CAN-EtherCAT/2

EtherCAT CAN FD or CAN CC Gateway

EtherCAT/2-FD
(C.2932.62)

Hardware Manual

For Products C.2932.62, C.2932.02

esd electronics gmbh
Vahrenwalder Str. 207 • 30165 Hannover • Germany
http://www.esd.eu
Phone: +49 (0) 511 3 72 98-0 • Fax: +49 (0) 511 3 72 98-68

Notes

The information in this document has been checked carefully and is considered to be entirely reliable. esd electronics makes no warranty of any kind regarding the material in this document and assumes no responsibility for any errors that may appear in this document. In particular, the descriptions and technical data specified in this document may not be constituted to be guaranteed product features in any legal sense.

esd electronics reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance, or design.

esd electronics gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

Tel.: +49-511-37298-0
Fax: +49-511-37298-68
E-Mail: info@esd.eu
Internet: www.esd.eu

This manual contains important information and instructions on safe and efficient handling of the CAN-EtherCAT/2. Carefully read this manual before commencing any work and follow the instructions.
The manual is a product component, please retain it for future use.

Links
esd electronics gmbh assumes no liability or guarantee for the content of Internet pages to which this document refers directly or indirectly. Visitors follow links to websites at their own risk and use them in accordance with the applicable terms of use of the respective websites.

Trademark Notices
CANopen® and CiA® are registered EU trademarks of CAN in Automation e.V.
Windows® is a registered trademark of Microsoft Corporation in the United States and other countries.
EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
All other trademarks, product names, company names or company logos used in this manual are reserved by their respective owners.

Document Information

Document file:	I:\Texte\Doku\MANUALS\CAN\CAN-EtherCAT2\CAN-EtherCAT-2_Manual_en_10.docx
Date of print:	2025-04-29
Document-type number:	DOC0800

Document History

The changes in the document listed below affect changes in the hardware as well as changes in the description of the facts, only.

Rev.	Chapter	Changes versus previous version	Date
1.0	-	First English version of the CAN-EtherCAT/2 manual	2025-04-29

Technical details are subject to change without further notice.

Classification of Warning Messages and Safety Instructions

This manual contains noticeable descriptions, warning messages and safety instructions, which you must follow to avoid personal injuries or death and property damage.

This is the safety alert symbol.
It is used to alert you to potential personal injury hazards. Obey all safety messages and instructions that follow this symbol to avoid possible injury or death.

DANGER, WARNING, CAUTION

Depending on the hazard level the signal words DANGER, WARNING or CAUTION are used to highlight safety instructions and warning messages. These messages may also include a warning relating to property damage.

DANGER
Danger statements indicate a hazardous situation which, if not avoided, will result in death or serious injury.

WARNING.
Warning statements indicate a hazardous situation that, if not avoided, could result in death or serious injury.

CAUTION
Caution statements indicate a hazardous situation that, if not avoided, could result in minor or moderate injury.

NOTICE

Notice statements are used to notify people on hazards that could result in things other than personal injury, like property damage.

NOTICE
This NOTICE statement indicates that the device contains components sensitive to electrostatic discharge.

NOTICE
This NOTICE statement contains the general mandatory sign and gives information that must be heeded and complied with for a safe use.

INFORMATION

INFORMATION
Notes to point out something important or useful.

Safety Instructions

- When working with the CAN-EtherCAT/2 follow the instructions below and read the manual carefully to protect yourself from injury and the CAN-EtherCAT/2 from damage.
- The assembly is classified as open equipment and must therefore be installed in a control cabinet that is designed for the specific environmental. The control cabinet should be made of metal to improve the electromagnetic immunity of the device. It should be equipped with a key locking mechanism to prevent any unauthorized access.
- Do not use damaged or defective cables to connect the CAN-EtherCAT/2 and follow the CAN wiring hints in chapter: "Correct Wiring of Galvanically Isolated CAN Networks".
- In case of damages to the device, which might affect safety, appropriate and immediate measures must be taken, that exclude an endangerment of persons and domestic animals and property.
- The galvanic isolation of the CAN-EtherCAT/2 has only functional tasks and is not a protection against hazardous electrical voltage.
- The CAN-EtherCAT/2 is a device of protection class III according to DIN EN 61140 and may only be operated on supply circuits that offer sufficient protection against dangerous voltages.
- External circuits connected to the ports of the CAN-EtherCAT/2 must be sufficiently protected against dangerous voltage.
- The user is responsible for compliance with the applicable national safety regulations.

- Do not open the housing of the CAN-EtherCAT/2 .
- The CAN-EtherCAT/2 must be securely installed before commissioning.
- The permitted operating position is specified as shown (Figure 2). Other operating positions are not allowed.
- Never let liquids get inside CAN-EtherCAT/2. Otherwise, electric shocks or short circuits may result.
- Protect the CAN-EtherCAT/2 from dust, moisture, and steam.
- Protect the CAN-EtherCAT/2 from shocks and vibrations.
- The CAN-EtherCAT/2 may become warm during normal use. Always allow adequate ventilation around the CAN-EtherCAT/2 and use care when handling
- Do not operate the CAN-EtherCAT/2 adjacent to heat sources and do not expose it to unnecessary thermal radiation. Ensure an ambient temperature as specified in the technical data.

DANGER

Hazardous Voltage - Risk of electric shock due to unintentional contact with uninsulated live parts with high voltages inside of the system into which the CAN-EtherCAT/2 is to be integrated.

→ All current circuits which are connected to the device have to be sufficiently protected against hazardous voltage, before you start with the installation.

→ Ensure the absence of voltage before starting any electrical work.

Qualified Personnel

This documentation is directed exclusively towards personnel qualified in control and automation engineering. The installation and commissioning of the product may only be carried out by qualified personnel, which is authorized to put devices, systems, and electric circuits into operation according to the applicable national standards of safety engineering.

Conformity

The CAN-EtherCAT/2 is an industrial product and meets the demands of the EU regulations and EMC standards printed in the conformity declaration at the end of this manual.

Warning: In a residential, commercial, or light industrial environment the CAN-EtherCAT/2 may cause radio interferences in which case the user may be required to take adequate measures.

Data Safety

This device is equipped with an Ethernet or other interface which is suitable to establish a connection to data networks. Depending on the software used on the device, these ports may allow attackers to compromise normal function, get illegal access or cause damage.
esd does not take responsibility for any damage caused by the device if operated at any networks. It is the responsibility of the device's user to take care that necessary safety precautions for the device's network port are in place.

Intended Use

The intended use of the CAN-EtherCAT/2 is operation as CAN-EtherCAT gateway.
The guarantee given by esd does not cover damages which result from improper use, usage not in accordance with regulations or disregard of safety instructions and warnings.

- The CAN-EtherCAT/2 is intended for installation in a control cabinet.
- The operation of the CAN-EtherCAT/2 in hazardous areas, or areas exposed to potentially explosive materials is not permitted.
- The operation of the CAN-EtherCAT/2 for medical purposes is prohibited.

Service Note

The CAN-EtherCAT/2 does not contain any parts that require maintenance by the user. The CAN-EtherCAT/2 does not require any manual configuration of the hardware. Unauthorized intervention in the device voids warranty claims

Disposal

Products marked with a crossed-out dustbin must not be disposed of with household waste. Devices which have become defective in the long run must be disposed in an appropriate way or must be returned to the manufacturer for proper disposal. Please, contribute to environmental protection.

Typographical Conventions

Throughout this manual the following typographical conventions are used to distinguish technical terms.

Convention	Example
File and path names	**`/dev/null`** or **`<stdio.h>`**
Function names	***open()***
Programming constants	`NULL`
Programming data types	`uint32_t`
Variable names	*Count*

Number Representation

All numbers in this document are base 10 unless designated otherwise. Hexadecimal numbers have a prefix of 0x, and binary numbers have a prefix of 0b. For example, 42 is represented as 0x2A in hexadecimal and 0b101010 in binary.

Table of Contents

List of Tables

List of Figures

Overview

1.1 About this Manual

This manual describes the hardware variants of the CAN-EtherCAT/2. The product is available in a variant with and a variant without CAN FD capability:

- EtherCAT CAN FD gateway: CAN-EtherCAT/2-FD
- EtherCAT CAN CC gateway: CAN-EtherCAT/2
  (successor of CAN-EtherCAT (C.2922.02))

The variants are collectively referred to as CAN-EtherCAT/2. Any specific differences between the CAN-EtherCAT/2 variants are explicitly noted in this manual. The available functions, options, and included components may vary depending on the selected CAN-EtherCAT/2 variant.

1.2 Description of CAN-EtherCAT/2

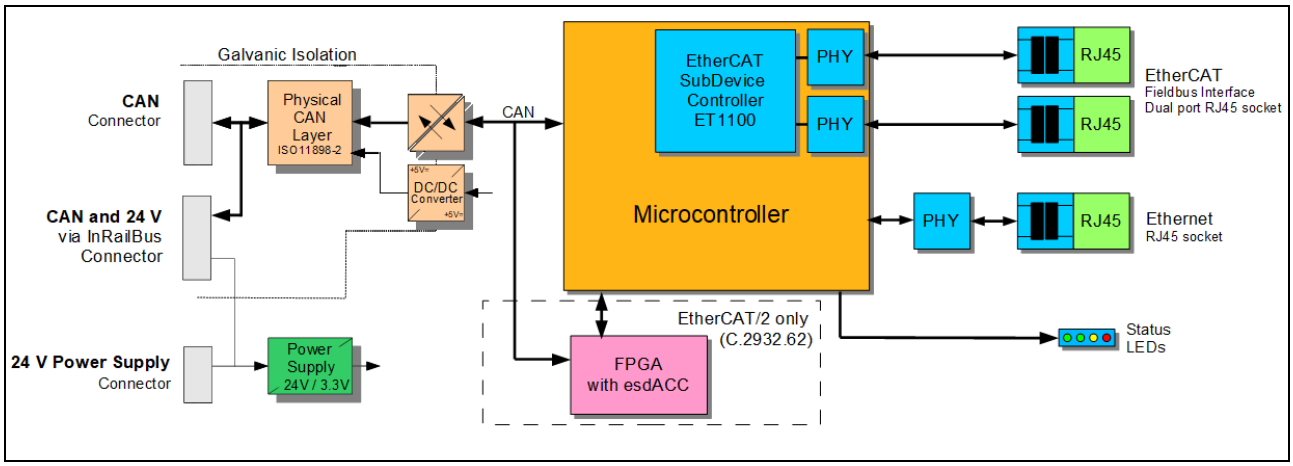


Figure 1: Block circuit diagram of CAN-EtherCAT/2(-FD)

The CAN-EtherCAT/2-FD gateway supports CAN FD implementations according to ISO 11898-1 and provides seamless connectivity between CAN FD modules and a real-time EtherCAT network. This capability enables the integration of CAN FD devices (including CAN CC) into an EtherCAT-based automation system and ensures real-time communication and interoperability across different networks. The CAN CC variant CAN-EtherCAT/2 (C.2932.02) is designed as replacement of CAN-EtherCAT (C.2922.02) and supports CAN CC implementations only.

Compliance with the EtherCAT Standard

The gateway operates with EtherCAT SubDevice functionality that is fully compliant with the ETG.5001 (CAN Interface – Module Profile No. 5000 (1)), ensuring standardized and efficient integration into EtherCAT networks. The firmware of the device can be updated via FoE.
The CAN-EtherCAT/2 can be easily configured via the EtherCAT MainDevice and standard EtherCAT configuration tools (e.g. esd EtherCAT Workbench). The configuration of the gateway is based on CAN over EtherCAT (CoE) protocol and process data exchange via EtherCAT PDOs.

Ethernet-over-EtherCAT (EoE) Support

Additionally, the CAN-EtherCAT/2 can be configured as an Ethernet switch port using the Ethernet-over-EtherCAT (EoE) protocol. This enables the extension of the EtherCAT network to include standard Ethernet devices, enhancing the network's flexibility and capability.

Ports
The CAN CC port of CAN-EtherCAT/2 is compliant with ISO 11898-2 and it supports bit rates from 10 kbit/s*[1] up to 1 Mbit/s. CAN-EtherCAT/2-FD comes with a CAN FD port and supports bit rates up to 8 Mbit/s.
The 100BASE-TX EtherCAT port is IEEE802.3 compatible and runs at 100 MBit/s.
The CAN port, as well as the EtherCAT port, are galvanically isolated.

1.3 Glossary

1.3.1 Abbreviations

Abbreviation	Term	Description
API	Application Programming Interface	
CAN	Controller Area Network	In this manual the term CAN only includes CAN CC and CAN FD. CAN XL is not supported
CAN CC	CAN classic	
CAN FD	CAN flexible data rate	
CPU	Central Processing Unit	
CiA	CAN in Automation	
ESC	EtherCAT SubDevice (legacy: Slave) Controller	
ECS	EtherCAT SubDevice (legacy: Slave)	
EUT	Equipment under Test	
HW	Hardware	
IDC	Insulation Displacement Connector	
I/O	Input/Output	
LSB	Least Significant Bit	
MDevice	Main Device	
MSB	Most Significant Bit	
n.a.	not applicable	
OS	Operating System	
QSPI	Quad serial peripheral interface	
SDK	Software Development Kit	
SoC	System-on-a-Chip	
SubDevice	SubordinateDevice	
USP	Unique Selling Point	

1.3.2 Inclusive Language

The term ***MainDevice*** (abbreviated ***MDevice***) replaces the term "Master" and ***SubordinateDevice*** (abbreviated ***SubDevice***) replaces the term "Slave". Despite that, API calls, data structures, macros, etc. which names contain the “old” terms are not renamed, because this would have the technical effect breaking (backward) compatibility. The latter is in no way intended to undermine the replacement of these terms or offend anyone.

1.4 References

(1) EtherCAT Technology Group, Modular Device Profile Part 3: MDP Fieldbus Gateway Profile Specifications, ETG.5001.3 V0.1.2

1 Limited by transceiver dominant timeout, typ. 1.25ms.

2 Hardware Installation

2.1 Connection Diagram

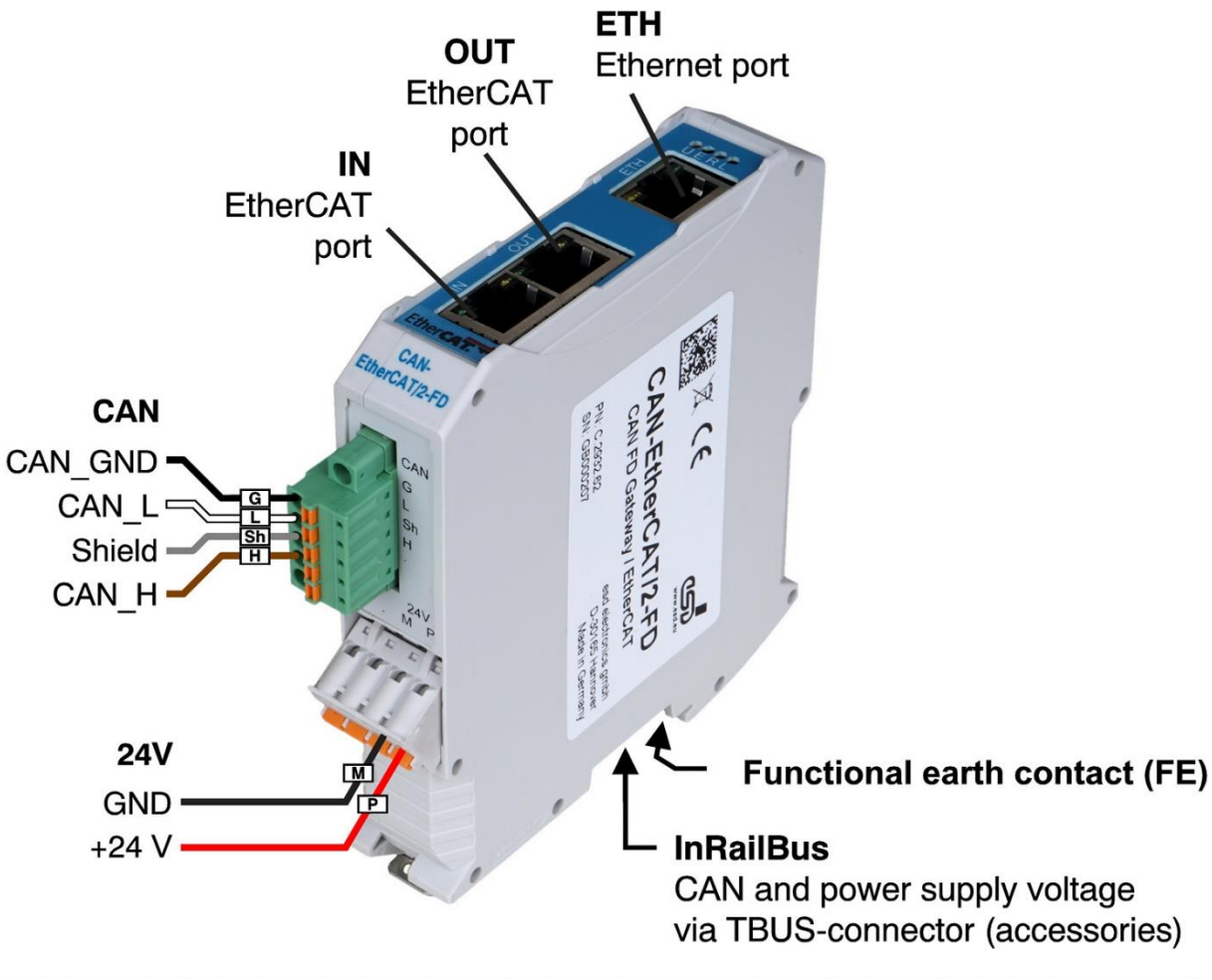


Figure 2: Connecting diagram of CAN-EtherCAT/2

See also page from page 72 for signal assignment of the CAN connectors.
For conductor connection and conductor cross section see page 77

NOTICE
Read chapter “Installing and Uninstalling Hardware” from page 17, before you start with the installation of the hardware!

3 LEDs

3.1 Position of the LEDs

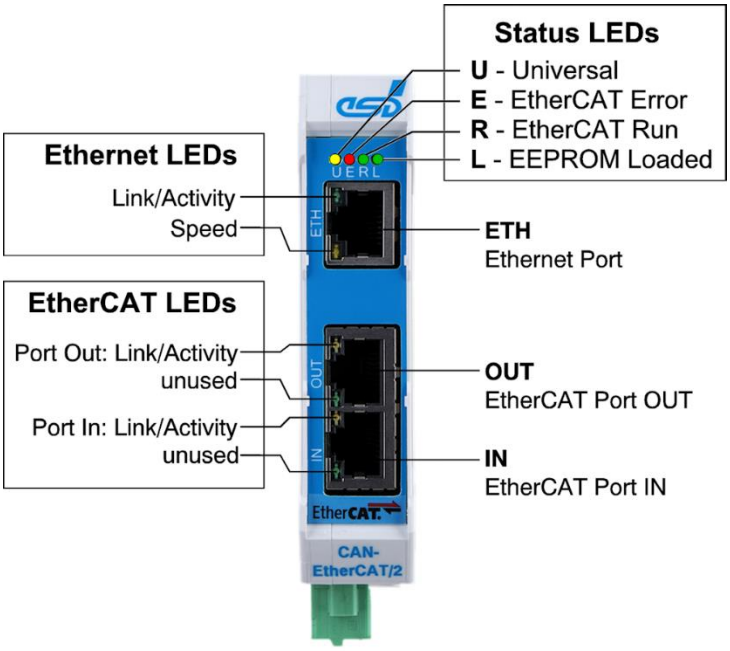


Figure 3: Connectors and LEDs

3.2 LED Indication

LEDs at Ethernet RJ45 connector *ETH:*

LED	Color	LED Status	Description
Link/ Activity	green or yellow depending on the socket used	off	No Ethernet link
		blinking	Ethernet link established, Ethernet activity (reception of Ethernet data packages)
		on	Ethernet link established, but no Ethernet activity
Speed	green or yellow depending on the socket used	off	10 MBit/s
		on	100 MBit/s

Table 1: Ethernet-LED functionality

LEDs at EtherCAT-RJ45 Connectors *IN* and *OUT*

LED	Color	LED Status	Description
Link/ Activity	green or yellow depending on the socket used	off	No EtherCAT link
		blinking	EtherCAT link established, EtherCAT activity (reception of EtherCAT data packages)
		on	EtherCAT link established, but no activity
Spare	yellow or green depending on the socket used	-	unused

Table 2: EtherCAT LED functionality (integrated in RJ45)

EtherCAT-LEDs *U, E, R, L*

LED Status	Description
blinking	LED repeats: 200 ms on, 200 ms off.
flicker	LED repeats: 50 ms on, 50 ms off.
single flash	LED repeats: 200 ms on, 1000 ms off.
double flash	LED repeats: 200 ms on, 200 ms off, 200 ms on, 1000 ms off.

Table 3: LED states (according to ETG.1300 documents)

LED	Color	Function	LED Status	Description
U	yellow	Universal	off	No information available
			flicker	FoE firmware transfer/update is in progress
			on	FoE firmware transfer finished. Visible only until the device is brought to INIT state - then the actual firmware update is started.
			any	Set by CoE object 0x2000.2, see 6.1.4.1
E	red	EtherCAT ERROR	off	No error
			blinking	State change failed
			single flash	State changed due to configuration error
			double flash	SM watchdog time out
R	green	EtherCAT RUN *	off	Init phase
			blinking	Pre-Operational phase
			single flash	Safe-Operational phase
			on	Operational phase
			flicker	Bootstrap phase
L	green	EEPROM Loaded *	off	Unable to retrieve ET1100 configuration from EEPROM
			on	Successful retrieval of ET1100 configuration from EEPROM

*Directly connected to the ET1100

Table 4: EtherCAT LED functionality

3.3 Side Views with Labels

Figure 4: CAN-EtherCAT/2 product label
(Example CAN CC variant)

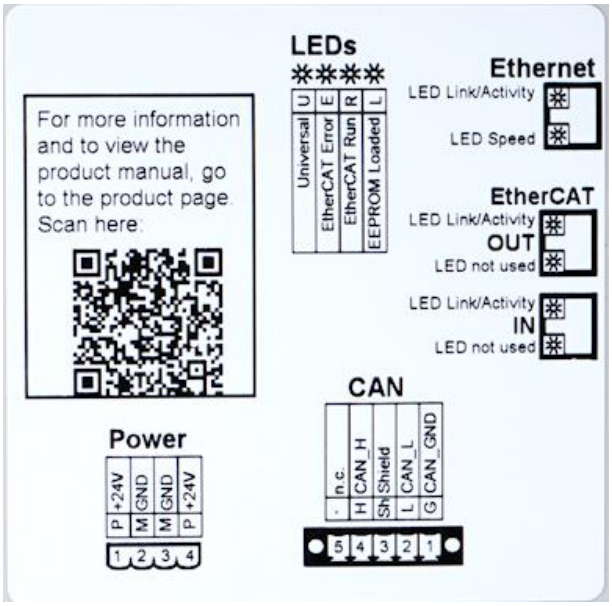


Figure 5: CAN-EtherCAT/2 connector label
(Example both variants)

The product label (Figure 4) shows among others the name, esd order No. (PN) and the serial number (SN).

Module Variant	CAN-EtherCAT/2-FD	CAN-EtherCAT/2 (Example Figure 4)
Name:	CAN-EtherCAT/2-FD	CAN-EtherCAT/2
PN (esd order No.):	C.2932.62	C.2932.02
SN (Serial number):	Individual number of the module	Individual number of the module e.g.: GB000213

Table 5: Description of the labels

The connector label (Figure 5) shows a description of the Connectors and LEDs.
Also refer to chapters “Connector Assignments” from page 72 and “LEDs” from page 14.

4 Installing and Uninstalling Hardware

To install or uninstall the CAN-EtherCAT/2, please follow the installation notes.

Step	Procedure	See Page
	**NOTICE** **Read the safety instructions at the beginning of this document carefully before you start with the hardware installation/!**	5
	**DANGER** Hazardous voltage - Risk of electric shock due to unintentional contact with uninsulated live parts with high voltages inside of the system into which the CAN-EtherCAT/2 is to be integrated. → The CAN-EtherCAT/2 is a device of protection class III according to DIN EN 61140 and may only be operated on supply circuits that offer sufficient protection against dangerous voltages. → External circuits connected to the ports of the CAN-EtherCAT/2 must be sufficiently protected against dangerous voltages. → Compliance with the applicable national safety regulations is the responsibility of the user. → Ensure the absence of voltage before starting any electrical work. → The plug connectors must not be plugged in or unplugged under voltage or load!	
To install, continue as described in chapter 4.1 ‘Installing the Hardware’. To uninstall, continue as described in chapter 4.2 ‘Uninstalling the Hardware’		

4.1 Installing the Hardware

Step	Procedure	See Page
1.	Follow the safety instructions at the beginning of chapter 4	17
2.	Mount the CAN-EtherCAT/2 module in the control cabinet. Connect the ports (power supply voltage, CAN, EtherCAT and Ethernet).	13
	See also chapter 9 for Connector Assignments	72
	**NOTICE** Incorrect wiring of the 24V power supply voltage can cause damage to the module! → Make sure to connect the cables correctly to the 24V cable connector! → Only use suitable cables for the plug. → If applicable, follow the instructions for using the InRailBus.	72
3.	Please note that the CAN bus must be terminated at both ends! esd offers special T-connectors and termination connectors for external termination. Additionally, the CAN_GND signal must be connected to earth at exactly one point in the CAN network. For details, please read chapter “Correct Wiring of Galvanically Isolated CAN Networks”.	78
4.	Before you switch on the supply voltage, check that all plug connectors are correctly seated. Switch on the 24 V-power supply voltage of the CAN-EtherCAT/2	-

5.	Copy the enclosed EtherCAT SubDevice information file (ESI) into the corresponding folder.	
6.	Configure the CAN-EtherCAT/2 gateway with an EtherCAT configurator.	

4.2 Uninstalling the Hardware

Step	Procedure	See Page
1.	Follow the safety instructions at the beginning of chapter 4	17
2.	Make sure that all connected interfaces and power supply are switched off.	
3.	Disconnect the CAN-EtherCAT/2 from the connected interfaces.	
4.	Use a screwdriver to pull the movable snap-on foot of the CAN-EtherCAT/2 downwards while swivelling the module upwards until it comes loose.	91
5.	Carefully take the CAN-EtherCAT/2 out.	

5 Configuration with an EtherCAT Configurator

5.1 CAN-EtherCAT Gateway Application Example

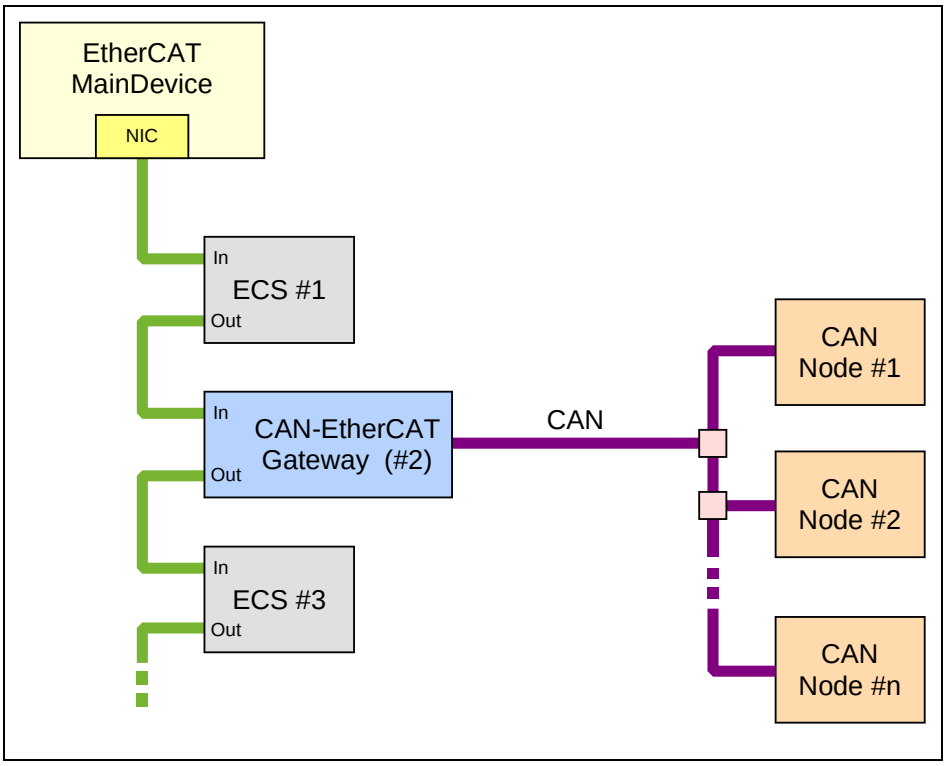


Figure 6: Basic function

The CAN-EtherCAT/2 gateway can take any position in an EtherCAT network.

5.2 Configuration Sequence, esd EtherCAT Workbench

The following chapter describes the configuration of the CAN port of the CAN-EtherCAT/2 gateway for example by means of the esd EtherCAT Workbench.
First, the EtherCAT SubDevice information file (ESI) must be copied. The file can be copied from the EtherCAT-CD which is included in delivery of the CAN-EtherCAT/2.
You can also download the ESI file from the CAN-EtherCAT/2 product page on our website: https://esd.eu/

Copy the following files to the corresponding folder:

- **ESD_CAN-EtherCAT2.xml**
- **CAN-EtherCAT2_OD.xml**
- **CAN-FD-EtherCAT2_OD.xml**

When the Workbench is running, click on the menu entry *Copy ESI file(s) to SubDevice library* (Under *Tools*) to copy the file, see Figure 7. Otherwise, the Workbench's start menu entry *Open SubDevice library folder* can be used to copy the file manually.

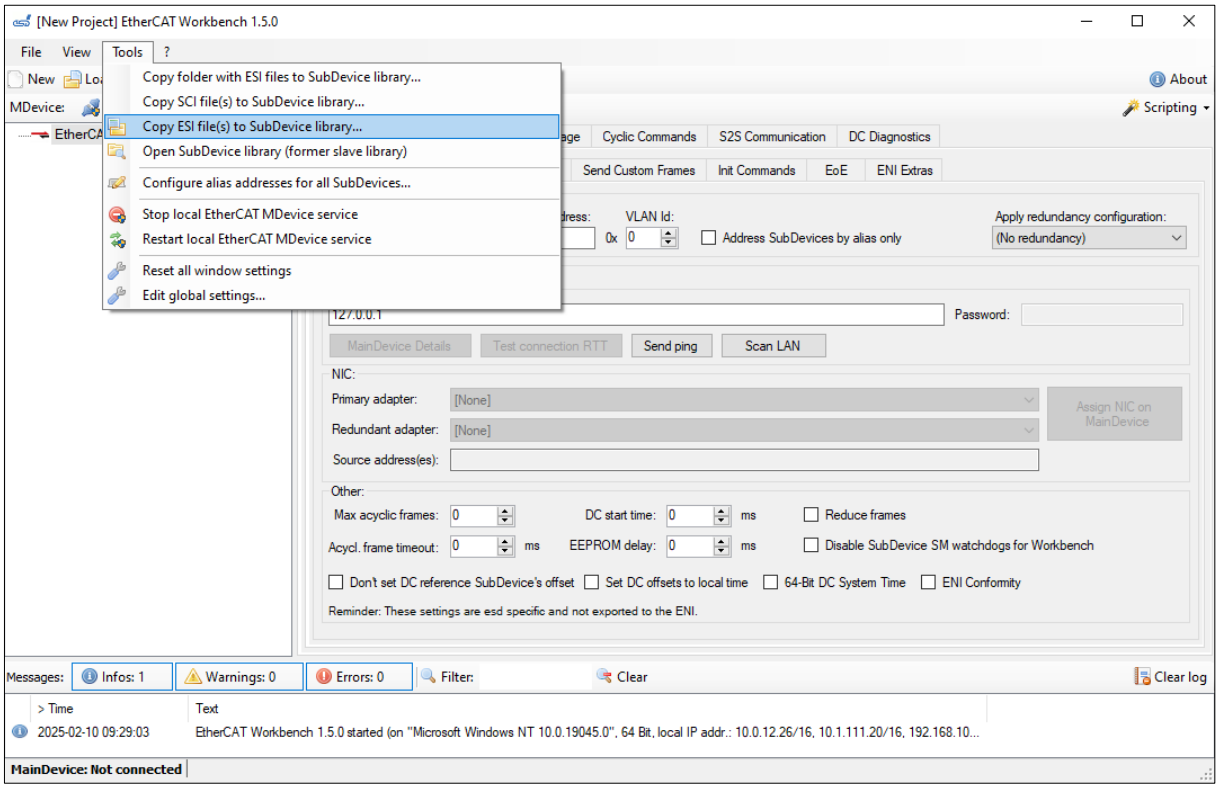


Figure 7: Installing ESI file with the Workbench

Now the Workbench must be (re)started and a network scan will show the device:

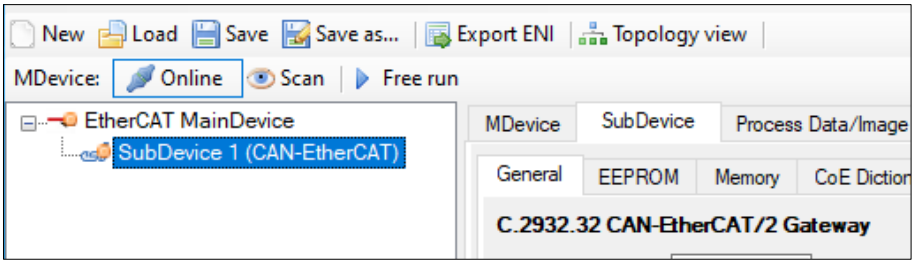


Figure 8: Tree view of SubDevices (detail)

Now go to the *CoE Dictionary* tab page and recreate the dictionary by the menu item *Recreate dict.*, *Online from SubDevice* as shown in Figure 9:

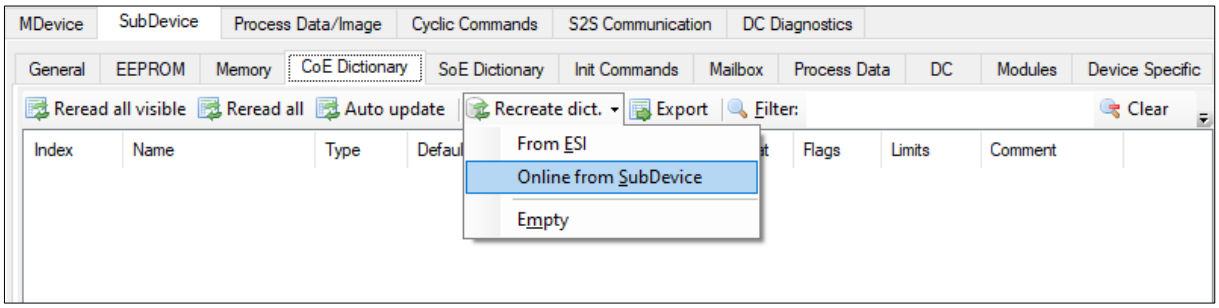


Figure 9: Recreating the CoE dictionary (detail)

Click *Reread all* to update the items, then select the *Process Data* tab page.

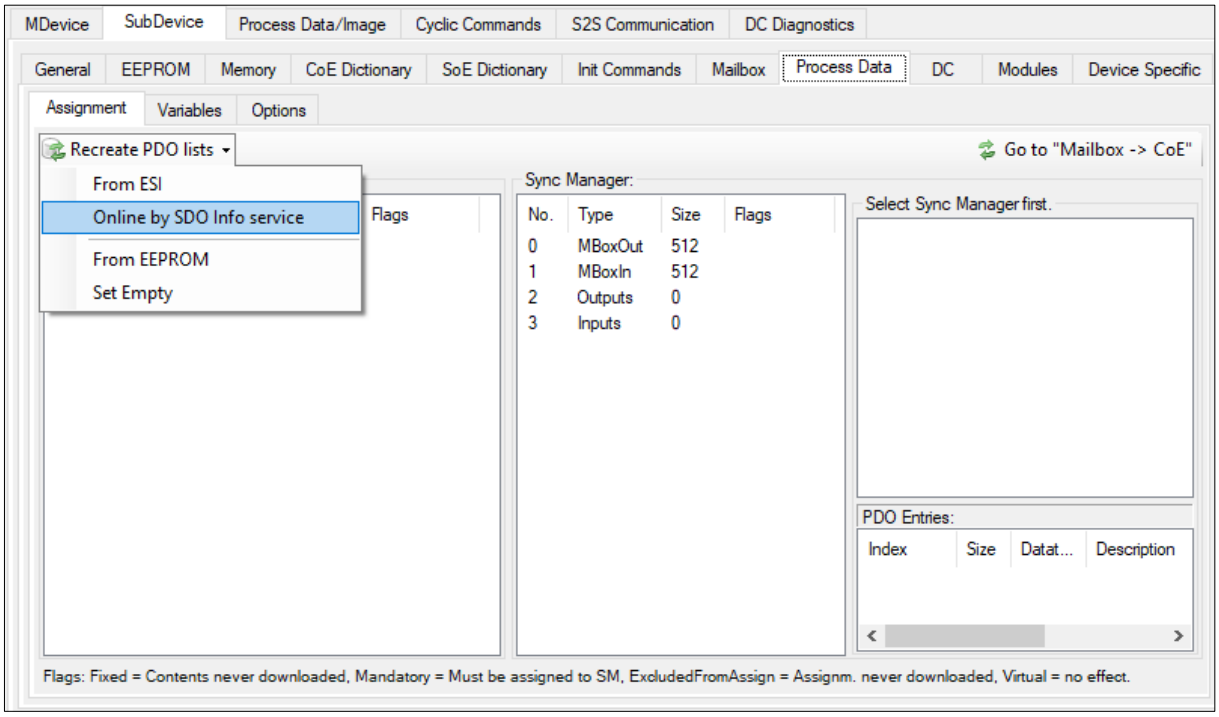


Figure 10: Recreating the list of available PDOs

Recreate list of available PDOs by SDO Info service as shown in Figure 10. As a result, the PDOs 0x1600 and 0x1a00 should be mapped:

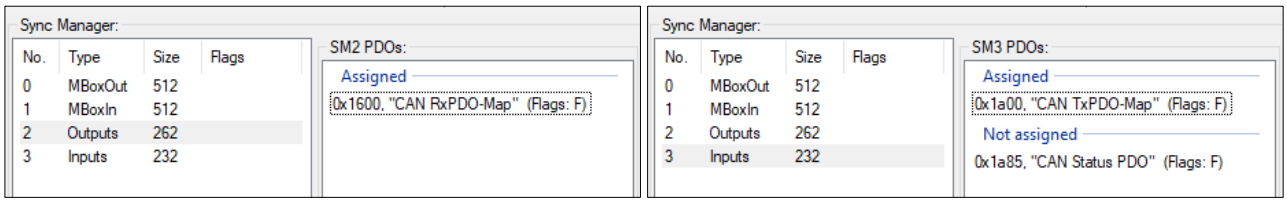


Figure 11: PDO Selection (Outputs / Inputs)

PDO 0x1600 contains the CAN Tx messages, as described in 6.1.3.6, 0x1a00 contains the Rx messages as described in 6.1.3.7. (The optional PDO 0x1a85 contains CAN Status information, see 6.1.3.8 and 6.1.5.11)

Different queue sizes or 29-bit CAN IDs
If you want to use extended CAN IDs or change Rx- or Tx- queue size, you must configure this in the CoE dictionary, right after you clicked “Reread all” as described above, but it must be done **before** you recreate the list of available PDOs.
To use 29-bit IDs for example, you must write object 0x8000.20 as described in 6.1.5.6. When the list of available PDOs is recreated afterwards you'll notice a PDO size change. (With the standard queue sizes for example, the 198 bytes for the outputs will change to 262 bytes.)

When changing the queue sizes or CAN ID type this must be done during SubDevice start up, too. Section 5.2.1 shows how this is done for the CAN bit rate object – this works for other objects as well.

5.2.1 Setting the Bit Rate during SubDevice Initialization

Go to the *SubDevice*, *Mailbox*, *CoE* tab page, right click in the *Init Commands* list and select *Append new item*:

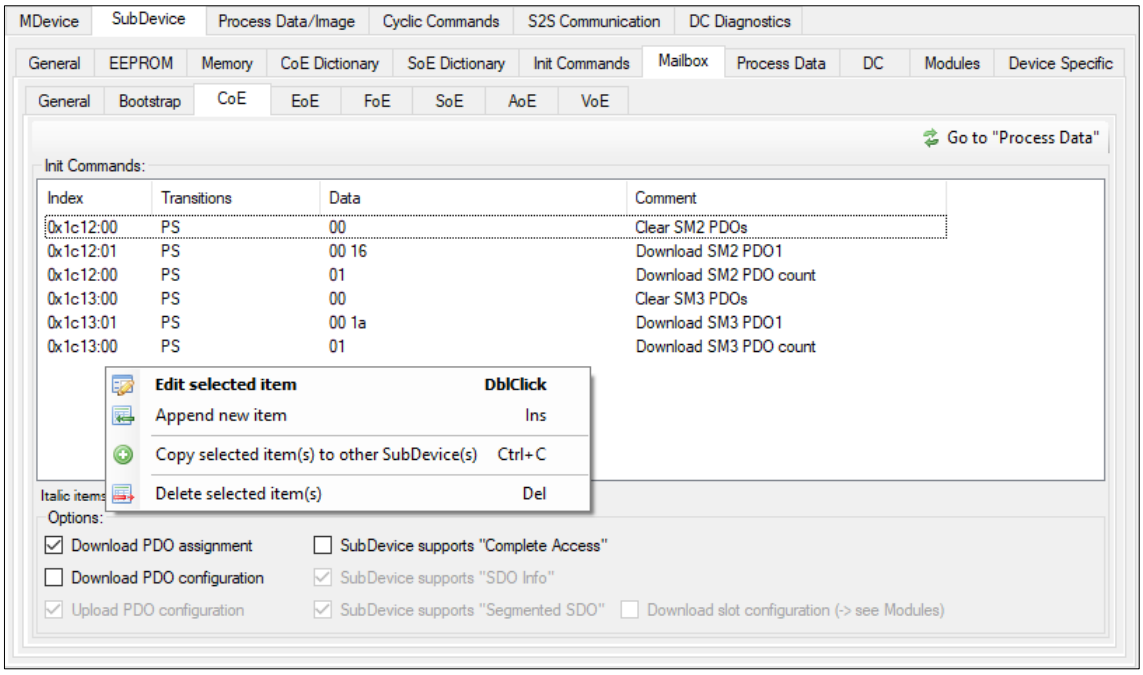


Figure 12: Appending CoE init command

In the following dialog window click *Select from objectdict.* and select the bit rate object (0xF800:02) in the context menu that appears. Now the dialog should look like this:

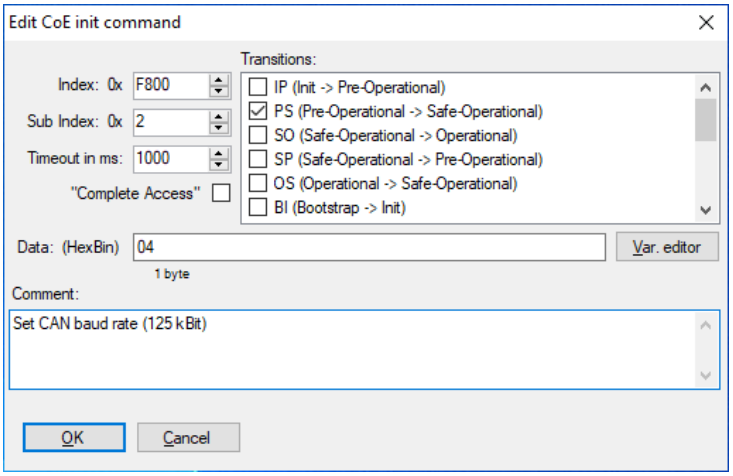


Figure 13: CoE init. command for bit rate object

Just enter the desired bit rate index (described in 6.1.5.12) at the *Data:* input box and leave the dialog with the *OK* button. (Other settings should be left untouched)
Now the *Init Commands* list contains an additional command that sets the bit rate during the SubDevice's *PreOp → SafeOp* transition.

This can be done for other objects, especially the *CAN Interface Configuration* objects (0x8000, see 6.1.5.6), too. In the example shown in Figure 14 other init commands (queue type, bit rate, filter table and identifier area) are selected.

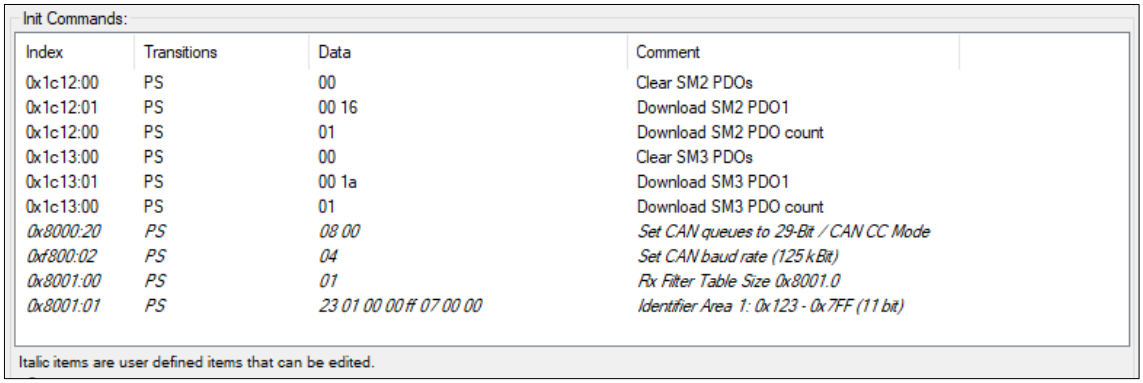


Figure 14: Other Init Commands (example)

Please refer to the EtherCAT Workbench Manual, chapter *CAN Interface* under *EtherCAT Network Configuration - > Device Specific* for further information about the tool to create the Init commands.

5.2.2 Export ENI

To export the ENI for the EtherCAT network and CAN-EtherCAT/2 you just configured click *Export MDevice configuration file (ENI)* in the *File* menu: (Or press “Ctrl+M” or click on the *Export ENI* tab in the Workbench's toolbar – they all do the same)

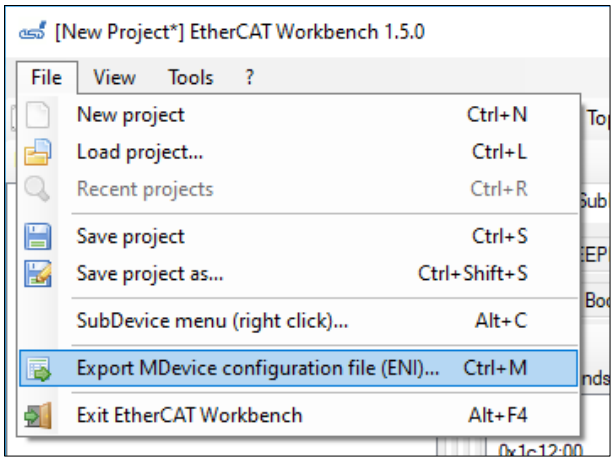


Figure 15: Exporting the ENI

5.3 Configuration Sequence, Beckhoff Configurator

The following chapter describes the configuration of the CAN port of the CAN-EtherCAT/2 gateway for example by means of the Beckhoff EtherCAT configurator.

First, the following enclosed EtherCAT SubDevice information files (ESI) must be copied to the corresponding folder:

- **ESD_CAN-EtherCAT2.xml**
- **CAN-EtherCAT2_OD.xml**
- **CAN-FD-EtherCAT2_OD.xml**

Using the EtherCAT configurator the folder may be, for example:
"`C:\Program Files\EtherCAT Configurator\EtherCAT`".

As soon as the EtherCAT configurator has recognized the CAN-EtherCAT/2, it will display it in the device tree view:

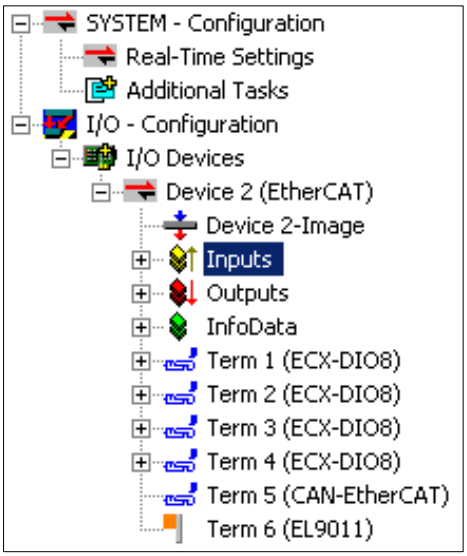


Figure 16: CAN-EtherCAT/2 in device tree view

Use the tab *CoE-Online* to display the object dictionary:

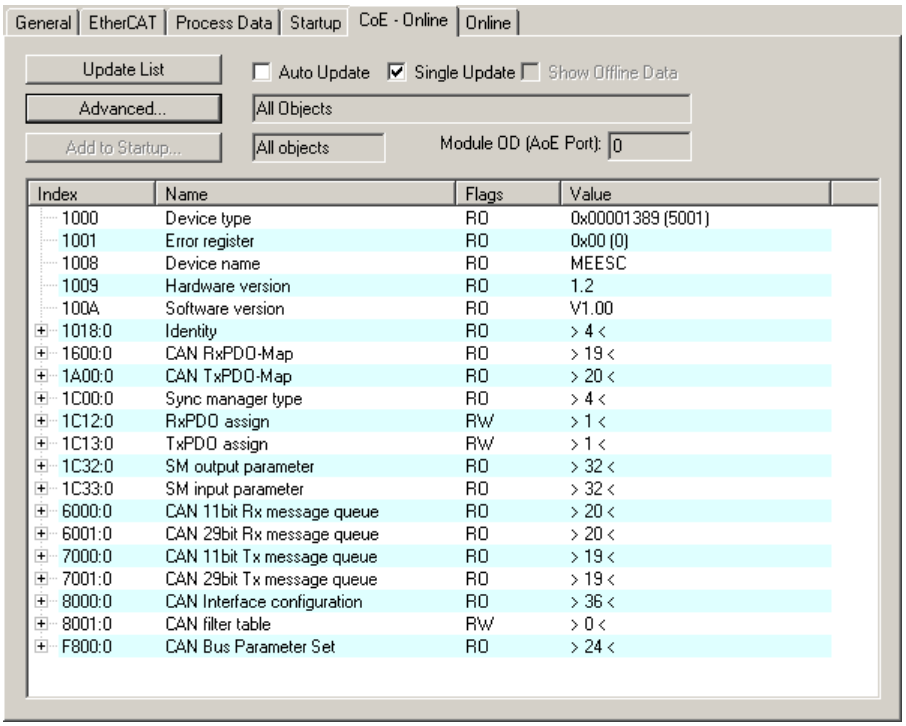


Figure 17: Object dictionary

The *Process Data* section will be initially empty. Click the *Load PDO info from device* button to read the data:

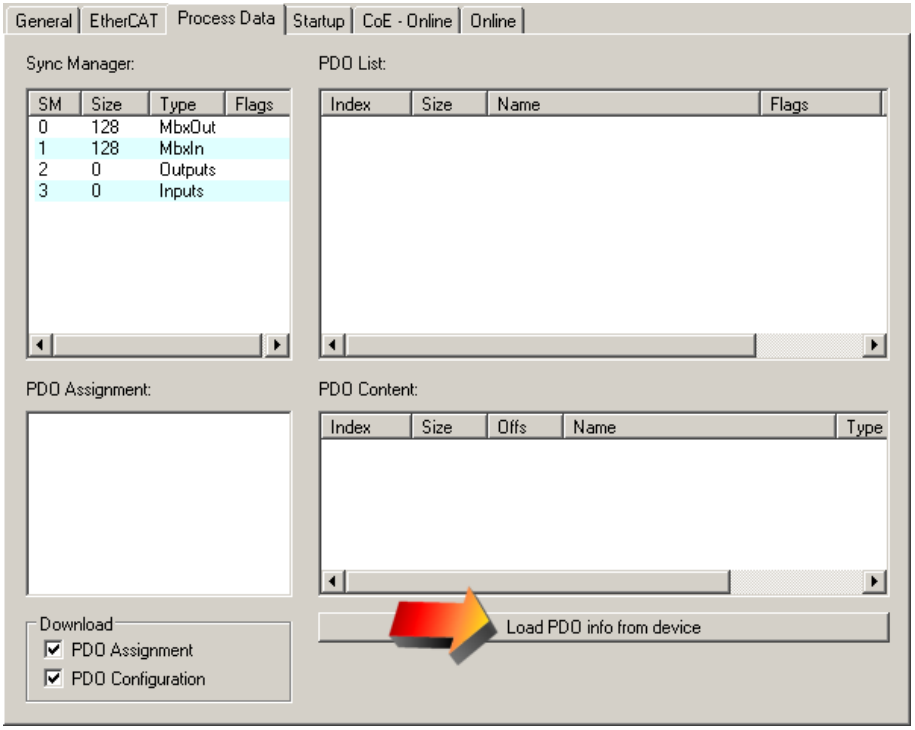


Figure 18: Loading process data

The configurator will now display the process data:

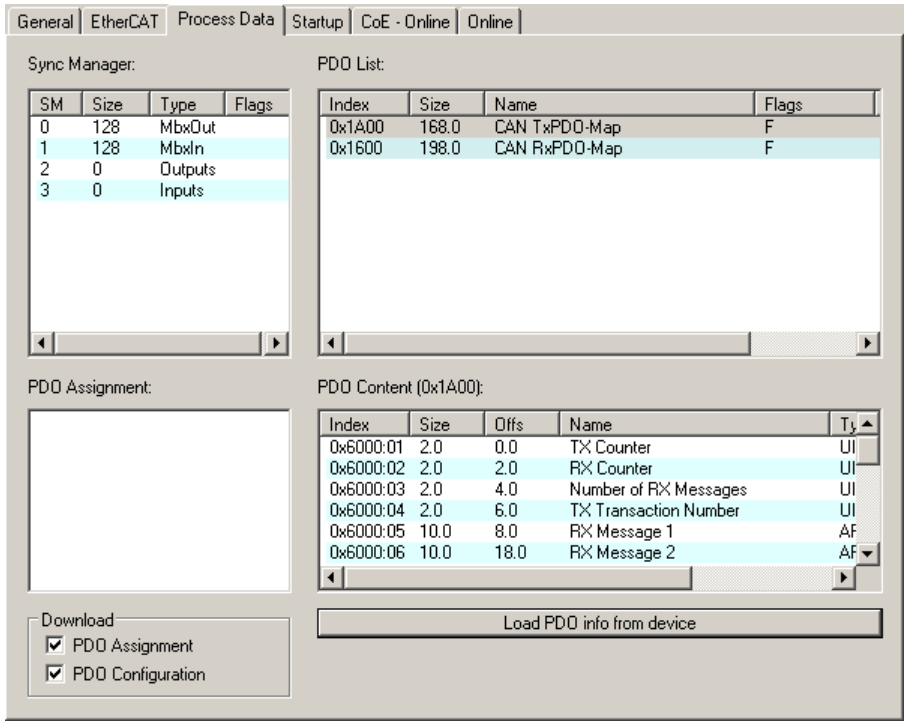


Figure 19: Process data display

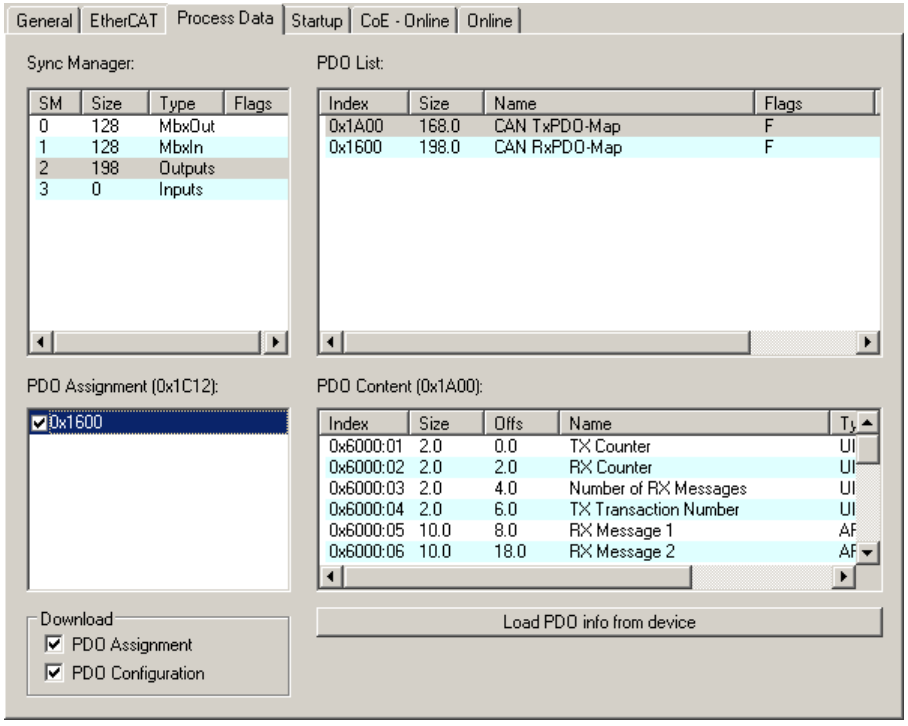


Figure 20: Process data (output PDOs chosen)

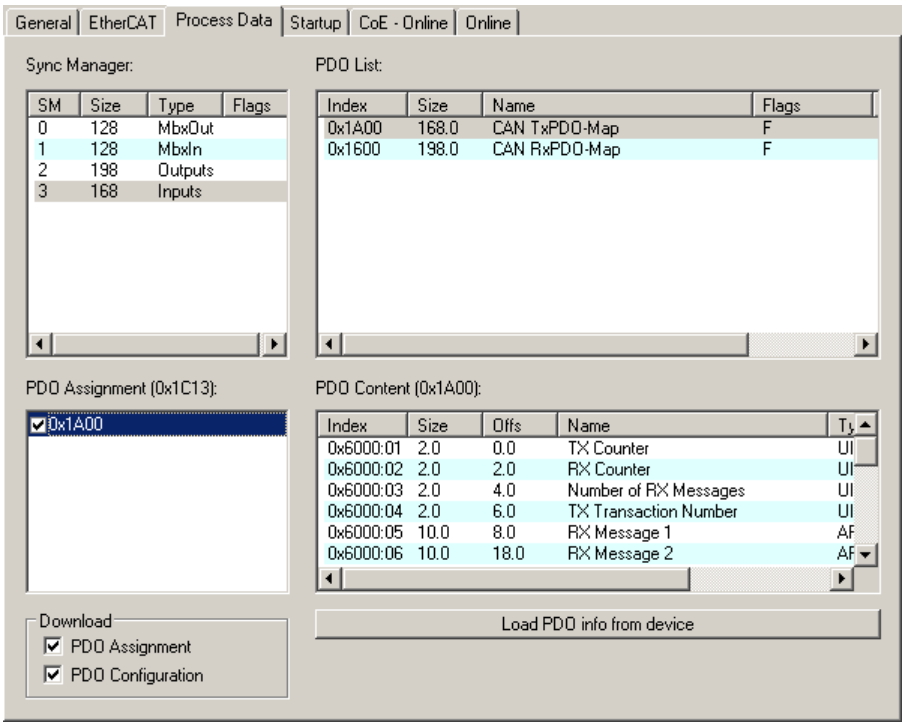


Figure 21: Process data (input PDOs chosen)

The CAN-EtherCAT/2 gateway will only go active on the CAN bus after the bit rate has been set (see chapter 6.1.5.12 from page 58). Consequently, it makes sense to set the bit rate right now.

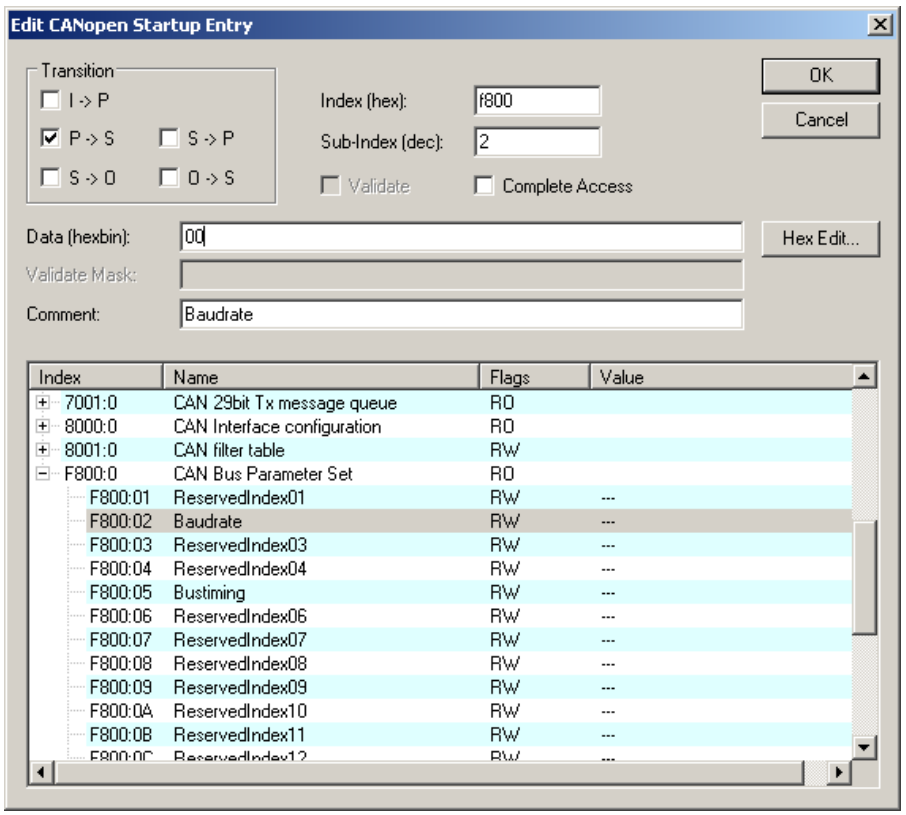


Figure 22: Bit rate setting during startup sequence

The final startup sequence may look like shown in the following example:

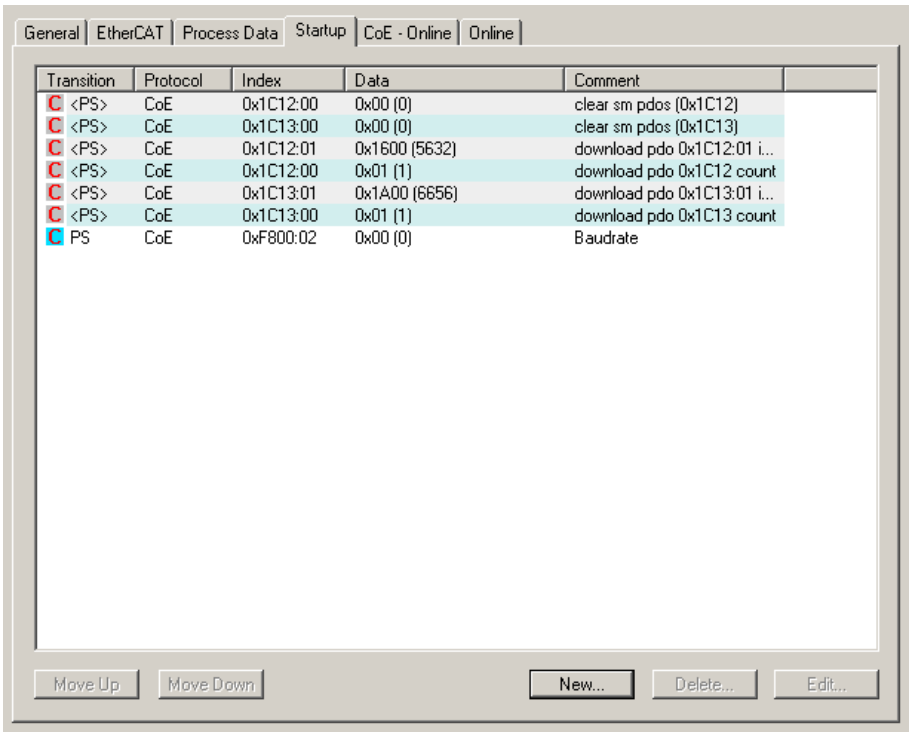


Figure 23: Startup - sequence example

Following the startup sequence, the EtherCAT network can be set active by calling the *Reload Devices* function by clicking *Main Menu/Actions*:

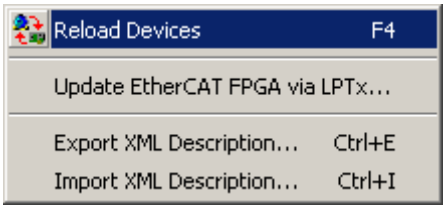


Figure 24: Reload Devices

The process image of the CAN port looks as follows:

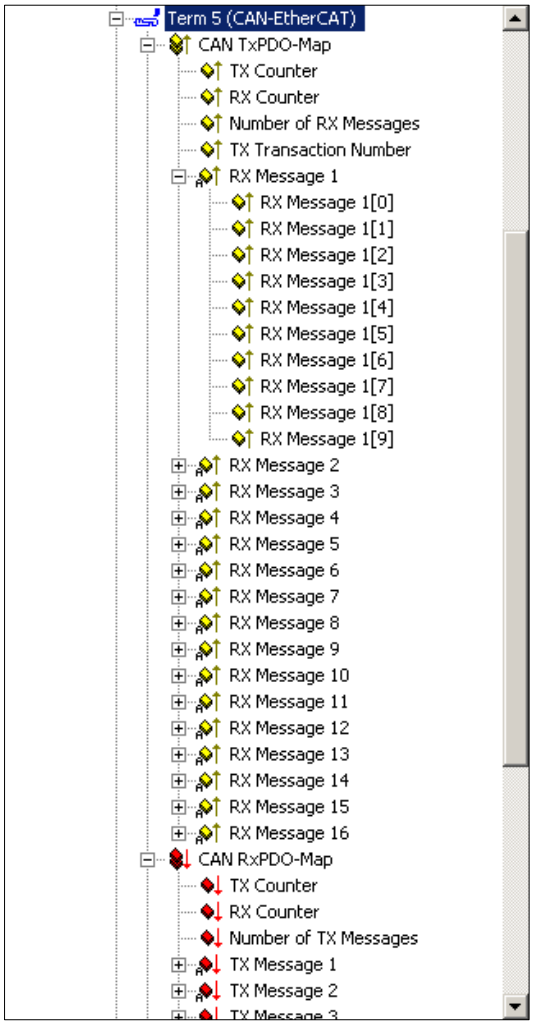


Figure 25:CAN port process image

5.3.1 Exporting EtherCAT Network Information (ENI)

To export the configuration file for an EtherCAT MDevice choose *Export Configuration File…*:

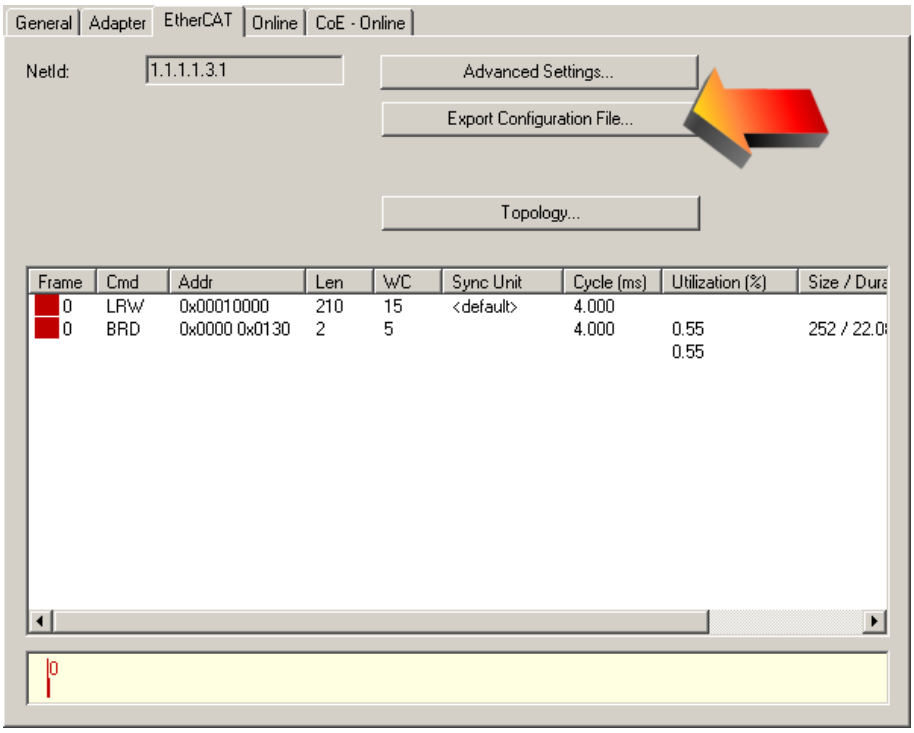


Figure 26:Exporting configuration file

6 EtherCAT Communication

6.1 CAN Port

The CAN port is based on a modular device profile (1) and it supports one CAN module. This module includes one CAN Tx message queue in the output area and one CAN Rx message queue in the input area.

6.1.1 Object Dictionary Structure

The object dictionary is composed of the following areas:

Index	Object Dictionary Areas
0x0000 ... 0x0FFF	Data Type Area
0x1000 ... 0x1FFF	Communication Area
0x2000 ... 0x5FFF	Vendor Specific Area
0x6000 ... 0x6FFF	Input Area (CAN Rx message queue)
0x7000 ... 0x7FFF	Output Area (CAN Tx message queue)
0x8000 ... 0x8FFF	Configuration Area (CAN interface configuration)
0xF000 ... 0xFFFF	Device Area

Table 6: Object dictionary structure

The following explains the definition of a standard and an extended CAN message queue. For CAN-EtherCAT/2-FD there is also a CAN FD message queue. For proper operation one of the CAN message queues must be chosen. This can be accomplished by writing the CAN interface settings object (0x8000). The RPDO and TPDO mapping objects (0x1600 and 0x1A00) will change accordingly.

6.1.1.1 Output Data

The CAN port output data includes the Tx message queue plus the control data for the Rx and TX message queues. The CAN port output data is always required.

6.1.1.2 Input Data

The CAN port input data includes the Rx message queue plus the status information for the Rx and Tx message queues. The CAN port input data is always required.

6.1.2 Object Dictionary

The CAN-EtherCAT/2 gateway layer 2 implementation supports the following objects:

Index	Name
0x1000	Device type
0x1008	Device name
0x1009	Hardware version
0x100A	Software version
0x1018	Identity object
0x1400	1. Receive PDO-Parameter
0x1600	1. Receive PDO-Mapping
0x1800	1. Transmit PDO-Parameter
0x1A00	1. Transmit PDO-Mappping CAN interface
0x1A85	CAN status PDO
0x1C00	Sync manager type
0x1C12	RPDO assign
0x1C13	TPDO assign
0x1C32	Output Sync Manager Parameter (Free Run Mode)
0x1C33	Input Sync Manager Parameter (Free Run Mode)
0x2000	Other settings
0x2010	Statistics
0x6000	CAN interface input (11-bit identifier)
0x6001	CAN interface input (29-bit identifier)
0x6002	CAN-EtherCAT/2-FD only: CAN interface input (CAN FD)
0x7000	CAN interface output (11-bit identifier)
0x7001	CAN interface output (29-bit identifier)
0x7002	CAN-EtherCAT/2-FD only: CAN interface output (CAN FD)
0x8000	CAN interface configuration
0x8001	CAN Rx filter table
0xF000	Modular device profile
0xF100	Link status
0xF108	CAN status
0xF800	CAN bus parameter set
0xFA01	Rx/Tx ethernet telegrams

Table 7: Implemented Objects

6.1.3 Standard Objects (0x1000 ... 0x1FFF)

6.1.3.1 Object 0x1000 Device Type

Index	Sub-Index	Description	Data Type	RW	Default
0x1000	0	*Device Type*	UINT32	RO	0x13881389

Variable Description

Device type of the EtherCAT SubDevice:
The low word contains the used CoE profile (5001). The high word contains the module profile according to the modular device profile (1)

6.1.3.2 Object 0x1008 Device Name

Index	Sub-Index	Description	Data Type	RW	Default
0x1008	0	*Device Name*	STRING	RO	“ECSGW-CAN”

Variable Description

Device name of the EtherCAT SubDevice.

6.1.3.3 Object 0x1009 Hardware Version

Index	Sub-Index	Description	Data Type	RW	Default
0x1009	0	*Hardware Version*	STRING	RO	

Variable Description

CAN-EtherCAT/2 gateway hardware version.

6.1.3.4 Object 0x100A Software Version

Index	Sub-Index	Description	Data Type	RW	Default
0x100A	0	*Software Version*	STRING	RO	

Variable Description

CAN-EtherCAT/2 gateway software version.

6.1.3.5 Object 0x1018 Identity

Index	Sub-Index	Description	Data Type	RW	Default
0x1018	0	*Number of sub-indices*	UINT8	RO	4
	1	*Vendor ID*	UINT32	RO	0x17 (23)
	2	*Product code*	UINT32	RO	2
	3	*Revision*	UINT32	RO	See below
	4	*Serial number*	UINT32	RO	See below

Variable Description

CAN-EtherCAT/2 gateway identification characteristics.

Vendor ID esd vendor-ID = 23

Product code CAN-EtherCAT/2 product code = 2

Revision CAN-EtherCAT/2 ESI revision number
Corresponds to the SubDevice revision number stored in its EEPROM ESI – used to determine which .xml version of the ESI file the configuration tool (e.g. the esd EtherCAT Workbench) shall use.

Serial number Serial number

6.1.3.6 Object 0x1600 RPDO-Map CAN-Interface

This object defines the CAN interface mapping into the EtherCAT input data.

The first three sub-indices contain the size of the Tx and Rx counters plus the number of Tx messages. The size of the CAN Rx message queue is configured using object 0x8000.

Object 0x8000 is also used to define the CAN message ID mode, either 11-bit (Object 0x7000), 29-bit (0x7001) or CAN FD (0x7002). Depending on the settings the contents of objects 0x7000, 0x7001 and 0x7002 are mapped in object 0x1600.

Object 0x1600 is always required and must be defined in the PDO Assign Object 0x1C12, sub-index 1.

Index	Sub-Index	Description	Data Type	RW	Default
0x1600	0	*Number of CAN-Messages+3*	UINT8	RO	
	1	*1. PDO Mapping entry* (object 0x700z (CAN interface output), entry 0x01 (Tx Counter))	UINT32	RO	
	2	*2. PDO Mapping entry* (object 0x700z (CAN interface output), entry 0x02 (Rx Counter))	UINT32	RO	
	3	*3. PDO Mapping entry* (object 0x700z (CAN interface output), entry 0x03 (Number of Tx Messages))	UINT32	RO	
	4	*4. PDO Mapping entry* (object 0x700z (CAN interface output), entry 0x04 (Tx Message 1))	UINT32	RO	
	...	..	...	...	
	m	*m. PDO Mapping entry* (object 0x700z (CAN interface output), entry m (Tx Message m-3))	UINT32	RO	

6.1.3.7 Object 0x1A00 TPDO-Map CAN-Interface

This object defines the CAN interface mapping into the EtherCAT output data.

The first three sub-indices contain the size of the Tx and Rx counters plus the number of Tx messages. The size of the CAN Tx message queue is configured using object 0x8000.

Object 0x8000 is also used to define the CAN message ID mode, either 11-bit (object 0x7000), 29 Bit (object 0x7001) or CAN FD (object 0x7002). Depending on the settings the contents of objects 0x6000, 0x6001 and 0x6002 are mapped in object 0x1A00.

Object 0x1A00 is always required and must be defined in the PDO Assign Object 0x1C13, sub-index 1.

Index	Sub-Index	Description	Data Type	RW	Default
0x1A00	0	Number of *CAN-Messages+4*	UINT8	RO	
	1	*1. PDO Mapping entry* (object 0x6000 (CAN interface input), entry 0x01 (Tx Counter))	UINT32	RO	
	2	*2. PDO Mapping entry* (object 0x6000 (CAN interface input), entry 0x02 (Rx Counter))	UINT32	RO	
	3	*3. PDO Mapping entry* (object 0x6000 (CAN interface input), entry 0x03 (Number of Rx Messages))	UINT32	RO	
	4	*4. PDO Mapping entry* (object 0x6000 (CAN interface input), entry 0x04 (Tx Transaction Number))	UINT32	RO	
	5	*5. PDO Mapping entry* (object 0x6000 (CAN interface input), entry 0x05 (Rx Message 1))	UINT32	RO	
	...	..		...	
	m	*m. PDO Mapping entry* (object 0x6000 (CAN interface input), entry m (Rx Message m-4))	UINT32	RO	

6.1.3.8 Object 0x1A85 CAN Status PDO

This object allows to map the CAN Status entries from object 0xF108. See 6.1.5.11 for details about the mapped entries.

Index	Sub-Index	Description	Data Type	RW	Default
0x1A85	0	*Max Subitem*	UINT8	RO	0xE (14)
	1	Object 0xF108 sub-index 0x01	UINT32	RO	
	2	Object 0xF108 sub-index 0x02	UINT32	RO	
	3	Object 0xF108 sub-index 0x03	UINT32	RO	
	4	*Padding (1 Bit)*	UINT32	RO	
	5	Object 0xF108 sub-index 0x05	UINT32	RO	
	6	Object 0xF108 sub-index 0x06	UINT32	RO	
	7	*Padding (10 Bit)*	UINT32	RO	
	8	Object 0xF108 sub-index 0x11	UINT32	RO	
	9	Object 0xF108 sub-index 0x12	UINT32	RO	
	10	Object 0xF108 sub-index 0x13	UINT32	RO	
	11	Object 0xF108 sub-index 0x14	UINT32	RO	
	12	*Padding (12 Bit)*	UINT32	RO	
	13	Object 0xF108 sub-index 0x21	UINT32	RO	
	14	Object 0xF108 sub-index 0x22	UINT32	RO	

6.1.3.9 Object 0x1C00 Sync Manager Type

Index	Sub-Index	Description	Data Type	RW	Default
0x1C00	0	*Number of sub-indices*	UINT8	RO	4
	1	*Sync-Manager Type Channel 1: Mailbox Write*	UINT8	RO	1
	2	*Sync-Manager Type Channel 2: Mailbox Read*	UINT8	RO	2
	3	*Sync-Manager Type Channel 3: Process Data Write (Outputs)*	UINT8	RO	3
	4	*Sync-Manager Type Channel 4: Process Data Read (Inputs)*	UINT8	RO	4

Parameter Description

Sync-Manager Type:

Sync-Manager Type Channel 1: Mailbox Write

Sync-Manager Type Channel 2: Mailbox Read

Sync-Manager Type Channel 3: Process Data Write (Outputs)

Sync-Manager Type Channel 4: Process Data Read (Inputs)

6.1.3.10 Object 0x1C12 RPDO-Assign

Object 0x1C12 assigns the mapping of the CAN interface RPDOs.

Index	Sub-Index	Description	Data Type	RW	Default
0x1C12	0	*Number of sub-indices*	UINT8	RW	1
	1	*CAN-Interface RPDO*	UINT16	RW	0x1600

6.1.3.11 Object 0x1C13 TPDO-Assign

Object 0x1C13 assigns the mapping of the CAN interface TPDOs.

Index	Sub-Index	Description	Data Type	RW	Default
0x1C13	0	*Number of sub-indices*	UINT8	RW	1
	1	*CAN-Interface TPDO*	UINT16	RW	0x1A00

6.1.3.12 Object 0x1C32 Output Sync Manager Parameter (Free Run Mode)

The following table describes only the defined sub-indices of object 0x1C32.

Index	Sub-Index	Description	Data Type	RW	Default
0x1C32	1	*Synchronization_type*	UINT16	R	0x0001
	4	*Synchronization_type_supported*		R	0x0002
	5	*Minimum_cycle_time*	-	R	0x3D090 (250000)
	11	*SM-event_missed*	-	-	0
	12	*Cycle_time_too_small*	-	-	0
	32	*Sync_error*	-	-	0

Used Output Sync Manager Parameters (Free Run Mode):
Parameter Description

Synchronization_type	0x0001: SM Synchronous - synchronized with SM 2 Event
Synchronization_type_supported	Bit 1: SM Synchronous supported
Minimum_cycle_time	Minimum cycle time = 250000 ns (0x3D090)

At the time of writing, Sub-index 11,12 and 32 are not fully implemented and will always read 0 (*SM-event_missed, Cycle_time_too_small, Sync_error)*

6.1.3.13 Object 0x1C33 Input Sync Manager Parameter (Free Run Mode)

The following table describes only the defined sub-indices of object 0x1C33.

Index	Sub-Index	Description	Data Type	RW	Default
0x1C33	1	*Synchronization_type*	UINT16	R	0x0022 (34)
	4	*Synchronization_type_supported*		R	0x0002
	5	*Minimum_cycle_time*	-	R	0x3D090 (250000)
	11	*SM-event_missed*	-	-	0
	12	*Cycle_time_too_small*	-	-	0
	32	*Sync_error*	-	-	0

Used Input Sync Manager Parameters (Free Run Mode):
Parameter Description

Synchronization_type	0x0022: SM-Synchronous with SM2 Event
Synchronization_type_supported	Same value as 0x1C32, sub-index 4
Minimum_cycle_time	Same value as 0x1C32, sub-index 5 (Minimum cycle time = 250000 ns (0x3D090))

At the time of writing, Sub-index 11,12 and 32 are not fully implemented and will always read 0 (*SM-event_missed, Cycle_time_too_small, Sync_error)*

6.1.4 Manufacturer Specific Objects (0x2000 – 0x5FFF)

6.1.4.1 Object 0x2000 Other Settings

NOTICE
The following CoE object (Index, Sub-Index) can only be written in *Pre-Operational* state:
Index: 0x2000, Sub-Index: 1

Index	Sub-Index	Name	Data Type	RW	Default
0x2000	0	*Number of sub-indices*	UINT8	RO	2
	1	*EoE IP Port local mode*	BOOL	RW	0
	2	*Custom LED state*	UINT8	RW	0

Parameter Description

EoE IP Port local mode This mode is not supported.
Sub-index 1 still exists but must remain '0'.

Custom LED state Used to overwrite the state of the "Universal" LED (section 3.2)
Values / State:
0: Off
1..4: Flash x1..Flash x4
13: Blink
14: Flicker
15: On
Reading this value reflects only the value that was last written – not the actual LED state. The state set by writing this object is overwritten whenever the CAN-EtherCAT/2 itself sets a state for the LED (i.e. the LED is usually turned off when the EtherCAT State changes)

6.1.4.2 Object 0x2010 Statistics

Index	Sub-Index	Name	Data Type	RW	Default
0x2010	0	*Number of sub-indices*	UINT8	RO	0x21 (33)
	1	*Reset*	UINT32	RW	
	2	*Cyclic handler time (min.)*	UINT32	RO	
	3	*Cyclic handler time (max.)*	UINT32	RO	
	4	*Cyclic handler time (avg.)*	UINT32	RO	
	5	*CAN handler time (min.)*	UINT32	RO	
	6	*CAN handler time (max.)*	UINT32	RO	
	7	*CAN handler time (avg.)*	UINT32	RO	
	8	*Watchdog triggered*	UINT32	RO	
	0x10	*EoE Frames EtherCAT Rx*	UINT32	RO	
	0x11	*EoE Frames EtherCAT Tx*	UINT32	RO	
	0x12	*EoE Frames EtherCAT Tx Error*	UINT32	RO	
	0x13	*EoE Frames EtherCAT Tx Overrun*	UINT32	RO	
	0x14	*EoE Frames Local Rx*	UINT32	RO	
	0x15	*EoE Frames Local Tx*	UINT32	RO	
	0x16	*EoE Frames Local Tx Error*	UINT32	RO	
	0x20	*App. CPU Usage (User)*	UINT8	RO	
	0x21	*App. CPU Usage (System)*	UINT8	RO	
	0x30	*CAN Frames TX Requested*	UINT32	RO	
	0x31	*CAN Frames TX*	UINT32	RO	
	0x32	*CAN Frames RX*	UINT32	RO	
	0x34	*CANDriver Controller overrun*	UINT32	RO	
	0x35	*CANDriver FIFO overrun*	UINT32	RO	
	0x36	*CANDriver Error Frames*	UINT32	RO	
	0x37	*CANDriver Aborted Frames*	UINT32	RO	
	0x38	*CANDriver RX Frames*	UINT32	RO	
	0x39	*CANDriver RX RTR Frames*	UINT32	RO	
	0x3A	*CANDriver RX Frames Ext.*	UINT32	RO	
	0x3B	*CANDriver RX RTR Frames Ext.*	UINT32	RO	
	0x3C	*CANDriver TX Frames*	UINT32	RO	
	0x3D	*CANDriver TX RTR Frames*	UINT32	RO	
	0x3E	*CANDriver TX Frames Ext.*	UINT32	RO	
	0x3F	*CANDriver TX RTR Frames Ext.*	UINT32	RO	

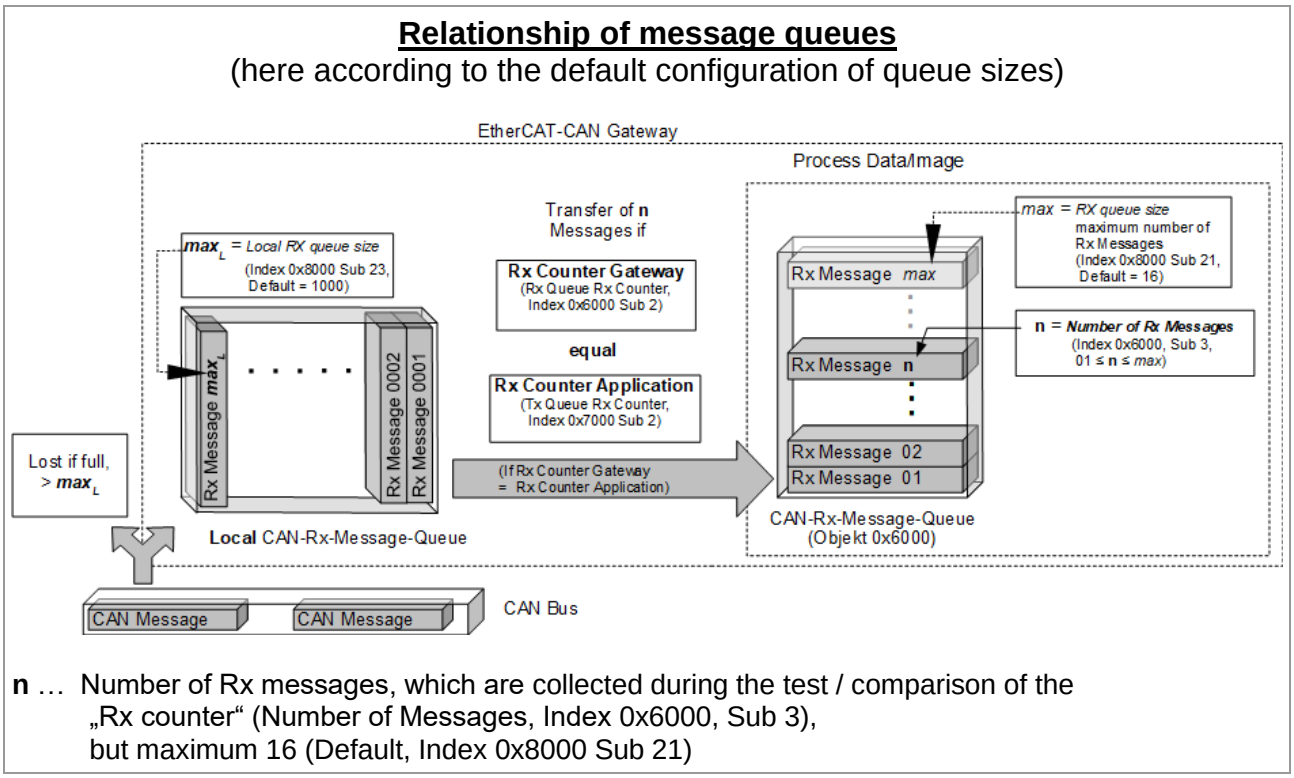
Parameter Description

Reset	When this object is written the statistics are reset. (Reading this value shows the time stamp of the last reset – milliseconds since device start up)
Cyclic handler time (min.)	For debugging purposes only. (Minimum time in application's cyclic handler within its last 10000 calls, in us)
Cyclic handler time (max.)	For debugging purposes only. (Maximum time in application's cyclic handler within its last 10000 calls, in us)
Cyclic handler time (avg.)	For debugging purposes only. (Average time in application's cyclic handler within its last 10000 calls, in us)
CAN handler time (min.)	For debugging purposes only. (Minimum time in application's CAN handler within its last 1000 calls, in us)
CAN handler time (max.)	For debugging purposes only. (Maximum time in application's CAN handler within its last 1000 calls, in us)
CAN handler time (avg.)	For debugging purposes only. (Average time in application's CAN handler within its last 1000 calls, in us)
Watchdog triggered	Times application watchdog was triggered due to missing process data (Outputs). (Watchdog value is calculated by ESC registers 0x0400 and 0x0420)
EoE Frames EtherCAT Rx	No. of Ethernet frames received from EtherCAT
EoE Frames EtherCAT Tx	No. of Ethernet frames sent to EtherCAT
EoE Frames EtherCAT Tx Error	No. of Ethernet frames that could not be sent to EtherCAT due to an error
EoE Frames EtherCAT Tx Overrun	No. of Ethernet frames that could not be sent to EtherCAT due to Tx buffer overrun (This kind of frame loss is likely to happen – higher level protocols on Ethernet side, such as TCP/IP, will handle this)
EoE Frames Local Rx	No. of Ethernet frames received on local Ethernet port
EoE Frames Local Tx	No. of Ethernet frames sent to local Ethernet port
EoE Frames Local Tx Error	No. of Ethernet frames that could not be sent to the local Ethernet port
App. CPU Usage (User/System)	For debugging purposes only. (Application's average CPU usage since last reading one of these two items, in percent)
CAN Frames TX Requested	Total No. of CAN frames that should have been sent – according to write accesses to “Tx Counter” objects. etc.
CAN Frames TX	Number of CAN frames that were successfully forwarded to the CAN driver
CAN Frames RX	Total No. of CAN frame successfully copied to the RX objects
CANDriver Controller overrun	CAN Driver statistics: No. of CAN Controller overruns
CANDriver FIFO overrun	CAN Driver statistics: No. of FIFO overruns
CANDriver Error Frames	CAN Driver statistics: No. of error frames
CANDriver Aborted Frames	CAN Driver statistics: No. of aborted frames
CANDriver RX Frames	CAN Driver statistics: No. of standard frames received
CANDriver RX RTR Frames	CAN Driver statistics: No. of standard RTR frames received
CANDriver RX Frames Ext.	CAN Driver statistics: No. of extended frames received
CANDriver RX RTR Frames Ext.	CAN Driver statistics: No. of extended RTR frames received
CANDriver TX Frames	CAN Driver statistics: No. of standard frames sent
CANDriver TX RTR Frames	CAN Driver statistics: No. of standard RTR frames sent
CANDriver TX Frames Ext.	CAN Driver statistics: No. of extended frames sent
CANDriver TX RTR Frames Ext.	CAN Driver statistics: No. of extended RTR frames sent

6.1.5 Profile Specific Objects (0x6000 – 0xFFFF)

These objects are identical for all EtherCAT SubDevices supporting the profile number 5000 (“CAN Interface”) (1).

6.1.5.1 Object 0x6000 CAN Rx Message Queue



n … Number of Rx messages, which are collected during the test / comparison of the „Rx counter“ (Number of Messages, Index 0x6000, Sub 3), but maximum 16 (Default, Index 0x8000 Sub 21)

Figure 27: SubDevice Relationship of the CAN Rx message queues

The number of transmitted Rx messages (**n**) is written in *Number of Rx Messages* (Sub-Index 3) and must not be changed until the "Rx Counters" are equal again.
For the chronological sequence see the example in Figure 29 on page 45.

Index	Sub-Index	Name	Data Type	RW	Default
0x6000	0	*Number of sub-indices*	UINT8	RO	
	1	*Tx Counter Gateway*	UINT16	RO	
	2	*Rx Counter Gateway*	UINT16	RO	
	3	*Number of Rx Messages*	UINT16	RO	
	4	*Tx Transaction Number*	UINT16	RO	
	5	*Rx Message 1*	OCTET-STRING[10]	RO	
	6	*Rx Message 2*	OCTET-STRING[10]	RO	
	:	:	:	:	
	m	*Rx Message (m-4)*	OCTET-STRING[10]	RO	

This object contains the CAN interface input messages with 11 Bit ID.

Parameter Description

Tx Counter Gateway

The Tx counter is incremented by the Gateway to indicate that the CAN Tx messages were copied from the output data to the local CAN send queue (see Figure 30).

Rx Counter Gateway

The Rx counter is incremented by the Gateway every time new CAN Rx data has arrived and the *Rx Counter Gateway* (0x6000, sub-index 02) is identical with *Rx Counter Application* (0x7000, sub-index 02). This indicates that new Rx data has been written into the process input data (see Figure 27).

Number of Rx Messages

Contains the number of CAN Rx messages in the following input data when the RX Counter was increased.

Tx Transaction Number

Contains the transaction number of the last sent Tx (see Figure 28)

Rx Message 1...(m-4)

1. to (m-4). CAN Rx message
The message is composed of the following components:

Bit 0-3:	CAN message length (0…8 bytes)
Bit 4:	RTR Bit
Bit 5-15:	CAN Identifier (11-bit CAN ID)
Bit 16-79:	CAN-Rx data

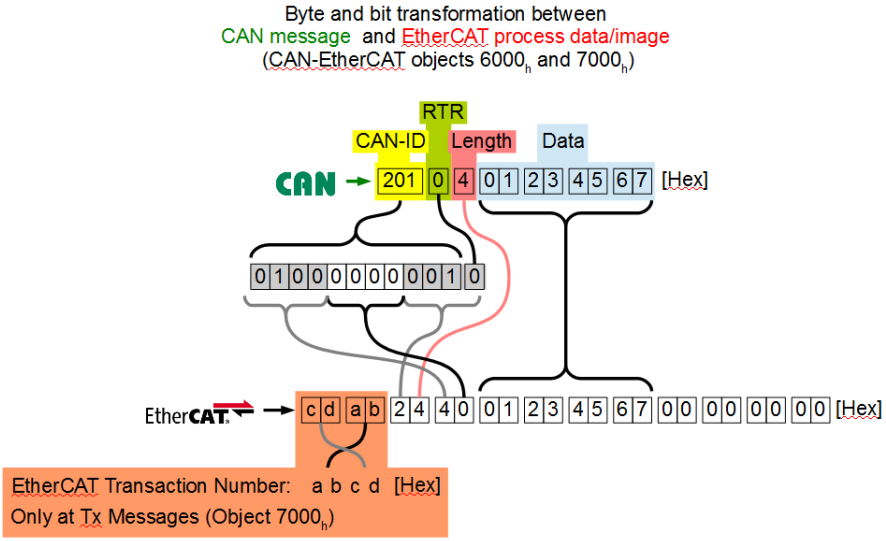


Figure 28: Formatting of the CAN data in the EtherCAT process image (11-bit Rx and Tx)

Example Sequence Rx-Counter

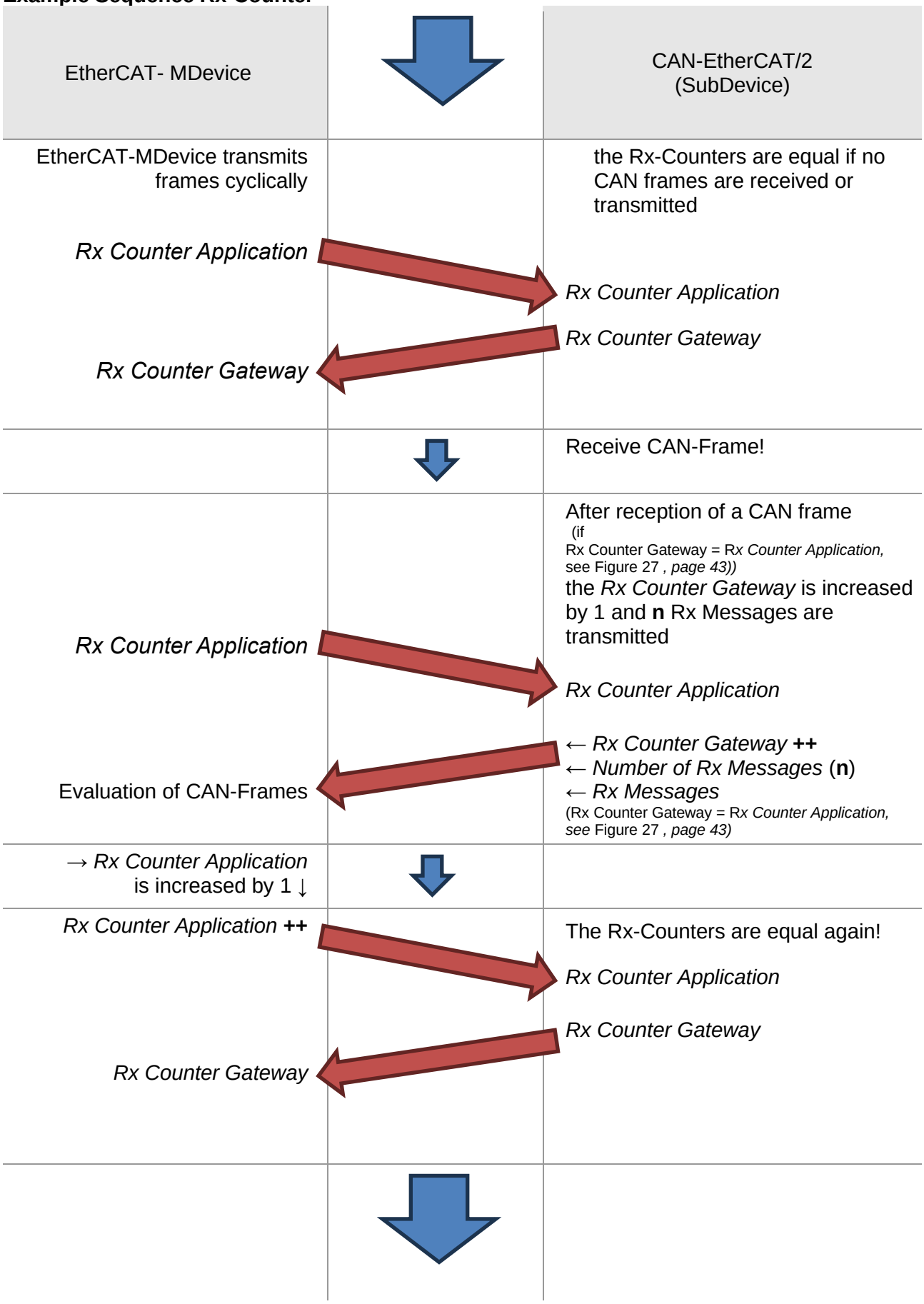


Figure 29: Chronological sequence of the Rx-Counters

6.1.5.2 Object 0x6001 Extended CAN Rx Message Queue

Index	Sub-Index	Name	Data Type	RW	Default
0x6001	0	*Number of sub-indices*	UINT8	RO	
	1	*Tx Counter Gateway*	UINT16	RO	
	2	*Rx Counter Gateway*	UINT16	RO	
	3	*Number of Rx Messages*	UINT16	RO	
	4	*Tx Transaction Number*	UINT16	RO	
	5	*Rx Message 1*	OCTET-STRING[14]	RO	
	6	*Rx Message 2*	OCTET-STRING[14]	RO	
	:	.	:	:	
	m	*Rx Message (m-4)*	OCTET-STRING[14]	RO	

This object contains the CAN interface input messages with 29-bit ID.
See Figure 27 for the relationship of the CAN-Rx-Message-Queues. For 29-Bit-Identifiers objects 0x6001 and 0x7001 are used instead of objects 0x6000 and 0x7000 accordingly.
For the chronological sequence see the example in Figure 29 on page 45.

Parameter Description

Tx Counter Gateway — The Tx counter is incremented by the Gateway to indicate that the CAN Tx messages were copied from the output data to the CAN send queue (see Figure 30).

Rx Counter Gateway — The Rx counter is incremented by the Gateway every time new CAN Rx data has arrived and the *Rx Counter Gateway* (0x6001, sub-index 02) is identical with *Rx Counter Application* (0x7001, sub-index 02). This indicates that new Rx data has been written into the process input data (see Figure 27).

Number of Rx Messages — Contains the number of CAN Rx messages in the following input data when the Rx counter was increased.

Tx Transaction Number — Contains the transaction number of the last sent Tx message (see Figure 28).

Rx Message 1...(m-4) — 1. to (m-4). CAN Rx message
The message is composed of the following components:

Bit 0-3:	CAN-Rx message length (0...8 byte)
Bit 4-15:	Reserved
Bit 16-44:	CAN Identifier (11- or 29-bit CAN identifier)
Bit 46:	RTR bit
Bit 47:	0 = 11-bit CAN identifier 1 = 29-bit CAN identifier
Bit 48-111:	CAN Rx data

6.1.5.3 Object 0x6002 CAN FD Rx Message Queue (CAN-EtherCAT/2-FD only)

Index	Sub-Index	Name	Data Type	RW	Default
0x6002	0	*Number of sub-indices*	UINT8	RO	
	1	*Tx Counter Gateway*	UINT16	RO	
	2	*Rx Counter Gateway*	UINT16	RO	
	3	*Number of Rx Messages*	UINT16	RO	
	4	*Tx Transaction Number*	UINT16	RO	
	5	*Rx Message 1*	OCTET-STRING[70]	RO	
	6	*Rx Message 2*	OCTET-STRING[70]	RO	
	:	*:*	:	:	
	m	*Rx Message (m-4)*	OCTET-STRING[70]	RO	

This object can only be used with the CAN FD variant CAN-EtherCAT/2-FD. It contains the CAN FD interface input messages. See Figure 27 for the relationship of the CAN-Rx-Message-Queues. For CAN FD objects 0x6002 and 0x7002 are used instead of objects 0x6000 and 0x7000 accordingly. For the chronological sequence see the example in Figure 29 on page 45.

Parameter Description

Tx Counter Gateway

The Tx counter is increased by the Gateway to indicate that the CAN Tx messages were copied from the output data to the CAN send queue (see Figure 30).

Rx Counter Gateway

The Rx counter is increased by the Gateway every time new CAN Rx data has arrived and the *Rx Counter Gateway* (0x6002, sub-index 02) is identical with *Rx Counter Application* (0x7002, sub-index 02). This indicates that new Rx data has been written into the process input data (see Figure 27).

Number of Rx Messages

Contains the number of CAN Rx messages in the following input data when the Rx counter was increased.

Tx Transaction Number

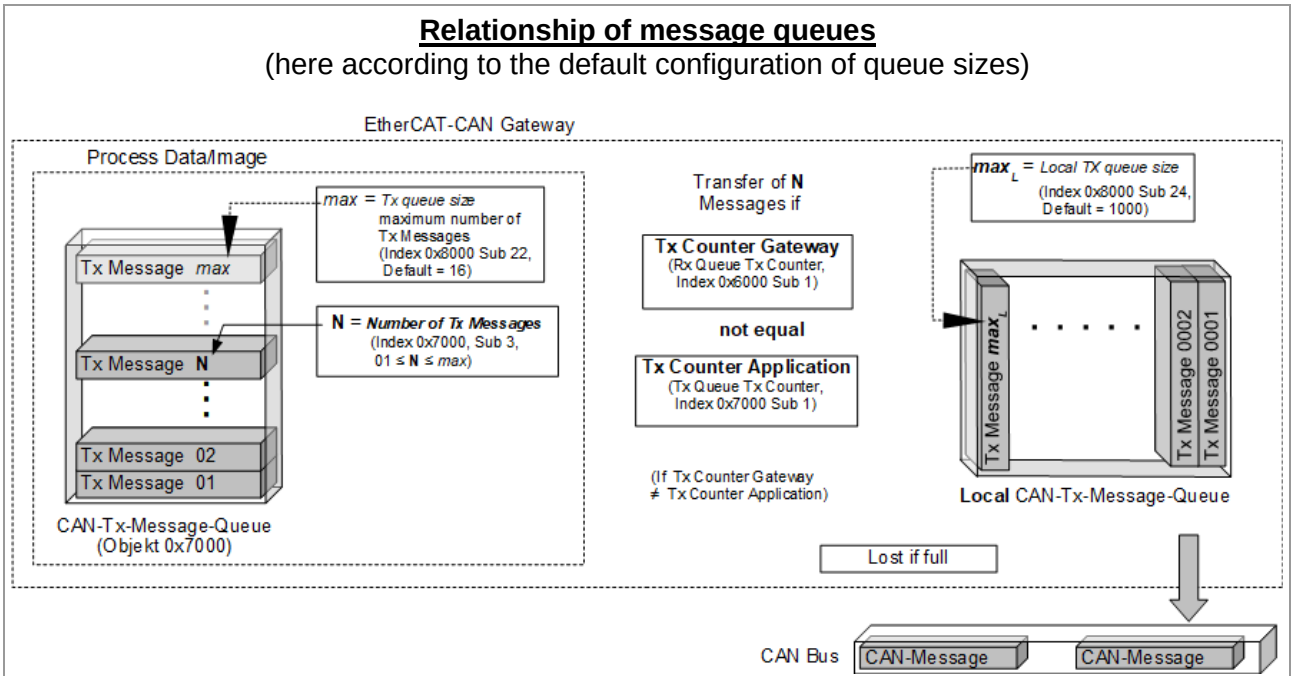
Contains the transaction number of the last sent Tx message (see Figure 28).

Rx Message 1...(m-4)

1. to (m-4). CAN Rx message,
Under normal circumstances, you cannot make the queue bigger than 20 messages before you run out of process RAM.
The message is composed of the following components:

Bit 0-7:	Length of CAN message (0...15 byte)
Bit 8-12:	reserved
Bit 13:	EDL - Bit (Enhanced Data Length)
Bit 14:	BSR - Bit (Bit Rate Switched)
Bit 15:	ESI - Bit (Error State Indicator)
Bit 16-44:	CAN Identifier
Bit 46:	RTR bit
Bit 47:	0 = 11-bit CAN identifier 1 = 29-bit CAN identifier
Bit 48-559:	CAN (FD) Rx data

6.1.5.4 Object 0x7000 CAN Tx Message Queue



N … Number of Tx messages, which are collected during the test / comparison of the „Tx counter“ (*Number of Tx Messages*, Index 0x7000, Sub 3), but maximum 16 (Default, *Tx queue size* Index 0x8000 Sub 22).

Figure 30: Relationship of the CAN Tx message queues

The number of effectively transmitted Tx messages (**N**) is written in *Number of Tx Messages* (Sub-Index 3) and will not be changed until the "Tx Counters" are equal again.
For the chronological sequence see the example in Figure 31 on page 50.

NOTICE
The following CoE objects (Index, Sub-Index) can only be written in *Pre-Operational* state: Index: 0x7000, Sub-Index: 1...m

Index	Sub-Index	Name	Data Type	RW	Default
0x7000	0	*Number of sub-indices*	UINT8	RO	
	1	*Tx Counter Application*	UINT16	RW	
	2	*Rx Counter Application*	UINT16	RW	
	3	*Number of Tx Messages*	UINT16	RW	
	4	*Tx Message 1*	OCTET-STRING[12]	RW	
	5	*Tx Message 2*	OCTET-STRING[12]	RW	
	:	:	:	:	
	m	*Tx Message (m-3)*	OCTET-STRING[12]	RW	

This object contains the CAN interface output messages with 11-bit ID.

The maximum value of the sub-index (*m*), and thus the number of Tx messages (*m-3*), is defined in the RxPDO-Mapping-Object (object 0x1600).

Parameter Description

Tx Counter Application

This counter must be incremented when or after writing the CAN Tx message to the output data (see Figure 30).

Rx Counter Application

This counter must be incremented by the EtherCAT MDevice application for each CAN Rx message list it has received and read. This indicates that the received Rx messages have been read (see Figure 27).

Number of Tx Messages

Contains the number of CAN Tx messages, which are transmitted with every increase of the Tx counter.

Tx Message 1...(m-3)

CAN Tx messages which are transmitted with every increase of the Tx counter.
The message is composed of the following components:

Bit 0-15:	Transaction Number The transaction number of the last transmitted CAN Tx message; readable in the input data.
Bit 16-19:	CAN message length (0…8 bytes)
Bit 20:	RTR bit
Bit 21-31:	CAN identifier (11-bit CAN ID)
Bit 32-95:	CAN Tx data

See Figure 28 on page 44.

Example Chronological Sequence Tx-Counter

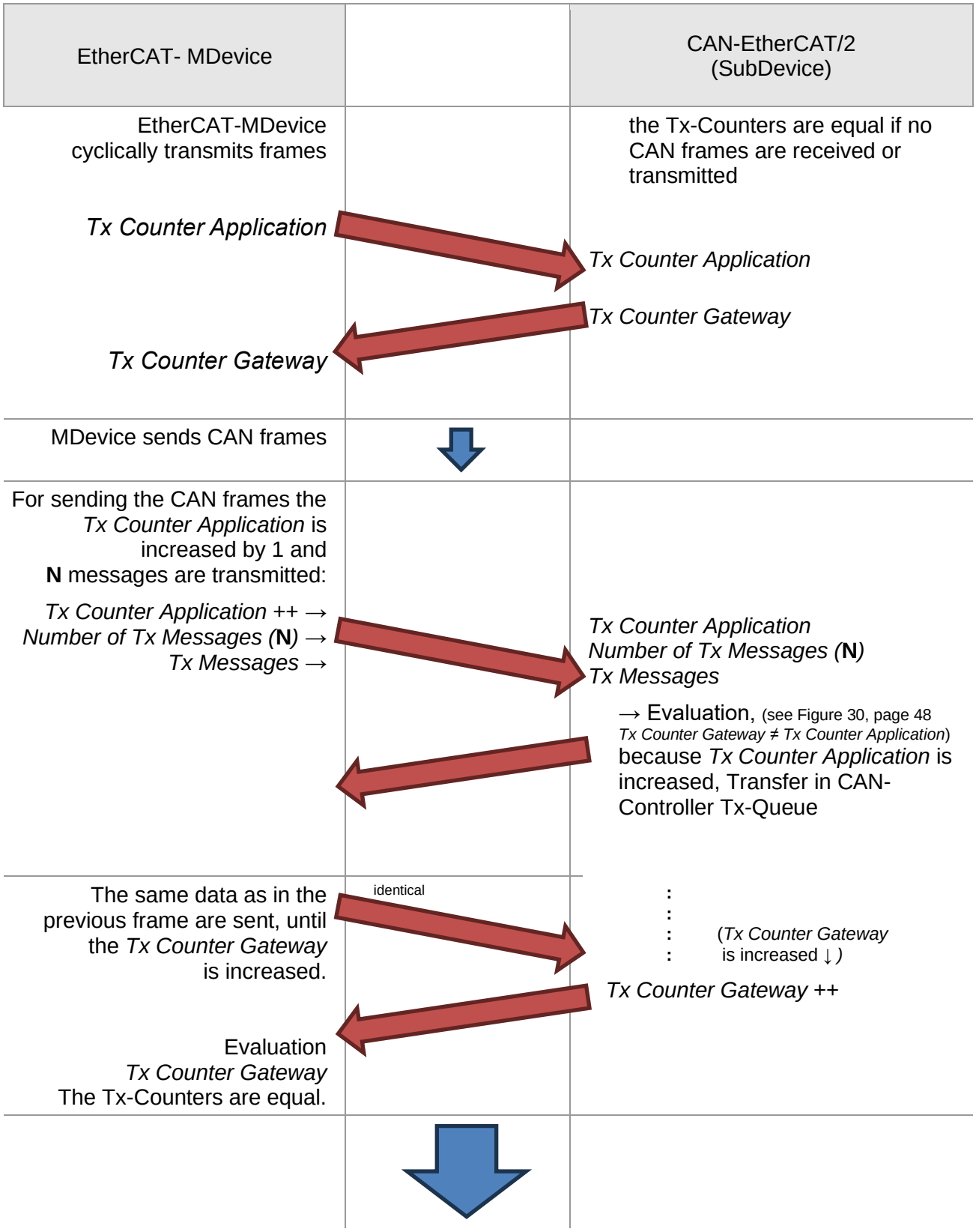


Figure 31: Chronological Sequence Tx-Counter

6.1.5.5 Object 0x7001 Extended CAN Tx Message Queue

NOTICE
The following CoE objects (Index, Sub-Index) can only be written in *Pre-Operational* state: Index: 0x7001, Sub-Index 1 ... m

Index	Sub-Index	Name	Data Type	RW	Default
0x7001	0	*Number of sub-indices*	UINT8	RO	
	1	*Tx Counter Application*	UINT16	RW	
	2	*Rx Counter Application*	UINT16	RW	
	3	*Number of Tx Messages*	UINT16	RW	
	4	*Tx Message 1*	OCTET-STRING[16]	RW	
	5	*Tx Message 2*	OCTET-STRING[16]	RW	
	:	:	:	:	
	m	*Tx Message (m-3)*	OCTET-STRING[16]	RW	

This object contains the CAN interface output messages with 29-Bit ID.
See Figure 30 for the relationship of the CAN-Tx-Message-Queues. For 29-Bit-Identifiers objects 0x6001 and 0x7001 are used instead of objects 0x6000 and 0x7000 accordingly.
For the chronological sequence see the example in Figure 31 on page 50.

Parameter Description

Tx Counter Application
This counter must be increased when or after writing the CAN Tx message to the output data (see Figure 30).

Rx Counter Application
This counter must be increased by the EtherCAT MDevice application for each CAN Rx message list it has received and read. This indicates that the received Rx messages have been read (see Figure 27).

Number of Tx Messages
Contains the number of CAN Tx messages which are transmitted with every increase of the Tx counter.

Tx Message 1...(m-3)
CAN Tx messages which are transmitted with every increase of the Tx counter.
The message is composed of the following components:

Bit 0-15:	Transaction Number The transaction number of the last transmitted CAN Tx message; readable in the input data.
Bit 16-31:	CAN message length (0...8 byte)
Bit 32-60:	CAN Identifier (11- or 29-bit CAN ID)
Bit 62:	RTR bit
Bit 63:	0 = 11-bit CAN identifier 1 = 29-bit CAN identifier
Bit 64-127:	CAN Tx data

6.1.5.6 Object 0x7002 CAN FD Tx Message Queue

NOTICE
The following CoE objects (Index, Sub-Index) can only be written in *Pre-Operational* state: Index: 0x7002, Sub-Index 1 ... 23

Index	Sub-Index	Name	Data Type	RW	Default
0x7002	0	*Number of sub-indices*	UINT8	RO	
	1	*Tx Counter Application*	UINT16	RW	
	2	*Rx Counter Application*	UINT16	RW	
	3	*Number of Tx Messages*	UINT16	RW	
	4	*Tx Message 1*	OCTET-STRING[72]	RW	
	5	*Tx Message 2*	OCTET-STRING[72]	RW	
	...	...	...	...	
	m	*Tx Message (m-3)*	OCTET-STRING[72]	RW	

This object can only be used with the CAN FD variant CAN-EtherCAT/2-FD. It contains the CAN FD interface output messages. See Figure 30 for the relationship of the CAN-Tx-Message-Queues. For CAN FD objects 0x6002 and 0x7002 are used instead of objects 0x6000 and 0x7000 accordingly. For the chronological sequence see the example in Figure 31 on page 50.

Parameter Description

Tx Counter Application
This counter must be incremented when or after writing the CAN Tx message to the output data (see Figure 30).

Rx Counter Application
This counter must be incremented by the EtherCAT MDevice application for each CAN Rx message list it has received and read. This indicates that the received Rx messages have been read (see Figure 27).

Number of Tx Messages
Contains the number of CAN Tx messages which are transmitted with every increase of the Tx counter.

Tx Message 1...(m-3)
CAN FD Tx messages which are transmitted with every increase of the Tx counter. Under normal circumstances, you cannot make the queue bigger than 20 messages before you run out of process RAM. The messages are composed of the following components:

Bit 0-15:	Transaction Number - The transaction number of the last CAN Tx message sent, readable in the input data.
Bit 16-23:	CAN FD message length (0...15)
Bit 24-28:	Reserved
Bit 29:	EDL - Bit (Enhanced Data Length)
Bit 30:	BSR - Bit (Bit Rate Switched)
Bit 31:	ESI - Bit (Error State Indicator)
Bit 32-60:	CAN Identifier (11- or 29-bit CAN ID)
Bit 62:	RTR-bit
Bit 63:	0 = 11-bit CAN identifier 1 = 29-bit CAN identifier
Bit 64-575:	CAN (FD) data

6.1.5.7 Object 0x8000 CAN-Interface-Configuration

NOTICE
The following CoE objects (Index, Sub-Index) can only be written in *Pre-Operational* state: Index: 0x8000, Sub-Index: 0x20, 0x21, 0x22

Index	Sub-Index	Name	Data Type	RW	Default
0x8000	0	*Number of sub-indices*	UINT8	RO	0x24 (36)
	1	*Node Address*	UINT16	RW	0x0000
	2 : 31	Reserved for future extensions	-	-	-
	32	*Flags*	UINT16	RW	0x0000
	33	*Rx queue size*	UINT8	RW	0x10 (16)
	34	*Tx queue size*	UINT8	RW	0x10 (16)
	35	*Local Rx queue size*	UINT16	RW	CAN CC variant: 0x80 (128) CAN FD variant: 0x20 (32)
	36	*Local Tx queue size*	UINT16	RW	CAN CC variant: 0x80 (128) CAN FD variant: 0x20 (32)

The CAN interface can be configured with this object.

Parameter Description

Node Address Must be set to 0 (to distinguish from the configuration of CANopen Servers)

Flags

Bit 0-2: Reserved for future extensions; must be 0

Bit 3: 0 = Standard CAN message queue (11-bit identifier),
1 = Extended CAN message queue (29-bit identifier)

Bit 4-8: Reserved for future extensions; must be 0

Bit 9: In Fast Mode, when a CAN frame is received, the module will copy the frame data to the process data, regardless of the rx counter, meaning CAN frames can be transferred every cycle, not every other cycle (after the rx counter has been increased by the master, without "Fast Mode"). The disadvantage is that if the master cannot keep up checking the process data for every frame (e.g. the process data is checked by some PLC with a different cycle than the EtherCAT cycle), the frames not processed will be lost.

Bit 10-14: Reserved for future extensions; must be 0

Bit 15: 0 = CAN CC message queue
1 = CAN FD message queue (CAN FD variant only)

CAN-EtherCAT/2: For the CAN CC variant bit 15 is always = 0
CAN-EtherCAT/2-FD: Bit 15 can be set to 0 or 1

Rx queue size	Number of Rx messages; CAN CC: m_{MAX} = 250*, CAN FD: m_{MAX} = 20*
Tx queue size	Number of Tx messages; CAN CC: m_{MAX} = 250*, CAN FD: m_{MAX} = 20*
Local Rx queue size	Maximum number of CAN RX messages which could be stored in the CAN interface, CAN CC: m_{MAX_L} = 0x80 (128), CAN FD: m_{MAX_L} = 0x20 (32)
Local Tx queue size	Maximum number of CAN TX messages which could be stored in the CAN interface, CAN CC: m_{MAX_L} = 0x80 (128), CAN FD: m_{MAX_L} = 0x20 (32)

** Also limited by SM size/configuration*

6.1.5.8 Object 0x8001 CAN-Rx-Filter-Table

Index	Sub-Index	Name	Data Type	RW	Default
0x8001	0	*Number of sub-indices*	UINT8	RW	
	1	*Identifier Area 1*	UINT64	RW	
	...	...	...	...	
	m	*Identifier Area m*	UINT64	RW	

This object assigns the CAN identifier areas, which are filled into the RX queue and are transmitted with the EtherCAT input data.

INFORMATION
For 29-Bit CAN identifiers bit 31 and 63 must be set!

In case this object is not configured, all received CAN messages will be assigned to the Rx queue and transmitted via the EtherCAT input data.

Parameter Description

Identifier Area 1 Byte 0-3: First identifier to be assigned to the Rx queue
Byte 4-7: Last identifier to be assigned to the Rx queue

Identifier Area m Byte 0-3: First identifier to be assigned to the Rx queue
Byte 4-7: Last identifier to be assigned to the Rx queue

m = max. 0xFF (255)

6.1.5.9 Object 0xF000 Modular Device Profile

Usually only needed by configuration tools, e.g. esd EtherCAT Workbench. See ETG.5100 documents for details.

Index	Sub-Index	Name	Data Type	RW	Default
0xF000	0	*Max. sub-index*	UINT8	RO	3
	1	*Index distance*	UINT16	RO	0x10 (16)
	2	*Maximum number of modules*	UINT16	RO	1
	3	*General configuration*	UINT32	RO	1

Parameter Description

Index distance	Specifies the index distance between two modules = 0x10
Maximum number of modules	
General configuration	Available entries in objects 0x8nn0 = 0x00000001

6.1.5.10 Object 0xF100 Link Status

Index	Sub-Index	Name	Data Type	RW	Default
0xF100	0	*Max. sub-index*	UINT8	RO	2
	1	*Link status*	UINT16	RO	-
	2	*Reserved*	UINT16	RO	0

Parameter Description

Link status	Status of the Ethernet link 0 = link_ok 1 = no_link

6.1.5.11 Object 0xF108 CAN Status

Index	Sub-Index	Name	Data Type	RW	Default
0xF108	0	*Max. sub-index*	UINT8	RO	0x22 (34)
	1	*Bus OFF* *(Read from CAN controller status byte)*	BOOL	RO	false
	2	*Warning Limit reached* *(Read from CAN controller status byte)*	BOOL	RO	false
	3	*Rx overflow* *(Read from CAN controller overrun counter)*	BOOL	RO	false
	4	*Reserved*	BIT1	RO	0
	5	*Tx overflow* *(Not served, always false)*	BOOL	RO	false
	6	*Ack error* *(Not served, always false)*	BOOL	RO	false
	7	*Reserved*	BIT2	RO	0
	8	*Reserved*	BIT8	RO	0
	17	*Reserved (by esd)*	BIT1	RO	0
	18	*Reserved (by esd)*	BIT1	RO	0
	19	*Reserved (by esd)*	BIT1	RO	0
	20	*Reserved (by esd)*	BIT1	RO	0
	33	*Rx error counter* *(Read from CAN controller Rx error counter byte)*	USINT	RO	0
	34	*Tx error counter* *(Read from CAN controller Tx error counter byte)*	USINT	RO	0

6.1.5.12 Object 0xF800: CAN Bus Parameter Set

Index	Sub-Index	Name	Data Type	RW	Default
0xF800	0	*Number of sub-indices*	UINT8	R	0x20 (32)
	1	*Reserved for CANopen MainDevice**	UINT8	RW	0x7F (127)
	2	*Bit rate*	UINT8	RW	0xFF (255)
	3	*Reserved for CANopen MainDevice**	UINT16	RW	0x0080 (128)
	4	*Reserved for CANopen MainDevice**	UINT32	RW	0
	5	*Bustiming Register CAN CC*	UINT32	RW	0x7FFFFFFF
	6 : 31	*Reserved for future extensions*			
	32	*Bustiming Register CAN FD*	UINT64	RW	0

* These reserved entries have no functionality, because the CAN-EtherCAT/2 gateway does not contain a CANopen MainDevice. The default values for these entries are defined in the specification.

This object contains the CAN CC bit rate or the nominal and the data bit rate for CAN FD.
The CAN CC and CAN FD bit rates can also be set via the bus timing registers.

Parameter Description

Bit rate

The CAN CC bit rate can be set according to Table 8 below:
CAN-EtherCAT/2-FD only: The CAN FD bit rate can be set according to Table 10 below:

CAN CC bit rate: This table shows the bit rate index used for CAN CC.

Parameter *Bit rate* [decimal]	CAN Bit rate [Kbit/s]
0	1000
1	800
2	500
3	250
4	125
5	100
6	50
7	20
8	10
255	CAN bit rate as defined in *Bustiming Register CAN CC* (sub-index 5), see Table 9

Table 8: Parameters CAN CC *Bit rate*

Bustiming Register CAN CC

Optionally the bit timing registers of the processor can be set directly.
If the *Bit rate* (Sub-index 2) is set to 0xFF (255), the bit rate is defined in the *Bustiming Register CAN CC* (sub-index 5) and interpreted as RAW-TSEG setting. See also Table 12 for the structure of the parameter *bit rate*. (Bit 31 UBR must be set to 1 to specify bustiming directly)

Bit	Description	Value range
0 - 9	arb.brp	0 … 255
16 -17	arb.sjw	0 … 255
20 - 23	arb.tseg1	0 … 127
24 - 26	arb.tseg2	0 … 127

Table 9: Bustiming Register CAN CC

CAN FD bit rate (CAN-EtherCAT/2-FD only):
This table shows the bit rate Index used for CAN FD.
It defines the nominal CAN FD bit rate (Arbitration bit rate)
and the CAN FD data bit rate.

Parameter *Bit rate* [decimal]	CAN FD Nominal Bit Rate	CAN FD Data Bit Rate
30	250 kbit/s	500 kbit/s
29	250 kbit/s	833 kbit/s
28	250 kbit/s	1 Mbit/s
27	250 kbit/s	1.538 Mbit/s
26	250 kbit/s	2 Mbit/s
25	250 kbit/s	4 Mbit/s
24	500 kbit/s	1 Mbit/s
23	500 kbit/s	2 Mbit/s
22	500 kbit/s	4 Mbit/s
21	500 kbit/s	5 Mbit/s
20	500 kbit/s	6.67 Mbit/s
19	500 kbit/s	8 Mbit/s
18	500 kbit/s	10 Mbit/s
17	1 Mbit/s	2 Mbit/s
16	1 Mbit/s	4 Mbit/s
15	1 Mbit/s	5 Mbit/s
14	1 Mbit/s	8 Mbit/s
13	1 Mbit/s	10 Mbit/s
10	CAN FD Bit rate as defined in parameter *Bustiming Register CAN FD (*Sub-index 32), see Table 11	

Table 10: Parameters CAN FD Bit rates

***Bustiming Register CAN FD* (CAN-EtherCAT/2-FD only)**
Optionally the bit timing registers of the processor can be set directly.
If the *Bit rate* (Sub-index 2) is set to 0x10, the bit rate is defined in the *Bustiming Register CAN FD* (sub-index 32) and interpreted as RAW-TSEG setting. See also Table 12 for the structure of the parameter *bit rate.*

Bit	Register	Description	Value range
0 - 7	BT0	arb.brp	0 … 255
8 - 15	BT1	arb.tseg1	0 … 255
16 - 23	BT2	arb.tseg2	0 … 127
24 - 31	BT3	arb.sjw	0 … 127
32 - 39	BT4	data.brp	
40 - 47	BT5	data.tseg1	0 … 31
48 - 55	BT6	data.tseg2	0 … 15
56 - 63	BT7	data.sjw	0 … 15

Table 11: Bustiming Register CAN FD

ARB_BRP must be the same as DATA_BRP

Structure of Bustiming Register CAN CC
The parameter *bit rate* defines the CAN CC bit rates.
The bits 28 to 31 of this 32-bit argument are used as configuration flags.
The combination of the User Bit Rate (UBR) bit 31 and the User Bit Rate Numerical (UBRN) bit 29 define the meaning of the value given as bit rate in bit 0 to 27 of this argument.
If supported by the CAN controller hardware, the Listen Only Mode (LOM via bit 30) and/or the Self-Test Mode (STM via bit 28) can be optionally set in addition to the bit rate, to enable the respective operation mode.

31 UBR	30 LOM	29 UBRN	28 STM	27	26... ...16	15... ...8	7... ...0
0	0/1	0	0/1	Reserved		Table index	
1	0/1	0	0/1	Reserved	CAN_BR (CAN Controller specific bit timing register values)		

Table 12: Structure of parameter *bit rate*

A combination of UBR = UBRN = 1 is not allowed!
UBRN is always = 0

Bit(s)	Value	Description
UBR	0	Use the pre-defined bit rate table (Table Index) (in combination with UBRN)
	1	Set the CAN controller bit rate registers directly (BTR0/BTR1)
LOM	0	Configure the bit rate in ‘active’ mode (normal operation)
	1	Configure the bit rate in ‘Listen-Only’ mode
UBRN	0	Use the pre-defined bit rate table
STM	0	Self-test mode disabled
	1	Self-test mode enabled: Transmit/receive without acknowledge from other CAN nodes
Table index	x	Use the bit rate in pre-defined
CAN_BR	x	CAN Controller specific bit timing register values: CAN-EtherCAT/2: CAN bit rate register of ccan , see Table 9 CAN-EtherCAT/2-FD: CAN bit rate register of esdACC, see Table 11

Table 13: Bits of parameter API-bit rate

When ‘User Bit Rate’ (UBR) is set to ‘0’ (‘User Bit Rate Numerical’ (UBRN) is always = 0), the bits 0 - 15 are interpreted as an index to a pre-defined bit rate table (Table index, see Table 14).

This allows the setting of CAN bit rates without detailed knowledge of the CAN controller hardware.

Table index	Bit rate [kBit/s]	Constant *1)
0	1000	NTCAN_BAUD_1000
0xE	800	NTCAN_BAUD_800
1	666.6	
2	500	NTCAN_BAUD_500
3	333.3	
4	250	NTCAN_BAUD_250
5	166	
6	125	NTCAN_BAUD_125
7	100	NTCAN_BAUD_100
0x10	83.3	
8	66.6	
9	50	NTCAN_BAUD_50
0xA	33.3	
0xB	20	NTCAN_BAUD_20
0xC	12.5	
0xD	10	NTCAN_BAUD_10

*1) The constants follow the CiA (CAN in Automation) recommendations.

Table 14: Pre-defined bit rate table

Constants and special features

Constant	Value	Function
NTCAN_BAUD_1000	0	Sets bit rate to 1000 kBit/s
NTCAN_BAUD_800	0xE	Sets bit rate to 800 kBit/s
NTCAN_BAUD_500	2	Sets bit rate to 500 kBit/s
NTCAN_BAUD_250	4	Sets bit rate to 250 kBit/s
NTCAN_BAUD_125	6	Sets bit rate to 125 kBit/s
NTCAN_BAUD_100	7	Sets bit rate to 100 kBit/s
NTCAN_BAUD_50	9	Sets bit rate to 50 kBit/s
NTCAN_NO_BAUDRATE	0x7FFF FFFF	The gateway cannot receive or transmit any message; stays passive on CAN bus
NTCAN_USER_BAUDRATE	0x8000 0000	Configure BTR register
NTCAN_LISTEN_ONLY_MODE	0x4000 0000	Configure listen only mode
NTCAN_USER_BAUDRATE_NUM	0x2000 0000	Numerical baudrate
NTCAN_SELF_TEST_MODE	0x1000 0000	Configure self-test mode

Table 15: Constant

Leaving the CAN Bus
The special constant NTCAN_NO_BAUDRATE can be used as an argument for the Parameter API-bit rate to force the hardware to leave the CAN bus and return to the Boot-Up condition (or to start it).

Listen Only Mode
This mode was developed for the purpose of CAN bus monitoring without effecting other CAN nodes.

With the Listen Only Mode (LOM) flag set to '0' the CAN controller works in regular active mode using the bit rate, which implies that messages can be received and transmitted.

Setting the LOM flag to '1' causes the CAN controller to operate in Listen Only Mode using the bit rate and can only receive messages.

CAN FD variant: LOM not supported at the time of writing

Numerical Baud Rate
The 'User Bit Rate Numerical' (UBRN) is not supported by the CAN-EtherCAT/2, so the value of UBRN is constant '0'.

Self-Test Mode
Use for testing only.
When enabled, the gateway is able to send into an empty bus. If there is no ACK by any node, the no-ACK error is ignored.

CAN CC variant
CAN-EtherCAT/2: The ccan will receive its own sent messages (only).
CAN FD variant
CAN-EtherCAT/2-FD: STM not supported at the time of writing

6.1.5.13 Object 0xFA01 Rx/Tx Ethernet Telegrams

Index	Sub-Index	Name	Data Type	RW	Default
0xFA01	0	*Max. sub-index*	UINT8	RO	3
	1	*Rx ethernet telegrams*	UINT16	RO	-
	2	*Tx ethernet telegrams*	UINT16	RO	-
	3	*reserved*	UINT16	RO	0

Parameter Description

Rx ethernet telegrams	Number of Ethernet Rx-telegrams
Tx ethernet telegrams	Number of Ethernet Tx-telegrams

6.2 EoE

The CAN-EtherCAT/2 supports two different EoE modes, it is selected by the MDevice sending the EoE configuration (Figure 32).

6.2.1 Switch Port Mode

This is the default mode. It is enabled when no EoE configuration is received. (i.e. usually, configuration tools don't send a configuration when this mode is selected) This mode is also set when an invalid EoE configuration is received.
Every Ethernet frame received from EtherCAT will be sent to Ethernet port and vice versa.

6.2.2 IP Port Mode

This mode is enabled when an EoE configuration is received. The EoE configuration must contain the IP address, subnet mask and default gateway. The MAC-Id value is ignored, the name server is optional.
In this mode the CAN-EtherCAT/2 also acts as DHCP server for a single DHCP client on its Ethernet port: The received IP configuration is offered to the DHCP client.

NOTICE
Always make sure only ONE CLIENT is connected to the EoE port when using this. As the DHCP server gives this IP to ANY client asking for an IP, connecting this to for example a company network can disturb the whole network.

6.2.3 Disabling EoE

The CAN-EtherCAT reads the “EoE enabled” bit from its EEPROM (cat. “General”, as defined in ETG.1000.6 documents). When this is set to 0 (checked at first change to PreOp after device start up), all EoE activity will be disabled and all EoE configuration options are ignored.

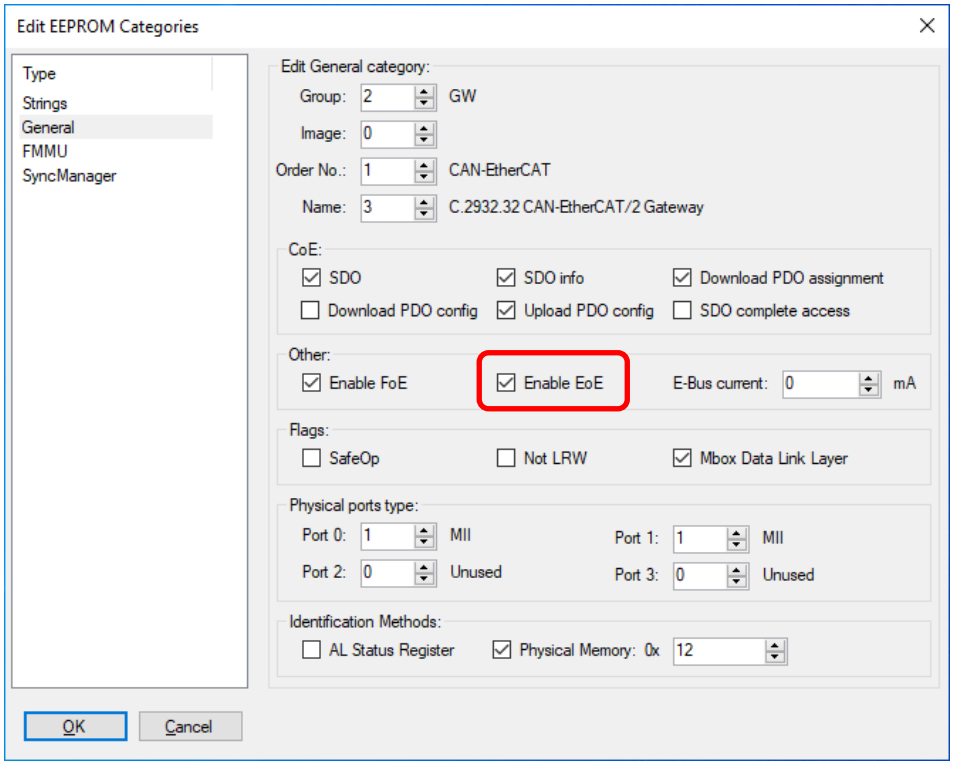


Figure 32: esd EtherCAT Workbench: Where to en/disable EoE

6.3 FoE

6.3.1 Firmware Update with the esd Workbench

NOTICE
Do not interrupt the power supply to the CAN-EtherCAT/2 gateway during a firmware update, as can lead to unpredictable operating states.

1. Make sure the SubDevice is connected, etc.
2. Set the SubDevice in the state *Bootstrap.* Therefore, choose the tab *SubDevice* and then *General. Now* click on the button *Bootstr.* as described in Figure 33. The *Current state:* of the CAN-EtherCAT/2 gateway is switched to *Bootstrap.* You can also click on the button *Bootstrap* in tab *FoE* to put the device into bootstrap mode, see Figure 34.

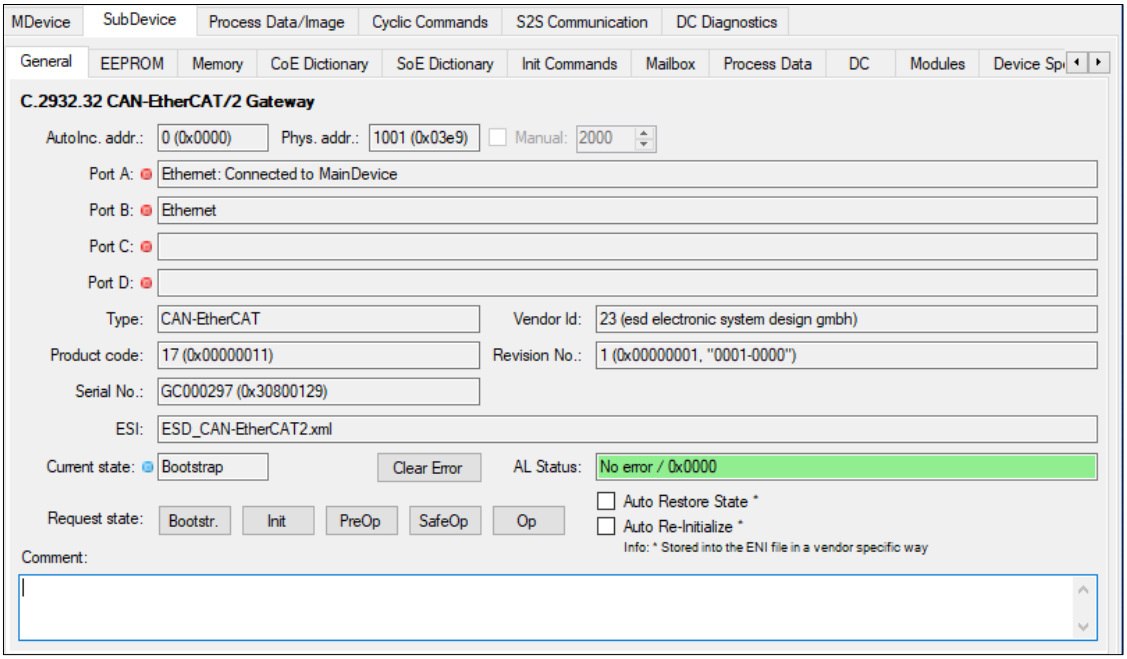


Figure 33: Firmware update via FoE

3. Select the tab *SubDevice*/*Mailbox* and choose the tab *FoE.*
   Enter „firmwareUpdate“ as name of the file in the input field *FoE filename:.*
   If you want to update an FPGA, the filename must contain “fpga”.
   Enter “fpgaUpdate” in the field *Fo filename*
   The *FoE password* has to be set to „0“.

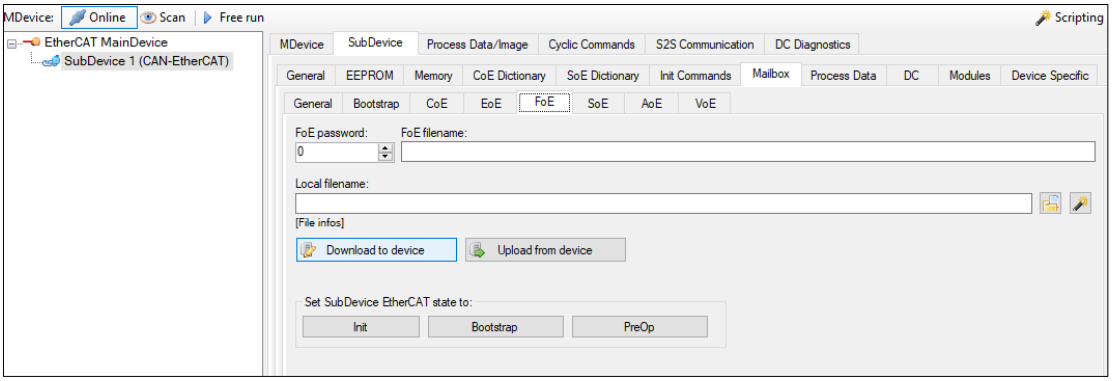


Figure 34: FoE file transfer dialog

4. Click the button *Download to device* (see Figure 34) and select the firmware file in the Windows selection dialog that appears.
   Confirm your settings with the OK button and wait until the file is transferred (a progress bar will appear, and the yellow LED is flickering while the transfer is in progress).

5. Click on the button *Init* to put the device in INIT state, starting the actual update.

6. Wait until the update procedure is completed (approx. 10 s)

7. Click on the button PreOp to get the mailboxes set up, to be able to do a CoE transfer.

8. Change to the tab *CoE Dictionary* under *SubDevice*.

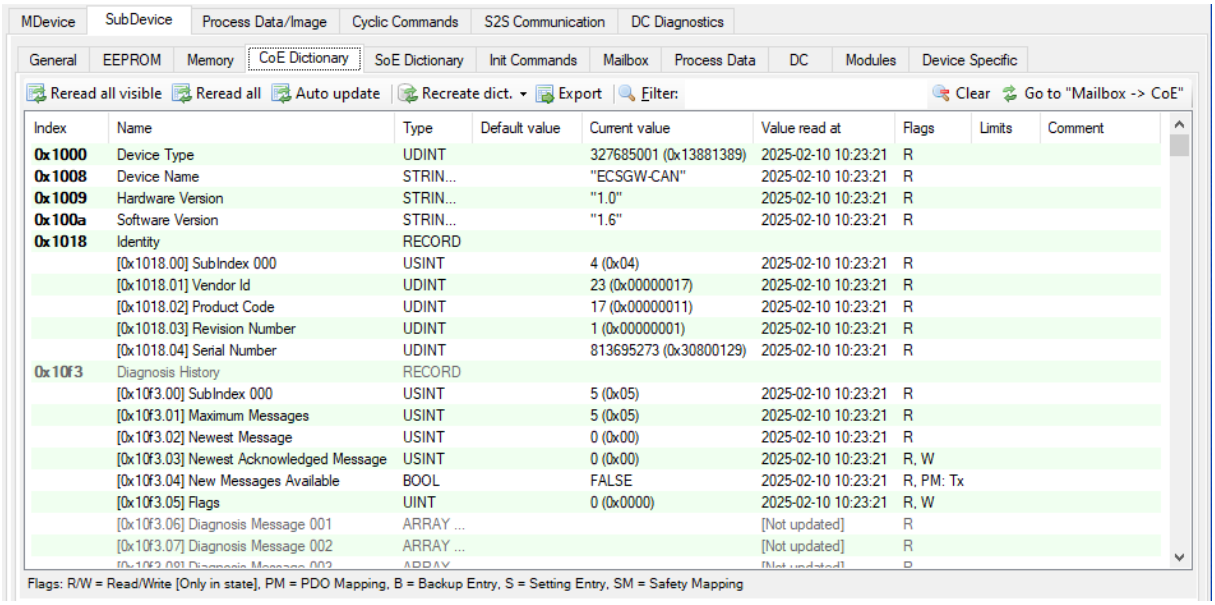


Figure 35: CAN-EtherCAT CoE Dictionary

9. Click on the button *Reread all* to ensure that the current objects are displayed.

10. Verify the current software version in object 0x100A, see Figure 35.

6.4 Firmware update with Beckhoff EtherCAT Configurator

NOTICE
Do not interrupt the CAN-EtherCAT/2 gateway power supply during a firmware update as this might result in unforeseeable operating conditions.

1. Make sure the SubDevice is connected, etc.
2. Set SubDevice to “Bootstrap”, by button *Bootstrap* (Figure 36)

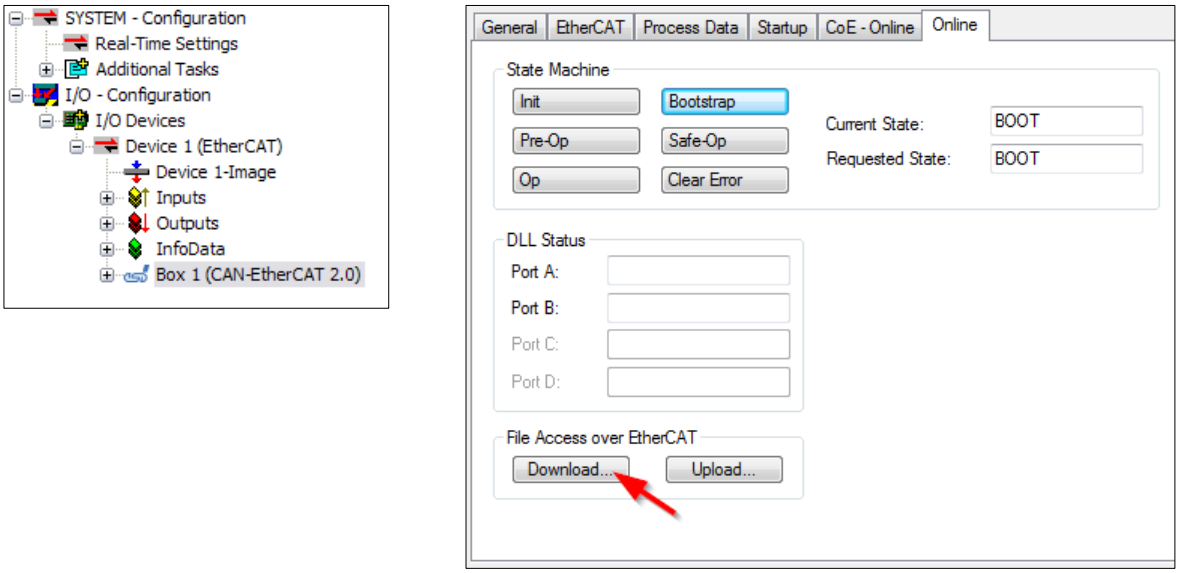


Figure 36: Firmware update by FoE

3. Click the *Download...* button and select the firmware file you received in the Windows file selection dialog that appears

4. Now a file transfer dialog (Figure 37) will appear: Set the file name string to “firmwareUpdate” and leave the password (hex) at “00000000”

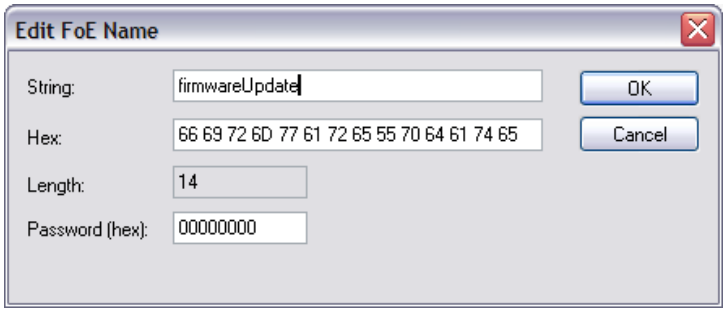


Figure 37: FoE file transfer dialog

5. Click *OK* and wait until the file is transferred (a progress bar will appear and the yellow LED is flickering while the transfer is in progress)

6. Click in the INIT button to start the actual update

7. Wait until the update is applied, approx. 10 seconds

8. Now click *Reload I/O Devices* in the Configurator toolbar (activating *Free run* is not needed) and switch to the *CoE - Online* tab page.

9. Verify the current version in object 0x100A,Figure 38.
(Make sure you're seeing online data: uncheck *Show Offline Data*, and perhaps do *Reload I/O Devices* again, etc.)

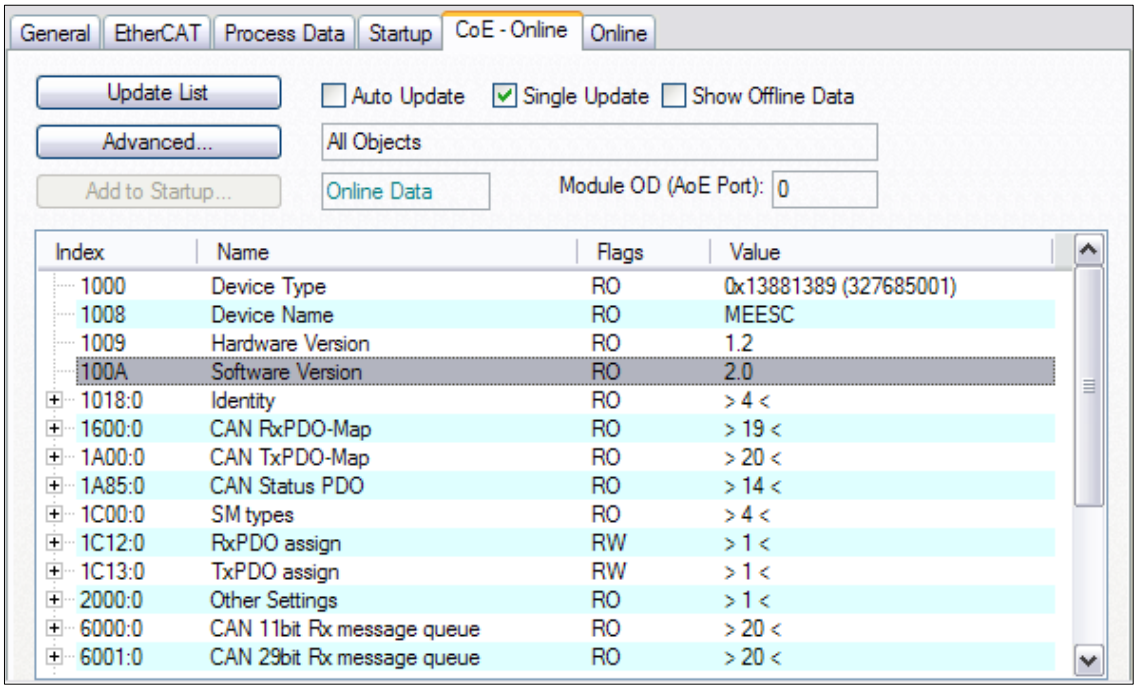


Figure 38: CAN-EtherCAT CoE dict., Software/Firmware version selected

7 Software Licenses

NOTICE
The software from esd and from third parties used in the CAN-EtherCAT/2 is subject to the license terms of the respective authors or rights holders. CAN-EtherCAT/2 may only be used in accordance with these license terms!
By using the CAN-EtherCAT/2 you agree to the terms of these software licenses.

The license terms of 3rd parties (3rd Party Licenses) can be downloaded from our website, see the following chapters.

7.1 3rd Party Software License Terms

License Name	Identifier (from SPDX License List)
Apache License 2.0	Apache-2.0
BSD 3-Clause "New" or "Revised" License	BSD-3-Clause
MIT License	MIT

Table 16: License and Identifier

- The CAN-EtherCAT/2 uses the opensource operating system FreeRTOS™.
  For the full license text see FreeRTOS, Amazon.com, Inc., Licence Details:
  https://www.freertos.org/a00114.html#license_comparison
  This also includes the MIT open source license.
  You can also download the text of the MIT License from our homepage, see Table 16.

- The CAN-EtherCAT/2 also uses the open source lwIP stack
  The lwIP stack is subject to the 3rd Party Software License Terms of BSD-3-Clause "Modified", see Table 16.

- CMSIS End User License Agreement, For the full text of the End user licence agreement for the cortex microcontroller software interface standard (CMSIS) deliverables see:
  CMSIS_END_USER_LICENCE_AGREEMENT.pdf

- The CAN-EtherCAT/2 uses the AX58200 specific Board Support Package (BSP) of ASIX.
  The BSP is subject to the 3rd Party Software License Terms of Apache-2.0, see Table 16.

7.1.1 Open-Source Software Copy

You may obtain a copy of the source code, if and as required under the license by sending a mail to oss-compliance@esd.eu

You may also obtain a copy of the source code, if and as required under the license, by sending a check or money of EUR 25.00 to:
esd electronics gmbh
Vahrenwalder Str. 207
30165 Hannover, Germany

8 Technical Data

8.1 General Technical Data

Power supply voltage	16 V … 32 V DC, Nominal: 24 V
Power consumption	CAN-EtherCAT/2-FD: Typical: 1.6 W Maximum: 2.16 W, CAN-EtherCAT/2: Typical: 1.1 W, Maximum: 1.68 W
Current consumption	CAN-EtherCAT/2-FD: $I_{24V\_TYPICAL}$ = 67 mA, I_{24V_MAX} = 90 mA), CAN-EtherCAT/2: $I_{24V\_TYPICAL}$ = 42 mA, I_{24V_MAX} = 70 mA)
Protective circuits	Reverse voltage protection Protection against transient overvoltages
Temperature range	-20°C ... +70°C ambient temperature
Humidity	Max. 90%, non-condensing
Protection class	IP20
Pollution degree	Maximum permissible according to DIN EN 61131-2: Pollution Degree 2
Housing	Plastic housing for carrier rail mounting NS35/7,5 DIN EN 60715
Form factor / Dimensions	Width: 22.5 mm, height: 99 mm, depth: 114.5 mm (without connectors)
Weight	Approx. 125 g

Table 17: General Data of the module

8.2 Connectors accessible from Outside

Name	Function, Ports	Type	Durability (e.g. grade, contact surface, mating cycles)
EtherCAT	EtherCAT Port IN	Dual port RJ45 socket with integrated transformer and LEDs	100 mating cycles
	EtherCAT Port OUT		
ETH	Ethernet	RJ45 socket with integrated transformer and LEDs	100 mating cycles
24V	24V-power supply	4-pin Phoenix Contact PCB terminal block MSTBO 2,5/ 4-G1L KMGY with pluggable connector FKCT 2,5/4-ST KMGY with push-in spring connection	25 mating cycles
InRailBus	CAN and 24V power supply via InRailBus	Card edge connector for mounting rail connector 5-pos. TBUS connector (optional accessories)	25 mating cycles
CAN	CAN (FD)	5-pin Phoenix Contact PCB terminal block MC 1,5/5-GF-3,81 with pluggable connector FK-MCP 1,5/5-STF-3,81 with push-in spring connection	25 mating cycles

Table 18: Connectors, accessible from outside

8.3 CAN Port

Number of CAN ports	CAN-EtherCAT/2-FD: 1x CAN FD CAN-EtherCAT/2: 1x CAN CC
CAN controller	CAN-EtherCAT/2-FD: esdACC in FPGA CAN-EtherCAT/2: integrated in CPU both acc. to ISO 11898-1 (CAN 2.0 A/B)
CAN protocol	According to ISO 11898-1
Physical CAN Layer	High-speed CAN port according to ISO 11898-2, CAN-EtherCAT/2-FD: CAN FD bit rate 10 kbit/s *[2] up to 8 Mbit/s CAN-EtherCAT/2: CAN CC bit rate 10 kbit/s *[2] up to 1 Mbit/s
Galvanic isolation	Separation by means of a digital isolator (transformer-based) and DC/DC-converter voltage over CAN isolation (CAN to EARTH; CAN to Host/System Ground: 1kV DC @ 1s (I < 1 mA))
Bus termination	Terminating resistor must be set externally, if required
Connector	Via CAN connector or via InRailBus connector, see 8.2

Table 19: Specification of the CAN port

8.4 Ethernet Port

Number of Ethernet ports	1
Standard	IEEE 802.3, 10BASE-T, 100BASE-TX
Bit rate	10/100 Mbit/s
Connection	Twisted Pair (compatible with IEEE 802.3), 100BASE-TX
Controller	integrated in CPU
Galvanic isolation	Via transformer integrated in RJ45 socket
Connector	RJ45 socket in the front panel with integrated LEDs (Link/Activity and Speed)

Table 20: Specification of the Ethernet port

2 Limited by transceiver dominant timeout, typ. 1.25ms.

8.5 EtherCAT Port

Number	1
Standard	IEEE 802.3, 100BASE-TX
Type	EtherCAT SubDevice
Controller	Beckhoff ET1100 compatible integrated in the CPU
Connection	Twisted Pair (compatible with IEEE 802.3), 100BASE-TX
Galvanic isolation	Via transformer, integrated in RJ45 socket
Name	Port IN and Port OUT
Connector	Dual port RJ45 socket with integrated LEDs for status indication (Link/Activity)

Table 21: Specification of the EtherCAT ports

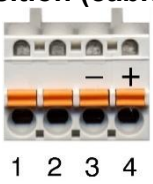
9 Connector Assignments

9.1 24V Power Supply Voltage

DANGER
The CAN-EtherCAT/2 is a device of protection class III according to DIN EN 61140 and may only be operated on supply circuits that offer sufficient protection against dangerous voltages.

Device socket: Phoenix Contact PCB header MSTBO 2,5/4-G1L-KMGY
Cable plug: Phoenix Contact pluggable connector FKCT 2,5/4-ST, 5.0 mm pitch, Push-in spring connection, included in the scope of delivery, (Phoenix Contact order No.: 19 21 90 0)
For conductor connection, cross section and stripping length see page 77.

Pin Position (cable plug):



Pin Assignment:

Device housing label			24V	
	.	.	M	P
Connector label	(none)	(none)	-	+

Pin	1	2	3	4
Signal	P24 (+ 24 V)	M24 (GND)	M24 (GND)	P24 (+ 24 V)

Please refer to the connecting diagram page 13.

INFORMATION
The P24 pins (pin 1, 4) and the M24 pins (pin 2, 3) are connected internally.

NOTICE
There is a connection between the 24V plug and the InRailBus so that the module can be supplied via the InRailBus. Note that this connection is not designed to feed the 24V supply voltage via the plug to the InRailBus!
Feeding through the 24V power supply voltage can cause damage on the module!
Further wiring via the plug is possible if the modules are mounted on the DIN rail without InRailBus connectors. Use pin 1 and 2 as input wiring and pins 3 and 4 as output to the next module for example (see chapter 12.4.3 and 12.4.3.2). Note that the limit values of the plug must not be exceeded and that voltage drops may occur in the plug.

→ Make sure to connect the cables correctly to the cable plug!
→ Use only suitable cables for the line plug.

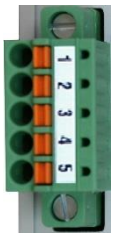
Signal Description:
P24... Power supply voltage (Nominal voltage = +24 V)
M24... Reference potential

9.2 CAN

Device connector: Phoenix Contact PCB header MC 1,5/5-GF-3,81
Cable plug: Phoenix Contact pluggable connector FK-MCP 1,5/5-STF-3,81,
Push-in spring-connection, 3,81 mm pitch
Phoenix Contact Order No.: 1851261, included in delivery
For conductor connection and conductor cross section see page 77.

Pin Position:

(cable plug)



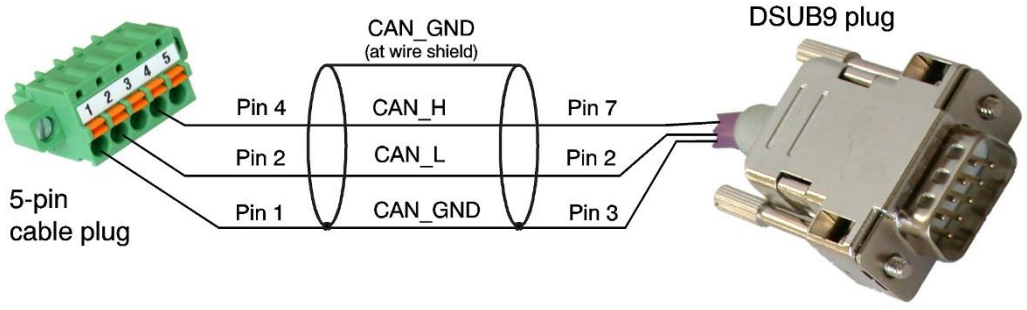
Pin Assignment:

Imprint	Signal	Pin
G	CAN_GND	1
L	CAN_L	2
Sh	Shield	3
H	CAN_H	4
●	-	5

Signal Description:

CAN_L, CAN_H …	CAN signal lines
CAN_GND …	Reference potential of the local CAN physical layer To ensure reliable CAN communication, this pin must always be connected!
Shield …	Pin for line shield connection if a separate cable shield is present/used (Connected to the potential of the DIN rail)
- …	Reserved, do not connect!

Recommendation of an adapter cable from 5-pin cable plug (here PCB connector FK-MCP1,5/5-STF_3,81 with spring-cage-connection) to DSUB9 connector:



The assignments of the DSUB9 connector and the cable plug are designed according to CiA 106.

INFORMATION
esd offers assembled CAN cables according to recommendations of CiA 303 part1 and CiA 106 as accessories, see Order Information, page 96.

9.2.1 CAN Port

The CAN bus signals are galvanically isolated from the other signals via digital isolator and DC/DC-converter.

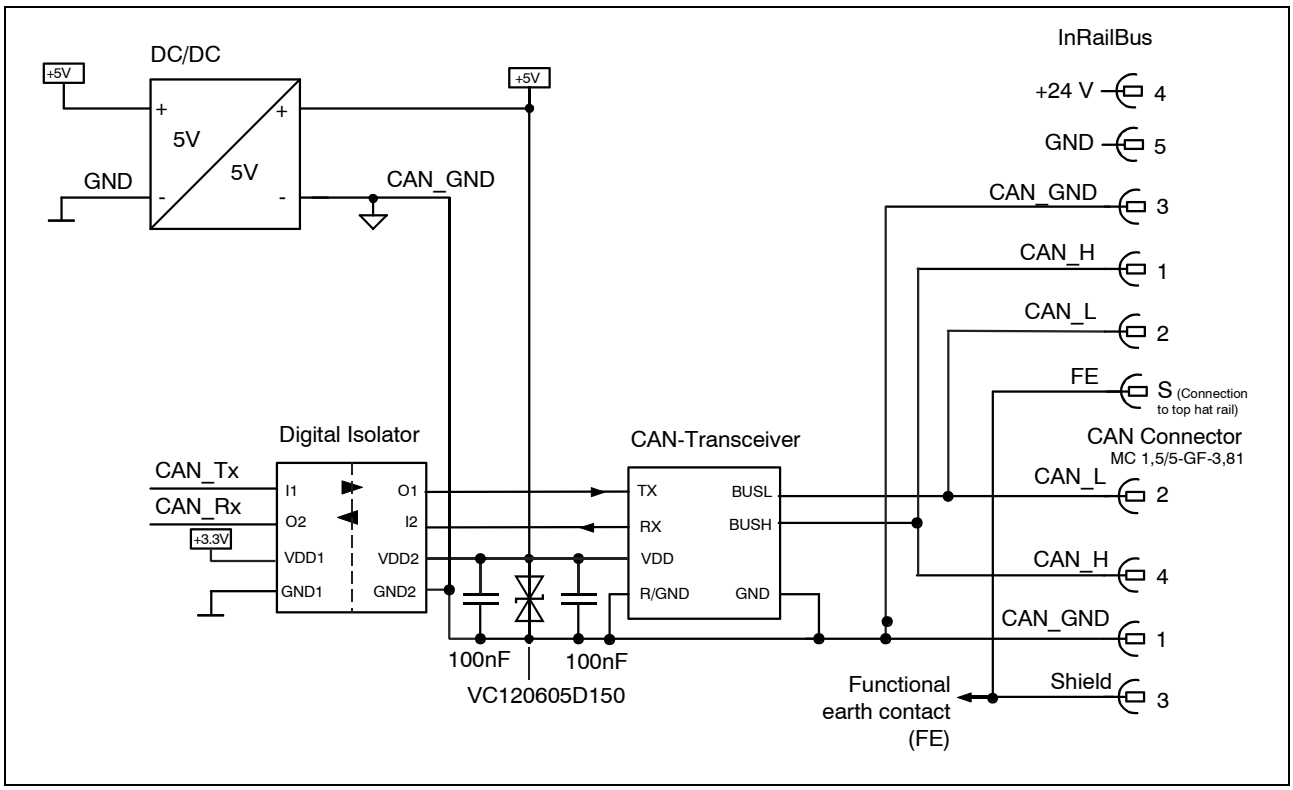


Figure 39: CAN port

The CAN port can be connected via the CAN connector or optionally via InRailBus (connector assignment see page 12

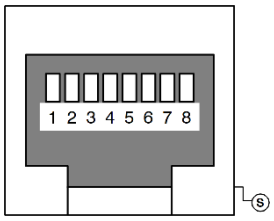
9.3 24 V and CAN via InRailBus

The power supply voltage and the CAN bus can optionally be fed via the InRailBus.
Use the mounting-rail bus connector (CAN-CBX-TBus) for the connection via the InRailBus, see Order Information InRailBus Accessories (page 88).
Read and follow the instructions for connecting power supply and CAN signals via InRailBus (see from page 89)!

9.4 Ethernet 100BASE-TX (ETH)

Device Connector: RJ45 socket, 8-pin
According to IEEE 802.3 ,
“Table 25–2—Twisted-pair MDI contact assignments”

Pin Position:



Pin Assignment:

Pin	Signal	Meaning
1	TxD+	Transmit Data +
2	TxD-	Transmit Data -
3	RxD+	Receive Data +
4	-	-
5	-	-
6	RxD-	Receive Data -
7	-	-
8	-	-

S	Shield	

Signal Description:

TxD+/-, RxD+/- ...	Ethernet data lines
- ...	Reserved for future applications, do not connect!
Shield...	Case shield, connected with the front panel of the CAN-EtherCAT/2

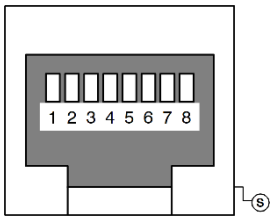
NOTICE
Cables of category CAT5e or higher must be used to grant the function in networks with 100 Mbit/s. esd grants the EU conformity of the product if the wiring is carried out with shielded twisted pair cables of category SF/UTP or better.

9.5 EtherCAT 100BASE-TX (IN, OUT)

Both EtherCAT ports have the same pin-assignment, each for the corresponding EtherCAT port.

Device connector: Dual port RJ45 sockets, 1X2, 8-pin
Ethernet 100BASE-TX, according to IEEE 802.3-2015,
“Table 25–2—Twisted-pair MDI contact assignments”

Pin Position:



Pin Assignment:

Pin	Signal	Meaning
1	Tx0+ (TxD+)	Transmit Data +
2	Tx0- (TxD-)	Transmit Data -
3	Rx0+ (RxD+)	Receive Data +
4	-	-
5	-	-
6	Rx0- (RxD-)	Receive Data -
7	-	-
8	-	-

S	Shield	

Signal Description:

Tx0+/-, Rx0+/- ...	Data lines of EtherCAT port
- ...	Reserved for future applications, do not connect!
Shield...	Case shield, connected with the front panel of the CAN-EtherCAT/2

NOTICE
Permissible cable types: Cables of category 5e or higher must be used to grant the function in networks with up to 100 Mbits/s.
esd grants the EU conformity of the product if the wiring is carried out with shielded twisted pair cables of class SF/UTP or better.

9.6 Conductor Connection / Conductor Cross Section

The following table contains an extract of the technical data of the cable plugs.

Characteristics	***Connector Type[3]***	
	Power Supply Voltage 24 V Cable connector	CAN Cable connector
Connector type plug component	FKCT 2,5/.-ST KMGY	FK-MCP 1,5/5-STF-3,81
Connection method	Push-in spring connection	Push-in spring connection
Stripping length	10 mm	9 mm
Nominal cross section	2.5 mm²	1.5 mm²
Conductor cross section rigid.	0.2 mm² … 2.5 mm²	0.14 mm² … 1.5 mm²
Conductor cross section flexible	0.2 mm² … 2.5 mm²	0.14 mm² … 1.5 mm²
Conductor cross section AWG	24 … 12	26 … 16
Conductor cross section flexible, with ferrule without plastic sleeve	0.25 mm² … 2.5 mm²	0.25 mm² … 1.5 mm²
Conductor cross section flexible, with ferrule with plastic sleeve	0.25 mm² … 2.5 mm²	0.25 mm² … 0.75 mm²
2 conductors with same cross section, stranded, TWIN ferrules with plastic sleeve, min./max.	0.5 mm² … 1.5 mm²	not allowed

3 Technical Data from Phoenix Contact website, printed circuit board connector, plug component

10 Correct Wiring of Galvanically Isolated CAN Networks

NOTICE
This chapter applies to CAN networks with bit rates up to 1 Mbit/s.
If you work with higher bit rates, as for example used for CAN FD, the information given in this chapter must be examined for applicability in each individual case.
For further information refer to the CiA® CAN FD guidelines and recommendations (https://www.can-cia.org/).

For the CAN wiring all applicable rules and regulations (EU, DIN), such as regarding electromagnetic compatibility, security distances, cable cross-section or material, must be obeyed.

10.1 CAN Wiring Standards

The flexibility in CAN network design is a major strength of the various extensions based on the original CAN standard ISO 11898-2, such as CANopen®, ARINC825, DeviceNet® and NMEA2000. However, taking advantage of this flexibility absolutely requires a network design that considers the interactions of all network parameters.

In some cases, the CAN organizations have adapted the scope of CAN in their specifications to enable applications outside the ISO 11898 standard. They have imposed system-level restrictions on data rate, line length and parasitic bus loads.

However, when designing CAN networks, a margin must always be planned for signal losses over the entire system and cabling, parasitic loads, network imbalances, potential differences against earth potential, and signal integrities. **Therefore, the maximum achievable number of nodes, bus lengths and stub lengths may differ from the theoretically possible number!**

esd has limited its recommendations for CAN wiring to the specifications of ISO 11898-2.
A description of the special features of the derived specifications CANopen, ARINC825, DeviceNet, and NMEA2000 is omitted here.

The consistent compliance with the ISO 11898-2 standard offers significant advantages:

1. Reliable operation due to proven design specifications
2. Minimization of error sources due to sufficient distance to the physical limits.
3. Easy maintenance because there are no "special cases" to consider for future network modifications and troubleshooting.

Of course, reliable networks can be designed according to the specifications of CANopen, ARINC825, DeviceNet and NMEA2000, **however it is strictly not recommended to mix the wiring guidelines of the various specifications!**

10.2 Light Industrial Environment (*Single* Twisted Pair Cable)

10.2.1 General Rules

NOTICE
esd grants the EU Conformity of the product if the CAN wiring is carried out with at least single shielded **single** twisted pair cables that match the requirements of ISO 11898-2. Single shielded *double* twisted pair cable wiring as described in chapter 10.3 ensures the EU Conformity as well.

The following **general rules** for CAN wiring with single shielded *single* twisted pair cable should be followed:

1	A suitable cable type with a wave impedance of about 120 Ω ±10% with an adequate conductor cross-section (≥ 0.22 mm²) must be used. The voltage drop over the wire must be considered.
2	For light industrial environment use at least a two-wire CAN cable, the wires of which must be assigned as follows: • Two twisted wires must be assigned to the data signals (CAN_H, CAN_L). • The cable shield must be connected to the reference potential (CAN_GND).
3	The reference potential CAN_GND must be connected to the functional earth (FE) at exactly **one** point.
4	A CAN bus line must not branch (exception: short cable stubs) and must be terminated with the characteristic impedance of the line (generally 120 Ω ±10%) at both ends (between the signals CAN_L and CAN_H and **not** at CAN_GND).
5	Keep cable stubs as short as possible (l < 0.3 m).
6	Select a working combination of bit rate and cable length.
7	Keep away cables from disturbing sources. If this cannot be avoided, double shielded wires are recommended.

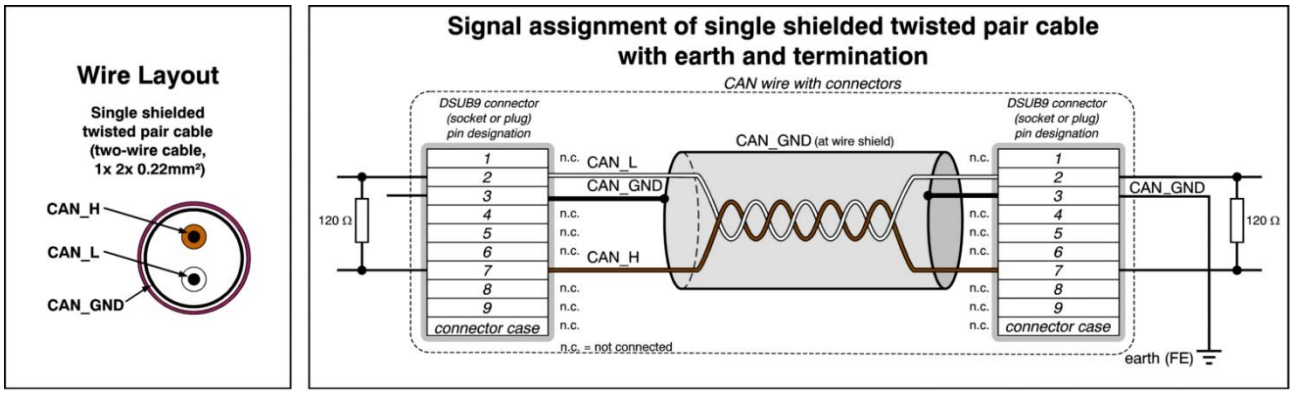


Figure 40: CAN wiring for light industrial environment

10.2.2 Cabling

- To connect CAN devices with just one CAN connector per net use a short stub (< 0.3 m) and a T-connector (available as accessory). If these devices are located at the end of the CAN network, the CAN terminator “CAN-Termination-DSUB9” can be used.

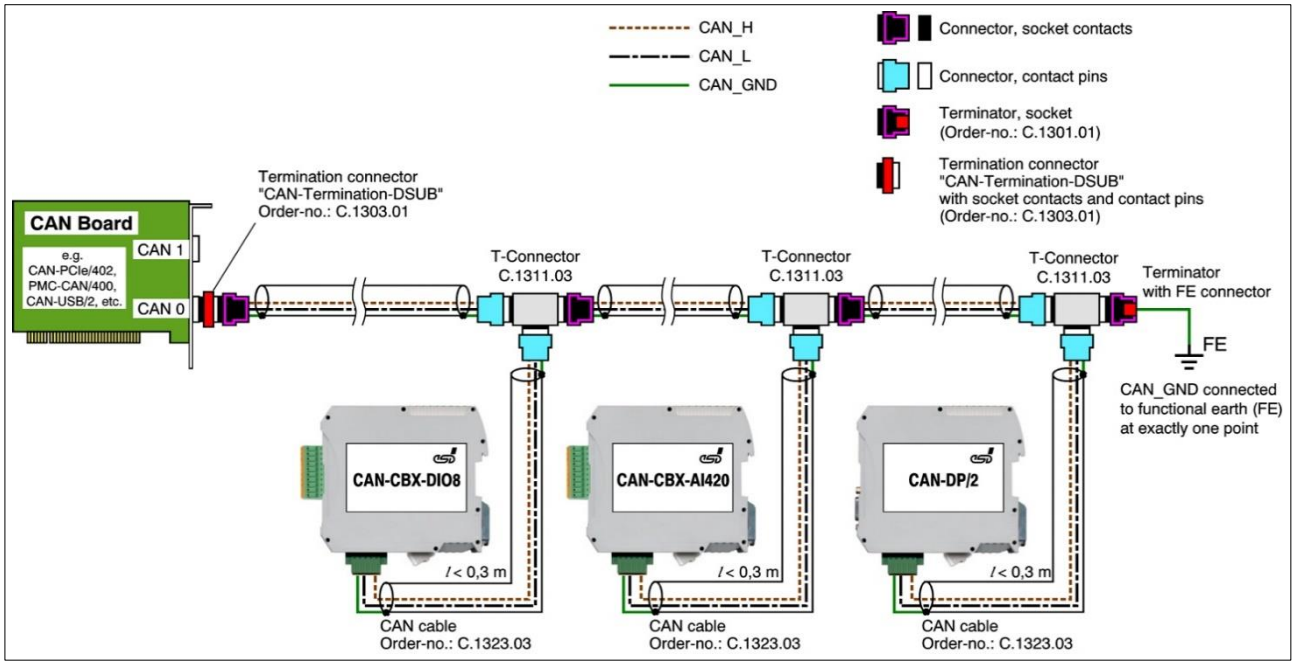


Figure 41: Example for proper wiring with single shielded single twisted pair wires

10.2.3 Branching

4. In principle the CAN bus must be realized in a line. The nodes are connected to the main CAN bus line via short cable stubs. This is normally realised by so called T-connectors. esd offers the CAN-T-Connector (Order No.: C.1311.03)
- If a mixed application of single twisted and double twisted cables cannot be avoided, ensure that the CAN_GND line is not interrupted!
- Deviations from the bus structure can be realized by using repeaters.

10.2.4 Termination Resistor

7. A termination resistor must be connected at both ends of the CAN bus.
   If an integrated CAN termination resistor is connected to the CAN interface at the end of the CAN bus, this integrated termination must be used instead of an external CAN termination resistor.
- 9-pole DSUB-termination connectors with integrated termination resistor and pin contacts and socket contacts are available from esd (order no. C.1303.01).
- For termination of the CAN bus and grounding of the CAN_GND, DSUB terminators with pin contacts (order no. C.1302.01) or socket contacts (order no. C.1301.01) and with additional functional earth contact are available.

10.3 Heavy Industrial Environment (Double Twisted Pair Cable)

10.3.1 General Rules

The following **general rules** for the CAN wiring with single shielded *double* twisted pair cable should be followed:

1	A suitable cable type with a wave impedance of about 120 Ω ±10% with an adequate conductor cross-section (≥ 0.22 mm²) must be used. The voltage drop over the wire must be considered.
2	For heavy industrial environment use a four-wire CAN cable, the wires of which must be assigned as follows: • Two twisted wires must be assigned to the data signals (CAN_H, CAN_L) and • The other two twisted wires must be assigned to the reference potential (CAN_GND). • The cable shield must be connected to functional earth (FE) at least at one point.
3	The reference potential CAN_GND must be connected to the functional earth (FE) at exactly **one** point.
4	A CAN bus line must not branch (exception: short cable stubs) and must be terminated with the characteristic impedance of the line (generally 120 Ω ±10%) at both ends (between the signals CAN_L and CAN_H and **not** to CAN_GND).
5	Keep cable stubs as short as possible (l < 0.3 m).
6	Select a working combination of bit rate and cable length.
7	Keep away CAN cables from disturbing sources. If this cannot be avoided, double shielded cables are recommended.

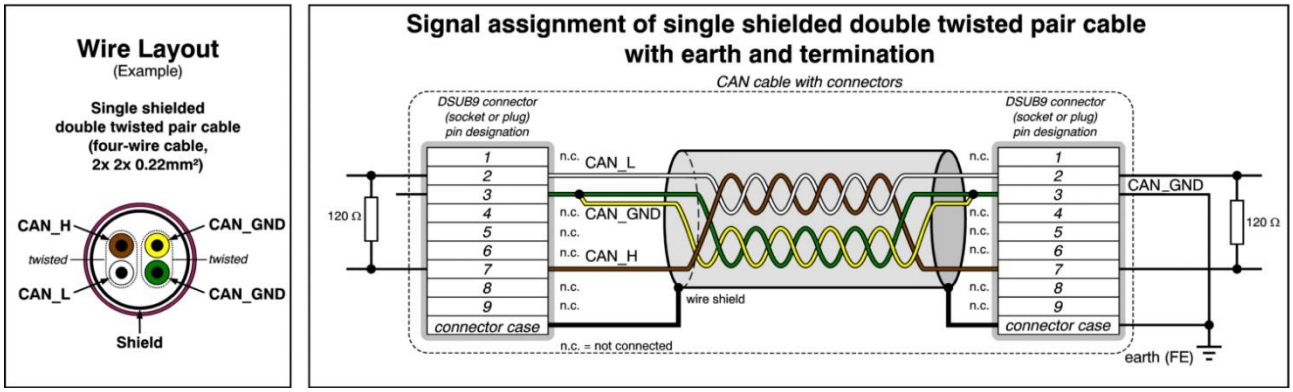


Figure 42: CAN wiring for heavy industrial environment

10.3.2 Device Cabling

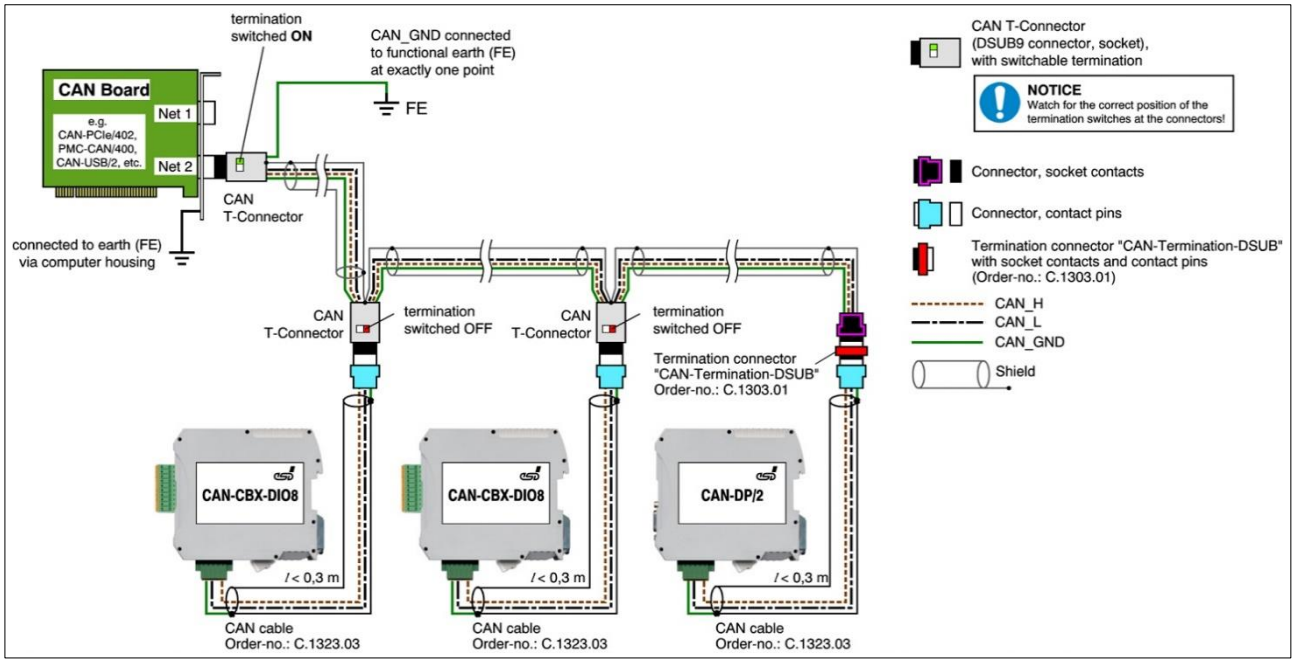


Figure 43: Example of proper wiring with single shielded double twisted pair cables

10.3.3 Branching

- In principle, the CAN bus must be realized in a line. The nodes are connected to the main CAN bus line via short cable stubs. This is usually realised via so called T-connectors. When using esd's CAN-T-Connector (order no.: C.1311.03) in heavy industrial environment and with four-wire twisted cables, it must be noted that the shield potential of the conductive DSUB housing is not looped through this type of T-connector. This interrupts the shielding. Therefore, you must take appropriate measures to connect the shield potentials, as described in the manual of the CAN-T-Connector. For further information on this, please refer to the CAN-T-Connector Manual (order no.: C.1311.21).
  Alternatively, a T-connector can be used, in which the shield potential is looped through, for example the DSUB9 connector from ERNI (ERBIC CAN BUS MAX, order no.:154039).
- If a mixed application of single twisted and double twisted cables cannot be avoided, ensure that the CAN_GND line is not interrupted!
- Deviations from the bus structure can be realized by using repeaters.

10.3.4 Termination Resistor

10. A termination resistor must be connected at both ends of the CAN bus.
    If an integrated CAN termination resistor is connected to the CAN interface at the end of the CAN bus, this integrated termination must be used instead of an external CAN termination resistor.
11. 9-pole DSUB-termination connectors with integrated termination resistor and pin contacts and socket contacts are available from esd (order no. C.1303.01).
12. 9-pole DSUB-connectors with integrated switchable termination resistor can be ordered for example from ERNI (ERBIC CAN BUS MAX, socket contacts, order no.:154039).

10.4 Electrical Grounding

13. For CAN devices with galvanic isolation the CAN_GND must be connected between the CAN devices.
14. CAN_GND should be connected to the earth potential (FE) at **exactly one** point of the network.
15. Each *CAN interface with electrical connection to earth potential* acts as a grounding point. For this reason, it is recommended not to connect more than one *CAN device with electrical connection to earth potential.*
16. Grounding can be done for example at a termination connector (e.g. order no. C.1302.01 or C.1301.01).

10.5 Bus Length

The bus length of a CAN network must be adapted to the set bit rate. The maximum values result from the fact that the time required for a bit to be transmitted in the bus system is shorter the higher the transmission rate is. However, as the line length increases, so does the time it takes for a bit to reach the other end of the bus. It should be noted that the signal is not only transmitted, but the receiver must also respond to the transmitter within a certain time. The transmitter, in turn, must detect any change in bus level from the receiver(s). Delay times on the line, the transceiver, the controller, oscillator tolerances and the set sampling time must be considered.
In the following table you will find guide values for the achievable bus lengths at certain bit rates.

Bit Rate [kbit/s]	Theoretical values of reachable wire length **with esd interface** l_{max} [m]	**CiA recommendations** (07/95) for reachable wire lengths l_{min} [m]	Standard values of the cross-section according to CiA 303-1 [mm²]
1000	37	25	0.25 to 0.34
800	59	50	0.34 to 0.6
666.$\overline{6}$	80	-	
500	130	100	
333.$\overline{3}$	180	-	
250	270	250	
166	420	-	0.5 to 0.6
125	570	500	
100	710	650	0.75 to 0.8
83.$\overline{3}$	850	-	
66.$\overline{6}$	1000	-	
50	1400	1000	
33.$\overline{3}$	2000	-	not defined in CiA 303-1
20	3600	2500	
12.5	5400	-	
10	7300	5000	

Table 22: Recommended cable lengths at typical bit rates (with esd-CAN interfaces)

Optical couplers are delaying the CAN signals. esd modules typically achieve a wire length of 37 m at 1 Mbit/s within a proper terminated CAN network without impedance disturbances, such as those caused by cable stubs > 0.3 m.

NOTICE
Please note that the cables, connectors, and termination resistors used in CANopen networks shall meet the requirements defined in ISO 11898-2.
In addition, further recommendations of the CiA, like standard values of the cross section, depending on the cable length, are described in the CiA recommendation CiA 303-1 (see CiA 303 CANopen Recommendation - Part 1: “Cabling and connector pin assignment,” Version 1.9.0, Table 2). Recommendations for pin-assignment of the connectors are described in CiA 106: “Connector pin-assignment recommendations ”.

10.6 Examples for CAN Cables

esd recommends the following two-wire and four-wire cable types for CAN network design. These cable types are used by esd for ready-made CAN cables, too.

10.6.1 Cable for Light Industrial Environment Applications (Two-Wire)

Manufacturer	Cable Type
U.I. LAPP GmbH Schulze-Delitzsch-Straße 25 70565 Stuttgart Germany www.lappkabel.com	e.g. UNITRONIC ®-BUS CAN UL/CSA (1x 2x 0.22) (UL/CSA approved) Part No.: 2170260 UNITRONIC ®-BUS-FD P CAN UL/CSA (1x 2x 0.25) (UL/CSA approved) Part No.: 2170272
ConCab GmbH Äußerer Eichwald 74535 Mainhardt Germany www.concab.de	e. g. BUS-PVC-C (1x 2x 0.22 mm²) Order No.: 93 022 016 (UL appr.) BUS-Schleppflex-PUR-C (1x 2x 0.25 mm²) Order No.: 94 025 016 (UL appr.)

10.6.2 Cable for Heavy Industrial Environment Applications (Four-Wire)

Manufacturer	Cable Type
U.I. LAPP GmbH Schulze-Delitzsch-Straße 25 70565 Stuttgart Germany www.lappkabel.com	e.g. UNITRONIC ®-BUS CAN UL/CSA (2x 2x 0.22) (UL/CSA approved) Part No.: 2170261 UNITRONIC ®-BUS-FD P CAN UL/CSA (2x 2x 0.25) (UL/CSA approved) Part No.: 2170273
ConCab GmbH Äußerer Eichwald 74535 Mainhardt Germany www.concab.de	e. g. BUS-PVC-C (2x 2x 0.22 mm²) Order No.: 93 022 026 (UL appr.) BUS-Schleppflex-PUR-C (2x 2x 0.25 mm²) Order No.: 94 025 026 (UL appr.)

INFORMATION
Ready-made CAN cables with standard or custom length can be ordered from **esd**.

11 CAN Troubleshooting Guide

The CAN Troubleshooting Guide is a guide to finding and eliminating the most common problems and errors when setting up CAN bus networks and CAN-based systems.

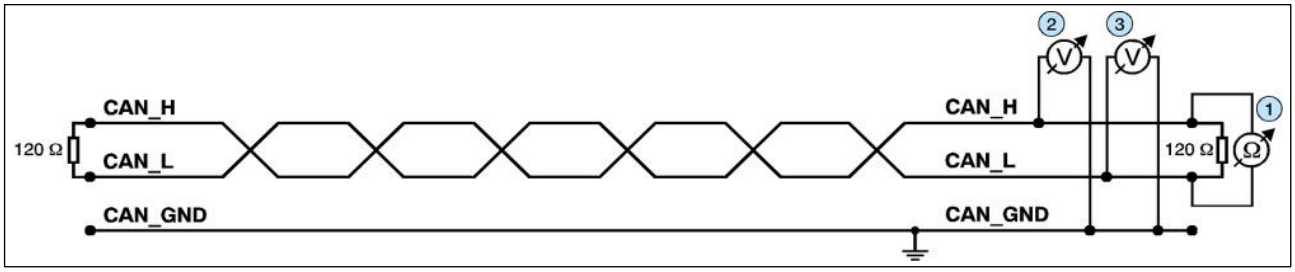


Figure 44: Simplified diagram of a CAN network

Termination
The bus termination is used to match impedance of a node to the impedance of the bus line used. If the impedance is mismatched, the transmitted signal is not completely absorbed by the load and will be partially reflected back into the transmission line.
If the impedances of the sources, transmission lines and loads are equal, the reflections are avoided. This test measures the total resistance of the two CAN data lines and the connected terminating resistors.

To **test this, please proceed as follows:**

1. Switch off the supply voltages of all connected CAN nodes.
2. Measure the DC resistance between CAN_H and CAN_L at one end of the network, measuring point ① (see figure above).

Expected result:
The measured value should be between 50 Ω and 70 Ω.

Possible causes of error:

- If the determined value is below 50 Ω, please make sure that:
  - There is no **short circuit** between CAN_H and CAN_L wiring.
  - **No more than two** terminating resistors are connected.
  - The transceivers of the individual nodes are not defective.

- If the determined value is higher than 70 Ω, please make sure that:
  - All CAN_H and CAN_L lines are correctly connected.
  - Two terminating resistors of 120 Ω each are connected to your CAN network (one at each end).

11.1 Electrical Grounding

The CAN_GND of the CAN network should be connected to the functional earth potential (FE) at only **one** point. This test indicates whether the CAN_GND is grounded at one or more points.

Please note that this test can only be performed with galvanically isolated CAN nodes.

To test this, please proceed as follows:

1. Disconnect the CAN_GND from the earth potential (FE).
2. Measure the DC resistance between CAN_GND and earth potential (see figure on the right).

Do not forget to reconnect CAN_GND to earth potential after the test!

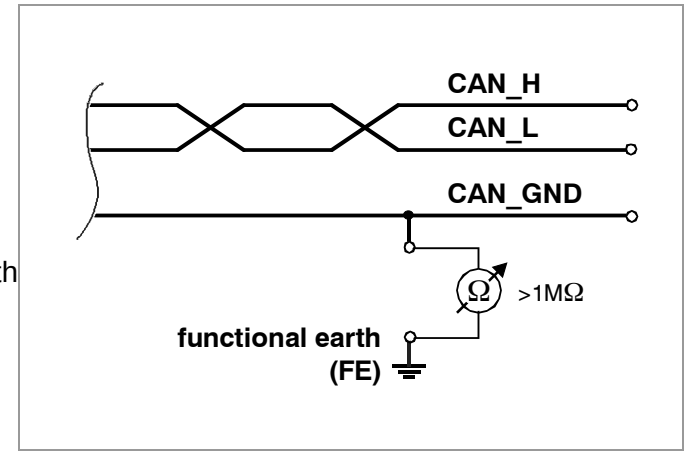


Figure 45: Simplified schematic diagram of ground test measurement

Expected result:
The measured resistance should be greater than 1 MΩ. If it is smaller, please search for additional grounding of the CAN_GND wires.

11.2 Short Circuit in CAN Wiring

A CAN bus might possibly still be able to transmit data even if CAN_GND and CAN_L are short-circuited. However, this will usually cause the error rate to rise sharply.
Ensure that there is no short circuit between CAN_GND and CAN_L!

11.3 Correct Voltage Levels on CAN_H and CAN_L

Each node contains a CAN transceiver that outputs differential signals. When the network communication is idle the CAN_H and CAN_L voltages are approximately 2.5 V measured to CAN_GND. Defective transceivers can cause the idle voltages to vary and disrupt network communication.

To test for defective transceivers, please proceed as follows:

1. Switch on all supply voltages.
2. Terminate all network communication.
3. Measure the DC voltage between CAN_H and CAN_GND, measuring point ②.
   (See “Simplified diagram of a CAN network” on previous page).
4. Measure the DC voltage between CAN_L and CAN_GND, measuring point ③.
   (See “Simplified diagram of a CAN network” on previous page).

Expected result:
The measured voltage should be between 2.0 V and 3.0 V.

Possible causes of error:

- If the voltage is lower than 2.0 V or higher than 3.0 V, it is possible that one or more nodes have defective transceivers.
    - If the voltage is lower than 2.0 V, please check the connections of the CAN_H and CAN_L lines.
- To find a node with a defective transceiver within a network, please check individually the resistances of the CAN transceivers of the nodes (see next section).

11.4 CAN Transceiver Resistance Test

CAN transceivers have circuits that control CAN_H and CAN_L. Experience shows that electrical damage can increase the leakage current in these circuits.

To measure the current leakage through the CAN circuits, please use an ohmmeter and proceed as follows:

1. Switch **off** the node ④ and **disconnect** it from the CAN network.
   (See figure below.)
2. Measure the DC resistance between CAN_H and CAN_GND, measuring point ⑤
   (See figure below.)
3. Measure the DC resistance between CAN_L and CAN_GND, measuring point ⑥
   (See figure below.)

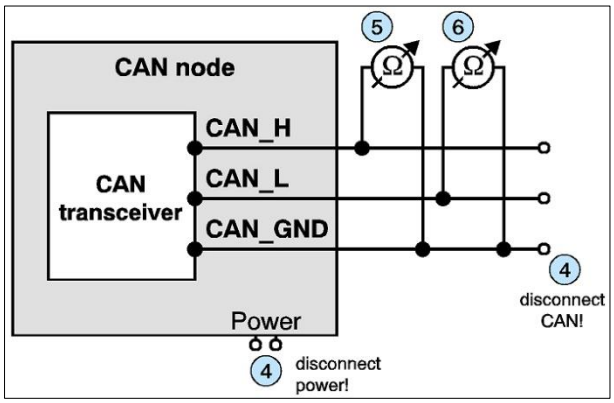


Figure 46: Measuring the internal resistance of CAN transceivers

Expected result:
The measured resistance should be greater than 10 kΩ for each measurement.

Possible causes of error:

- If the resistance is significantly lower, the CAN transceiver may be defective.
- Another indication of a defective CAN transceiver is a very high deviation of the two measured input resistances (>> 200 %).

11.5 Support by esd

If you have followed the troubleshooting steps in this troubleshooting guide and still cannot find a solution to your problem, our support team can help.
Please contact our support by email to support@esd.eu or by phone **+49-511-37298-130.**

12 Optional InRailBus

12.1 Order Information InRailBus Accessories

Type	Properties	Order No.
Accessories		
 **CAN-CBX-TBus**	DIN-rail bus connector of the CBX-InRailBus for CAN-CBX modules (ME 22,5 TBUS 1,5/ 5-ST-3,81 KMGY)	C.3000.01
 **CAN-CBX-TBus-Connector-Socket**	Terminal plug of the CBX-InRailBus for the connection of the +24V power supply voltage and the CAN interface (MCVR 1,5/5-ST-3,81 AU), socket contacts	C.3000.02
 **CAN-CBX-TBus-Connector-Plug**	Terminal plug of the CBX-InRailBus for the connection of the +24V power supply voltage and the CAN-Interface (IMC 1,5/ 5-ST-3,81 AU), pin contacts	C.3000.03

12.2 Conductor Connection/Conductor Cross Section

The following table contains an extract of the technical data of the cable plugs.

Characteristics	*Connector Type*[4]	
	CAN-CBX-TBus-Connector-Socket	CAN-CBX-TBus-Connector-Plug
Connector type plug component	MCVR 1,5/5-ST-3,81 AU	IMC 1,5/ 5-ST-3,81 AU
Connection method	Screw connection with tension sleeve	Screw connection with tension sleeve
Stripping length	7 mm	7 mm
Nominal cross section	1.5 mm²	1.5 mm²
Conductor cross section rigid.	0.14 mm² … 1.5 mm²	0.14 mm² … 1.5 mm²
Conductor cross section flexible	0.14 mm² … 0.5 mm²	0.14 mm² … 1.5 mm²
Conductor cross section AWG	28 … 16	28 … 16
Conductor cross section flexible, with ferrule without plastic sleeve	0.25 mm² … 1.5 mm²	0.25 mm² … 1.5 mm²
Conductor cross section flexible, with ferrule with plastic sleeve	0.25 mm² … 0.5 mm²	0.25 mm² … 0.5 mm²
2 conductors with same cross section, stranded, TWIN ferrules with plastic sleeve, min./max.	0.5 mm² ... 0.5 mm²	0.5 mm² ... 0.5 mm²

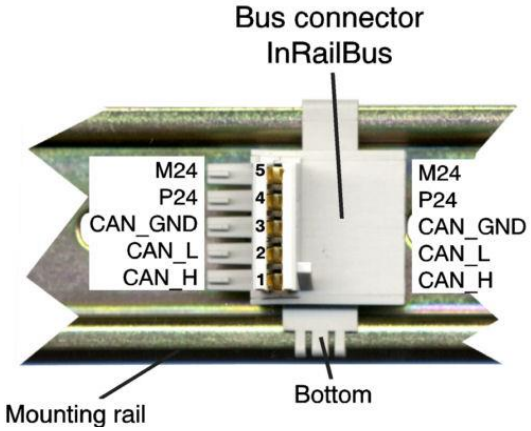
4 Technical Data from Phoenix Contact website, printed circuit board connector, plug component

12.3 Connector Assignment 24V and CAN via InRailBus (Option)

DANGER
The CAN-EtherCAT/2 is a device of protection class III according to DIN EN IEC 61140 and may only be operated on supply circuits that offer sufficient protection against dangerous voltages.

Connector type: Mounting-rail bus connector of the CBX-InRailBus
Phoenix Contact ME 22,5 TBUS 1,5/5-ST-3,81 KMGY

Connector View:



Pin Assignment:

Pin	Signal
5	M24 (GND)
4	P24 (+24 V)
3	CAN_GND
2	CAN_L
1	CAN_H
S	FE (PE_GND)

Signal Description:

CAN_L, CAN_H ... CAN signals
CAN_GND ... reference potential of the local CAN-Physical layers
P24... power supply voltage +24 V
M24... reference potential
FE... functional earth contact (EMC) (connected to mounting rail potential)

12.4 Using InRailBus (Option)

This chapter describes the installation of the module using InRailBus as an example for CAN-CBX-modules.

12.4.1 Installation of the Module when using the InRailBus Connector

If the CAN bus signals and the power supply voltage shall be fed via the InRailBus, please proceed as follows:

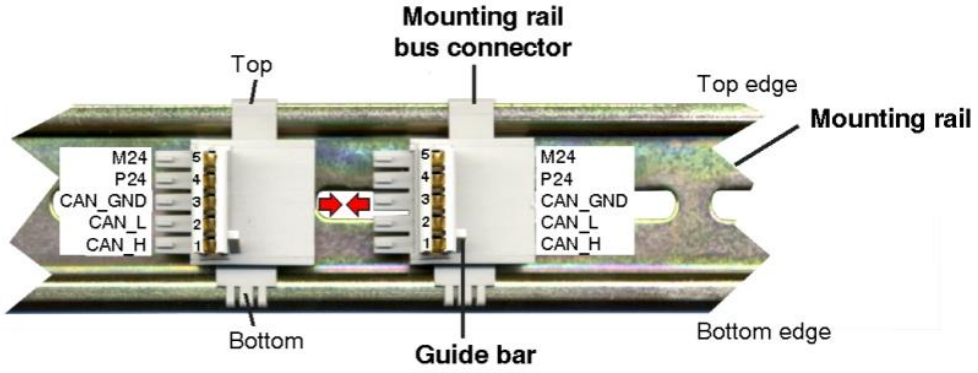


Figure 47: Mounting rail with bus connector

1. Position the InRailBus connector on the mounting rail and snap it onto the mounting rail using slight pressure. Plug the bus connectors together to contact the communication and power signals (in parallel with one). The bus connectors can be plugged together before or after the CAN-CBX module is plugged on.

2. Hold the CAN-CBX module tilted backwards at a slight angle and place it on the bus connector so that the DIN rail guideway is placed on the top edge of the mounting rail.

Figure 48: Mounting CAN-CBX modules

3. Now swivel the CAN-CBX module onto the mounting rail by moving the module downwards according to the direction of the arrow in Figure 48. The housing is mechanically guided by the guide bar of the bus connector.

4. When mounting the CAN-CBX module the moveable snap-on foot snaps onto the bottom edge of the mounting rail.
   The module is now firmly seated on the mounting rail and is connected to the InRailBus via the bus connector. If necessary, connect the bus connectors to each other and connect the +24 V supply voltage and the CAN interface to the InRailBus as described below.

Figure 49: Mounted CAN-CBX module

12.4.2 Connecting via the InRailBus

To connect the power supply and the CAN signals via the InRailBus, a terminal plug is needed. The terminal plug is not included in the scope of delivery and must be ordered separately (order no.: C.3000.02, see order information for InRailBus Accessories).

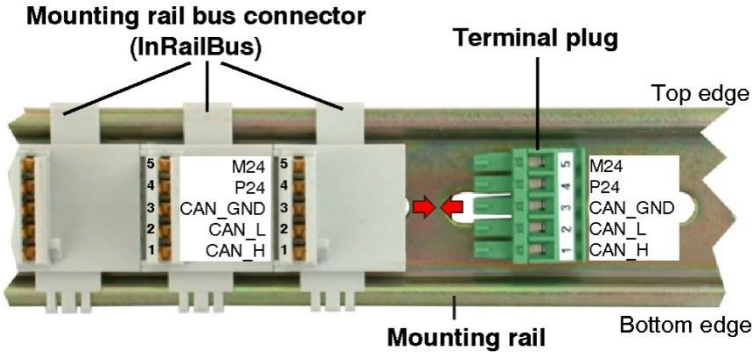


Figure 50: Mounting rail with InRailBus and terminal plug

Insert the terminal plug from the right into the socket side of the outer mounting rail bus connector of the InRailBus, as shown in Figure 50. Then connect the CAN interface and the supply voltage via the terminal plug.

12.4.3 Connection of the Supply Voltage

DANGER
Hazardous Voltage - **Risk of electric shock** due to unintentional contact with uninsulated live parts with high voltages inside of the system into which the CAN-CBX-module is to be integrated.

→ Read the safety instructions at the beginning of this document (from page 5) carefully before you start with the hardware installation!
→ Ensure the absence of voltage before starting any electrical work.
→ Switch off the power supply, before you connect it to the system.

DANGER
The CAN-EtherCAT/2 is a device of protection class III according to DIN EN IEC 61140 and may only be operated on supply circuits that offer sufficient protection against dangerous voltages.

There are two ways to feed the 24 V power supply voltage into the CBX station:
- via the terminal plug of the InRailBus, see 12.4.3.1
- via the 24 V connector of the first module in the CBX station, see 12.4.3.2

NOTICE
The two connections for the 24 V power supply (via InRailBus or 24 V connector) are connected internally and must not be supplied by two independent power sources at the same time!
Connecting 24 V at both connectors will cause damage to the CAN-CBX module.
Also read the chapter on the assignment of the 24 V connector for further information.

12.4.3.1 Connection of the Power Supply Voltage via InRailBus

NOTICE
If you feed the 24V power supply via the terminal plug of the InRailBus (see Figure below), the maximum load current must not exceed $I_{MAX\_LOAD\_InRailBus}$ = 8 A.

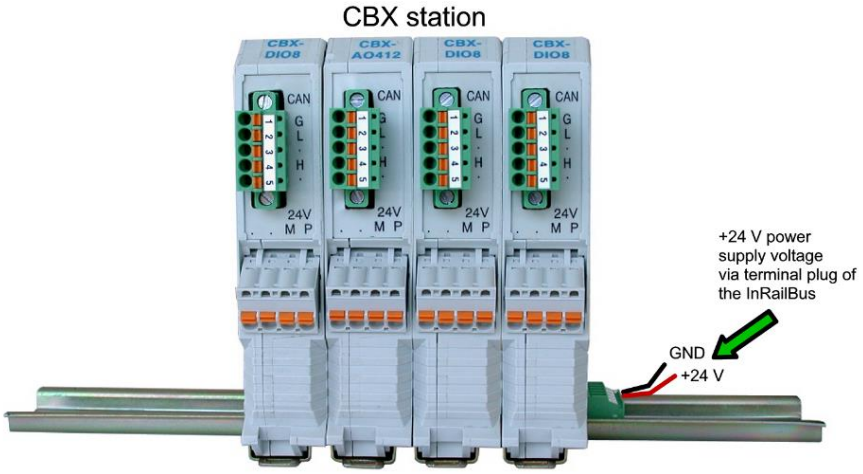


Figure 51: Connection via terminal plug

12.4.3.2 Connection of the Power Supply Voltage via 24 V Connector

NOTICE
Note that the connection between the 24V plug and the InRailBus is not designed to feed the 24V supply voltage via the plug to the InRailBus!
If the modules are mounted on the DIN rail without InRailBus connectors, it is allowed to bridge the power supply from one module to another (see Figure below), but the maximum load current must not exceed $I_{MAX\_LOAD\_24V\_plug}$ = 2 A.

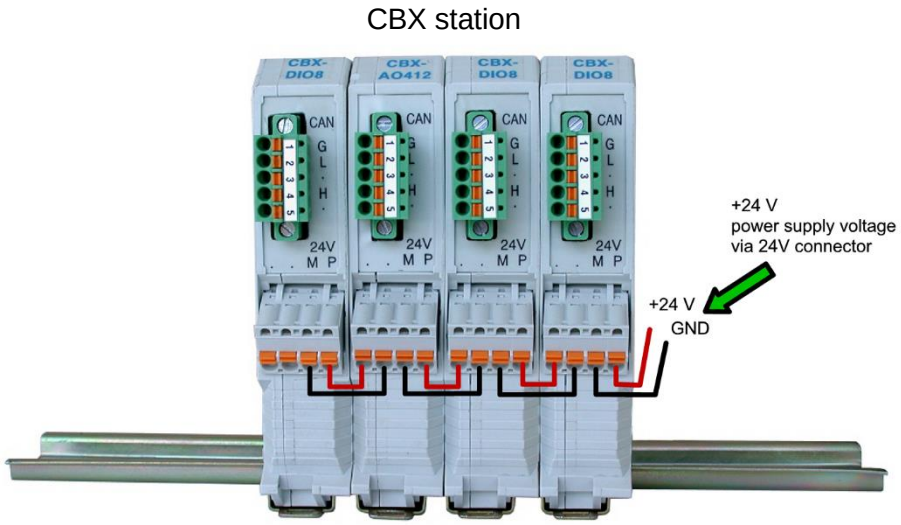


Figure 52: Connection via 24V Connector

12.4.3.3 Earthing of the Mounting Rail

NOTICE
The module is connected to the mounting rail via its functional earth contact. This improves the stability against electromagnetic disturbances. The mounting rail must therefore be connected to a suitable functional earth contact in the environment or in the installation. It must be ensured that the impedance of the connection is kept low.
The functional earth contact of the module does not ensure electrical safety.

12.4.4 Connection of CAN

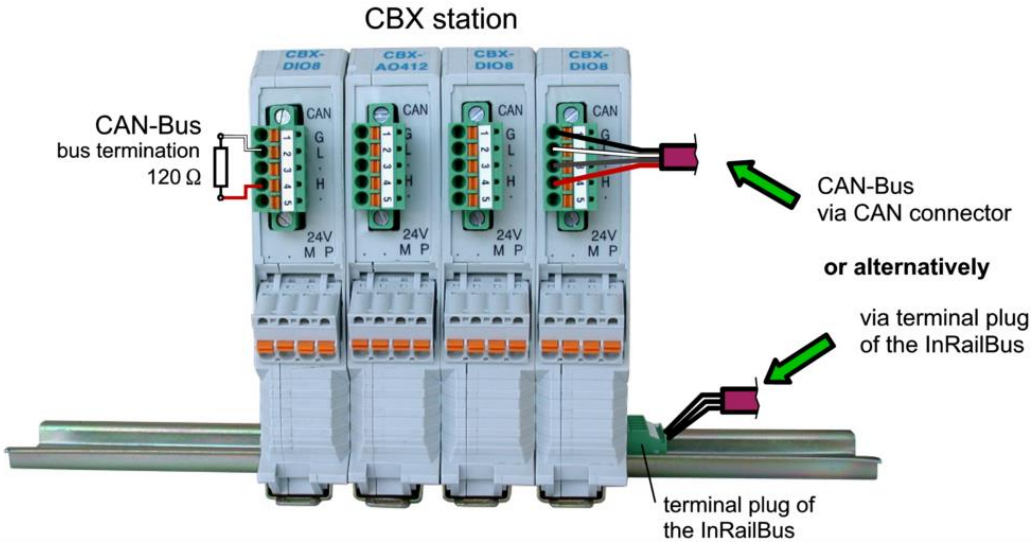


Figure 53: Connecting the CAN signals to the CAN-CBX station

In general, the CAN signals can be fed in via the InRailBus or via the CAN connector of the first CAN-CBX module in the CAN-CBX station. The signals are then connected through the CAN-CBX station via the InRailBus. The CAN signals may be lead through via the CAN connector of the CAN-CBX module mounted at the other end of the CBX station. However, the CAN signals must not be connected via the CAN connectors of the middle CAN-CBX modules of the CBX station, as this would lead to impermissible branching.
Please note that a bus terminating resistor must be connected to the CAN-CBX module located at the end of the InRailBus if the CAN bus ends there (see Figure 53).

12.4.5 Remove the CAN-CBX Module from InRailBus

If the CAN-CBX module is only connected via the InRailBus, proceed as follows when removing it:
Release the module from the mounting rail by moving the snap-on foot (see Figure 49) downwards (e.g., with a screwdriver). This releases the module from the bottom edge of the mounting rail, and it can be removed.

INFORMATION
It is possible to remove individual devices from the CBX station without interrupting the InRailBus connection, because the contact chain will not be disrupted.

13 Declaration of Conformity

EU-KONFORMITÄTSERKLÄRUNG
EU DECLARATION OF CONFORMITY

Adresse / *Address*
esd electronics gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

esd erklärt, dass das Produkt *esd declares, that the product*	Typ, Modell, Artikel-Nr. *Type, Model, Article No.*
CAN-EtherCAT/2 **CAN-EtherCAT/2-FD**	**C.2932.02** **C.2932.62**
die Anforderungen der Normen *fulfills the requirements of the standards*	**EN 61000-6-2:2005,** **EN 61000-6-3:2007/A1:2011**
gemäß folgendem Prüfbericht erfüllt. *according to test certificate.*	**EMVP No.: 0224-202307**
Das Produkt entspricht damit der EU-Richtlinie „EMV" *Therefore, the product conforms to the EU Directive 'EMC'*	**2014/30/EU**
Das Produkt entspricht den EU-Richtlinien „RoHS" *The product conforms to the EU Directives 'RoHS'*	**2011/65/EU, 2015/863/EU**

Diese Erklärung verliert ihre Gültigkeit, wenn das Produkt nicht den Herstellerunterlagen entsprechend eingesetzt und betrieben wird, oder das Produkt abweichend modifiziert wird.
This declaration loses its validity if the product is not used or run according to the manufacturer's documentation or if non-compliant modifications are made.

Name / *Name*	T. Bielert
Funktion / *Title*	QM-Beauftragter / *QM Representative*
Datum / *Date*	Hannover, 2024-07-15

Rechtsgültige Unterschrift / *authorized signature*

I:\Texte\Doku\MANUALS\CAN\CAN-EtherCAT2\EU-Konformitaetserklaehrung\CAN-EtherCAT2_EU_Declaration_of_Conformity_2024-07-15.docx

14 Order Information

14.1 Hardware

Type	Properties	Order No.
CAN-EtherCAT/2-FD	EtherCAT-to-**CAN FD** gateway for DIN rail mounting. Can be integrated as EtherCAT SubDevice with CAN FD frames as process data image or according to ETG Modular Device Profile No. 5000. Galvanically isolated **CAN FD** port according to ISO-11898 with up to 8 Mbit/s. The CAN FD port is fully backwards compatible with CAN CC The CAN-EtherCAT/2-FD is configured via the EtherCAT MainDevice and standard EtherCAT configuration tools. The module can also be used as an Ethernet switch port device for the integration of Ethernet into EtherCAT.	C.2932.62
CAN-EtherCAT/2	Successor of CAN-EtherCAT (C.2922.02) EtherCAT to **CAN CC** gateway for DIN rail mounting. Can be integrated as EtherCAT SubDevice with CAN frames as process data image or according to ETG Modular Device Profile No. 5000. Galvanically isolated **CAN CC** port according to ISO-11898 with up to 1 Mbit/s. The CAN-EtherCAT/2 is configured via the EtherCAT MainDevice and standard EtherCAT configuration tools. The module can also be used as Ethernet switch port device for the integration of Ethernet into EtherCAT.	C.2932.02
Accessories:		
CAN-Cable-S, 0.3 m (plug)	CAN cable assembly, metallised plastic housing, 0.3 m length, 1x DSUB9 plug and 3 wire end sleeves,	C.1323.03
CAN-Cable-S, 0.3 m (socket)	CAN cable assembly, metallised plastic housing, 0.3 m length,1x DSUB9 socket and 3 wire end sleeves,	C.1323.04

Table 23: Order information hardware

14.2 Manuals

PDF Manuals
For the availability of the manuals see the table below.
Please download the manuals as PDF documents from our esd website https://www.esd.eu for free.

Manuals		Order No.
CAN-EtherCAT/2-ME	Hardware manual in English	C.2932.21

Table 24: Available Manuals

Printed Manuals
If you need a printout of the manual additionally, please contact our sales team (sales@esd.eu) for a quotation. Printed manuals may be ordered for a fee.