Manual

06/16 MN050003 EN

CANopen Gateway
XN-312-GW-CAN

Powering Business Worldwide

Brands and products are trademarks or registered trademarks of their owners.

Break-Down Service
Please call your local representative:
http://eaton.eu/aftersales or www.eaton.com/CustomerSupport

or
Hotline After Sales Service:
+49 (0) 180 5 223822 (de, en)
AfterSalesEGBonn@eaton.com

Eaton Industries GmbH
Hein-Moeller-Straße 7-11,
53115 Bonn, Germany

Eaton Corporation
W126 N7250 Flint Drive
Menomonee Falls,
WI 53051, USA

Contact your regional Eaton office for additional technical support

Americas	Europe – Middle East – Africa – Asia Pacific
Eaton Corporation	**Eaton Industries GmbH**
Electrical Sector	Electrical Sector EMEA
+1.800.426.9184	+49 (0) 180 5 223822 (de, en)
North America 1-877-ETN-CARE	www.eaton.eu/aftersales
www.eaton.com/CustomerSupport	www.eaton.eu/xn300

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 2014, publication date 12/14
2nd edition 2016, publication date 02/16
3rd edition 2016, publication date 06/16

Authors: Thomas Hettwer, Andreas Lüngen
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-CAN gateway.

The gateway is an integral part of the XN300 system, as are the slice modules with designation XN-312.

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. MN050003.

The standard EDS files can also be found in the Support Center by entering the search key word "EDS".

The instruction leaflet can be found by going to the Download Center - Documentation page in the Customer Support area and entering "IL050017ZU" into the search box.

Information on XN-322-... slice modules can be found in the following documents:

- Manual "XN-300 Slice modules", MN050002

Download Center

EDS files, the XN-300 Assist engineering tool, the XSOFT-CODESYS-2 and XSOFT-CODESYS-3 software described in this manual, and updates for the operating system for XN-312-… can all be downloaded from the Eaton Download Center on the Internet at:

http://www.eaton.eu/software or TRCAutomation@eaton.com

0.1 List of revisions

The following significant amendments have been introduced since previous issues:

Edition date	Page	Keyword	New	Changes
02/16	45	Bus utilization → Section "3.2.1 Creating field bus communication CANopen"		✓
	178	Table SDO value for sensor selection → Section "7.14.15 Sensor Type Selection Channel x (Object 0x5070 bis 0x5073)"		✓
	314, 315	Choosing period duration → Chapter 7 "Product-specific CAN objects XN300 slice modules"	✓	

06/16	178	Expansion for sensor selection„XN-322-4AI-PTNI“		✓
		The following chapters were added:		
	106	→ Section "7.1 XN-322-8DI-PD"	✓	
	109	→ Section "7.2 XN-322-16DI-PD"	✓	
	125	→ Section "7.6 XN-322-20DI-ND"	✓	
	128	→ Section "7.7 XN-322-4DO-RNO"	✓	
	131	→ Section "7.8 XN-322-8DO-P05"	✓	
	143	→ Section "7.11 XN-322-8DIO-PD05"	✓	
	148	→ Section "7.12 XN-322-16DIO-PD05"	✓	
	153	→ Section "7.13 XN-322-16DIO-PC05"	✓	
	234	→ Section "7.19 XN-322-4AIO-U2"	✓	
	266	→ Section "7.21 XN-322-4AIO-I"	✓	
	284	→ Section "7.22 XN-322-8AIO-I"	✓	

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 maintenance 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

- COB-ID - Communication OBject IDentifier
- DIP - Dual Inline Package
- EDS - Electronic Data Sheet
- PDO - Process Data Objects
- RPDO - Receive Process Data Objects
- SDO - Service Data Objects
- SSI - Synchronous Serial Interface
- TPDO - Transmit Process Data Objects
- XN300 - Device series, including the XN-312 gateway and XN-322 slice modules

Following designations XSOFT-CODESYS-2 are used:

- Module - System bus module
- Station
- Station address - Address of the field bus module

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 XN-312-GW-CAN gateway

1.1 Proper use

XN-312-GW-CAN gateways can be used to establish a connection between a higher-level PLC and a system bus with its modules.

These gateways use the CANopen protocol to communicate with higher-level PLCs. XN-312-GW-CAN gateways are part of the XN300 system, which, in addition to the gateways themselves, includes several I/O slice modules. These slice modules include both digital and analog I/O modules, as well as various technology modules with counting, weighing, and motor drive functionalities. These modules can be joined together without the use of tools in order to form a system block. All XN300 system slice modules communicate 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.

1.2 Overview of functions

XN-312-GW-CAN gateways can be used to connect a system bus to CANopen, and make it possible to access the data of up to 32 I/O modules using CANopen. 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 feature a config check function that can be activated with DIP switches. When this function is used, the gateway will check the I/O slice modules that are present on the system bus and compare the resulting actual configuration with the previously stored target configuration. If the actual configuration is different from the target configuration, it will not be possible for the gateway to switch to the "Operational" operating state.

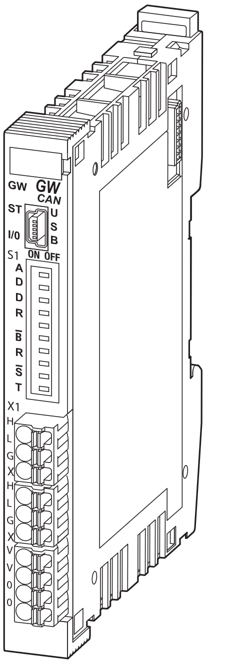


Figure 1: The XN-312-GW-CAN gateway can be used to establish a connection to a CANopen field bus.

The connection to the field bus needs to be established by connecting an FMC 1.5/3-ST-3.5 three-pin connector to X1. An internal connection makes it possible to directly connect an additional module to the field bus by connecting it to X2.

The gateway's 24-V POW power supply needs to be connected to X3. Two additional terminals are connected in parallel.

A system bus is used for data communication 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 diagnostic interface on XN-312-GW-CAN gateways can be used to connect them to a computer in order to use the XN-Assist planning and commissioning program. The interface can also be used to update the operating system.

1.3 Device overview

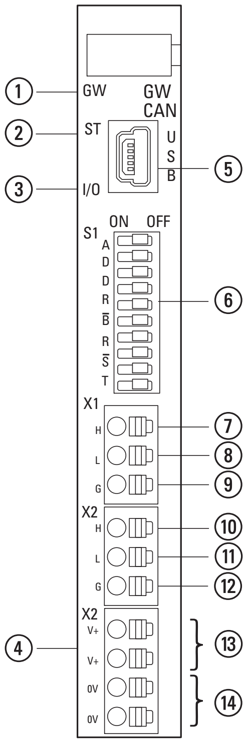


Figure 2: Front view XN-312-GW-CAN

① GW LED, system bus 5 VDC status
② LED ST, status CANopen
③ I/O LED, I/O slice module configuration status
④ LED POW
⑤ Mini-USB diagnostic interface
⑥ DIP switches for configuring the address for the node IDs, firmware update, baud rate, config check, bus termination resistor
⑦ CAN_H
⑧ CAN_L
⑨ GND
⑩ CAN_H
⑪ CAN_L
⑫ GND
⑬ Power supply, 24 V DC POW
⑭ 0 V

1.4 List of I/O slice module products

The "XN300 slice modules", MN050002" manual describes all the I/O slice modules that can be combined into a system block with this gateway in detail. As of this writing, the following XN300 slice modules are available:

Power supply modules

- XN-322-4PS-20
- XN-322-18PD-M
- XN-322-18PD-P

Digital I/O modules

- XN-322-8DI-PD
- XN-322-16DI-PD
- XN-322-20DI-PD
- XN-322-20DI-PF
- XN-322-20DI-PCNT
- XN-322-20DI-ND
- XN-322-8DO-P05
- XN-322-12DO-P17
- XN-322-16DO-P05
- XN-322-8DIO-PD05
- XN-322-16DIO-PD05
- XN-322-16DIO-PC05

Analog I/O modules

- XN-322-4AI-PTNI
- XN-322-7AI-U2PT
- XN-322-8AI-I
- XN-322-10AI-TEKT
- XN-322-8AO-U2
- XN-322-4AIO-U2
- XN-322-8AIO-U2
- XN-322-4AIO-I
- XN-322-8AIO-I

Technology Modules

- XN-322-2DMS-WM
- XN-322-1DCD-B35
- XN-322-1CNT-8DIO
- XN-322-2SSI

Relay modules

- XN-322-4DO-RNO

1.5 Important data for engineering

In connection with the I/O slice modules, the CAN gateway will appear as a modular module on the field bus. This manual focuses on the CAN gateway exclusively.

Gateway	field bus
XN-312-GW-CAN	CANopen, according to DS301.4 profile

Each XN300 module is presented as a separate module, with its data, to the field bus master.

The following is the maximum number of I/O slice modules that can be run with the gateway:

Gateway	Maximum number of I/O slice modules on the system bus
XN-312-GW-CAN	32

Make sure to take the maximum volume of data that can be transmitted on the field bus into account. The corresponding limit will be defined by the field bus.

The following applies to CANopen: 4 TPDOs and 4 RPDOs with 8 bytes of data each will be provided to each field bus module, i.e., a total of 32 bytes in each direction. Additional COB-IDs can be used with an offset of 32, 64, and 96 bytes, meaning that a total of 4 * 32 = 128 bytes of payload data are available in each direction. This means that there are a max. of 16 TPDOs and 16 RPDOs available.

Another 8 TPDOs and 8 RPDOs, with 8 bytes of data each, are available per direction. However, these objects will not be mapped automatically, and need to be mapped manually instead. Accordingly, for each of the PDOs from 17 through 24 that are in use, the user needs to enter a COB-ID that is not yet being used in the system.

This means that the total number of TPDOs and RPDOs available is max. 24 TPDOs and 24 RPDOs.
Accordingly, 128 bytes + 64 bytes = 192 bytes of payload data are available for each direction.

NOTICE

When using these COB-IDs, none of the modules used on the same CAN network should have
Node ID of XN-312 gateway + offset (32/64/96)
as this will result in frame collisions on the bus.

Field bus-slave	Max. input data [Byte]	Max. output data [Byte]
CANopen	128	128

→ For information on the I/O slice modules' input and output data volumes, please refer to the appendix in manual "XN300 slice modules", MN050002".

1.5.1 Field bus connection

There are two field bus connectors on the gateway: X1 and X2. The ST LED on the XN-312 gateway will show the CANopen status for this bus module. The field bus' status will be indicated as specified in CiA303. The connectors are three-pin sockets and are internally connected in parallel as an input and an outgoer. Accordingly, two three-pin plugs are included with each gateway.

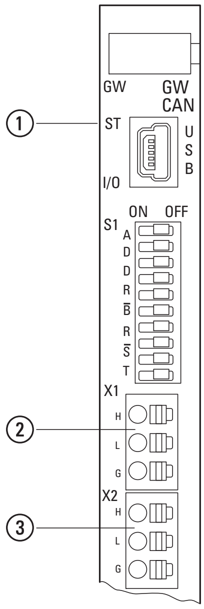


Figure 3: The configuration of the terminal sockets at the gateway

① LED ST red/green
② X1 field bus connection 1
③ X2 field bus connection 2

Table 1: What the signals from the ST diagnostic LED for the CANopen field bus interface mean; for a more detailed explanation of each status Table , page 45.

ST LED	Status	CANopen status	Meaning
Green	Continuous light green	OPERATIONAL	Operation correct
	Flashing	PRE-OPERATIONAL	PRE-OPERATIONAL
	Flashing pattern SINGLE FLASH	STOP	HALT

ST LED	Status	CANopen status	Meaning
Red	Continuous light red	BUS OFF	The field bus has been turned off due to an error, e.g., • Wiring faults • Module with wrong baud rate on bus
	Flashing pattern SINGLE FLASH	:	Warning! Errors have occurred on the field bus
	Flashing pattern DOUBLE FLASH		Error! GUARD or HEARTBEAT has occurred on the field bus Excessive bus load, values for guarding settings set too low, connection problems.
red/ green	Flickering		Baud rate detection active
off	–	CANopen disabled	CANopen stack disabled. The following are possible causes: • XN300-Assist actively running wiring test • Baud rate detection started • Error on field bus

Reminder

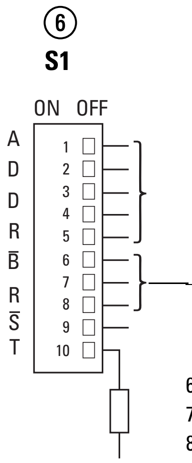
The following states are defined in CiA301: PRE-OPERATIONAL, OPERATIONAL, STOPPED, and INITIALIZED.

Status	Meaning	Explanation
PRE-OPERATIONAL	The device has been started and has checked in on the CANopen bus Outputs are not being written to PDOs are not being written to the master Inputs are not being read The device is being initialized for CANopen	
OPERATIONAL	The device has been initialized and is active Outputs are being written to PDOs are being written to the master Inputs are being read	
STOPPED	Outputs are not being written to Inputs are not being read No monitoring active The device has already been initialized for CANopen, but is passive	
INITIALIZATION	Device booting	As soon as the device's internal initialization is complete, the device will automatically switch to the PRE-OPERATIONAL state.

1.5.2 Baud rate

DIP switches 6 to 8 can be used to set a fixed baud rate for the gateway. If DIP switches 6 to 8 are in the OFF position, the automatic baud rate detection function, as described in CiA801, will be enabled. If the gateway is in the middle of automatically detecting the baud rate, the ST LED will flash red/green and the gateway will adopt the field bus master's baud rate. Baud rates of 100 kBaud and 800 kBaud can only be set using automatic baud rate detection, i.e., they cannot be set using the DIP switches.

Cyclical CAN frames need to be sent on the field bus in order for the baud rate to be detected successfully. Otherwise, the gateway will not be able to detect the CAN baud rate.



	Auto detection	Baud rate [kBaud] 10	20	50	125	250	500	1000
6	OFF	ON	OFF	ON	OFF	ON	OFF	ON
7	OFF	OFF	ON	ON	OFF	OFF	ON	ON
8	OFF	OFF	OFF	OFF	ON	ON	ON	ON

Figure 4: DIP switches for setting the baud rate

→ If the device is unable to automatically determine the baud rate, use the DIP switches to set a fixed baud rate or add another CAN module to the system.

1.5.3 Valid device field bus addresses

DIP switches 1 to 5 can be used to set the gateway's address. If all the DIP switches are in the OFF position, the gateway will start in boot loader mode. When the gateway is in this mode, XN300-Assist can be used to update the gateway's firmware using the USB port. For more details on this, please refer to the "Updating the operating system" section in the online help for XN300-Assist.

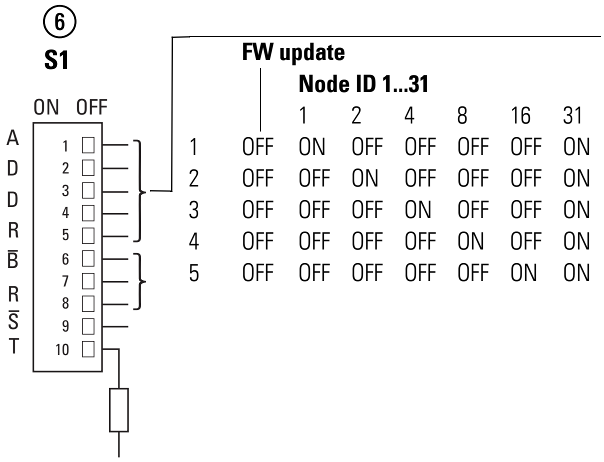


Figure 5: DIP switches for setting the gateway's node ID

Data bus	Address	Address setting with dip switch	Valid address range
CANopen	Node ID	1 - 5	1 - 31

1.5.4 Enabling and disabling the config check function

If DIP switch 9 is in the "OFF" position, a configuration comparison/config check will not be carried out. Instead, the gateway will take the actual configuration on the system bus and store it as the target configuration in its device memory. The I/O LED will flash green.

"Actual configuration" refers to the current configuration of the I/O slice modules on the system bus.

This means that the target configuration will be the I/O slice module configuration that the user has defined on the system bus at the time the system is turned on with DIP switch 9 = "OFF."

If DIP switch 9 is in the "ON" position instead, the gateway will check whether the current actual configuration on the system bus matches the target configuration.

If they match, the I/O LED will display a continuous green light and the gateway will be ready for operation. If the configuration list and the detected modules do not match, the I/O LED will display a continuous red light. In this case, it will not be possible to switch the gateway to the OPERATIONAL state, and the gateway will not start.

For more information, please refer to → Figure 26, page 40.

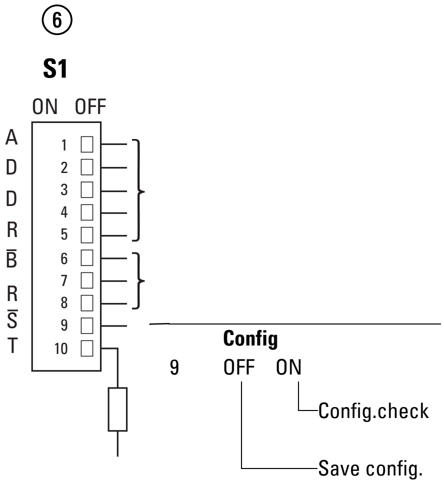


Figure 6: DIP switch for enabling and disabling the config check function

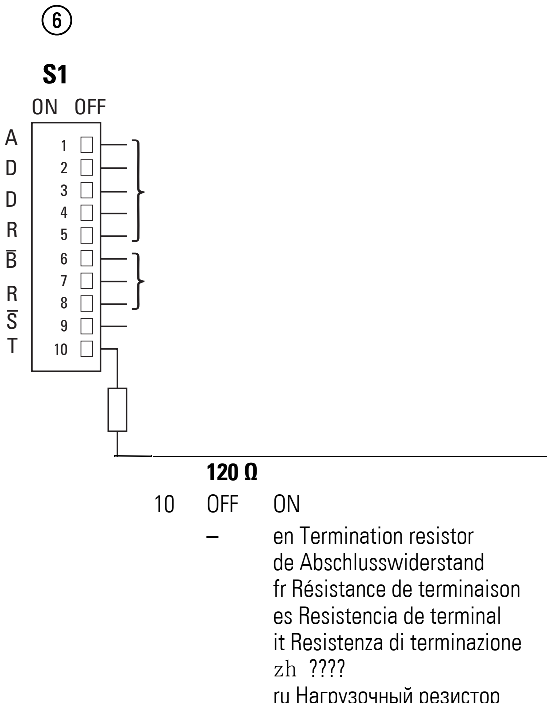


Figure 7: DIP switch for activating the bus termination resistor

2 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 gateway is installed in the following order:

- Adjust the field bus address at the device.
- Set the Baud rate on the device.
- If the gateway is the first or last module on the field bus, use the DIP switches to activate the bus termination resistor in the device.
- 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.

2.1 Setting the gateway's field bus address

Before mounting the device, set the field bus address using the DIP switches (DIP = dual in-line package) at the front of the gateway.

NOTICE

Changes made to the dip switches' settings will take effect only after the power supply is turned off and back on again.

Follow the example below in order to configure an address on the gateway - 18 (decimal) is used as an example in this case:

$18_{dec} = 16 + 2 = 1 \cdot 2^4 + 1 \cdot 2^1$

The address is to be set at the dip switches.

▶ Set the CANopen address using DIP switches 1 - 5; refer to → Section "1.5.3 Valid device field bus addresses", page 23. DIP switches 1 and 4 must be switched to ON in the example below.

DIP Switch					
1	2	3	4	5	Description
2^0	2^1	2^2	2^3	2^4	
OFF	ON	OFF	OFF	ON	Setting the gateway's field bus address to a value of 18_{dec}

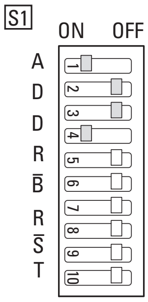


Figure 8: Example: Setting field bus address 18_{dec} on the gateway

2.2 Setting the Baud rate

DIP switches 6 to 8 can be used to set a fixed baud rate for the gateway → Section “1.5.2 Baud rate”, page 22. If you want to use the automatic baud rate detection function (as described in CiA 801) instead, follow the steps below:

▶ Set DIP switches 6 to 8 to OFF.

When the gateway is attempting to detect the baud rate, the ST LED will flash red and green and the gateway will adopt the field bus master's baud rate.

Cyclical CAN frames need to be sent on the field bus in order for the baud rate to be detected successfully. Otherwise, the gateway will not be able to detect the CAN baud rate.

→ If the device is unable to automatically determine the baud rate, use the DIP switches to set a fixed baud rate or add another CAN module to the system.

2.3 Activate the bus termination resistor for CANopen

If the XN-312-GW-CAN gateway is the last module on the field bus, the integrated bus termination resistor must be activated using DIP switch 10.

▶ Set DIP switch 10 to the "ON" position.

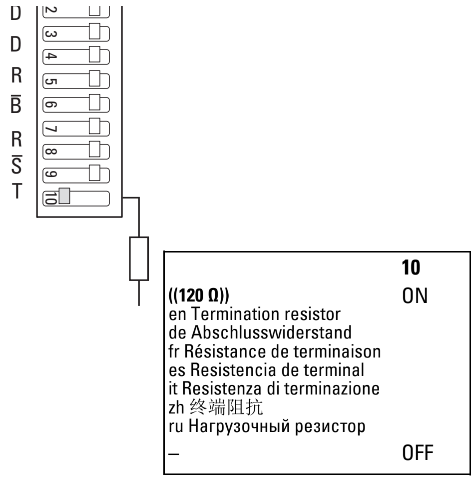


Figure 9: Activated bus termination resistor

2.4 mounting

2.4.1 Installation prerequisites

Install the XN-312-GW-CAN gateway in a switch cabinet, 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.

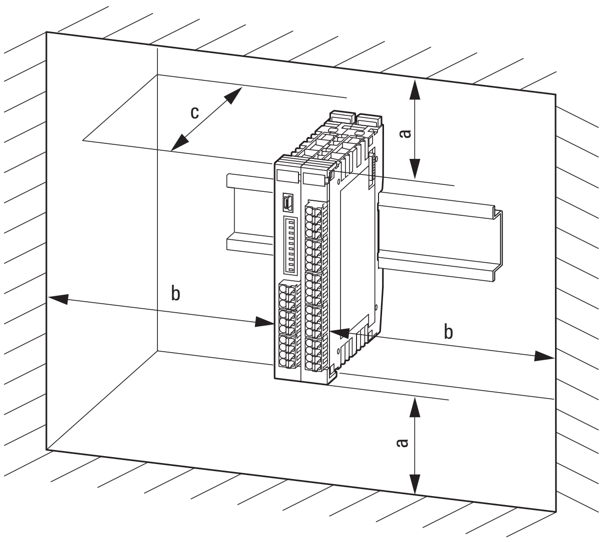


Figure 10: Horizontal installation only!

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

2.4.2 Mounting the system block on the DIN-rail

Before mounting the device, make sure that you have configured the device's field bus address and, if necessary, that the bus termination resistor (if any) has been activated with the corresponding DIP switch.

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.
- Open the locking tabs on the sides of the XN300 slice modules by pulling on the front cover (blue). Make sure that all locking tabs (blue) are on the front so that they will engage the next slice module. The front cover's stay-put function is meant to help with this.

→ 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.

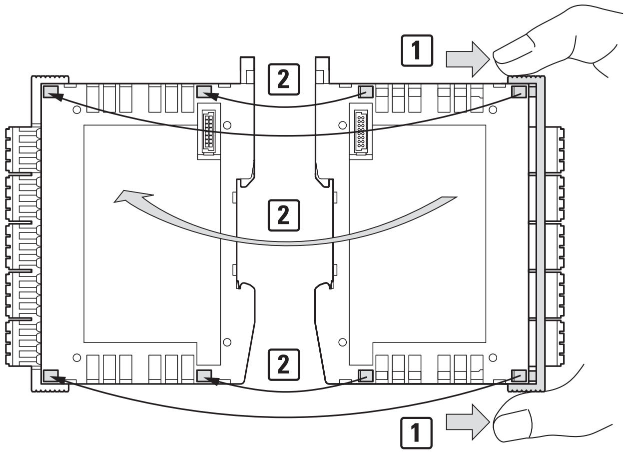


Figure 11: Joining the gateway and an XN300 slice module to form a system block

- 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.

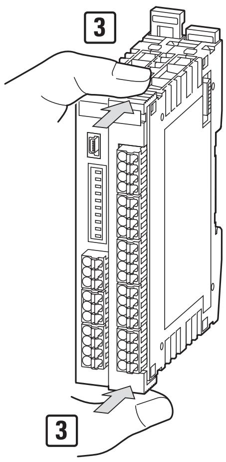


Figure 12: Locking the system block in place

- 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,

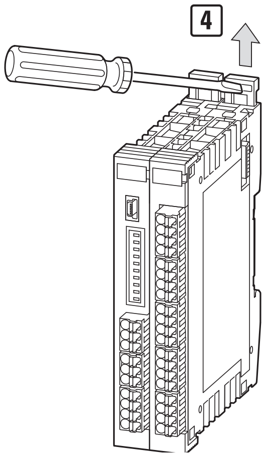


Figure 13: Securing the system block on the DIN-rail

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

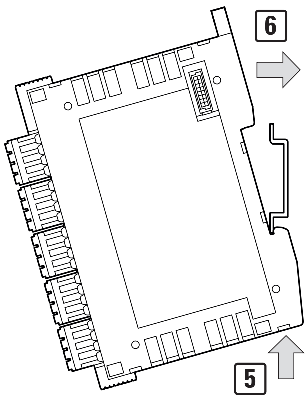


Figure 14: Placing the system block against the bottom edge of the DIN-rail

- 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.

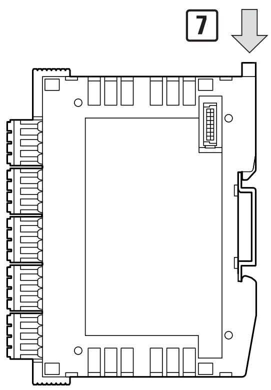


Figure 15: Locking the system block into place on the DIN-rail

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

2.5 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.

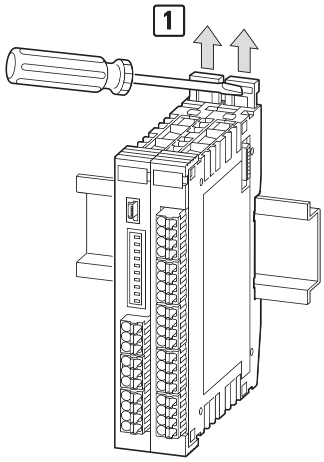


Figure 16: Disengaging the system block

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

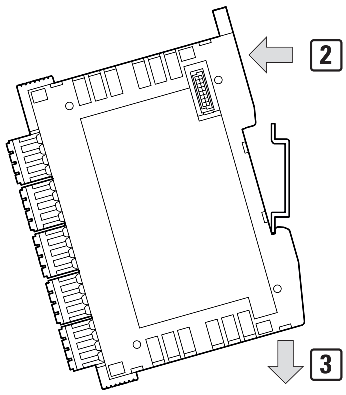


Figure 17: Placing the system block against the bottom edge of the DIN-rail

▶ 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.

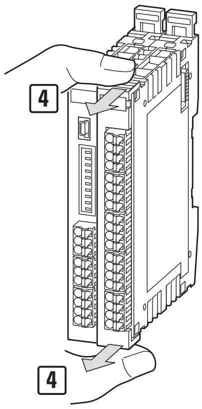


Figure 18: Disengaging the front cover

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

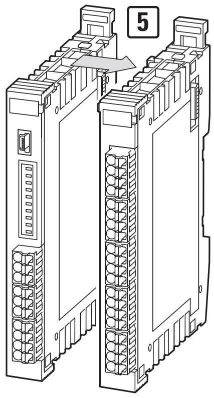


Figure 19: Separating the gateway and the XN300 slice modules from the system block

2.6 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 CANopen field bus and the XN300 system are galvanically isolated from each other.

Common

- 0V
- ⏚

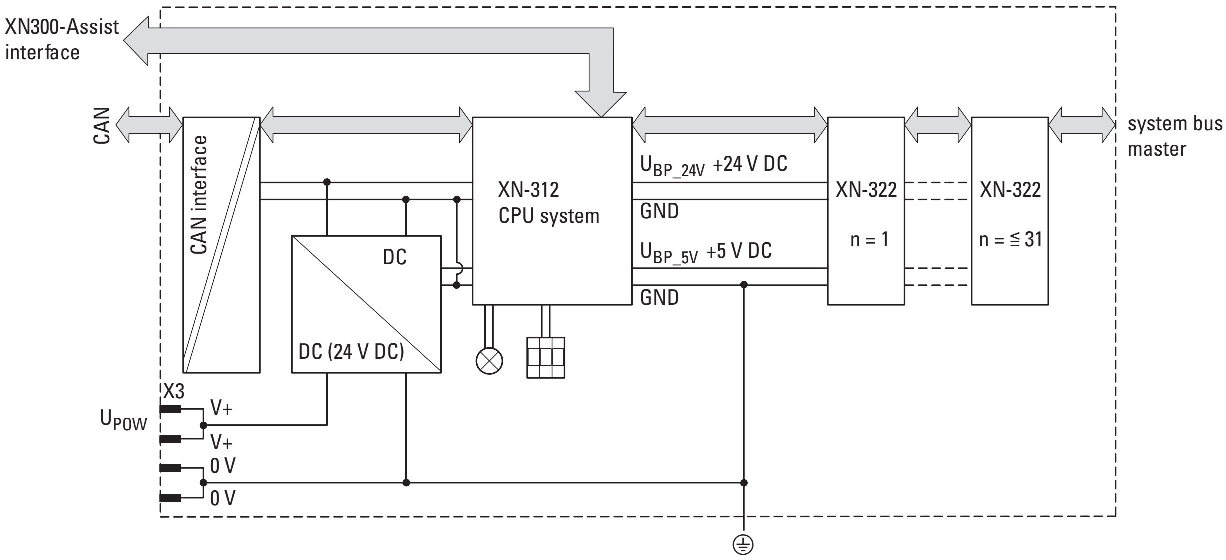


Figure 20: Functional diagram of XN300 system

2.7 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 XN-322-4PS-20 power supply modules. You can use the XN300-Assist software program for assistance with the corresponding calculations. In addition, the program will automatically let you know if additional power supply modules are required for your design.

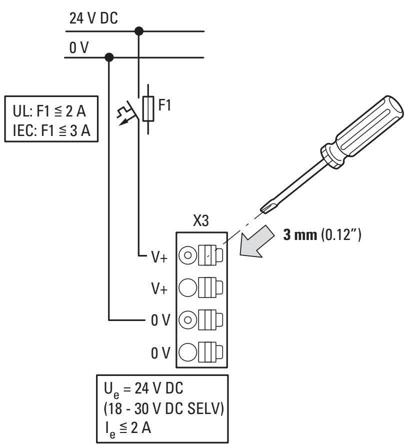


Figure 21: Connection of power supply

▶ 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

2.8 Connect field bus

CANopen

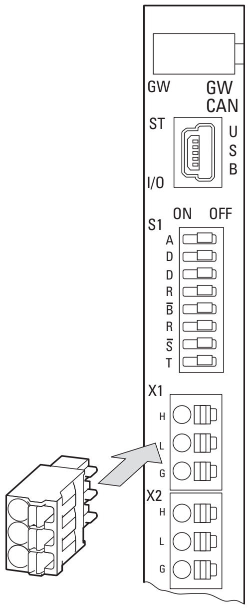


Figure 22: Connection of CANopen

The following CAN bus cable is recommended in line with the requirements in ISO 11898:

- UNITRONIC bus LD, Messrs. LAPPKABEL
  - 2 x 2 x 0.22 mm²
  - Surge impedance: 100 – 120 Ohm
  - Effective capacitance: 800 Hz, max. 60 nF/km

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

▶ Insert the CAN bus cable's wires into the push-in terminals on the three-pin FMC 1.5/3-ST-3.5 contact plug.
▶ Connect the contact plug to the X1 field bus interface on the gateway.
▶ You can use the X2 field bus interface on the gateway to connect additional field bus modules.

2.8.1 Maximum cable length

The maximum cable length for a CAN bus cable depends on the baud rate being used. The following table provides an overview of the possible Baud rates and the corresponding maximum cable lengths:

Table 2: Max. cable lengths

Baud rate (kBaud)	Maximum cable length (m)
10	5000
20	2500
50	1000

Baud rate (kBaud)	Maximum cable length (m)
100	650
125	500
250	250
500	100
800	50
1000	25

It may be necessary to use repeaters when using cable lengths of 1,000 m or more.

2.9 Connect 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 program to run real-time diagnostics on the system bus
- Updating the operating system

You can use any of the following programming cables:

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

2.10 Connection example

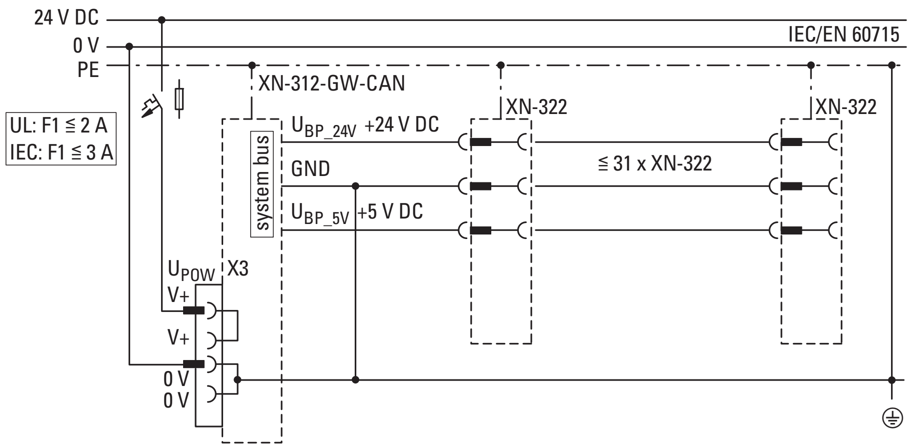


Figure 23: Connecting example for XN-312-GW-CAN gateway in XN300 system

2.11 Wiring in accordance with EMC requirements

Undesired faults can occur on the field bus 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

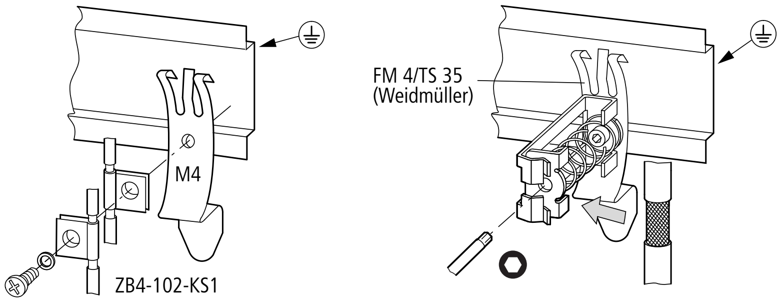


Figure 24: Field bus shielded by using a shield

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

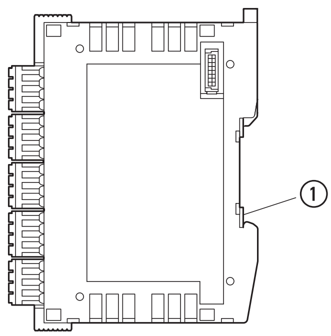


Figure 25: XN-312-GW-CAN side view

(1) Functional earth

3 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., motors start up unexpectedly.

Startup takes place in several steps:

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

3.1 Take system bus into operation

DANGER

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

When you replace XN300 slice modules without switching off the power supply, the XN-312-GW-CAN gateway will carry out a software-controlled RESET.

The system bus can be placed into operation with or without a configuration check. Accordingly, decide whether the configuration check function on the gateway should be enabled or not (it will be disabled by default).

If the type or number of connected modules changes and the configuration check function is enabled, you will need to disable the configuration check function and start the gateway with the function disabled. When you do so, the gateway will read the modified target configuration, after which you can re-enable the configuration check function.

As an alternative to commissioning the station using the configuration check based on the DIP switches on the device, you can have the PLC run the configuration check instead. This option is described in detail in the following chapter → Chapter 8 “Station variants”, page 355.

3.1.1 Switching the gateway on with or without a config check

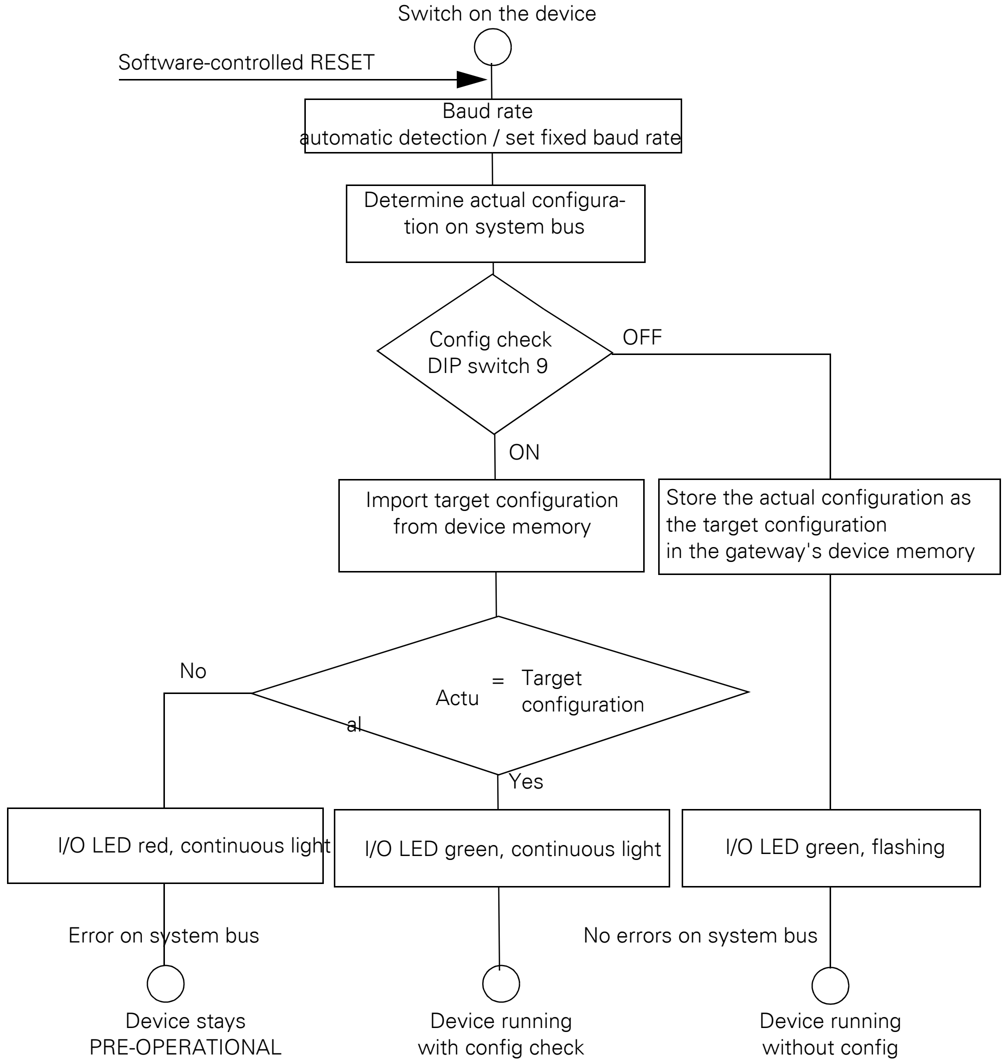


Figure 26: Switching on the device with and without a config check

Switching the gateway on without a config check

If you want to put the gateway into operation without a config check, make sure that the config check function is disabled:

▶ DIP switch 9 must be in the "OFF" position.
▶ Switch on the power supply.

The gateway will determine which I/O slice modules are currently on the system bus and will store that configuration as the target configuration in its device memory.

The I/O LED will flash green to indicate that the config check is disabled.

Switching the gateway on with a config check

If you want to put the gateway into operation with a config check, follow the steps below:

▶ Make sure that all the I/O slice modules in the target configuration are actually present on the system bus.

Storing the target configuration in the gateway's device memory

The target configuration needs to be imported the first time the device is commissioned, whenever a module is replaced, and whenever the module layout on the system bus changes.

Requirements that must be met in order to import the target configuration:

- All slice modules must be locked in place together with the gateway in the form of a system block and must be connected to the system bus.
- Voltage POW is applied to the gateway and the POW LED lights up.
- The planned power supply modules must be connected.
- The status LEDs on the I/O slice modules must be on or must be flashing.

To import the target configuration, follow the steps below:

▶ Set DIP switch 9 to the "OFF" position.
▶ Switch on the device's power supply.

The gateway will determine which I/O slice modules are currently on the system bus and will store that configuration as the target configuration in its device memory. The I/O LED will flash green to indicate that the device is running without a config check.

▶ Set DIP switch 9 to the "ON" position.
▶ In order for the switch position change to be applied, switch the device's power supply off and then back on.

When the device is switched on, the workflow in → Figure 26, page 40 will be followed. The green I/O led will show a continuous light to indicate that the device is running with the config check function on.

→ The position of DIP switch 9 will be read only when the device is powered on.

If DIP switch 9 is switched to the OFF position during operation, it will be necessary to switch the device's power supply off and then back on.

→ If the I/O LED shows a continuous red light, this means that no addresses have been assigned to any of the XN300 modules. Check the system bus and check the system block to make sure that all I/O slice modules are properly locked in place with each other.

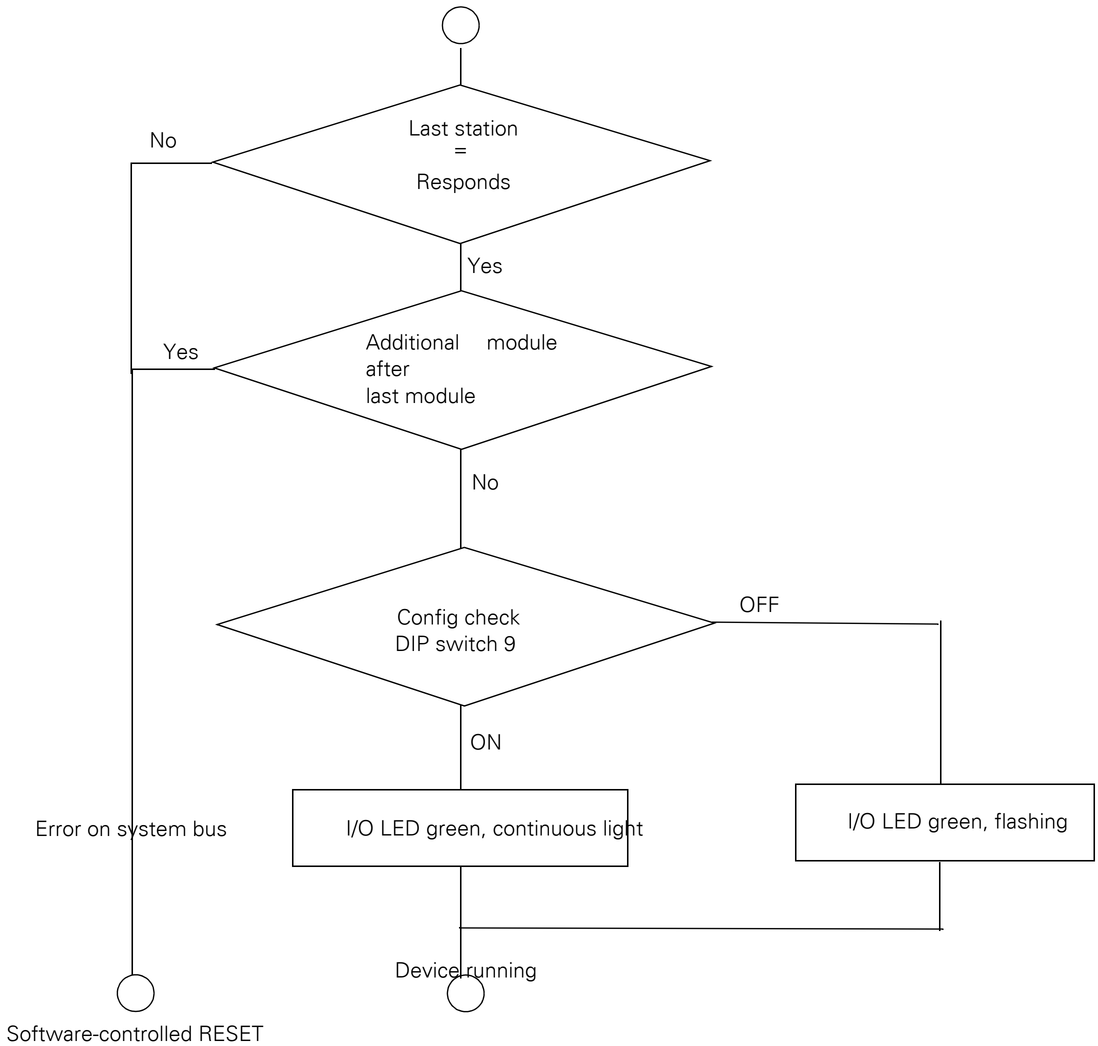
3.1.1.1 In operation

Regardless of the config check setting, the gateway will check whether the actual configuration changes during operation.

If there are any configuration changes during operation, the gateway will be reset.

One possible cause of a configuration change is when a part of the system block becomes disengaged on purpose or by accident. Another possible cause is a module that has dropped out from the system bus.

Depending on whether the config check function is enabled or disabled, the gateway will either start with the modified configuration (config check disabled) or will stay in safe mode (config check enabled – check failed).



→ Figure 23

Figure 27: Device operation with and without config check

NOTICE

If the gateway is run with the config check function disabled, only the modules up to the device that has dropped out will be assigned an address and stored. It will not be possible to use the remaining modules until the defective module is replaced and the configuration is imported again.

Import the target configuration in the following cases:

- Initial Commissioning
- Replacement of a defective module
- When there is a change in the I/O slice module layout in the system block

The online and real-time functions featured by the XN300-Assist program offer a wide range of options for displaying and diagnosing the system bus, even without an active PLC → Section “3.5 XN300-Assist”, page 48.

3.1.1.2 Switching on when the target configuration is stored

If a target configuration is already stored on the gateway, DIP switch 9 can be switched to the "ON" position while the gateway is being powered on. The gateway will then check the connected modules when booting up. Otherwise, the current actual configuration will be used to overwrite the previously stored target configuration → Section “3.3 LEDs on the device”, page 45.

3.2 Take the CANopen field bus into operation

If you are putting the field bus into operation for the first time, you will first have to set up the PLC programming environment
→ Chapter 5 “Connecting the PLC to the gateway using CODESYS”, page 53.

Generally speaking, the following steps must be carried out when commissioning the gateway for the first time:

1. Open or create the project in the PLC programming environment.
2. Install the appropriate EDS files in the PLC programming tool.
3. Select the field bus master in the PLC programming environment (target system) and set a Baud rate, e.g., 250 kBaud.
4. Select the gateway as a field bus module
5. Select the XN300 slice modules; this step is skipped when using a project-specific EDS file.
6. Configure the gateway's and the XN300 modules' parameters in the PLC programming tool.
   The I/O slice modules' parameters will be stored in the object dictionary.

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

- If default mapping (index 0x6401/x) is used for analog modules, PDO-based data transmission for these I/O slice modules must be enabled by setting the value of object index 0x6423, subindex 0 to TRUE → Section “7.15.3 Analog Input Global Interrupt Enable (Object 0x6423)”, page 187.
- Regulate the bus load on the CAN field bus
  The following procedure is recommended in order to prevent the XN-312-GW-CAN gateway from generating a heavy load on the CAN field bus:
  - Use internal module filters; please consult manual "XN300 slice modules", MN050002".
  - If using the default mapping (index 0x6401/x), a transmission filter can be configured for each individual payload data object (SDO objects: AI_DELTA_VALUE, index 0x6426/x).
    When this filter is configured, the gateway will update and transmit the corresponding value only if this value is different from the last transmitted value by more than the configured delta → Section “7.14.6 Analog Input Interrupt Delta Unsigned (Object 0x6426)”, page 170.
  - By setting an inhibit time or selecting a synchronous transmission type, TPDO properties can be configured in such a way as to avoid impermissible bus loads → Section “6.8.1 Objects 1800hex to 180Fhex Transmit PDO Parameter”, page 96.

3.2.1 Creating field bus communication CANopen

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

The green ST 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 or the baud rate is changed on the gateway, the change will not take effect for the communication on the field bus until after the gateway is switched off and then back on.

3.2.2 POW status display

Description	**POW**
Device ready for operation	on
If the other three gateway LEDs show a continuous red light, this means that the firmware or hardware is faulty. If they show continuous orange light, the gateway is in Firmware Update mode.	off

In operation

If an I/O 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.

3.3 LEDs on the device

Explanation of LED statuses

The following table explains, in detail, how the LEDs will behave for each of the statuses listed.

LED status	Description
off	LED shows no light at all
To	LED shows a continuous light
Flickering	Cyclical flashing (10 Hz), LED on for 50 ms / off for 50 ms
Flashing	Cyclical flashing (2.5 Hz), LED on for 200 ms / off for 200 ms
Flashing pattern: SINGLE FLASH	Cyclical single flash, LED on for 200 ms / off for 1,000 ms!
Flashing pattern: DOUBLE FLASH	Cyclical double flash, LED on for 200 ms / off for 200 ms / on for 200 ms / off for 1,000 ms

GW LED

GW LED	Status	Meaning
Green	Continuous light green	Firmware running properly
off	–	Firmware error or boot loader mode active

I/O LED

I/O LED	Status	Meaning
Green	Continuous light green	Actual configuration = Target configuration
	Flashing	Config check disabled
Red	Continuous light red	Actual configuration ≠ Target configuration
off	–	Configuration has not been checked yet.

ST LED

ST LED	Status	Designation	Meaning
Green	Continuous light green	OPERATIONAL	Process data being transferred
	Flashing	PRE-OPERATIONAL	PRE-OPERATIONAL
	Flashing pattern: SINGLE FLASH	STOPPED	HALT
Red	Continuous light red	BUS OFF	The field bus has been turned off due to an error, e.g., • Wiring faults, • - Module with wrong baud rate on bus
	Flashing pattern: SINGLE FLASH	ERROR WARNING	Warning! • Wiring faults, • Module with wrong baud rate on bus
	Flashing pattern: DOUBLE FLASH	Guarding error	Depending on the relevant setting (OD 0x1029), the gateway will internally switch to the PRE-OPERATIONAL state (default), to the STOPPED state, or will remain in its current state.
red/green	Flickering	INITIALISATION	Automatic baud rate detection in progress
off	–	CANopen disabled	The field bus has been deactivated. Possible causes include: • XN300-Assist wiring test active • A project is currently being loaded onto the device

POW-LED

POW-LED	Meaning
Green	5 V system bus power working properly
off	No power on system bus

LEDs on I/O slice modules

The I/O modules can come with various LEDs, with the specific LEDs depending on the modules' functionality. Regardless of this, all I/O slice modules will feature a POW LED and a User LED, with the exception of power supply modules.

POW-LED	Status	Meaning
Green	Continuous light green	I/O slice module active
	Green flashing	Fault in communication with gateway, I/O slice module not connected.
	off	No power on system bus

USER LED	Status	Meaning
Yellow	off	Application specific
	Flashing pattern: SINGLE FLASH	Application specific
	Flashing pattern: SINGLE FLASH, INVERTED	Application specific
	To	Application specific

3.4 Note on alarms

The module does not send any alarms. The diagnostic data must be transmitted using PDO or SDO access.

3.5 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
- Generating a project-specific EDS file
- Status indication of the inputs/outputs
- Wiring test
- Importing actual configurations
- 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.

4 Description files for CANopen

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

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

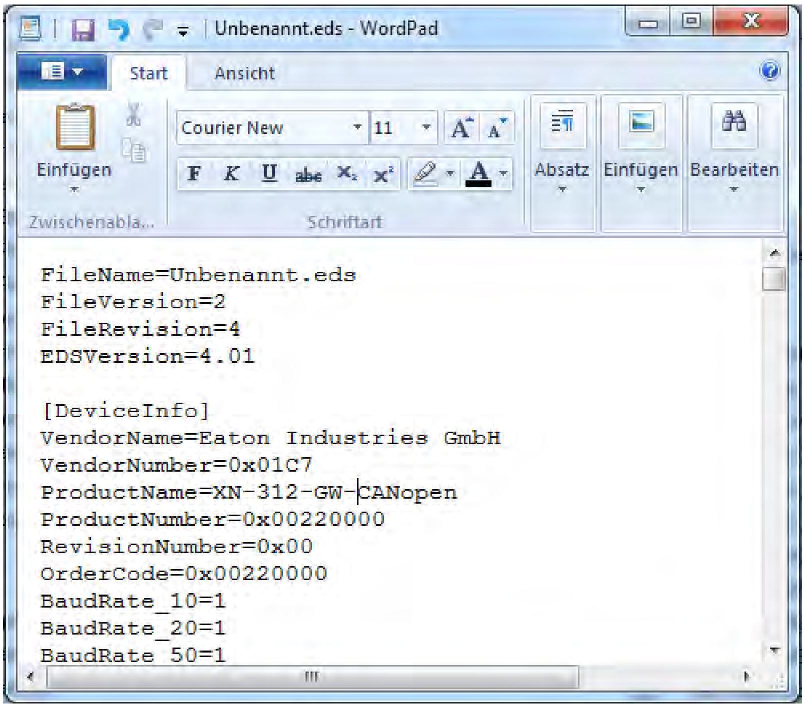


Figure 28: Header of an EDS file for XN-312-GW-CAN

In order to be able to run the gateway on a CANopen field bus, the SWD gateway's properties must be stored in a description file. This file follows specific standards for the corresponding field bus and can be loaded into the PLC programming environment for the field bus master being used.

The EDS file contains all the properties of a gateway in its role as a field bus module.

This means that new EDS file versions need to be created whenever new I/O slice modules are released.

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

Make sure that you are working with the latest version of the standard EDS files in your PLC programming environment. Check whether there are available updates for the files at our Download Center → Page 11.

Updating the operating system

Eaton may make new operating systems available in order to fix firmware errors in the CAN gateway and/or provide new functionalities, Download center → Page 11.

In the case of XN-312-GW-CAN, the operating system needs to be updated via the diagnostic interface using XN300-Assist.

For information on how to update the operating system, please refer to the online help for XN300-Assist. Open the "Search" tab and enter "Update operating system" into the search box.

4.1 Standard EDS files

There might be multiple versions of each EDS file for the CANopen gateway.

The version number of the EDS file being used must match the version number of the operating system loaded onto the gateway. This is the only way to ensure that the range of functions in the EDS file and in the XN300 gateway will match.

Within each version, the EDS file with the highest revision number must be selected, as this is the only file that will contain the description for all currently available
XN300 system I/O slice modules.

The first EDS file for the XN-312-GW-CAN CANopen gateway is named "XN-312-GW-CAN_V0102.eds".

To download the standard EDS file, please visit our Download Center → Page 11. To find it quickly, simply enter "XN300" or "EDS" as a search term.

4.2 Project-specific EDS file

In addition to the standard EDS file, you can also use XN300-Assist to generate a project-specific EDS file. To do this, you can either generate a project-specific EDS file directly from the modular project view or read the XN300 slice modules online. The selection of XN300 slice modules on the system bus will already be assigned to the XN-312-GW-CAN field bus module.

If you load the project-specific EDS file into the PLC programming tool, you will not need to add I/O slice modules to the configuration → Section “5.2.3.2 Configuring the XN-322 slice modules”, page 57.

Use the option of using a project-specific EDS file in the following cases:

- The CANopen configurator you are using has problems processing a modular EDS file.
- You want to avoid selecting I/O slice modules by accident later on.

Follow the steps below in order to generate a project-specific EDS file:

▶ Start XN300-Assist.

- ▶ Switch to the Communication view by clicking on the "View | Communication View" menu option.
- ▶ Click the "Online" button.

The program will show the actual configuration, but this configuration will not be in the Project view yet.

- ▶ Click on the button "Device =>PC".

The actual configuration will be loaded into the Project view.

- ▶ Click on the “Offline” button.
- ▶ Switch to the Project view by clicking on the "View | Communication View" menu option.
- ▶ Click on the menu option "Project | Export EDS file".
- ▶ Specify the project folder as the storage location and confirm with "Save."

XN300 assist stores the EDS file under the name of the current project with the appropriate extension, such as "XN300_Project.eds". This can be installed in XSOFT-CODESYS-2 in the same way as the standard EDS file.

→ You will not be able to modify project-specific EDS files with the configuration tool. The changes must be made exclusively with XN300-Assist.

4.3 Installing the EDS file

The procedure for installing the EDS file you want to use will depend on the configurator being used. Following are examples using XSOFT-CODESYS-2 and XSOFT-CODESYS-3.

4.3.1 XSOFT-CODESYS-2

To install the EDS file, follow the steps below:

- ▶ Select and download the latest version of the EDS file.
- ▶ Save and unzip the *.zip file to a project folder of your choice.

Then use one of the following two methods:

Method 1

- ▶ Move the EDS file to the folder that your programming system uses for description files.
  In the case of XSOFT-CODESYS-2, this folder will be the configuration directory:
  <PROGRAM>\Common Files\CAA-Targets\Eaton Automation\V2.3.9 SP<SERVICEPACK>\PLCConf, where
  - <PROGRAM> = Program folder (e.g.: "C:\Program Files (x86)")
  - <SERVCEPACK> = ServicePack being used (e.g.: "3").
- ▶ After adding a new EDS file, save the project and load it again.

Method 2

- Alternatively, move the EDS file to a folder of your choice.
- Then, in XSOFT-CODESYS-2, select the <Project | Options | Category: Directories> menu option.
- Enter the path to the EDS file into the "Configuration files" field.
- Confirm by clicking on "OK."
- After adding a new configuration path, save the project and load it again.

4.3.2 XSOFT-CODESYS-3

You will need to import the EDS file into the programming system you are using. To install the EDS file, follow the steps below:

- Select and download the latest version of the EDS file.
- Save and unzip the *.zip file to a project folder of your choice.
- In XSOFT-CODESYS-3, select the <Tools | Install device …> menu option.
- Check the filter selection to the right of the "Filename" field. The selection must say "EDS and DCF files."
- Select the EDS file in the project folder.
- Click on "OK" on the "The device has been successfully installed" prompt.

5 Connecting the PLC to the gateway using CODESYS

You can either use XSOFT-CODESYS-2 or XSOFT-CODESYS-3 as the configuration software for the gateway.

The gateway can be connected to any CAN master by using the appropriate EDS file.

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

5.1 Connecting the programming computer, the PLC, and the CAN XN300 station

Before commissioning a Eaton PLC (e.g., XC202, XC201, EC4P, XV-152, XV300, XC-152) with the configuration software, the PLC will need to be connected to a programming computer. To do this, the Eaton PLC first needs to be connected to the XN-312-GW-CAN gateway using a CAN bus cable.

5.2 Configuring the system with XSOFT-CODESYS-2

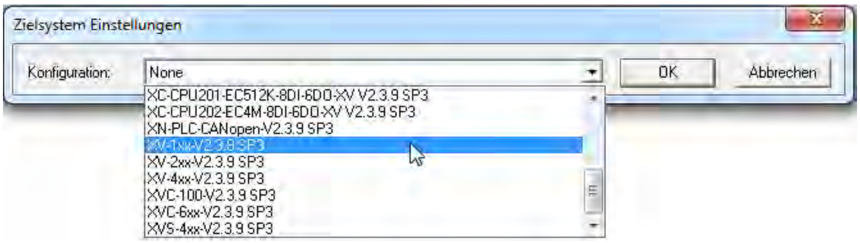
The information below uses the XSOFT-CODESYS-2 Version 2.3.9 SP5 configuration program and the XN-312-GW-CAN_V0102.eds EDS file. Moreover, the example below uses the following devices:

- CANopen master XV-152
- CANopen slave XN-312-GW-CAN
- XN300 slice modules:
  - XN-322-20DI-PCNT
  - XN-322-20DI-PCNT
  - XN-322-10AI-TEKT

5.2.1 Starting XSOFT-CODESYS-2 and creating a new project

▶ Start XSOFT-CODESYS-2 and open a new project by clicking on < File I New >.

▶ The "Target Settings" dialog box will appear. Select the PLC model being used in your application from the options in the drop-down menu.



▶ Confirm the PLC information being displayed by clicking on "OK."
  - The "New POU" dialog box will appear.

▶ Click on "OK" to exit the "New POU" dialog box so that you can finish configuring the XV-152.

5.2.2 Adding the CAN master

Follow the steps below to enable communication via CANopen:

▶ To configure the PLC, start by opening the "Resources" tab.

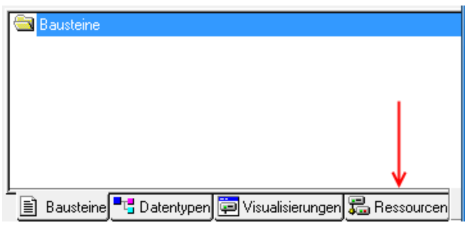


Figure 29: Selecting the configuration mode

▶ Select the "PLC Configuration" node in the configuration structure.

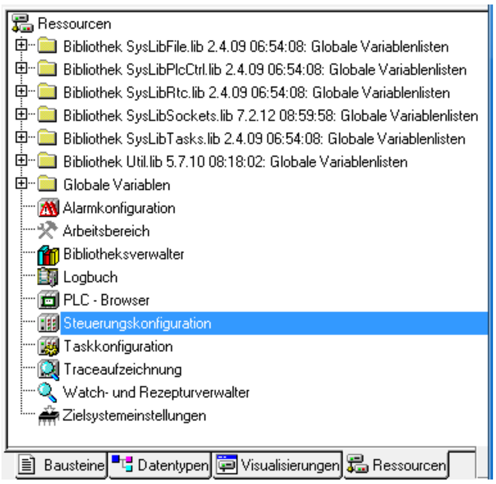


Figure 30: Selecting the PLC Configuration node

- ▶ Double-click on the "PLC Configuration" node.
- ▶ Then click on:
  Insert menu – <Append Subelement I CanMaster...>.
  - The appropriate CAN master will be added to the PLC configuration.
  - The pane on the right will show the tabs used to configure the CAN master.
- ▶ Open the "CAN parameters" tab and check the settings there.

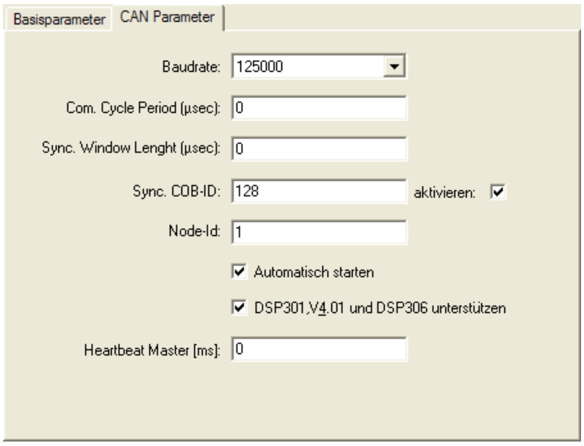


Figure 31: CAN parameters for the CAN master

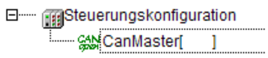
- The master's baud rate must match the slave's baud rate (the slave in this case is the XN-312-GW-CAN gateway).
- The maximum bus length must be used as the basis for determining the maximum transfer rate (→ Section " Maximum cable length (m)", page 36).
- The transfer rate for the XN-312-GW-CAN gateway needs to be set using the corresponding DIP switches (→ Section "1.5.2 Baud rate").
- The three input fields that follow, i.e., "Com. Cycle Period [µsec]," "Sync. Window Length [µsec]," and "Sync. COB-ID," can be used to activate a synchronization message.

– The master's "Node-Id" must fall within a range of 1 to 127 and must only be assigned once on the entire network, i.e., it must be a unique node ID.
– When enabled, the "Automatic startup" function ensures that the CAN bus will be initialized and started. If the options is not enabled, the CAN bus will need to be started from inside the project.
– When enabled, the "Support DSP301,V4.01 and DSP306" function makes it possible to adjust the cycle time for the heartbeat function, among other things.

→ For detailed information, please refer to the "XSoft-CoDeSys-2 PLC Programming" user manual or to the online help for CODESYS, which can be opened by pressing the <F1> function key.

5.2.3 Adding the CAN slaves

▶ To add the CAN slave (XN-312-GW-CAN in this case) to your configuration, right-click on the "CanMaster[xxx]" node.



▶ Then select the <Append Subelement> option.
- A list with all the slaves that can be configured will be shown.

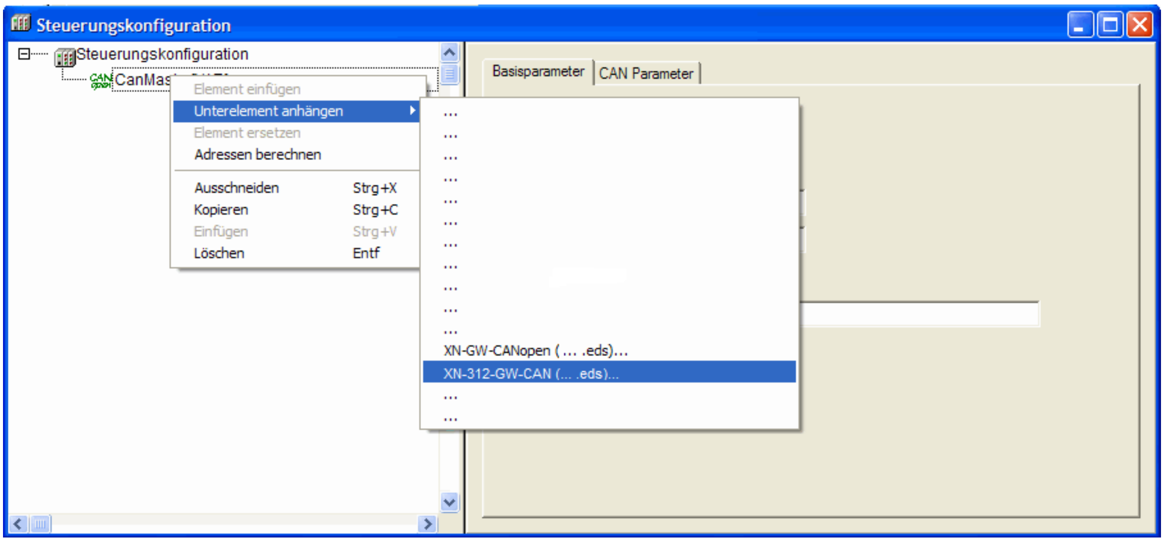


Figure 32: Appending a CAN slave

▶ Select the appropriate CAN slave.
- The CAN slave you selected will be added to the PLC configuration.

▶ If the CAN slaves you want are not found in the list, update the X-SOFT-CODESYS-2 version you are using or load the appropriate EDS file → Section "4.3 Installing the EDS file", page 51.
- You can find updates and EDS files by visiting the Download Center on our website → Page 11.

5.2.3.1 Configuring the XN312 gateway

The pane on the right shows the tabs used to configure the CAN slave.

- ▶ Open the "CAN parameters" tab and check the settings there.
  - The slave's "Node ID":
    – Needs to be set using the DIP switches on the device
    – must be in the range 1 to 31.
    – And must only be assigned once on the entire network, i.e., it must be a unique node ID

→ Make sure that the node ID set with the DIP switches matches the node ID shown under the "CAN parameters" tab!
Change the node ID if necessary!

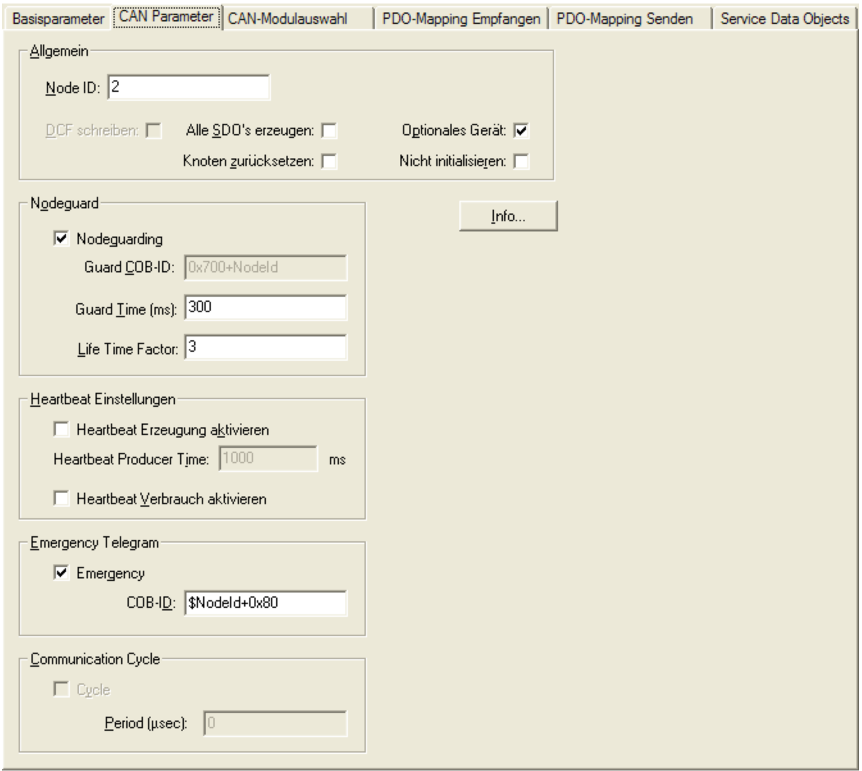


Figure 33: CAN parameters for the CAN slave (XN-312-GW-CAN in this case)

→ For detailed information, please refer to the "XSoft-CoDeSys-2 PLC Programming" user manual or to the online help for CODESYS, which can be opened by pressing the <F1> function key.

5.2.3.2 Configuring the XN-322 slice modules

- ▶ Open the "CAN Module Selection" tab in order to configure the station.

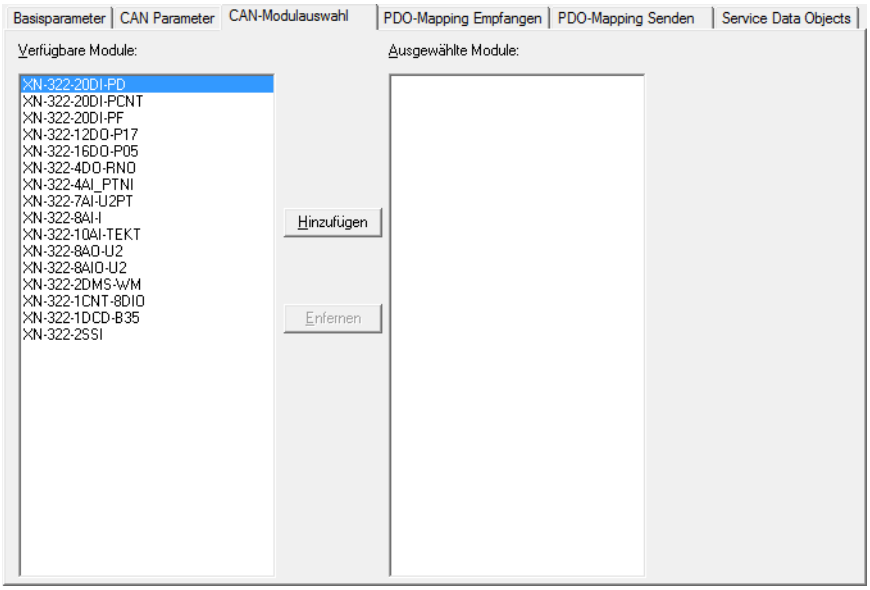


Figure 34: CAN Module Selection tab for CAN slave (XN-312-GW-CAN in this case)

- This tab will show two panes:
  - The pane on the left will show a list with the various module models available.
  - The pane on the right will show a list with the module models in your station.

▶ To configure an XN-322 slice module, select the module model on the left pane and then click on:
<Add>.

→ When adding additional XN300 slice modules, you can choose where you want to place them in the list:
Simply select the module model in the left pane and the position you want in the right pane.
Click on <Add>.
The selected XN300 slice module will be placed directly **above** the module you selected in the right pane.

Disabled process data objects

Certain process data objects are not enabled automatically. These objects need to be manually enabled in order for them to be included in process data transfers (→ Chapter 5 “Connecting the PLC to the gateway using CODESYS”, page 59).

Every time a change is made to the configuration, a prompt indicating that there are disabled process data objects will be shown.

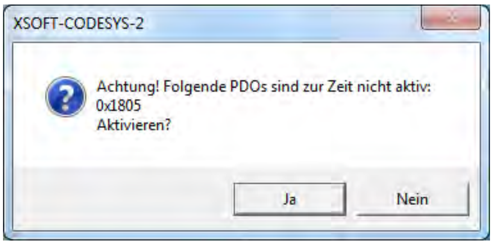


Figure 35: Prompt indicating that there are disabled PDOs

For the default mapping, the gateway provides 8 PDOs for digital data transfers and 8 PDOs for analog data transfers. In other words, 8 x 8 bytes or 8 x 4 words can be mapped automatically. This applies both to the receive and transmit directions!

- A maximum of **64** bytes for digital input channels
- A maximum of **64** bytes for digital output channels
- A maximum of **32** words for analog input channels
- A maximum of **32** words for analog output channels
- Specialty modules not based on the CiA401 profile are not included in this default mapping → "Specialty modules" section, page 16.

If the XN300 station has more input or output channels than the default mapping can handle, or if it includes specialty modules, the process data objects must be enabled manually.

5.2.3.3 Enabling the default PDOs / manually enabling process data

Certain XN300 slice modules provide additional payload data that can be manually mapped to unused PDOs. The appropriate subsections in → Section “7 Product-specific CAN objects XN300 slice modules”, page 105 contain tables listing these PDOs for each XN300 slice module. For example, the PDOs for the XN322-20DI-PD can be found in Table , page 112.

For a description on how to manually map the aforementioned payload data to PDOs, please consult the manual for your configuration tool.

5.2.4 Adding CANopen communication libraries

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

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

5.3 Configuring the system with XSOFT-CODESYS-3

The information below uses the XSOFT-CODESYS-3 Version 3.5.6 configuration program in standard mode, as well as the XN-312-GW-CAN_V0102.eds EDS file. Moreover, the example below uses the following devices:

- CANopen manager XC-152
- CANopen device XN-312-GW-CAN
- XN300 slice modules:
  - XN-322-20DI-PCNT
  - XN-322-20DI-PCNT
  - XN-322-10AI-TEKT

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

Make sure that you are running XSOFT-CODESYS-3 in standard mode. If you do not, you may be unable to carry out some of the steps below.

Start XSOFT-CODESYS-3 in standard mode

▶ Click on < Tools I Options | Features >.
▶ Click on the "Predefined feature sets..." button.
▶ Select the "Standard" option from the drop-down menu.

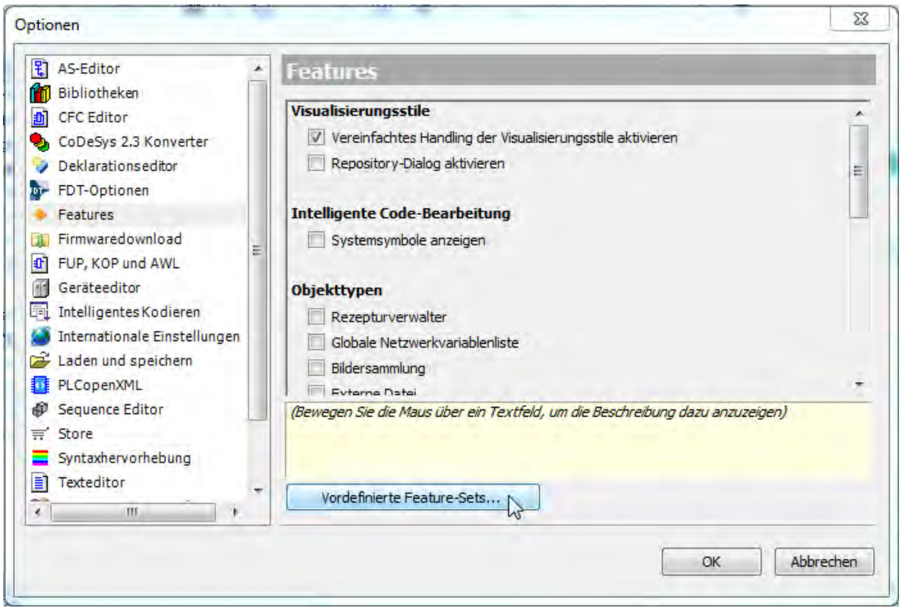


Figure 36: "Options" dialog box with "Predefined feature sets..." button

Creating Project

▶ After starting XSOFT-CODESYS-3, open a new project by clicking on < File I New Project...>.
▶ The "Standard Project" dialog box will appear. Select the "Standard project" option in the "Templates" pane 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., "XC-152".

▶ Now use the "PLC_PRG in" drop-down menu to select the programming language you want, e.g., "Ladder Logic Diagram (LD)".

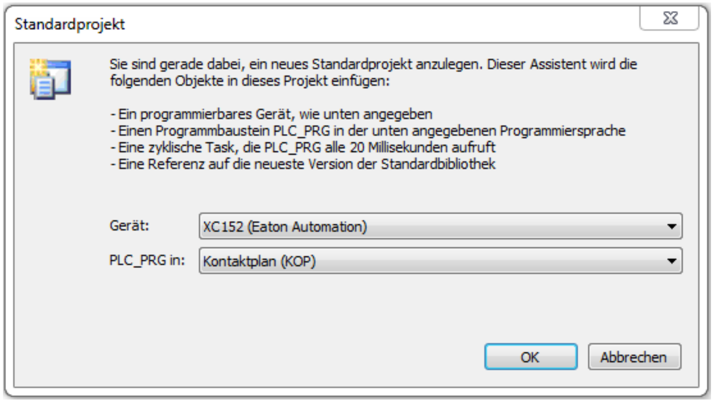


Figure 37: Configuring a standard project

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

The device will be added to the project.

5.3.2 Adding the CAN manager

Follow the steps below to enable communication via CANopen:

▶ In the "Devices" pane on the left, select the device you added previously, e.g., "Device (XC-152)".

▶ 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.

▶ Expand the "CANbus" node under the "Fieldbusses" node. Select "CAN-bus" and click on "Add Device."

The field bus will be added to the configuration structure.

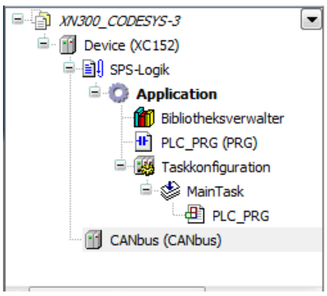


Figure 38: Configuration structure with bus system selected

▶ Right-click on "CANbus (CANbus)" and select the <Add Device...> option.

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

▶ Expand the "CANopen" node under the "Fieldbusses" node. Then expand the "CANopenManager" node, select "CANopen Manager," and click on "Add Device."

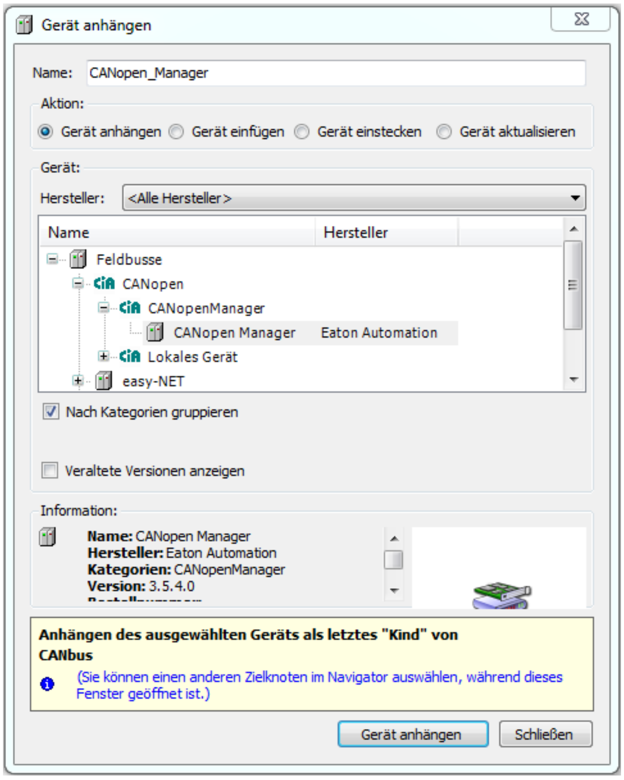


Figure 39: "Add Device" dialog box

The device will be configured as a CANopen manager.

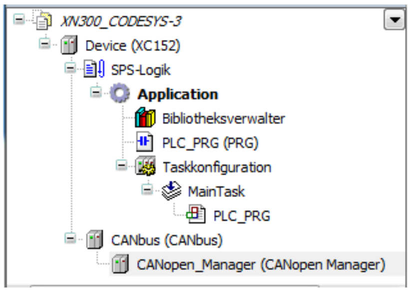


Figure 40: "Devices" pane with CAN manager in the configuration structure

- ▶ Double-click on CANopen_Manager (CANopen Manager).
- ▶ Check the settings under the "CANopen Manager" tab.
  - – The "Node ID" for the CAN manager must fall within a range of 1 to 127 and must only be assigned once on the entire network, i.e., it must be a unique node ID.
  - – If the "Autostart CANopenManager" function is enabled, the CANopen manager will start automatically (switch to the OPERATIONAL state) after all mandatory slaves are ready. If the option is disabled, the CANopen manager will have to be started by the application.

– If the "Start Slaves" function is enabled, the CANopen manager will start the slaves. If the option is disabled, the slaves will have to be started by the application.

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

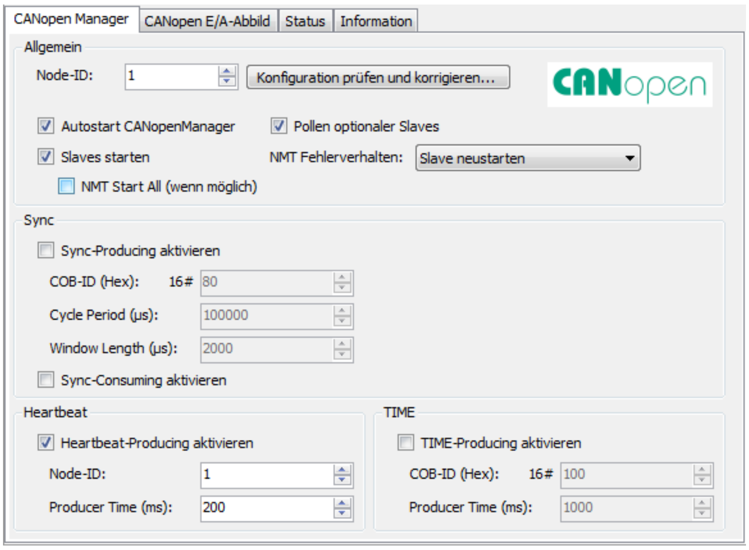


Figure 41: CAN parameters for CANopen manager

Setting the baud rate for the CANopen manager

The baud rate for the CANopen manager (XC-152 in this example) and the baud rate for the CANopen device (XN-312-GW-CAN in this case) must match.

The maximum bus length must be used as the basis for determining the maximum transfer rate (→ Section " Maximum cable length (m)", page 36).

The transfer rate for the XN-312-GW-CAN gateway needs to be set using the corresponding DIP switches (→ Section "1.5.2 Baud rate").

Follow the steps below to set the transfer rate for the CANopen manager:

▶ In the "Devices" configuration structure, double-click on <CANbus (CANbus)>.
▶ Open the "CANbus" tab and select the baud rate you want, e.g., <125000>.

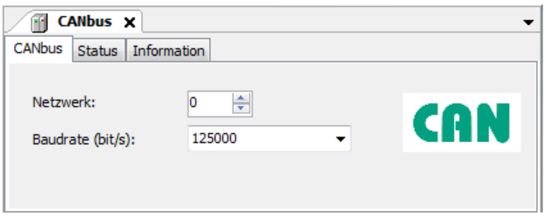


Figure 42: CANopen manager baud rate

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

5.3.3 Adding CANopen devices

- ▶ To add the XN-312-GW-CAN CANopen device to your configuration, right-click on "CANopen Manager (CANopen Manager)."
- ▶ Click on the <Add Device…> option.

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

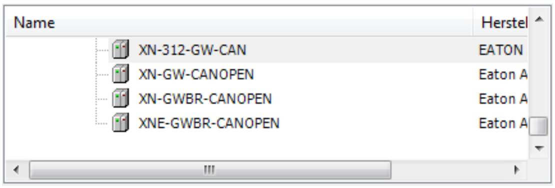


Figure 43: Adding a CANopen device

- ▶ Select the appropriate CANopen device.
- ▶ Click on the <Add Device> button.

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

- ▶ If the CAN slaves you want are not found in the list, update the X-SOFT-CODESYS-3 version you are using or load the appropriate EDS file → Section “4.2 Project-specific EDS file”, page 50.
  You can find updates and EDS files by visiting the Download Center on our website → Page 11.

5.3.4 Configuring the XN312 gateway

Check the settings for the CANopen device's parameters.

Node ID

- ▶ In the "Devices" configuration structure, double-click on the CANopen device you added previously, e.g., "XN_312_GW_CAN (XN-312-GW-CAN)."
- ▶ Open the "CANopen Remote Device" tab and check the following:
  - The device's "Node ID"
    - Is set using the DIP switches on the device
    - must be in the range 1 to 31.
    - And must only be assigned once on the entire network, i.e., it must be a unique node ID

→ Make sure that the node ID set with the DIP switches matches the node ID shown under the "CAN parameters" tab!
Change the node ID if necessary!

5.3.4.1 Configuring XN-322 slice modules

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

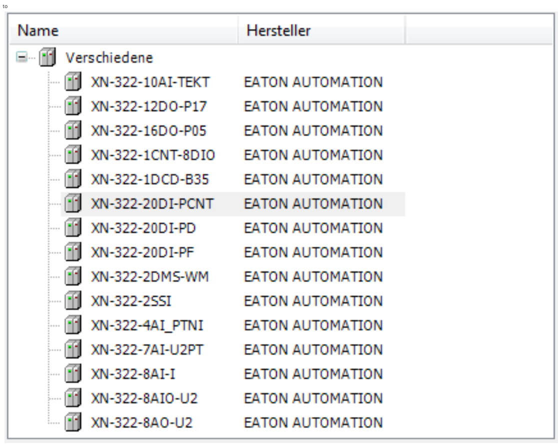


Figure 44: "Add Device" dialog box for selecting CAN modules for the CANopen device (XN-312-GW-CAN in this case)

- Select the I/O slice module you want, e.g., "XN-322-20DI-PCNT."
- Then click on the <Add Device> button.

The I/O slice module will be added to the "Devices" configuration structure.

→ 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 CANopen device.

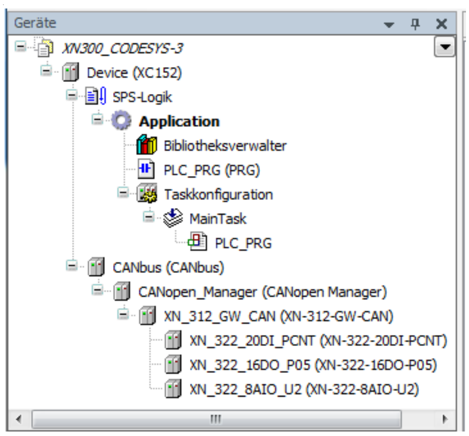


Figure 45: "Devices" configuration structure

5.3.4.2 Automatic PDO mapping

Automatic PDO mapping will be enabled by default. This means that, for each I/O slice module, certain process data will be mapped to PDOs by default. For more information, please refer to the corresponding tables for CANopen objects and vendor-specific objects in section
→ Chapter 7 “Product-specific CAN objects XN300 slice modules”, page 105.

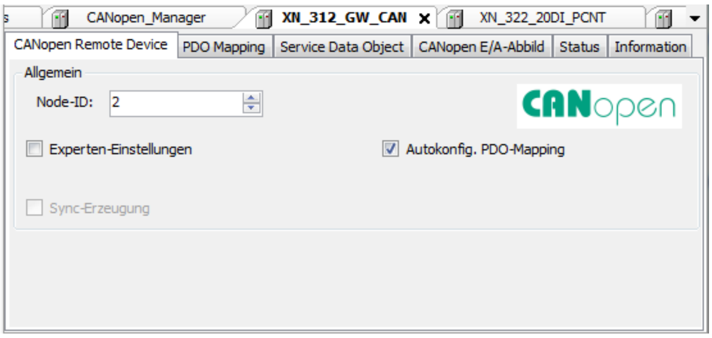


Figure 46: "CANopen Remote Device" tab: automatic PDO mapping enabled

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

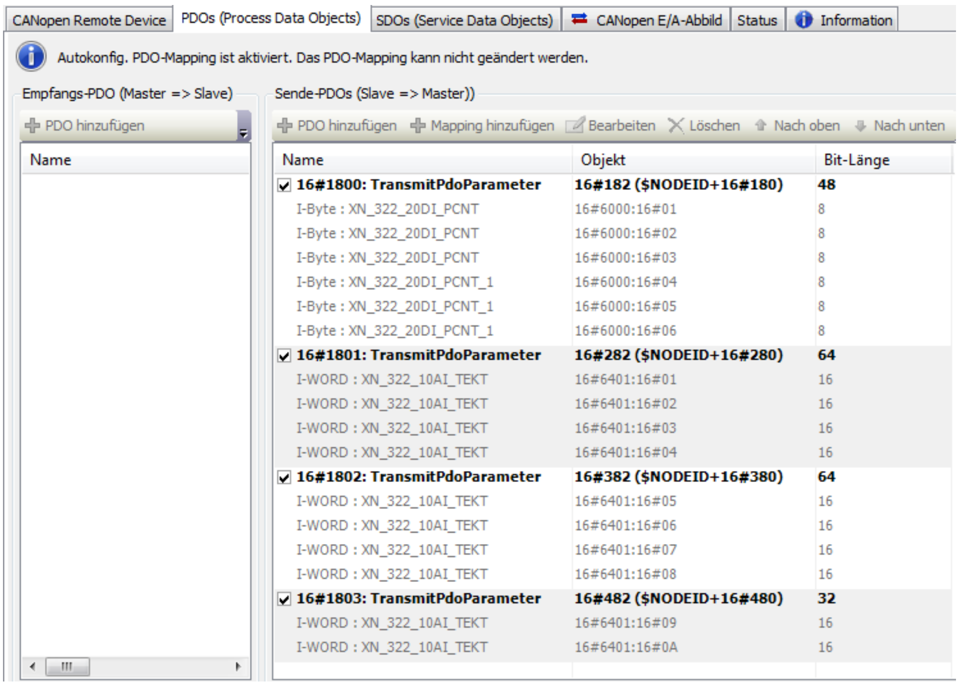


Figure 47: Mapping table in "PDO Mapping" tab

If the mapping table does not include all of the I/O slice modules' process data, you will need to map the unmapped data manually.

Automatic mapping limits

If I/O slice modules are added to the CANopen device, the corresponding addresses will be assigned automatically. A PDO contains a maximum of 8 bytes of process data. In XSOFT-CODESYS-3, automatic PDO mapping can be used successfully until one of the following limits is reached:

- The maximum number of PDOs will depend on the number of PDOs supported by the gateway; please refer to → Chapter 10 “Appendix”, page 376.
- A maximum of **64** digital input channels
- A maximum of **64** digital output channels
- A maximum of **12** analog input channels
- A maximum of **12** analog output channels
- No specialty modules

This default "mapping" and activation of process data is specified in accordance with communication profile CiA DS-301.

→ If the XN300 station has more input or output channels than specified above, or if it includes specialty modules, the process data objects must be enabled manually (→ Chapter 5 “Connecting the PLC to the gateway using CODESYS”, page 59).

5.3.4.3 Manual mapping

Certain XN300 slice modules provide information that needs to be mapped manually to unused PDOs. The appropriate subsections in → Section “7 Product-specific CAN objects XN300 slice modules”, page 105 contain tables listing these PDOs for each XN300 slice module. For example, the PDOs for the XN322-20DI-PD can be found in Table , page 112.

In this case, as well as in cases in which you want to change the mapping for the PDOs, you will need to map the corresponding data manually. The following example shows how to add an additional PDO and then add the process data for the XN-322-20DI-CNT device's counter to the mapping table.

→ For more detailed information, please consult the following documents:

- The manual for your configuration tool
- The online help for CODESYS, which can be opened by pressing the <F1> function key
- System description CiA Draft Standard DSP30

▶ Ideally, you should first use default mapping and add the I/O slice modules as devices to the CANopen device before doing any manual mapping → Section “5.3.4.2 Automatic PDO mapping”.

Switching to manual mapping

Follow the steps below to switch to manual mapping:

- Double-click on the CANopen device and open the "CANopen Remote Device" tab.
- Enable the "Enable Expert Settings" checkbox.
- Disable the "Autoconfig PDO Mapping" checkbox.
- Open the "PDO Mapping" tab.

Manually adding a PDO

The pane on the left will show the receive PDOs (RPDOs), while the pane on the right will show the send PDOs (TPDOs).

- If necessary, you can add a new PDO by clicking on the "Add PDO..." button.
  Do not exceed the maximum number of 16 RPDOs and 16 TPDOs! The program will show an error message otherwise!
- Double-click on a PDO that has been added, e.g., "TransmitPdoParameter," and configure its properties.

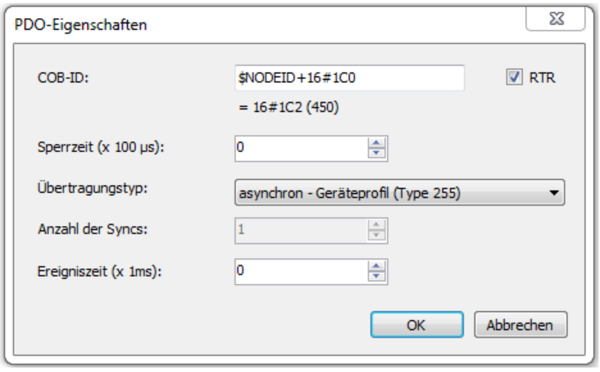
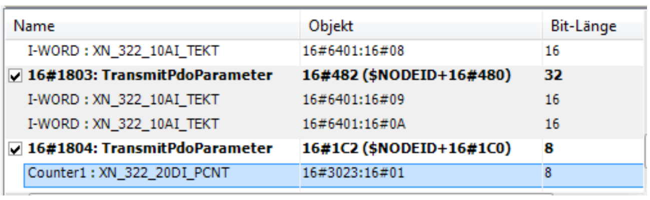


Figure 48: Selecting a transmission type for the manually mapped PDO

Defining the process data for a PDO

- Select the PDO you want, e.g., "TransmitPdoParameter."
- Click on the "Add Mapping..." button.
- In the "Select item from object directory" dialog box, select the item you want, e.g., "Counter1:XN_322_20DI_PCNT"
- Confirm with OK.

The item you selected will be added to the PDO.



Name	Objekt	Bit-Länge
I-WORD : XN_322_10AI_TEKT	16#6401:16#08	16
16#1803: TransmitPdoParameter	16#482 ($NODEID+16#480)	32
I-WORD : XN_322_10AI_TEKT	16#6401:16#09	16
I-WORD : XN_322_10AI_TEKT	16#6401:16#0A	16
16#1804: TransmitPdoParameter	16#1C2 ($NODEID+16#1C0)	8
Counter1 : XN_322_20DI_PCNT	16#3023:16#01	8

Figure 49: Manually added TPDO with process data for counter 1

5.3.5 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 CANopen device, e.g., "XN_312_GW_CAN (XN-312-GW-CAN)."
- ▶ Open the "Service Data Object" tab.
- ▶ Click on the "New..." button.

A list of all available SDO objects will be shown.

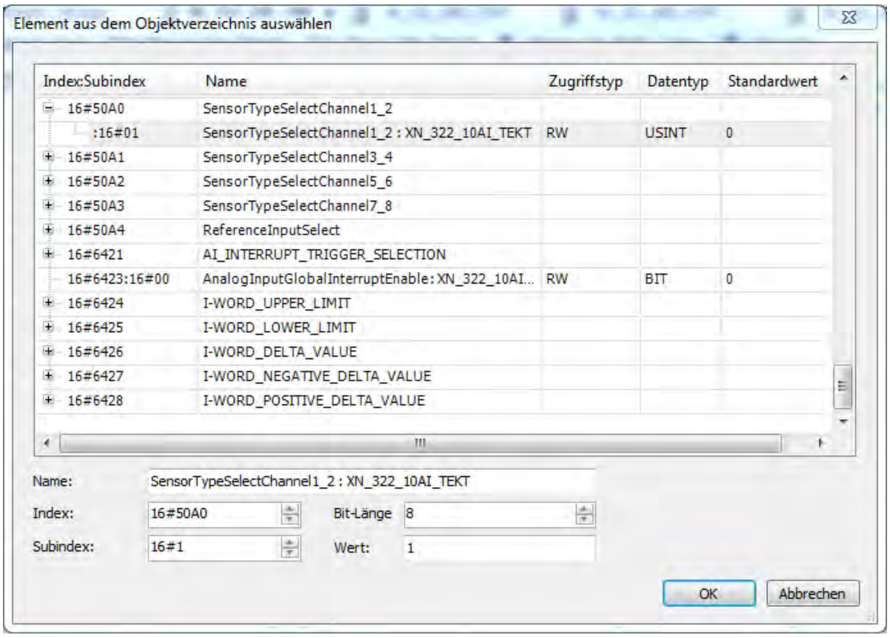


Figure 50: List of all available SDO objects

- ▶ 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 "Service Data Object" tab will show default device initialization values with a gray font and modified device initialization values with a black font.

5.3.6 Adding CANopen communication libraries

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

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

6 Object dictionary for XN-312-GW-CAN gateway

CANopen has three different types of objects:

1. Standard objects as defined in CiA301
2. Vendor-specific objects (2000 – 5FFF), e.g., used to configure all of the device's parameters, e.g., node ID
3. Profile-specific objects as defined in CiA401 (6000 – 7FFF); these objects are used for default mapping purposes if the user does not use any mapping.

6.1 Overview of objects 0x1000 to 0x1AFF

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

The entries in the CANopen 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 CAN bus.

For more information on the type and use of the listed CANopen-specific entries in the object dictionary, please refer to /CiA301/. General overview of all objects

Index	page	Name
1000_{hex}	→ Page 73	Object 1000hex: Device type (device profile number + additional information)
1001_{hex}	→ Page 82	Object 1001hex Error Register
1003_{hex}	→ Page 83	Object 1003hex Error History
1005_{hex}	→ Page 87	Object 1005hex Sync COB-ID
1008_{hex}	→ Page 74	Object 1008hex: Device name
1009_{hex}	→ Page 74	Object 1009hex: Manufacturer hardware version
$100A_{hex}$	→ **Page 75**	Object 100Ahex: Manufacturer software version
$100C_{hex}$	→ Page 84	Object 100Chex Guard Time
$100D_{hex}$	→ Page 84	Object 100Dhex Life Time Factor
1014_{hex}	→ Page 88	Object 1014hex Emcy COB-ID
1016_{hex}	→ Page 75	Object 1016hex Consumer Heartbeat Time
1017_{hex}	→ Page 86	Object 1017hex Producer Heartbeat Time
1018_{hex}	→ Page 75	Object 1018hex: Identity object
1027_{hex}	→ Page 90	Module Identification Number (Object 0x1027)
1029_{hex}	→ Page 83	Objekt 1029hex Error Behaviour Object/Communication Error (rw)
1200_{hex}	→ Page 90	Object 1200hex: Server SDO parameters
1400_{hex} to $140F_{hex}$	→ Page 91	Objects 1400hex to 140Fhex Receive PDO parameter

Index	page	Name
1600_{hex} to $160F_{hex}$	→ Page 93	Objects 1600hex to 160Fhex Receive PDO Mapping Parameter
1800_{hex} to $180F_{hex}$	→ Page 96	Objects 1800hex to 180Fhex Transmit PDO Parameter
$1A00_{hex}$ to $1A0F_{hex}$	→ Page 100	Objects 1A00hex to 1A0Fhex Transmit PDO Mapping Parameter
A description of the following objects can be found in the user manuals for the corresponding XN-322 slice modules		

The following table provides an overview of the supported entries in the object dictionary as defined in communication profile CiA DS-301.

The **Index (hex)** column specifies the entry's position in the object dictionary.

The **Object** column specifies the object's type.

The **Name** column specifies the pre-defined symbolic name for the entry.

The **Type** column specifies the entry's data type as pre-defined in CiA DS-301.

The **Access** column specifies the access options for the entry, where:

- rw = read and write
- rww (read/write/write) = read and write for SDOs, write only for PDOs
- ro = read only
- wo = write only
- const = constant / read only

The **M/O** column specifies whether the entry is mandatory or optional.

Table 3: List of objects for the communication profile

Index (hex)	object	Name	Part no.	Access	M/O
1000_{hex}	VAR	device type	Unsigned32	const	M
1001_{hex}	ARRAY	error register	Unsigned8	ro	M
1003_{hex}		error history	Unsigned32		M
1005_{hex}	VAR	COB-ID sync message	Unsigned32	rw	O
1008_{hex}	VAR	manufacturer device name	Vis string	const	O
1009_{hex}	VAR	manufacturer hardware version	Vis string	const	O
$100A_{hex}$	VAR	manufacturer software version	Vis string	const	O
$100C_{hex}$	VAR	Guard time	Unsigned16	rw	O
$100D_{hex}$	VAR	life time factor	Unsigned8	rw	O
1014_{hex}	VAR	COB-ID emergency	Unsigned32	rw	O
1016_{hex}	ARRAY	Consumer heartbeat time	Unsigned16	rw	O

Index (hex)	object	Name	Part no.	Access	M/O
1017_{hex}	VAR	Producer Heartbeat Time	Unsigned16	rwProduct description	O
1018_{hex}	RECORD	Identity Object	Identity	ro	O
1027_{hex}	ARRAY	Module list	Unsigned16	ro	M
1029_{hex}		error behavior object		rw	
Server SDO parameter (22_{hex})					
1200_{hex}	RECORD	1. Server SDO parameter	SDOParameter	ro	O
Receive PDO communication parameter (20_{hex})					
1400_{hex}	RECORD	1. receive PDO parameter	PDOCommPar	rw	O
1401_{hex}	RECORD	2. receive PDO parameter	PDOCommPar	rw	O
1402_{hex}	RECORD	3. receive PDO parameter	PDOCommPar	rw	O
...	...	...	...	...	...
$140F_{hex}$	RECORD	16. receive PDO parameter	PDOCommPar	rw	O
Receive PDO mapping parameter (21_{hex})					
1600_{hex}	ARRAY	1. receive PDO mapping	PDOMapping	rw	O
1601_{hex}	ARRAY	2. receive PDO mapping	PDOMapping	rw	O
1602_{hex}	ARRAY	3. receive PDO mapping	PDOMapping	rw	O
...	...	...	...	...	...
$160F_{hex}$	ARRAY	16. receive PDO mapping	PDOMapping	rw	O
Transmit PDO communication parameter (20_{hex})					
1800_{hex}	RECORD	1. transmit PDO Parameter	PDOCommPar	rw	O
1801_{hex}	RECORD	2. transmit PDO parameter	PDOCommPar	rw	O
1802_{hex}	RECORD	3. transmit PDO parameter	PDOCommPar	rw	O
...	...	...	...	...	...
$180F_{hex}$	RECORD	32. transmit PDO parameter	PDOCommPar	rw	O
Transmit PDO mapping parameter (21_{hex})					
$1A00_{hex}$	ARRAY	1. transmit PDO mapping	PDOMapping	rw	O
$1A01_{hex}$	ARRAY	2. transmit PDO mapping	PDOMapping	rw	O
$1A02_{hex}$	ARRAY	3. transmit PDO mapping	PDOMapping	rw	O
...	...	...	...	...	...
$1A0F_{hex}$	ARRAY	32. transmit PDO mapping	PDOMapping	rw	O

6.2 Information regarding the XN300 station

6.2.1 Object 1000_{hex}: Device type

Object 1000_{hex} contains the XN300 system block's type and function.

A value of 008F 0191$_{hex}$ indicates that the profile is supported in accordance with CiA Draft Standard Proposal 401.

Table 4: Description of object 1000$_{hex}$

Feature	Sub index	Description / Value
Name		Device type
Object code		VAR
PDO mapping		No
Data type	Sub index 00$_{hex}$	Unsigned32
Access	Sub index 00$_{hex}$	ro
Default value XN-312-GW_CAN	Sub index 00$_{hex}$	008F0191$_{hex}$

6.2.2 Object 1008$_{hex}$: Device name

Object 1008$_{hex}$ contains the vendor-specific device name.

Table 5: Description of object 1008$_{hex}$

Feature	Sub index	Description / Value
Name		Device Name
Object code		VAR
PDO mapping		No
Data type	Sub index 00$_{hex}$	Visible string
Access	Sub index 00$_{hex}$	ro
Default value XN-312-GW_CAN	Sub index 00$_{hex}$	–

6.2.3 Object 1009$_{hex}$: Manufacturer hardware version

Object 1009$_{hex}$ contains the designation for the hardware version.

Table 6: Description of object 1009$_{hex}$

Feature	Sub index	Description / Value
Name		Manufacturer Hardware Version
Object code		VAR
PDO mapping		No
Data type	Sub index 00$_{hex}$	Visible string
Access	Sub index 00$_{hex}$	ro
Default value XN-312-GW-CAN	Sub index 00$_{hex}$	XN-GW-CANopen

→ The description of the values corresponds to the object dictionary used as of this writing (Product description).

6.2.4 Object 100A$_{hex}$: Manufacturer software version

Object 100A$_{hex}$ contains the designation for the software version.

Table 7: Description of object 100A$_{hex}$

Feature	Sub index	Description / Value
Name		Manufacturer Software Version
Object code		VAR
PDO mapping		No
Data type	Sub index 00$_{hex}$	Visible string
Access	Sub index 00$_{hex}$	ro
Default value XN-312-GW-CAN	Sub index 00$_{hex}$	–

→ The description of the values corresponds to the object dictionary used as of this writing (Product description).

6.2.5 Object 1018$_{hex}$: Identity object

Object 1018$_{hex}$ contains general information regarding the XN-312 gateway.

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

The vendor-specific revision number (sub-index 03$_{hex}$) consists of a major revision number and a minor revision number. The main revision number is used to specify a specific CANopen behavior, meaning that the number needs to be increased whenever the CANopen functionality is expanded. Meanwhile, the minor revision number is used to distinguish between different versions that have the same CANopen behavior.

Table 8: Description of 1018$_{hex}$

Feature	Sub index	Description / Value
Name		Identity Object
Object code		RECORD
PDO mapping		No
Data type	Sub index 00$_{hex}$	Unsigned8
	Sub index 01$_{hex}$ to 04$_{hex}$	Unsigned32
Access	Sub index 00$_{hex}$	ro
	Sub index 01$_{hex}$ to 04$_{hex}$	ro

1) The revision numbers are up-to-date as of this writing (Product description).

Feature	Sub index	Description / Value
Function	Sub index 00_{hex}	Number of entries
	Sub index 01_{hex}	Manufacture ID
	Sub index 02_{hex}	Product code (here:article no.)
	Sub index 03_{hex}	Revision Number
	Sub index 04_{hex}	Serial Number
Default value XN-312-GW-CAN	Sub index 00_{hex}	No
	Sub index 01_{hex}	$0x1C7_{hex}$
	Sub index 02_{hex}	Article No.: 178782 XN-312--GW-CAN
	Sub index 03_{hex}	Revision numbers[1]: 0x010100A5 = $V01.01.165_{dez}$
	Sub index 04_{hex}	No

1) The revision numbers are up-to-date as of this writing (Product description).

Revision Number

As of this writing, the revision number corresponds to the latest software version. This revision number has the following format:

Format: 0xMMmmRRRR

- MM = Major version,
- mm = Minor version
- RRRR = Build / Revision

Table 9: Bit structure for revision number

Bit value	MSB		LSB
Bit	31 …24	23 …16	15 … 0
Meaning	Major revision number	Minor revision number	Build / Revision
Example	0000 0001	0000 0001	0000 0000 1010 0101

6.2.6 Module Identification Number (Object 0x1027)

Object 0x1027 contains the module IDs for all the XN300 slice modules present in an XN300 system block.

The number of listed modules is stored in sub-index 00_{hex}.

Meanwhile, consecutive sub-indexes $01_{hex} \leqq nh \leqq 1F_{hex}$ (31 dec) are used to describe the XN-322 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.

→ There is an individual identification number for each XN-322 slice module model

Feature	Description / Value	EDS
Name	Module Identification Number	[MxSubExt1027] ParameterName=ModuleID ObjectType=0x7 DataType=0x0006 AccessType=ro DefaultValue=<default> PDOMapping=0 Count=1
Description	ModuleID	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	00: Count of Module Identification Numbers	
	01 … 1F$_{hex}$: Module position	
Access	ro	
Default value	<default$_{dec}$>	

Design of the data bytes:

Sub index 1≦n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	m2	B1	B0
	MSB															LSB

XN300 slice module	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$_{dec}$

XN300 slice module	Default value <default> decimal
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}

6.2.7 Serial Number XN-312-CANopen (Object 0x4000)

Object 0x4000 can be used to read the XN-312-GW-CAN gateway's serial number using SDO-based access.

Feature	**Description / Value17**	**EDS**
Name	Serial number XN-312-CANopen	ParameterName=SerialNumber XN-312-CANopen ObjectType=0x7 DataType=0x09 AccessType=const PDOMapping=0
Description	SerialNumber XN-312-CANopen	
Object Code	VAR	
Mapping	SDO	
Data type	Visible string (max.12Byte)	
Access	const	

6.2.8 Serial Number (Object 0x4001)

Object 0x4001 can be used to read the XN300 slice modules' serial numbers using SDO-based access.

Feature	**Description / Value17**	**EDS**
Name	Serial Number	[MxSubExt4001] ParameterName=SerialNumber ObjectType=0x7 DataType=0x09 AccessType=ro PDOMapping=0 Count=1
Description	SerialNumber	
Object Code	ARRAY	
Mapping	SDO	
Data type	Visible string (max.12Byte)	
Sub index	00: Number of serial numbers	
	01 … $1F_{hex}$: Module position	
Access	const	

6.2.9 User LED 1…16 (Object 0x4002)

Object 0x4002 can be used to read and write the states of the User LEDs on XN300 slice modules 1 to 16 using SDO-based access.

Feature	Description / Value17	EDS
Name	User LEDs 1..16(BitMask)	ParameterName=User LEDs 1..16(BitMask) ObjectType=0x7 DataType=0x0007 AccessType=rw PDOMapping=0 DefaultValue=0
Description	User LEDs 1..16(BitMask)	
Object Code	VAR	
Mapping	SDO	
Data type	UNSIGNED32	
Access	rw	
Default value	0	

Design of the data bytes:

Sub index 1≦n≦254	Byte 1								Byte 0:							
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	B3	m2	B1	B0
	TN 8		TN 7		TN 6		TN 5		TN 4		TN 3		TN 2		TN 1	
	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x

Sub index 1≦n≦254	Byte 3								Byte 2							
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	B3	m2	B1	B0
	TN 16		TN 15		TN 14		TN 13		TN 12		TN 11		TN 10		TN 9	
	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x

The following User LED states are possible:

Bit x+1	Bit x	Meaning Stations 1 to 16
0	0	User LED off
0	1	User LED flashing (200 ms ON, 1000 ms OFF)
1	0	User LED flashing (1000 ms ON, 200 ms OFF)
1	1	User LED on

6.2.10 User LED 17…32 (Object 0x4003)

Object 0x4003 can be used to read and write the states of the User LEDs on XN300 slice modules 17 to 32 using SDO-based access.

Feature	Description / Value17	EDS
Name	User LEDs 17..32(BitMask)	ParameterName=User LEDs 17..32(BitMask) ObjectType=0x7 DataType=0x0007 AccessType=rw PDOMapping=0 DefaultValue=0
Description	Status LED user 17..32(BitMask)	
Object Code	VAR	
Mapping	SDO	
Data type	UNSIGNED32	
Access	rw	
Default value	0	

Design of the data bytes:

Sub index 1≦n≦254	Byte 1								Byte 0:							
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	B3	m2	B1	B0
	TN 34		TN 23		TN 22		TN 21		TN 20		TN 19		TN 18		TN 17	
	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x

Sub index 1≦n≦254	Byte 3								Byte 2							
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	B3	m2	B1	B0
	TN 32		TN 31		TN 30		TN 29		TN 28		TN 27		TN 26		TN 25	
	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x	Bit x+1	Bit x

The following User LED states are possible:

Bit x+1	x	Meaning Stations 1 to 16
0	0	User LED off
0	1	User LED flashing (200 ms ON, 1000 ms OFF)
1	0	User LED flashing (1000 ms ON, 200 ms OFF)
1	1	User LED on

6.2.11 User LED Control (Object 0x4004)

Object 0x4004 can be used to control the state of the module's User LED.

Feature	Description / Value	EDS
Name	UserLEDControl	[MxSubExt4004] ParameterName=UserLEDControl ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	User LED Control	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	$0x00_{hex}$	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	b7	b6	b5	b4	B3	m2	B1	B0
	MSB							LSB

Bit B1	Bit B0	Meaning
0	0	User LED off
0	1	User LED flashing (200 ms ON, 1000 ms OFF)
1	0	User LED flashing (1000 ms ON, 200 ms OFF)
1	1	User LED on

6.2.12 Boot Loader Version (Object 0x400A)

Object 0x400A can be used to read the gateway's boot loader version using SDO-based access.

Feature	Description / Value17	EDS
Name	Boot loader version	[[400A] ParameterName=Boot loader version(String) ObjectType=0x7 DataType=0x0009 AccessType=ro PDOMapping=0
Description	Boot loader version	
Object Code	VAR	
Mapping	SDO	
Data type	Visible string (max.12 Byte)	
Access	ro	

6.2.13 Product Name Gateway (Object 0x400B)

Object 0x400B can be used to read the gateway's product name using SDO-based access.

Feature	Description / Value17	EDS
Name	Product name	ParameterName=Product name ObjectType=0x7 DataType=0x0009 AccessType=ro PDOMapping=0
Description	Product name	
Object Code	VAR	
Mapping	SDO	
Data type	Visible string (max.20 Byte)	
Access	ro	

6.2.14 Product Name (Object 0x400C)

Object 0x400C can be used to read the XN300 slice modules' product names using SDO-based access.

Feature	Description / Value	EDS
Name	Product Name	[MxSubExt400C] ParameterName=Product name ObjectType=0x7 DataType=0x0009 AccessType=ro PDOMapping=0 Count=1
Description	ProductName	
Object Code	ARRAY	
Mapping	SDO	
Data type	Visible string (max. 20 Bytes)	
Sub index	$00_{hex:}$ Number of product names	
	01 … $1F_{hex}$: Module position	
Access	ro	

6.3 Error Register

6.3.1 Object 1001_{hex} Error Register

Object 1001_{hex} contains the error register for the XN-312-GW-CAN gateway. In other words, this object holds the gateway's internal faults in a single byte as per DS301.

Table 10: Description of object 1001_{hex}

Feature	Sub index	Description / Value
Name		Error Register
Object code		VAR
PDO mapping		Yes

Feature	Sub index	Description / Value
Data type	Sub index 00_{hex}	Unsigned8
Access	Sub index 00_{hex}	ro
Default value XN300	Sub index 00_{hex}	0x0

Error Register							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
generic	Current	Voltage	0	Comm.	0	0	Manu.

Abbreviations	
generic	Group fault
Current	Short-circuit on output/ Current fault
Voltage	Voltage fault
Comm.	communication error (communication fault)
Manu.	manufacturer specific error

6.3.2 Object 1003_{hex} Error History

Object 1003_{hex} is used to log the gateway's last six EMCY frames. These frames can be read by the CAN master as a history at any time. In other words, this object holds the gateway's internal faults in a single byte

Table 11: Description of object 1003_{hex}

Feature	Sub index	Description / Value
Name		Error Register
Object code		VAR
PDO mapping		Yes
Data type	Sub index 00_{hex}	Unsigned32; number of error entries
Access	Sub index 00_{hex}	ro
Default value XN300	Sub index 00_{hex}	no; number of error entries
Data type	Sub index 01_{hex} … $0F_{hex}$	Unsigned32; error entries

6.3.3 Objekt 1029_{hex} Error Behaviour Object/Communication Error (rw)

Object 1029_{hex} (error behavior object/communication error (rw)) defines what the response to a guarding error will be. For more information, please refer to CiA DS301, sec. 7.5.2.32, Default = Switch to Pre-Operational.

Value	Description
0x00	Change to Pre-operational NMT state (only if currently in Operational NMT state)
0x01	No change of the NMT state
0x02	Change to NMT state Stopped

6.4 Operational readiness monitoring

6.4.1 Object 100C$_{hex}$ Guard Time

A message will be sent to the CANopen nodes at fixed intervals, with the interval's length being defined by the "guard time" object. If a node fails to respond with the specified guard COB-ID, it will be assigned the "Timeout" status. The number of failed attempts before the CANopen node is categorized as "not OK" is specified by the "life time factor" object (object 100D$_{hex}$).

Guard time = 0: Node guarding disabled.

→ Node guarding cannot be used together with the heartbeat protocol.

Table 12: Description of object 100C$_{hex}$

Feature	Sub index	Description / Value
Name		Guard Time
Object code		VAR
PDO mapping		No
Data type	Sub index 00$_{hex}$	Unsigned 16
Access	Sub index 00$_{hex}$	rw
Default value XN312-GW-CAN	Sub index 00$_{hex}$	200 [ms]

6.4.2 Object 100D$_{hex}$ Life Time Factor

The number of failed "node guarding" attempts before a CANopen node is categorized as "not OK" is specified with the "life time factor" object.

The length of the interval between these attempts is specified by the "guard time" object (object 100C$_{hex}$).

Table 13: Description of object $100D_{hex}$

Feature	Sub index	Description / Value
Name		Life Time Factor
Object code		VAR
PDO mapping		No
Data type	Sub index 00_{hex}	Unsigned 8
Access	Sub index 00_{hex}	rw
Default value XN-312-GW-CAN0	Sub index 00_{hex}	200 [ms]

6.4.3 Object 1016_{hex} Consumer Heartbeat Time

The heartbeat protocol is used to monitor the operational readiness of other CANopen nodes.

All nodes can register when a CANopen node drops out!

Object 1016_{hex} defines the interval at which a heartbeat is expected.

When using the heartbeat protocol, a device generates a heartbeat at a specific interval (please refer to "Object 1017_{hex}" "Producer Heartbeat Time" for more information). A different device then receives this heartbeat and monitors the interval.

The aforementioned interval needs to be longer than the transmitter's cycle time (please refer to "Object 1017_{hex}" for more information). Heartbeat monitoring starts after the first heartbeat frame is received.
If the consumer heartbeat time = 0, monitoring for the corresponding CANopen node will be disabled.

The interval is set as a multiple of 1 ms.

The heartbeat protocol should be viewed as an alternative to node/life guarding that, in contrast to guarding, does not use remote frames. Node guarding uses one-to-one monitoring, i.e., when using node guarding, one CANopen node will monitor one other CANopen node. In contrast, the heartbeat protocol can be used to implement one-to-many monitoring, i.e., every CANopen node for which the heartbeat protocol is enabled can be configured in such a way as to monitor the heartbeat of every other CANopen node, including the CAN master.

The XN-312-GW-CAN gateway can monitor a maximum of four other CANopen nodes as a heartbeat consumer.

Table 14: Description of object 1016_{hex}

Feature	Sub index	Description / Value
Name		Consumer heartbeat time
Object code		ARRAY
PDO mapping		No

Feature	Sub index	Description / Value
Data type	Sub index 00_{hex}	Unsigned8
	Sub index 01_{hex}	Unsigned16
Access	Sub index 00_{hex}	const
	Sub index 01_{hex}	rw
Default value XN-312-GW-CAN	Sub index 00_{hex}	4
	Sub index 01_{hex}	0

Table 15: Structure of consumer heartbeat time entry (Unsigned16)

	msb		lsb
Bits	31 to 24	23 to 16	15 to 0
Value	reserved (Default: 00_{hex})	Node ID	Heartbeat time
Data type	–	Unsigned8	Unsigned16

6.4.4 Object 1017_{hex} Producer Heartbeat Time

A CANopen node uses heartbeat signals to tell all the other nodes on the CANopen network that it is ready for operation, even if there have not been any data transfers for an extended period of time.

Object 1017_{hex} defines the interval (cycle time) for this heartbeat.

All nodes can register when a CANopen node drops out!

If the interval = 0, a heartbeat will not be used. The object's content is interpreted as a multiple of 1 ms.

A node cannot use the heartbeat function in combination with node guarding. If the two monitoring mechanisms are enabled simultaneously, the node will use the heartbeat protocol only.

Table 16: Description of object 1017_{hex}

Feature	Sub index	Description / Value
Name		Producer Heartbeat Time
Object code		VAR
PDO mapping		No
Data type	Sub index 00_{hex}	Unsigned16
Access	Sub index 00_{hex}	rw
Default value XN-312-GW-CAN	Sub index 00_{hex}	0

6.5 Synchronization and emergency identifiers

6.5.1 Object 1005$_{hex}$ Sync COB-ID

Object 1005$_{hex}$ defines the COB-ID to which SYNC is sent and, accordingly, the synchronization object's (SYNC) priority. The XN-312-GW-CAN gateway cannot generate SYNC messages, but can receive them. When a synchronization frame is received, the frame will trigger the cyclical transmission of PDOs if these PDOs have been configured with the "cyclic -synchronous" or "acyclic - synchronous" transmission type.

Default COB-ID = 0x80.

Outputs can be set simultaneously and/or inputs can be read simultaneously.

Table 17: Description of object 1005$_{hex}$

Feature	Sub index	Description / Value
Name		SYNC COB-ID
Object code		VAR
PDO mapping		No
Data type	Sub index 00$_{hex}$	Unsigned32
Access	Sub index 00$_{hex}$	rw
Default value XN-312-GW-CAN	Sub index 00$_{hex}$	0000 0080$_{hex}$

Table 18: Structure of SYNC COB-ID entry (Unsigned32)

	msb				lsb
bits	**31**	**30**	**29**	**28 to 11**	**10 to 0**
11 Bit ID	X	0	0	00 0000 0000 0000 0000	00010000000 11 Bit identifier
29 Bit ID	X	0	1	29 Bit Identifier	

1) XN-312-GW-CAN

Table 19: Description of SYNC COB-ID entry

Bit number	Value	Description
31 (MSB)	X	Fixed
30	0 1	Module will not generate a SYNC message Module will generate a SYNC message
29	0 1	11-Bit-ID (CAN 2.0A) 29-Bit-ID (CAN 2.0B)
28 to 11	0 X	if Bit 29 = 0 If bit 29 = 1: Bits 28 to 11 of SYNC COB-ID
10 to 0 (lsb)	X	Bits 10 to 0 of SYNC COB-ID

1) XN-312-GW-CAN

→ Bit 30 is static, i.e., its value cannot be changed.

6.5.2 Object 1014_{hex} Emcy COB-ID

Object 1014_{hex} contains the COB-ID used by the gateway to transmit its own error messages on the bus.

Default COB ID = 0x80 + Node ID.

Table 20: Description of object 1014_{hex}

Feature	Sub index	Description / Value
Name		Emcy COB-ID
Object code		VAR
PDO mapping		No
Data type	Sub index 00_{hex}	Unsigned32
Access	Sub index 00_{hex}	rw
Default value XN-312-GW-CAN	Sub index 00_{hex}	0000 0080_{hex}+Node ID

Table 21: Structure of EMCY COB-ID entry (Unsigned32)

	msb				lsb
bits	**31**	**30**	**29**	**28 to 11**	**10 to 0**
11 Bit ID	0/1	0	0	00 0000 0000 0000 0000	11 Bit Identifier
29 Bit ID	0/1	0	1	29 Bit Identifier	

Table 22: Description of EMCY COB-ID entry

Bit number	Value	Description
31 (MSB)	0 1	EMCY present / valid EMCY not present / invalid
30	0	Reserved (always 0)
29	0 1	11-bit ID (CAN 2.0A) –> In the case of XN-312-GW-CAN 29 Bit ID (CAN 2.0B)
28 to 11	0 X	if Bit 29 = 0 If bit 29 = 1: Bits 28 to 11 of 29-bit COB-ID
10 to 0 (LSB)	X	Bits 10 to 0 of COB-ID

6.6 Service data transmission

Object 1200_{hex}: Server SDO parameters

Object 1200_{hex} defines the priority for transmitting SDO1. The data's priority is defined by the identifier/COB-ID.

Table 23: Description of object 1200_{hex}

Feature	Sub index	Description / Value
Name		Server SDO parameter
Object code		RECORD
PDO mapping		No
Data type	Sub index 00_{hex}	Unsigned8
	Sub index 01_{hex}	Unsigned32
	Sub index 02_{hex}	Unsigned32
Function	Sub index 00_{hex}	Number of entries
	Sub index 01_{hex}	COB-ID Client->Server
	Sub index 02_{hex}	COB-ID Server->Client
Access	Sub index 00_{hex}	ro
	Sub index 01_{hex}	• ro: Object 1200_{hex}
	Sub index 02_{hex}	• ro: Object 1200_{hex}
Default valueXN300	Sub index 00_{hex}	2
	Sub index 01_{hex}	• Node ID + 00000600_{hex}: Object 1200_{hex}
	Sub index 02_{hex}	• Node ID + 00000580_{hex}: Object 1200_{hex}

6.7 Process output data transmission

Together with objects 1600_{hex} to $160F_{hex}$, objects 1400_{hex} to $140F_{hex}$ define which output data should be transmitted with which priority and which transmission type using RPDO transfers.

"RPDO" stands for "receive process data object."

Object 1400_{hex} defines the priority and the transmission type for RPDO**1** and refers to the process data objects received by the gateway.

Information defining which data should be sent with this PDO is stored in object 0x1600.
Sub-index 0: Size of appended data
Sub-index 1–8: Object index / sub-index / length of data entry 1–8

Accordingly, object 1401_{hex} provides this information for RPDO**2** together with object 1601_{hex}, etc.

The data's priority is defined with the identifier/COB-ID.

By default, the value for object 1400_{hex} will have already been configured.

This will be the case for all PDOs. Moreover, the first 4 RPDOs and TPDOs will be enabled by default.

This means that an XN300 system block with a maximum of 32 digital outputs and/or 32 analog outputs will take care of the RPDO-based transmission of process output data by itself if the default mapping for these input/output variables falls within the first four PDOs. The variables will be processed after starting without any additional configuration required. PDO1 contains byte variables, PDOs 2 to 4 contain word variables.

6.7.1 Objects 1400_{hex} to $140F_{hex}$ Receive PDO parameter

Objects 1400_{hex} to $140F_{hex}$ define the priority and transmission type for RPDO1 to RPDO16.

The priority is defined with the identifier/COB-ID using sub-index 01_{hex}. The most significant bit in sub-index 01_{hex} can be used to define the remaining content as invalid/valid. This means that the corresponding most significant hex number is >8.

The transmission type is defined using sub-index 02_{hex}. table 27, page 93 shows the available transmission types.

Objects 1600_{hex} to $160F_{hex}$ define which data content should be transmitted with RPDO1 to RPDO16.

Table 24: Description of objects 1400_{hex} to $141F_{hex}$

Feature	Sub-Index	Description / Value
Name		Receive PDO Parameter
Object code		RECORD
PDO mapping		No
Data type	Sub index 00_{hex}	Unsigned8
	Sub index 01_{hex}	Unsigned32
	Sub index 02_{hex}	Unsigned8
Function	Sub index 00_{hex}	Number of entries
	Sub index 01_{hex}	PDO COB-ID ()
	Sub index 02_{hex}	Transmission type (Table 27, page 93)
Access	Sub index 00_{hex}	ro
	Sub index 01_{hex}	rw
	Sub index 02_{hex}	rw

Feature	Sub-Index	Description / Value
Default value XN312-GW-CAN	Sub index 00_{hex}	02_{hex}
	Sub index 01_{hex}	• Node ID + 00000200_{hex}: Object 1400_{hex} • Node ID + 00000300_{hex}: Object 1401_{hex} • Node ID + 00000400_{hex}: Object 1402_{hex} • Node ID + 00000500_{hex}: Object 1403_{hex} • Node ID + $80000XXX_{hex}$: Objects 1404_{hex} to $140F_{hex}$: Identifiers are configured, but not yet enabled • 80000000_{hex}: Object $140F_{hex}$ to $141F_{hex}$
	Sub index 02_{hex}	FF_{hex} (255)

COB-ID:

Sub-index 01_{hex} has the following structure:

Table 25: Structure of COB-ID entry (sub-index 01_{hex})

Bits	msb 31	30	29	28 to 11	lsb 10 to 0
11 Bit ID	0 / 1	0 / 1	0	00 0000 0000 0000 0000	11 Bit Identifier
29 Bit ID	0 / 1	0 / 1	1	29 Bit Identifier	

Table 26: Description of COB-ID entry (sub-index 01_{hex})

Bit number	Value	Meaning
31 (msb)	0	PDO exists / is valid
	1	PDO does not exist / is invalid
30	0	RTR allowed for this PDO
	1	RTR not allowed for this PDO
29	0	11-Bit-ID (CAN 2.0A) (standard application)
	1	29-Bit-ID (CAN 2.0B)
28 to 11	0	if Bit 29=0 (standard application)
	X	If bit 29 = 1: Bits 28 to 11 of COB-ID
10 to 0 (lsb)	X	Bits 10 to 0 of COB-ID

Transmission type:

Sub-index 02_{hex} can have the following values:

Table 27: Description of transmission type

Value	PDO transmission				
	Cyclic	Acyclic	Synchro-nous	Asynchro-nous	Upon RTR only
0		X	X		
1	X		X		
2 to 254	reserved				
255				X	

6.7.2 Objects 1600_{hex} to $160F_{hex}$ Receive PDO Mapping Parameter

Objects 1600_{hex} to $160F_{hex}$ specify which data contents should be transmitted with RPDO1 to RPDO32. The data contents themselves (process output data in this case) are represented with product-specific mappable objects.

The process output data for the digital channels will be entered into objects 6200_{hex} for example.

A description of these objects can be found in this manual as indicated below:

→ Section “7.10.1 Write Digital Output 8-bit (Object 0x6200)”, page 140.

The entry in sub-indexes 01_{hex} to 40_{hex} of objects 1600_{hex} to $161F_{hex}$ includes the object number, the sub-index, and the length of the data content that needs to be transmitted with the corresponding RPDO.

An RPDO can transmit a maximum of 8 bytes (64 bits).

The number of sub-indexes will depend on the data length, and the user themselves are responsible for calculating and entering this number in the case of large projects (see below).

A data length of 8 bits will require 8 sub-indexes in order to represent a total of 64 bits. Meanwhile, a data length of 1 bit will require 64 sub-indexes for a total of 64 bits.

By default, objects 1600_{hex} to 1603_{hex} (RPDO1 to RPDO4) already reference the values for the first 64 digital output channels and the values for the first 12 analog output channels. The prerequisite for this is that the values be represented with object 6200_{hex} (digital values) and object 6411_{hex} (analog values).

Table 28: Description of objects 1600_{hex} to $161F_{hex}$

Feature	Sub-Index	Description / Value
Name		Receive PDO mapping parameter
Object code		RECORD
PDO mapping		No
Data type	Sub index 00_{hex}	Unsigned8
	Sub index 01_{hex} to Sub index 40_{hex}	Unsigned32
Function		
1600_{hex} to 1603_{hex}	Table 29, page 95	
1604_{hex} to $161F_{hex}$	Sub index 00_{hex}	Number of entries (depends on the data length)
	Sub index 01_{hex}	1. Mapped application object
	Sub index 02_{hex}	2. Mapped application object
	Sub index 03_{hex}	3. Mapped application object
	...	...
	Sub index 40_{hex}	64. Mapped application object
Access	Sub index 00_{hex}	rw
	Sub index 01_{hex} to Sub index 08_{hex}	rw
Default value XN300		
1600_{hex} to 1603_{hex}	Table 29, page 95	
1604_{hex} to $161F_{hex}$	Sub index 00_{hex} to Sub index 40_{hex}	No

→ The number of mapping objects that are automatically generated when the gateway starts will depend on the current physical structure of the XN300 station.

Table 29: Set default values for objects 1600_{hex} to 1603_{hex}

Object	Sub-Index	Default value XN300	Description	Applies to
1600_{hex}	01_{hex}	6200 0108_{hex}	1. Mapping object (digital output)	RPDO1
	02_{hex}	6200 0208_{hex}	2. Mapping object (digital output)	
	03_{hex}	6200 0308_{hex}	3. Mapping object (digital output)	
	04_{hex}	6200 0408_{hex}	4. Mapping object (digital output)	
	05_{hex}	6200 0508_{hex}	5. Mapping object (digital output)	
	06_{hex}	6200 0608_{hex}	6. Mapping object (digital output)	
	07_{hex}	6200 0708_{hex}	7. Mapping object (digital output)	
	08_{hex}	6200 0808_{hex}	8. Mapping object (digital output)	
1601_{hex}	01_{hex}	6411 0110_{hex}	1. Mapping object (analog output)	RPDO2
	02_{hex}	6411 0210_{hex}	2. Mapping object (analog output)	
	03_{hex}	6411 0310_{hex}	3. Mapping object (analog output)	
	04_{hex}	6411 0410_{hex}	4. Mapping object (analog output)	
1602_{hex}	01_{hex}	6411 0510_{hex}	1. Mapping object (analog output)	RPDO3
	02_{hex}	6411 0610_{hex}	2. Mapping object (analog output)	
	03_{hex}	6411 0710_{hex}	3. Mapping object (analog output)	
	04_{hex}	6411 0810_{hex}	4. Mapping object (analog output)	
1603_{hex}	01_{hex}	6411 0910_{hex}	1. Mapping object (analog output)	RPDO4
	02_{hex}	6411 $0A10_{hex}$	2. Mapping object (analog output)	
	03_{hex}	6411 $0B10_{hex}$	3. Mapping object (analog output)	
	04_{hex}	6411 $0C10_{hex}$	4. Mapping object (analog output)	

The following structure applies to the parameters of sub-indexes 01_{hex} to 40_{hex}:

Table 30: Structure of PDO mapping entries

msb		lsb
Object index (16 bits)	Sub index (8 Bit)	Object length (8 bits)

6.8 Process input data transmission

Together with objects 1A00$_{hex}$ to 1A0F$_{hex}$, objects 1800$_{hex}$ to 180F$_{hex}$ define which input data should be transmitted with which priority and which transmission type using TPDO transfers.

"TPDO" stands for "transmit process data object" and refers to process data transmitted by the gateway.

Object 1800$_{hex}$ defines the priority, the minimum transmission inhibit time, the maximum pause between two transmissions, and the transmission type for TPDO**1**. Object 1A00$_{hex}$ specifies the object index, sub-index, and data length for the data that needs to be transmitted with TPDO**1**.

Accordingly, object 1801$_{hex}$ provides this information for TPDO**2** together with object 1A01$_{hex}$, etc.

The data's priority is defined with the identifier/COB-ID.

A default mapping is already configured for all PDOs: 1600 to 160F and 1A00 to 1A0F. The COB-ID that needs to be used is already configured for all PDOs as well: 1400/1 to140F/1 and 1A00/1 to 1A0F/1. The first 4 PDOs, 0 to 3, are enabled and allowed to transmit. PDOs 4 to F need to be manually enabled.

This means that an XN300 station with a maximum of 64 digital inputs and 12 analog inputs will take care of the TPDO-based transmission of process input data by itself.

6.8.1 Objects 1800$_{hex}$ to 180F$_{hex}$ Transmit PDO Parameter

Objects 1800$_{hex}$ to 180F$_{hex}$ define the priority, the transmission type, the minimum inhibit time, and the maximum pause between two transmissions for TPDO1 to TPDO32.

The priority is defined with the identifier/COB-ID using sub-index 01$_{hex}$. The most significant bit in sub-index 01$_{hex}$ can be used to define the remaining content as invalid/valid. This means that the corresponding most significant hex number is >8.

The transmission type is defined using sub-index 02$_{hex}$. table 27, page 93 shows the available transmission types..

Sub-index 03$_{hex}$ defines the minimum transmission inhibit time.

Sub-index 05$_{hex}$ is used to define the maximum pause between two transmissions.

Objects 1A00$_{hex}$ to 1A0F$_{hex}$ define which data content should be transmitted with TPDO1 to TPDO32.

Table 31: Description of objects 1800_{hex} to $181F_{hex}$

Feature	Sub-Index	Description / Value
Name		Transmit PDO communication parameter
Object code		RECORD
PDO mapping		No
Parameter name	Sub index 00_{hex}	Number of entries
	Sub index 01_{hex}	COB-ID ()
	Sub index 02_{hex}	Transmission type (Table 34, page 99)
	Sub index 03_{hex}	Inhibit time (→ Chapter 6 “Object dictionary for XN-312-GW-CAN gateway”, page 99)
	Sub index 04_{hex}	Reserved
	Sub index 05_{hex}	Event timer (→ Section “ The "maximum pause between two transmissions" parameter”, page 99)
Data type	Sub index 00_{hex}	Unsigned8
	Sub index 01_{hex}	Unsigned32
	Sub index 02_{hex}	Unsigned8
	Sub index 03_{hex}	Unsigned16
	Sub index 04_{hex}	Unsigned8
	Sub index 05_{hex}	Unsigned16
Access	Sub index 00_{hex}	ro
	Sub index 01_{hex} to 05_{hex}	rw
Default value XN300	Sub index 00_{hex}	5
	Sub index 01_{hex}	• Index 1800_{hex}: 0000 0180_{hex} + Node ID • Index 1801_{hex}: 0000 0280_{hex} + Node ID • Index 1802_{hex}: 0000 0380_{hex} + Node ID • Index 1803_{hex}: 0000 0480_{hex} + Node ID • Indexes 1804_{hex} to $181F_{hex}$: invalid
	Sub index 02_{hex}	255
	Sub index 03_{hex} to 05_{hex}	0000_{hex}

COB-ID:

Sub-index 01_{hex} has the following structure:

Table 32: Structure of COB-ID entry (sub-index 01_{hex})

	msb				lsb
Bits	31	30	29	28 to 11	10 to 0
11 Bit ID	0 / 1	0 / 1	0	00 0000 0000 0000 0000	11 Bit Identifier
29 Bit ID	0 / 1	0 / 1	1	29 Bit Identifier	

Table 33: Description of COB-ID entry (sub-index 01_{hex})

Bit number	Value	Meaning
31 (msb)	0	PDO exists / is valid
	1	PDO does not exist / is invalid
30	0	RTR allowed for this PDO
	1	RTR not allowed for this PDO
29	0	11-Bit-ID (CAN 2.0A) (standard application)
	1	29-Bit-ID (CAN 2.0B)
28 to 11	0	if Bit 29=0 (standard application)
	X	If bit 29 = 1: Bits 28 to 11 of COB-ID
10 to 0 (lsb)	X	Bits 10 to 0 of COB-ID

Transmission type:

Sub-index 02_{hex} can have the following values:

Table 34: Description of transmission type

Transmission Type	PDO transmission				
	Cyclic	Acyclic	Synchro-nous	Asynchro-nous	Upon RTR only
0		X	X		
1	X		X		
2 to 252	reserved				
253				X	X
254	reserved				
255				X	

The "minimum inhibit time" parameter

The "inhibit time" parameter can be used to define an inhibit time between two transmissions in order to prevent high-priority messages from taking over the bus completely.
Sub-index 03_{hex} is used to define the inhibit time as a multiple of 100 µs. However, since the XN-312-GW-CAN gateway's internal clock has a resolution of 1 ms, inhibit time values smaller than 10 x 100 µs should not be used.

The "maximum pause between two transmissions" parameter

Sub-index 05_{hex} is used to define the time by which a TPDO must be transmitted at the latest even if there is no other event causing a transmission. When the event timer elapses, this will be detected as an event. However, any time an event causes a transmission, the event timer will be reset and restart.
The object's value will be interpreted as a multiple of 1 ms.

6.8.2 Objects $1A00_{hex}$ to $1A0F_{hex}$ Transmit PDO Mapping Parameter

Objects $1A00_{hex}$ to $1A0F_{hex}$ specify which data contents should be transmitted with TPDO1 to TPDO32. The data contents themselves (process input data in this case) are represented with product-specific mappable objects.

The process input data for the digital channels is entered into objects 6000_{hex}, $6200_{hex,}$ etc., for example.

For a description of these objects, please refer to manual "XN300 slice modules", MN050002".

The entry in sub-indexes 01_{hex} to 40_{hex} of objects $1A00_{hex}$ to $1A0F_{hex}$ includes the object number, the sub-index, and the length of the data content that needs to be transmitted with the corresponding TPDO.

A TPDO can transmit a maximum of 8 bytes.

By default, objects $1A00_{hex}$ to $1A03_{hex}$ (TPDO1 to TPDO4) already reference the values for the first 64 digital input channels and the values for the first 12 analog input channels. The prerequisite for this is that the values be represented with object 6000_{hex} (digital values) and object 6401_{hex} (analog values).

Table 35: Description of objects $1A00_{hex}$ to $1A1_{hex}$

Feature	Sub-Index	Description / Value
Type name		Transmit PDO mapping parameter
Object code		RECORD
PDO mapping		No
Data type	Sub index 00_{hex}	Unsigned8
	Sub index 01_{hex} to Sub index 40_{hex}	Unsigned32
Function		
$1A00_{hex}$ to $1A03_{hex}$	See note concerning mapping objects, Page 101	
$1A03_{hex}$ to $1A1F_{hex}$	Sub index 00_{hex}	Number of entries (depends on the data length)
	Sub index 01_{hex}	1. Mapped application object
	Sub index 02_{hex}	2. Mapped application object
	Sub index 03_{hex}	3. Mapped application object
	...	...
	Sub index 40_{hex}	64. Mapped application object
Access	Sub index 00_{hex}	rw
	Sub index 01_{hex} to Sub index 08_{hex}	rw
Default value XN300		
$1A00_{hex}$ to $1A03_{hex}$	See note concerning mapping objects, Page 101	
$1A04_{hex}$ to $1A1F_{hex}$	Sub index 00_{hex} to Sub index 40_{hex}	No

→ The number of mapping objects that are automatically generated when the gateway starts will depend on the current physical structure of the XN300 station.

Table 36: Set default values for objects $1A00_{hex}$ to $1A03_{hex}$

Object	Sub-Index	Default value XN300	Description	Applies to
$1A00_{hex}$	01_{hex}	$6000\ 0108_{hex}$	1. Mapping object (digital input)	TPDO1
	02_{hex}	$6000\ 0208_{hex}$	2. Mapping object (digital input)	
	03_{hex}	$6000\ 0308_{hex}$	3. Mapping object (digital input)	
	04_{hex}	$6000\ 0408_{hex}$	4. Mapping object (digital input)	
	05_{hex}	$6000\ 0508_{hex}$	5. Mapping object (digital input)	
	06_{hex}	$6000\ 0608_{hex}$	6. Mapping object (digital input)	
	07_{hex}	$6000\ 0708_{hex}$	7. Mapping object (digital input)	
	08_{hex}	$6000\ 0808_{hex}$	8. Mapping object (digital input)	
$1A01_{hex}$	01_{hex}	$6401\ 0110_{hex}$	1. Mapping object (analog input)	TPDO2
	02_{hex}	$6401\ 0210_{hex}$	2. Mapping object (analog input)	
	03_{hex}	$6401\ 0310_{hex}$	3. Mapping object (analog input)	
	04_{hex}	$6401\ 0410_{hex}$	4. Mapping object (analog input)	
$1A02_{hex}$	01_{hex}	$6401\ 0510_{hex}$	1. Mapping object (analog input)	TPDO3
	02_{hex}	$6401\ 0610_{hex}$	2. Mapping object (analog input)	
	03h	$6401\ 0710_{hex}$	3. Mapping object (analog input)	
	04_{hex}	$6401\ 0810_{hex}$	4. Mapping object (analog input)	
$1A03_{hex}$	01_{hex}	$6401\ 0910_{hex}$	1. Mapping object (analog input)	TPDO4
	02_{hex}	$6401\ 0A10_{hex}$	2. Mapping object (analog input)	
	03_{hex}	$6401\ 0B10_{hex}$	3. Mapping object (analog input)	
	04_{hex}	$6401\ 0C10_{hex}$	4. Mapping object (analog input)	

The following structure applies to the parameters of sub-indexes 01_{hex} to 40_{hex}:

Table 37: Structure of PDO mapping entries

msb		**lsb**
Object index (16 bits)	Sub index (8 Bit)	Object length (8 bits)

7 Product-specific CAN objects XN300 slice modules

The following sections are used to list the various product-specific object dictionary entries. The detailed description of the CAN objects for each XN300 module is organized as follows:

The **Feature** column provides a simple functional description of the entry.
The **Description/Value** column specifies either the entry's data type or its specific value.
The **EDS** column specifies how the object is described in the EDS file.

7.1 XN-322-8DI-PD

This module supports the provision of data for digital inputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

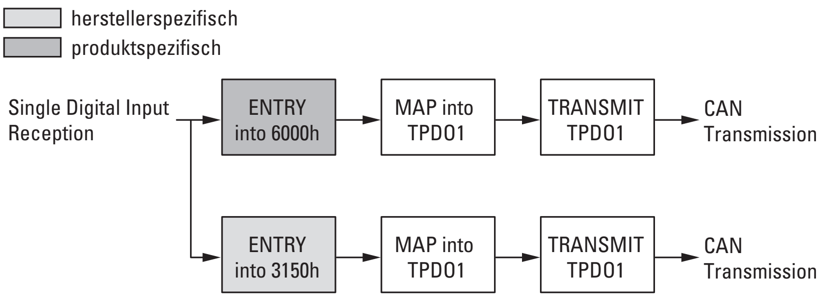


Figure 51: Block diagram showing the various CANopen objects for digital inputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6000	UNSIGNED8	I-BYTE	Digital Input 8-bit	Default	ro	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object

Manufacturer-specific objects

Index range for the XN-322-8DI-PD module: x150 to x15F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x3150	UNSIGNED8	Input1_8	Read Digital Input 1_8	Manual	ro	PDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDCon-trol	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO

7.1.1 Read Digital Input 8-Bit (Object 0x6000)

Object 0x6000 represents the digital value of the channels' digital input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-BYTE	[MxSubExt6000] ParameterName=I-BYTE ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Input 8-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.1.2 Read Digital Inputs (Object 0x3150)

Object 0x3150 represents the channels' formatted digital input values in a byte.

Feature	Description / Value	EDS
Name	Input1_8	[MxSubExt3150] ParameterName=Input1_8 ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Inputs	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

7.2 XN-322-16DI-PD

This module supports the provision of data for digital inputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

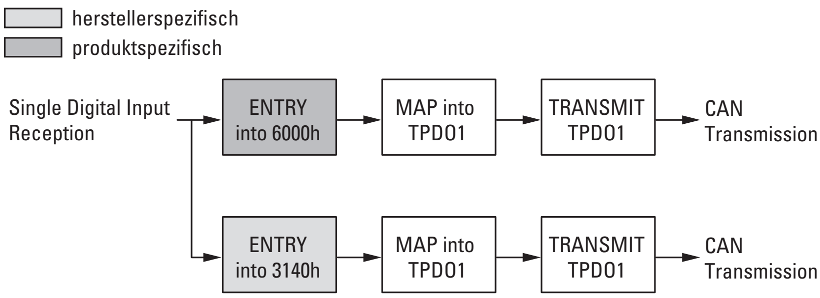


Figure 52: Block diagram showing the various CANopen objects for digital inputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6000	UNSIGNED8	I-BYTE	Digital Input 8-bit	Default	ro	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object

Manufacturer-specific objects

Index range for the XN-322-16DI-PD module: x150 to x15F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x3140	UNSIGNED16	Input1_16	Read Digital Input 1_16	Manual	ro	PDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDCon-trol	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO

7.2.1 Read Digital Input 8-Bit (Object 0x6000)

Object 0x6000 represents the digital value of the channels' digital input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-BYTE	[MxSubExt6000] ParameterName=I-BYTE ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=2
Description	Read Digital Input 8-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.2.2 Read Digital Inputs (Object 0x3140)

Object 0x3140 represents the channels' formatted digital input values in a word.

Feature	Description / Value	EDS
Name	Input1_16	[MxSubExt3140] ParameterName=Input1_16 ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Inputs	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	1 Byte								Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0	B7	B6	B5	B4	B3	B2	B1	B0
	DI 16	DI 15	DI 14	DI 13	DI 12	DI 11	DI 10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

7.3 XN-322-20DI-PD

This module supports the provision of data for digital inputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

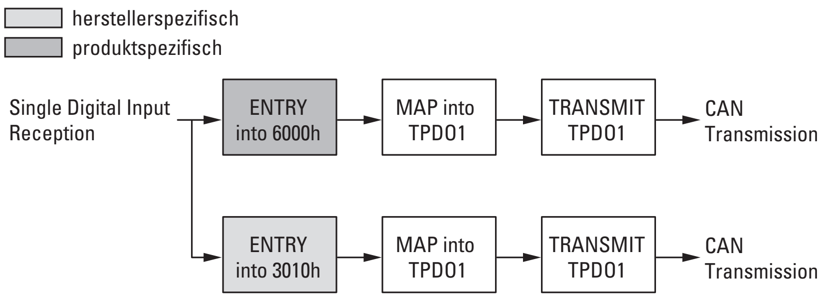


Figure 53: Block diagram showing the various CANopen objects for digital inputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6000	UNSIGNED8	I-BYTE	Digital Input 8-bit	Default	ro	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object

Manufacturer-specific objects

Index range for the XN-322-20DI-PD module: x010 to x01F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module ID number → Section "6.2.6 Module Identification Number (Object 0x1027)"	–	ro	SDO
0x3010	UNSIGNED32	Input1_20	Digital input channels 1 to 20	Manual	ro	PDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO

7.3.1 Read Digital Input 8-Bit (Object 0x6000)

Object 0x6000 represents the digital value of the channels' digital input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-BYTE	[MxSubExt6000] ParameterName=I-BYTE ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=3
Description	Read Digital Input 8-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.3.2 Read Digital Inputs (Object 0x3010)

Object 0x3010 represents the channels' formatted digital input values in a double word.

Feature	Description / Value	EDS
Name	Input1_20	[MxSubExt3010] ParameterName=Input1_20 ObjectType=0x7 DataType=0x0007 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Inputs	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	$0000\ 0000_{hex}$	

Design of the data bytes:

Sub index 1≦n≦254	**1 Byte**								**Byte 0:**							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
																LSB
	DI16	D15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

Sub index 1≦n≦254	**3 Byte**								**Byte 2**							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															
	–	–	–	–	–	–	–	–	–	–	–	–	DI20	DI19	DI18	DI17

7.4 XN-322-20DI-PF

This module supports the provision of data for digital inputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

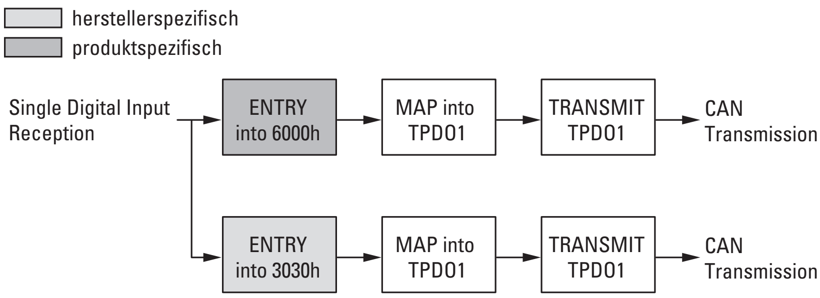


Figure 54: Block diagram showing the various CANopen objects for digital inputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6000	UNSIGNED8	I-BYTE	Digital Input 8-bit	Default	ro	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object

Manufacturer-specific objects

Index range for the XN-322-20DI-PF module: x030 to x03F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section “6.2.6 Module Identification Number (Object 0x1027)”, page 76	–	ro	SDO
0x3030	UNSIGNED32	Input1_20	Read Digital Input 1_20	Manual	ro	PDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDCon-trol	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO

7.4.1 Read Digital Input 8-Bit (Object 0x6000)

Object 0x6000 represents the digital value of the channels' digital input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-BYTE	[MxSubExt6000] ParameterName=I-BYTE ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=3
Description	Read Digital Input 8-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.4.2 Read Digital Inputs (Object 0x3030)

Object 0x3030 represents the channels' formatted digital input values in a double word.

Feature	Description / Value	EDS
Name	Input1_20	[MxSubExt3030] ParameterName=Input1_20 ObjectType=0x7 DataType=0x0007 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Inputs	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
																LSB
	DI16	D15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

Sub index 1≦n≦254	3 Byte								Byte 2							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															
	–	–	–	–	–	–	–	–	–	–	–	–	DI20	DI19	DI18	DI17

7.5 XN-322-20DI-PCNT

This module supports the provision of data for digital inputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

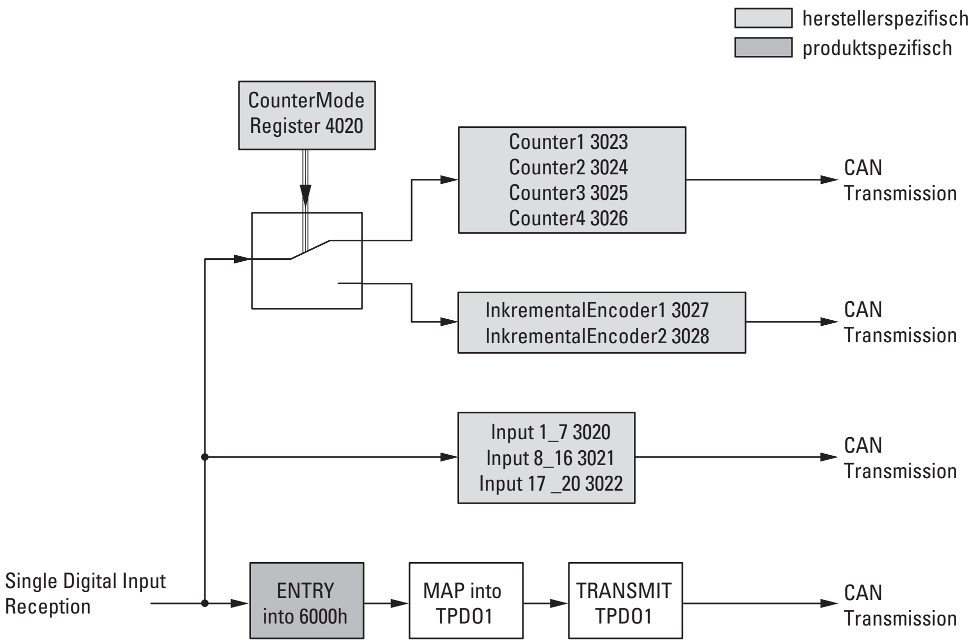


Figure 55: Block diagram showing the various CANopen objects for digital inputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
6000	UNSIGNED8	I-BYTE	Digital Input 8-bit	Default	ro	PDO

Manufacturer-specific objects

Index range for the XN-322-20DI-PCNT module: x020 to x02F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x3020	UNSIGNED8	Input1_8	Read Digital Input 1_8	Manual	ro	PDO
0x3021	UNSIGNED8	Input9_16	Read Digital Input 9_16	Manual	ro	PDO
0x3022	UNSIGNED8	Input17_20	Read Digital Input 17_20	Manual	ro	PDO
0x3023	UNSIGNED8	Counter1	Counter Register 1	Manual	ro	PDO
0x3024	UNSIGNED8	Counter2	Counter Register 2	Manual	ro	PDO
0x3025	UNSIGNED8	Counter3	Counter Register 3	Manual	ro	PDO
0x3026	UNSIGNED8	Counter4	Counter Register 4	Manual	ro	PDO
0x3027	UNSIGNED16	Incremental Encoder1	Incremental Encoder 1 Register	Manual	ro	PDO
0x3028	UNSIGNED16	Incremental Encoder2	Incremental Encoder 2 Register	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO
0x4020	UNSIGNED8	CounterMo-deRegister	Counter Mode Register	–	ro	SDO

→ Make sure to only use the data relevant to the selected operating mode. If applicable, content in non-relevant registers may also change. The operating mode can be selected using the counter mode register.

7.5.1 Read Digital Input 8-Bit (Object 0x6000)

Object 0x6000 represents the digital value of the channels' digital input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-BYTE	[MxSubExt6000] ParameterName=I-BYTE ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=3
Description	Read Digital Input 8-Bit	
Object Code	VAR	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.5.2 Digital Inputs x_y (Object 0x3020 to 0x3022)

Objects 0x3020 to 0x3022 represent the channels' formatted digital input values.

Feature	Description	Value	EDS
Name	Read Digital Input x_y		[MxSubExt302x] ParameterName=Inputx_y ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Input1_8	3020	
	Input9_16	3021	
	Input17_20	3022	
Object Code	VAR		
Mapping	PDO		
	Manual		
Data type	UNSIGNED8		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
DI16	DI15	D14	DI13	DI12	DI11	DI10	DI9

Byte 2:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–	–	DI20	DI19	DI18	DI17

7.5.3 Counter Register (Object 0x3023-0x3026)

Objects 0x3023 to 0x3026 are 8-bit count registers assigned to the corresponding inputs.

Any writing command to counter mode register 0x4020 will result in all the count registers object 0x3023 to 0x3028 being reset to 0x00.

ACHTUNG NOTICE

To reset the count registers, write to counter mode register 0x4020 (this register is used to select an operating mode).

Feature	Description	Value	EDS
Name	Counter1	3023	[MxSubExt302x] ParameterName=Counterx ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
	Counter2	3024	
	Counter3	3025	
	Counter4	3026	
Description	Counter Register Input x		
Object Code	VAR		
Mapping	PDO		
	Manual		
Data type	UNSIGNED8		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

Object 0x3023

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Object 0x3024

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Object 0x3025

Sub index 1≦n≦254	Byte 2							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Object 0x3026

Sub index 1≦n≦254	3 Byte							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.5.4 Incremental Encoder Register (Object 0x3027 - 0x3028)

Objects 0x3027 to 0x3028 are 16-bit count registers assigned to the corresponding inputs.

Any writing command to counter mode register 0x4020 will result in all the count registers object 0x3023 to 0x3028 being reset to 0x00.

ACHTUNG NOTICE

To reset the count registers, write to counter mode register 0x4020 (this register is used to select an operating mode).

IncrementalEncoder1 : Input 1 and Input 2

IncrementalEncoder2 : Input 3 and Input 4

The signals are evaluated with AB quadrature mode and X4 encoding.

Feature	**Description**	**Value**	**EDS**
Name	IncrementalEncoder1	3027	[MxSubExt302x] ParameterName=IncrementalEncoderx ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
	IncrementalEncoder2	3028	
Description	Incremental Encoder Register x		
Object Code	VAR		
Mapping	PDO		
	Manual		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

Sub index 1≦n≦254	**1 Byte**								**Byte 0:**							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

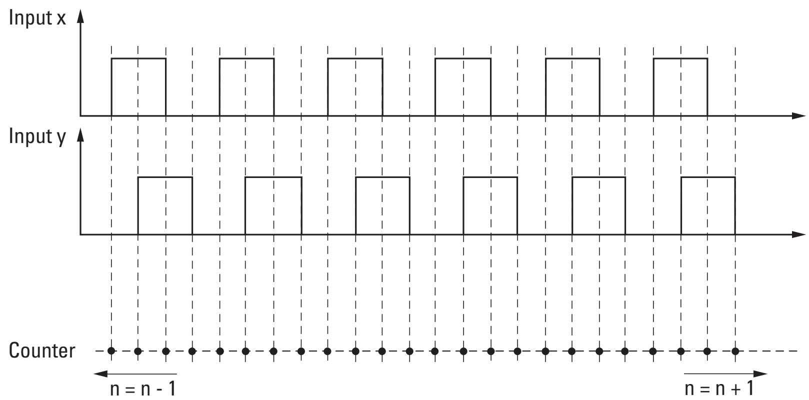


Figure 56: Timing diagram showing how the inputs of the incremental encoder register in the XN-322-20DI-PCNT module count with X4 encoding

7.5.5 Counter Mode Register (Object 0x4020)

Object 0x4020 can be used to configure the counter inputs and, accordingly, select the operating mode you want to use. In addition, any writing command to counter mode register object 0x4020 will reset count registers 0x3023 through 0x3028 to 0x00.

Feature	Description / Value	EDS
Name	Counter Mode Register	[MxSubExt4020] ParameterName=CounterModeRegister ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	CounterModeRegister	
Object Code	VAR	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	$0x00_{hex}$	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Meaning	Notes
0	Input 1/2	0 = Counter Mode 1 = Incremental Encoder Mode	
1	Input 3/4	0 = Counter Mode 1 = Incremental Encoder Mode	
2 – 7		reserved	

7.6 XN-322-20DI-ND

This module supports the provision of data for digital inputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

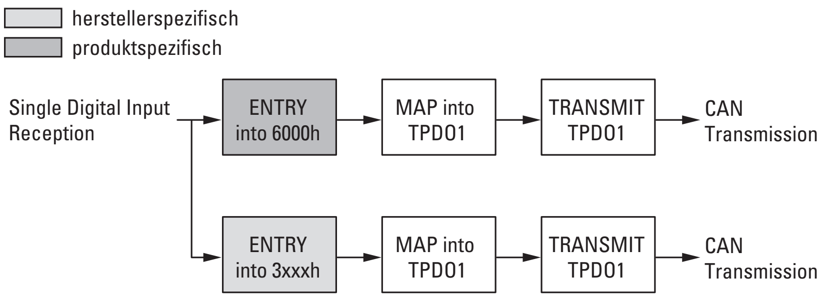


Figure 57: Block diagram showing the various CANopen objects for digital inputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6000	UNSIGNED8	I-BYTE	Digital Input 8-bit	Default	ro	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object 0x6000)"

Manufacturer-specific objects

Index range for the XN-322-20DI-ND module: x130 to x13F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x3130	UNSIGNED32	Input1_20	Read Digital Input 1_20	Manual	ro	PDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO

7.6.1 Read Digital Input 8-Bit (Object 0x6000)

Object 0x6000 represents the digital value of the channels' digital input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-BYTE	[MxSubExt6000] ParameterName=I-BYTE ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=3
Description	Read Digital Input 8-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.6.2 Read Digital Inputs (Object 0x3130)

Object 0x3130 represents the channels' formatted digital input values in a double word.

Feature	Description / Value	EDS
Name	Input1_20	[MxSubExt3130] ParameterName=Input1_20 ObjectType=0x7 DataType=0x0007 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Inputs	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.7 XN-322-4DO-RNO

This module supports the provision of data for digital outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

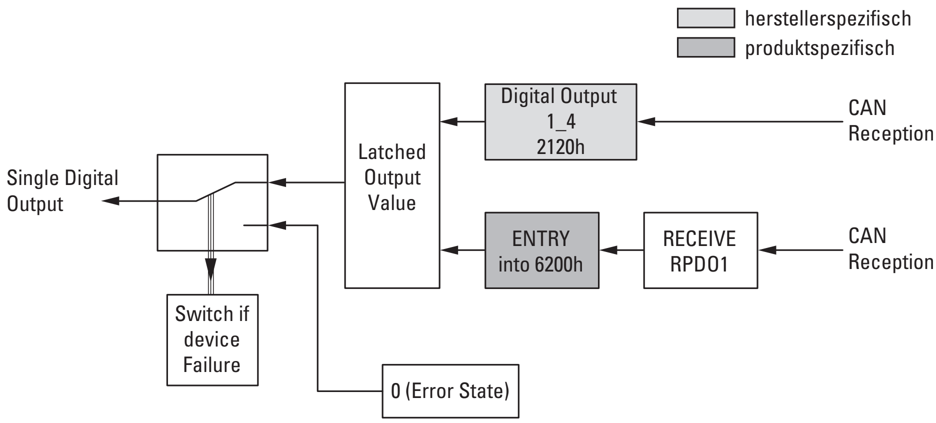


Figure 58: Block diagram showing the various CANopen objects for digital outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6200	UNSIGNED8	Q-BYTE	Write Digital Output 8-bit	Default	rww	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object 0x6000)"

Manufacturer-specific objects

Index range for the XN-322-4DO-RNO module: x120 to x12F

CAN Object Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x2120	UNSIGNED8	Output 1_4	Write Digital Output 1_4	Manual	rww	PDO

CAN Object Index (hex)	Data Type	Name	Function	Mapping	Access	
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO

7.7.1 Write Digital Output 8-bit (Object 0x6200)

Object 0x6200 transmits the digital value of the channels' digital signal outputs. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-Byte	[MxSubExt6200] ParameterName=Q-Byte ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Output 8-Bit	
Object Code	VAR	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.7.2 Write Digital Output (Object 0x2120)

Object 0x2120 transmits the value of the channels' digital signal outputs in a 8-bit word.

Feature	Description / Value	EDS
Name	Output1_4	[MxSubExt2120] ParameterName=Output1_4 ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Outputs	
Object Code	VAR	
Mapping	PDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	–	–	–	–	DO3	DO3	DO2	DO1

7.8 XN-322-8DO-P05

This module supports the provision of data for digital outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

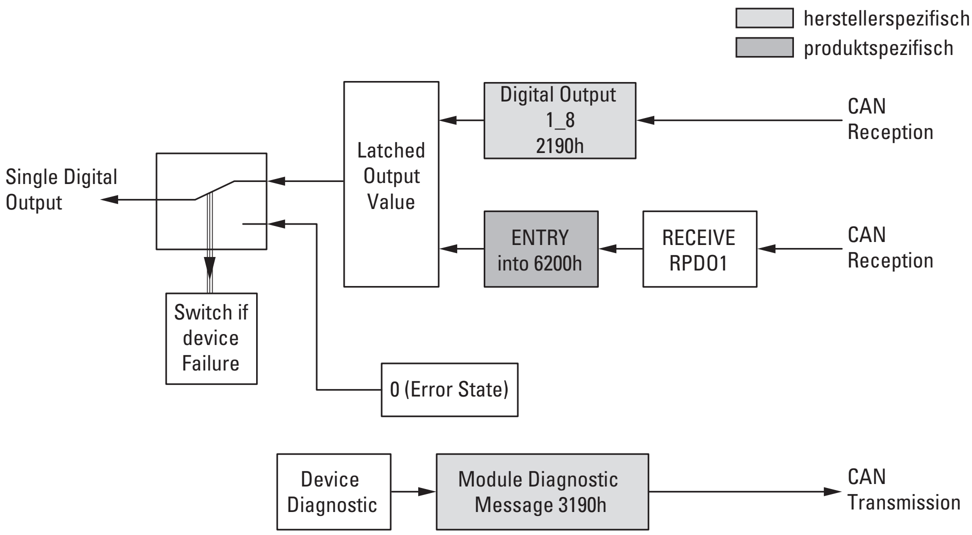


Figure 59: Block diagram showing the various CANopen objects for digital outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6200	UNSIGNED8	Q-BYTE	Write Digital Output 8-bit	Default	rww	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object 0x6000)"

Manufacturer-specific objects

Index range for the XN-322-8DO-P05 module: x190 to x19F

CAN Object Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section “6.2.6 Module Identification Number (Object 0x1027)”, page 76	–	ro	SDO
0x2190	UNSIGNED8	Output 1_8	Write Digital Output 1_8	Manual	rww	PDO
0x3190	UNSIGNED8	InputVoltageState	Input Voltage State Bit 0: DC 24V Output 1..8 OK	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO

7.8.1 Write Digital Output 8-bit (Object 0x6200)

Object 0x6200 transmits the digital value of the channels' digital signal outputs. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-Byte	[MxSubExt6200] ParameterName=Q-Byte ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Output 8-Bit	
Object Code	VAR	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.8.2 Write Digital Output (Object 0x2190)

Object 0x2190 transmits the value of the channels' digital signal outputs in a byte.

Feature	Description / Value	EDS
Name	Output1_8	[MxSubExt2190] ParameterName=Output1_8 ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Outputs	
Object Code	VAR	
Mapping	PDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.8.3 Input Voltage State (Object 0x3190)

Object 0x3190 contains status information on the module's supply voltage:

Bit 0: DC 24V, Output 1 to 8 OK

Feature	Description / Value	EDS
Name	InputVoltageState	[MxSubExt3190] ParameterName=InputVoltageState ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Status of supply voltage	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

Data bit	Designation	Meaning	Notes
0	Power supply, outputs 1–8:	0 = No power 1 = 24 V supply voltage OK at 1+	
1-7		reserved	

7.9 XN-322-12DO-P17

This module supports the provision of data for digital outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

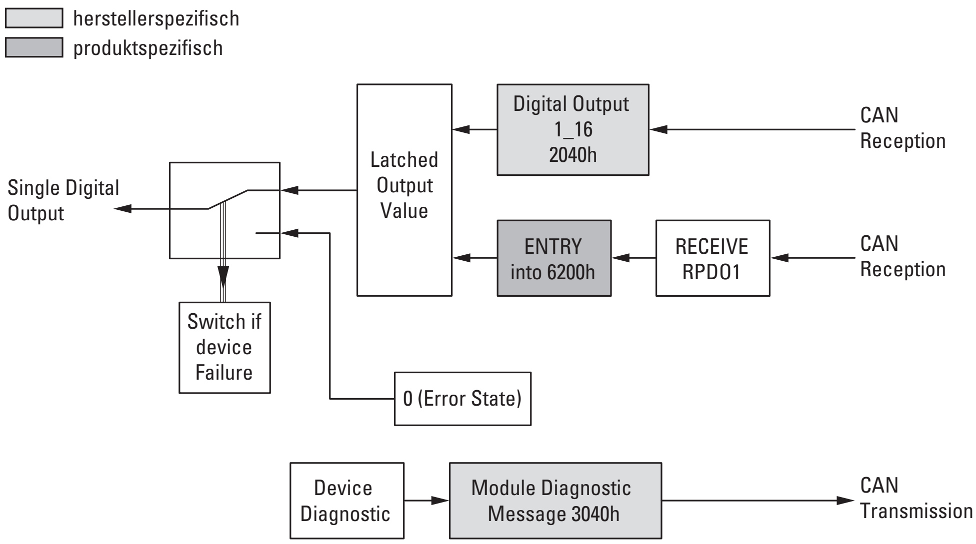


Figure 60: Block diagram showing the various CANopen objects for digital outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6200	UNSIGNED8	Q-BYTE	Write Digital Output 8-Bit	Default	rww	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object 0x6000)“

Manufacturer-specific objects

Index range for the XN-322-12DO-P17 module: x040 to x04F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section “6.2.6 Module Identification Number (Object 0x1027)”, page 76	–	ro	SDO
0x2040	UNSIGNED16	Output1_12	Write Digital Output 1-12	Manual	rww	PDO
0x3040	UNSIGNED8	InputVoltageState	Input Voltage State Bit 0: DC 24V Output 1..4 OK Bit 1: DC 24V Output 5..8 OK Bit 2: DC 24V Output 9..12 OK	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO

7.9.1 Write Digital Output 8-bit (Object 0x6200)

Object 0x6200 transmits the digital value of the channels' digital signal outputs. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-Byte	[MxSubExt6200] ParameterName=Q-Byte ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=2
Description	Write Digital Output 8-Bit	
Object Code	VAR	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≤n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.9.2 Write Digital Output (Object 0x2040)

Object 0x2040 transmits the value of the channels' digital signal outputs in a 16-bit word.

Feature	Description / Value	EDS
Name	Output1_12	[MxSubExt2040] ParameterName=Output1_12 ObjectType=0x7 DataType=0x0006 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Outputs	
Object Code	VAR	
Mapping	PDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	0000_{hex}	

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.9.3 InputVoltageState(Object 0x3040)

Object 0x3040 contains information on the module's power supply.

Feature	Description / Value	EDS
Name	InputVoltageState	[MxSubExt3040] ParameterName=InputVoltageState ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Input Voltage State	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

Data bit	Designation	Meaning	Notes
0	Power supply, outputs 1–4:	0 = No power 1 = Power OK (24 V)	
1	Power supply, outputs 5–8:	0 = No power 1 = Power OK (24 V)	
2	Power supply, outputs 9–12:	0 = No power 1 = Power OK (24 V)	
3-7		reserved	

7.10 XN-322-16DO-P05

This module supports the provision of data for digital outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

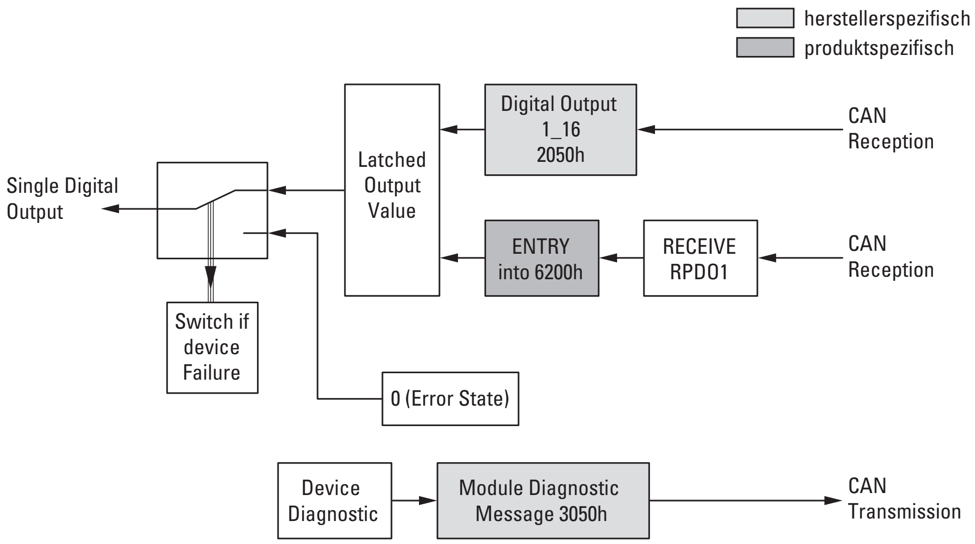


Figure 61: Block diagram showing the various CANopen objects for digital outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6200	UNSIGNED8	Q-BYTE	Write Digital Output 8-bit	Default	rww	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object 0x6000)“

Manufacturer-specific objects

Index range for the XN-322-16DO-P05 module: x050 to x05F

CAN Object Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x2050	UNSIGNED16	Output 1_16	Write Digital Output 1_16	Manual	rww	PDO
0x3050	UNSIGNED8	InputVoltageState	Input Voltage State Bit 0: DC 24V Output 1..8 OK Bit 1: DC 24V Output 9..16 OK	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO

7.10.1 Write Digital Output 8-bit (Object 0x6200)

Object 0x6200 transmits the digital value of the channels' digital signal outputs. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-Byte	[MxSubExt6200] ParameterName=Q-Byte ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=2
Description	Write Digital Output 8-Bit	
Object Code	VAR	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≤n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.10.2 Write Digital Output (Object 0x2050)

Object 0x2050 transmits the value of the channels' digital signal outputs in a 16-bit word.

Feature	Description / Value	EDS
Name	Output1_16	[MxSubExt2050] Parameter-Name=Output1_16 ObjectType=0x7 DataType=0x0006 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Outputs	
Object Code	VAR	
Mapping	PDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	0000_{hex}	

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.10.3 Input Voltage State (Object 0x3050)

Object 0x3050 contains status information on the module's supply voltage:

Bit 0: DC 24V, Output 1 to 8 OK

Bit 1: DC 24V, Output 9 to 16 OK

Feature	Description / Value	EDS
Name	InputVoltageState	[MxSubExt3050] ParameterName=InputVoltageState ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Status of supply voltage	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

Data bit	Designation	Description	Notes
0	Power supply, outputs 1–8:	0 = No power 1 = Power OK (24 V)	
1	Power supply, outputs 9–16:	0 = No power 1 = Power OK (24 V)	
2-7		reserved	

7.11 XN-322-8DIO-PD05

This module supports the provision of data for digital inputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

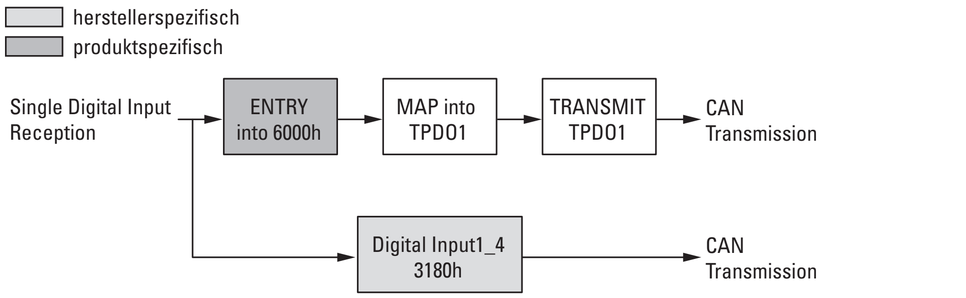


Figure 62: Block diagram showing the various CANopen objects for digital inputs

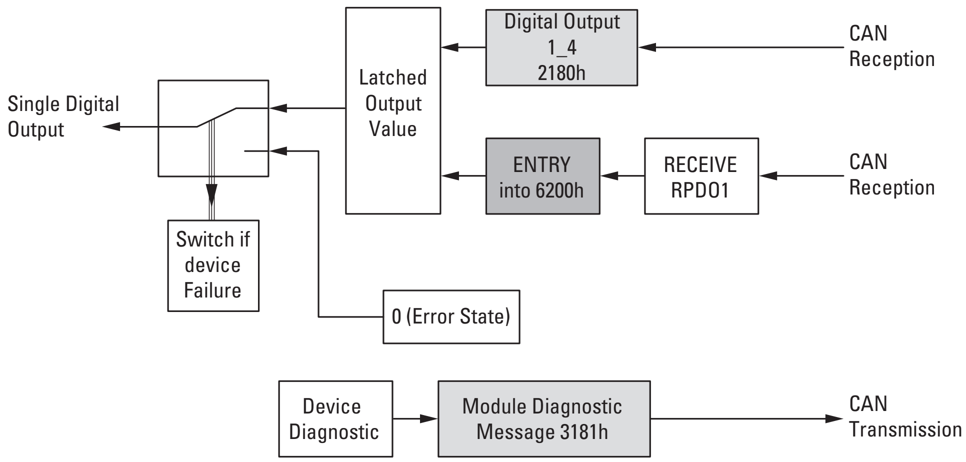


Figure 63: Block diagram showing the various CANopen objects for digital outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6000	UNSIGNED8	I-BYTE	Digital Input 8-bit	Default	ro	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object 0x6000)“
0x6200	UNSIGNED8	Q-BYTE	Write Digital Output 8-bit	Default	rww	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object 0x6000)“

Manufacturer-specific objects

Index range for the XN-322-8DIO-PD05 module: x180 to x18F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module ID number → Section "6.2.6 Module Identification Number (Object 0x1027)"	–	ro	SDO
0x2180	UNSIGNED8	Output 1_4	Write Digital Output 1_4	Manual	rww	PDO
0x3180	UNSIGNED8	Input1_4	Read Digital Output 1_4	Manual	ro	PDO
0x3181	UNSIGNED8	InputVoltageState	Input Voltage State	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO

7.11.1 Read Digital Input 8-Bit (Object 0x6000)

Object 0x6000 represents the digital value of the channels' digital input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-BYTE	[MxSubExt6000] ParameterName=I-BYTE ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Input 8-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.11.2 Write Digital Output 8-bit (Object 0x6200)

Object 0x6200 transmits the digital value of the channels' digital signal outputs. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-Byte	[MxSubExt6200] ParameterName=Q-Byte ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Output 8-Bit	
Object Code	VAR	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.11.3 Write Digital Output (Object 0x2180)

Object 0x2180 transmits the value of the channels' digital signal outputs in a byte.

Feature	Description / Value	EDS
Name	Output1_4	[MxSubExt2180] ParameterName=Output1_4 ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Outputs	
Object Code	VAR	
Mapping	PDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	0000_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
					DO4	DO3	DO2	DO1

7.11.4 Read Digital Inputs (Object 0x3180)

Object 0x3180 represents the channels' formatted digital input values in a byte.

Feature	Description / Value	EDS
Name	Input1_4	[MxSubExt3180] ParameterName=Input1_4 ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Inputs	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
					DI4	DI3	DI2	DI1

7.11.5 Input Voltage State (Object 0x3181)

Object 0x3181 contains status information on the module's supply voltage:

Feature	Description / Value	EDS
Name	InputVoltageState	[MxSubExt3181] ParameterName=InputVoltageState ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Status of supply voltage	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

Data bit	Designation	Meaning	Notes
0	01 to 04	0 = No power 1 = Power OK (24 V)	
1-7		reserved	

7.12 XN-322-16DIO-PD05

This module supports the provision of data for digital inputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

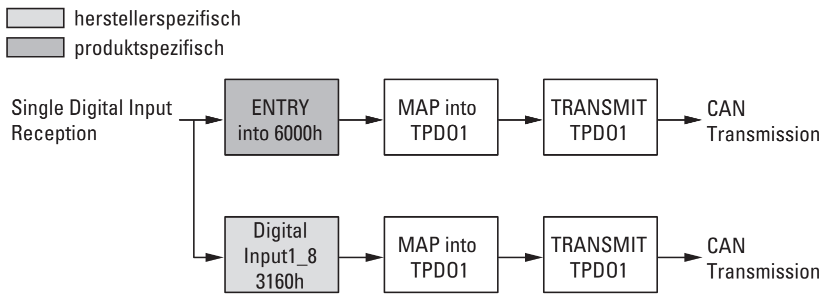


Figure 64: Block diagram showing the various CANopen objects for digital inputs

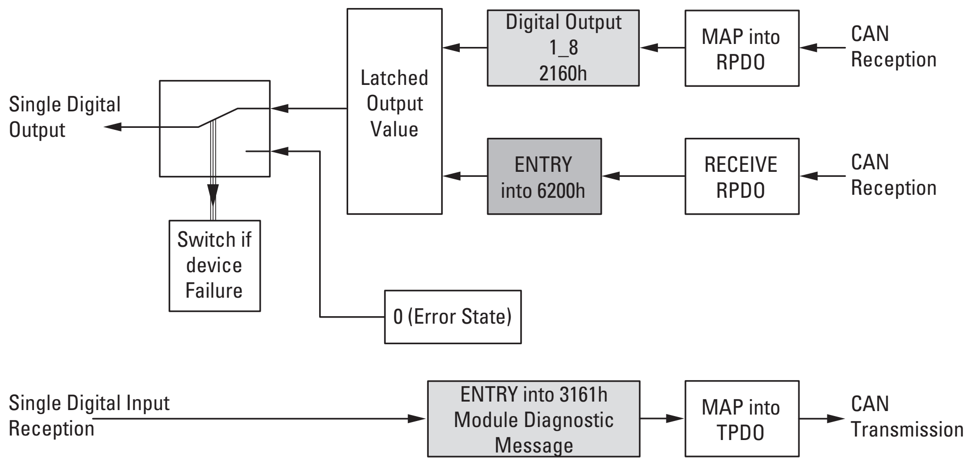


Figure 65: Block diagram showing the various CANopen objects for digital outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6000	UNSIGNED8	I-BYTE	Digital Input 8-bit	Default	ro	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object 0x6000)"
0x6200	UNSIGNED8	Q-BYTE	Write Digital Output 8-bit	Default	rww	PDO	→ Section "7.12.2 Write Digital Output 8-bit (Object 0x6200)"

Manufacturer-specific objects

Index range for the XN-322-16DIO-PD05 module: x160 to x16F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module ID number → Section "6.2.6 Module Identification Number (Object 0x1027)“	–	ro	SDO
0x2160	UNSIGNED8	Output 1_8	Write Digital Output 1_8	Manual	rww	PDO
0x3160	UNSIGNED8	Input1_8	Read Digital Output 1_8	Manual	ro	PDO
0x3161	UNSIGNED8	InputVoltageState	Input Voltage State	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO

7.12.1 Read Digital Input 8-Bit (Object 0x6000)

Object 0x6000 represents the digital value of the channels' digital input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-BYTE	[MxSubExt6000] ParameterName=I-BYTE ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Input 8-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.12.2 Write Digital Output 8-bit (Object 0x6200)

Object 0x6200 transmits the digital value of the channels' digital signal outputs. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-Byte	[MxSubExt6200] ParameterName=Q-Byte ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Output 8-Bit	
Object Code	VAR	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.12.3 Write Digital Output (Object 0x2160)

Object 0x2160 transmits the value of the channels' digital signal outputs in a byte.

Feature	Description / Value	EDS
Name	Output1_8	[MxSubExt2160] ParameterName=Output1_8 ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Outputs	
Object Code	VAR	
Mapping	PDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1

7.12.4 Read Digital Inputs (Object 0x3160)

Object 0x3160 represents the channels' formatted digital input values in a byte.

Feature	Description / Value	EDS
Name	Input1_8	[MxSubExt3160] ParameterName=Input1_8 ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Inputs	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

7.12.5 Input Voltage State (Object 0x3161)

Object 0x3161 contains status information on the module's supply voltage.

Feature	Description / Value	EDS
Name	InputVoltageState	[MxSubExt3161] ParameterName=InputVoltageState ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Status of supply voltage	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

Data bit	Designation	Meaning	Notes
0	01 to 08	0 = No power 1 = Power OK (24 V)	
1-7		reserved	

7.13 XN-322-16DIO-PC05

This module supports the provision of data for digital inputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

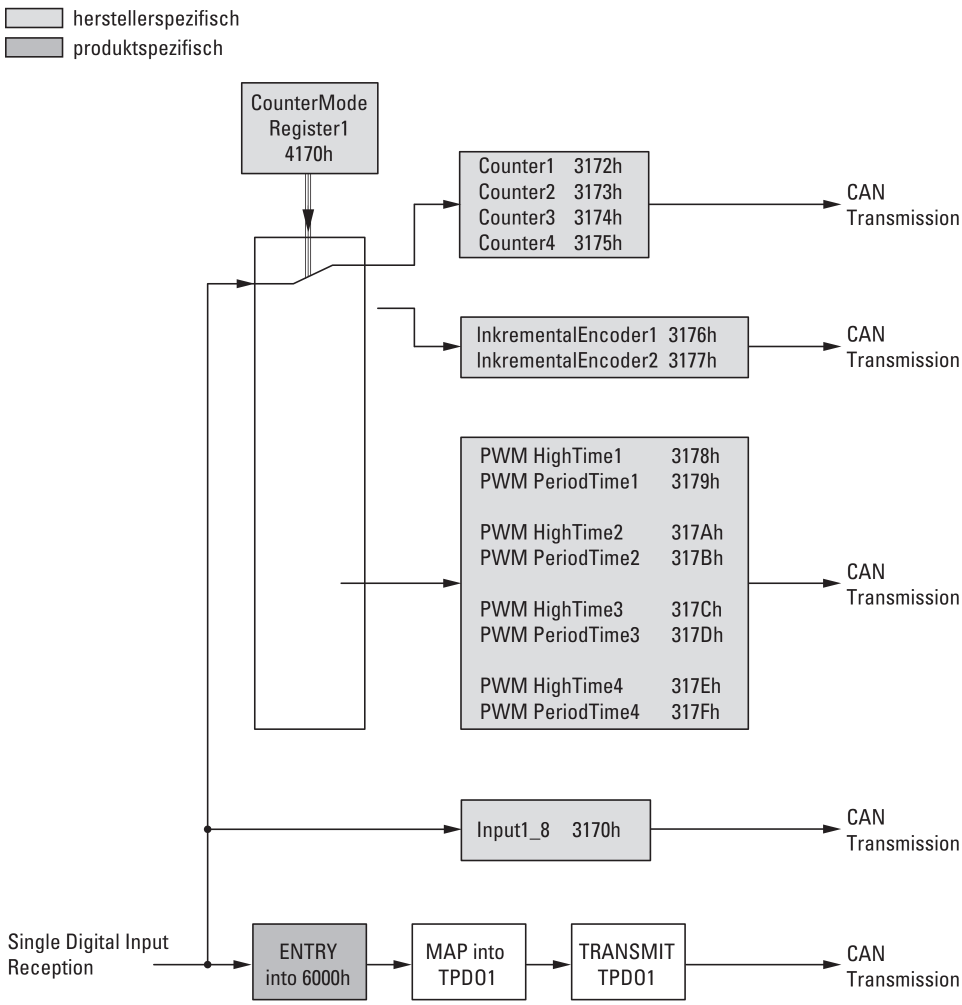


Figure 66: Block diagram showing the various CANopen objects for digital inputs

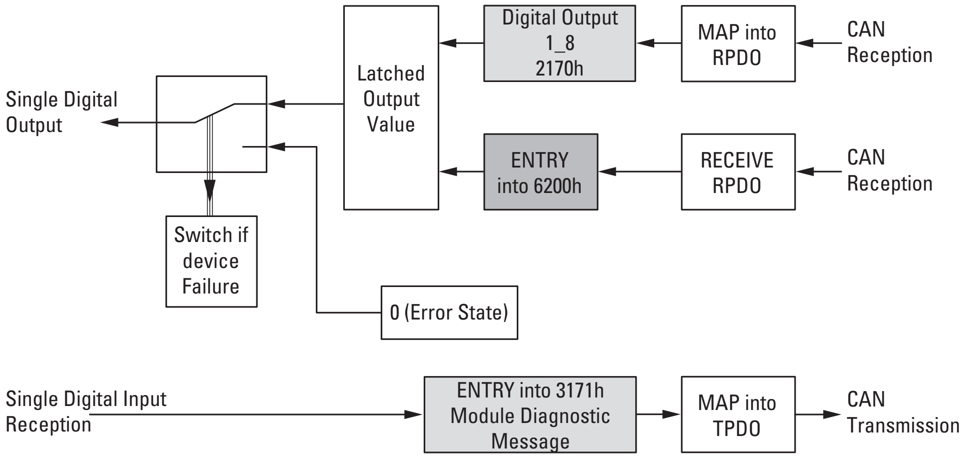


Figure 67: Block diagram showing the various CANopen objects for digital outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access		
0x6000	UNSIGNED8	I-BYTE	Digital Input 8-bit	Default	ro	PDO	→ Section "7.1.1 Read Digital Input 8-Bit (Object 0x6000)"
0x6200	UNSIGNED8	Q-BYTE	Write Digital Output 8-bit	Default	rww	PDO	→ Section "7.12.2 Write Digital Output 8-bit (Object 0x6200)"

Manufacturer-specific objects

Index range for the XN-322-16DIO-PC05 module: x170 to x17F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module ID number → Section "6.2.6 Module Identification Number (Object 0x1027)"	–	ro	SDO
0x2170	UNSIGNED8	Output 1_8	Write Digital Output 1_8	Manual	rww	PDO
0x3170	UNSIGNED8	Input1_8	Digital input channels 1 to 8	Manual	ro	PDO
0x3171	UNSIGNED8	InputVoltageState	Input Voltage State	Manual	ro	PDO
0x3172	UNSIGNED8	Counter1	Counter Register 1	Manual	ro	PDO
0x3173	UNSIGNED8	Counter2	Counter Register 2	Manual	ro	PDO
0x3174	UNSIGNED8	Counter3	Counter Register 3	Manual	ro	PDO
0x3175	UNSIGNED8	Counter4	Counter Register 4	Manual	ro	PDO
0x3176	UNSIGNED16	Incremental Encoder1	Incremental Encoder Register 1/2	Manual	ro	PDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x3177	UNSIGNED16	Incremental Encoder2	Incremental Encoder Register 3/4	Manual	ro	PDO
0x3178	UNSIGNED16	PWMHighTime1	PWM High Time 1	Manual	ro	PDO
0x3179	UNSIGNED16	PWMPeriod1	PWM Period1	Manual	ro	PDO
0x317A	UNSIGNED16	PWMHighTime2	PWM High Time 2	Manual	ro	PDO
0x317B	UNSIGNED16	PWMPeriod2	PWM Period2	Manual	ro	PDO
0x317C	UNSIGNED16	PWMHighTime3	PWM High Time 3	Manual	ro	PDO
0x317D	UNSIGNED16	PWMPeriod3	PWM Period3	Manual	ro	PDO
0x317E	UNSIGNED16	PWMHighTime4	PWM High Time 4	Manual	ro	PDO
0x317F	UNSIGNED16	PWMPeriod4	PWM Period4	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO
0x4170	UNSIGNED8	CounterModeRegister	Counter Mode Register	–	ro	SDO

→ Make sure to only use the data relevant to the selected operating mode. If applicable, content in non-relevant registers may also change. The operating mode can be selected using the counter mode register.

7.13.1 Read Digital Input 8-Bit (Object 0x6000)

Object 0x6000 represents the digital value of the channels' digital input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-BYTE	[MxSubExt6000] ParameterName=I-BYTE ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Input 8-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.13.2 Write Digital Output 8-bit (Object 0x6200)

Object 0x6200 transmits the digital value of the channels' digital signal outputs. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-Byte	[MxSubExt6200] ParameterName=Q-Byte ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Output 8-Bit	
Object Code	VAR	
Mapping	PDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.13.3 Write Digital Output (Object 0x2170)

Object 0x2170 transmits the value of the channels' digital signal outputs in a byte.

Feature	Description / Value	EDS
Name	Output1_8	[MxSubExt2170] ParameterName=Output1_8 ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Outputs	
Object Code	VAR	
Mapping	PDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1

7.13.4 Read Digital Inputs (Object 0x3170)

Object 0x3170 represents the channels' formatted digital input values in a byte.

Feature	Description / Value	EDS
Name	Input1_8	[MxSubExt3170] ParameterName=Input1_8 ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Inputs	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

7.13.5 Input Voltage State (Object 0x3171)

Object 0x3171 contains status information on the module's supply voltage.

Feature	Description / Value	EDS
Name	InputVoltageState	[MxSubExt3171] ParameterName=InputVoltageState ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Status of supply voltage	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

Data bit	Designation	Description	Notes
0	01 to 08	0 = No power 1 = Power OK (24 V)	
1-7		reserved	

7.13.6 Counter Register (Object 0x3172-0x3175)

Objects 0x3172 to 0x3175 are 8-bit count registers assigned to the corresponding inputs. Any writing command to counter mode register object 0x4170 will result in counter registers 0x3172 through 0x3177 being reset to 0x00.

> *ACHTUNG NOTICE*
> Writing to counter mode register 0x4170 (which is used to select an operating mode) will reset the count registers.

Feature	Description	Value	EDS
Name	Counter1	3172	[MxSubExt317x] ParameterName=Counterx ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
	Counter2	3173	
	Counter3	3174	
	Counter4	3175	
Description	Counter Register Input x		
Object Code	VAR		
Mapping	PDO		
	Manual		
Data type	UNSIGNED8		
Sub index	01 … FE_{hex}		
Access	ro		

Data byte structure for objects 0x3172 to 0x3175:

Sub index 1≤n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.13.7 Incremental Encoder Register (Object 0x3176 bis 0x3177)

Objects 0x3176 to 0x3177 are 16-bit count registers assigned to the corresponding inputs.

Any writing command to counter mode register 0x4170 will result in all the count registers object 0x3172 to 0x3177 being reset to 0x00.

ACHTUNG NOTICE

To reset the count registers, write to counter mode register 0x4170 (this register is used to select an operating mode).

IncrementalEncoder1 : Input 1 and Input 2

IncrementalEncoder2 : Input 3 and Input 4

The signals are evaluated with AB quadrature mode and X4 encoding.

Feature	Description	Value	EDS
Name	IncrementalEncoder1	3176	[MxSubExt317x] ParameterName=IncrementalEncoderx ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
	IncrementalEncoder2	3177	
Description	Incremental Encoder Register x		
Object Code	VAR		
Mapping	PDO		
	Manual		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

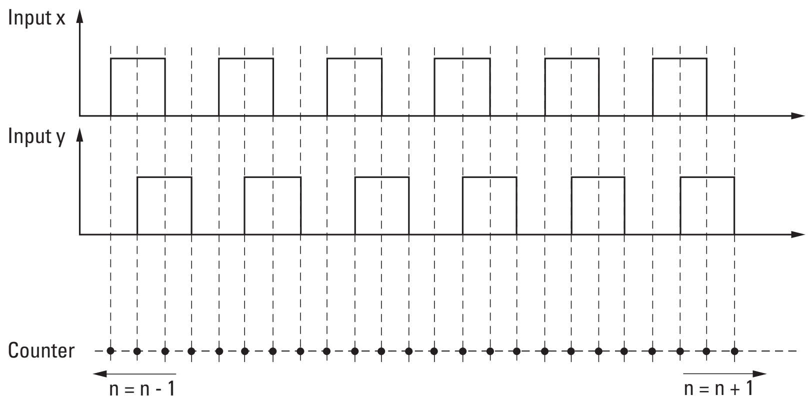


Figure 68: Timing diagram showing how inputs (x/y) = (1/2) or (3/4) of the incremental encoder register in the XN-322-16DIO-PC05 module count with X4 encoding

7.13.8 PWM High Time (Object 0x3178, 0x317A, 0x317C, 0x317E)

Objects 0x3178, 0x317A, 0x317C, 0x317E are 16-bit count registers used to record the high time for the signal present at the corresponding inputs.

This high time is the time that passes between the rising and falling edges of the signal at the digital input. When a rising signal edge is detected, a counter will start being incremented every µs. Then, when the corresponding falling signal edge is detected, the counter value will be recorded in the corresponding 16-bit PwmHighTime(x) count register. Once the value is transferred to PwmHighTime(x), the counter will be reset.

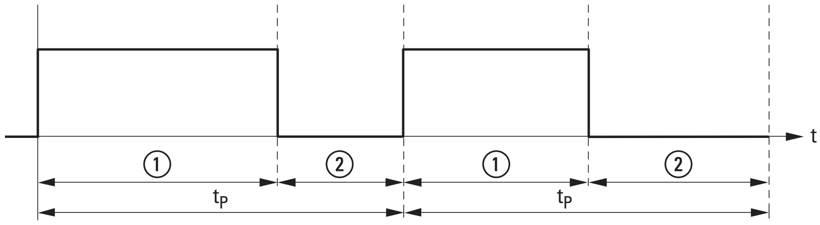


Figure 69: PWM signal measurement

① High Time

② Low Time

Feature	Description	Value	EDS
Name	PwmHighTime1	0x3178	[MxSubExt317x] ParameterName=PwmHighTimex ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
	PwmHighTime2	0x317A	
	PwmHighTime3	0x317C	
	PwmHighTime4	0x317E	
Description	PWM High Time x		
Object Code	VAR		
Mapping	PDO		
	Manual		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.13.9 PWM Period Time (Object 0x3179, 0x317B, 0x317D, 0x317F)

Objects 0x3179, 0x317B, 0x317D, 0x317F are 16-bit count registers used to record the period for the signal present at the corresponding inputs.

Period t_p will be the time that passes between the rising edges of the signal at the digital input. When the first rising signal edge is detected, a counter will start being incremented every µs. Then, when the second rising signal edge is detected, the counter value will be recorded in the corresponding 16-bit PwmPeriodTime(x) counter register. Once the value is transferred to PwmPeriodTime(x), the counter will be reset.

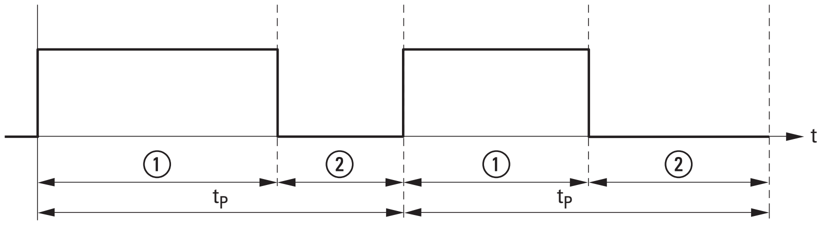


Figure 70: PWM signal measurement

① High Time
② Low Time

Feature	**Description**	**Value**	**EDS**
Name	PwmPeriodTime1	0x3179	[MxSubExt317x] ParameterName=PwmPeriodTimex ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
	PwmPeriodTime2	0x317B	
	PwmPeriodTime3	0x317D	
	PwmPeriodTime4	0x317F	
Description	PWM Period Time x		
Object Code	VAR		
Mapping	PDO		
	Manual		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

Sub index 1≤n≤254	**1 Byte**								**Byte 0:**							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.13.10 Counter Mode Register (Object 0x4170)

Object 0x4170 can be used to configure the counter inputs and, accordingly, select the operating mode you want to use. In addition, any writing command to counter mode register object 0x4170 will reset count registers 0x3172 through 0x3177 to 0x00.

Feature	Description / Value	EDS
Name	CounterModeRegister1	[MxSubExt4170] ParameterName=CounterModeRegister1 ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	Counter Mode Register 1	
Object Code	VAR	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	$0x00_{hex}$	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–	–	Input 3/4		Input 1/2	

Bit B1	Bit B0	Designation	Description
0	0	Input 1/2	Counter Mode
0	1		Incremental Encoder Mode
1	1		PWM Time Measuring Mode
Bit B3	**B2**	**Designation**	**Description**
0	0	Input 3/4	Counter Mode
0	1		Incremental Encoder Mode
1	1		PWM Time Measuring Mode

7.14 XN-322-4AI-PTNI

This module supports the provision of data for analog inputs and analog outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

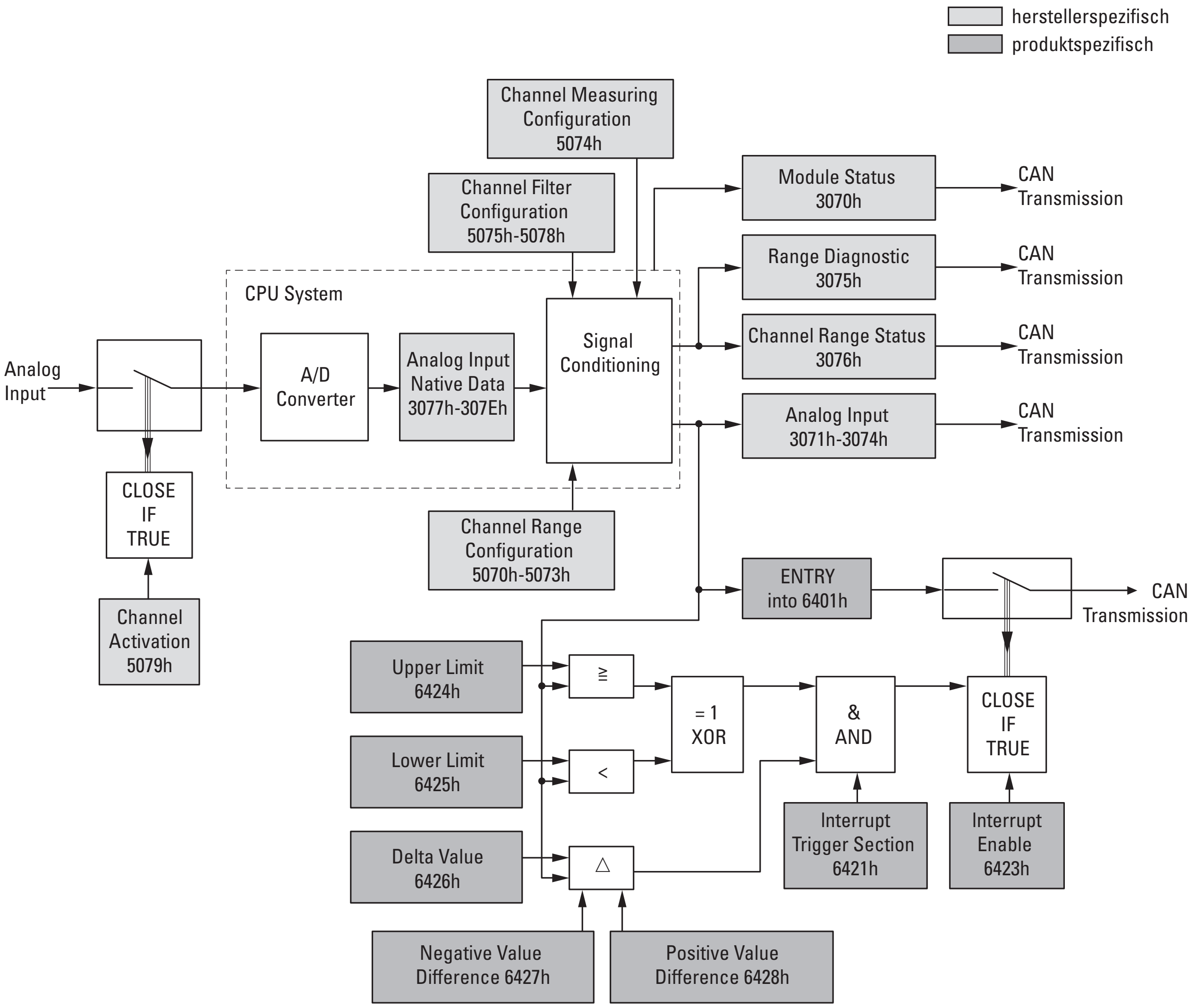


Figure 71: Block diagram showing the various CANopen objects for analog inputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x6401	INTEGER16	I-WORD	Read Analog Input 16-bit	Default	ro	PDO
0x6421	UNSIGNED8	AI_INTERRUPT_TRIGGER_SELECTION	Analog Input Interrupt Trigger Selection	-	rw	SDO
0x6423	BOOLEAN	AnalogInputGlobalInterruptEnable	Analog Input Global Interrupt Enable	-	rw	SDO
0x6424	INTEGER32	AI_UPPER_LIMIT	Analog Input Interrupt Upper Limit Integer	-	rw	SDO
0x6425	INTEGER32	AI_LOWER_LIMIT	Analog Input Interrupt Lower Limit Integer	-	rw	SDO
0x6426	UNSIGNED32	AI_DELTA_VALUE	Analog Input Interrupt Delta Unsigned	-	rw	SDO
0x6427	UNSIGNED32	AI_NEGATIVE_DELTA_VALUE	Analog Input Interrupt Negative Delta Unsigned	-	rw	SDO
0x6428	UNSIGNED32	AI_POSITIVE_DELTA_VALUE	Analog Input Interrupt Positive Delta Unsigned	-	rw	SDO

Manufacturer-specific objects

Index range for the XN-322-4AI-PTNI module: x070 to x07F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section “6.2.6 Module Identification Number (Object 0x1027)”, page 76	–	ro	SDO
0x3070	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x3071	INTEGER16	InputChannel1	Input Channel 1	Manual	ro	PDO
0x3072	INTEGER16	InputChannel2	Input Channel 2	Manual	ro	PDO
0x3073	INTEGER16	InputChannel3	Input Channel 3	Manual	ro	PDO
0x3074	INTEGER16	InputChannel4	Input Channel 4	Manual	ro	PDO
0x3075	UNSIGNED8	WireBreakDiag	Wire Break Diagnostic Messages	Manual	ro	PDO
0x3076	UNSIGNED8	RangeDiag	Range Diagnostic Message	Manual	ro	PDO
0x3077	INTEGER16	NativeDataAI1	Analog Input 1 Native Data	Manual	ro	PDO
0x3078	INTEGER16	NativeDataAI2	Analog Input 2 Native Data	Manual	ro	PDO
0x3079	INTEGER16	NativeDataAI3	Analog Input 3 Native Data	Manual	ro	PDO
0x307A	INTEGER16	NativeDataAI4	Analog Input 4 Native Data	Manual	ro	PDO
0x307B	INTEGER16	NativeDataAI5	Analog Input 5 Native Data	Manual	ro	PDO
0x307C	INTEGER16	NativeDataAI6	Analog Input 6 Native Data	Manual	ro	PDO
0x307D	INTEGER16	NativeDataAI7	Analog Input 7 Native Data	Manual	ro	PDO
0x307E	INTEGER16	NativeDataAI8	Analog Input 8 Native Data	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO

0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	-	ro	SDO
0x4070	UNSIGNED16	FirmwareVersion	Firmware Version	-	ro	SDO
0x5070	UNSIGNED8	SensorSelectChannel1	Sensor Type Selection Channel 1	-	rw	SDO
0x5071	UNSIGNED8	SensorSelectChannel2	Sensor Type Selection Channel 2	-	rw	SDO
0x5072	UNSIGNED8	SensorSelectChannel3	Sensor Type Selection Channel 3	-	rw	SDO
0x5073	UNSIGNED8	SensorSelectChannel4	Sensor Type Selection Channel 4	-	rw	SDO
0x5074	UNSIGNED8	ChannelMeasuringConfig	Channel Measuring Configuration (two-wire/three-wire measurement)	-	rw	SDO
0x5075	UNSIGNED16	FilterConfigChannel1	Filter Configuration Channel 1	-	rw	SDO
0x5076	UNSIGNED16	FilterConfigChannel2	Filter Configuration Channel 2	-	rw	SDO
0x5077	UNSIGNED16	FilterConfigChannel3	Filter Configuration Channel 3	-	rw	SDO
0x5078	UNSIGNED16	FilterConfigChannel4	Filter Configuration Channel 4	-	rw	SDO
0x5079	UNSIGNED8	ChannelActivation	Channel Activation	-	rw	SDO

7.14.1 Read Analog Input 16-Bit (Object 0x6401)

Object 0x6401 represents the channels' formatted analog input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Read Analog Input 16-Bit	[MxSubExt6401] ParameterName=I-WORD ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=4
Description	I-WORD	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE hex	
Access	ro	
Default value	0 x 0000_{hex}	

Design of the data bytes:

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

7.14.2 Analog Input Interrupt Trigger Selection (Object 0x6421)

Object 0x6421 defines which events will trigger an interrupt for the corresponding channel so that the channel's analog input value (object 0x6401) will be transmitted when the event occurs.

Feature	Description / Value	EDS
Name	AI_INTERRUPT_TRIGGER_SELECTION	[MxSubExt6421] Parameter-Name=AI_INTERRUPT_TRIGGER_SELECTION ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=7 LowLimit=0 HighLimit=31 PDOMapping=0 Count=4
Description	Analog Input Interrupt Trigger Selection	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	$0x07_{hex}$	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Description	Notes
0	0 = Upper limit not exceeded 1 = Upper limit exceeded	
1	0 = Input not below lower limit 1 = Input below lower limit	
2	0 = Input not changed by more than delta 1 = Input changed by more than delta	
3	0 = Input not reduced by more than negative delta 1 = Input reduced by more than negative d	
4	0 = Input not reduced by more than positive delta 1 = Input reduced by more than positive delta	
5 – 7	reserved	

7.14.3 Analog Input Global Interrupt Enable (Object 0x6423)

Object 0x6423 enables and disables the transmission of data via object 0x6401 in general.

Feature	Description / Value	EDS
Name	AnalogInputGlobalInterruptEnable	[MxFixed6423] ParameterName=AnalogInputGlobalInterruptEnable ObjectType=0x7 DataType=0x0001 AccessType=rw DefaultValue=0 PDOMapping=0
Description	Analog Input Global Interrupt Enable	
Object Code	Variable	
Mapping	SDO	
Data type	BOOLEAN	
Access	rw	
Default value	FALSE	
Object Code	Variable	

Default value

- FALSE (0)
  Object 0x6401 is not enabled and cannot transmit any analog input values.
- TRUE (1)
  Object 0x6401 is enabled and can transmit analog input values.

7.14.4 Analog Input Interrupt Upper Limit Integer (Object 0x6424)

Object 0x6424 defines an upper limit and affects the transmission of object 0x6401 accordingly → Figure 71, page 164.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) is equal to or greater than the upper limit (Upper Limit 0x6424).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_UPPER_LIMIT	[MxSubExt6424] ParameterName=AI_UPPER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=4
Description	Analog Input Interrupt Upper Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.14.5 Analog Input Interrupt Lower Limit Integer (Object 0x6425)

Object 0x6425 defines a lower limit and affects the transmission of object 0x6401 accordingly → Figure 71, page 164.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) falls below the lower limit (Lower Limit 0x6425).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_LOWER_LIMIT	[MxSubExt6425] ParameterName=AI_LOWER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=4
Description	Analog Input Interrupt Lower Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	$0000\ 0000_{hex}$	

Design of the data bytes:

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

Sub index 1≤n≤254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.14.6 Analog Input Interrupt Delta Unsigned (Object 0x6426)

Object 0x6426 defines the fluctuation range (delta) relative to the last transmitted value both when counting up and down and affects the transmission of object 0x6401 accordingly → Figure 71, page 164.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The difference between the current value (ENTRY into 0x6401) and the last transmitted value is equal to or greater than the fluctuation range (Value Difference 0x6426).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_DELTA_VALUE	[MxSubExt6426] ParameterName=AI_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=4
Description	Analog Input Interrupt Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01... FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≤n≤254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.14.7 Analog Input Interrupt Negative Delta Unsigned (Object 0x6427)

Object 0x6427 defines the negative fluctuation range (delta) relative to the last transmitted value when counting down and affects the transmission of object 0x6401 accordingly → Figure 71, page 164.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is less than the last transmitted value and the difference is greater than or equal to the fluctuation range (Negative Value Difference 0x6427).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_NEGATIVE_DELTA_VALUE	[MxSubExt6427] Parameter-Name=AI_NEGATIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=4
Description	Analog Input Interrupt Negative Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE $_{ex}$	
Access	rw	
Default value	00000000$_{hex}$	

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.14.8 Analog Input Interrupt Positive Delta Unsigned (Object 0x6428)

Object 0x6428 defines the negative fluctuation range (delta) relative to the last transmitted value when counting up and affects the transmission of object 0x6401 accordingly → Figure 71, page 164.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is greater than the last transmitted value. The difference's magnitude is greater than the fluctuation range (Positive Value Difference 0x6428).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

If the current value is less than the last transmitted value, object 0x6401 will be transmitted. If this behavior is not desired, object 0x6421 must be enabled and disabled accordingly.

Feature	Description / Value	EDS
Name	AI_POSITIVE_DELTA_VALUE	[MxSubExt6428] Parameter-Name=AI_POSITIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=4
Description	Analog Input Interrupt Positive Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															LSB

7.14.9 Module Diagnostic Messages (Object 0x3070)

Object 0x3070 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	Module Diagnostic Messages	[MxSubExt3070] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	ModuleDiag	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–					

Data bit	Description
0	reserved
1	0 = sync OK 1 = lno sync
2	0 = OK 1 = FLASH Data CRC Error
3	0 = OK 1 =RAM Data CRC Error
4	0 = OK 1 = Inconsistent FLASH Data
5 – 7	reserved

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Meaning
8 – 15		reserved

7.14.10 Input Channel x (Object 0x3071 to 0x3074)

Objects 0x3071 to 0x3074 contain the formatted (integer) analog input values.

Feature	Description / Value		EDS
Name	Input Channel x		[M13SubExt307x] ParameterName=InputChannelx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
Description	InputChannel1	Object 0x3071	
	InputChannel2	Object 0x3072	
	InputChannel3	Object 0x3073	
	InputChannel4	Object 0x3074	
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.14.11 Wire Break Diagnostic Messages (Object 0x3075)

Object 3075 contains status information on the module's channels.

Feature	Description / Value	EDS
Name	WireBreakDiag	[MxSubExt3075] ParameterName=WireBreakDiag ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Wire Break Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description	Notes
0		0 = Channel 1 OK 1 = Channel 1 wire breakage	
1		0 = Channel 2 OK 1 = Channel 2wire breakage	
2		0 = Channel 3 OK 1 = Channel 3 wire breakage	-
3		0 = Channel 4 OK 1 = Channel 4 wire breakage	
4 – 7		reserved	

7.14.12 Range Diagnostic Messages (Object 0x3076)

Object 0x3076 contains status information on the channels' measurements.

Feature	Description / Value	EDS
Name	Range Diagnostic Messages	[MxSubExt3076] ParameterName=RangeDiag ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	RangeDiag	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	0000_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Meaning
0		0 = Channel 1 OK 1 = Channel 1 over range
1		0 = Channel 2 OK 1 = Channel 2 over range
2		0 = Channel 3 OK 1 = Channel 3 over range
3		0 = Channel 4 OK 1 = Channel 4 over range
4		0 = Channel 5 OK 1 = Channel 5 over range
5		0 = Channel 6 OK 1 = Channel 6 over range
6		0 = Channel 7 OK 1 = Channel 7 over range
7		0 = Channel 8 OK 1 = Channel 8 over range

7.14.13 Native Data Analog Input x (Object 0x3077 to 0x307E)

Objects 0x3077 to 0x307E contain the analog input values as raw values.

Feature	Description / Value		EDS
Name	NativeData AnalogInput Channelx		[MxSubExt307x] ParameterName=NativeDataAIx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
Description	NativeDataA1	Object 0x3077	
	NativeDataA2	Object 0x3078	
	NativeDataA3	Object 0x3079	
	NativeDataA4	Object 0x307A	
	NativeDataA5	Object 0x307B	
	NativeDataA6	Object 0x307C	
	NativeDataA7	Object 0x307D	
	NativeDataA8	Object 0x307E	
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.14.14 Firmware Version (Object 0x4070)

Object 0x4070 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	Firmware Version	[MxSubExt4070] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	FirmwareVersion	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

7.14.15 Sensor Type Selection Channel x (Object 0x5070 bis 0x5073)

Objects 0x5070 to 0x5073 can be used to configure a channel's range.

Feature	Description / Value		EDS
Name	Sensor Type Configuration Channel x		[MxSubExt5070] ParameterName=SensorSelectChannelx ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0
Description	SensorSelectChannel1	Object 0x5070	
	SensorSelectChannel2	Object 0x5071	
	SensorSelectChannel3	Object 0x5072	
	SensorSelectChannel4	Object 0x5073	
Object Code	ARRAY		
Mapping	SDO		
	Manual		
Data type	UNSIGNED8		
Sub index	01 … FE_{hex}		
Access	rw		
Default value	00_{hex}		

Design of the data bytes:

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

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

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

SDO value for sensor type	Bit	Temperature range	Resistance range	Resolution in °C
0	0000 0000	-200 ... +150 °C	18.5 ... 157.3 Ω	1/10
1	0000 0001	-200 ... +850 °C	18.5 ... 390.5 Ω	1/10
2	0000 0010	-200 ... +150 °C	39.0 ... 314 Ω	1/10
3	0000 0011	-200 ... +850 °C	39.0 ... 780 Ω	1/10
4	0000 0100	-200 ... +150 °C	92.6 ... 786.6 Ω	1/10
5	0000 0101	-200 ... +850 °C	92.6 ... 1952.4 Ω	1/10
6	0000 0110	-200 ... +150 °C	185.2 ... 1573.3 Ω	1/10
7	0000 0111	-200 ... +850 °C	185.2 ... 3904.8 Ω	1/10
8	0000 1000	-60 ... +150 °C	69.5 ... 198.7 Ω	1/10
9	0000 1001	-60 ... +250 °C	69.5 ... 290.1 Ω	1/10
10	0000 1010	-60 ... +150 °C	743.0 ... 1987.0 Ω	1/10
11	0000 1011	-60 ... +250 °C	743.0 ... 2800.0 Ω	1/10
12	0000 1100	0	250	1/10
13	0000 1101	0	500	1/10
14	0000 1110	0	1000	1/10
15	0000 1111	0	2500	1/10
16	0001 0000	0	5000	1/10
17	0001 0001	-50 ... +150 °C	1035.9 ... 4575.3 Ω	1/10
18 19	0001 0010 0001 0011	-55 ... +150 °C	450.0 … 2211.0 Ω	1/10
20	0001 0100	-55 ... +150 °C	485.0 … 2189.0 Ω	1/10
21	0001 0101	-55 ... +150 °C	495.0 … 2233.0 Ω	1/10
22	0001 0110	-55 ... +150 °C	450.0 … 2211.0 Ω	1/10
23	0001 0111	-40 ... +300 °C	359.0 … 2624.0 Ω	1/10
24	0001 1000	-40 ... +300 °C	359.0 … 2624.0 Ω	1/10
25	0001 1001	-200 ... +150 °C	18.5 ... 157.3 Ω	1/100

7.14.16 Channel Measuring Configuration (Object 0x5074)

Object 0x5074 must be used to configure the channels' measuring method.

Feature	Description / Value	EDS
Name	Channel1MeasuringConfig	[M13SubExt5074 ParameterName=ChannelMeasuringConfig ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	Channel Measuring Configuration	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE$_{hex}$	
Access	rw	
Default value	00$_{hex}$	

Byte 0 syntax:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designa-tion	Description
0	Channel 1	0 = channel 1 2-wire measuring method 1 = channel 1 3-wire measuring method
1	Channel 2	0 = channel 2 2-wire measuring method 1 = channel 2 3-wire measuring method
2	Channel 3	0 = channel 3 2-wire measuring method 1 = channel 3 3-wire measuring method
3	Channel 4	0 = channel 4 2-wire measuring method 1 = channel 4 3-wire measuring method
4 – 7		reserved

7.14.17 Filter Configuration Channel x (Object 0x5075 to 0x5078)

Objects 0x5075 to 0x5078 can be used to configure the software filter for a channel.

Feature	Description / Value		EDS
Name	Filter Configuration Channel x		[M7SubExt5075] ParameterName=FilterConfigChannelx ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
Description	FilterConfigChannel1	Object 0x5075	
	FilterConfigChannel2	Object 0x5076	
	FilterConfigChannel3	Object 0x5077	
	FilterConfigChannel4	Object 0x5078	
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	rw		
Default value	00_{hex}		

Design of the data bytes:

Low-pass cut-off frequency in Hz (example: 50 Hz => 0 x 0032_{hex})

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

Example

Low-pass cut-off frequency in Hz

50 Hz = 0032_{hex}

The following settings are valid:

Low-pass cut-off frequency	Register value
Filter disabled (default)	0x0000
10 Hz	0x000A
25 Hz	0x0019
50 Hz	0x0032
100 Hz	0x0064

7.14.18 Channel Activation (Object 0x5079)

Object 0x5079 can be used to activate the measuring channel.

Feature	Description / Value	EDS
Name	Channel Activation	[MxSubExt5079] ParameterName=ChannelActivation ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	ChannelActivation	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00_{hex}	

Data byte 0 syntax

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Meaning
0	Channel 1	0 = channel 1 inactive 1 = channel 1 active
1	Channel 2	0 = channel 2inactive 1 = channel 2active
2	Channel 3	0 = channel 3inactive 1 = channel 3 active
3	Channel 4	0 = channel4 inactive 1 = channel 4 active
4 – 7		reserved

7.15 XN-322-7AI-U2PT

This module supports the provision of data for analog inputs as per the specifications in CiA401. The behavior of the I/O slice modules can be configured with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

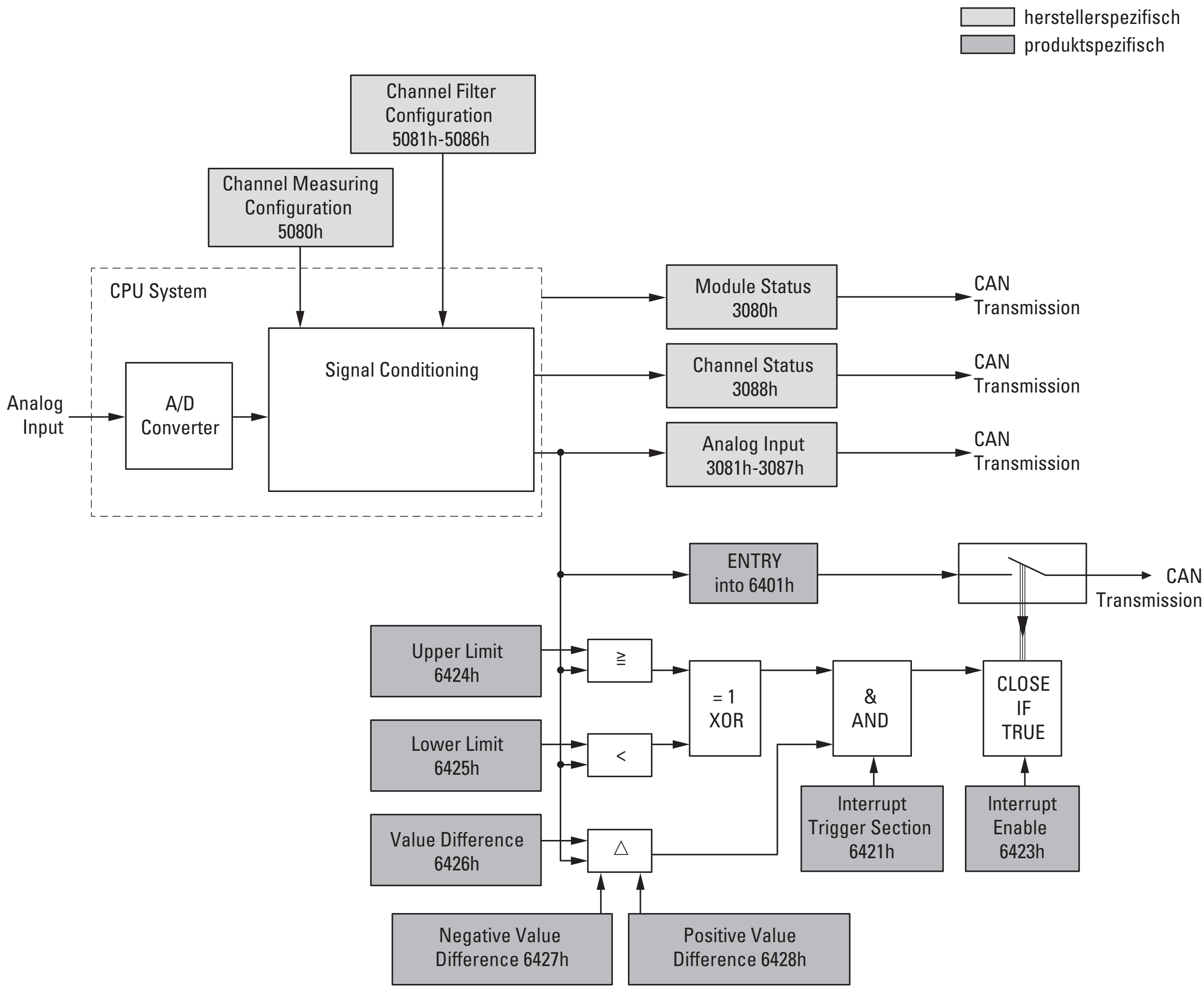


Figure 72: Block diagram showing the various CANopen objects for analog inputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x6401	INTEGER16	I-WORD	Read Analog Input 16-bit	Default	ro	PDO
0x6421	UNSIGNED8	AI_INTERRUPT_TRIGGER_SELECTION	Analog Input Interrupt Trigger Selection	–	rw	SDO
0x6423	BOOLEAN	AnalogInputGlobalInterruptEnable	Analog Input Global Interrupt Enable	–	rw	SDO
0x6424	INTEGER32	AI_UPPER_LIMIT	Analog Input Interrupt Upper Limit Integer	–	rw	SDO
0x6425	INTEGER32	AI_LOWER_LIMIT	Analog Input Interrupt Lower Limit Integer	–	rw	SDO
0x6426	UNSIGNED32	AI_DELTA_VALUE	Analog Input Interrupt Delta Unsigned	–	rw	SDO
0x6427	UNSIGNED32	AI_NEGATIVE_DELTA_VALUE	Analog Input Interrupt Negative Delta Unsigned	–	rw	SDO
0x6428	UNSIGNED32	AI_POSITIVE_DELTA_VALUE	Analog Input Interrupt Positive Delta Unsigned	–	rw	SDO

Manufacturer-specific objects

Index range for the XN-322-7AI-U2PT module: x080 to x08F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x3080	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x3081	INTEGER16	InputChannel1	Input Channel 1	Manual	ro	PDO
0x3082	INTEGER16	InputChannel2	Input Channel 2	Manual	ro	PDO
0x3083	INTEGER16	InputChannel3	Input Channel 3	Manual	ro	PDO
0x3084	INTEGER16	InputChannel4	Input Channel 4	Manual	ro	PDO
0x3085	INTEGER16	InputChannel5	Input Channel 5	Manual	ro	PDO
0x3086	INTEGER16	InputChannel6	Input Channel 6	Manual	ro	PDO
0x3087	INTEGER16	InputChannel7	Input Channel 7	Manual	ro	PDO
0x3088	UNSIGNED16	WireBreakDiag	Wire Break Diagnostic Messages	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO
0x4080	UNSIGNED16	FirmwareVersion	Angabe der FW Version	–	ro	SDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x5080	UNSIGNED16	ChannelMeasuring-Config	Channel Measuring Configuration	–	rw	SDO
0x5081	UNSIGNED16	FilterConfigChannel1	Filter Configuration Channel 1	–	rw	SDO
0x5082	UNSIGNED16	FilterConfigChannel2	Filter Configuration Channel 2	–	rw	SDO
0x5083	UNSIGNED16	FilterConfigChannel3	Filter Configuration Channel 3	–	rw	SDO
0x5084	UNSIGNED16	FilterConfigChannel4	Filter Configuration Channel 4	–	rw	SDO
0x5085	UNSIGNED16	FilterConfigChannel5	Filter Configuration Channel 5	–	rw	SDO
0x5086	UNSIGNED16	FilterConfigChannel6	Filter Configuration Channel 6	–	rw	SDO

7.15.1 Read Analog Input 16-Bit (Object 0x6401)

Object 0x6401 represents the channels' formatted analog input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Read Analog Input 16-Bit	[MxSubExt6401] ParameterName=I-WORD ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=7
Description	I-WORD	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

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

7.15.2 Analog Input Interrupt Trigger Selection (Object 0x6421)

Object 0x6421 defines which events will trigger an interrupt for the corresponding channel so that the channel's analog input value (object 0x6401) will be transmitted when the event occurs.

Feature	Description / Value	EDS
Name	Analog Input Interrupt Trigger Selection	[MxSubExt6421] Parameter-Name=AI_INTERRUPT_TRIGGER_SELECTION ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=7 LowLimit=0 HighLimit=31 PDOMapping=0 Count=7
Description	AI_INTERRUPT_TRIGGER_SELECTION	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	07_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Meaning	Notes
0		0 = Upper limit not exceeded 1 = Upper limit exceeded	
1		0 = Input not below lower limit 1 = Input below lower limit	
2		0 = Input not changed by more than delta 1 = Input changed by more than delta	
3		0 = Input not reduced by more than negative delta 1 = Input reduced by more than negative d	
4		0 = Input not reduced by more than positive delta 1 = Input reduced by more than positive delta	
5 – 7		reserved	

7.15.3 Analog Input Global Interrupt Enable (Object 0x6423)

Object 0x6423 enables and disables the transmission of data via object 0x6401 in general.

Feature	Description / Value	EDS
Name	AnalogInputGlobalInterruptEnable	[MxFixed6423] ParameterName=AnalogInputGlobalInterruptEnable ObjectType=0x7 DataType=0x0001 AccessType=rw DefaultValue=0 PDOMapping=0
Description	Analog Input Global Interrupt Enable	
Object Code	Variable	
Mapping	SDO	
Data type	BOOLEAN	
Access	rw	
Default value	FALSE	
Object Code	Variable	

Default value

- FALSE (0)
  Object 0x6401 is not enabled and cannot transmit any analog input values.
- TRUE (1)
  Object 0x6401 is enabled and can transmit analog input values.

7.15.4 Analog Input Interrupt Upper Limit Integer (Object 0x6424)

Object 0x6424 defines an upper limit and affects the transmission of object 0x6401 accordingly → Figure 72, page 183.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) is equal to or greater than the upper limit (Upper Limit 0x6424).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_UPPER_LIMIT	[MxSubExt6424] ParameterName=AI_UPPER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=7
Description	Analog Input Interrupt Upper Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.15.5 Analog Input Interrupt Lower Limit Integer (Object 0x6425)

Object 0x6425 defines a lower limit and affects the transmission of object 0x6401 accordingly → Figure 72, page 183.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) falls below the lower limit (Lower Limit 0x6425).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_LOWER_LIMIT	[MxSubExt6425] ParameterName=AI_LOWER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=7
Description	Analog Input Interrupt Lower Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

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

Sub index 1≤n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.15.6 Analog Input Interrupt Delta Unsigned (Object 0x6426)

Object 0x6426 defines the fluctuation range (delta) relative to the last transmitted value both when counting up and down and affects the transmission of object 0x6401 accordingly → Figure 72, page 183.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The difference between the current value (ENTRY into 0x6401) and the last transmitted value is equal to or greater than the fluctuation range (Value Difference 0x6426).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_DELTA_VALUE	[MxSubExt6426] ParameterName=AI_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=7
Description	Analog Input Interrupt Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01... FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≤n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.15.7 Analog Input Interrupt Negative Delta Unsigned (Object 0x6427)

Object 0x6427 defines the negative fluctuation range (delta) relative to the last transmitted value when counting down and affects the transmission of object 0x6401 accordingly → Figure 72, page 183.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is less than the last transmitted value and the difference is greater than or equal to the fluctuation range (Negative Value Difference 0x6427).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_NEGATIVE_DELTA_VALUE	[MxSubExt6427] Parameter-Name=AI_NEGATIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=7
Description	Analog Input Interrupt Negative Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE $_{ex}$	
Access	rw	
Default value	00000000$_{hex}$	

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.15.8 Analog Input Interrupt Positive Delta Unsigned (Object 0x6428)

Object 0x6428 defines the negative fluctuation range (delta) relative to the last transmitted value when counting up and affects the transmission of object 0x6401 accordingly → Figure 72, page 183.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is greater than the last transmitted value. The difference's magnitude is greater than the fluctuation range (Positive Value Difference 0x6428).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

If the current value is less than the last transmitted value, object 0x6401 will be transmitted. If this behavior is not desired, object 0x6421 must be enabled and disabled accordingly.

Feature	Description / Value	EDS
Name	AI_POSITIVE_DELTA_VALUE	[MxSubExt6428] Parameter-Name=AI_POSITIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=7
Description	Analog Input Interrupt Positive Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															LSB

7.15.9 Module Diagnostic Messages (Object 0x3080)

Object 0x3080 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	ModuleDiag	[MxSubExt3080] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Module Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	0000_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–						

Data bit	Designation	Meaning	Notes
0		reserved	
1		0 = sync OK 1 = no sync	
2		0 = OK 1 = FLASH Data CRC Error	-
3		0 = OK 1 =RAM Data CRC Error	
4		0 = OK 1 = EEProm version not valid	
5-7		reserved	

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
0 – 7		reserved

7.15.10 Input Channel x (Object 0x3081 to 0x3087)

Objects 3081 to 3087 represent the analog input values.

Feature	Description / Value		EDS
Name	InputChannel1	Object 0x3081	[M8SubExt308x] ParameterName=InputChannelx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
	InputChannel2	Object 0x3082	
	InputChannel3	Object 0x3083	
	InputChannel4	Object 0x3084	
	InputChannel5	Object 0x3085	
	InputChannel6	Object 0x3086	
	InputChannel7	Object 0x3087	
Description	Input Channel x		
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.15.11 Wire Break Diagnostic Messages (Object 0x3088)

Object 0x3088 contains status information on the device's channels.

Feature	Description / Value	EDS
Name	WireBreakDiag	[MxSubExt3088] ParameterName=WireBreakDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Wire Break Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB
	Channel 2								Channel 1							

Byte 0:

Data bit	Desig-nation	Description	Notes
0		0 = OK 1 = channel 1 wire breakage	
1		0 = OK 1 = channel 2 wire breakage	
2		0 = OK 1 = channel 3 wire breakage	
3		0 = OK 1 = channel 4 wire breakage	
4		0 = OK 1 = channel 5 wire breakage	
5		0 = OK 1 = channel 6 wire breakage	
6		0 = OK 1 = channel 7 wire breakage	
7		0 = OK 1 = Short circuit channel 1 when KTY/ PT parameterization	

Byte 1:

Data bit	Desig-nation	Description	Notes
0		0 = OK 1 = Short circuit channel 7 when KTY/ PT parameterization	
1		0 = OK 1 =Low Voltage U Reference	
2		0 = OK 1 = Overcurrent U Reference	-
3 – 7		reserved	

7.15.12 Firmware Version (Object 0x4080)

Object 0x4080 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	FirmwareVersion	[MxSubExt4080] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	Firmware Version	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

7.15.13 Channel Measuring Configuration (Object 0x5080)

Object 0x5080 can be used to configure the measuring method for a channel.

Feature	Description / Value	EDS
Name	ChannelMeasuringConfig	[MxSubExt5080 ParameterName=ChannelMeasuringConfig ObjectType=0x7 DataType=0x0006 AccessType=rw DefaultValue=1 PDOMapping=0 Count=1
Description	Channel Measuring Configuration	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000_{hex}	

Data word structure:

Sub index 1≤n≤254	Word															
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Word 0:

Data bit	Designation	Description	Notes
0	Analog Input 1	0 = Analog measurement ± 10V 1 = Temperature measurement	
1		0 = KTY10 Sensor 1 = PT1000 Sensor	
2		0 = Differential measurement 1 = AI1- grounded measurement	
3	Analog Input 2	0 = Differential measurement 1 = AI1- grounded measurement	
4	Analog Input 3	0 = Differential measurement 1 = AI1- grounded measurement	
5	Analog Input 4	0 = Differential measurement 1 = AI1- grounded measurement	
6	Analog Input 5	0 = Differential measurement 1 = AI1- grounded measurement	
7	Analog Input 6	0 = Differential measurement 1 = AI1- grounded measurement	
8	Sensor selection	0: KTY10 Sensor 1: Pt1000 Sensor	
9-15		reserved	

7.15.14 Filter Configuration Channel x (Object 0x5081 to 0x5086)

Objects 0x5081 to 0x5086 can be used to configure the software filter for a channel.

Feature	Description / Value		EDS
Name	FilterConfigChannel1	Object 0x5081	[MxSubExt508x] ParameterName=FilterConfigChannelx ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
	FilterConfigChannel2	Object 0x5082	
	FilterConfigChannel3	Object 0x5083	
	FilterConfigChannel4	Object 0x5084	
	FilterConfigChannel5	Object 0x5085	
	FilterConfigChannel6	Object 0x5086	
Description	Filter Configuration Channel x		
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	rw		
Default value	0000_{hex}		

Design of the data bytes:

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

Example

Low-pass cut-off frequency in Hz

50 Hz = 0032_{hex}

The following settings are valid:

Low-pass cut-off frequency	Register value
Filter disabled (default)	0x0000
10 Hz	0x000A
25 Hz	0x0019
50 Hz	0x0032
100 Hz	0x0064
250 Hz	0x00FA
500 Hz	0x01F4
1000 Hz	0x03E8

7.16 XN-322-8AI-I

This module supports the provision of data for analog inputs and analog outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

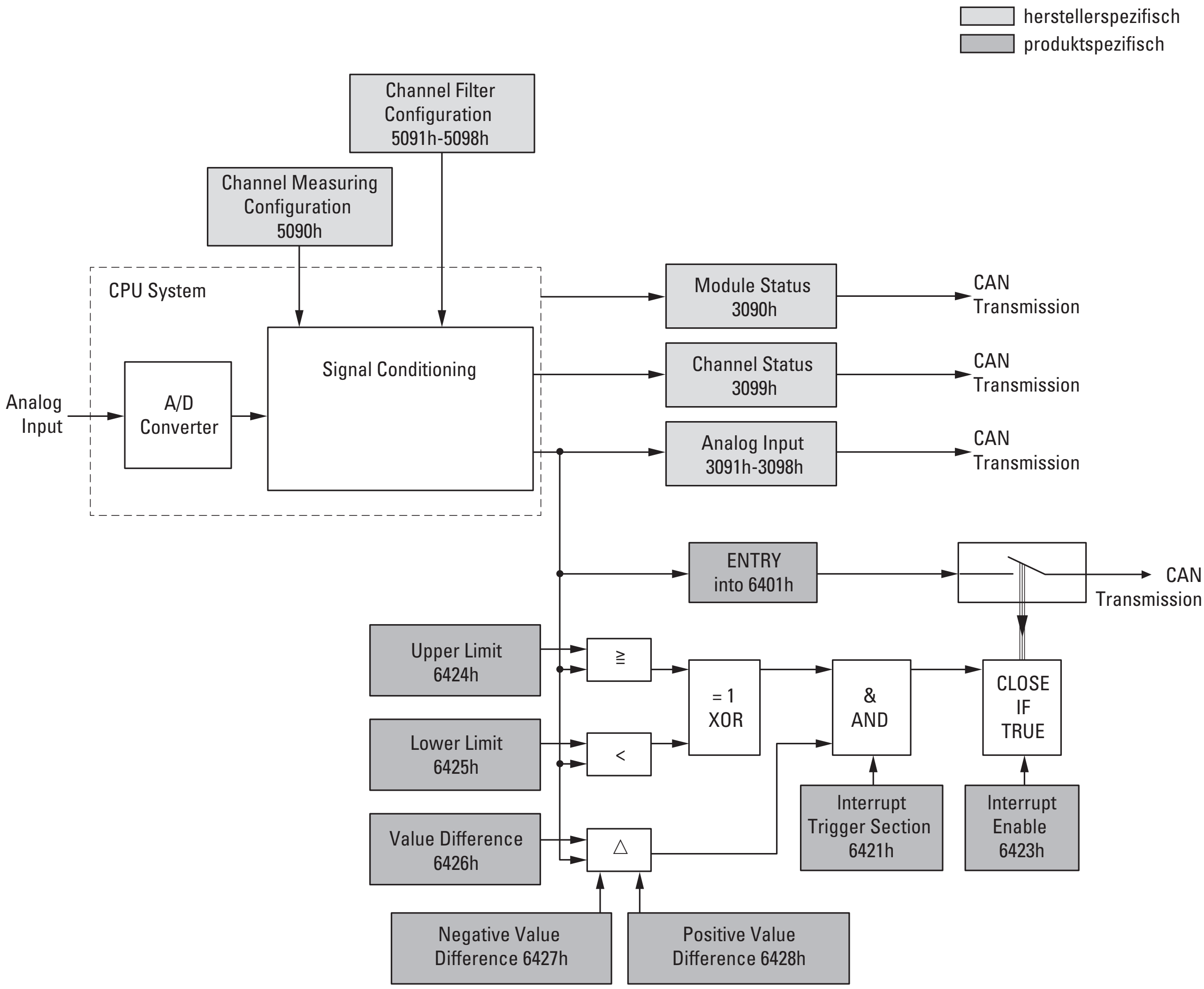


Figure 73: Block diagram showing the various CANopen objects for analog inputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x6401	INTEGER16	I-WORD	Read Analog Input 16-bit	Default	ro	PDO
0x6421	UNSIGNED8	AI_INTERRUPT_TRIGGER_SELECTION	Analog Input Interrupt Trigger Selection	–	rw	SDO
0x6423	BOOLEAN	AnalogInputGlobalInterruptEnable	Analog Input Global Interrupt Enable	–	rw	SDO
0x6424	INTEGER32	AI_UPPER_LIMIT	Analog Input Interrupt Upper Limit Integer	–	rw	SDO
0x6425	INTEGER32	AI_LOWER_LIMIT	Analog Input Interrupt Lower Limit Integer	–	rw	SDO
0x6426	UNSIGNED32	AI_DELTA_VALUE	Analog Input Interrupt Delta Unsigned	–	rw	SDO
0x6427	UNSIGNED32	AI_NEGATIVE_DELTA_VALUE	Analog Input Interrupt Negative Delta Unsigned	–	rw	SDO
0x6428	UNSIGNED32	AI_POSITIVE_DELTA_VALUE	Analog Input Interrupt Positive Delta Unsigned	–	rw	SDO

Manufacturer-specific objects

Index range for the XN-322-8AI-I module: x090 to x09F

Index (hex)	Data Type	Name	Function	Mappin g	Access	
0x1027	UNSIGNED16	ModuleD	Module Identification Number → Section “6.2.6 Module Identification Number (Object 0x1027)”, page 76	–	ro	SDO
0x3090	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x3091	INTEGER16	InputChannel1	Input Channel 1	Manual	ro	PDO
0x3092	INTEGER16	InputChannel2	Input Channel 2	Manual	ro	PDO
0x3093	INTEGER16	InputChannel3	Input Channel 3	Manual	ro	PDO
0x3094	INTEGER16	InputChannel4	Input Channel 4	Manual	ro	PDO
0x3095	INTEGER16	InputChannel5	Input Channel 5	Manual	ro	PDO
0x3096	INTEGER16	InputChannel6	Input Channel 6	Manual	ro	PDO
0x3097	INTEGER16	InputChannel7	Input Channel 7	Manual	ro	PDO
0x3098	INTEGER16	InputChannel8	Input Channel 8	Manual	ro	PDO
0x3099	UNSIGNED16	WireBreakDiag	Wire Break Diagnostic Messages	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO
0x4090	UNSIGNED16	FirmwareVersion	Firmware Version	-	ro	SDO
0x5090	UNSIGNED16	ChannelMeasuringConfig	Channel Measuring Configuration	–	rw	SDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x5091	UNSIGNED16	FilterConfigChannel1	Filter Configuration Channel 1	–	rw	SDO
0x5092	UNSIGNED16	FilterConfigChannel2	Filter Configuration Channel 2	–	rw	SDO
0x5093	UNSIGNED16	FilterConfigChannel3	Filter Configuration Channel 3	–	rw	SDO
0x5094	UNSIGNED16	FilterConfigChannel4	Filter Configuration Channel 4	–	rw	SDO
0x5095	UNSIGNED16	FilterConfigChannel5	Filter Configuration Channel 5	–	rw	SDO
0x5096	UNSIGNED16	FilterConfigChannel6	Filter Configuration Channel 6	–	rw	SDO
0x5097	UNSIGNED16	FilterConfigChannel7	Filter Configuration Channel 7	–	rw	SDO
0x5098	UNSIGNED16	FilterConfigChannel8	Filter Configuration Channel 8	–	rw	SDO

7.16.1 Read Analog Input 16-Bit (Object 0x6401)

Object 0x6401 represents the channels' formatted analog input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Read Analog Input 16-Bit	[MxSubExt6401] ParameterName=I-WORD ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=8
Description	I-WORD	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE hex	
Access	ro	
Default value	0 x 0000_{hex}	

Design of the data bytes:

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

7.16.2 Analog Input Interrupt Trigger Selection (Object 0x6421)

Object 0x6421 defines which events will trigger an interrupt for the corresponding channel so that the channel's analog input value (object 0x6401) will be transmitted when the event occurs.

Feature	Description / Value	EDS
Name	Analog Input Interrupt Trigger Selection	[MxSubExt6421] Parameter- Name=AI_INTERRUPT_TRIGGER_SELECTION ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=7 LowLimit=0 HighLimit=31 PDOMapping=0 Count=8
Description	AI_INTERRUPT_TRIGGER_SELECTION	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	$0x07_{hex}$	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
0		0 = Upper limit not exceeded 1 = Upper limit exceeded
1		0 = Input not below lower limit 1 = Input below lower limit
2		0 = Input not changed by more than delta 1 = Input changed by more than delta
3		0 = Input not reduced by more than negative delta 1 = Input reduced by more than negative d
4		0 = Input not reduced by more than positive delta 1 = Input reduced by more than positive delta
5 – 7		reserved

7.16.3 Analog Input Global Interrupt Enable (Object 0x6423)

Object 0x6423 enables and disables the transmission of data via object 0x6401 in general.

Feature	Description / Value	EDS
Name	Analog Input Global Interrupt Enable	[MxFixed6423] ParameterName=AnalogInputGlobalInterruptEnable ObjectType=0x7 DataType=0x0001 AccessType=rw DefaultValue=0 PDOMapping=0
Description	AnalogInputGlobalInterruptEnable	
Object Code	Variable	
Mapping	SDO	
Data type	BOOLEAN	
Access	rw	
Default value	FALSE	
Object Code	Variable	

Default value

- FALSE (0)
  Object 0x6401 is not enabled and cannot transmit any analog input values.
- TRUE (1)
  Object 0x6401 is enabled and can transmit analog input values.

7.16.4 Analog Input Interrupt Upper Limit Integer (Object 0x6424)

Object 0x6424 defines an upper limit and affects the transmission of object 0x6401 accordingly → Figure 73, page 198.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) is equal to or greater than the upper limit (Upper Limit 0x6424).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_UPPER_LIMIT	[MxSubExt6424] ParameterName=AI_UPPER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=8
Description	Analog Input Interrupt Upper Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.16.5 Analog Input Interrupt Lower Limit Integer (Object 0x6425)

Object 0x6425 defines a lower limit and affects the transmission of object 0x6401 accordingly → Figure 73, page 198.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) falls below the lower limit (Lower Limit 0x6425).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_LOWER_LIMIT	[MxSubExt6425] ParameterName=AI_LOWER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=8
Description	Analog Input Interrupt Lower Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

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

Sub index 1≤n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.16.6 Analog Input Interrupt Delta Unsigned (Object 0x6426)

Object 0x6426 defines the fluctuation range (delta) relative to the last transmitted value both when counting up and down and affects the transmission of object 0x6401 accordingly → Figure 73, page 198.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The difference between the current value (ENTRY into 0x6401) and the last transmitted value is equal to or greater than the fluctuation range (Value Difference 0x6426).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_DELTA_VALUE	[MxSubExt6426] ParameterName=AI_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=8
Description	Analog Input Interrupt Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01... FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≤n≤254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.16.7 Analog Input Interrupt Negative Delta Unsigned (Object 0x6427)

Object 0x6427 defines the negative fluctuation range (delta) relative to the last transmitted value when counting down and affects the transmission of object 0x6401 accordingly → Figure 73, page 198.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is less than the last transmitted value and the difference is greater than or equal to the fluctuation range (Negative Value Difference 0x6427).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_NEGATIVE_DELTA_VALUE	[MxSubExt6427] Parameter-Name=AI_NEGATIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=8
Description	Analog Input Interrupt Negative Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE $_{ex}$	
Access	rw	
Default value	00000000$_{hex}$	

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.16.8 Analog Input Interrupt Positive Delta Unsigned (Object 0x6428)

Object 0x6428 defines the negative fluctuation range (delta) relative to the last transmitted value when counting up and affects the transmission of object 0x6401 accordingly → Figure 73, page 198.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is greater than the last transmitted value. The difference's magnitude is greater than the fluctuation range (Positive Value Difference 0x6428).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

If the current value is less than the last transmitted value, object 0x6401 will be transmitted. If this behavior is not desired, object 0x6421 must be enabled and disabled accordingly.

Feature	Description / Value	EDS
Name	AI_POSITIVE_DELTA_VALUE	[MxSubExt6428] ParameterName=AI_POSITIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=8
Description	Analog Input Interrupt Positive Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															LSB

7.16.9 Module Diagnostic Messages (Object 0x3090)

Object 0x3090 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	Module Diagnostic Messages	[MxSubExt3090] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	ModuleDiag	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–					

Data bit	Designation	Meaning
0		reserved
1		0 = sync OK 1 = no sync
2		0 = OK 1 = FLASH Data CRC Error
3		0 = OK 1 =RAM Data CRC Error
4		0 = OK 1 = Inconsistent FLASH Data
5 – 7		reserved

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
8 – 15		reserved

7.16.10 Input Channel x (Object 0x3091 to 0x3098)

Objects 0x3091 to 0x3098 represent the formatted (integer) analog input values.

Feature	Description / Value		EDS
Name	Input Channel x		[MxSubExt309x] ParameterName=InputChannelx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
Description	InputChannel1	Object 0x3091	
	InputChannel2	Object 0x3092	
	InputChannel3	Object 0x3093	
	InputChannel4	Object 0x3094	
	InputChannel5	Object 0x3095	
	InputChannel6	Object 0x3096	
	InputChannel7	Object 0x3097	
	InputChannel8	Object 0x3098	
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.16.11 Wire Break Diagnostic Messages (Object 0x3099)

Object 0x3099 contains status information on the module's channels.

Feature	Description / Value	EDS
Name	WireBreakDiag	[Mx]SubExt3099] ParameterName=WireBreakDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Wire Break Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Description
0	0 = Channel 1 OK 1 = Channel 1 wire breakage
1	0 = Channel 2 OK 1 = Channel 2wire breakage
2	0 = Channel 3 OK 1 = Channel 3 wire breakage
3	0 = Channel 4 OK 1 = Channel 4 wire breakage
4	0 = Channel 5OK 1 = Channel 5wire breakage
5	0 = Channel 6OK 1 = Channel 6wire breakage
6	0 = Channel 7OK 1 = Channel 7 wire breakage
7	0 = Channel 8 OK 1 = Channel 8 wire breakage

7.16.12 Firmware Version (Object 0x4090)

Object 0x4090 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	Firmware Version	[MxSubExt4090] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	FirmwareVersion	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

7.16.13 Channel Measuring Configuration (Object 0x5090)

Object 0x5090 must be used to configure the channels' measuring method.

Feature	Description / Value	EDS
Name	Channel1MeasuringConfig	[MxSubExt5090 ParameterName=ChannelMeasuringConfig ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
Description	Channel Measuring Configuration	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000_{hex}	

Design of the data bytes:

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

Byte 0:

Data bit	Designation	Description
0	Channel 1	0 = measurement range 0 – 20mA 1 = measurement range 4 – 20mA
1	Channel 2	0 = measurement range 0 – 20mA 1 = measurement range 4 – 20mA
2	Channel 3	0 = measurement range 0 – 20mA 1 = measurement range 4 – 20mA
3	Channel 4	0 = measurement range 0 – 20mA 1 = measurement range 4 – 20mA
4	Channel 5	0 = measurement range 0 – 20mA 1 = measurement range 4 – 20mA
5	Channel 6	0 = measurement range 0 – 20mA 1 = measurement range 4 – 20mA
6	Channel 7	0 = measurement range 0 – 20mA 1 = measurement range 4 – 20mA
7	Channel 8	0 = measurement range 0 – 20mA 1 = measurement range 4 – 20mA

Byte 1:

Data bit	Designation	Description
7 – 15		reserved

7.16.14 Channel x Filter Configuration (Object 0x5091 to 0x5098)

Objects 0x5091 to 0x5098 can be used to configure the software filter for a channel.

Feature	Description / Value		EDS
Name	Channel x Filter Configuration		[MxSubExt509x] ParameterName=ChannelxFilterConfig ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
Description	FilterConfigChannel1	Object 0x5091	
	FilterConfigChannel2	Object 0x5092	
	FilterConfigChannel3	Object 0x5093	
	FilterConfigChannel4	Object 0x5094	
	FilterConfigChannel5	Object 0x5095	
	FilterConfigChannel6	Object 0x5096	
	FilterConfigChannel7	Object 0x5097	
	FilterConfigChannel8	Object 0x5098	
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	rw		
Default value	0000_{hex}		

Design of the data bytes:

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

Low-pass cut-off frequency in Hz (example: 50 Hz => 0 x 00 32 hex)

The following settings are valid:

Low-pass cut-off frequency	Register value
Filter disabled	0x0000
10 Hz	0x000A
25 Hz	0x0019
50 Hz	0 x0032
100 Hz	0x0064
250 Hz	0x00FA
500 Hz	0x01F4
1000 Hz (Default)	0x03E8

7.17 XN-322-10AI-TEKT

This module supports the provision of data for analog inputs as per the specifications in CiA401. The behavior of the I/O slice modules can be configured with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

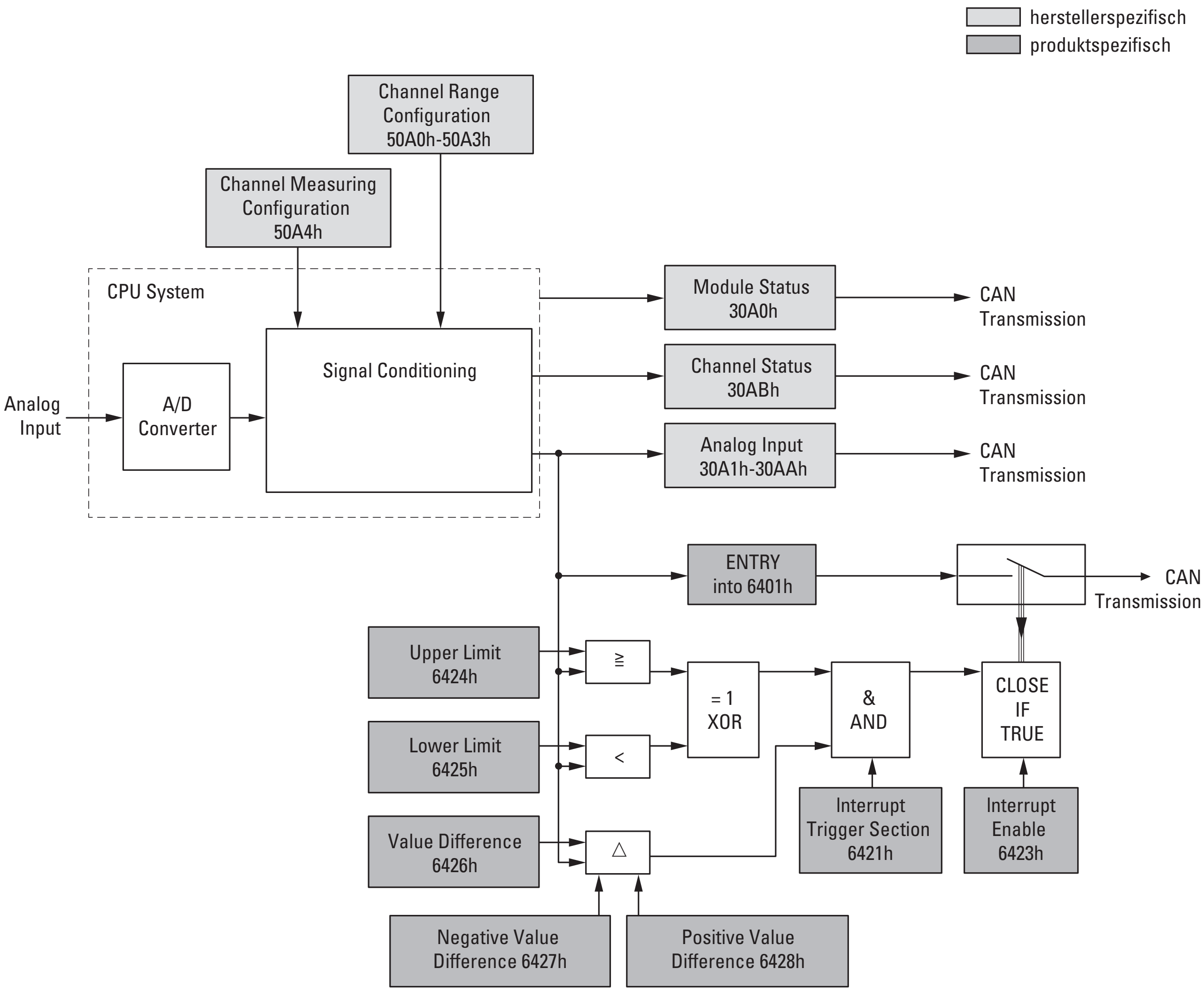


Figure 74: Block diagram showing the various CANopen objects

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x6401	INTEGER16	I-WORD	Read Analog Input 16-bit	Default	ro	PDO
0x6421	UNSIGNED8	AI_INTERRUPT_TRIGGER_SELECTION	Analog Input Interrupt Trigger Selection	–	rw	SDO
0x6423	BOOLEAN	AnalogInputGlobalInterruptEnable	Analog Input Global Interrupt Enable	–	rw	SDO
0x6424	INTEGER32	AI_UPPER_LIMIT	Analog Input Interrupt Upper Limit Integer	–	rw	SDO
0x6425	INTEGER32	AI_LOWER_LIMIT	Analog Input Interrupt Lower Limit Integer	–	rw	SDO
0x6426	UNSIGNED32	AI_DELTA_VALUE	Analog Input Interrupt Delta Unsigned	–	rw	SDO
0x6427	UNSIGNED32	AI_NEGATIVE_DELTA_VALUE	Analog Input Interrupt Negative Delta Unsigned	–	rw	SDO
0x6428	UNSIGNED32	AI_POSITIVE_DELTA_VALUE	Analog Input Interrupt Positive Delta Unsigned	–	rw	SDO

Manufacturer-specific objects

Index range for the XN-322-10AI-TEKT module: x0A0 to x0AF

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section “6.2.6 Module Identification Number (Object 0x1027)”, page 76	–	ro	SDO
0x30A0	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x30A1	INTEGER16	InputChannel1	Input Channel 1	Manual	ro	PDO
0x30A2	INTEGER16	InputChannel2	Input Channel 2	Manual	ro	PDO
0x30A3	INTEGER16	InputChannel3	Input Channel 3	Manual	ro	PDO
0x30A4	INTEGER16	InputChannel4	Input Channel 4	Manual	ro	PDO
0x30A5	INTEGER16	InputChannel5	Input Channel 5	Manual	ro	PDO
0x30A6	INTEGER16	InputChannel6	Input Channel 6	Manual	ro	PDO
0x30A7	INTEGER16	InputChannel7	Input Channel 7	Manual	ro	PDO
0x30A8	INTEGER16	InputChannel8	Input Channel 8	Manual	ro	PDO
0x30A9	INTEGER16	ReferenceInput1	Input Reference 1	Manual	ro	PDO
0x30AA	INTEGER16	ReferenceInput2	Input Reference 2	Manual	ro	PDO
0x30AB	UNSIGNED16	WireBreakDetect	Wire Break Detection	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO
0x40A0	UNSIGNED16	FirmwareVersion	Angabe der FW Version	–	ro	SDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x50A0	UNSIGNED8	SensorTypeSelectChannel1_2	Sensor Type Selection Channel 1_2	–	rw	SDO
0x50A1	UNSIGNED8	SensorTypeSelectChannel3_4	Sensor Type Selection Channel 3_4	–	rw	SDO
0x50A2	UNSIGNED8	SensorTypeSelectChannel5_6	Sensor Type Selection Channel 5_6	–	rw	SDO
0x50A3	UNSIGNED8	SensorTypeSelectChannel7_8	Sensor Type Selection Channel 7_8	–	rw	SDO
0x50A4	UNSIGNED8	ReferenceInputSelect	Reference Input Select Cold-junction compensation configuration (KTY1,KTY2)	–	rw	SDO

7.17.1 Read Analog Input 16-Bit (Object 0x6401)

Object 0x6401 represents the channels' formatted analog input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-WORD	[MxSubExt6401] ParameterName=I-WORD ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=10
Description	Read Analog Input 16-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

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

7.17.2 Analog Input Interrupt Trigger Selection (Object 0x6421)

Object 0x6421 defines which events will trigger an interrupt for the corresponding channel so that the channel's analog input value (object 0x6401) will be transmitted when the event occurs.

Feature	Description / Value	EDS
Name	AI_INTERRUPT_TRIGGER_SELECTION	[MxSubExt6421] Parameter-Name=AI_INTERRUPT_TRIGGER_SELECTION ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=7 LowLimit=0 HighLimit=31 PDOMapping=0 Count=10
Description	Analog Input Interrupt Trigger Selection	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	$0x07_{hex}$	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
0		0 = Upper limit not exceeded 1 = Upper limit exceeded
1		0 = Input not below lower limit 1 = Input below lower limit
2		0 = Input not changed by more than delta 1 = Input changed by more than delta
3		0 = Input not reduced by more than negative delta 1 = Input reduced by more than negative d
4		0 = Input not reduced by more than positive delta 1 = Input reduced by more than positive delta
5 – 7		reserved

7.17.3 Analog Input Global Interrupt Enable (Object 0x6423)

Object 0x6423 enables and disables the transmission of data via object 0x6401 in general.

Feature	Description / Value	EDS
Name	AnalogInputGlobalInterruptEnable	[MxFixed6423] ParameterName=AnalogInputGlobalInterruptEnable ObjectType=0x7 DataType=0x0001 AccessType=rw DefaultValue=0 PDOMapping=0
Description	Analog Input Global Interrupt Enable	
Object Code	Variable	
Mapping	SDO	
Data type	BOOLEAN	
Access	rw	
Default value	FALSE	
Object Code	Variable	

Default value

- FALSE (0)
  Object 0x6401 is not enabled and cannot transmit any analog input values.
- TRUE (1)
  Object 0x6401 is enabled and can transmit analog input values.

7.17.4 Analog Input Interrupt Upper Limit Integer (Object 0x6424)

Object 0x6424 defines an upper limit and affects the transmission of object 0x6401 accordingly → Figure 74, page 213.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) is equal to or greater than the upper limit (Upper Limit 0x6424).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_UPPER_LIMIT	[MxSubExt6424] ParameterName=AI_UPPER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=10
Description	Analog Input Interrupt Upper Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.17.5 Analog Input Interrupt Lower Limit Integer (Object 0x6425)

Object 0x6425 defines a lower limit and affects the transmission of object 0x6401 accordingly → Figure 74, page 213.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) falls below the lower limit (Lower Limit 0x6425).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_LOWER_LIMIT	[MxSubExt6425] ParameterName=AI_LOWER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=10
Description	Analog Input Interrupt Lower Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

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

Sub index 1≤n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.17.6 Analog Input Interrupt Delta Unsigned (Object 0x6426)

Object 0x6426 defines the fluctuation range (delta) relative to the last transmitted value both when counting up and down and affects the transmission of object 0x6401 accordingly → Figure 74, page 213.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The difference between the current value (ENTRY into 0x6401) and the last transmitted value is equal to or greater than the fluctuation range (Value Difference 0x6426).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_DELTA_VALUE	[MxSubExt6426] ParameterName=AI_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=10
Description	Analog Input Interrupt Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	$01... FE_{hex}$	
Access	rw	
Default value	$0000\ 0000_{hex}$	

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.17.7 Analog Input Interrupt Negative Delta Unsigned (Object 0x6427)

Object 0x6427 defines the negative fluctuation range (delta) relative to the last transmitted value when counting down and affects the transmission of object 0x6401 accordingly → Figure 74, page 213.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is less than the last transmitted value and the difference is greater than or equal to the fluctuation range (Negative Value Difference 0x6427).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_NEGATIVE_DELTA_VALUE	[MxSubExt6427] ParameterName=AI_NEGATIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=10
Description	Analog Input Interrupt Negative Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE $_{ex}$	
Access	rw	
Default value	00000000$_{hex}$	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.17.8 Analog Input Interrupt Positive Delta Unsigned (Object 0x6428)

Object 0x6428 defines the negative fluctuation range (delta) relative to the last transmitted value when counting up and affects the transmission of object 0x6401 accordingly → Figure 74, page 213.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is greater than the last transmitted value. The difference's magnitude is greater than the fluctuation range (Positive Value Difference 0x6428).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

If the current value is less than the last transmitted value, object 0x6401 will be transmitted. If this behavior is not desired, object 0x6421 must be enabled and disabled accordingly.

Feature	Description / Value	EDS
Name	AI_POSITIVE_DELTA_VALUE	[MxSubExt6428] ParameterName=AI_POSITIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=10
Description	Analog Input Interrupt Positive Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															LSB

7.17.9 Module Diagnostic Messages (Object 0x30A0

Object 0x30A0 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	Module Diagnostic Messages	[MxSubExt30A0] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	ModuleDiag	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–					

Data bit	Designation	Description
0		0 = 24VDC OK 1 = 24VDC not OK
1		0 = sync OK 1 = no sync
2		0 = OK 1 = FLASH Data CRC Error
3		0 = OK 1 =RAM Data CRC Error
4		0 = OK 1 = Inconsistent FLASH Data
5-7		reserved

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Description
0 – 7		reserved

7.17.10 Input Channel x (Object 0x30A1 to 0x30A8)

Objects 0x30A1 to 0x30A8 represent the analog input values.

Feature	Description / Value		EDS
Name	Input Channel x		[MxSubExt306x] ParameterName=InputChannelx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
Description	InputChannel1	Object 0x30A1	
	InputChannel2	Object 0x30A2	
	InputChannel3	Object 0x30A3	
	InputChannel4	Object 0x30A4	
	InputChannel5	Object 0x30A5	
	InputChannel6	Object 0x30A6	
	InputChannel7	Object 0x30A7	
	InputChannel8	Object 0x30A8	
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.17.11 Reference Input Channel x (Object 0x30A9 to 0x30AA)

Objects 0x30A9 to 0x30AA represent the analog input values formatted as integers.

Feature	Description / Value		EDS
Name	Input Channel x		[MxSubExt30Ax] ParameterName=ReferenceInputx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
Description	ReferenceInput1	Object 0x30A9	
	ReferenceInput2	Object 0x30AA	
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.17.12 Wire Break Diagnostic Messages (Object 0x30AB)

Object 0x30AB contains status information on the module's channels

Feature	Description / Value	EDS
Name	Wire Break Detection	[MxSubExt30AB] ParameterName=WireBreakDetect ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	WireBreakDetect	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–					

Data bit	Desig-nation	Meaning
0		0 = OK 1 = Channel 1 cable breakage
1		0 = OK 1 = Channel 2 cable breakage
2		0 = OK 1 = Channel 3 cable breakage
3		0 = OK 1 = Channel 4 cable breakage
4		0 = OK 1 = Channel 5 cable breakage
5		0 = OK 1 = Channel 6 cable breakage
6		0 = OK 1 = Channel 7 cable breakage
7		0 = OK 1 = Channel 8 cable breakage

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
0		0 = OK 1 = Channel 9 cable breakage
1		0 = OK 1 = Channel 10 cable breakage
2		0 = OK 1 = channel 9 cable short circuit
3		0 = OK 1 = channel 10cable short circuit
4-7		reserved

7.17.13 Firmware Version (Object 0x40A0)

Object 0x40A0 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	Firmware Version	[MxSubExt40A0] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	FirmwareVersion	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

Byte 0 = Minor-Version, Byte 1 = Major-Version

7.17.14 Channel x Sensor Type Selection (Object 0x50A0, 0x50A3)

Objects 0x50A0 and 0x50A3 can be used to configure the range or input gain for a channel. The low nibble will configure analog input n, while the high nibble will configure analog input n+1.

Feature	Description / Value		EDS
Name	RangeConfig Channelx		[MxSubExt50Ax] ParameterName=SensorSelectChannelx ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=0 PDOMapping=0 Count=1
Description	SensorSelectChannel1_2	Object 0x50A0	
	SensorSelectChannel3_4	Object 0x50A1	
	SensorSelectChannel5_6	Object 0x50A2	
	SensorSelectChannel7_8	Object 0x50A3	
Object Code	ARRAY		
Mapping	SDO		
Data Type	UNSIGNED8		
Sub index	01 … FE_{hex}		
Access	rw		
Default value	00_{hex}		

Design of the data bytes:

Byte 0:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB
	AI2				AI1			
	AI4				AI3			
	AI6				AI5			
	AI8				AI7			

Object	AI	Bit	Description
50A0	AI1	Bit 0-3	Table 38, page 227
	AI2	Bit 4-7	
50A1	AI3	Bit 0-3	
	AI4	Bit 4-7	
50A2	AI5	Bit 0-3	
	AI6	Bit 4-7	
50A3	AI7	Bit 0-3	
	AI8	Bit 4-7	

Table 38: Defining measuring ranges for analog inputs with the high and low data nibbles

Hexadecimal value Bit 0-3 Bit 4-7	Part no.	Measuring range
0_{hex}	J	0 … +690 °C
1_{hex}	K	0…+940 °C
2_{hex}	T	0…+400 °C
3_{hex}	E	0…+520 °C
4_{hex}	N	0… +1080 °C
5_{hex}	S	0…+1760 °C
6_{hex}	R	0… +1760 °C
7_{hex}	B	0… +1820 °C
8_{hex}	L	0…+680 °C
9_{hex}	U	0…+590 °C
$(A-F)_{hex}$	reserved	

7.17.15 Reference Input Select (Object 0x50A4)

Object 0x50A4 can be used to assign KTY cold-junction compensation to a channel.

Feature	Description / Value	EDS
Name	Reference Input Select	[MxSubExt50A4 ParameterName=ReferenceInputSelect ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=0 PDOMapping=0 Count=1
Description	ReferenceInputSelect	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
								LSB

Byte 0:

Data bit	Designation	Description
0	Channel 1	0 = KTY 1 1 = KTY 2
1	Channel 2	0 = KTY 1 1 = KTY 2
2	Channel 3	0 = KTY 1 1 = KTY 2
3	Channel 4	0 = KTY 1 1 = KTY 2
4	Channel 5	0 = KTY 1 1 = KTY 2
5	Channel 6	0 = KTY 1 1 = KTY 2
6	Channel 7	0 = KTY 1 1 = KTY 2
7	Channel 8	0 = KTY 1 1 = KTY 2

7.18 XN-322-8AO-U2

This module supports the provision of data for analog outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

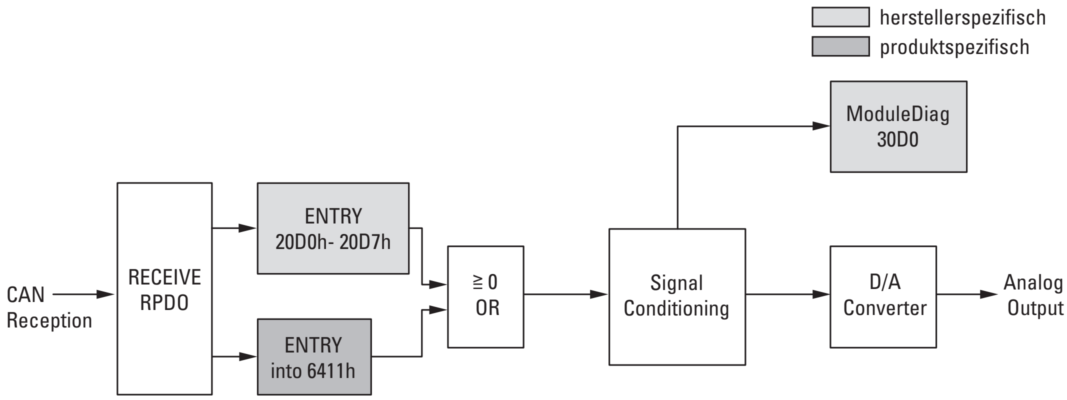


Figure 75: Block diagram showing the various CANopen objects for analog outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x6411	INTEGER16	Q-WORD	Write Analog Output 16-bit	Default	rww	PDO

Manufacturer-specific objects

Index range for the XN-322-8AO-U2 module: x0D0 to x0DF

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x20D0	INTEGER16	OutputChannel1	Output Channel 1	Manual	rw	PDO
0x20D1	INTEGER16	OutputChannel2	Output Channel 2	Manual	rw	PDO
0x20D2	INTEGER16	OutputChannel3	Output Channel 3	Manual	rw	PDO
0x20D3	INTEGER16	OutputChannel4	Output Channel 4	Manual	rw	PDO
0x20D4	INTEGER16	OutputChannel5	Output Channel 5	Manual	rw	PDO
0x20D5	INTEGER16	OutputChannel6	Output Channel 6	Manual	rw	PDO
0x20D6	INTEGER16	OutputChannel7	Output Channel 7	Manual	rw	PDO
0x20D7	INTEGER16	OutputChannel8	Output Channel 8	Manual	rw	PDO
0x30D0	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO
0x40D0	UNSIGNED16	FirmwareVersion	Firmware Version	–	ro	SDO

7.18.1 Write Analog Output 16-Bit (Object 0x6411)

Object 0x6411 transmits the digital value of the channels' analog signal outputs. The object's data is automatically entered into the receive PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-WORD	[MxSubExt6411] ParameterName=Q-WORD ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=1 Count=8
Description	Write Analog Output 16-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	0000_{hex}	

Design of the data bytes:

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

7.18.2 Output Channel x (Object 0x20D0 to 0x20D7)

Objects 0x20D0 to 0x20D7 transmit the digital value of the channels' analog signal outputs.

Feature	Description / Value		EDS
Name	OutputChannel1	Object 0x20D0	[MxSubExt20Dx] ParameterName=OutputChannelx ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=1 Count=1
	OutputChannel2	Object 0x20D1	
	OutputChannel3	Object 0x20D2	
	OutputChannel4	Object 0x20D3	
	OutputChannel5	Object 0x20D4	
	OutputChannel6	Object 0x20D5	
	OutputChannel7	Object 0x20D6	
	OutputChannel8	Object 0x20D7	
Description	Output Channel x		
Object Code	ARRAY		
Mapping	PDO		
	MANUAL		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	rww		
Default value	00000_{hex}		

Design of the data bytes:

Sub index 1≦n≦254	**1 Byte**								**Byte 0:**							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.18.3 Module Diagnostic Messages (Object 0x30D0)

Object 0x30D0 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	ModuleDiag	[MxSubExt30D0] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Module Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	–	–					

Data bit	Description
0	reserved
1	0 = sync OK 1 = no sync
2	0 = OK 1 = FLASH Data CRC Error
3	0 = OK 1 =RAM Data CRC Error
4	0 = OK 1 = Inconsistent FLASH Data
5 – 7	reserved

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Description
8 – 15	reserved

7.18.4 Firmware Version (Object 0x40D0)

Object 0x40D0 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	FirmwareVersion	[MxSubExt40D0] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	Firmware Version	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

7.19 XN-322-4AIO-U2

This module supports the provision of data for analog inputs and analog outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

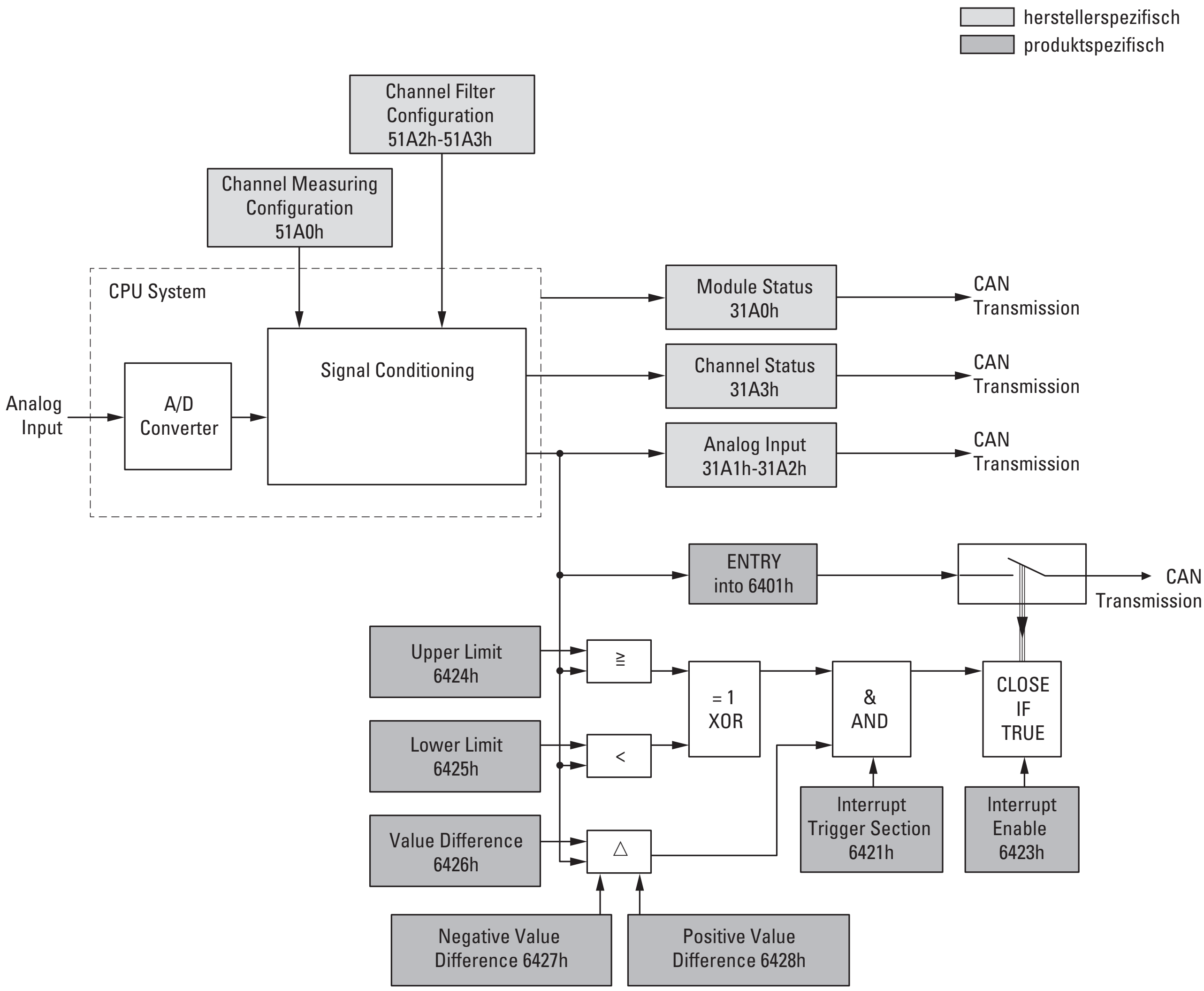


Figure 76: Block diagram showing the various CANopen objects for analog inputs

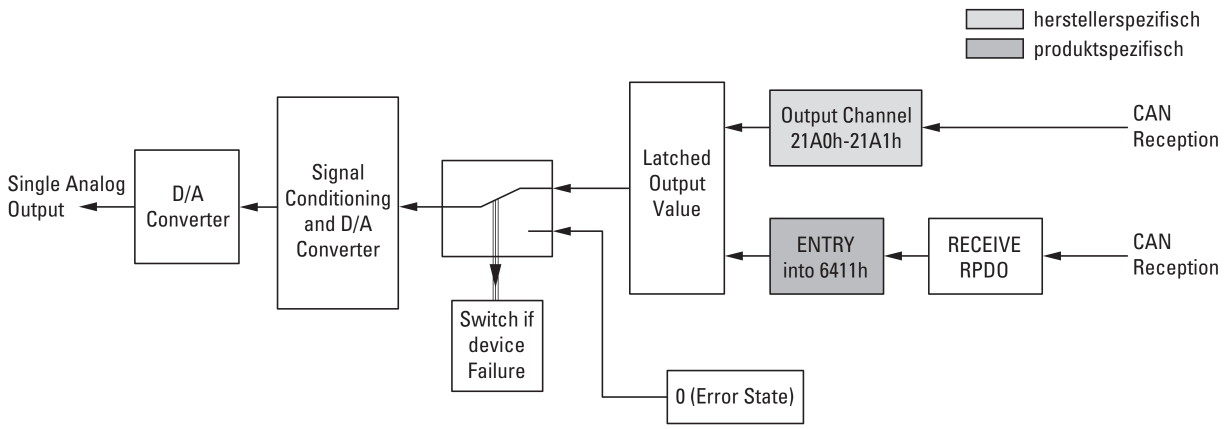


Figure 77: Block diagram showing the various CANopen objects for analog outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x6401	INTEGER16	I-WORD	Read Analog Input 16-bit	Default	ro	PDO
0x6411	INTEGER16	Q-WORD	Write Analog Output 16-bit	Default	rww	PDO
0x6421	UNSIGNED8	AI_INTERRUPT_TRIGGER_SELECTION	Analog Input Interrupt Trigger Selection	–	rw	SDO
0x6423	BOOLEAN	AnalogInputGlobalInterruptEnable	Analog Input Global Interrupt Enable	–	rw	SDO
0x6424	INTEGER32	AI_UPPER_LIMIT	Analog Input Interrupt Upper Limit Integer	–	rw	SDO
0x6425	INTEGER32	AI_LOWER_LIMIT	Analog Input Interrupt Lower Limit Integer	–	rw	SDO
0x6426	UNSIGNED32	AI_DELTA_VALUE	Analog Input Interrupt Delta Unsigned	–	rw	SDO
0x6427	UNSIGNED32	AI_NEGATIVE_DELTA_VALUE	Analog Input Interrupt Negative Delta Unsigned	–	rw	SDO
0x6428	UNSIGNED32	AI_POSITIVE_DELTA_VALUE	Analog Input Interrupt Positive Delta Unsigned	–	rw	SDO

Manufacturer-specific objects

Index range for the XN-322-4AIO-U2: x1A0 to x1AF

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x21A0	INTEGER16	OutputChannel1	Output Channel 1	Manual	rww	PDO
0x21A1	INTEGER16	OutputChannel2	Output Channel 2	Manual	rww	PDO
0x31A0	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x31A1	INTEGER16	InputChannel1	Input Channel 1	Manual	ro	PDO
0x31A2	INTEGER16	InputChannel2	Input Channel 2	Manual	ro	PDO
0x31A3	UNSIGNED16	WireBreakDiag	Wire Break Diagnostic Messages	Manual	ro	PDO

0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO
0x41A0	UNSIGNED16	FirmwareVersion	Firmware Version	–	ro	SDO
0x51A0	UNSIGNED16	AnalogInputSelection	Analog Input Selection	–	rw	SDO
0x51A1	UNSIGNED16	FilterConfigChannel1	Filter Configuration Channel 1	–	rw	SDO
0x51A2	UNSIGNED16	FilterConfigChannel2	Filter Configuration Channel 2	–	rw	SDO

7.19.1 Read Analog Input 16-Bit (Object 0x6401)

Object 0x6401 represents the channels' formatted analog input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-WORD	[MxSubExt6401] ParameterName=I-WORD ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=2
Description	Read Analog Input 16-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.19.2 Write Analog Output 16-Bit (Object 0x6411)

Object 0x6411 transmits the digital value of the channels' analog signal outputs. The object's data is automatically entered into the receive PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-WORD	[MxSubExt6411] ParameterName=Q-WORD ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=1 Count=2
Description	Write Analog Output 16-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

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

7.19.3 Analog Input Interrupt Trigger Selection (Object 0x6421)

Object 0x6421 defines which events will trigger an interrupt for the corresponding channel so that the channel's analog input value (object 0x6401) will be transmitted when the event occurs.

Feature	Description / Value	EDS
Name	AI_INTERRUPT_TRIGGER_SELECTION	[MxSubExt6421] Parameter- Name=AI_INTERRUPT_TRIGGER_SELECTION ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=7 LowLimit=0 HighLimit=31 PDOMapping=0 Count=2
Description	Analog Input Interrupt Trigger Selection	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	07_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Meaning
0		0 = Upper limit not exceeded 1 = Upper limit exceeded
1		0 = Input not below lower limit 1 = Input below lower limit
2		0 = Input not changed by more than delta 1 = Input changed by more than delta
3		0 = Input not reduced by more than negative delta 1 = Input reduced by more than negative d
4		0 = Input not reduced by more than positive delta 1 = Input reduced by more than positive delta
5 – 7		reserved

7.19.4 Analog Input Global Interrupt Enable (Object 0x6423)

Object 0x6423 enables and disables the transmission of data via object 0x6401 in general.

Feature	Description / Value	EDS
Name	AnalogInputGlobalInterruptEnable	[MxFixed6423] ParameterName=AnalogInputGlobalInterruptEnable ObjectType=0x7 DataType=0x0001 AccessType=rw DefaultValue=0 PDOMapping=0
Description	Analog Input Global Interrupt Enable	
Object Code	Variable	
Mapping	SDO	
Data type	BOOLEAN	
Access	rw	
Default value	FALSE	
Object Code	Variable	

Default value

- FALSE (0)
  Object 0x6401 is not enabled and cannot transmit any analog input values.
- TRUE (1)
  Object 0x6401 is enabled and can transmit analog input values.

7.19.5 Analog Input Interrupt Upper Limit Integer (Object 0x6424)

Object 0x6424 defines an upper limit and affects the transmission of object 0x6401 accordingly → Figure 76, page 234.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) is equal to or greater than the upper limit (Upper Limit 0x6424).
- Object 0x6421 is set to enable.

- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_UPPER_LIMIT	[MxSubExt6424] ParameterName=AI_UPPER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=2
Description	Analog Input Interrupt Upper Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.19.6 Analog Input Interrupt Lower Limit Integer (Object 0x6425)

Object 0x6425 defines a lower limit and affects the transmission of object 0x6401 accordingly → Figure 76, page 234.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) falls below the lower limit (Lower Limit 0x6425).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_LOWER_LIMIT	[MxSubExt6425] ParameterName=AI_LOWER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=2
Description	Analog Input Interrupt Lower Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.19.7 Analog Input Interrupt Delta Unsigned (Object 0x6426)

Object 0x6426 defines the fluctuation range (delta) relative to the last transmitted value both when counting up and down and affects the transmission of object 0x6401 accordingly → Figure 76, page 234.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The difference between the current value (ENTRY into 0x6401) and the last transmitted value is equal to or greater than the fluctuation range (Value Difference 0x6426).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_DELTA_VALUE	[MxSubExt6426] ParameterName=AI_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=2
Description	Analog Input Interrupt Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01... FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.19.8 Analog Input Interrupt Negative Delta Unsigned (Object 0x6427)

Object 0x6427 defines the negative fluctuation range (delta) relative to the last transmitted value when counting down and affects the transmission of object 0x6401 accordingly → Figure 76, page 234.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is less than the last transmitted value and the difference is greater than or equal to the fluctuation range (Negative Value Difference 0x6427).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_NEGATIVE_DELTA_VALUE	[MxSubExt6427] ParameterName=AI_NEGATIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=2
Description	Analog Input Interrupt Negative Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE $_{ex}$	
Access	rw	
Default value	00000000$_{hex}$	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.19.9 Analog Input Interrupt Positive Delta Unsigned (Object 0x6428)

Object 0x6428 defines the negative fluctuation range (delta) relative to the last transmitted value when counting up and affects the transmission of object 0x6401 accordingly → Figure 76, page 234.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is greater than the last transmitted value. The difference's magnitude is greater than the fluctuation range (Positive Value Difference 0x6428).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

If the current value is less than the last transmitted value, object 0x6401 will be transmitted. If this behavior is not desired, object 0x6421 must be enabled and disabled accordingly.

Feature	Description / Value	EDS
Name	AI_POSITIVE_DELTA_VALUE	[MxSubExt6428] ParameterName=AI_POSITIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=2
Description	Analog Input Interrupt Positive Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															LSB

7.19.10 Output Channel x (Object 0x21A0 bis 0x21A1)

Objects 0x21A0 to 0x21A1 transmit the digital value of the channels' analog signal outputs.

Feature	Description / Value		EDS
Name	Output Channel x		[MxSubExt21Ax] ParameterName=OutputChannelx ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=1 Count=1
Description	OutputChannel1	Object 0x21A0	
	OutputChannel2	Object 0x21A1	
Object Code	ARRAY		
Mapping	PDO , MANUAL		
Data type	INTEGER16		
Sub index	01 … FE hex		
Access	rww		
Default value	0 x 0000 hex		

Design of the data bytes:

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

7.19.11 Module Diagnostic Messages (Object 0x31A0)

Object 0x31A0 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	ModuleDiag	[MxSubExt31A0] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Module Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–					

Data bit	Designation	Description
0		reserved
1		0 = sync OK 1 = no sync
2		0 = OK 1 = FLASH Data CRC Error
3		0 = OK 1 =RAM Data CRC Error
4		0 = OK 1 = Inconsistent FLASH Data
5 – 7		reserved

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	–

Data bit	Desig-nation	Description
8 – 15		reserved

7.19.12 Input Channel x (Object 0x31A1 to 0x31A2)

Objects 0x31A1 to 0x31A2 represent the formatted (integer) analog input values.

Feature	Description / Value		EDS
Name	InputChannelx		[MxSubExt31Ax] ParameterName=InputChannelx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
Description	InputChannel1	Object 0x31A1	
	InputChannel2	Object 0x31A2	
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.19.13 Wire Break Diagnostic Messages (Object 0x31A3)

Object 0x31A3 contains status information on the module's channels.

Feature	Description / Value	EDS
Name	WireBreakDiag	[MxSubExt31A3] ParameterName=WireBreakDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Wire Break Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
0		0 = Channel 1 OK 1 = Channel 1 wire breakage
1		0 = Channel 2 OK 1 = Channel 2wire breakage
2		reserved
3		reserved
4		0 = OK 1 = Reference Low Voltage
5		0 = OK 1 = Reference OverCurrent
6 – 7		reserved

7.19.14 Firmware Version (Object 0x41A0)

Object 0x41A0 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	FirmwareVersion	[MxSubExt41A0] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	Firmware Version	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

7.19.15 Analog Input Selection (Object 0x51A0)

Object 0x51A0 must be used to configure the channels' measuring method.

Feature	Description / Value	EDS
Name	AnalogInputSelection	[MxSubExt51A0 ParameterName=AnalogInputSelection ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
Description	Analog Input Selection	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000_{hex}	

Design of the data bytes:

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

Byte 0:

Data bit	Desig-nation	Description
0	Channel 1	0 = Differential measurement 1 = AI1-grounded measurement
1	Channel 2	0 = Differential measurement 1 = AI2-grounded measurement
2 – 7		reserved

Byte 1:

Data bit	Desig-nation	Description
8 – 15		reserved

7.19.16 Filter Configuration Channel x (Object 0x51A1 bis 0x51A2)

Objects 0x51A1 to 0x51A2 can be used to configure the software filter for a channel.

Feature	Description / Value		EDS
Name	FilterConfigChannelx		[MxSubExt51Ax] ParameterName=FilterConfigChannelx ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
Description	FilterConfigChannel1	Object 0x51A1	
	FilterConfigChannel2	Object 0x51A2	
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	rw		

Design of the data bytes:

Low-pass cut-off frequency in Hz (example: 50 Hz => 0032_{hex})

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

Example

Low-pass cut-off frequency in Hz

50 Hz = 0032_{hex}

The following settings are valid:

Low-pass cut-off frequency	Register value
Filter disabled (default)	0x0000
10 Hz	0x000A
25 Hz	0x0019
50 Hz	0 x0032
100 Hz	0x0064
250 Hz	0x00FA
500 Hz	0x01F4
1000 Hz	0x03E8

7.20 XN-322-8AIO-U2

This module supports the provision of data for analog inputs and analog outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

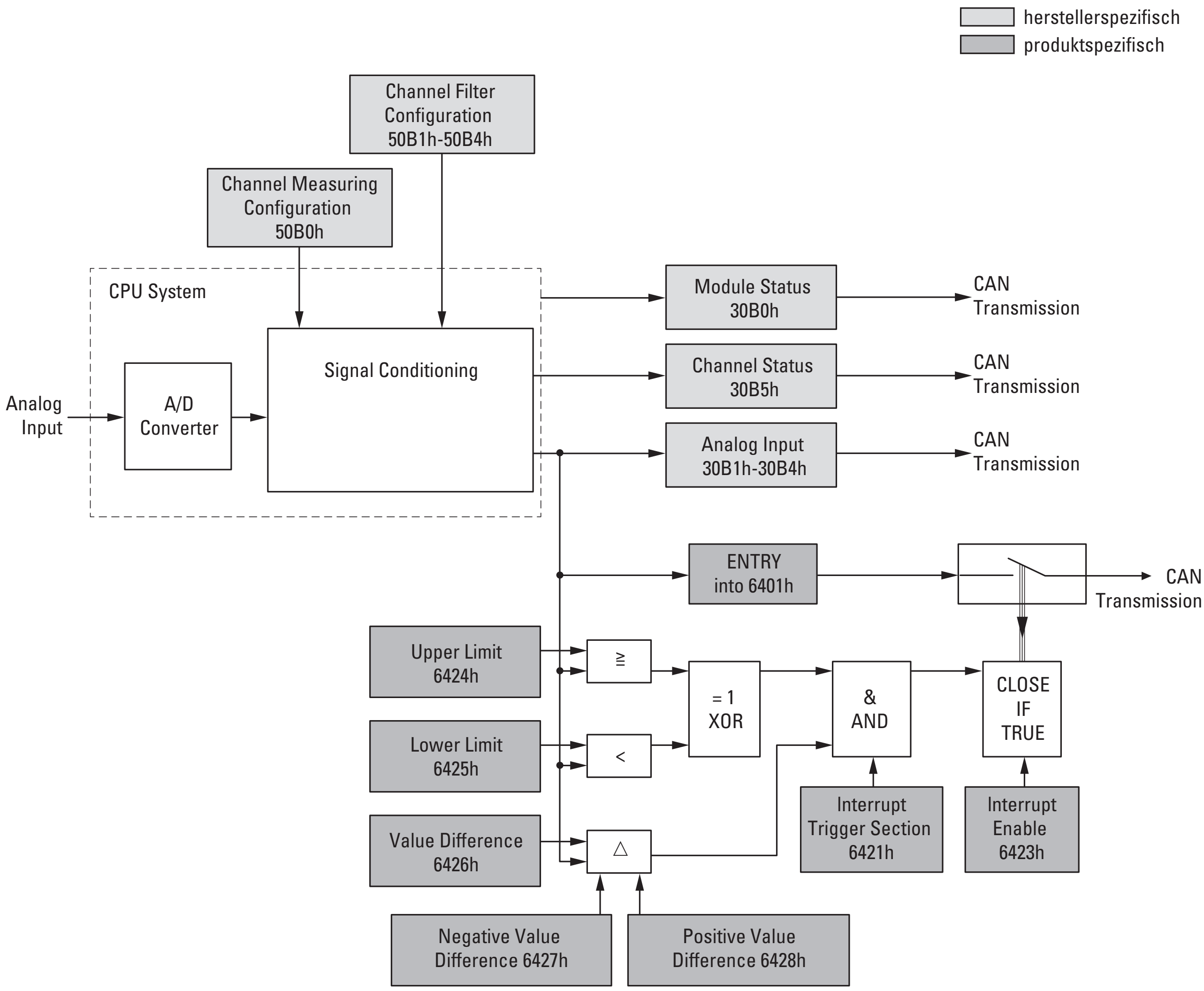


Figure 78: Block diagram showing the various CANopen objects for analog inputs

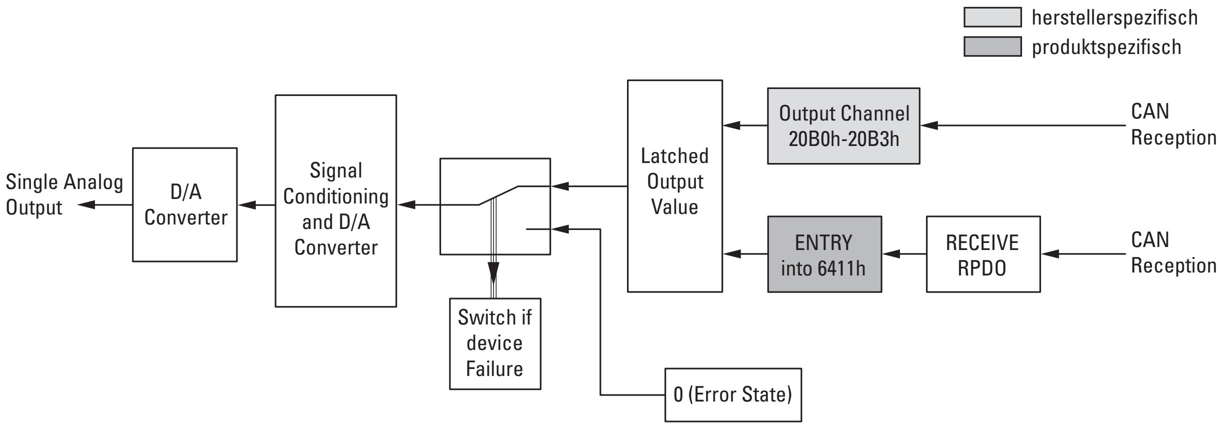


Figure 79: Block diagram showing the various CANopen objects for analog outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x6401	INTEGER16	I-WORD	Read Analog Input 16-bit	Default	ro	PDO
0x6411	INTEGER16	Q-WORD	Write Analog Output 16-bit	Default	rww	PDO
0x6421	UNSIGNED8	AI_INTERRUPT_TRIGGER_SELECTION	Analog Input Interrupt Trigger Selection	–	rw	SDO
0x6423	BOOLEAN	AnalogInputGlobalInterruptEnable	Analog Input Global Interrupt Enable	–	rw	SDO
0x6424	INTEGER32	AI_UPPER_LIMIT	Analog Input Interrupt Upper Limit Integer	–	rw	SDO
0x6425	INTEGER32	AI_LOWER_LIMIT	Analog Input Interrupt Lower Limit Integer	–	rw	SDO
0x6426	UNSIGNED32	AI_DELTA_VALUE	Analog Input Interrupt Delta Unsigned	–	rw	SDO
0x6427	UNSIGNED32	AI_NEGATIVE_DELTA_VALUE	Analog Input Interrupt Negative Delta Unsigned	–	rw	SDO
0x6428	UNSIGNED32	AI_POSITIVE_DELTA_VALUE	Analog Input Interrupt Positive Delta Unsigned	–	rw	SDO

Manufacturer-specific objects

Index range for the XN-322-8AIO-U2: x0B0 to x0BF

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x20B0	INTEGER16	OutputChannel1	Output Channel 1	Manual	rww	PDO
0x20B1	INTEGER16	OutputChannel2	Output Channel 2	Manual	rww	PDO
0x20B2	INTEGER16	OutputChannel3	Output Channel 3	Manual	rww	PDO
0x20B3	INTEGER16	OutputChannel4	Output Channel 4	Manual	rww	PDO
0x30B0	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x30B1	INTEGER16	InputChannel1	Input Channel 1	Manual	ro	PDO
0x30B2	INTEGER16	InputChannel2	Input Channel 2	Manual	ro	PDO

0x30B3	INTEGER16	InputChannel3	Input Channel 3	Manual	ro	PDO
0x30B4	INTEGER16	InputChannel4	Input Channel 4	Manual	ro	PDO
0x30B5	UNSIGNED16	WireBreakDiag	Wire Break Diagnostic Messages	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section “6.2.8 Serial Number (Object 0x4001)”, page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section “6.2.11 User LED Control (Object 0x4004)”, page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section “6.2.14 Product Name (Object 0x400C)”, page 82	–	ro	SDO
0x40B0	UNSIGNED16	FirmwareVersion	Firmware Version	–	ro	SDO
0x50B0	UNSIGNED16	AnalogInputSelection	Analog Input Selection	–	rw	SDO
0x50B1	UNSIGNED16	FilterConfigChannel1	Filter Configuration Channel 1	–	rw	SDO
0x50B2	UNSIGNED16	FilterConfigChannel2	Filter Configuration Channel 2	–	rw	SDO
0x50B3	UNSIGNED16	FilterConfigChannel3	Filter Configuration Channel 3	–	rw	SDO
0x50B4	UNSIGNED16	FilterConfigChannel4	Filter Configuration Channel 4	–	rw	SDO

7.20.1 Read Analog Input 16-Bit (Object 0x6401)

Object 0x6401 represents the channels' formatted analog input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	Read Analog Input 16-Bit	[MxSubExt6401] ParameterName=I-WORD ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=4
Description	I-WORD	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

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

7.20.2 Write Analog Output 16-Bit (Object 0x6411)

Object 0x6411 transmits the digital value of the channels' analog signal outputs. The object's data is automatically entered into the receive PDOs (default mapping).

Feature	Description / Value	EDS
Name	Write Analog Output 16-Bit	[MxSubExt6411] ParameterName=Q-WORD ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=1 Count=4
Description	Q-WORD	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

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

7.20.3 Analog Input Interrupt Trigger Selection (Object 0x6421)

Object 0x6421 defines which events will trigger an interrupt for the corresponding channel so that the channel's analog input value (object 0x6401) will be transmitted when the event occurs.

Feature	Description / Value	EDS
Name	Analog Input Interrupt Trigger Selection	[MxSubExt6421] Parameter- Name=AI_INTERRUPT_TRIGGER_SELECTIO N ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=7 LowLimit=0 HighLimit=31 PDOMapping=0 Count=4
Description	AI_INTERRUPT_TRIGGER_SELECTION	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	07_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
0		0 = Upper limit not exceeded 1 = Upper limit exceeded
1		0 = Input not below lower limit 1 = Input below lower limit
2		0 = Input not changed by more than delta 1 = Input changed by more than delta
3		0 = Input not reduced by more than negative delta 1 = Input reduced by more than negative d
4		0 = Input not reduced by more than positive delta 1 = Input reduced by more than positive delta
5 – 7		reserved

7.20.4 Analog Input Global Interrupt Enable (Object 0x6423)

Object 0x6423 enables and disables the transmission of data via object 0x6401 in general.

Feature	Description / Value	EDS
Name	Analog Input Global Interrupt Enable	[MxFixed6423] ParameterName=AnalogInputGlobalInterruptEnable ObjectType=0x7 DataType=0x0001 AccessType=rw DefaultValue=0 PDOMapping=0
Description	AnalogInputGlobalInterruptEnable	
Object Code	Variable	
Mapping	SDO	
Data type	BOOLEAN	
Access	rw	
Default value	FALSE	
Object Code	Variable	

Default value

- FALSE (0)
  Object 0x6401 is not enabled and cannot transmit any analog input values.
- TRUE (1)
  Object 0x6401 is enabled and can transmit analog input values.

7.20.5 Analog Input Interrupt Upper Limit Integer (Object 0x6424)

Object 0x6424 defines an upper limit and affects the transmission of object 0x6401 accordingly → Figure 78, page 250.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) is equal to or greater than the upper limit (Upper Limit 0x6424).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_UPPER_LIMIT	[MxSubExt6424] ParameterName=AI_UPPER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=4
Description	Analog Input Interrupt Upper Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.20.6 Analog Input Interrupt Lower Limit Integer (Object 0x6425)

Object 0x6425 defines a lower limit and affects the transmission of object 0x6401 accordingly → Figure 78, page 250.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) falls below the lower limit (Lower Limit 0x6425).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_LOWER_LIMIT	[MxSubExt6425] ParameterName=AI_LOWER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=4
Description	Analog Input Interrupt Lower Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.20.7 Analog Input Interrupt Delta Unsigned (Object 0x6426)

Object 0x6426 defines the fluctuation range (delta) relative to the last transmitted value both when counting up and down and affects the transmission of object 0x6401 accordingly → Figure 78, page 250.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The difference between the current value (ENTRY into 0x6401) and the last transmitted value is equal to or greater than the fluctuation range (Value Difference 0x6426).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_DELTA_VALUE	[MxSubExt6426] ParameterName=AI_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=4
Description	Analog Input Interrupt Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01... FE$_{hex}$	
Access	rw	
Default value	0000 0000$_{hex}$	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.20.8 Analog Input Interrupt Negative Delta Unsigned (Object 0x6427)

Object 0x6427 defines the negative fluctuation range (delta) relative to the last transmitted value when counting down and affects the transmission of object 0x6401 accordingly → Figure 78, page 250.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is less than the last transmitted value and the difference is greater than or equal to the fluctuation range (Negative Value Difference 0x6427).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_NEGATIVE_DELTA_VALUE	[MxSubExt6427] ParameterName=AI_NEGATIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=4
Description	Analog Input Interrupt Negative Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE $_{ex}$	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

	3 Byte								Byte 2							
Sub index 1≦n≦254	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.20.9 Analog Input Interrupt Positive Delta Unsigned (Object 0x6428)

Object 0x6428 defines the negative fluctuation range (delta) relative to the last transmitted value when counting up and affects the transmission of object 0x6401 accordingly → Figure 78, page 250.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is greater than the last transmitted value. The difference's magnitude is greater than the fluctuation range (Positive Value Difference 0x6428).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

If the current value is less than the last transmitted value, object 0x6401 will be transmitted. If this behavior is not desired, object 0x6421 must be enabled and disabled accordingly.

Feature	Description / Value	EDS
Name	AI_POSITIVE_DELTA_VALUE	[MxSubExt6428] ParameterName=AI_POSITIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=4
Description	Analog Input Interrupt Positive Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															LSB

7.20.10 Output Channel x (Object 0x20B0 to 0x20B3)

Objects 0x20B0 to 0x20B3 transmit the digital value of the channels' analog signal outputs.

Feature	Description / Value		EDS
Name	OutputChannel1	Object 0x20B0	[MxSubExt20Bx] ParameterName=OutputChannelx ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=0 Count=1
	OutputChannel2	Object 0x20B1	
	OutputChannel3	Object 0x20B2	
	OutputChannel4	Object 0x20B3	
Description	Output Channel x		
Object Code	ARRAY		
Mapping	PDO , MANUAL		
Data type	INTEGER16		
Sub index	01 … FE hex		
Access	rww		
Default value	0 x 0000 hex		

Design of the data bytes:

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

7.20.11 Module Diagnostic Messages (Object 0x30B0)

Object 0x30B0 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	ModuleDiag	[MxSubExt3060] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Module Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–					

Data bit	Designation	Description
0		0 = 24VDC OK 1 = 24VDC not OK
1		0 = sync OK 1 = no sync
2		0 = OK 1 = FLASH Data CRC Error
3		0 = OK 1 =RAM Data CRC Error
4		0 = OK 1 = Inconsistent FLASH Data
5 – 7		reserved

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Description
8 – 15		reserved

7.20.12 Input Channel x (Object 0x30B1 to 0x30B4)

Objects 0x30B1 to 0x30B4 represent the formatted (integer) analog input values.

Feature	Description / Value		EDS
Name	InputChannel1	Object 0x30B1	[MxSubExt30Bx] ParameterName=InputChannelx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
	InputChannel2	Object 0x30B2	
	InputChannel3	Object 0x30B3	
	InputChannel4	Object 0x30B4	
Description	Input Channel x		
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.20.13 Wire Break Diagnostic Messages (Object 0x30B5)

Object 0x30B5 contains status information on the module's channels.

Feature	Description / Value	EDS
Name	Wire Break Detection	[MxSubExt30B5] ParameterName=WireBreakDetect ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	WireBreakDetect	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
0		0 = Channel 1 OK 1 = Channel 1 wire breakage
1		0 = Channel 2 OK 1 = Channel 2wire breakage
2		0 = Channel 3 OK 1 = Channel 3 wire breakage
3		0 = Channel 4 OK 1 = Channel 4 wire breakage
4		0 = OK 1 = LowReference
5		0 = OK 1 = AI OverCurrent
6 – 7		reserved

7.20.14 Firmware Version (Object 0x40B0)

Object 0x40B0 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	Firmware Version	[MxSubExt40B0] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	FirmwareVersion	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

7.20.15 Analog Input Selection (Object 0x50B0)

Object 0x50B0 must be used to configure the channels' measuring method.

Feature	Description / Value	EDS
Name	Analog Input Selection	[MxSubExt50B0 ParameterName=AnalogInputSelection ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
Description	AnalogInputSelection	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000_{hex}	

Design of the data bytes:

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

Byte 0:

Data bit	Designation	Description
0	Channel 1	0 = Differential measurement 1 = AI1-grounded measurement
1	Channel 2	0 = Differential measurement 1 = AI2-grounded measurement
2	Channel 3	0 = Differential measurement 1 = AI3-grounded measurement
3	Channel 4	0 = Differential measurement 1 = AI4-grounded measurement
4 – 7		reserved

Byte 1:

Data bit	Designation	Description
8 – 15		reserved

7.20.16 Filter Configuration Channel x (Object 0x50B1 to 0x50B4)

Objects 0x50B1 to 0x50B4 can be used to configure the software filter for a channel.

Feature	Description / Value		EDS
Name	Filter Configuration Channel x		[MxSubExt50Bx] ParameterName=FilterConfigChannelx ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
Description	FilterConfigChannel1	Object 0x50B1	
	FilterConfigChannel2	Object 0x50B2	
	FilterConfigChannel3	Object 0x50B3	
	FilterConfigChannel4	Object 0x50B4	
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	rw		
Default value	0000_{hex}		

Design of the data bytes:

Low-pass cut-off frequency in Hz (example: 50 Hz => 0032_{hex})

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

Example

Low-pass cut-off frequency in Hz

50 Hz = 0032_{hex}

The following settings are valid:

Low-pass cut-off frequency	Register value
Filter disabled (default)	0x0000
10 Hz	0x000A
25 Hz	0x0019
50 Hz	0 x0032
100 Hz	0x0064
250 Hz	0x00FA
500 Hz	0x01F4
1000 Hz	0x03E8

7.21 XN-322-4AIO-I

This module supports the provision of data for analog inputs and analog outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

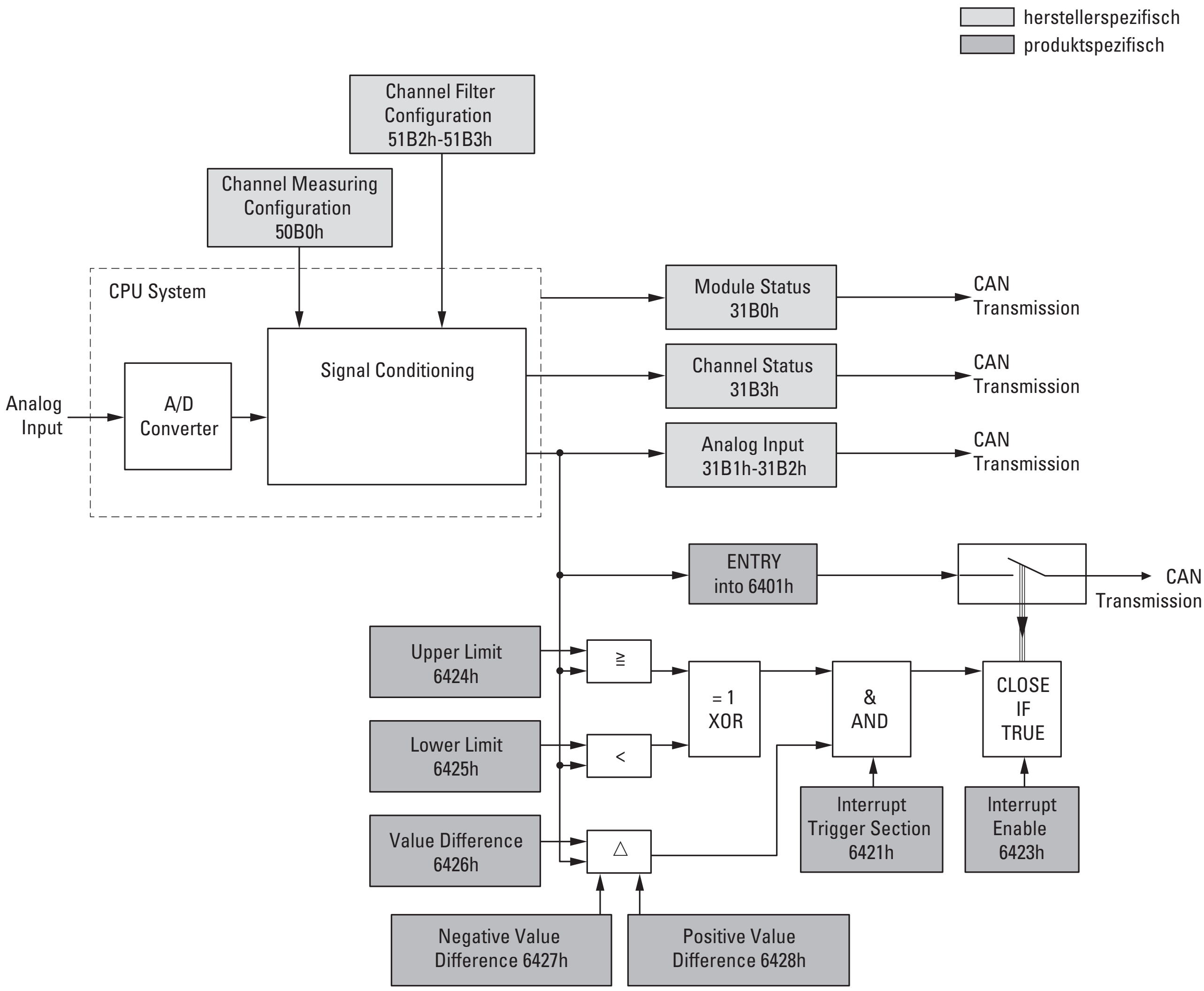


Figure 80: Block diagram showing the various CANopen objects for analog inputs

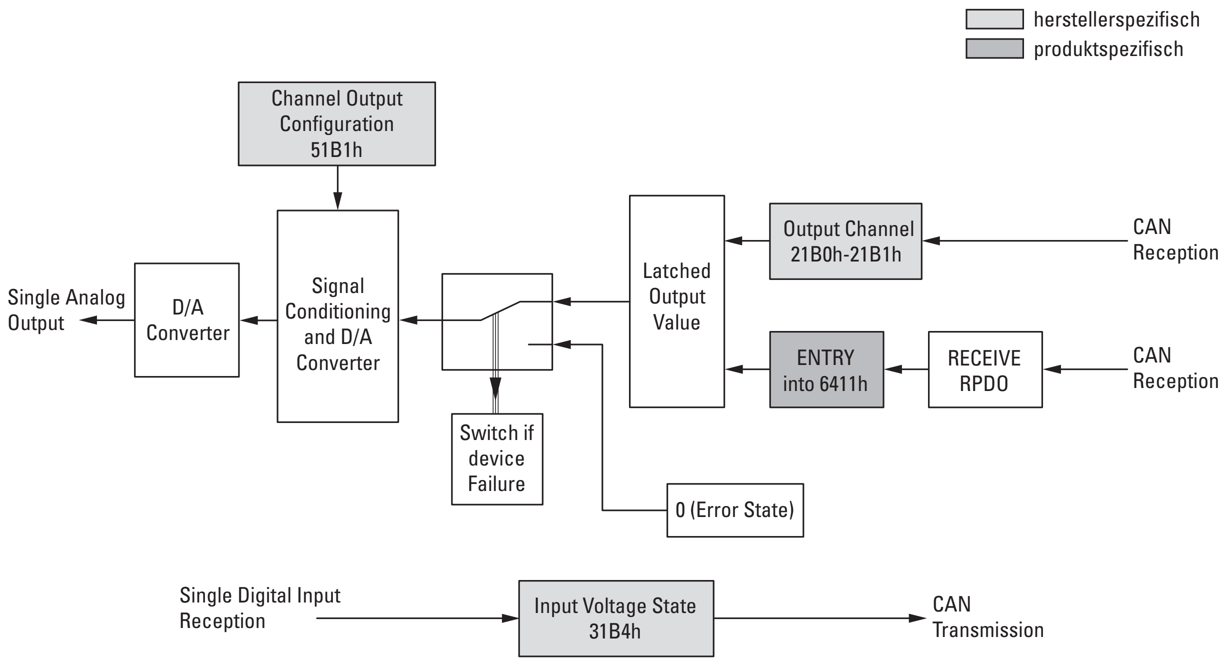


Figure 81: Block diagram showing the various CANopen objects for analog outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x6401	INTEGER16	I-WORD	Read Analog Input 16-bit	Default	ro	PDO
0x6411	INTEGER16	Q-WORD	Write Analog Output 16-bit	Default	rww	PDO
0x6421	UNSIGNED8	AI_INTERRUPT_TRIGGER_SELECTION	Analog Input Interrupt Trigger Selection	–	rw	SDO
0x6423	BOOLEAN	AnalogInputGlobalInterruptEnable	Analog Input Global Interrupt Enable	–	rw	SDO
0x6424	INTEGER32	AI_UPPER_LIMIT	Analog Input Interrupt Upper Limit Integer	–	rw	SDO
0x6425	INTEGER32	AI_LOWER_LIMIT	Analog Input Interrupt Lower Limit Integer	–	rw	SDO
0x6426	UNSIGNED32	AI_DELTA_VALUE	Analog Input Interrupt Delta Unsigned	–	rw	SDO
0x6427	UNSIGNED32	AI_NEGATIVE_DELTA_VALUE	Analog Input Interrupt Negative Delta Unsigned	–	rw	SDO
0x6428	UNSIGNED32	AI_POSITIVE_DELTA_VALUE	Analog Input Interrupt Positive Delta Unsigned	–	rw	SDO

Manufacturer-specific objects

Index range for the XN-322-4AIO-I: x1B0 to x1BF

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x21B0	INTEGER16	OutputChannel1	Output Channel 1	Manual	rww	PDO
0x21B1	INTEGER16	OutputChannel2	Output Channel 2	Manual	rww	PDO
0x31B0	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x31B1	INTEGER16	InputChannel1	Input Channel 1	Manual	ro	PDO
0x31B2	INTEGER16	InputChannel2	Input Channel 2	Manual	ro	PDO
0x31B3	UNSIGNED8	ChannelDiag	Channel Diagnostic Messages	Manual	ro	PDO
0x31B4	UNSIGNED8	InputVoltageState	Input Voltage State	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO
0x41B0	UNSIGNED16	FirmwareVersion	Firmware Version	–	ro	SDO
0x51B0	UNSIGNED8	InputChannelConfig	Channel Measuring Configuration (Measurement range 0…20mA/4…20mA)	–	rw	SDO
0x51B1	UNSIGNED8	OutputChannelConfig	Channel Output Configuration (Measurement range 0…20mA)	–	rw	SDO
0x51B2	UNSIGNED16	FilterConfigChannel1	Filter Configuration Channel 1	–	rw	SDO
0x51B3	UNSIGNED16	FilterConfigChannel2	Filter Configuration Channel 2	–	rw	SDO

7.21.1 Read Analog Input 16-Bit (Object 0x6401)

Object 0x6401 represents the channels' formatted analog input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-WORD	[MxSubExt6401] ParameterName=I-WORD ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=2
Description	Read Analog Input 16-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

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

7.21.2 Write Analog Output 16-Bit (Object 0x6411)

Object 0x6411 transmits the digital value of the channels' analog signal outputs. The object's data is automatically entered into the receive PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-WORD	[MxSubExt6411] ParameterName=Q-WORD ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=1 Count=2
Description	Write Analog Output 16-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

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

7.21.3 Analog Input Interrupt Trigger Selection (Object 0x6421)

Object 0x6421 defines which events will trigger an interrupt for the corresponding channel so that the channel's analog input value (object 0x6401) will be transmitted when the event occurs.

Feature	Description / Value	EDS
Name	AI_INTERRUPT_TRIGGER_SELECTION	[MxSubExt6421] Parameter-Name=AI_INTERRUPT_TRIGGER_SELECTION ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=7 LowLimit=0 HighLimit=31 PDOMapping=0 Count=2
Description	Analog Input Interrupt Trigger Selection	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	07_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Meaning
0		0 = Upper limit not exceeded 1 = Upper limit exceeded
1		0 = Input not below lower limit 1 = Input below lower limit
2		0 = Input not changed by more than delta 1 = Input changed by more than delta
3		0 = Input not reduced by more than negative delta 1 = Input reduced by more than negative d
4		0 = Input not reduced by more than positive delta 1 = Input reduced by more than positive delta
5 – 7		reserved

7.21.4 Analog Input Global Interrupt Enable (Object 0x6423)

Object 0x6423 enables and disables the transmission of data via object 0x6401 in general.

Feature	Description / Value	EDS
Name	AnalogInputGlobalInterruptEnable	[MxFixed6423] ParameterName=AnalogInputGlobalInterruptEnable ObjectType=0x7 DataType=0x0001 AccessType=rw DefaultValue=0 PDOMapping=0
Description	Analog Input Global Interrupt Enable	
Object Code	Variable	
Mapping	SDO	
Data type	BOOLEAN	
Access	rw	
Default value	FALSE	
Object Code	Variable	

Default value

- FALSE (0)
  Object 0x6401 is not enabled and cannot transmit any analog input values.
- TRUE (1)
  Object 0x6401 is enabled and can transmit analog input values.

7.21.5 Analog Input Interrupt Upper Limit Integer (Object 0x6424)

Object 0x6424 defines an upper limit and affects the transmission of object 0x6401 accordingly → Figure 80, page 266.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) is equal to or greater than the upper limit (Upper Limit 0x6424).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_UPPER_LIMIT	[MxSubExt6424] ParameterName=AI_UPPER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=2
Description	Analog Input Interrupt Upper Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.21.6 Analog Input Interrupt Lower Limit Integer (Object 0x6425)

Object 0x6425 defines a lower limit and affects the transmission of object 0x6401 accordingly → Figure 80, page 266.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) falls below the lower limit (Lower Limit 0x6425).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_LOWER_LIMIT	[MxSubExt6425] ParameterName=AI_LOWER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=2
Description	Analog Input Interrupt Lower Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.21.7 Analog Input Interrupt Delta Unsigned (Object 0x6426)

Object 0x6426 defines the fluctuation range (delta) relative to the last transmitted value both when counting up and down and affects the transmission of object 0x6401 accordingly → Figure 80, page 266.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The difference between the current value (ENTRY into 0x6401) and the last transmitted value is equal to or greater than the fluctuation range (Value Difference 0x6426).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_DELTA_VALUE	[MxSubExt6426] ParameterName=AI_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=2
Description	Analog Input Interrupt Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01... FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

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

Sub index 1≤n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.21.8 Analog Input Interrupt Negative Delta Unsigned (Object 0x6427)

Object 0x6427 defines the negative fluctuation range (delta) relative to the last transmitted value when counting down and affects the transmission of object 0x6401 accordingly → Figure 80, page 266.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is less than the last transmitted value and the difference is greater than or equal to the fluctuation range (Negative Value Difference 0x6427).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_NEGATIVE_DELTA_VALUE	[MxSubExt6427] Parameter-Name=AI_NEGATIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=2
Description	Analog Input Interrupt Negative Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE $_{ex}$	
Access	rw	
Default value	00000000$_{hex}$	

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.21.9 Analog Input Interrupt Positive Delta Unsigned (Object 0x6428)

Object 0x6428 defines the negative fluctuation range (delta) relative to the last transmitted value when counting up and affects the transmission of object 0x6401 accordingly → Figure 80, page 266.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is greater than the last transmitted value. The difference's magnitude is greater than the fluctuation range (Positive Value Difference 0x6428).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

If the current value is less than the last transmitted value, object 0x6401 will be transmitted. If this behavior is not desired, object 0x6421 must be enabled and disabled accordingly.

Feature	Description / Value	EDS
Name	AI_POSITIVE_DELTA_VALUE	[MxSubExt6428] ParameterName=AI_POSITIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=2
Description	Analog Input Interrupt Positive Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≤n≤254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															LSB

7.21.10 Output Channel x (Object 0x21B0 to 0x21B1)

Objects 0x21B0 to 0x21B1 transmit the digital value of the channels' analog signal outputs.

Feature	Description / Value		EDS
Name	OutputChannelx		[MxSubExt21Bx] ParameterName=OutputChannelx ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=1 Count=1
Description	OutputChannel1	Object 0x21B0	
	OutputChannel2	Object 0x21B1	
Object Code	ARRAY		
Mapping	PDO		
	MANUAL		
Data type	INTEGER16		
Sub index	01 … FE hex		
Access	rww		
Default value	0 x 0000 hex		

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.21.11 Module Diagnostic Messages (Object 0x31B0)

Object 0x31B0 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	ModuleDiag	[MxSubExt31B0] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Module Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–					

Data bit	Designation	Description
0		reserved
1		0 = sync OK 1 = no sync
2		0 = OK 1 = FLASH Data CRC Error
3		0 = OK 1 =RAM Data CRC Error
4		0 = OK 1 = Inconsistent FLASH Data
5		0 = OK 1 = Invalid configuration
6 – 7		reserved

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Description
8 – 15		reserved

7.21.12 Input Channel x (Object 0x31B1 to 0x31B2)

Objects 0x31B1 to 0x31B2 represent the formatted (integer) analog input values.

Feature	Description / Value		EDS
Name	InputChannel1	Object 0x31B1	[MxSubExt31Bx] ParameterName=InputChannelx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
	InputChannel2	Object 0x31B2	
Description	InputChannelx		
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.21.13 Channel Diagnostic Messages (Object 0x31B3))

Object 0x31B3 contains status information on the module's channels.

Feature	Description / Value	EDS
Name	ChannelDiag	[MxSubExt31B3] ParameterName=ChannelDiag ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Channel Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Description
0		0 = Input channel 1 OK 1 = Input channel 1 Wire breakage
1		0 = Input channel 2 OK 1 = Input channel 2 Wire breakage
2		reserved
3		reserved
4		0 = OK 1 = Input channel 1 Over Range
5		0 = OK 1 = Input channel 2 Over Range
6 – 7		reserved

7.21.14 Input Voltage State (Object 0x31B4)

Object 0x31B4 contains status information on the supply voltage for the module's two inputs and outputs.

Feature	Description / Value	EDS
Name	InputVoltageState	[MxSubExt31B4] ParameterName=InputVoltageState ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Status of supply voltage	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

Data bit	Designation	Description
0-6	–	reserved
7	Power supply of analog inputs 1+, 2+ analog outputs 1+, 2+	0 = No power 1 = Power supply 24 VDC OK

7.21.15 Firmware Version (Object 0x41B0)

Object 0x41B0 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	FirmwareVersion	[MxSubExt41B0] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	Firmware Version	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

7.21.16 Input Channel Configuration (Object 0x51B0)

Object 0x51B0 must be used to configure the channels 0…20 mA or 4…20 mA measuring method.

Feature	Description / Value	EDS
Name	InputChannelConfig	[MxSubExt51B0 ParameterName=InputChannelConfig ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 DefaultValue=0xF0 Count=1
Description	Input Channel Configuration	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	$F0_{hex}$	

Design of the data bytes:

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

Byte 0:

Data bit Bit 1	Bit 0	Designation	Meaning
0	0	Channel 1	Measurement range 0…20mA
0	1		Measurement range 4…20mA
1	0		–
1	1		Input disabled

Data bit Bit 3	Bit 2	Designation	Description
0	0	Channel 2	Measurement range 0…20mA
0	1		Measurement range 4…20mA
1	0		–
1	1		Input disabled

Data bit	Designation	Description
4 – 7		reserved

7.21.17 Output Channel Configuration (Object 0x51B1)

Object 0x51B1 can be used to configure the output channels. It is used to define which current should correspond to the current output value based on the selected output range.

Feature	Description / Value	EDS
Name	OutputChannelConfig	[MxSubExt51B1] ParameterName=OutputChannelConfig ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 DefaultValue=0xF0 Count=1
Description	Output Channel Configuration	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	$F0_{hex}$	

Design of the data bytes:

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

Byte 0:

Data bit Bit 1	Bit 0	Designation	Description
0	0	Channel 1	Output range 0…20mA
0	1		–
1	0		–
1	1		Output disabled

Data bit Bit 3	Bit 2	Designation	Description
0	0	Channel 2	Output range 0…20mA
0	1		–
1	0		–
1	1		Output disabled

Data bit	Designation	Description
4 – 7		reserved

7.21.18 Filter Configuration Channel x (Object 0x51B2 to 0x51B3)

Objects 0x51B2 to 0x51B3 can be used to configure the software filter for a channel.

Feature	Description / Value		EDS
Name	FilterConfigChannelx		[MxSubExt51Bx] ParameterName=FilterConfigChannelx ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
Description	FilterConfigChannel1	Object 0x51B2	
	FilterConfigChannel2	Object 0x51B3	
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	rw		

Design of the data bytes:

Low-pass cut-off frequency in Hz (example: 50 Hz => 0032_{hex})

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

Example

Low-pass cut-off frequency in Hz

50 Hz = 0032_{hex}

The following settings are valid:

Low-pass cut-off frequency	Register value
Filter disabled (default)	0x0000
10 Hz	0x000A
25 Hz	0x0019
50 Hz	0 x0032
100 Hz	0x0064
250 Hz	0x00FA
500 Hz	0x01F4
1000 Hz	0x03E8

7.22 XN-322-8AIO-I

This module supports the provision of data for analog inputs and analog outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

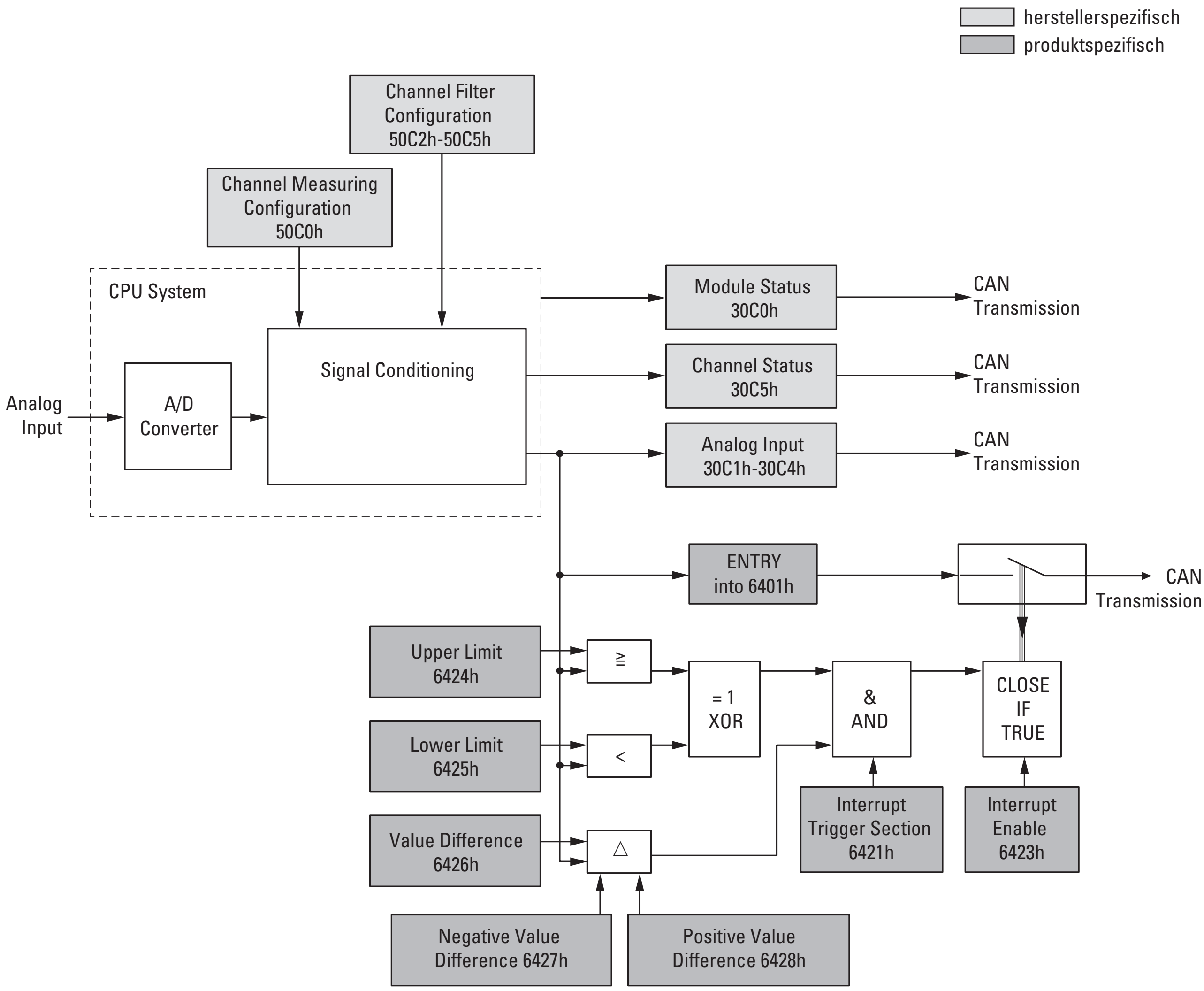


Figure 82: Block diagram showing the various CANopen objects for analog inputs

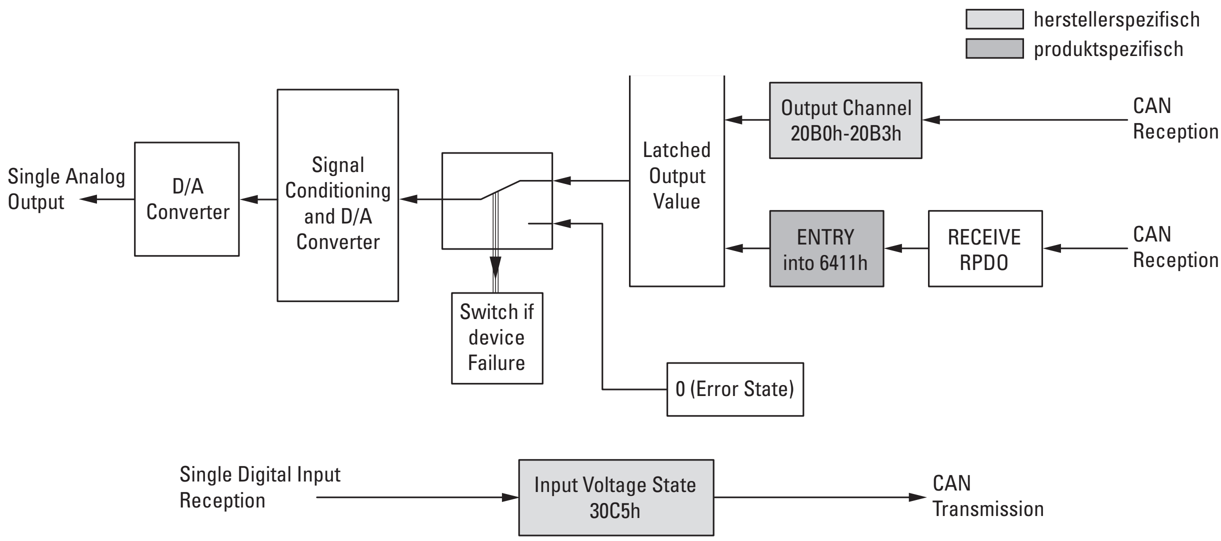


Figure 83: Block diagram showing the various CANopen objects for analog outputs

Product-specific CANopen objects

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x6401	INTEGER16	I-WORD	Read Analog Input 16-bit	Default	ro	PDO
0x6411	INTEGER16	Q-WORD	Write Analog Output 16-bit	Default	rww	PDO
0x6421	UNSIGNED8	AI_INTERRUPT_TRIGGER_SELECTION	Analog Input Interrupt Trigger Selection	–	rw	SDO
0x6423	BOOLEAN	AnalogInputGlobalInterruptEnable	Analog Input Global Interrupt Enable	–	rw	SDO
0x6424	INTEGER32	AI_UPPER_LIMIT	Analog Input Interrupt Upper Limit Integer	–	rw	SDO
0x6425	INTEGER32	AI_LOWER_LIMIT	Analog Input Interrupt Lower Limit Integer	–	rw	SDO
0x6426	UNSIGNED32	AI_DELTA_VALUE	Analog Input Interrupt Delta Unsigned	–	rw	SDO
0x6427	UNSIGNED32	AI_NEGATIVE_DELTA_VALUE	Analog Input Interrupt Negative Delta Unsigned	–	rw	SDO
0x6428	UNSIGNED32	AI_POSITIVE_DELTA_VALUE	Analog Input Interrupt Positive Delta Unsigned	–	rw	SDO

Manufacturer-specific objects

Index range for the XN-322-8AIO-I: x0C0 to x0CF

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x20C0	INTEGER16	OutputChannel1	Output Channel 1	Manual	rww	PDO
0x20C1	INTEGER16	OutputChannel2	Output Channel 2	Manual	rww	PDO
0x20C2	INTEGER16	OutputChannel3	Output Channel 3	Manual	rww	PDO
0x20C3	INTEGER16	OutputChannel4	Output Channel 4	Manual	rww	PDO

0x30C0	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x30C1	INTEGER16	InputChannel1	Input Channel 1	Manual	ro	PDO
0x30C2	INTEGER16	InputChannel2	Input Channel 2	Manual	ro	PDO
0x30C3	INTEGER16	InputChannel3	Input Channel 3	Manual	ro	PDO
0x30C4	INTEGER16	InputChannel4	Input Channel 4	Manual	ro	PDO
0x30C5	UNSIGNED8	ChannelDiag	Channel Diagnostic Messages	Manual	ro	PDO
0x30C6	UNSIGNED8	InputVoltageState	Input Voltage State Bit 0: DC 24V Output 1..8 OK Bit 1: DC 24V Output 9..16 OK	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO
0x40C0	UNSIGNED16	FirmwareVersion	Firmware Version	–	ro	SDO
0x50C0	UNSIGNED8	InputChannelConfig	Channel Measuring Configuration (Measurement range 0…20mA/4…20mA)	–	rw	SDO
0x50C1	UNSIGNED8	OutputChannelConfig	Channel Output Configuration (Measurement range 0…20mA)	–	rw	SDO
0x50C2	UNSIGNED16	FilterConfigChannel1	Filter Configuration Channel 1	–	rw	SDO
0x50C3	UNSIGNED16	FilterConfigChannel2	Filter Configuration Channel 2	–	rw	SDO
0x50C3	UNSIGNED16	FilterConfigChannel3	Filter Configuration Channel 3	–	rw	SDO
0x50C4	UNSIGNED16	FilterConfigChannel4	Filter Configuration Channel 4	–	rw	SDO

7.22.1 Read Analog Input 16-Bit (Object 0x6401)

Object 0x6401 represents the channels' formatted analog input values. The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Name	I-WORD	[MxSubExt6401] ParameterName=I-WORD ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=4
Description	Read Analog Input 16-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

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

7.22.2 Write Analog Output 16-Bit (Object 0x6411)

Object 0x6411 transmits the digital value of the channels' analog signal outputs. The object's data is automatically entered into the receive PDOs (default mapping).

Feature	Description / Value	EDS
Name	Q-WORD	[MxSubExt6411] ParameterName=Q-WORD ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=1 Count=4
Description	Write Analog Output 16-Bit	
Object Code	ARRAY	
Mapping	PDO	
	Default	
Data type	INTEGER16	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

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

7.22.3 Analog Input Interrupt Trigger Selection (Object 0x6421)

Object 0x6421 defines which events will trigger an interrupt for the corresponding channel so that the channel's analog input value (object 0x6401) will be transmitted when the event occurs.

Feature	Description / Value	EDS
Name	AI_INTERRUPT_TRIGGER_SELECTION	[MxSubExt6421] Parameter-Name=AI_INTERRUPT_TRIGGER_SELECTIO N ObjectType=0x7 DataType=0x0005 AccessType=rw DefaultValue=7 LowLimit=0 HighLimit=31 PDOMapping=0 Count=4
Description	Analog Input Interrupt Trigger Selection	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	07_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Description
0		0 = Upper limit not exceeded 1 = Upper limit exceeded
1		0 = Input not below lower limit 1 = Input below lower limit
2		0 = Input not changed by more than delta 1 = Input changed by more than delta
3		0 = Input not reduced by more than negative delta 1 = Input reduced by more than negative d
4		0 = Input not reduced by more than positive delta 1 = Input reduced by more than positive delta
5 – 7		reserved

7.22.4 Analog Input Global Interrupt Enable (Object 0x6423)

Object 0x6423 enables and disables the transmission of data via object 0x6401 in general.

Feature	Description / Value	EDS
Name	AnalogInputGlobalInterruptEnable	[MxFixed6423] ParameterName=AnalogInputGlobalInterruptEnable ObjectType=0x7 DataType=0x0001 AccessType=rw DefaultValue=0 PDOMapping=0
Description	Analog Input Global Interrupt Enable	
Object Code	Variable	
Mapping	SDO	
Data type	BOOLEAN	
Access	rw	
Default value	FALSE	
Object Code	Variable	

Default value

- FALSE (0)
  Object 0x6401 is not enabled and cannot transmit any analog input values.
- TRUE (1)
  Object 0x6401 is enabled and can transmit analog input values.

7.22.5 Analog Input Interrupt Upper Limit Integer (Object 0x6424)

Object 0x6424 defines an upper limit and affects the transmission of object 0x6401 accordingly → Figure 82, page 284.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) is equal to or greater than the upper limit (Upper Limit 0x6424).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_UPPER_LIMIT	[MxSubExt6424] ParameterName=AI_UPPER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=4
Description	Analog Input Interrupt Upper Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	$01 \dots FE_{hex}$	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.22.6 Analog Input Interrupt Lower Limit Integer (Object 0x6425)

Object 0x6425 defines a lower limit and affects the transmission of object 0x6401 accordingly → Figure 82, page 284.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The analog input value (ENTRY into 0x6401) falls below the lower limit (Lower Limit 0x6425).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_LOWER_LIMIT	AI_LOWER_LIMIT[MxSubExt6425] ParameterName=AI_LOWER_LIMIT ObjectType=0x7 DataType=0x0004 AccessType=rw DefaultValue=0x00000000 PDOMapping=0 Count=4
Description	Analog Input Interrupt Lower Limit Integer	
Object Code	ARRAY	
Mapping	SDO	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000 0000_{hex}	

Design of the data bytes:

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

Sub index 1≤n≤254	4 Byte								3 Byte							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.22.7 Analog Input Interrupt Delta Unsigned (Object 0x6426)

Object 0x6426 defines the fluctuation range (delta) relative to the last transmitted value both when counting up and down and affects the transmission of object 0x6401 accordingly → Figure 82, page 284.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The difference between the current value (ENTRY into 0x6401) and the last transmitted value is equal to or greater than the fluctuation range (Value Difference 0x6426).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_DELTA_VALUE	[MxSubExt6426] ParameterName=AI_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=4
Description	Analog Input Interrupt Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	$01... FE_{hex}$	
Access	rw	
Default value	$0000\ 0000_{hex}$	

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.22.8 Analog Input Interrupt Negative Delta Unsigned (Object 0x6427)

Object 0x6427 defines the negative fluctuation range (delta) relative to the last transmitted value when counting down and affects the transmission of object 0x6401 accordingly → Figure 82, page 284.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is less than the last transmitted value and the difference is greater than or equal to the fluctuation range (Negative Value Difference 0x6427).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

Feature	Description / Value	EDS
Name	AI_NEGATIVE_DELTA_VALUE	[MxSubExt6427] ParameterName=AI_NEGATIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=4
Description	Analog Input Interrupt Negative Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE $_{ex}$	
Access	rw	
Default value	00000000$_{hex}$	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	Not relevant								Not relevant							

7.22.9 Analog Input Interrupt Positive Delta Unsigned (Object 0x6428)

Object 0x6428 defines the negative fluctuation range (delta) relative to the last transmitted value when counting up and affects the transmission of object 0x6401 accordingly → Figure 82, page 284.

The specified fluctuation range refers to the data transmitted in object 0x6401 and accordingly must be specified as an unsigned value with the same value representation.

Object 0x6401 will be transmitted (CAN transmission) if all of the following conditions are met:

- The current value (ENTRY into 0x6401) is greater than the last transmitted value. The difference's magnitude is greater than the fluctuation range (Positive Value Difference 0x6428).
- Object 0x6421 is set to enable.
- The global interrupt in object 0x6423 has been enabled with: TRUE.

If the current value is less than the last transmitted value, object 0x6401 will be transmitted. If this behavior is not desired, object 0x6421 must be enabled and disabled accordingly.

Feature	Description / Value	EDS
Name	AI_POSITIVE_DELTA_VALUE	[MxSubExt6428] ParameterName=AI_POSITIVE_DELTA_VALUE ObjectType=0x7 DataType=0x0007 AccessType=rw DefaultValue=0x00000000 HighLimit=0x0000FFFF PDOMapping=0 Count=4
Description	Analog Input Interrupt Positive Delta Unsigned	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	00000000_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															LSB

7.22.10 Output Channel x (Object 0x20C0 bis 0x20C3)

Objects 0x20C0 to 0x20C3 transmit the digital value of the channels' analog signal outputs.

Feature	Description / Value		EDS
Name	OutputChannel1	Object 0x20C0	[MxSubExt20Cx] ParameterName=OutputChannelx ObjectType=0x7 DataType=0x0003 AccessType=rww PDOMapping=1 Count=1
	OutputChannel2	Object 0x20C1	
	OutputChannel3	Object 0x20C2	
	OutputChannel4	Object 0x20C3	
Description	OutputChannelx		
Object Code	ARRAY		
Mapping	PDO		
	MANUAL		
Data type	INTEGER16		
Sub index	01 … FE hex		
Access	rww		

Design of the data bytes:

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

7.22.11 Module Diagnostic Messages (Object 0x30C0)

Object 0x30C0 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	ModuleDiag	[MxSubExt30C0] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Module Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–					

Data bit	Designation	Meaning
0		reserved
1		0 = sync OK 1 = no sync
2		0 = OK 1 = FLASH Data CRC Error
3		0 = OK 1 =RAM Data CRC Error
4		0 = OK 1 = Inconsistent FLASH Data
5		0 = OK 1 = Invalid configuration
6 – 7		reserved

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Designation	Description
8 – 15		reserved

7.22.12 Input Channel x (Object 0x30C1 bis 0x30C4)

Objects 0x31B1 to 0x31B2 represent the formatted (integer) analog input values.

Feature	Description / Value		EDS
Name	Input Channel x		[MxSubExt30Cx] ParameterName=InputChannelx ObjectType=0x7 DataType=0x0003 AccessType=ro PDOMapping=1 Count=1
Description	InputChannel1	Object 0x30C1	
	InputChannel2	Object 0x30C2	
	InputChannel3	Object 0x30C3	
	InputChannel4	Object 0x30C4	
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

7.22.13 Channel Diagnostic Messages (Object 0x30C5))

Object 0x30C5 contains status information on the module's channels.

Feature	Description / Value	EDS
Name	ChannelDiag	[MxSubExt30C5] ParameterName=ChannelDiag ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Channel Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
0		0 = Input channel 1 OK 1 = Input channel 1 wire breakage
1		0 = Input channel 2 OK 1 = Input channel 2 wire breakage
2		0 = Input channel 3 OK 1 = Input channel 3 wire breakage
3		0 = Input channel 4 OK 1 = Input channel 4 wire breakage
4		0 = OK 1 = Input channel 1 Over Range
5		0 = OK 1 = Input channel 2 Over Range
6		0 = OK 1 = Input channel 3 Over Range
7		0 = OK 1 = Input channel 4 Over Range

7.22.14 Input Voltage State (Object 0x30C6)

Object 0x30C6 contains status information on the module's supply voltage:

Feature	Description / Value	EDS
Name	InputVoltageState	[MxSubExt30C6] ParameterName=InputVoltageState ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Status of supply voltage	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	00_{hex}	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

Data bit	Designation	Description	Notes
0-6	–	reserved	
7	Power supply of analog inputs 1+, 2+, 3+, 4+ analog outputs 1+, 2+, 3+, 4+	0 = No power 1 = Power supply 24 VDC OK	

7.22.15 Firmware Version (Object 0x40C0)

Object 0x40C0 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	FirmwareVersion	[MxSubExt40C0] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	Firmware Version	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.22.16 Input Channel Configuration (Object 0x50C0)

Object 0x50C0 can be used to set the channels' measuring method to 0–20 mA or 4–20 mA.

Feature	Description / Value	EDS
Name	InputChannelConfig	[MxSubExt50C0 ParameterName=InputChannelConfig ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	Input Channel Configuration	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Byte 0:

Data bit Bit 1	Bit 0	Designation	Description
0	0	Channel 1	Measurement range 0…20mA
0	1		Measurement range 4…20mA
1	0		–
1	1		Input disabled

Data bit Bit 3	Bit 2	Designation	Description
0	0	Channel 2	Measurement range 0…20mA
0	1		Measurement range 4…20mA
1	0		–
1	1		Input disabled

Data bit Bit 5	Bit 4	Designation	Description
0	0	Channel 3	Measurement range 0…20mA
0	1		Measurement range 4…20mA
1	0		–
1	1		Input disabled

Data bit Bit 7	Bit 6	Designation	Description
0	0	Channel 4	Measurement range 0…20mA
0	1		Measurement range 4…20mA
1	0		–
1	1		Input disabled

7.22.17 Output Channel Configuration (Object 0x50C1)

Object 0x50C1 can be used to configure the output channels. It is used to define which current should correspond to the current output value based on the selected output range.

Feature	Description / Value	EDS
Name	OutputChannelConfig	[MxSubExt50C1] ParameterName=OutputChannelConfig ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	Output Channel Configuration	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000_{hex}	

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Byte 0:

Data bit Bit 1	Bit 0	Designation	Meaning
0	0	Channel 1	Output range 0…20mA
0	1		–
1	0		–
1	1		Output disabled

Data bit Bit 3	Bit 2	Designation	Description
0	0	Channel 2	Output range 0…20mA
0	1		–
1	0		–
1	1		Output disabled

Data bit Bit 5	 Bit 4	Designation	Description
0	0	Channel 3	Output range 0…20mA
0	1		–
1	0		–
1	1		Output disabled

Data bit Bit 7	 Bit 6	Designation	Description
0	0	Channel 4	Output range 0…20mA
0	1		–
1	0		–
1	1		Output disabled

7.22.18 Filter Configuration Channel x (Object 0x50C2 bis 0x50C5)

Objects 0x50C2 to 0x50C5 can be used to configure the software filter for a channel.

Feature	Description / Value		EDS
Name	FilterConfigChannelx		[MxSubExt50Cx] ParameterName=FilterConfigChannelx ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0 Count=1
Description	FilterConfigChannel1	Object 0x50C2	
	FilterConfigChannel2	Object 0x50C3	
	FilterConfigChannel3	Object 0x50C4	
	FilterConfigChannel4	Object 0x50C5	
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	rw		

Design of the data bytes:

Low-pass cut-off frequency in Hz (example: 50 Hz => 0032_{hex})

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

Example

Low-pass cut-off frequency in Hz

50 Hz = 0032_{hex}

The following settings are valid:

Low-pass cut-off frequency	Register value
Filter disabled (default)	0x0000
10 Hz	0x000A
25 Hz	0x0019
50 Hz	0x0032
100 Hz	0x0064
250 Hz	0x00FA
500 Hz	0x01F4
1000 Hz	0x03E8

7.23 XN-322-2DMS-WM

This module supports the provision of data for analog inputs as per the specifications in CiA401. The behavior of the I/O slice modules can be configured with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

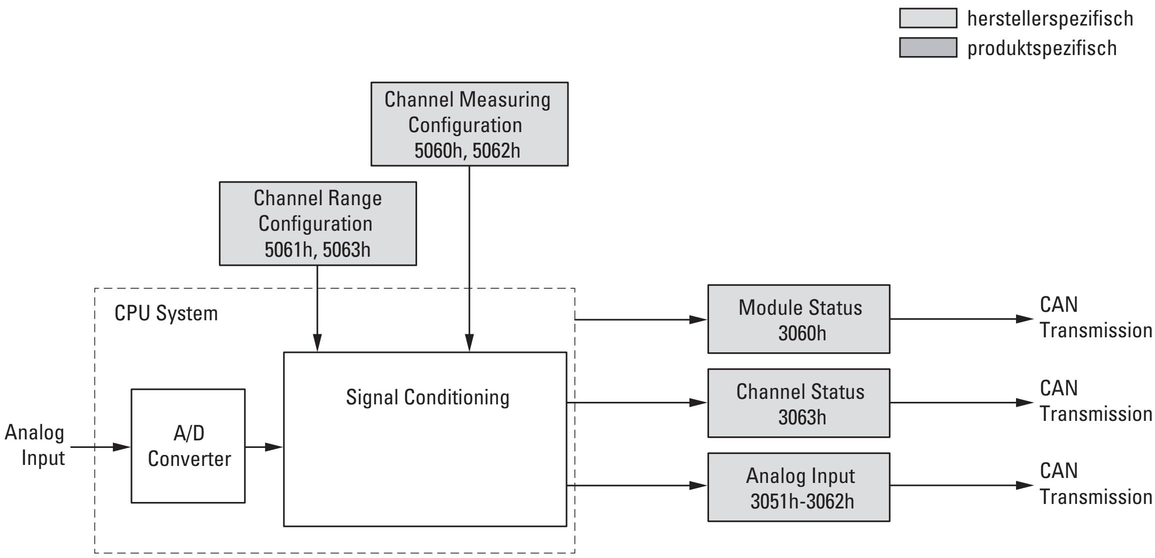


Figure 84: Block diagram showing the various CANopen objects

Product-specific CANopen objects

- None -

Manufacturer-specific objects

Index range for the XN-322-2DMS-WM module: x060 to x06F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x3060	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x3061	INTEGER32	InputChannel1	Input Channel 1	Manual	ro	PDO
0x3062	INTEGER32	InputChannel2	Input Channel 2	Manual	ro	PDO
0x3063	UNSIGNED16	ADCDiag	Analog Digital Converter Diagnostic Messages	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO
0x4060	UNSIGNED16	FirmwareVersion	Firmware Version	–	ro	SDO
0x5060	UNSIGNED16	MeasuringConfig Channel1	Measuring Configuration Channel 1	–	rw	SDO
0x5061	UNSIGNED16	RangeConfig Channel1	Range Configuration Channel 1	–	rw	SDO
0x5062	UNSIGNED16	MeasuringConfig Channel2	Measuring Configuration Channel 2	–	rw	SDO
0x5063	UNSIGNED16	RangeConfig Channel2	Range Configuration Channel 2	–	rw	SDO
0x5064	INTEGER32	ZeroScale Channel1	Zero-Scale Channel 1	–	ro	SDO
0x5065	INTEGER32	FullScale Channel1	Full-Scale Channel 1	–	ro	SDO
0x5066	INTEGER32	ZeroScale Channel2	Zero-Scale Channel 2	–	ro	SDO
0x5067	INTEGER32	FullScale Channel2	Full-Scale Channel 2	–	ro	SDO

7.23.1 Module Diagnostic Messages (Object 0x3060)

Object 0x3060 contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	ModuleDiag	[MxSubExt3060] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Module Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–						

Data bit	Desig-nation	Description
0		reserved
1		0 = sync OK 1 = Ino sync
2		0 = OK 1 = FLASH Data CRC Error
3		0 = OK 1 =RAM Data CRC Error
4		0 = OK 1 = Inconsistent FLASH Data
5		0 = OK 1 = Bridge 1 DC not OK
6		0 = OK 1 = Bridge 2DC not OK
7		0 = OK 1 = Offset ADC1 not valid

Byte 1:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	CLASS	CLASS	CLASS	TYPE	TYPE	TYPE

Data bit	Desig-nation	Description
0		0 = OK 1 = Offset ADC2 not valid
1		0 = OK 1 = Filter ADC1 not ready
2		0 = OK 1 = Filter ADC2 not ready
3 – 7		reserved

7.23.2 Input Channel x (Object 0x3061 to 0x3062)

Objects 0x3061 to 0x3062 represent the analog input values

Feature	Description / Value		EDS
Name	Channel1Input	Object 0x3061	[MxSubExt306x] ParameterName=ChannelxInput ObjectType=0x7 DataType=0x0004 AccessType=ro PDOMapping=1 Count=1
	Channel2Input	Object 0x3062	
Description	Input Channel x		
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	INTEGER32		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

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

	4 Byte								**3 Byte**							
Sub index 1≦n≦254	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															

7.23.3 Analog Digital Converter Diagnostic Messages (Object 0x3063)

Object 0x3063 contains status information on the device's analog-to-digital converter.

Feature	Description / Value	EDS
Name	ADCDiag	[MxSubExt3063] ParameterName=ADCDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Analog Digital Converter Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB
	Channel 2 (ADC2)								Channel 1(ADC1)							

Byte 0:

Data bit	Designation	Description
0 – 4		reserved
5		0 = OK 1 = Reference Voltage Error
6		0 = OK 1 = ADC Range Error
7		0 = measurement is off 1 = measurement is active

Byte 1:

Data bit	Designation	Description
0 – 4		reserved
5		0 = OK 1 = Reference Voltage Error
6		0 = OK 1 = ADC Range Error
7		0 = measurement is off 1 = measurement is active

7.23.4 Firmware Version (Object 0x4060)

Object 0x4060 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	FirmwareVersion	[MxSubExt4060] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	Firmware Version	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≤n≤254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.23.5 Measuring Configuration Channel x (Object 0x5060, 0x5062)

Objects 0x5060 and 0x5062 can be used to configure special measuring settings for a channel.

Feature	Description / Value		EDS
Name	MeasuringConfigChannel1	Object 0x5060	[MxSubExt506x] ParameterName=MeasuringConfigChannelx ObjectType=0x7 DataType=0x0006 AccessType=rw DefaultValue=2 PDOMapping=0 Count=1
	MeasuringConfigChannel2	Object 0x5062	
Description	MeasuringConfig Channelx		
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	rw		
Default value	0002_{hex}		

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Byte 0:

Data bit	Desig-nation	Description
0 – 7		Filter depth of ADC 1–1023 (Default = 1)

Byte 1:

Data bit	Desig-nation	Description
0 – 1		Filter depth of ADC 1–1023 (Default = 1)
2		0 ≧ SINC4 Filter (Default) 1 ≧ SINC3 Filter
3 – 5		0 = Continuous conversion mode (default) 1 – 5 = reserved 6 = System zero-scale calibration 7 = System full-scale calibration
6 – 7		reserved

7.23.6 Range Configuration Channel x (Object 0x5061, 0x5063)

Objects 0x5061 and 0x5063 can be used to configure the range or input gain for a channel.

Feature	Description / Value		EDS
Name	RangeConfig Channelx		[MxSubExt506x] ParameterName=RangeConfigChannelx ObjectType=0x7 DataType=0x0006 AccessType=rw DefaultValue=3 PDOMapping=0 Count=1
Description	RangeConfigChannel1	Object 0x5061	
	RangeConfigChannel2	Object 0x5063	
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	rw		
Default value	0003_{hex}		

Design of the data bytes:

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

Byte 0:

Data bit	Desig-nation	Description
0 – 2		0 = GAIN 1 (+/- 120mV) 1, 2 ≧ reserved (don't use) 3 = GAIN 8 (+/- 15mV) 4 = GAIN 16 (+/- 7.5mV) 5 = GAIN 32 (+/- 3.75mV) 6 = GAIN 64 (+/- 1.875mV) 7 => reserved (don't use)
3 – 7		reserved

Byte 1:

Data bit	Desig-nation	Description
0 – 7		reserved

7.23.7 Zero-Scale Channel x (Object 0x5064, 0x5066)

Objects 0x5064 and 0x5066 store the value of the measurement from the zero-scale calibration.

Feature	Description / Value		EDS
Name	Zero-Scale Channel x		[MxSubExt506x] ParameterName=ZeroScaleChannelx ObjectType=0x7 DataType=0x0004 AccessType=ro DefaultValue=0 PDOMapping=0 Count=1
Description	ZeroScaleChannel1	Object 0x5064	
	ZeroScaleChannel2	Object 0x5066	
Object Code	ARRAY		
Mapping	SDO		
Data type	INTEGER32		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≤n≦254	3 Byte								Byte 2							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.23.8 Full-Scale Channel x (Object 0x5065, 0x5067)

Objects 0x5065 and 0x5067 store the value of the measurement from the full-scale calibration.

Feature	Description / Value		EDS
Name	Full-Scale Channel x		[MxSubExt506x] ParameterName=FullScaleChannelx ObjectType=0x7 DataType=0x0004 AccessType=ro DefaultValue=0 PDOMapping=0 Count=1
Description	FullScaleChannel1	Object 0x5065	
	FullScaleChannel2	Object 0x5067	
Object Code	ARRAY		
Mapping	SDO		
Data type	INTEGER32		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

Sub index 1≤n≦254	3 Byte								Byte 2							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.24 XN-322-1DCD-B35

This module supports the provision of data per the specifications in CiA401. The behavior of the I/O slice modules can be configured with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

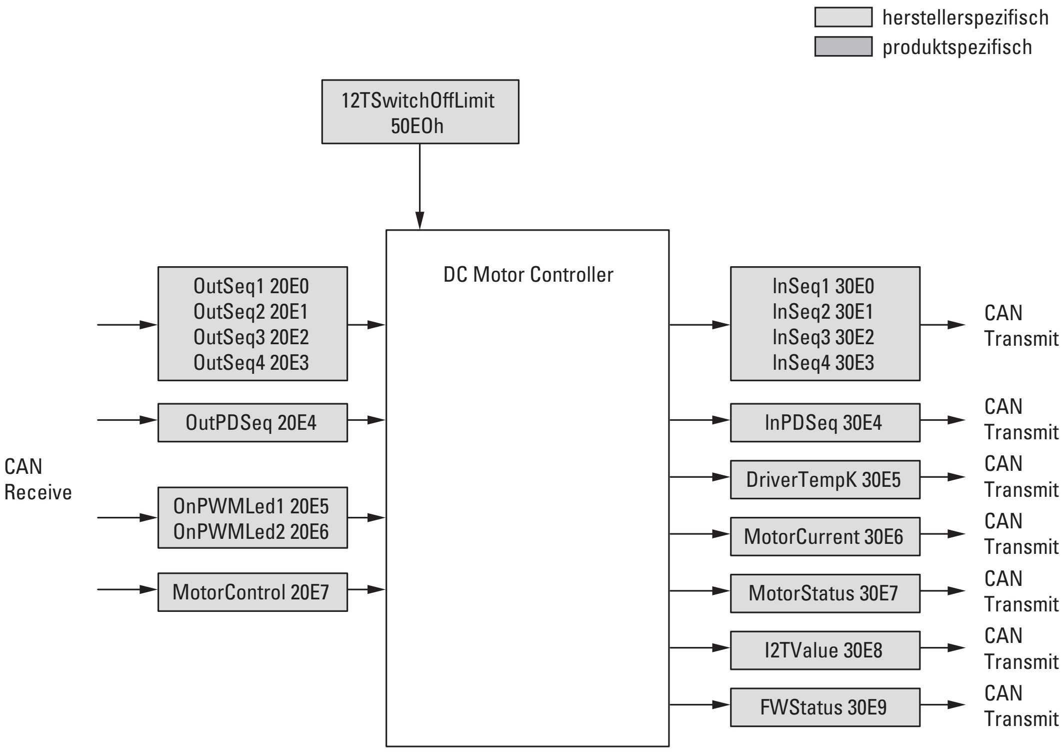


Figure 85: Block diagram showing the various CANopen objects

Manufacturer-specific objects

Index range for the XN-322-2DCD-B35 module: x0E0 to x0EF

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number; → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x20E0	UNSIGNED16	WRSeq1	Write PWM Sequence Data Seq. 1	Manual	ro	PDO
0x20E1	UNSIGNED16	WRSeq2	Write PWM Sequence Data Seq. 2	Manual	ro	PDO
0x20E2	UNSIGNED16	WRSeq3	Write PWM Sequence Data Seq. 3	Manual	ro	PDO
0x20E3	UNSIGNED16	WRSeq4	Write PWM Sequence Data Seq. 4	Manual	ro	PDO
0x20E4	UNSIGNED16	WRPeriodDurationSeq	Write Period Duration of Sequence Cycle	Manual	ro	PDO
0x20E5	UNSIGNED8	TonLED1	ON Time PWM LED 1 (20mA)	Manual	wo	PDO
0x20E6	UNSIGNED8	TonLED2	ON Time PWM LED 2 (350mA)	Manual	wo	PDO
0x20E7	UNSIGNED16	MotorControl	Motor Control Register	Manual	wo	PDO
0x30E0	UNSIGNED16	RDSeq1	Read PWM Sequence Data Seq. 1	Manual	ro	PDO
0x30E1	UNSIGNED16	RDSeq2	Read PWM Sequence Data Seq. 2	Manual	ro	PDO
0x30E2	UNSIGNED16	RDSeq3	Read PWM Sequence Data Seq. 3	Manual	ro	PDO
0x30E3	UNSIGNED16	RDSeq4	Read PWM Sequence Data Seq. 4	Manual	ro	PDO
0x30E4	UNSIGNED16	RDPeriodDurationSeq	Read Period Duration of Sequence Cycle	Manual	ro	PDO
0x30E5	UNSIGNED16	DCDTempK	DC Driver Temperature in °K	Manual	ro	PDO
0x30E6	UNSIGNED16	DCMotorCurrent	DC Motor Current in mA	Manual	ro	PDO
0x30E7	UNSIGNED16	DCMotorDiag	DC Motor Diagnosis	Manual	ro	PDO
0x30E8	UNSIGNED16	DCMotorStatus	DC Motor Status	Manual	ro	PDO
0x30E9	UNSIGNED32	DCMotorI2T	DC Motor I^2T Value	Manual	ro	PDO
0x30EA	UNSIGNED16	ModuleDiag	Module Diagnostic Messages	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product name → Section "6.2.14 Product Name (Object 0x400C)", page 82	-	ro	SDO
0x40E0	UNSIGNED16	FirmwareVersion	Firmware Version	–	ro	SDO
0x40E1	UNSIGNED16	PreScaleLED1	PWM Prescaler Register LED1	–	ro	SDO
0x40E2	UNSIGNED16	PreScaleLED2	PWM Prescaler Register LED2	–	ro	SDO

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x40E3	UNSIGNED8	PDLED1	PWM Period Duration Register LED1	–	ro	SDO
0x40E4	UNSIGNED8	PDLED2	PWM Period Duration Register LED2	–	ro	SDO
0x40E5	UNSIGNED32	DCMotorI2TLimit	DC Motor I2T Value Limit	–	rw	SDO

7.24.1 Write PWM Sequenz Data (Object 0x20E0 to 0x20E3)

ACHTUNG NOTICE

Select period duration t_p in line with the output sequence:

$t_p = t_{OutputSequence} = t_{OutputSequence1} + \ldots + t_{OutputSequence4}$

Failure to do so may result in undesired XN300 slice module states.

The four objects 0x20E0 to 0x20E3 are used to transmit the PWM output sequence for motor control to the XN300 module.

Feature	Description / Value		EDS
Name	WRSeq1	Object 0x20E0	[MxSubExt20Ex] ParameterName=WRSeq1 ObjectType=0x7 DataType=0x0006 AccessType=wo PDOMapping=1 Count=1
	WRSeq2	Object 0x20E1	
	WRSeq3	Object 0x20E2	
	WRSeq4	Object 0x20E3	
Description	Write PWM Sequence Data		
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	wo		
Default value	0000_{hex}		

Design of the data bytes:

Sub index 1≦n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	Motor rotation				TC	Time value for determining the pulse width $t_{OutputSequence\ n}$										

Data bit	Designation	Meaning
0 – 10	$t_{OutputSequence\ n}$	Time value for determining the pulse width
11	TC	0 = relative time counter 1 = absolute time counter
12		0 = OFF 1 = left high ON
13		0 = OFF 1 = right high ON
14		0 = OFF 1 = left low ON
15		0 = OFF 1 = right low ON

7.24.2 Write Period Duration of Sequence Cycle (Object 0x20E4)

ACHTUNG NOTICE

Select period duration t_p in line with the output sequence:
$t_p = t_{OutputSequence} = t_{OutputSequence1} + ... + t_{OutputSequence4}$
Failure to do so may result in undesired XN300 slice module states.

Object 0x20E4 can be used to transmit the duration of the PWM output sequence t_p for motor control to the module.

$$\text{Period duration} = t_P = \frac{\text{WRPeriodDurationSeq}}{\text{Clock frequency}}$$

Feature	Description / Value	EDS
Name	WRPeriodDurationSeq	[MxSubExt20E4] ParameterName=WRPeriodDurationSeq ObjectType=0x7 DataType=0x0006 AccessType=wo PDOMapping=1 Count=1
Description	Write Period Duration of Sequence Cycle	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	wo	
Default value	0000_{hex}	

Design of the data bytes:

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

7.24.3 ON Time PWM LED x (Object 0x20E5, 0x20E6)

Objects 0x20E5 (LED1, 20 mA) and 0x20E6 (LED2, 350 mA) can be used to transmit the duty factor of the LED PWM output to the module.

The register's content determines the LEDs' duty cycle within the defined period duration. A value of 0xFF corresponds to a duty cycle of 100%.

Feature	Description / Value		EDS
Name	TonLED1	Object 0x20E5	[MxSubExt20Ex] ParameterName=TonLEDx ObjectType=0x7 DataType=0x0005 AccessType=wo PDOMapping=1 Count=1
	TonLED2	Object 0x20E6	
Description	ON Time PWM LED		
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	UNSIGNED8		
Sub index	01 … FE_{hex}		
Access	wo		
Default value	00000_{hex}		

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Byte 0:

Data bit	Desig-nation	Description	Notes
0 – 1		SSI Shift Register Frequency	→ Table 40
2		0 = OK 1 = Gray Code Decoding	
3		0 = Incremental Encoder Mode 1 = SSI busy (1= busy) (read only)	
4		Reserved	
5		0 = 1 = Error Reset (1 = clear error) (read, write)	
6		0 = 1 = Start with Sync (1= enable)	
7		0 = 1 = Continuous Sensor Read	

Duty cycle of PWM output for LED 1:

$t_{on} = t_{on}LED1 * 180$ ns

Period duration: PD = 45.9 µs → f = 21.8 kHz

Duty cycle of PWM output for LED 2:

$t_{on} = t_{on}LED2 * 1100$ ns

Period duration: PD = 2805 µs → f = 3.56 kHz

7.24.4 Motor Control (Object 0x20E7)

Motor control object 0x20E7 can be used to activate the sequence output and the status feedback in object 0x30E8 → Page 323.

Feature	Description / Value	EDS
Name	MotorControl	[MxSubExt20E7] ParameterName=MotorControl ObjectType=0x7 DataType=0x0006 AccessType=wo PDOMapping=1 Count=1
Description	Motor Control Register	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	wo	
Default value	0000_{hex}	

Design of the data bytes:

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

Bit	Description
0	1 => Activate motor sequence output
1	reserved
2	reserved
3	1 = Activate status message: High motor current (I^2t = 16 A^2s exceeded)
4	1 = Activate status message: Motor current (3.5 A) too high (additional peripheral reset)
5	1 = Activate status message: DC of module supply OK
6	1 = Activate status message: DC of motor supply OK
7	1 = Activate status message: LED peripheral reset
8	1 = Activate status message: Motor peripheral reset
9	1 = Activate status message: Motor induced voltage too high (additional peripheral reset)
10	1 = Activate overtemperature shutdown
11	1 = Reset 95 °C overtemperature (clear status message)
12	1 = Activate i^2t error shutdown
13	1 = Reset i^2t error (clear status message)
14	1 = Activate PWM 20 mA LED
15	1 = Activate PWM 3500 mA LED

7.24.5 Read PWM Sequenz Data (Object 0x30E0 to 0x30E3)

Objects 0x30E0 to 0x30E3 can be used to read the motor control PWM output sequence data from the module.

Feature	Description / Value		EDS
Name	RDSeq1	Object 0x30E0	[MxSubExt30Ex] ParameterName=RDSeq1 ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
	RDSeq2	Object 0x30E1	
	RDSeq3	Object 0x30E2	
	RDSeq4	Object 0x30E3	
Description	Read PWM Sequence Data		
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	ro		

Design of the data bytes:

	1 Byte								Byte 0:							
Sub index 1≦n≦254	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB					Time value / clock frequency										

Data bit	Desig-nation	Description
0 – 7		Time value / clock frequency
8 – 10		Time value / clock frequency
11		0 = relative time counter 1 = absolute time counter
12		0 = OFF 1 = left high ON
13		0 = OFF 1 = right high ON
14		0 = OFF 1 = left low ON
15		0 = OFF 1 = right low ON

7.24.6 Read Period Duration of Sequence Cycle (Object 0x30E4)

Object 0x30E4 can be used to read the duration of the motor control PWM output sequence.

$$\text{Period Time} = \frac{\text{RDPeriodDurationSeq}}{\text{Clock frequency}}$$

Feature	Description / Value	EDS
Name	RDPeriodDurationSeq	[MxSubExt30E4] ParameterName=RDPeriodDurationSeq ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Read Period Duration of Sequence Cycle	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	
Default value	0000_{hex}	

Design of the data bytes:

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

7.24.7 DC Driver Temperature (Object 0x30E5)

Object 0x30E5 represents the analog input value of the internal motor driver temperature in kelvins.

Feature	Description / Value	EDS
Name	DCDTempK	[MxSubExt30E5] ParameterName=DCDTempK ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	DC Driver Temperature (in °K)	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

7.24.8 DC Motor Current (Object 0x30E6)

Object 0x30E6 represents the analog input value of the internal motor current in mA. The total of the currents in both bridge sections that is relevant to the module's temperature rise will be measured.

Feature	Description / Value	EDS
Name	DCMotorCurrent	[MxSubExt30E6] ParameterName=DCMotorCurrent ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	DC Motor Current	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

7.24.9 DC Motor Diagnosis (Object 0x30E7)

Motor diagnostic object 0x30E7 can be used to read the module's "stored" system status. When read, the diagnostic messages will be reset. The over-temperature and i²t error error messages can only be reset using the motor control register.

Feature	Description / Value	EDS
Name	DCMotorDiag	[MxSubExt30E7] ParameterName=DCMotorDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	DC Motor Diagnosis Register	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

Bit	Description
0	reserved
1	1 = Incorrect sequence configuration. Sequence cannot be carried out.
2	1 = Incorrect sequence period duration configured. Sequence cannot be carried out.
3	1 = High motor current (I^2t = 16 A^2s exceeded)
4	1 = Motor current (3.5 A) too high (reset has been carried out)
5	1 = DC of module supply OK
6	1 = DC of motor supply OK
7	1 = Peripherie-Reset LED
8	1 = Motor peripheral reset
9	1 = Motor induced voltage too high (peripheral reset carried out)

Bit	Description
10	1 = 95 °C overtemperature
11	reserved
12	1 = i²t error shutdown activated
13 – 15	reserved

7.24.10 DC Motor Status (Object 0x30E8)

Motor status object 0x30E8 can be used to read the module's system status.

Feature	Description / Value	EDS
Name	DCMotorStatus	[MxSubExt30E8] ParameterName=DCMotorStatus ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	DC Motor Status Register	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

Bit	Description
0	1 = Motor sequence output is active
1	reserved
2	reserved
3	1 = High motor current (I^2t = 16 A^2s exceeded)
4	1 = Motor current (3.5 A) too high (reset has been carried out)
5	1 = DC of module supply OK
6	1 = DC of motor supply OK
7	1 = Peripherie-Reset LED
8	1 = Motor peripheral reset
9	1 = Motor induced voltage too high (peripheral reset carried out)
10	1 = 95 °C overtemperature

Bit	Description
11	reserved
12	1 = i²t error shutdown activated
13 – 15	reserved

7.24.11 DC motor I²T value (object 0x30E9)

Object 0x30E9 represents the analog input value of the internal measurement of motor energy I²t in [(1/160) A²s].

Feature	Description / Value	EDS
Name	DCMotorI2T	[MxSubExt30E9] ParameterName=DCMotorI2T ObjectType=0x7 DataType=0x0007 AccessType=ro PDOMapping=1 Count=1
Description	DC Motor I2T Value	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															

7.24.12 Module Diagnostic Messages (Object 0x30EA)

Object 0x30EA contains status information on the module's general operating status.

Feature	Description / Value	EDS
Name	ModuleDiag	[MxSubExt30EA] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Module Diagnostic Messages	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

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

Byte 0:

Data bit	Desig-nation	Description
0		0 = OK 1 = Supply Voltage NOK
1		0 = OK 1 = no sync
2		0 = OK 1 = FLASH Data CRC Error
3		0 = OK 1 = RAM Data CRC Error
4		0 = OK 1 = Inconsistent FLASH Data
5 – 7		reserved

7.24.13 Firmware Version (Object 0x40E0)

Object 0x40E0 can be used to read the firmware version using SDO-based access.

Feature	Description / Value	EDS
Name	FirmwareVersion	[MxSubExt40E0] ParameterName=FirmwareVersion ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	Firmware Version	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≤n≦254	1 Byte								Byte 0:							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															LSB

7.24.14 PWM Prescaler Register LED x (Object 0x40E1, 0x40E2)

Objects 0x40E1 and 0x40E2 can be used to read the PWM pre-scaler register for LED control.

Feature	Description / Value		EDS
Name	PreScaleLED1	Object 0x40E1	[MxSubExt40Ex] ParameterName=PreScaleLEDx ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
	PreScaleLED2	Object 0x40E2	
Description	PWM Prescaler Register LED x		
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED16		
Sub index	01 … FE_{hex}		
Access	ro		
Default value	0002_{hex}		

Design of the data bytes:

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

7.24.15 PWM Period Duration Register LED x (Object 0x40E3, 0x40E4)

Objects 0x40E3 and 0x40E4 can be used to read the set PWM period duration for LED control.

Feature	Description / Value		EDS
Name	PDLED1	Object 0x40E3	[MxSubExt40Ex] ParameterName=PDLEDx ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=0 Count=1
	PDLED2	Object 0x40E4	
Description	PWM Period Duration Register LED x		
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED8		
Sub index	01 … FE_{hex}		
Access	ro		
Default value	0002_{hex}		

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

7.24.16 DC Motor I2T Value Limit (Object 0x40E5)

Object 0x40E5 represents the analog shutdown value for the internal measurement of motor energy I²t in [(1/160) A²s].

Feature	Description / Value	EDS
Name	DCMotorI2TLimit	[MxSubExt40E5] ParameterName=DCMotorI2TLimit ObjectType=0x7 DataType=0x0007 AccessType=rw PDOMapping=1 Count=1
Description	DC Motor I2T Value Limit	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default	0000 0400_{hex}	

Design of the data bytes:

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

Sub index 1≦n≦254	4 Byte								3 Byte							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															

7.25 XN-322-1CNT-8DIO

This module supports the provision of data for analog inputs and analog outputs as per the specifications in CiA401. It allows for process data to be accessed with various vendor-specific objects.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

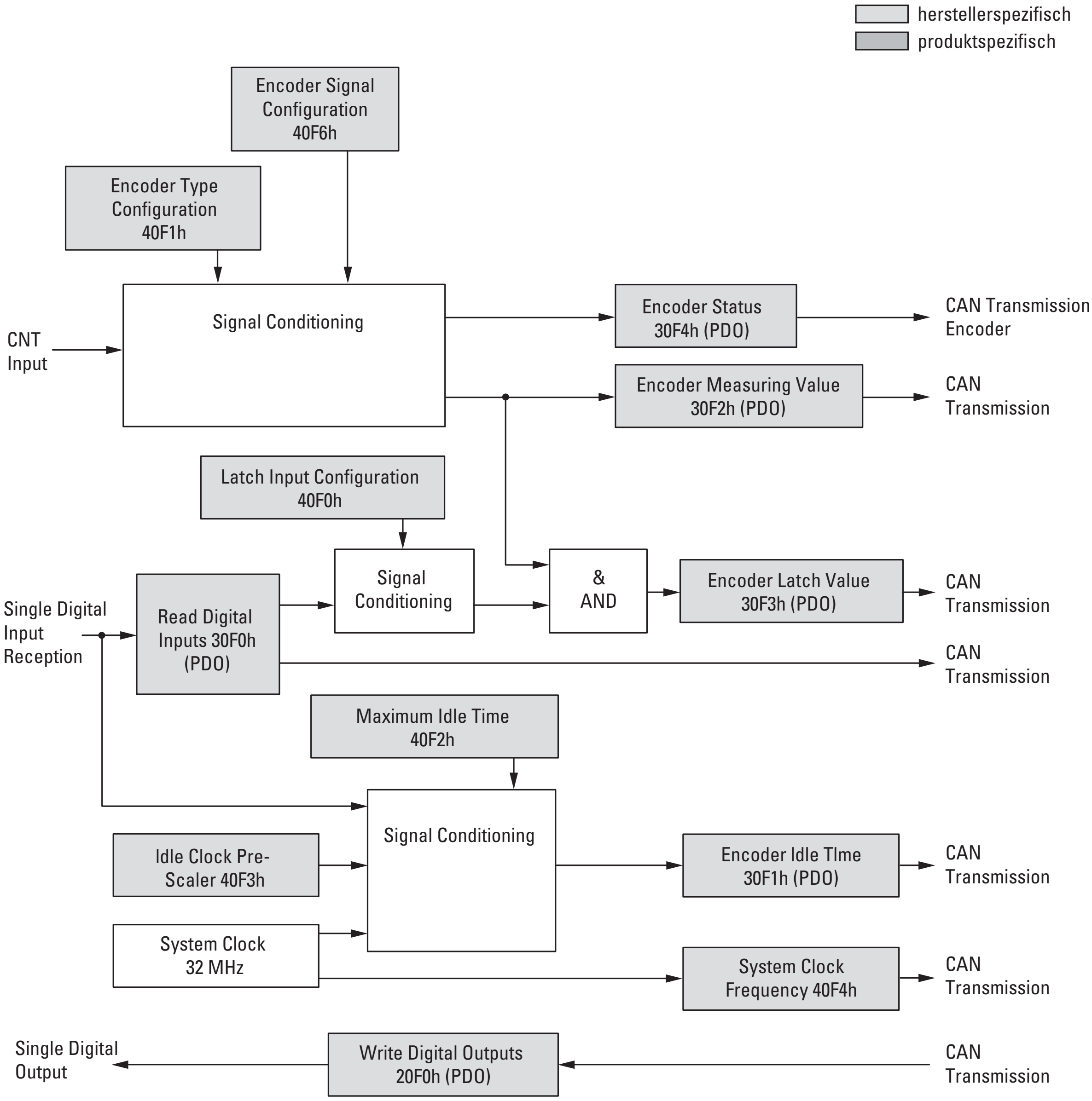


Figure 86: Block diagram showing the various CANopen objects for analog inputs

Manufacturer-specific objects

Index range for the XN-322-1CNT-8DIO module: x0F0 to x0FF

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x20F0	UNSIGNED8	Output1_4	Write Digital Outputs	Manual	rww	PDO
0x30F0	UNSIGNED8	Input1_4	Read Digital Inputs	Manual	ro	PDO
0x30F1	SIGNED32	IdleTime	Encoder Idle Time	Manual	ro	PDO
0x30F2	UNSIGNED16	CounterValue	Encoder Count Value	Manual	ro	PDO
0x30F3	UNSIGNED16	LatchValue	Encoder Latch Value	Manual	ro	PDO
0x30F4	UNSIGNED8	EncoderStatus	Encoder Status	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO
0x40F0	UNSIGNED8	LatchConfig	Latch Input Configuration	–	rw	SDO
0x40F1	UNSIGNED8	EncoderConfig	Encoder Type Configuration	–	rw	SDO
0x40F2	SIGNED32	MaxIdleTime	Maximum Idle Time	–	rw	SDO
0x40F3	UNSIGNED8	IdleClock	Idle Clock Pre-Scaler	–	rw	SDO
0x40F4	UNSIGNED8	SystemClock	System Clock Frequency	–	ro	SDO
0x40F5	UNSIGNED16	CounterValueSDO	Encoder Measuring Value SDO	–	ro	SDO
0x40F6	UNSIGNED8	SignalConfig	Encoder Signal Configuration	–	rw	SDO
0x40F7	UNSIGNED8	EncoderStatusSDO	Encoder Status SDO	–	ro	SDO
0x40F8	UNSIGNED8	LatchValueSDO	Encoder Latch Value SDO	–	ro	SDO

7.25.1 Write Digital Outputs (Object 0x20F0)

Object 0x20F0 transmits the digital value of the channels' digital signal outputs.

Feature	Description / Value	EDS
Name	Output1_4	[MxSubExt20F0] ParameterName=Output1_4 ObjectType=0x7 DataType=0x0005 AccessType=rww PDOMapping=1 Count=1
Description	Write Digital Outputs	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rww	
Default value	00_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	**Byte 0:**							
	B7	B6	B5	B4	B3	B2	B1	B0
					DO4	DO3	DO2	DO1

7.25.2 Read Digital Input (Object 0x30F0)

Object 0x30F0 transmits the digital value of the channels' digital signal inputs in a double word.

Feature	Description / Value	EDS
Name	Input1_4	[MxSubExt30F0] ParameterName=Input1_4 ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Digital Input	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	**Byte 0:**							
	B7	B6	B5	B4	B3	B2	B1	B0
					DI4	DI3	DI2	DI1

7.25.3 Encoder Idle Time (Object 0x30F1)

Object 0x30F1 contains the number of pulses triggered by the internal time reference (IdleClock) since the last encoder count value increment (rising edge on signal A). This register makes it possible to measure frequencies and speed.

Feature	Description / Value	EDS
Name	IdleTime	[MxSubExt30F1] ParameterName=IdleTime ObjectType=0x7 DataType=0x0004 AccessType=ro PDOMapping=1 Count=1
Description	Encoder Idle Time	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	INTEGER32	
Sub index	01 … FE_{hex}	
Access	ro	

An internal register will be internally incremented at the idle clock time interval defined in object 9x40F3 and copied to the encoder idle time register with every rising edge on the counter input (A) (X1 encoding), after which it will be reset.

This means that the content of the encoder idle time register provides a measurement of the time interval between two count pulses.

The counting direction will be determined using the second channel of counter input (B), with the sign (MSB) being interpreted as the direction of rotation.

This measurement will yield the "pulses per second" frequency as per the formula below:

$$f_{IPS} = \frac{f_{CLK}}{\text{(Encoder Idle Time * Idle Clock Pre-Scaler}}$$

f_{IPS}: "Pulses per second" frequency
f_{CLK} : Frequency Clock

If the speed increases, the value in the encoder idle time register will decrease, i.e., the time measured will be shorter. If the speed decreases, a larger number of pulses will be added, so that the time measured will be longer. If the register's content reaches its maximum value, it must be assumed that the motor has stopped or that there is an open wire. The comparison between the content of the internal count register and the specified maximum value in the maximum idle time register will cause the incrementing operation to be stopped, the counter value to be copied to the encoder idle time register, and the counting operation to then be restarted. The direction of rotation will then be indicated as being positive.

The maximum value that will result in the incrementing operation being stopped must be entered into the MaxIdleTime register.

The quality of the measurement will depend on the entries in the idle clock pre-scaler, the defined MaxIdleTime, and the number of pulses recorded in the encoder idle time register. The goal is to use the register ranges optimally while limiting the maximum time for the integration interval.

In regard to determining the rpm revolutions per minute, the encoder's resolution with X1 encoding and, if applicable, the gear must be taken into account. The following applies:

$$\text{Rpm} = \frac{f_{IPS} * 60\ [\text{s/min}]}{\text{Encoder resolution}[\text{i/r}]}$$

Timing diagram for the signals:

Signal A : Internal counter latch and restart if there is a rising edge. The internal IdleClock pulses are counted between two rising edges on signal A. When the second edge occurs, the total value is copied to the encoder idle time register and the internal counter is restarted.

Signal B : Positive direction of rotation if there is a rising edge on B after the counter starts; negative direction of rotation if there is a falling edge on B after the counter starts

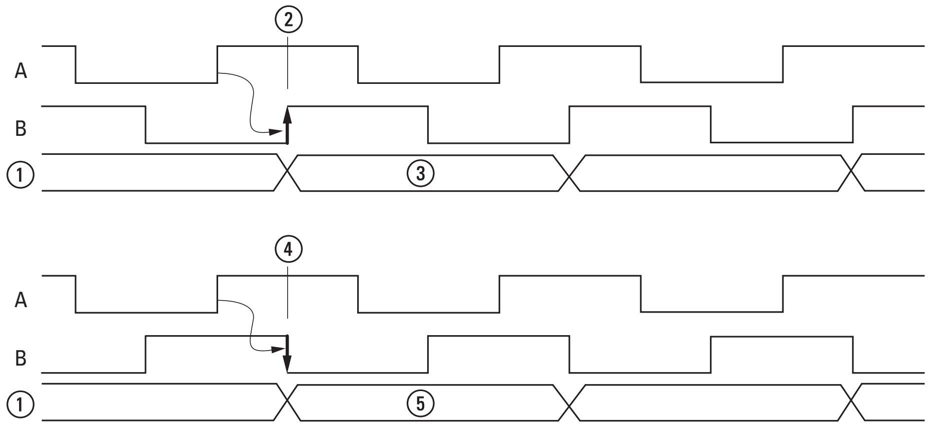


Figure 87: Signal diagram for signals A and B

① Period counter
② Rising edge
③ Positive value
④ Falling edge
⑤ Negative value

The signals are evaluated based on the X1 encoding for both signals (A and B). If the signal sequence does not follow this method, the edge will not be evaluated and the idle time register will report the maximum value -> motor stop also if there is an open wire for signal A, $\overline{A}$, B, or $\overline{B}$.

7.25.4 Encoder Measuring Value (Object 0x30F2)

Object 0x30F2 transmits the measured value, the number of counted pulses as per the counting method defined in object 0x40F6.

Feature	Description / Value	EDS
Name	CounterValue	MxSubExt30F2] ParameterName=CounterValue ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Encoder Count Value	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

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

7.25.5 Encoder Latch Value (Object 0x30F3)

Object 0x30F3 transmits the measured value stored with a latch operation.

Feature	Description / Value	EDS
Name	LatchValue	[MxSubExt30F3] ParameterName=LatchValue ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=1 Count=1
Description	Encoder Latch Value	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	UNSIGNED16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

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

7.25.6 Encoder Status (Object 0x30F4)

Object 0x30F4 transmits the counter's status signals.

Feature	Description / Value	EDS
Name	EncoderStatus	[MxSubExt30F4] ParameterName=EncoderStatus ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Encoder Status	
Object Code	VAR	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

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

The content of bit 5 will be automatically reset after the register is read. This ensures that the zero state, which will only be present for a short time, can also be read. This must be taken into account if the object is being continuously read with the PDO.

Byte 0:

7	6	5	4	3	2	1	0
–	–						

7.25.7 Latch Input Configuration (Object 0x40F0)

Object 0x40F0 is used to configure the digital inputs in the add-on function in terms of whether, and when, the current counter value should be stored in the LATCH register.

<table>
<tr><th>Feature</th><th>Description / Value</th><th>EDS</th></tr>
<tr><td>Name</td><td>LatchConfig</td><td rowspan="7">[MxSubExt40F0]
ParameterName=LatchConfig
ObjectType=0x7
DataType=0x0005
AccessType=rw
PDOMapping=0
Count=1</td></tr>
<tr><td>Description</td><td>Latch Input Configuration</td></tr>
<tr><td>Object Code</td><td>ARRAY</td></tr>
<tr><td>Mapping</td><td>SDO</td></tr>
<tr><td>Data type</td><td>UNSIGNED8</td></tr>
<tr><td>Sub index</td><td>01 … FE hex</td></tr>
<tr><td>Access</td><td>ro</td></tr>
</table>

Design of the data bytes:

<table>
<tr><td rowspan="3">Sub index
1≤n≦254</td><td colspan="8">Byte 0:</td></tr>
<tr><td>B7</td><td>B6</td><td>B5</td><td>B4</td><td>B3</td><td>B2</td><td>B1</td><td>B0</td></tr>
<tr><td colspan="2">Input 4</td><td colspan="2">Input 3</td><td colspan="2">Input 2</td><td colspan="2">Input 1</td></tr>
</table>

Byte 0:

7	6	5	4	3	2	1	0
–	–						

<table>
<tr><th colspan="2">Input</th><th>Description</th></tr>
<tr><td colspan="2">Input 1</td><td></td></tr>
<tr><td>Bit 1</td><td>Bit 0</td><td></td></tr>
<tr><td>0</td><td>0</td><td>No latch</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>0</td><td>Falling and rising edges</td></tr>
<tr><td colspan="2"></td><td></td></tr>
<tr><td colspan="2">Input 2</td><td></td></tr>
<tr><td>Bit 3</td><td>Bit 2</td><td></td></tr>
<tr><td>0</td><td>0</td><td>No latch</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>0</td><td>Falling and rising edges</td></tr>
<tr><td colspan="2"></td><td></td></tr>
</table>

Input		Description
Input 3		
Bit 5	Bit 4	
0	0	No latch
0	1	Rising edge
1	0	Falling edge
0	0	Falling and rising edges
Input 4		
Bit 7	Bit 6	
0	0	No latch
0	1	Rising edge
1	0	Falling edge
0	0	Falling and rising edges

→ If multiple inputs are used for the latch function, their signals will be OR'd.

7.25.8 Encoder Type Configuration (Object 0x40F1)

Object 0x40F1 can be used to configure the type of input signal.

Feature	Description / Value	EDS
Name	EncoderConfig	[MxSubExt40F1] ParameterName=EncoderConfig ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	Encoder Type Configuration	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE hex	
Access	rw	
Default	0 x 00 hex	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Byte 0:

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	Conf

Data bit	Designation	Description
0		0= TTL signal encoder 1 = RS422 signal encoder

7.25.9 Maximum Idle Time (Object 0x40F2)

Object 0x40F2 will contain the following information depending on the type of access used:

- READ
  Content of encoder idle time register (object 0x30F1)
- WRITE
  Maximum value that the internal count register and, accordingly, the encoder idle time register (object 0x30F1) are allowed to assume

Object (40F2) can be used to define the maximum integration interval for the encoder idle time. If the internal count register is incremented all the way to this value, its count will cease to be incremented, it will be copied to the encoder idle time register, and counting will be restarted.

The value must be interpreted as a motor stop or open wire. This means that the content of maximum idle time can be used to define the time after which a motor stop will be detected.

$$T_{max} = \frac{\text{Maximum Idle Time * Idle Clock Pre-Scaler}}{f_{CLK}}$$

Feature	Description / Value	EDS
Name	MaxIdleTime	[MxSubExt40F2] ParameterName=MaxIdleTime ObjectType=0x7 DataType=0x0004 AccessType=rw PDOMapping=1 Count=1
Description	Maximum Idle Time	
Object Code	ARRAY	
Mapping	SDO	
	Manual	
Data type	INTEGER32	
Sub index	01 … FE hex	
Access	rw	

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								yte 2							
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
	MSB															

7.25.10 Idle Clock Pre-Scaler (Object 0x40F3)

Object 0x40F3 transmits the pre-scaling factor for the speed measurement.

Feature	Description / Value	EDS
Name	IdleClock	[MxSubExt40F3] ParameterName=IdleClock ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	Idle Clock Pre-Scaler	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE hex	
Access	rw	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	IdleClockPreScaler							

The IdleReference prescaler is used to determine the time base for measuring the time between two edges on the counter inputs. The following applies:

$$\text{TdleClock} = \frac{\text{Clock Idle Time}}{\text{Idle Clock Pre-Scaler}}$$

f_{CLK} : 32000000 Hz
Clock [Hz] is an internal module time base.

Example

If a value of 255 (dec) is loaded onto the idle clock prescaler, this will yield an idle reference time of 8 µs. This corresponds to an f_{IDS} of 125490 Hz.

$$\text{IdleClock} = \frac{32000000\text{ Hz}}{255} = 125490\text{ Hz}$$

A lower value for the IdleClockPreScaler will make it possible to obtain more accurate speed measurements at higher speeds. On the other hand, a higher value will increase the measuring accuracy at low speeds.

7.25.11 System Clock Frequency (Object 0x40F4)

Object 0x40F4 delivers the internal module system clock frequency in MHz.

Feature	Description / Value	EDS
Name	SystemClock	[MxSubExt40F4] ParameterName=SystemClock ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=0 Count=1
Description	System Clock Frequency	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	System Clock Frequency in MHz							

Internal clock frequency: 32 MHz (20 hex).

7.25.12 Encoder Measuring Value SDO (Object 0x40F5)

Object 0x40F5 transmits the measured value, the number of pulses counted (like object 0x30F2), as an SDO.

Feature	Description / Value	EDS
Name	CounterValueSDO	[MxSubExt40F5] ParameterName=CounterValueSDO ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	Encoder Measuring Value	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

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

7.25.13 Encoder Signal Configuration (Object 0x40F6)

Object 0x40F6 is used to configure the counter's encoding.

Feature	Description / Value	EDS
Name	SignalConfig	[MxSubExt40F6] ParameterName= SignalConfig ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	Encoder Signal Configuration	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	SignalConfig							

Byte 0:

Data bit		Designa-tion	Description
0 – 1			reserved
2			0 = Normal evaluation 1 = Inverted R zero position evaluation
3			0 = Normal evaluation 1 = Inverted phase B evaluation
4 – 5			
Bit 5	Bit 4		
0	0		
0	1		
1	0		
1	1		
6 – 7			reserved

In AB mode, the phase shift of the A and B input signals is used to determine the pulse and direction. To do this, signals A and B are evaluated for rising and falling edges.

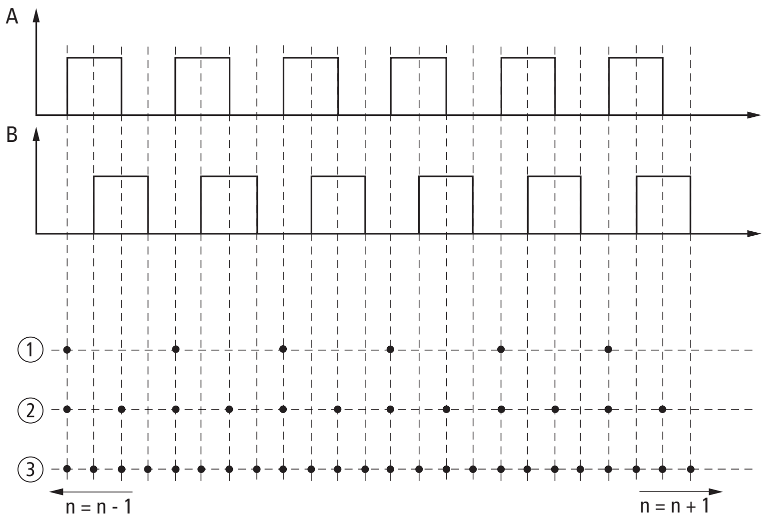


Figure 88: Signal diagram

1 1-way
2 2-way
3 4-way

The dots represent the points at which the count changes.

If the signal sequence is followed in the direction the arrow is pointing (towards the right), this corresponds to a positive counting direction. If it is followed against it, this corresponds to a negative counting direction.

7.25.14 Encoder Status SDO (Object 0x40F7)

Object 0x30F4 transmits the counter's status signals as an SDO.

Feature	Description / Value	EDS
Name	EncoderStatusSDO	[MxSubExt40F7] ParameterName= EncoderStatusSDO ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=0 Count=1
Description	Encoder Status SDO	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED8	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0

Byte 0:

Design of the data bytes:

Byte 0:

Sub-Index 1≦n≦254

7	6	5	4	3	2	1	0
–	–	–					

Data bit	Desig-nation	Description
0 – 3		0 = 24VDC OK 1 = 24VDC not OK
4		0 = 1 = Current position is zero position
5		0 = Register has been read 1 = Zero position has been crossed and not yet read.
6 – 7		reserved

Bit 5 will be automatically reset after the register is read. This ensures that the zero state, which will only be present for a short time, can also be read. This must be taken into account if the object is being continuously read.

7.25.15 Encoder Latch Value SDO (Object 0x40F8)

Object 0x40F8 transmits the stored measured value, the number of pulses counted after a latch event (like object 0x30F3), as an SDO.

Feature	Description / Value	EDS
Name	LatchValueSDO	[MxSubExt40F8] ParameterName=LatchValueSDO ObjectType=0x7 DataType=0x0006 AccessType=ro PDOMapping=0 Count=1
Description	Encoder Latch Value SDO	
Object Code	ARRAY	
Mapping	SDO	
Data type	UNSIGNED16	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	**Byte 0:**							
	B7	B6	B5	B4	B3	B2	B1	B0

7.26 XN-322-2SSI

This module supports the provision of data with various vendor-specific objects as per the specifications in CiA401.

A distinction is drawn between the following CANopen objects within this context:

- System-specific objects that the CANopen standard requires for a system.
- Product-specific objects that the CANopen standard requires for a product group, e.g., for analog modules.
- Vendor-specific objects that the CANopen standard does not require and that the manufacturer implements for data communication purposes.

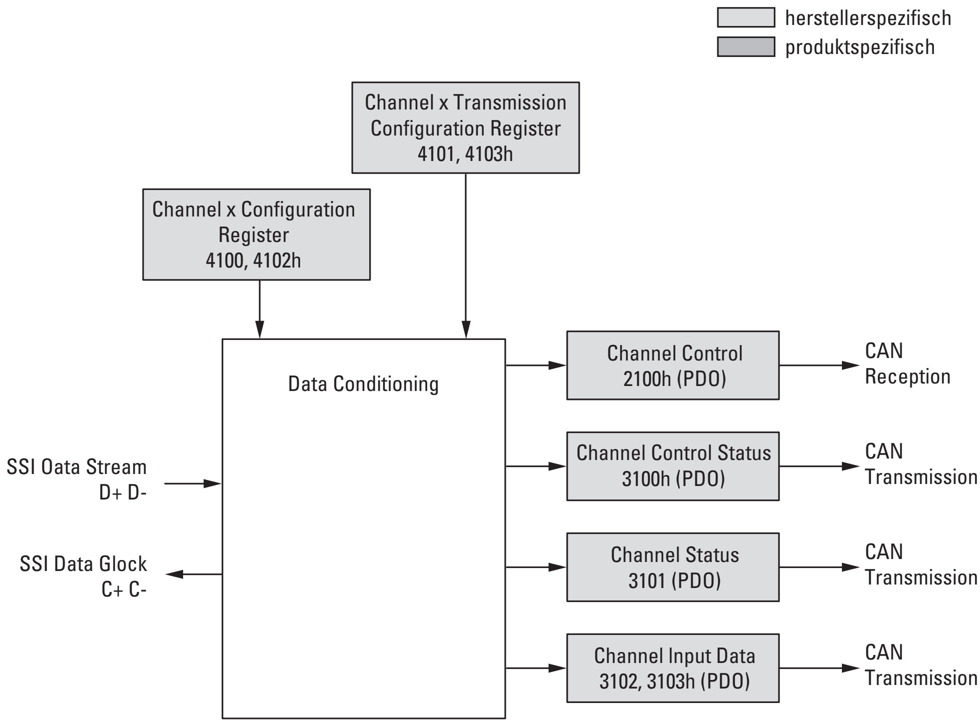


Figure 89: Block diagram showing the various CANopen objects for transmitting data streams

Manufacturer-specific objects

Index range for the XN-322-2SSI module: x100 to x10F

Index (hex)	Data Type	Name	Function	Mapping	Access	
0x1027	UNSIGNED16	ModuleID	Module Identification Number → Section "6.2.6 Module Identification Number (Object 0x1027)", page 76	–	ro	SDO
0x2100	UNSIGNED8	StartReadCycle	Start Read Cycle	Manual	wo	PDO
0x3100	UNSIGNED8	ReadCycleState	Read Cycle State	Manual	ro	PDO
0x3101	UNSIGNED8	ModuleDiag	Module Diagnosis	Manual	ro	PDO
0x3102	UNSIGNE32	InputChannel1	Input Data Channel 1	Manual	ro	PDO
0x3103	UNSIGNE32	InputChannel2	Input Data Channel 2	Manual	ro	PDO
0x4001	VISIBLE STRING	SerialNumber	Serial Number → Section "6.2.8 Serial Number (Object 0x4001)", page 78	–	const	SDO
0x4004	UNSIGNED8	UserLEDControl	User LED Control → Section "6.2.11 User LED Control (Object 0x4004)", page 81	–	rw	SDO
0x400C	VISIBLE STRING	ProductName	Product Name → Section "6.2.14 Product Name (Object 0x400C)", page 82	–	ro	SDO
0x4100	UNSIGNED8	ConfigurationRegisterChannel1	Configuration Register Channel 1	–	rw	SDO
0x4101	UNSIGNED8	StateRegisterChannel1	State Register Channel 1	–	rw	SDO
0x4102	UNSIGNED8	ConfigurationRegisterChannel2	Configuration Register Channel 2	–	rw	SDO
0x4103	UNSIGNED8	StateRegisterChannel2	State Register Channel 2	–	rw	SDO

7.26.1 Start Read Cycle (Object 0x2100)

Object 0x2100 contains the write access to data used to control the module's channels.

Feature	Description / Value	EDS
Name	StartReadCycle	[MxSubExt2100] ParameterName=StartReadCycle ObjectType=0x7 DataType=0x0005 AccessType=wo PDOMapping=1 Count=1
Description	Start Read Cycle	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE hex	
Access	wo	
Default value	0000_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Byte 0:

Data bit	Designation	Description
0		0 = 1 = Start Read Cycle Channel 1
1		0 = 1 = Start Read Cycle Channel 2
2 – 7		reserved

7.26.2 Read Cycle State (Object 0x3100)

Object 0x3100 contains the read access to data used to control the module's channels

Feature	Description / Value	EDS
Name	ReadCycleState	[MxSubExt3100] ParameterName=ReadCycleState ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Read Cycle State	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Byte 0:

Data bit	Desig-nation	Description
0		0 = 1 = Start Read Cycle Channel 1
1		0 = 1 = Start Read Cycle Channel 2
2 – 7		reserved

7.26.3 Module Diagnosis (Object 0x3101)

Object 0x3101 contains data on the channels' status.

Feature	Description / Value	EDS
Name	ModuleDiag	[MxSubExt3101] ParameterName=ModuleDiag ObjectType=0x7 DataType=0x0005 AccessType=ro PDOMapping=1 Count=1
Description	Module Diagnosis	
Object Code	ARRAY	
Mapping	PDO	
	Manual	
Data type	UNSIGNED8	
Sub index	01 … FE hex	
Access	ro	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Byte 0:

Data bit	Desig- nation	Description
0		0 = 1 = Channel 1 „started“
1		0 = 1 = Channel 1 „busy“
2		0 = 1 = Channel 1 „toggle“
3		0 = 1 = Channel 1 SSI Error/ Invalid Z-Position
4		0 = 1 = Channel 2„started“
5		0 = 1 = Channel 2„busy“
6		0 = 1 = Channel 2„toggle“
7		0 = 1 = Channel 2SSI Error/ Invalid Z-Position

7.26.4 Input Channel x (Object 0x3102 to 0x3103)

Objects 0x3103 and 0x3104 represent the encoder's formatted 32-bit values.

Feature	Description / Value		EDS
Name	Input Channel x		[MxSubExt310x] ParameterName=InputChannelx ObjectType=0x7 DataType=0x0007 AccessType=ro PDOMapping=1 Count=1
Description	InputChannel1	Object 0x3102	
	InputChannel2	Object 0x3103	
Object Code	ARRAY		
Mapping	PDO		
	Manual		
Data type	UNSIGNED32		
Sub index	01 … FE hex		
Access	ro		

Design of the data bytes:

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

Sub index 1≦n≦254	3 Byte								Byte 2							
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	MSB															

7.26.5 Channel Configuration Register (Object 0x4100, 0x4102)

Objects 0x4100 and 0x4102 contain configuration information regarding the channel's function.

Feature	Description / Value		EDS
Name	Configuration Register Channel1		[M8SubExt410x] ParameterName=ConfigurationRegisterChannelx ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	ConfigurationRegisterChannel1	Object 0x4100	
	ConfigurationRegisterChannel2	Object 0x4102	
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED8		
Sub index	01 … FE hex		
Access	rw		

Design of the data bytes:

Sub index 1≤n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Byte 0:

Data bit	Designation	Description	Notes
0 – 5		SSI Shift Register Size (max.32 Bit)	→ Table 39
6		0 = OK 1 = Read Value Twice	
7		0 = Incremental Encoder Mode 1 = SSI Encoder Mode	

Table 39: SSI Shift Register Size

	Bit					
SSI Shift Register Size [Bit]	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
…						
30	0	1	1	1	0	1
31	0	1	1	1	1	0
32	1	0	0	0	0	0

7.26.6 Channel Transmission Configuration Register (Object 0x4101, 0x4103)

Objects 0x4101 and 0x4103 contain configuration information regarding the channel's data transfer.

Feature	Description / Value		EDS
Name	State Register Channel		[M8SubExt410x] Parameter Name=StateRegisterChannel1 ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 Count=1
Description	StateRegisterChannel1	Object 0x4101	
	StateRegisterChannel2	Object 0x4103	
Object Code	ARRAY		
Mapping	SDO		
Data type	UNSIGNED8		
Sub index	01 … FE hex		
Access	rw		

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Byte 0:

Data bit	Designation	Description	Notes
0 – 1		SSI Shift Register Frequency	→ Table 40
2		0 = OK 1 = Gray Code Decoding	
3		0 = Incremental Encoder Mode 1 = SSI busy (1= busy) (read only)	
4		Reserved	
5		0 = OK 1 = Error Reset (1 = clear error) (read, write)	
6		0 = OK 1 = Start with Sync (1= enable)	
7		0 = OK 1 = Continuous Sensor Read	

Table 40: SSI Shift Register Frequency

SSI Shift Register Frequency	Bit	
	B1	B0
125 kHz	0	0
250 kHz	0	1
500 kHz	1	0
1 MHz	1	1

8 Station variants

8.1 Definition of Terms

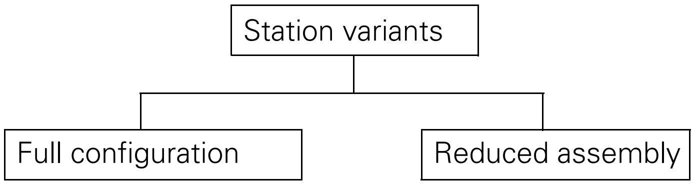


Figure 90: Station variant definitions

- Station variants
  The "station variants" operating mode on the XN300 gateway is based on a "full configuration" and one or more "reduced assemblies," i.e., one or more subsets of the full configuration. This mode makes it possible to keep addressing an unchanged full configuration on the field bus side even when there is a reduced assembly on the gateway's module side.
- Full configuration
  Configured maximum setup for the XN300 gateway. This full configuration is used to define the addresses that should be used to communicate with the individual XN300 slice modules independently of which XN300 slice modules are actually physically connected to the XN300 gateway.
- Reduced assembly
  The term "reduced assembly" refers to a permissible subset of the full configuration. In special cases, a reduced assembly may actually be the same as the full configuration.
- Actual configuration
  The XN300 slice modules that are physically present and currently being used with the gateway.
- Target configuration
  The target configuration is the configuration for the XN300 slice modules on the system bus that is read by the XN300 gateway and stored in its memory when the gateway is switched on with DIP switch 9 = "OFF."

8.2 Reason why this mode was implemented

The "station variants" operating mode enables users – and OEM users in particular – to use a variety of increasingly expanding XN300 system configurations with the XN-312-GW-CAN gateway without having to make changes to the corresponding user program.

To get started, the user needs to create a PLC program with an XN300 gateway as a CANopen node. This XN300 gateway needs to be created with the desired full configuration with all XN300 slice modules in the program.

The station variant that is actually being used with the gateway can correspond to the full configuration or to a subset of it (in the latter case, not all XN300 slice modules need to be present). Regardless of the configuration used, however, the order of the XN300 slice modules must not be altered.

With the help of the function block, the user program will determine whether one of the permitted reduced assemblies is physically connected to the gateway. Once this occurs, a set of program code modified as necessary for the specific reduced assembly can be executed in the application.

8.3 How the "station variants" operating mode works

The gateway can be run in "standard" mode or "station variants" mode. Setting SDO 0x4005, "Activation StationVariants Mode," to TRUE will make the gateway switch to the "station variants" mode. XN300 Assist can be used to determine whether the "station variants" mode has been selected.

There are two initialization phases.

Initialization phase 1

During initialization phase 1, the XN300 gateway's CANopen status will be PRE-OPERATIONAL.

If you want to use the "station variants" functionality in the user program, the PLC must set the "Activation StationVariants Mode" entry to TRUE during initialization phase 1. Initialization phase 1 ends with the "Start-Remote-Node" NMT frame.

Initialization phase 2

During initialization phase 2, the XN300 gateway's CANopen status will be OPERATIONAL.

In the "station variants" operating status, the defined full configuration will be stored on the XN300 gateway as a list of XN300 slice modules in a permissible order. The information on the station variant to be used on the gateway will also be stored as a list. These lists are written to the gateway via the PLC.

As long as there is information on the full configuration/reduced assembly on the gateway, the gateway will use the aforementioned lists for the following purposes:

- To run a validation check against the target configuration
- To create a default mapping and an object dictionary based on the full configuration

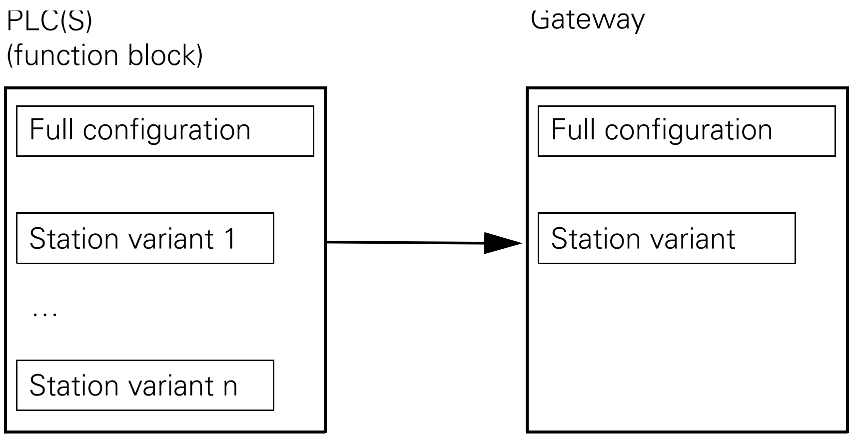


Figure 91: How the station variants are stored on the PLC and on the gateway

With regard to the user program, the gateway will behave as though the full configuration were connected. In fact, address mapping will be based on the full configuration.

After receiving the "Start-Remote-Node" NMT frame, the gateway will switch to the "OPERATIONAL" status.

If, during initialization phase 1, the "Activation StationVariants Mode" entry = FALSE, the gateway will send the inputs to the PLC and forward the output data received from the PLC to the outputs as soon as it switches to the "OPERATIONAL" status. In this case, the gateway will be in "standard" mode.

If, on the other hand, the "Activation StationVariants Mode" entry = TRUE, the inputs will not be sent to the PLC and the output data received from the PLC will not be forwarded (yet) to the outputs even when the "OPERATIONAL" status is reached.

After this, the PLC will need to check the full configuration list stored on the gateway.

The PLC can handle the list as follows:

- Approve
- Edit the existing list / create a new list
- delete

If the list is approved, the gateway will start, the outputs will be connected to the hardware, and the EMCY frame will be canceled.

If the PLC edits or deletes the list or creates a new one, this will trigger a gateway reset. In this case, the gateway will restart with the data from the new list.

The full configuration check carried out by the PLC consists of the following steps:

4. A list with all permitted station variants is stored on the PLC. A checksum can be calculated based on any of the corresponding lists.
5. The PLC reads the checksum for the station variant stored on the gateway.

6. If this checksum matches one of the checksums on the PLC, this means that the selected station variant on the gateway is permitted and can be used.
7. If the checksums do not match, the actual configuration will be read from the gateway. The actual configuration will then be compared with the permitted station variants found in the list on the PLC.
8. If the PLC finds a matching station variant, it will load the corresponding list onto the gateway. If it does not, the PLC will delete the list on the gateway.

8.3.1 General principles / behavior of the gateway and the PLC

As soon as the config check is active, it will always be carried out regardless of the station variant. Please note that the config check is used to check whether the actual configuration matches the target configuration that has been stored – the check for permitted station variants is carried out afterwards.

→ Please note that if the config check is active and you want to put a different station variant into operation, you will first need to store the new station variant as a target configuration in the device's memory → "Switching the gateway on with a config check" section on page 38.

What happens if the "station variants" operating mode is enabled?

- On the field bus side, the gateway will behave as though the full configuration were physically connected. The address mapping used will be based on the full configuration.
- Attempts to access variable values from missing nodes will return an undefined value.
- Outputs written with the use of PDOs will not be set until the PLC has confirmed the selected station variant by writing the checksum back to the gateway.

When will the gateway exit "station variants" mode?

The gateway will exit the "station variants" operating mode and delete its lists for the full configuration and for the selected station variants when one of the following events occurs:

- A change to the actual configuration for the XN300 slice modules is detected on the gateway.
- The gateway receives a start-remote-node command without the enable signal for the "station variants" operating mode being activated before, i.e., "Activation StationVariants Mode" = FALSE.

How does the enable process in which the PLC sends an enable signal work?

The function block (FB) asks the gateway for the checksum. After receiving this checksum, the FB checks whether one of the station variants stored on the PLC has the same checksum as the station variant on the gateway (this check practically verifies that the station variant on the gateway corresponds to a permitted reduced assembly). If it does, the PLC writes the checksum back to the gateway, which in turn interprets this as an enable signal. The gateway will not start transferring data until it receives this enable signal.

Initialization phases

When the "station variants" mode is active, there will be two initialization phases on the PLC. The first one is initialization phase 1, which will be carried out by the CANopen master. During this phase, the CANopen master will carry out the mapping, write the "Activation StationVariants Mode" object to the gateway, and switch the gateway to the "OPERATIONAL" status. No data will be transferred yet at this point, i.e., in order for data to start being transferred, the enable process needs to send an enable signal first.

8.3.2 Startup behavior

The user needs to create an application. The desired full configuration needs to be set up in the PLC configuration.

The enable signal for the "station variants" operating mode needs to be turned on in the configuration settings for the XN-321-GW-CAN gateway (object 0x4005/Sub0 == 0x01).

Information regarding the full configuration and the permitted reduced assembly variants will be passed to the "StationVariants" function block.

When the gateway starts, the application will run through the following steps by calling the "XN300_StationVariants" function block instance:

1. The gateway is started and communication via the CAN bus is enabled; PRE-OPERATIONAL CANopen status. Continue to step 2.
2. Initialization phase 1 for the gateway is carried out by the PLC's CANopen master.
   During this phase, the "Activation StationVariants Mode" entry, SDO 0x4005/0, must be set to 1. Continue to step 3.
3. The PLC's CANopen master switches the gateway to "OPERATIONAL" status. Continue to step 4.
4. The "XN300_StationVariants" function block instance queries whether the "Activation StationVariants Mode" entry on the gateway has a value of TRUE.

If it does: continue to step 5.

If it does not, continue to step 12.
The entry may not have a value of TRUE if, for example, object 0x4005 was not set because the "station variants" operating mode is not desired or because the gateway features old firmware that does not support this operating mode and, accordingly, SDO 0x4005.

5. The function block instance compares the checksum for the station variant on the gateway with the checksum for the station variant expected / permitted by the application (SDO 0x4006/0).
   - If the check fails because a matching station variant could not be found: continue to step 6.

- If the check is OK, i.e., if a matching station variant is found: continue to step 10.

6. The function block instance reads the target configuration from the gateway and checks whether this target configuration matches one of the station variants stored on the PLC.
   - If the target configuration matches one of the station variants on the PLC: continue to step 7.
   - If a match cannot be found: continue to step 13.
7. The function block instance writes the full configuration to the gateway (SDO 0x4007). Continue to step 8.
8. The function block instance writes information regarding the reduced assembly to the gateway (SDO 0x4008). Continue to step 9.
9. The checksum is written (SDO 0x4006/0), causing the gateway to store the full configuration/reduced assembly and reboot. Continue to step 1.
10. The function block instance writes the checksum back (SDO 0x4006/0), giving the gateway the enable signal the latter needs in order to start transferring data. Continue to step 11.
11. Normal operation – data exchange.
12. "ERROR_NOT_ACTIVATED" error: no enable signal in the application, the gateway is in "standard" mode.
13. "ERROR_NO_SUBASSEMBLY" error: no enable signal in the application, no data transfers by the gateway.

8.3.3 "Station Variants" function block

The "StationVariants" function block is found in the EA_XN300.library library and can be used in XSOFT-CODESYS 3.5.xx and higher.

The functionality implemented in this library can be used to access XN-312-GW-CAN CANopen nodes.

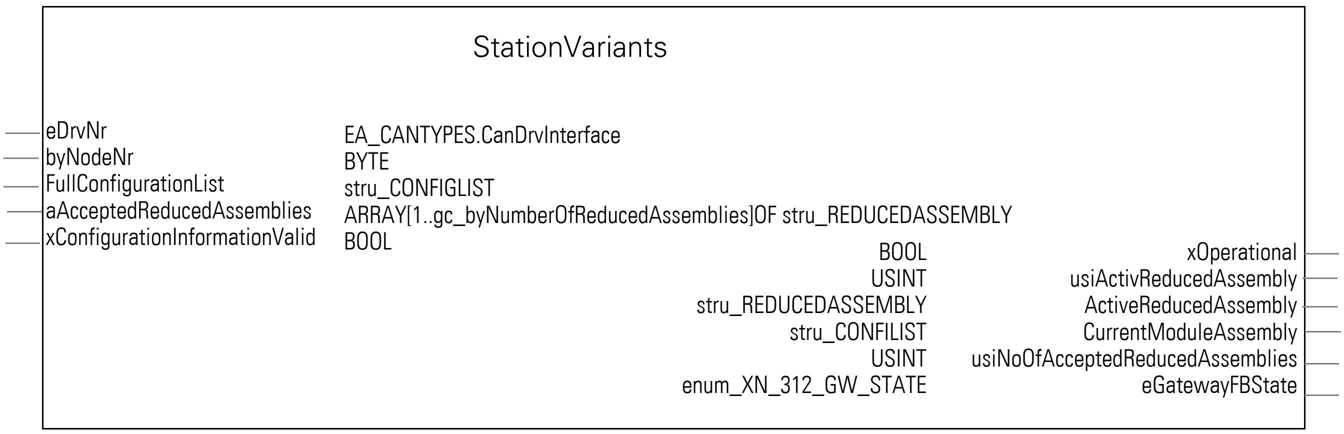


Figure 92: StationVariants function block in the EA_XN300 library

For more information on the function block, you can consult the EA_XN300_CAN.library library as soon as you add it to your project. The EA_XN300_CAN_de (German) and EA_XN300_CAN_en (English) PDF files can be accessed by using the following command sequence:

Library manager->EA_XN300_CAN->Documentation.

8.3.4 Overview

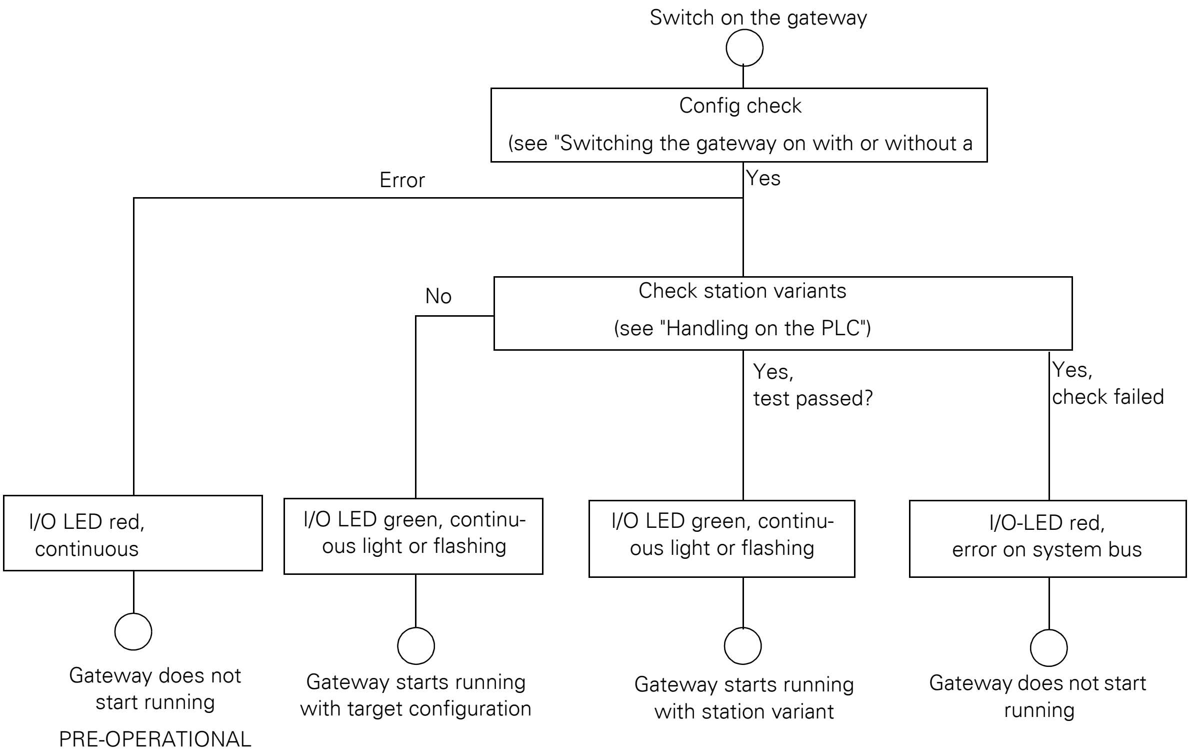


Figure 93: Possible gateway operating modes

8.3.5 Handling on the PLC

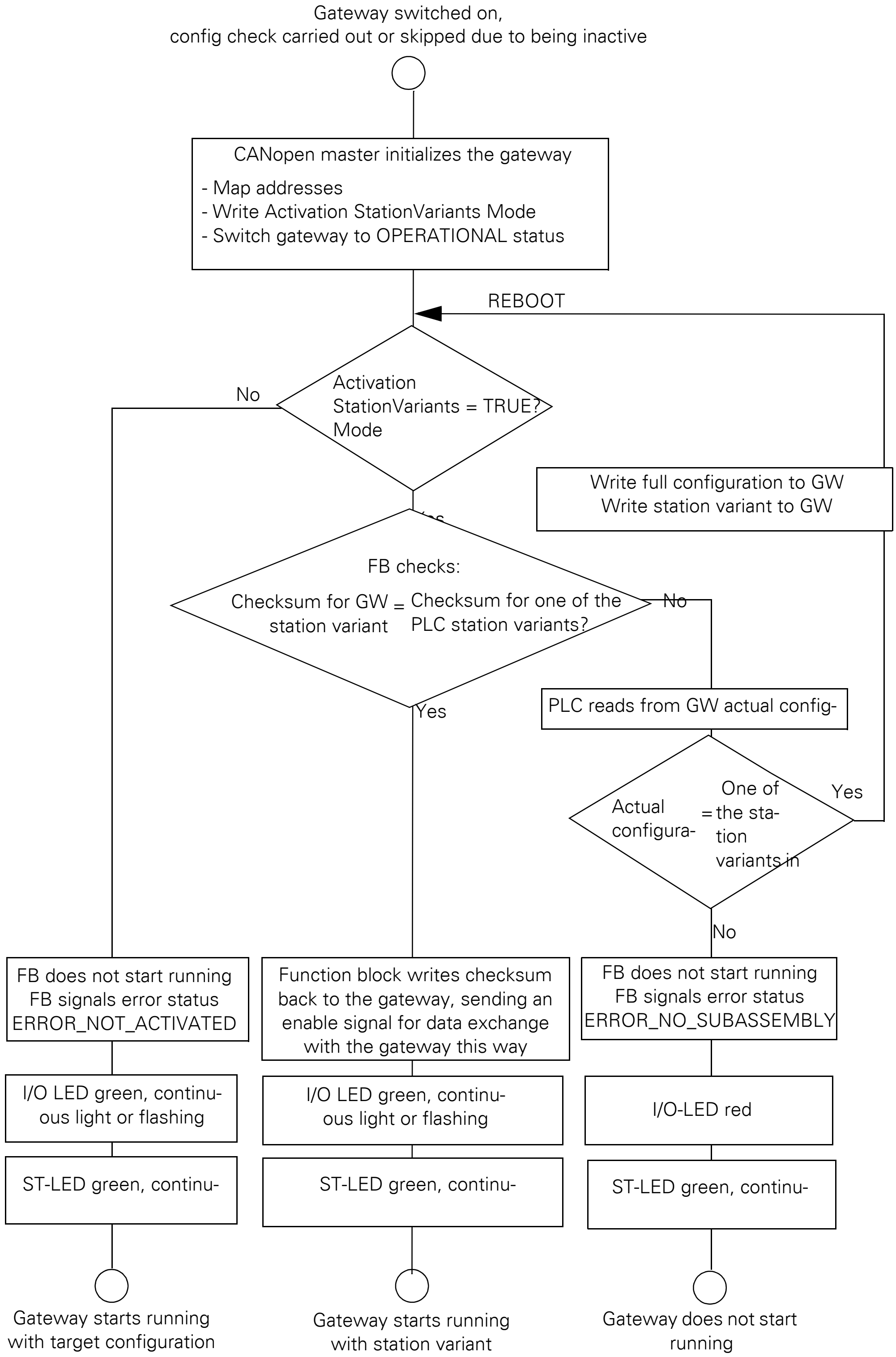


Figure 94: PLC handling when the gateway is switched with and without station variants

8.4 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
- Generating a project-specific EDS file
- Status indication of the inputs/outputs
- Wiring test
- Importing actual configurations
- 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.

XN300-Assist version 1.10 and higher can show when the "station variants" operating mode is in use.

If the "station variants" operating mode is active, all the XN300 slice modules created in the full configuration will be shown in XN300-Assist (modules that are not physically present will be transparent).

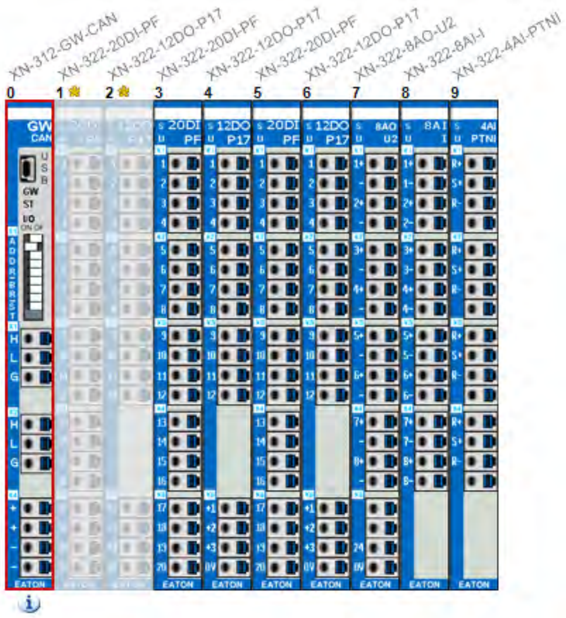


Figure 95: XN300-Assist with full configuration and reduced assembly

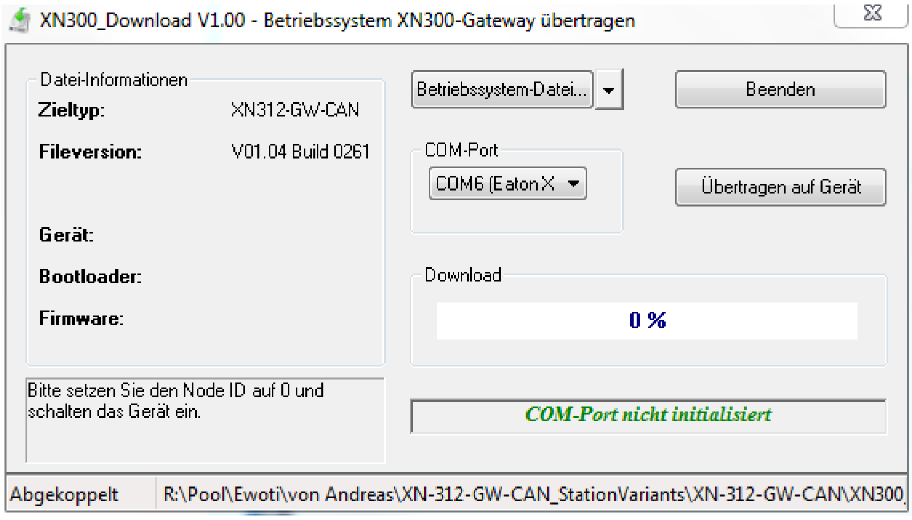
8.5 Loading new firmware onto the gateway

In order to be able to load new firmware onto the gateway, you will first need to install XN300-Assist on your computer.

To load new firmware onto the gateway, follow the steps below:

- ▶ Switch DIP switches 1 to 5 on the gateway to the OFF position. This will set the node ID to 0.
- ▶ Start the gateway with the set node ID of 0.
  The gateway will be in update mode.
- ▶ Use a mini USB cable to connect the XN-312-GW-CAN gateway to your computer's USB port.
- ▶ If drivers for the mini USB cable are not installed, install the drivers by using the following command sequence:
  Windows START-> All Programs-> Eaton-> XN300-Assist-> XN-300-GW-USB driver installation.

- ▶ Start XN300-Assist.
- ▶ Open the "?" menu and select the "Updating the Operating System..." option.
- ▶ Click on the arrow next to the "Operating system file..." button and select the new XN300_CANVxxxxxxx.bin firmware file.
- ▶ Use the drop-down menu for the COM port to select the port on your PC into which the mini USB cable is plugged.
- ▶ Start transferring the firmware by clicking on the "Transfer to device" button.



Once the transfer is successfully completed, you will get a corresponding message. The GW LED will show a solid green light.

▶ Now switch the gateway off, change the node ID to the value you want (1–31), and switch the gateway back on.

The gateway will start running with the new firmware.

For running the gateway in "station variants" mode

▶ Set the node ID for the gateway, e.g., to <2>.
▶ Check that the node ID has been entered in XSOFT-CODESYS Register ...
▶ Power cycle the gateway by turning it off and then back on.
▶ Select the gateway in the PLC configuration and, in the "General" tab, enable the "Autoconfig. PDO Mapping" option. This will enable default mapping.

→ TIP: A gateway without baud rate autodetection will result in faster initialization, as automatic baud rate detection will be disabled.

Setting a fixed baud rate

To turn off automatic baud rate detection and set a fixed baud rate instead, follow the steps below:

▶ Use DIP switches 6–8 to set the baud rate you want on the gateway, e.g., 125 kHz.
▶ In XSOFT-CODESYS, select the gateway and, in the "Baud rate" field in the "PLC settings" tab, select the same baud rate you just set, e.g., 125 kHz.

8.6 CAN objects for station variants

8.6.1 Activation StationVariants Mode (Object 0x4005)

Object 0x4005 must be set to 0x01 during the gateway's initialization phase. Otherwise, the lists for the full configuration and the station variant will be deleted from the gateway and the gateway will be restarted.

"Activation StationVariants Mode"=TRUE tells the gateway that it must not start transferring data immediately after switching to the "OPERATIONAL" CANopen status, and that it must instead wait for an enable signal from the enable process.

The object's data is automatically entered into the transmit PDOs (default mapping).

Feature	Description / Value	EDS
Type name	Activation StationVariants Mode	[4005] ParameterName=Activation StationVariants Mode ObjectType=0x7 DataType=0x0005 AccessType=rw PDOMapping=0 DefaultValue=0
Description	Activation StationVariants Mode	
Object Code	ARRAY	
Mapping	SDO	
	Default	
Data type	UNSIGNED8	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000_{hex}	

Design of the data byte:

Data bit	Description	Notes
0	0 = Start gateway with target configuration 1 = Switch gateway to "station variants" operating status	
1-7	reserved	

8.6.2 Checksum (internal) (Object 0x4006)

Object 0x4006 is addressed exclusively from the IEC library. The returned checksum provides clear feedback indicating which reduced assembly is active on the gateway.

Feature	Description / Value	EDS
Type name	INTERNAL_CS_MC	[4006] Parameter name=INTERNAL_CS_MC ObjectType=0x7 DataType=0x0007 AccessType=rw PDOMapping=0 DefaultValue=0
Description	NTERNAL_CS_MC	
Object Code	ARRAY	
Mapping	SDO	
	Default	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000_{hex}	

Design of the data bytes:

Sub index 1≦n≦254	Byte 0:							
	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

8.6.3 Module list (internal) (Object 0x4007)

Object 0x4007 is addressed exclusively from the IEC library. The module list is used to tell the gateway what the full configuration specified by the PLC is. The gateway then uses this information together with the bitmask for the nodes that are present in order to identify physically missing nodes and substitute them when dealing with the field bus.

Subindex 0 contains the number of XN300 slice modules in the full configuration, while subindexes 1 to 32 contain the module ID for the corresponding module in the full configuration.

Feature	Description / Value	EDS
Type name	INTERNAL_ML_MC	[4007] Parameter name=INTERNAL_ML_MC ObjectType=0x8 SubNumber=1 [4007sub0] Parameter name=INTERNAL_NM_MC ObjectType=0x7 DataType=0x0005 AccessType=rw [4007sub1..32] Parameter name=INTERNAL_NM_MC ObjectType=0x7 DataType=0x0006 AccessType=rw PDOMapping=0
Description	INTERNAL_ML_MC	
Object Code	ARRAY	
Mapping	SDO	
	Default	
Data type	UNSIGNED16	
Sub index	00 count of modules for full configuration	
	01 first module ID of full configuration	
	02 … FE_{hex} next module ID of full configuration	
Access	ro	
Default value	0000_{hex}	

8.6.4 Available Modules (internal) (Object 0x4008)

Object 0x4008 is addressed exclusively from the IEC library. While the PLC uses object 0x4007 to tell the gateway what the full configuration is, the gateway uses this information in order to identify physically missing nodes and substitute them when dealing with the field bus. Object 0x4008 represents a bitmask used to identify, by means of a double word, which of the individual XN300 slice modules in object 0x4007 are present and which are absent.

Feature	Description / Value	EDS
Type name	INTERNAL_BM_MC	[4008] Parameter name=INTERNAL_BM_MC ObjectType=0x7 DataType=0x0007 AccessType=rw PDOMapping=0 DefaultValue=0
Description	INTERNAL_BM_MC	
Object Code	ARRAY	
Mapping	SDO	
	Default	
Data type	UNSIGNED32	
Sub index	01 … FE_{hex}	
Access	rw	
Default value	0000_{hex}	

Design of the data bytes:

Data bit	Designation	Description
0… 31		0 = Module from full configuration not found in station variant 1 = Module from full configuration found in station variant

9 What Happens If …?

If the XN-312-GW-CAN is not behaving as expected, the following tips may help fix the problem. Only qualified persons should test electrical voltages while the device is in operation.

Problem	Solution
Unable to find the module under configuration in the EDS file.	You are using an old EDS file version. Load and use the the current EDS file, → Section “4.2 Project-specific EDS file”, page 50.
LED ST flickers red/green Communication with gateway not possible.	The gateway is in the middle of detecting the CAN baud rate. Check the wiring of the CAN bus. Make sure that the CAN Master is connected correctly. Set a fixed baud rate, → Section “1.5.2 Baud rate”, page 22.
No transmission of analog values (Index 0x6401)	Make sure that the analog values are enabled: Object Index 0x6423/ Sub0=1 (AnalogInputGlobalInputEnable), → Section “7.15.3 Analog Input Global Interrupt Enable (Object 0x6423)”, page 187
Gateway with enabled monitoring boots continuously	With enabled analog values, make sure that the load of the CAN bus doesn't exceed 70%. • Use the filter settings for analog modules to reduce the bus load (see description of specific product) • The filter settings for DS-401: For the analog value 0x6401/Sub x the delta values can be set at 0x6426/Sub x. • Use the possibility to provide a PDO with a sending delay (Inhibit-Time) or switch the PDOs transmission type to SYNCRON or RTR-only.
CAN master indicates a guarding error	

10 Appendix

10.1 Approvals and national approvals for XN300 system devices

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

Product standards	• IEC/EN → Section " Standards", page 373; • UL 508 (INDUSTRIAL CONTROL EQUIPMENT); • CE-mark
UL File No.	XN-312-…, XN-322--…: E135462, XN322-1DCD-B35: E172143
NA Certification	cULus
Degree of protection	IEC: IP20

10.2 Dimensions

		XN-312-GW-CAN
Dimensions (H × D × W)	mm	115 x 72 x 12.5
	inch	4.53 × 2.83 × 0.49
Space units (SU) width		
Weight	kg	0.080
	lb	0.18
mounting		DIN-rail IEC EN 60715, 35 mm
Mounting position		horizontal

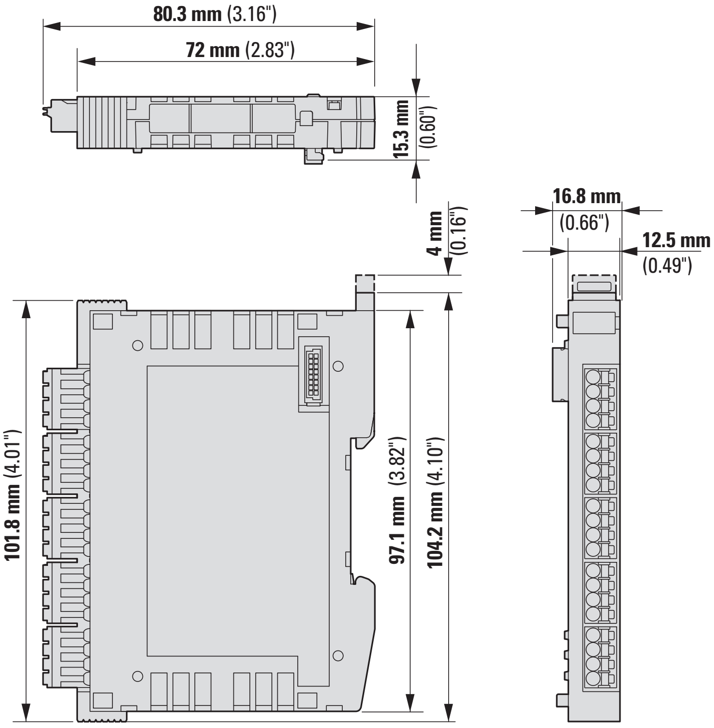


Figure 96: Dimensions XN-312-GW-CAN

10.3 Technical Data

CANopen gateway			XN-312-GW-CAN
General			
Standards			IEC/EN 61131-2
Dimensions (W x H x D) (without plug)		mm	115 x 72 x 12.5
Weight		kg	approx. 0.080
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			prevent with suitable measures
Storage		°C	-40…80
Relative humidity, non-condensing (IEC/EN 60068-2-30)		%	5…95
Ambient mechanical conditions			
Degree of protection, IEC/EN 60529			IP20
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	9
Drop to IEC/EN 60068-2-31		mm	50
Free fall, packaged (IEC/EN 60068-2-32)		m	0.3
Mounting position			horizontal
Supply voltage			
Rated operating voltage	U_e	V	24 DC
admissible range		V	19.2 - 30 DC
Residual ripple of input voltage		%	5
Protection against polarity reversal			Yes
Rated operational current	I_e	mA	100 max.
Heat dissipation at 24 V DC		W	2.4
Potential isolation between supply and 24/5 VDC bus voltage			No
Bridging voltage dips			
Duration of dip		ms	10
Repetition rate		s	1
Field bus interface CANopen			Field bus interface CANopen

CANopen gateway			XN-312-GW-CAN
Potential isolation between supply and field bus			Yes
Insulation test voltage	U_i	V AC	500
Measurement of the clearance and creepage distance			IEC/EN 61131-2
CANopen – Baud rates		kBps	10, 20, 50, 125, 250, 500, 1000
Electromagnetic compatibility (EMC)			
Overvoltage category			II
Pollution degree			2
Electrostatic discharge (IEC/EN 6113-2:2008, ESD)			
Air discharge (Level 3)		kV	8
Contact discharge (Level 2)		kV	4
Electromagnetic fields (IEC/EN 61131-2:2008)			1
(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
CANopen bus cables		kV	1
Surge (IEC/EN 61131-2:2008, Level 1)			
Supply		kV	0.5
CAN		kV	1
Radiated RFI (IEC/EN 61131-2:2008, level 3)		V	10

10.4 XN-322 slice modules

The following slice module models are available as part of the XN300 system and can be combined as necessary:

Table 41: List of available XN322- ... slice modules

Part no.	Description
Power supply modules	XN-322-4PS-20 XN-322-18PD-M XN-322-18PD-P
Digital I/O modules	XN-322-8DI-PD XN-322-16DI-PD XN-322-20DI-PD XN-322-20DI-PF XN-322-20DI-PCNT XN-322-20DI-ND XN-322-8DO-P05 XN-322-12DO-P17 XN-322-16DO-P05 XN-322-8DIO-PD05 XN-322-16DIO-PD05 XN-322-16DIO-PC05
Analog I/O modules	XN-322-4AI-PTNI XN-322-7AI-U2PT XN-322-8AI-I XN-322-10AI-TEKT XN-322-8AO-U2 XN-322-4AIO-U2 XN-322-8AIO-U2 XN-322-4AIO-I XN-322-8AIO-I
Technology Modules	XN-322-2DMS-WM XN-322-1DCD-B35 XN-322-1CNT-8DIO XN-322-2SSI XN-322-4DO-RNO

10.5 Firmware versions

The following XN-322-… slice modules, as well as the station variants, are supported only by firmware version 1.04 or higher and EDS file version 1.04 or higher. For up-to-date information on the firmware and the EDS files, please visit the Download Center → Page 11.

Table 42: List of available XN-322- ... slice modules

Part no.	Description
Digital I/O modules	XN-322-8DI-PD XN-322-16DI-PD XN-322-20DI-ND XN-322-8DO-P05 XN-322-8DIO-PD05 XN-322-16DIO-PD05 XN-322-16DIO-PC05
Analog I/O modules	XN-322-4AIO-U2 XN-322-4AIO-I XN-322-8AIO-I

10.6 Maximum number of process data objects

Table 43: Maximum number of process data objects

Gateway	RPDO	TPDO
XN-322-GW-CAN	16	16

Index

O

Eaton is dedicated to ensuring that reliable, efficient and safe power 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 or www.eaton.com .

Eaton addresses worldwide:
www.eaton.com/Worldwide Sites

E-Mail: info-bonn@eaton.com
TRCAutomation@eaton.com
Internet: www.eaton.eu/xn300
www.eaton.com/xn300

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

06/16 MN050003 EN ICPD-MOC

Powering Business Worldwide