN-322-8DI-PD and XN-322-16DO-P05 do not have any start parameters.

Use X1 to connect XN-322-4AI-PTNI to a temperature sensor as specified in the MN050002 "XN300 slice modules" manual. Select a 3-wire configuration. We are assuming that the temperature sensor is a Pt100 resistance thermometer with a temperature range of -200 to +850°C.

To configure the start parameters for XN-322-4AI-PTNI and XN-322-8AIO-U2, follow the steps below:

XN-322-4AI-PTNI

- Make sure that you are logged out.
- Open the XN_322_4AI_PTNI/Start parameters tab by double-clicking in the configuration tree.
- To select the temperature sensor type for channel 1, assign AI1Config a value of 1. 1 corresponds to a Pt100 with a range of -200 to +850°C (8026#1).
- To select a 3-wire configuration for channel 1, assign the AI_MeasureMethode start parameter a value of 1 (802C#2).
- To enable channel 1, assign AI_Active Bit 0 a value of 1. (802C#1)
- To select a filter frequency of 1 Hz, assign AI1_Filterfreq a value of 1 (8029#1).

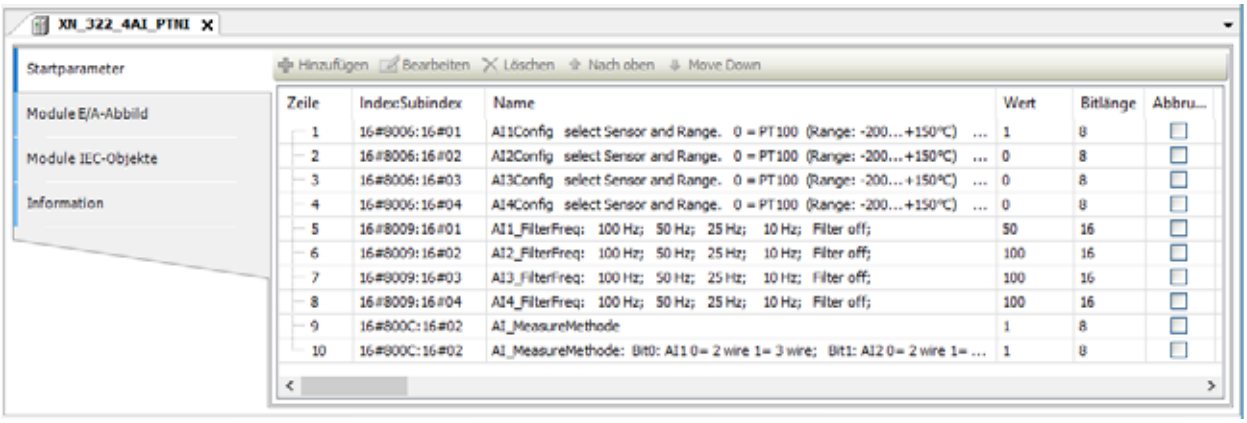


Figure 50: XN_322_4AI_PTNI/Start parameters tab

XN-322-8AIO-U2

- Make sure that you are logged out.
- Open the XN_322_8AIO_U2/Start parameters tab by double-clicking in the configuration tree.
- To configure channel 1 as a potentiometer measurement, assign the AI1Config start parameter a value of 1. 1 corresponds to a measurement in which analog input AI1 is connected to GND.
- To select a filter frequency of 1 Hz for channel 1, assign AI1_Filterfreq a decimal value of 1.
- To select a resolution within the measuring range / reading representation of -10 to +10V with 0.3 mV/LSB, assign the FullRes16Bit start parameter a value of 1.

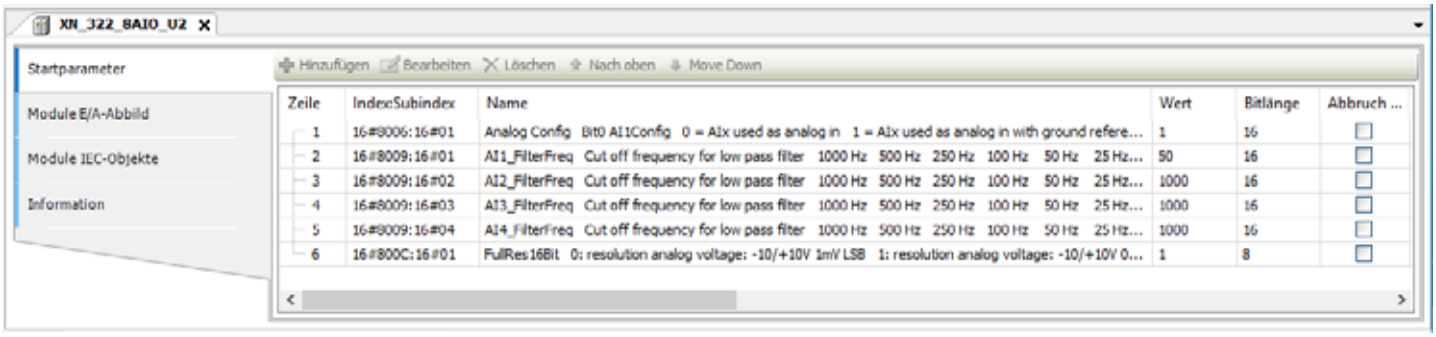


Figure 51: XN_322_8AIO_U2/Startparameter tab

12.4 Monitor IEC objects

IEC objects are used primarily by experts.

This tab in the generic device editor lists "objects" that make it possible to access the device from the IEC application. In online mode, it serves as a monitoring view for the settings for the Ethercat gateway itself.

In online mode, you can use the IEC object table as a monitoring view. It will also show the current value, the address, and the comments for the function block tags on the channel, as well as provide the option to write and force values.

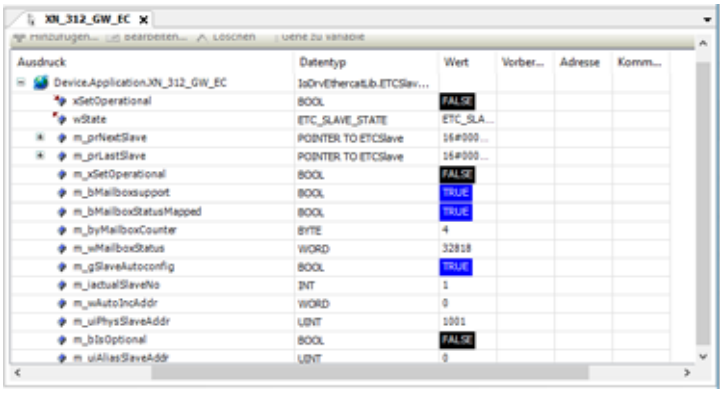


Figure 52: XN_312_GW_EC/EtherCAT IEC objects online tab

12.5 Going online and loading the program onto the PLC

Please note that data can only be shown and monitored in XSOFT-CODESYS-3 if used in a program that is found on the PLC.

In order to be able to load a program onto the PLC and display and monitor the states of a running program, the programming system must go online with the PLC.

Before that, we will be creating a small sample program for our sample project. The program will set the outputs on the XN-322-16DO-P05 and read the inputs on the XN-322-4AI-PTNI.

The following screenshot shows one possible small sample program:

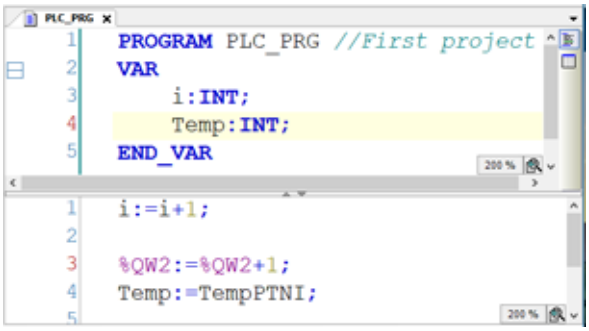
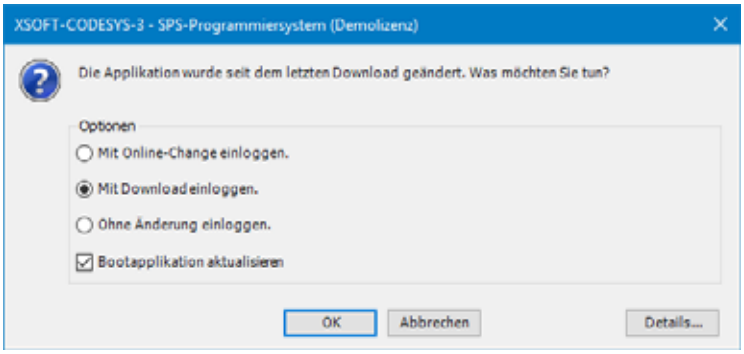


Figure 53: Sample program in XSOFT-CODESYS-3

Follow the steps below with the loaded project and the sample program above (or another one of your choice):

- Click on the "Create" / "Compile" menu option.
  A binary file that can be run by the PLC will be generated using the source code.
- Click on the "Online" / "Log in" menu option.
  The following dialog appears.:



- Select the "Log in with download" option.
  The "Update boot application" option will be enabled automatically. Leave it enabled.
- Confirm by clicking on OK.
- Start the program by pressing function key [F5] or by clicking on the icon in the toolbar.

The program will be loaded onto the PLC's RAM and will then be run. In addition, a boot application will be generated so that this will be the program that is run every time the PLC boots.

green icons in the configuration tree indicate that communication is starting and the program is running.

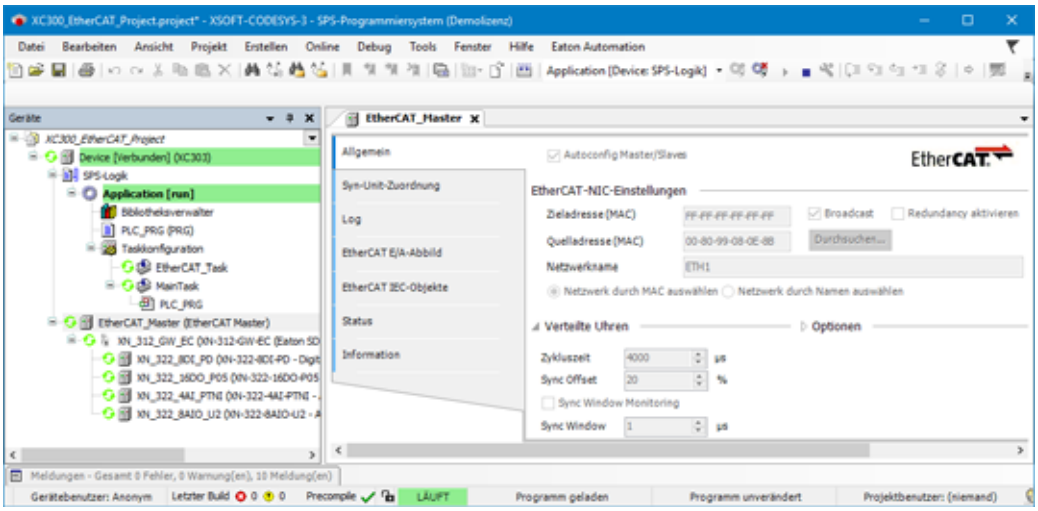


Figure 54: XSOFT-CODESYS-3 in ONLINE mode

12.6 Diagnostics messages

If the planned configuration on the Ethercat gateway does not match the actual configuration (i.e., the physically present configuration), the EtherCAT master will not start running. If communication cannot be successfully established, you can use the following diagnostic options:

EtherCAT_Master/ diagnostics

The EtherCAT_Master/General tab provides a general description in the "Diagnostic message" field.

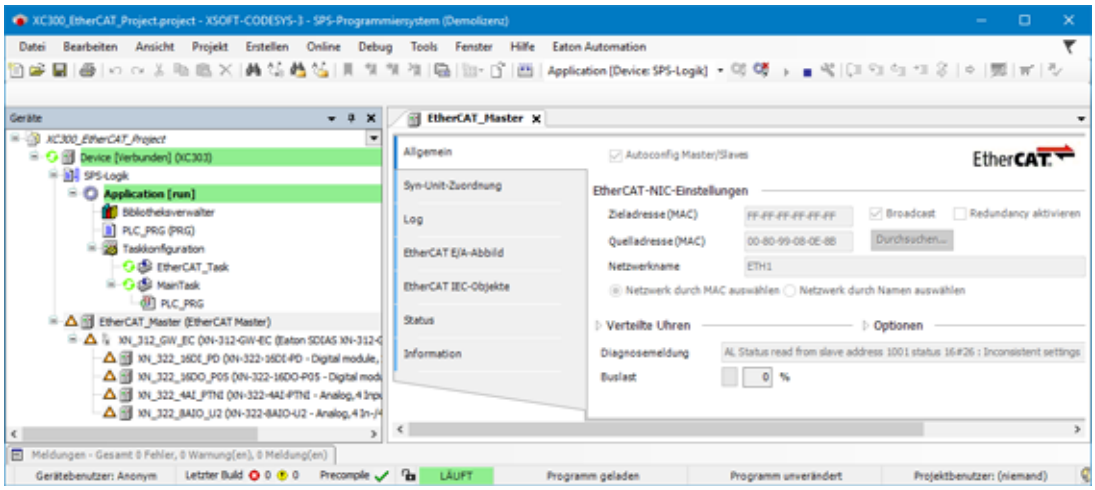


Figure 55: Diagnostic message in EtherCAT_Master/General tab

CoE diagnostics for Ethercat gateway

ErrorCode

For more information, please refer to → Section “4.3.6 Error info Object 0xF111”, page 38.

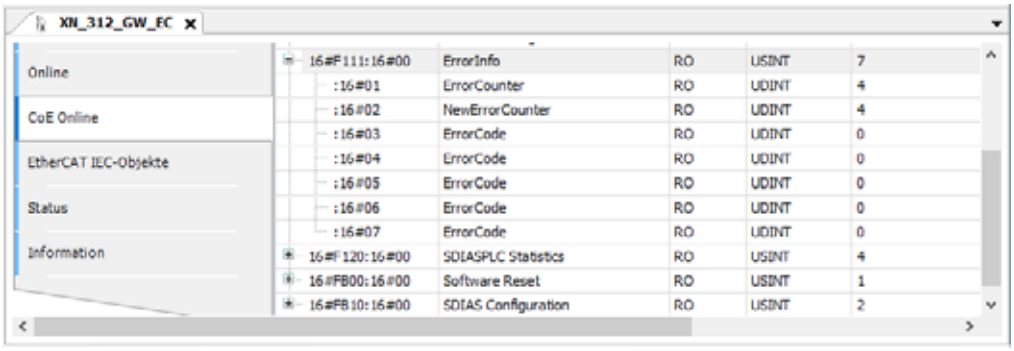


Figure 56: ErrorCode diagnostic message in XN_312_GW_EC/CoE tab

Error settings

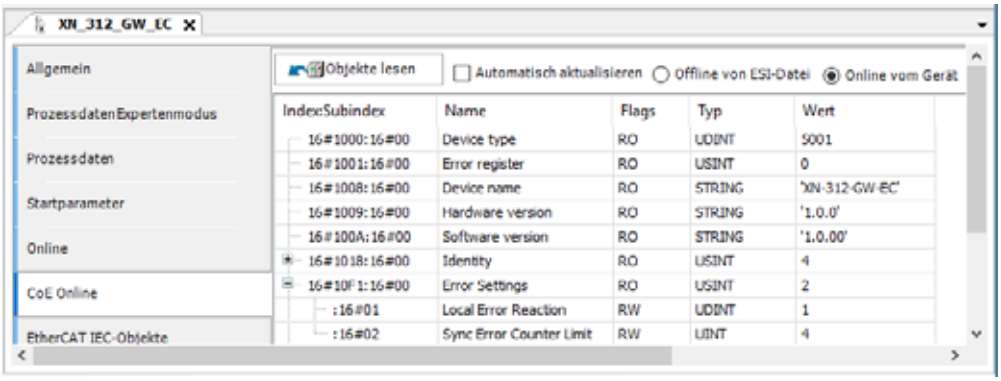


Figure 57: Diagnostic message in XN_312_GW_EC/CoE/Online from device tab

12.7 What will be transferred

Object range

0x6000 – cyclical output process data

0x7000 – cyclical input process data

0x8000 – parameters, i.e., SDOs

0x9000 – module information as SDOs, such as ModuleID, serial number, HW version, firmware version, FPGA version

12.7.1 Implementing the start parameter PDOs and SDOs

The SDOs for the individual XN300 slice modules can be viewed and edited in the "Start parameters" tab for the corresponding XN300 slice module.

The objects are found in address range 0x8000. The mapping of these objects to the objects that must be edited for XN-312-GW-EC gateway in the "Start parameters" tab is as follows:

The zeroes in the middle of the object address are replaced with the ModuleID for the relevant module. The ModuleID starts with 0 and increases in line with the node order.

Example: In our sample project, the object address for the sensor selection at analog input AI1 under the "XN_322_4AI_PTNI/Start parameters" tab is 0x8006:16#01hex. This object is represented under a different address in the XN-312-GW-EC gateway.

In tab "XN_312_GW_EC/Start parameters“ = 0x8026:16#01_{hex}, since it is the third node in the XN300 slice module block and has ModuleID 2.

XN300 slice module	ModuleID	Object address in slice module/ Start parameters tab	Object address in XN300 XN_312_GW_EC/ Start parameters tab	Inputs in XN300 XN_312_GW_EC/ Process data tab	Outputs in XN300 XN_312_GW_EC/ Process data tab
XN-322-8DI-PD	0	–	–	0x600x	–
XN-322-16DO-P05	1	–	–	0x601x	0x701x
XN-322-4AI-PTNI	2	0x800x	0x802x	0x602x	–
XN-322-8AIO-U2	3	0x800z	0x803z	0x603x	0x703x

12.7.2 Start parameters for XN-312-GW-EC gateway

The objects in the 0x8000 address range are parameters and accordingly correspond to the SDOs. This is in contrast to the process data that is sent cyclically, i.e., the PDOs. The SDOs can be viewed for all XN300 slice modules under the "XN_312_GW_EC“/“Start parameters“ tab. Changes must be made in the "Start parameters" tab for the relevant XN300 slice module.

When the program starts, these parameters are sent by the XN-312-GW-EC gateway to the XN300 slice modules.

The following specific SDOs and start parameters are the ones in the sample project

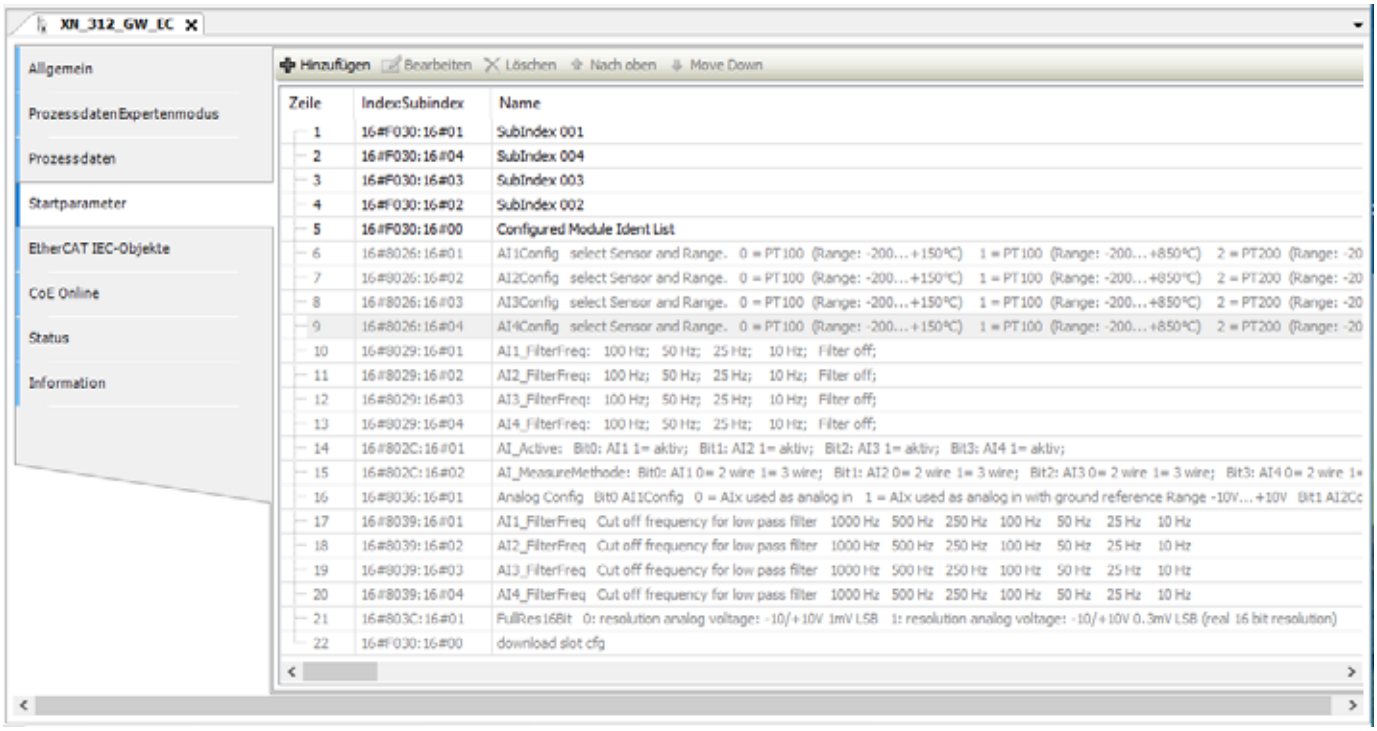


Figure 58: XN_312_GW_EC/Start parameters tab

Start parameters for XN-312-GW-EC gateway; the read start parameters for all XN300 slice modules

Object index (hex)	Data type	Name	Function	Mapping	Access	
0xF030	UINT	ModuleID	Module Identification Number → Section "4.3.2 Configured module ident list ((0×F030)", page 36	–	ro	SDO
Start parameters for XN-322-4AI-PTNI						
0x8021#01	USINT	UserLEDControl	User LED Control → Section "4.2.3 Configuration data object area (0×8xxx)", page 32	–	rw	SDO
0x8026#01	USINT	SensorSelectChannel1	Sensor Type Selection Channel 1	-	rw	SDO
0x8026#02	USINT	SensorSelectChannel2	Sensor Type Selection Channel 2	-	rw	SDO
0x8026#03	USINT	SensorSelectChannel3	Sensor Type Selection Channel 3	-	rw	SDO
0x8026#04	USINT	SensorSelectChannel4	Sensor Type Selection Channel 4	-	rw	SDO

Object index (hex)	Data type	Name	Function	Mapping	Access	
0x802C#2	USINT	ChannelMeasuringConfig	Channel Measuring Configuration (two-wire/three-wire measurement)	-	rw	SDO
0x8029#1	UINT	FilterConfigChannel1	Filter Configuration Channel 1	-	rw	SDO
0x8029#2	UINT	FilterConfigChannel2	Filter Configuration Channel 2	-	rw	SDO
0x8029#3	UINT	FilterConfigChannel3	Filter Configuration Channel 3	-	rw	SDO
0x8029#4	UINT	FilterConfigChannel4	Filter Configuration Channel 4	-	rw	SDO
0x802C#1	USINT	ChannelActivation	Channel Activation	-	rw	SDO
Start parameters for XN-322-8AIO-U2						
0x8036	UINT	AnalogInputSelection	Analog Input Selection	–	rw	SDO
0x8039#1	UINT	FilterConfigChannel1	Filter Configuration Channel 1	–	rw	SDO
0x8039#2	UINT	FilterConfigChannel2	Filter Configuration Channel 2	–	rw	SDO
0x8039#3	UINT	FilterConfigChannel3	Filter Configuration Channel 3	–	rw	SDO
0x8039#4	UINT	FilterConfigChannel4	Filter Configuration Channel 4	–	rw	SDO
0x803C#1	~~~''''					

12.7.3 Process data XN-312-GW-EC gateway

The process data for all XN300 slice modules is provided to the XN-312-GW-EC gateway. This data can be found under the XN_312_GW_EC/Process data tab.

Outputs are in the 0x7000 range. For the outputs, this means that:

- All 0x7010 objects are XN-322-16DO-P05 outputs.
- All 0x7031 objects are XN-322-8AIO-U2 outputs (below)

The inputs are transmitted with objects in the 6000 range. For the inputs, this means that:

- Object 0x6000 – all XN-322-8DI-PD inputs
- Object 0x601F – XN-322-16DO-P05 voltage OK input
- All 0x6021 objects – XN-322-4AI-PTNI inputs
- All 0x6031 objects – XN-322-8AIO-U2 inputs

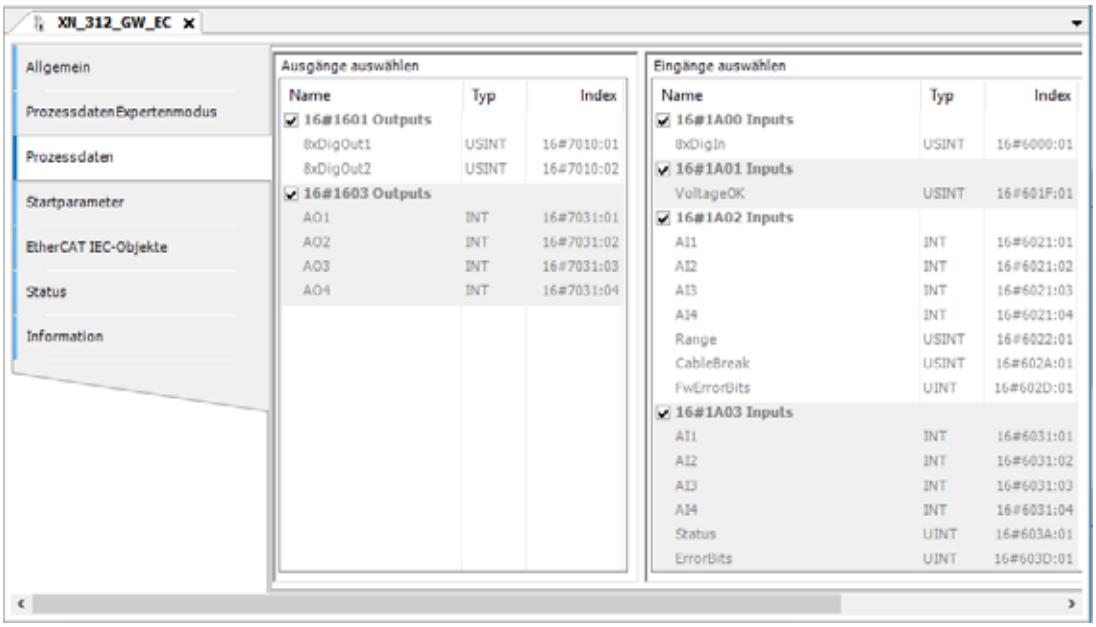
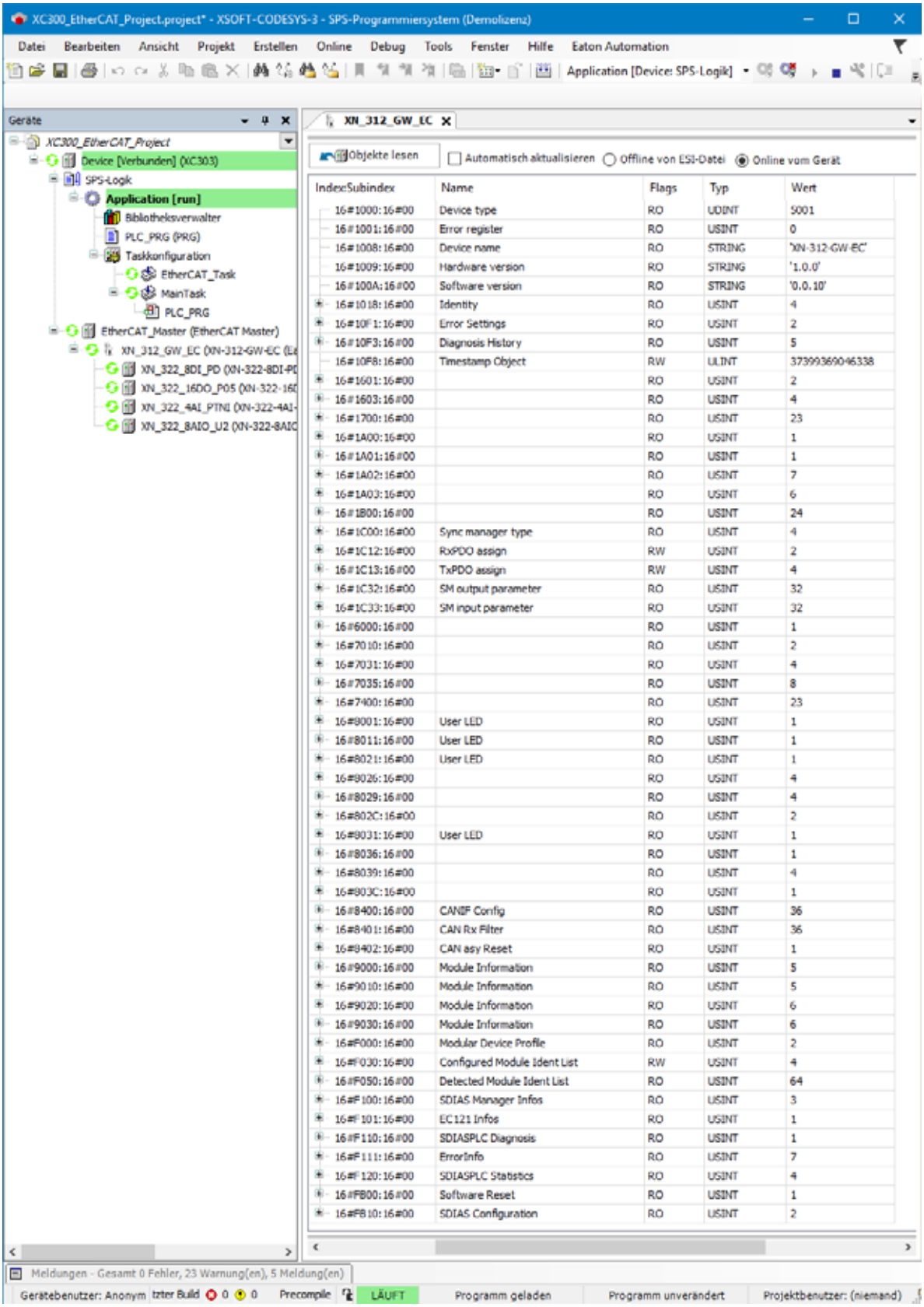


Figure 59: XN_312_GW_EC/Process data tab with process data for the gateway and all XN300 slice modules offline

12.7.4 ONLINE CoE XN-312-GW-EC gateway

CoE (CAN application layer over EtherCAT) is the complete Ethercat gateway object dictionary to which the objects of all XN300 slice modules are mapped. It contains the process data and the start parameters. If communication is offline, the CoE will be generated based on the description files for the individual devices. If the CoE is in online communication, the current values will be shown.

▶ Go online; please refer to the following as well → Section "12.5 Going online and loading the program onto the PLC".
▶ Enable the "Update automatically" option.
▶ Enable the "Online from device" option


XC300_EtherCAT_Project.project* - XSOFT-CODESYS-3 - SPS-Programmiersystem (Demolizenz)
Datei Bearbeiten Ansicht Projekt Erstellen Online Debug Tools Fenster Hilfe Eaton Automation
Application [Device: SPS-Logik]
Geräte
XC300_EtherCAT_Project
Device [Verbunden] (XC303)
SPS-Logik
Application [run]
Bibliotheksverwalter
PLC_PRG (PRG)
Taskkonfiguration
EtherCAT_Task
MainTask
PLC_PRG
EtherCAT_Master (EtherCAT Master)
XN_312_GW_EC
XN_312_GW_EC
Objekte lesen
Automatisch aktualisieren
Offline von ESI-Datei
Online vom Gerät
Index:Subindex | Name | Flags | Typ | Wert
16#1000:16#00 | Device type | RO | UDINT | 5001
16#1001:16#00 | Error register | RO | USINT | 0
16#1008:16#00 | Device name | RO | STRING | 'XN-312-GW-EC'
16#1009:16#00 | Hardware version | RO | STRING | '1.0.0'
16#100A:16#00 | Software version | RO | STRING | '0.0.10'
16#1018:16#00 | Identity | RO | USINT | 4
16#10F1:16#00 | Error Settings | RO | USINT | 2
16#10F3:16#00 | Diagnosis History | RO | USINT | 5
16#10F8:16#00 | Timestamp Object | RW | ULINT | 37399369046338
16#1601:16#00 | | RO | USINT | 2
16#1603:16#00 | | RO | USINT | 4
16#1700:16#00 | | RO | USINT | 23
16#1A00:16#00 | | RO | USINT | 1
16#1A01:16#00 | | RO | USINT | 1
16#1A02:16#00 | | RO | USINT | 7
16#1A03:16#00 | | RO | USINT | 6
16#1B00:16#00 | | RO | USINT | 24
16#1C00:16#00 | Sync manager type | RO | USINT | 4
16#1C12:16#00 | RxPDO assign | RW | USINT | 2
16#1C13:16#00 | TxPDO assign | RW | USINT | 4
16#1C32:16#00 | SM output parameter | RO | USINT | 32
16#1C33:16#00 | SM input parameter | RO | USINT | 32
16#6000:16#00 | | RO | USINT | 1
16#7010:16#00 | | RO | USINT | 2
16#7031:16#00 | | RO | USINT | 4
16#7035:16#00 | | RO | USINT | 8
16#7400:16#00 | | RO | USINT | 23
16#8001:16#00 | User LED | RO | USINT | 1
16#8011:16#00 | User LED | RO | USINT | 1
16#8021:16#00 | User LED | RO | USINT | 1
16#8026:16#00 | | RO | USINT | 4
16#8029:16#00 | | RO | USINT | 4
16#802C:16#00 | | RO | USINT | 2
16#8031:16#00 | User LED | RO | USINT | 1
16#8036:16#00 | | RO | USINT | 1
16#8039:16#00 | | RO | USINT | 4
16#803C:16#00 | | RO | USINT | 1
16#8400:16#00 | CANIF Config | RO | USINT | 36
16#8401:16#00 | CAN Rx Filter | RO | USINT | 36
16#8402:16#00 | CAN asy Reset | RO | USINT | 1
16#9000:16#00 | Module Information | RO | USINT | 5
16#9010:16#00 | Module Information | RO | USINT | 5
16#9020:16#00 | Module Information | RO | USINT | 6
16#9030:16#00 | Module Information | RO | USINT | 6
16#F000:16#00 | Modular Device Profile | RO | USINT | 2
16#F030:16#00 | Configured Module Ident List | RW | USINT | 4
16#F050:16#00 | Detected Module Ident List | RO | USINT | 64
16#F100:16#00 | SDIAS Manager Infos | RO | USINT | 3
16#F101:16#00 | EC121 Infos | RO | USINT | 1
16#F110:16#00 | SDIASPLC Diagnosis | RO | USINT | 1
16#F111:16#00 | ErrorInfo | RO | USINT | 7
16#F120:16#00 | SDIASPLC Statistics | RO | USINT | 4
16#FB00:16#00 | Software Reset | RO | USINT | 1
16#FB10:16#00 | SDIAS Configuration | RO | USINT | 2
Meldungen - Gesamt 0 Fehler, 23 Warnung(en), 5 Meldung(en)
Gerätebenutzer: Anonym
Precompile
LÄUFT
Programm geladen
Programm unverändert
Projektbenutzer: (niemand)

12.8 Finding XN300 slice module PDOs and SDOs

If the objects are mapped, the process data for each individual XN300 slice module will be found in the corresponding XN300 slice module/Module I/O image table tab. The „ XN-300 Slice Modules“, MN050002-EN manual describes the process data in the "memory layout" subsection for each relevant XN300 slice module.

12.8.1 XN-322-8DI-PD

This XN300 slice module does not map any SDOs to the object dictionary. Only PDOs can be shown under the XN_322_8DI_PD/Module I/O image table tab.

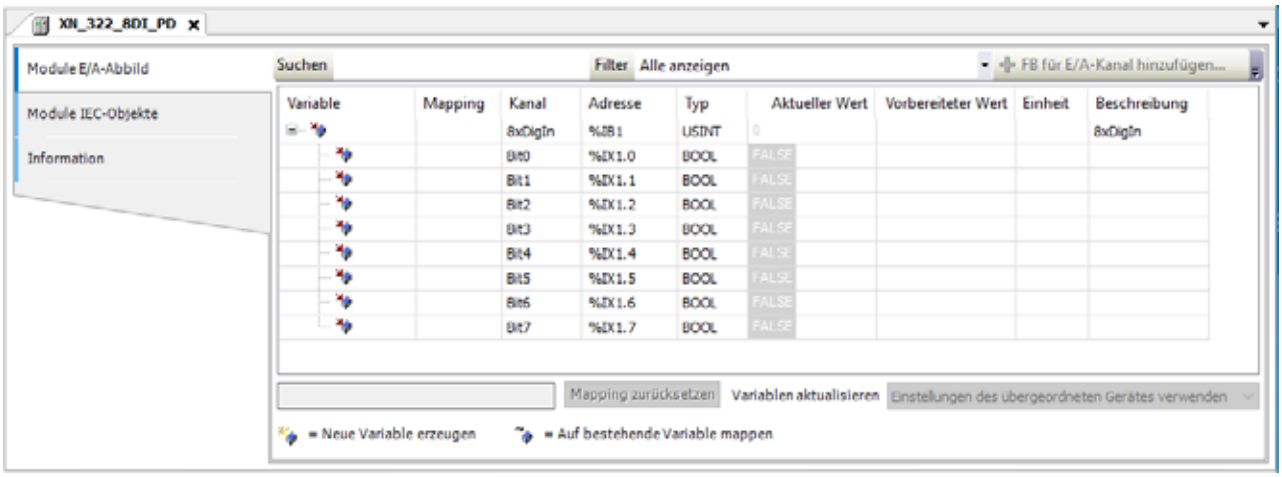


Figure 60: XN_322_8DI_PD/Module I/O image table tab

The XN300 slice module's process data is also stored in the XN-312-GW-EC gateway process data, in the $6xxx_{hex}$ object range. In the 0x6xxx range, since they are inputs exclusively. The ModuleID in the sample project is 0, which is why the inputs are stored in object 0x6000.

Index (hex)	Data type	Name	Function	Mapping	Access	
0x1A00 :#01		TxPDO mapping	→ Section “4.1.7 TxPDO mapping object (0×1A00…0×1BFF)”, page 29			
0x6000	USINT	Input1_8	Read Digital Input 1_8		ro	PDO
0x8001 :#01	USINT	UserLEDCon-trol	User LED Control → Section “4.2.3 Configuration data object area (0×8xxx)”, page 32	–	rw	SDO

Index (hex)	Data type	Name	Function	Mapping	Access	
0x9000 : #02	UINT	DeviceID	Module Identification Number → Section "5.1 Module ID number", page 43.	–	ro	SDO
0x9000 :#05	VISIBLE STRING	SerialNumber	Serial number	–	const	SDO

The module information, such as the ModuleID, serial number, HW version, and firmware version is shown during online communication in the Ethercat gateway CoE in the 0x9000 objects. In the case of XN-322-8DI-PD, specifically in object 9000, since it is the first node on the system bus; please refer to → Section "12.7.4 ONLINE CoE XN-312-GW-EC gateway", page 90.

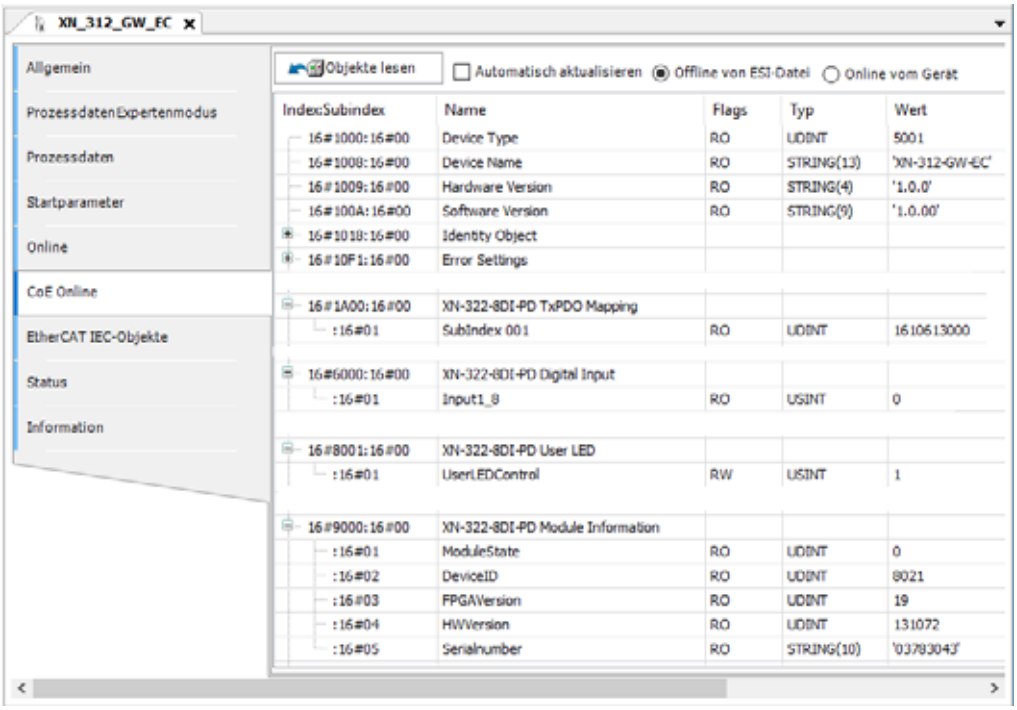


Figure 61: XN_312_GW_EC/CoE Online tab excerpt

12.8.2 XN-322-16DO-P05

This XN300 slice module does not map any SDOs to the object dictionary. Only PDOs can be shown under the XN_322_16DO_P05/Module I/O image table tab.

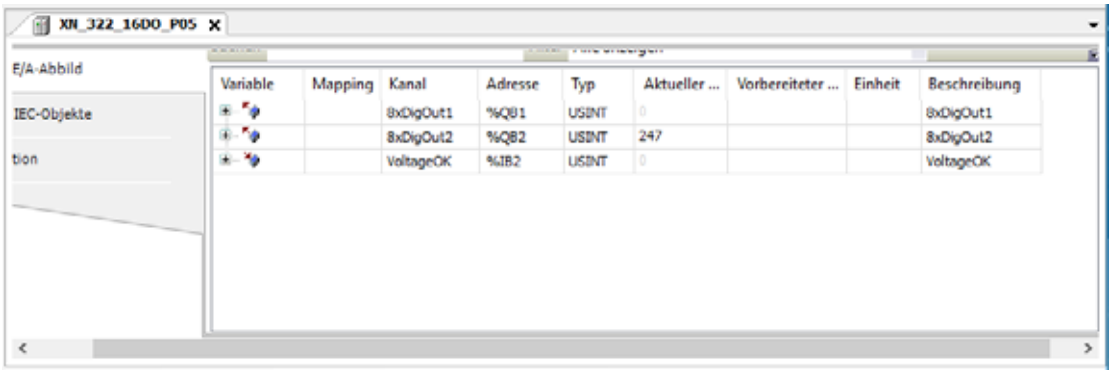


Figure 62: XN_322_16DO_P05/Module I/O image table tab

The XN300 slice module's process data is also stored in the XN-312-GW-EC gateway process data, in the 0x7xxx object range. In the 0x7xxx range, since they are inputs exclusively. The ModuleID in the sample project is 1, and the outputs are accordingly stored in object 0x7010.

Object index (hex)	Data type	Name	Function	Mapping	Access	
0x1601: #01		RxPDO mapping	→ Section "4.1.6 RxPDO mapping object (0×1600…0×17FF)", page 28		ro	PDO
0x1601: #02		RxPDO mapping			ro	
0x1A01: #01		TxPDO mapping	→ Section "4.1.7 TxPDO mapping object (0×1A00…0×1BFF)", page 29		ro	
0x601F: #01	USINT	VoltageOk	Input Voltage State Bit 0: DC 24V Output 1..16 OK		ro	PDO
0x7010: #01	USINT	Output 1_8	Write digital output 1_8		rww	PDO
0x7010: #02	USINT	Output 9_16	Write digital output 9_16 → Section "4.2.2 Output data object area (0×7xxx)", page 32		rww	PDO
0x8011: #01	USINT	UserLEDControl	User LED Control → Section "4.2.3 Configuration data object area (0×8xxx)", page 32	–	rw	SDO
0x9010:#02	UINT	DeviceID	Module Identification Number → Section "5.1 Module ID number", page 43.	–	ro	SDO
0x9010:#05	VISIBLE STRING	SerialNumber	Serial number → Section " Serial number", page 34.	–	const	SDO

The module information, such as the DeviceID, serial number, HW version, and FirmwareVersion is shown during online communication in the Ethercat gateway CoE in the 9xxx objects. In the case of XN-322-16DO-P05 specifically in object 9010 because it is the second node on the system bus; please refer to → Section "12.7.4 ONLINE CoE XN-312-GW-EC gateway", page 90.

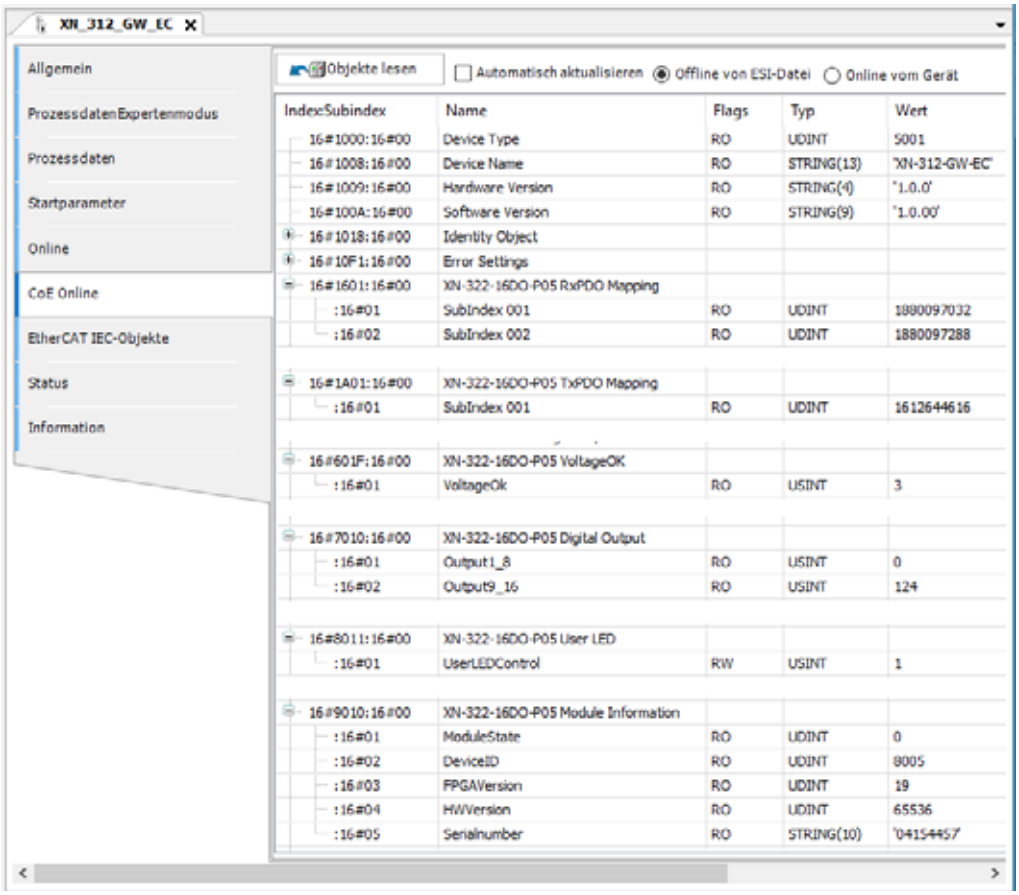


Figure 63: XN_312_GW_EC/CoE Online tab excerpt

12.8.3 XN-322-4AI-PTNI

SDO start parameters for XN-322-4AI-PTNI

The SDOs can be shown under the XN_322_4AI_PTNI/Start parameters tab.

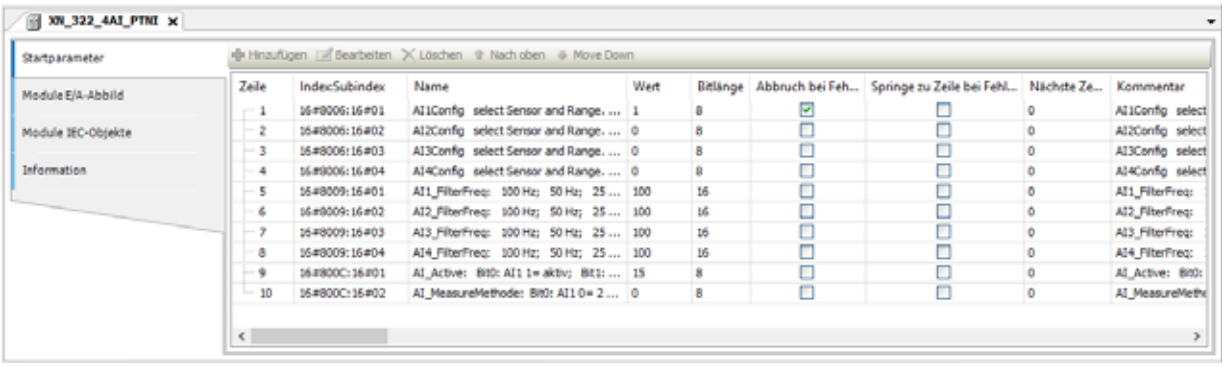


Figure 64: XN-322-4AI-PTNI start parameters

PDO process data for XN-322-4AI-PTNI

The PDOs can be shown under the XN_322_4AI_PTNI/Module I/O image table tab.

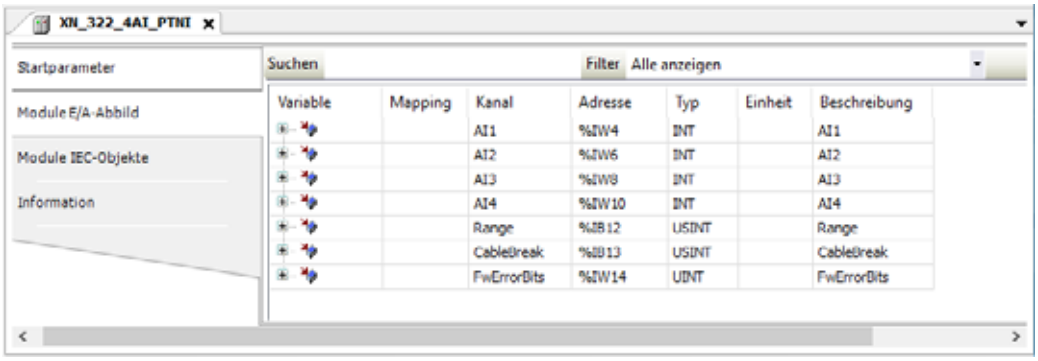


Figure 65: XN-322-4AI-PTNI module input/output parameters

The XN300 slice module's process data is also stored in the XN-312-GW-EC gateway process data. The analog inputs are stored in the 0x6xxx object range. The ModuleID in the sample project is 2, and the inputs are accordingly stored in object 0x6021. The module does not have any outputs.

The description of the objects for the analog inputs; please refer to
→ Section “13.4.5 Analog input”, page 109.

12.8.4 XN-322-8AIO-U2

SDO start parameters for XN-322-8AIO-U2

The SDOs can be shown under the XN_322_8AIO_U2/Start parameters tab.

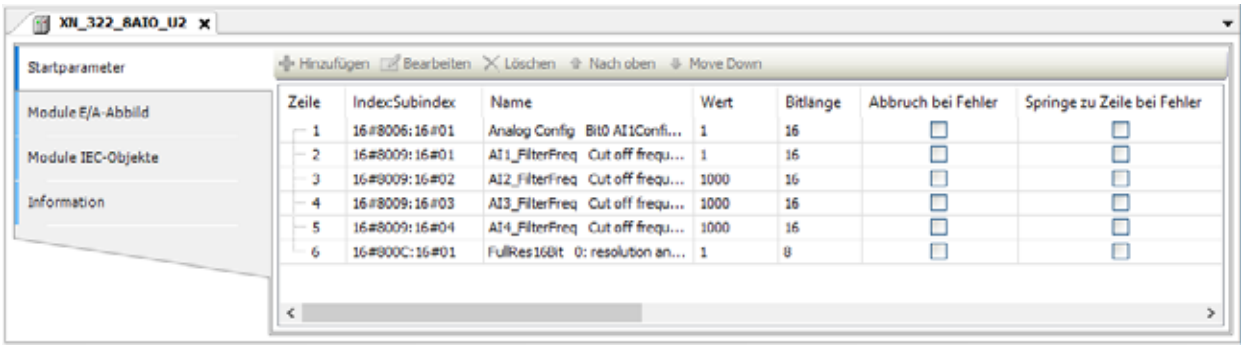


Figure 66: Start parameters for XN-322-8AIO-U2

PDO process data for XN-322-8AIO-U2

The PDOs can be shown under the XN_322_8AIO_U2/Module I/O image table tab.

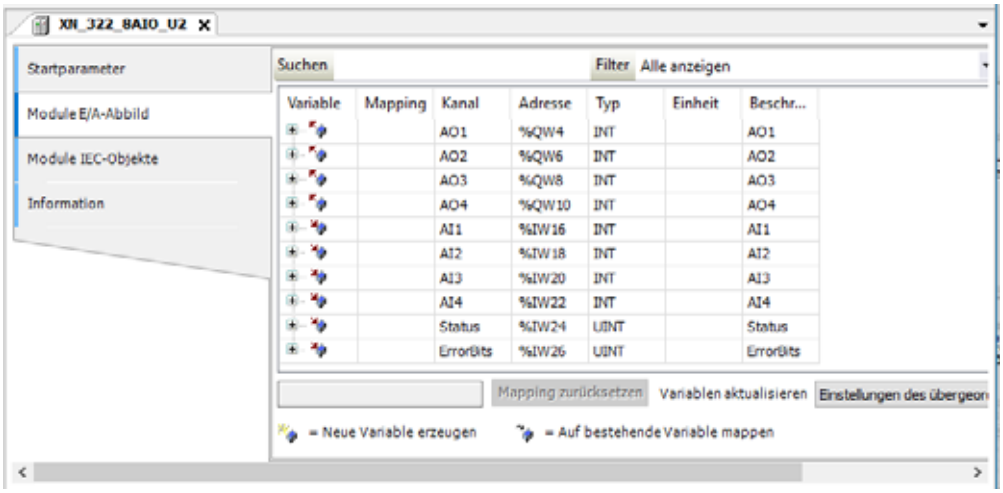


Figure 67: XN-322-8AIO-U2 module I/O parameters

The XN300 slice module's process data is also stored in the gateway process data. The analog inputs are stored in the 0x6xxx object range. The ModuleID in the sample project is 3, and the inputs are accordingly stored in object 0x6031. The analog outputs are stored in the 0x7031 object range.

The description of the objects for the analog inputs and outputs; please refer to → Section “13.4.7 Analog input/output”, page 112.

13 Appendix

13.1 Approvals

Product standards	• EN 61131-2 (programmable logic controllers); • UL 61010-2-201 (industrial controllers); • CE-mark
Approvals	• CE-mark • cULus (in preparation)

13.2 Dimensions

		XN-312-GW-EC
Dimensions (H × D × W)	mm	105 x 72 x 25
	inch	4.13 × 2.83 × 0.98
Space units (SU) width		
Weight	kg	0.085
	lb	0.19
Mounting		DIN-rail IEC EN 60715, 35 mm
Mounting position		horizontal

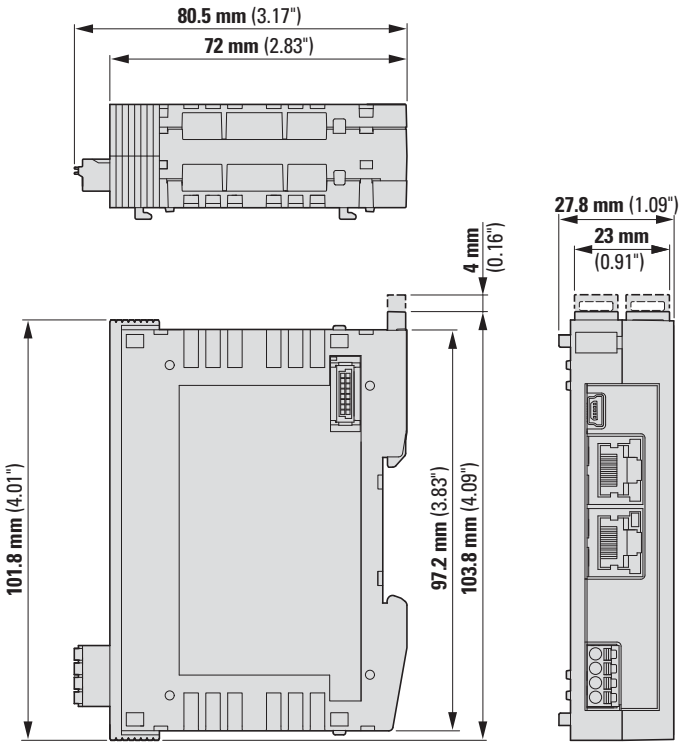


Figure 68: XN-312-GW-EC dimensions

13.3 Technical data

EtherCAT gateway			**XN-312-GW-EC**
General			
Standards			IEC/EN 61131-2
Dimensions (W x H x D) (without plug)		mm	115 x 72 x 23
Weight		kg	0.085
Mounting			DIN-rail IEC/EN 60715, 35 mm
Connection type			System plug
Ambient climatic conditions			
Operating ambient temperature (IEC 60068-2)		°C	-25…+60
Condensation			Must be prevented with suitable measures
Storage		°C	-40…85
Relative humidity, non-condensing (IEC/EN 60068-2-30)		%	0…95
Ambient mechanical conditions			
Degree of protection, IEC/EN 60529			IP20_x
Vibrations (IEC/EN 61131-2:2008)			
constant amplitude 3.5 mm		Hz	5…8.4
Constant acceleration 1 g		Hz	8.4…150
Mechanical shock resistance (IEC/EN 61131-2:2008) semi-sinusoidal 15 g/11 ms		Impact resis-tances	18
Free fall, packaged (IEC/EN 60068-2-32)		m	0.3
Mounting position			horizontal
Installation altitude		m	0…2000
Specifications for connection to supply voltage			
Rated operating voltage	U_e	V	24 DC
admissible range		V	18 - 30 DC
Residual ripple of input voltage		%	5
Protection against polarity reversal			Yes
rated operational current	I_e	A	2.5 max.
Internal power loss with max. permissible load		W	2.5 W
Galvanic isolation between power supply and 24/5 VDC system bus voltage			No
Bridging voltage dips			
Duration of dip		ms	10
Repetition rate		s	1

EtherCAT gateway			**XN-312-GW-EC**
Field bus interface			EtherCAT
Potential isolation between supply and field bus			Yes
Measurement of the clearance and creepage distance			IEC/EN 61131-2
EtherCAT – baud rate		Mbps	100
2 x shielded RJ45 ports			
Max. cable length between 2 stations		m	100
Propagation delay		µ	1
Potential separation		V	500
FMMU			3
Sync manager			4
RAM process data		kB	8
Synchronization			64 bit Distibuted clock
Process image			Modular process image
Asynchronous data exchange			
CoE protocol			
FoE protocol			
FSoE protocol			
Maximum number of XN300 slice modules			32
Electromagnetic compatibility (EMC)			
Overvoltage category			II
Pollution degree			2
Electrostatic discharge (IEC/EN 61131-2:2008, ESD)			
Air discharge (Level 3)		kV	8
Contact discharge (Level 2)		kV	4
Electromagnetic Fields (IEC/EN 61131-2:2008)			
(80 …. 1000) MHz		V/m	10
(1.4 … 2) GHz		V/m	3
(2 … 2.7) GHz		V/m	1
Radio interference suppression			EN 55011 Class A
Burst (IEC/EN 61131-2:2008, Level 3)			
Supply cables		kV	2
EtherCAT bus cables		kV	1
Surge (IEC/EN 61131-2:2008, Level 1)			

EtherCAT gateway			**XN-312-GW-EC**
Supply		kV	0.5
Field bus		kV	1
Radiated RFI (IEC/EN 61131-2:2008, level 3)		V	10

13.4 XN300 slice module objects

13.4.1 Data types

The following table shows a list of all data types for the objects and their corresponding definition

Data type	Designation	Description
INT	Signed integer	16 bit, with sign
DINT	Signed double integer	32 bit, with sign
SINT	Signed small integer	8 bit, with sign
UINT	Unsigned integer	16 bit, unsigned
USINT	Unsigned small integer	8 bit, unsigned
UDINT	Unsigned double integer	32 bit, unsigned
Visible string		String
VAR		Array of tags of various data types
ARRAY		Array of tags with the same data type
RECORD		

13.4.2 Digital input

EtherCAT Object	Data type	XN-322-8DI-PD Name	XN-322-16DI-PD Name	XN-322-20DI-PD Name	XN-322-20DI-PF Name	XN-322-20DI-ND Name	XN-322-20DI-PCNT Name
16#6xx0		Digital input	Digital input	Digital input	Digital input	Digital input	Digital input
16#6xx0 : #01	USINT	Input1_8	Input1_8	Input1_8	Input1_8	Input1_8	Input1_8
16#6xx0 : #02	USINT	–	Input9_16	Input9_16	Input9_16	Input9_16	Input9_16
16#6xx0 : #03	USINT	–	–	Input17_20	Input17_20	Input17_20	Input17_20
16#6xx4		–	–	–	–	–	Counter
16#6xx4 : #01	USINT	–	–	–	–	–	Counter1_X1/I1
16#6xx4 : #02	USINT	–	–	–	–	–	Counter2_X1/I2
16#6xx4 : #03	USINT	–	–	–	–	–	Counter3_X1/I3
16#6xx4 : #04	USINT	–	–	–	–	–	Counter4_X1/I4
16#8xx1		User LED	User LED	User LED	User LED	User LED	User LED
16#8xx1 : #01	USINT	User LED control[0]	User LED control[0]	User LED control[0]	User LED control[0]	User LED control[0]	User LED control[0]
16#8xx4		–	–	–	–	–	Counter Mode
16#8xx4 : #01	USINT	–	–	–	–	–	Counter Mode
16#9xx1		Module information	Module information	Module information	Module information	Module information	Module information
16#9xx1 : #01	UDINT	module status	module status	module status	module status	module status	module status
16#9xx1 : #02	UDINT	device ID	device ID	device ID	device ID	device ID	device ID
16#9xx1 : #03	UDINT	FPGA version	FPGA version	FPGA version	FPGA version	FPGA version	FPGA version
16#9xx1 : #04	UDINT	HW version	HW version	HW version	HW version	HW version	HW version
16#9xx1 : #05	Visible string	Serial number	Serial number	Serial number	Serial number	Serial number	Serial number
16#9xx1 : #06	UDINT	–	–	–	–	–	–

13.4.3 Digital output

EtherCAT Object	Data type	XN-322-12DO-P17 Name	XN-322-16DO-P05 Name	XN-322-8DO-P05 Name	XN-322-4DO-RNO Name
16#6xxF		Module status	Module status	Module status	Module status
16#6xxF : #01	USINT	SupplyVoltageState	SupplyVoltageState	SupplyVoltageState	–
16#7xxx0		DigitalOutput	DigitalOutput	DigitalOutput	DigitalOutput
16#7xxx0 : #01	USINT	Output1_8	Output1_8	Output1_8	Output1_4
16#7xxx0 : #02	USINT	Output9_12	Output9_16	–	–
16#8xx1		User LED	User LED	User LED	User LED
16#8xx1 : #01	USINT	User LED control[0]	User LED control[0]	User LED control[0]	User LED control[0]
16#9xx0		Module information	Module information	Module information	Module information
16#9xx0 : #01	UDINT	module status	module status	module status	module status
16#9xx0 : #02	UDINT	device ID	device ID	device ID	device ID
16#9xx0 : #03	UDINT	FPGA version	FPGA version	FPGA version	FPGA version
16#9xx0 : #04	UDINT	HW version	HW version	HW version	HW version
16#9xx0 : #05	Visible string	Serial number	Serial number	Serial number	Serial number
16#9xx0 : #06	UDINT	–	–	–	–

13.4.4 Digital input/output

EtherCAT Object index	Data type	XN-322-16DIO-PD05 Name	XN-322-16DIO-PC05 Name	XN-322-8DIO-PD05 Name
16#6xx0		Digital input	Digital input	Digital input
16#6xx0 : #01	USINT	Input1_8	Input1_8	Input1_8
16#6xx4		–	Counter	–
16#6xx4 : #01	UINT	–	EncoderCounter1_X1/I1 I2	–
16#6xx4 : #02	UINT	–	EncoderCounter2_X1/I3I4	–
16#6xx5		–	PWMPeriod	–
16#6xx5 : #01	UINT		PwmHighTime1_X1/I1	
16#6xx5 : #02	UINT		PwmHighTime2_X1/I2	
16#6xx5 : #03	UINT		PwmHighTime3_X1/I3	
16#6xx5 : #04	UINT		PwmHighTime4_X1/I4	
16#6xx5 : #05	UINT		PwmPeriod1_X1/I1	
16#6xx5 : #06	UINT		PwmPeriod2_X1/I2	
16#6xx5 : #07	UINT		PwmPeriod3_X1/I3	
16#6xx5 : #08	UINT		PwmPeriod4_X1/I4	
16#6xx6		–	Timestamp (object for internal purposes only)	–
16#6xx6 : #01	UINT	–	Timestamp1	–
16#6xx6 : #02	UINT	–	Timestamp2	–
16#6xx6 : #03	UINT		Timestamp3	
16#6xx6 : #04	UINT		Timestamp4	
16#6xxF		Module status	Module status	Module status
16#6xxF : #01	USINT	SupplyVoltageState	SupplyVoltageState	SupplyVoltageState
0x7xx0		DigitalOutput	DigitalOutput	DigitalOutput
0x7xx0 : #01	USINT	Output1_8	Output1_8	Output1_4
0x8xx1		User LED	User LED	User LED
0x8xx1 : #01	USINT	User LED control[0]	User LED control[0]	User LED control[0]
0x8xx4		–	CounterMode	–

EtherCAT Object index	Data type	XN-322-16DIO-PD05 Name	XN-322-16DIO-PC05 Name	XN-322-8DIO-PD05 Name
0x8xx4	USINT	–	CounterModeRegister	–
0x9xx1		Module information	Module information	Module information
0x9xx1 : #01	UDINT	module status	module status	module status
0x9xx1 : #02	UDINT	device ID	device ID	device ID
0x9xx1 : #03	UDINT	FPGA version	FPGA version	FPGA version
0x9xx1 : #04	UDINT	HW version	HW version	HW version
0x9xx1 : #05	Visible string	Serial number	Serial number	Serial number
0x9xx1 : #06	UDINT	–	–	–

13.4.5 Analog input

EtherCAT Object	Data type	XN-322-8AI-I Name	XN-322-7AI-U2PT Name	XN-322-4AI-PTNI Name	XN-322-10AI-TEKT Name
16#6xx1		Analog input	Analog input	Analog input	Analog input
16#6xx1 : #01	USINT	InputChannelAI1_X1/1+1-	InputChannelAI1_X1/1+1-	InputChannelAI1_X1	InputChannelAI1_X1/1+1-
16#6xx1 : #02	USINT	InputChannelAI2_X1/2+2-	InputChannelAI2_X1/2+2-	InputChannelAI2_X2	InputChannelAI2_X1/2+2-
16#6xx1 : #03	USINT	InputChannelAI3_X2/3+3-	InputChannelAI3_X2/3+3-	InputChannelAI3_X3	InputChannelAI3_X2/3+3-
16#6xx1 : #04	USINT	InputChannelAI4_X2/4+4-	InputChannelAI4_X2/4+4-	InputChannelAI4_X4	InputChannelAI4_X2/4+4-
16#6xx1 : #05	USINT	InputChannelAI5_X3/5+5-	InputChannelAI5_X3/5+5-	–	InputChannelAI5_X3/5+5-
16#6xx1 : #06	USINT	InputChannelAI6_X3/6+6-	InputChannelAI6_X3/6+6-	–	InputChannelAI6_X3/6+6-
16#6xx1 : #07	USINT	InputChannelAI7_X4/7+7-	InputChannelAI7_X5/K+K-	–	InputChannelAI7_X4/7+7-
16#6xx1 : #07	USINT	InputChannelAI8_X4/8+8-	–	–	InputChannelAI8_X4/8+8-
16#6xx1 : #09	USINT	–	–	–	InputChannelKTY1_X5/K1+K1-
16#6xx1 : #0A	USINT	–	–	–	InputChannelKTY2_X5/K2+K2-
16#6xx1 : #0B	USINT	–	–	–	InternalTempKTY1
16#6xx1 : #0C	USINT	–	–	–	InternalTempKTY2
16#6xx2		–	–	Range diagnostics	–
16#6xx2 : #01	USINT	–	–	OverUnderflowDiag	–
16#6xxA		System diagnostics	System diagnostics	System diagnostics	System diagnostics
16#6xxA : #01	USINT	WireBreakDiag	WireBreakDiag	WireBreakDiag	WireBreakDiag
16#8xx1		User LED	User LED	User LED	User LED
16#8xx1 : #01	USINT	User LED control[0]	User LED control[0]	User LED control[0]	User LED control[0]
16#8xx6		Channel configuration	Channel configuration	Channel configuration	Channel configuration
16#8xx6 : #01	UINT	InputChannelConfig [0]	InInputChannelConfig [0]	(USINT) SensorSelectionAI1	(UDINT) SensorTypSelection
16#8xx6 : #02	UINT	–	–	(USINT) SensorSelectionAI2	(UINT) ColdJunctionAssignment
16#8xx6 : #03	UINT	–	–	(USINT) SensorSelectionAI3	–
16#8xx6 : #04	UINT	–	–	(USINT) SensorSelectionAI4	–
16#8xx7		Range control	–		–

EtherCAT Object	Data type	XN-322-8AI-I Name	XN-322-7AI-U2PT Name	XN-322-4AI-PTNI Name	XN-322-10AI-TEKT Name
16#8xx7 : #01	UINT	UnderRangeLimit [0000]	–	–	–
16#8xx9		Filter configuration	Channel configuration	Channel configuration	–
16#8xx9 : #01	UINT	FilterConfigAI1 [0]	FilterConfigAI1 [0]	FilterConfigAI1 [0]	–
16#8xx9 : #02	UINT	FilterConfigAI2 [0]	FilterConfigAI2 [0]	FilterConfigAI2 [0]	–
16#8xx9 : #03	UINT	FilterConfigAI3 [0]	FilterConfigAI3 [0]	FilterConfigAI3 [0]	–
16#8xx9 : #04	UINT	FilterConfigAI4 [0]	FilterConfigAI4 [0]	FilterConfigAI4 [0]	–
16#9xx1		Module information	Module information	Module information	Module information
16#9xx1 : #01	UDINT	module status	module status	module status	module status
16#9xx1 : #02	UDINT	device ID	device ID	device ID	device ID
16#9xx1 : #03	UDINT	FPGA version	FPGA version	FPGA version	FPGA version
16#9xx1 : #04	UDINT	HW version	HW version	HW version	HW version
16#9xx1 : #05	Visible string	Serial number	Serial number	Serial number	Serial number
16#9xx1 : #06	UDINT	Firmware version	Firmware version	Firmware version	Firmware version

13.4.6 Analog output

EtherCAT Object	Data type	XN-322-8AO-U2 Name
16#6xxD		Analog output
16#6xxD : #01	INT	OutputChannelAI1_X1/1+-
16#6xxD : #02	INT	OutputChannelAI2_X1/2+-
16#6xxD : #03	INT	OutputChannelAI3_X2/3+-
16#6xxD : #04	INT	OutputChannelAI4_X2/4+-
16#6xxD : #05	INT	OutputChannelAI5_X3/5+-
16#6xxD : #06	INT	OutputChannelAI6_X3/6+-
16#6xxD : #07	INT	OutputChannelAI7_X4/7+-
16#6xxD : #07	INT	OutputChannelAI8_X4/8+-
16#6xxD : #08	INT	–
16#8xx1		User LED
16#8xx1 : #01	USINT	User LED control[0]
16#9xx1		Module information
16#9xx1 : #01	UDINT	module status
16#9xx1 : #02	UDINT	device ID
16#9xx1 : #03	UDINT	FPGA version
16#9xx1 : #04	UDINT	HW version
16#9xx1 : #05	Visible string	Serial number
16#9xx1 : #06	UDINT	Firmware version

13.4.7 Analog input/output

EtherCAT Object	Data type	XN-322-4AI0-I Name	XN-322-8AI0-I Name	XN-322-4AI0-U2 Name	XN-322-8AI0-U2 Name
16#6xx1		Analog input	Analog input	Analog input	Analog input
16#6xx1 : #01	USINT	InputChannelAI 1_X1/1+1-	InputChannelAI 1_X1/1+1-	InputChannelAI 1_X1/1+1-	InputChannelAI 1_X1/1+1-
16#6xx1 : #02	USINT	InputChannelAI 2_X1/2+2-	InputChannelAI 2_X1/2+2-	InputChannelAI 2_X2/2+2-	InputChannelAI 2_X1/2+2-
16#6xx1 : #03	USINT	–	InputChannelAI 3_X2/3+3-	–	InputChannelAI 3_X2/3+3-
16#6xx1 : #04	USINT	–	InputChannelAI 4_X2/4+4-	–	InputChannelAI 4_X2/4+4-
16#6xx2		Range diagnostics	Range diagnostics	–	–
16#6xx2 : #01	UINT	OverUnderflowDiag	OverUnderflowDiag	–	–
16#6xxA		–	–	Channel diagnostics	Channel diagnostics
16#6xxA : #01	UINT	–	–	WireBreakDiag	WireBreakDiag
16#6xxD		Module diagnostics	Module diagnostics	Module diagnostics	Module diagnostics
16#6xxD : #01	UINT	ModuleDiag	ModuleDiag	ModuleDiag	ModuleDiag
16#7xx1		Analog output	Analog output	Analog output	Analog output
16#7xx1 : #01	INT	OutputChannel AO1_X2/1+1-	OutputChannel AO1_X3/1+1-	OutputChannel AO1_X3/1+-	OutputChannel AO1_X3/1+-
16#7xx1 : #02	INT	OutputChannel AO2_X2/2+2-	OutputChannel AO2_X3/2+2-	OutputChannel AO2_X3/2+-	OutputChannel AO2_X3/2+-
16#7xx1 : #03	INT	–	OutputChannel AO3_X4/3+3-	–	OutputChannel AO2_X4/3+-
16#7xx1 : #04	INT	–	OutputChannel AO4_X4/4+4-	–	OutputChannel AO2_X4/4+-
16#8xx1		User LED	User LED	User LED	User LED
16#8xx1 : #01	USINT	User LED control[0]	User LED control[0]	User LED control[0]	User LED control[0]
16#8xx6		Channel configuration	Channel configuration	Channel configuration	Channel configuration

EtherCAT Object	Data type	XN-322-4AI0-I Name	XN-322-8AI0-I Name	XN-322-4AI0-U2 Name	XN-322-8AI0-U2 Name
16#8xx6 : #01	USINT	InputChannel-Config [0]	InputChannel-Config [0]	InputChannel-Config [0]	InputChannel-Config [0]
16#8xx6 : #02	UINT	UnderRange-Limit [3000]	UnderRange-Limit [3000]	–	–
16#8xx6 : #03	USINT	OutputChannel-Config [0]	OutputChannel-Config [0]	–	–
16#8xx9		Filter configuration	Filter configuration	Filter configuration	Filter configuration
16#8xx9 : #01	UINT	FilterConfigAI1 [1000]	FilterConfigAI1 [1000]	FilterConfigAI1 [1000]	FilterConfigAI1 [1000]
16#8xx9 : #02	UINT	FilterConfigAI2 [1000]<	FilterConfigAI2 [1000]	FilterConfigAI2 [1000]	FilterConfigAI2 [1000]
16#8xx9 : #03	UINT	–	FilterConfigAI3 [1000]	–	FilterConfigAI3 [1000]
16#8xx9 : #04	UINT	–	FilterConfigAI4 [1000]	–	FilterConfigAI4 [1000]
16#8xxC		–	–	Features configuration	Features configuration
16#8xxC : #01	USINT	–	–	Resolution [0]	Resolution [0]
16#9xx1		Module information	Module information	Module information	Module information
16#9xx1 : #01	UDINT	module status	module status	module status	module status
16#9xx1 : #02	UDINT	device ID	device ID	device ID	device ID
16#9xx1 : #03	UDINT	FPGA version	FPGA version	FPGA version	FPGA version
16#9xx1 : #04	UDINT	HW version	HW version	HW version	HW version
16#9xx1 : #05	Visible string	Serial number	Serial number	Serial number	Serial number
16#9xx1 : #06	UDINT	Firmware version	Firmware version	Firmware version	Firmware version

13.4.8 Function module XN-322-2SSI

EtherCAT Object	Data type	XN-322-2SSI Name
16#6xx0		Data input
16#6xx0 : #01	INT	InputDataChannel1_X1
16#6xx0 : #02	INT	InputDataChannel2_X2
16#6xxF		Module status
16#6xxF : #01	USINT	DataTransmissionState
16#7xx9		Module control
16#7xx9 : #01	UINT	StartReadCycle
16#8xx1		User LED
16#8xx1 : #01	USINT	User LED control[0]
16#8xx7		Transmission control
16#8xx7 : #01	USINT	SystemPeriodTime [0]
16#8xx7 : #02	USINT	SyncEnable [0]
16#8xx7 : #03	USINT	SyncOut0Multiplier [0]
16#8xx7 : #04	USINT	SyncOut0EnableRegister [0]
16#8xxA		Transmission configuration
16#8xxA : #01	USINT	ConfigRegChannel1 [0]
16#8xxA : #02	USINT	ControlRegChannel1 [0]
16#8xxA : #03	USINT	ConfigRegChannel2 [0]
16#8xxA : #04	USINT	ControlRegChannel2 [0]
16#9xx1		Module information
16#9xx1 : #01	UDINT	module status
16#9xx1 : #02	UDINT	device ID
16#9xx1 : #03	UDINT	FPGA version
16#9xx1 : #04	UDINT	HW version
16#9xx1 : #05	Visible string	Serial number
16#9xx1 : #06	–	–

13.4.9 Function module XN-322-2DMS-WM

EtherCAT Object	Data type	XN-322-2DMS-WM Name
16#6xx1		Analog input
16#6xx1 : #01	DINT	InputChannel1_X1X2
16#6xx1 : #02	DINT	InputChannel2_X1X2
16#6xxA		System diagnostics
16#6xxA : #01	USINT	ADCDiagChannel1
16#6xxA : #02	USINT	ADCDiagChannel2
16#6xxD		Module diagnostics
16#6xxD : #01	UINT	ModuleDiag
16#8xx1		User LED
16#8xx1 : #01	USINT	User LED control[0]
16#8xx6		Channel configuration
16#8xx6 : #01	UINT	MeasuringConfigChannel1 [0]
16#8xx6 : #02	UINT	RangeConfigChannel1 [0]
16#8xx6 : #03	UINT	MeasuringConfigChannel2 [0]
16#8xx6 : #04	UINT	RangeConfigChannel2 [0]
16#8xx6 : #05	DINT	ZeroScaleOffsetChannel1 [0]
16#8xx6 : #06	DINT	FullScaleOffsetChannel1 [0]
16#8xx6 : #07	DINT	ZeroScaleOffsetChannel2 [0]
16#8xx6 : #07	DINT	FullScaleOffsetChannel2 [0]
16#9xx1		Module information
16#9xx1 : #01	UDINT	module status
16#9xx1 : #02	UDINT	device ID
16#9xx1 : #03	UDINT	FPGA version
16#9xx1 : #04	UDINT	HW version
16#9xx1 : #05	Visible string	Serial number
16#9xx1 : #06	UDINT	Firmware version

13.4.10 Function module XN-322-1DCD-B35

EtherCAT Object	Data type	XN-322-1DCD-B35 Name
16#6xx1		Analog values
16#6xx1 : #01	INT	DCDTemperature_K
16#6xx1 : #02	INT	DCMotorCurrent
16#6xx1 : #03	DINT	DCMotor_I2T
16#6xx2		Range diagnostics
16#6xx2 : #01	UINT	OverUnderflowDiag
16#6xxA		System diagnostics
16#6xxA : #01	UINT	DCMotorDiagLatch
16#6xxA : #02	UINT	DCMotorDiag
16#6xxA : #03	UINT	FWStatusReg
16#6xxD		
16#6xxD : #01	UINT	–
16#7xx4		LED operation
16#7xx4 : #01	USINT	TonPwmLED1
16#7xx4 : #02	USINT	TonPwmLED2
16#7xxA		Operation control
16#7xxA : #01	UINT	PeriodDuration
16#7xxA : #02	UINT	MotorControlReg
16#7xxB		Sequence control
16#7xxB : #01		DataSequence1
16#7xxB : #02		DataSequence2
16#7xxB : #03		DataSequence3
16#7xxB : #04		DataSequence4
16#8xx1		User LED
16#8xx1 : #01	USINT	User LED control[0]
16#8xxA		I2T Threshold Configuration
16#8xxA : #01	USINT	ClearOnWriteReg [1]

EtherCAT Object	Data type	XN-322-1DCD-B35 Name
16#8xxA : #02	UDINT	I2TSwitchOffThreshold [0]
16#9xx1		Module information
16#9xx1 : #01	UDINT	module status
16#9xx1 : #02	UDINT	device ID
16#9xx1 : #03	UDINT	FPGA version
16#9xx1 : #04	UDINT	HW version
16#9xx1 : #05	Visible string	Serial number
16#9xx1 : #06	UDINT	Firmware version

13.5 Examples showing how to configure the EtherCAT master in XSOFT-CODESYS-3

13.5.1 Enabling distributed clocks

To enable distributed clocks, the cycle time must be set in the EtherCAT master. The cycle time will be automatically transmitted to the EtherCAT nodes. The default value is 4 ms and can be adjusted under the "General" tab. For more information, please refer to the CODESYS online help (enter "distributed clocks" as a search term).

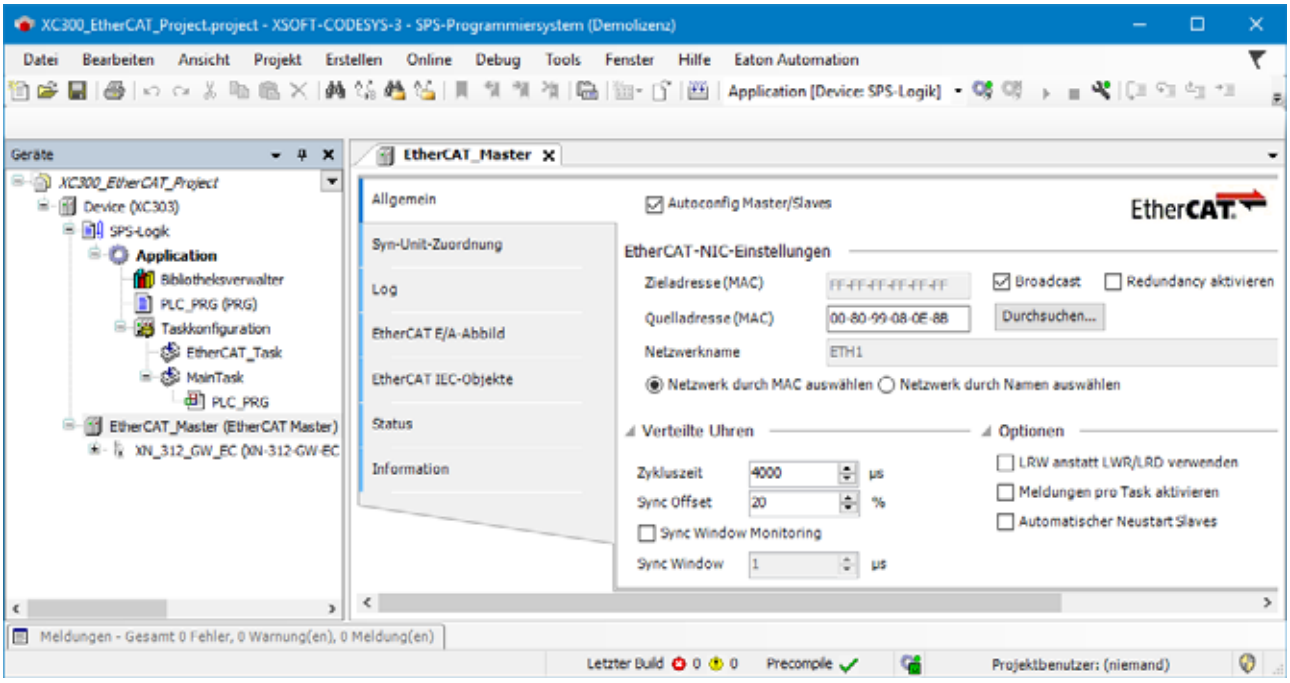


Figure 69: XSOFT-CODESYS-3 setting EtherCAT Master tab General section Distributed clocks

13.5.2 Redundancy

In addition, redundancy must be configured on the EtherCAT master by enabling the "Enable redundancy" option under the General tab. For information on the remaining configuration steps, please refer to the CODESYS online help (enter "configuring redundant PLC devices" as a search term).

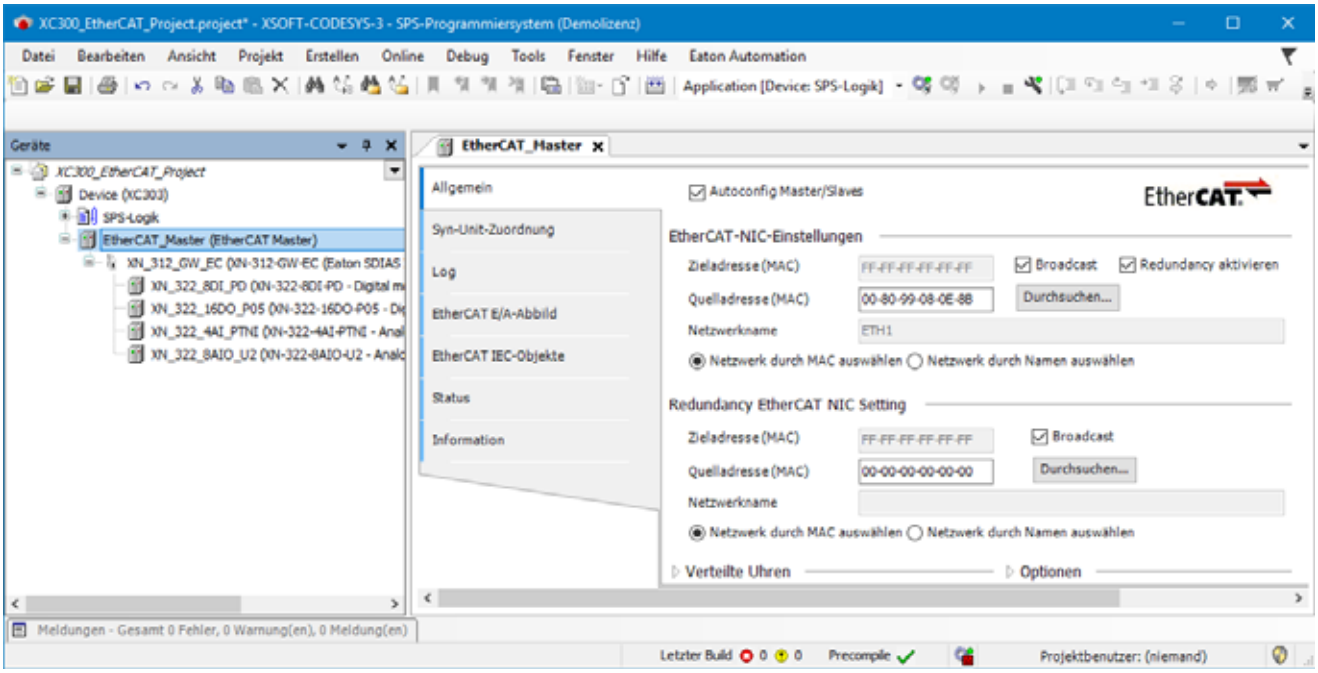


Figure 70: XSOFT-CODESYS-3 settings, EtherCAT Master General tab, Redundancy section

13.5.3 Watchdog

The watchdog times are configured on the XN-312-GW-EC Ethercat gateway. For information on the remaining configuration steps, please refer to the CODESYS online help (enter "Configuration tab" as a search term -> Watchdog).

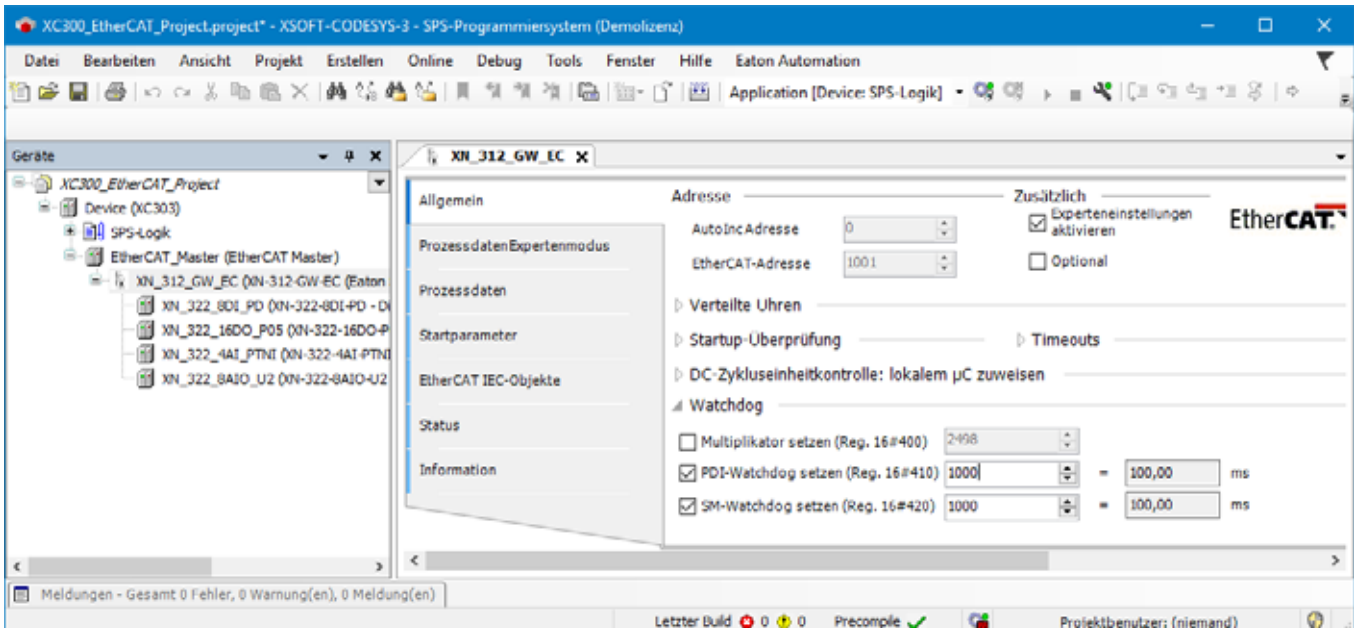


Figure 71: XSOFT-CODESYS-3 settings, EtherCAT Master General tab, Watchdog section

13.6 System limits

Table 4: System limits

	Gateway XN-312-GW-EC
Maximum number of nodes	32
Task cycle time (min/max)	500 µs to 32ms
Number of nodes on network	Depends exclusively on the EtherCAT master.
Data volume limitations	
Maximum PDO size	1194 bytes Rx and Tx each
	The relationship between the timing on the system bus and EtherCAT communication can be adjusted with the ISO start time (CoE SDO). Default value: 50%
Operating mode limitations	
Free run mode	In free run mode, the system bus cycle runs independently of the EtherCAT cycle This needs to be taken into account for the data refresh time.
DC synchronous mode	In DC synchronous mode, a maximum of 200 synchronous EtherCAT frames are allowed to miss the time window when using default settings. If necessary, task cycle times and task priorities must be optimized.
EtherCAT master with high jitter	In EtherCAT systems with high jitter, the "Sync Error counter limit" value must be increased to the maximum value of 65535 under tab XN_312_GW_EC/CoE object 0x10F1 Error Settings in the #02 sub-index if necessary.

13.7 Further reading and links

For more information on additional devices and modules, please visit the following links.

Product information

For up-to-date information, please consult the product page on the Internet

http://www.eaton.eu/XN300

Cybersecurity

For tips and recommendations on how to minimize security risks when using the product, please refer to document MZ050008EN, "Secure Hardening Guideline for EtherCAT Gateway XN-312-GW-EC." Please note that this document is only available in English.

Download Center — Documentation

You can find the documents on the Download Center - Documentation page by entering the document name.

http://www.eaton.eu/documentation

File type	Title	Designation
Mounting instruction	XN-312-GW-EC	IL050025ZU
Mounting instruction	XC300	IL050018ZU
Manual	XN-312-GW-EC	MN050010
Manual	XN300 slice modules	MN050002
Manual	XC300	MN050005
Manual	Secure Hardening Guideline for EtherCAT Gateway XN-312-GW-EC	MZ050008EN

Download Center — Software

The XSOFT-CODESYS-3 software described in this manual, as well as updates for the XN-312-… operating system, EDS files, application examples, and the XN300-Assist engineering tool, are available on the Internet from the Eaton Download Center — Software page:

http://www.eaton.eu/software

Category	Title	Designation
Software	XSOFT-CODESYS-3	XSOFT-CODESYS Vx.x.x SPx
Firmware Updates	XN300	XN-312-GW-EC FW Vx.xx

Category	Title	Designation
Firmware Updates	XC300	Included in: XSOFT-CODESYS Vx.x.x SPx
Tools	XN300-Assist	XN300-Assist Vx.xx
Application examples	XC300	XC300 Application examples CODESYS V3 for slice modules XN-322-20DI-PCNT XN-322-1CNT-8DIO, XN-322-1DCD-B35 XN-322-2SSI XN-322-2DMS-WM

References

[1] EtherCAT Technology Group, http://www.ethercat.org

[2] Industrial Communication with Fieldbus and Ethernet, 2010, VDE Verlag, pp. 151–168

[3] IEEE 802.3ae-2002: CSMA/CD Access Method and Physical Layer Specification

[4] IEC 61158-3…6-12, Industrial communication networks – Fieldbus specifications - Part 3–12: Data-link layer service definition – Part 4–12:Data-link layer protocol specification – Part 5–12: Application layer service definition – Part 6–12: Application layer protocol specification - Type 12 elements (EtherCAT).

Index

Eaton is dedicated to ensuring that reliable, efficient and safe power supply is available when it's needed most. With unparalleled knowledge of electrical power management across industries, experts at Eaton deliver customized, integrated solutions to solve our customers' most critical challenges.

Our focus is on delivering the right solution for the application. But, decision makers demand more than just innovative products. They turn to Eaton for an unwavering commitment to personal support that makes customer success a top priority. For more information, visit www.eaton.eu

Eaton addresses worldwide:
https://www.eaton.com/us/en-us/locate/global-locations.html

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

07/20 MN050010-EN Doku/Doku/Eb 07/20

Powering Business Worldwide