Documentation for

EP6001 and EP6002

Serial Interface Modules (RS232 or RS422/RS485)

Version: 2.1.0
Date: 2018-10-19

BECKHOFF

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, DE102004044764, DE102007017835 with corresponding applications or registrations in various other countries.

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents: EP0851348, US6167425 with corresponding applications or registrations in various other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of instructions

In this documentation the following instructions are used.
These instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow this safety instruction directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow this safety instruction endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow this safety instruction can lead to injuries to persons.

NOTE

Damage to environment/equipment or data loss

Failure to follow this instruction can lead to environmental damage, equipment damage or data loss.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation Issue Status

Version	Comment
2.1.0	• Chapter *Notes on the documentation* updated • Structural update • RS232 level corrected • Chapter *Mounting* updated
2.0.1	• Nut torques for connectors updated
2.0.0	• Migration • EP6001-0002 added • Chapter Mounting updated • Conductor losses 7/8" added • Chapter Cabling updated • EP6001, EP6002 Status-LEDs updated • Chapter Signal connection added • EP6001-0002 - Status-LEDs added • Basics communication - EtherCAT added • EP600x-0002 - Interface modes updated • EP600x-0002 - Basic function principles updated • EP600x-0002 - Choice of the interface type updated • EP6002-0002 - Sample program 1 updated • EP6002-0002 - Sample program 2 updated • EP6001-0002 - Object overview added • EtherCAT Box accessories updated
1.3.0	• Power Connection updated
1.2.0	• Description of choosing the interface type via System Manager corrected • Object description extended • Notes on RS485 mode extended • Chapter *EtherCAT connection* updated
1.1.0	• Description of the M12 sockets corrected • Technical data updated
1.0.0	• Extended ambient temperature range • Sample programs added • Special conditions for ATEX added • Description of the power connection updated • Overview of EtherCAT cables extended
0.1	• First preliminary version for EP6002-0002

Firmware and hardware versions

This documentation refers to the firmware and hardware version that was applicable at the time the documentation was written.

The module features are continuously improved and developed further. Modules having earlier production statuses cannot have the same properties as modules with the latest status. However, existing properties are retained and are not changed, so that older modules can always be replaced with new ones.

The firmware and hardware version (delivery state) can be found in the batch number (D-number) printed on the side of the EtherCAT Box.

Syntax of the batch number (D-number)

D: WW YY FF HH	Example with D no. 29 10 02 01:
WW - week of production (calendar week)	29 - week of production 29
YY - year of production	10 - year of production 2010
FF - firmware version	02 - firmware version 02
HH - hardware version	01 - hardware version 01

2 Product overview

2.1 Module overview

EP600x-0002

Module	Signal connection	Number of channels – serial interfaces	Comment
EP6001-0002 [▶ 12]	4 x M12	1	Narrow housing
EP6002-0002 [▶ 16]	4 x M12	2	Narrow housing

2.2 EtherCAT Box - Introduction

The EtherCAT system has been extended with EtherCAT Box modules with protection class IP 67. Through the integrated EtherCAT interface the modules can be connected directly to an EtherCAT network without an additional Coupler Box. The high-performance of EtherCAT is thus maintained into each module.

The extremely low dimensions of only 126 x 30 x 26.5 mm (h x w x d) are identical to those of the Fieldbus Box extension modules. They are thus particularly suitable for use where space is at a premium. The small mass of the EtherCAT modules facilitates applications with mobile I/O interface (e.g. on a robot arm). The EtherCAT connection is established via screened M8 connectors.

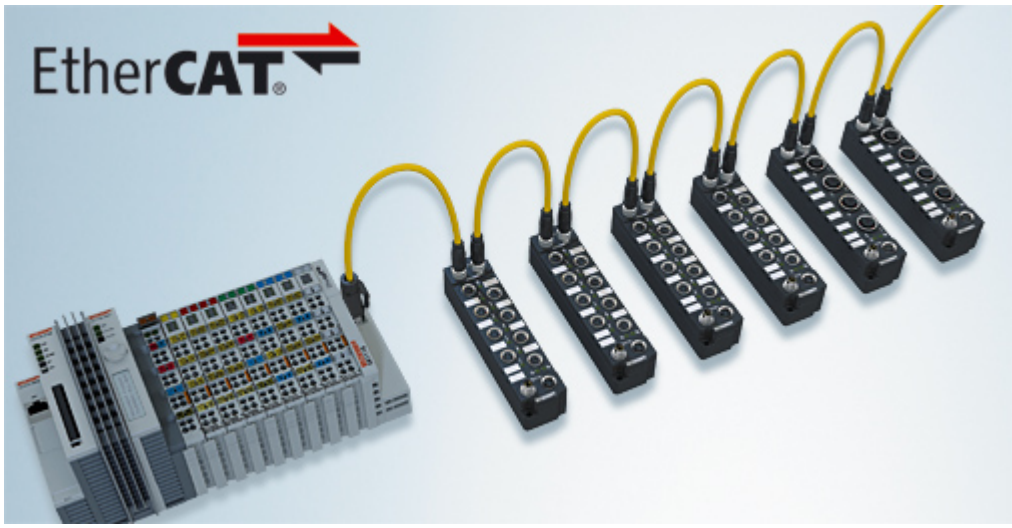


Fig. 1: EtherCAT Box Modules within an EtherCAT network

The robust design of the EtherCAT Box modules enables them to be used directly at the machine. Control cabinets and terminal boxes are now no longer required. The modules are fully sealed and therefore ideally prepared for wet, dirty or dusty conditions.

Pre-assembled cables significantly simplify EtherCAT and signal wiring. Very few wiring errors are made, so that commissioning is optimized. In addition to pre-assembled EtherCAT, power and sensor cables, field-configurable connectors and cables are available for maximum flexibility. Depending on the application, the sensors and actuators are connected through M8 or M12 connectors.

The EtherCAT modules cover the typical range of requirements for I/O signals with protection class IP67:

- digital inputs with different filters (3.0 ms or 10 µs)
- digital outputs with 0.5 or 2 A output current
- analog inputs and outputs with 16 bit resolution
- Thermocouple and RTD inputs
- Stepper motor modules

XFC (eXtreme Fast Control Technology) modules, including inputs with time stamp, are also available.

Fig. 2: EtherCAT Box with M8 connections for sensors/actuators

Fig. 3: EtherCAT Box with M12 connections for sensors/actuators

Basic EtherCAT documentation

You will find a detailed description of the EtherCAT system in the Basic System Documentation for EtherCAT, which is available for download from our website (www.beckhoff.com) under Downloads.

XML files

You will find XML files (XML Device Description Files) for Beckhoff EtherCAT modules on our website (www.beckhoff.com) under Downloads, in the Configuration Files area.

2.3 EP6001-0002

2.3.1 EP6001-0002 - Introduction

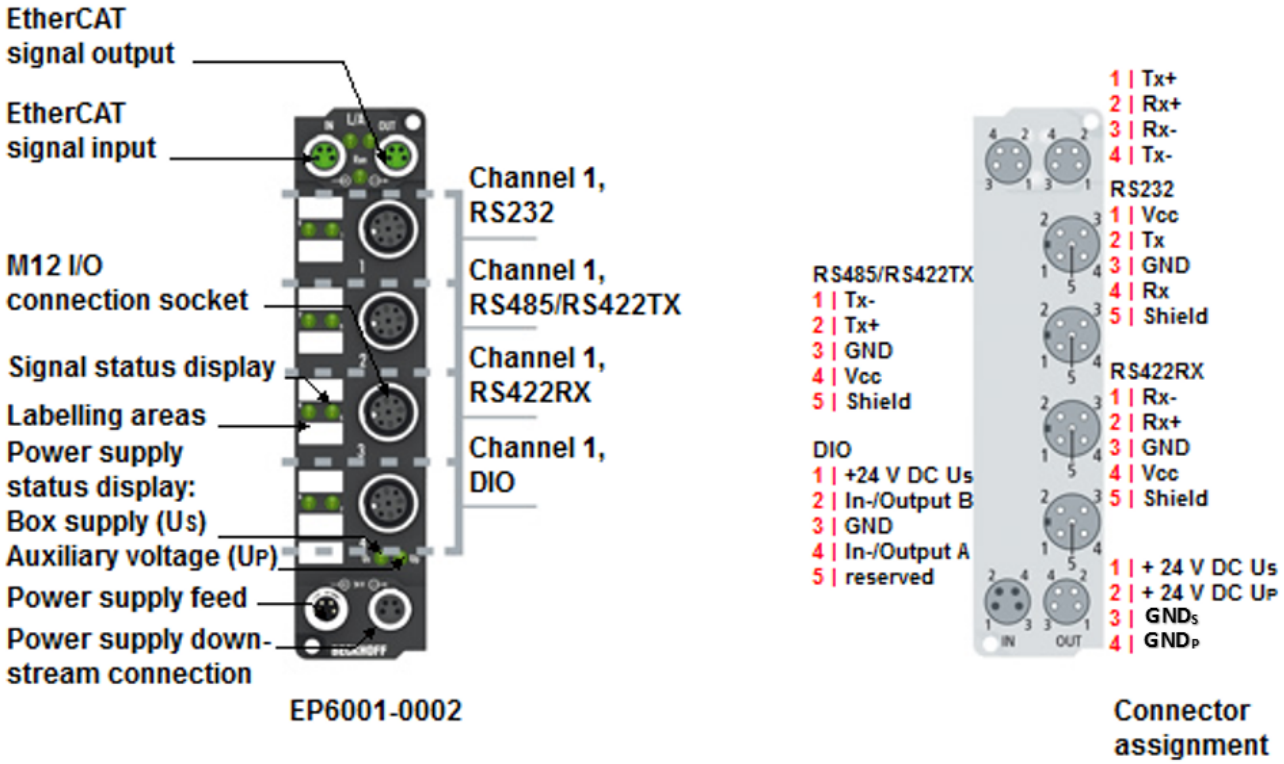


Fig. 4: EP6001-0002

1-channel serial interface, RS232, RS422/RS485

The EP6001-0002 serial interface module allows the connection of devices with an RS232 or an RS422/RS485 interface. The module transmits the data in a fully transparent manner to the higher-level automation device. The active serial communication channel functions independently of the higher-level bus system in full duplex mode at up to 115,200 baud, while a 864 bytes receive buffer and a 128 byte send buffer are available.

The 1-channel version has an increased end device power supply of up to 1 A; the connector assignment depends on the interface selected.

The two integrated digital inputs/outputs allow the connection of additional sensors/actuators in order, for example, to trigger the reading process of the barcode reader or, depending on the result, to initiate an action.

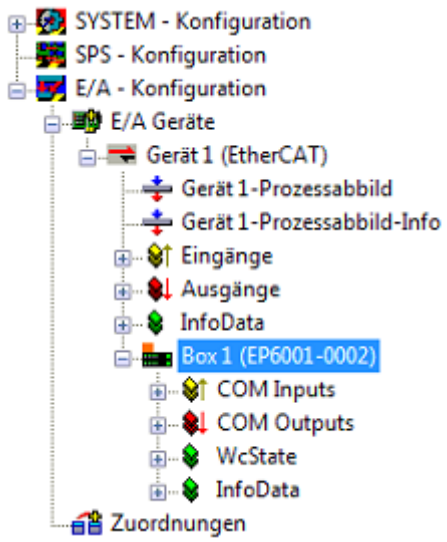
In conjunction with the TwinCAT Virtual Serial COM Driver the EP6001-0002 can be used as a normal Windows COM interface.

2.3.2 EP6001-0002 – Technical data

Technical data	EP6001-0002
Fieldbus [▶ 46]	EtherCAT
Fieldbus connection [▶ 25]	2 x M8 socket (green)
Nominal voltage	24 V_{DC} (-15 %/+20 %)
Data transfer channels	1
Number of digital inputs/outputs	2, 24 V_{DC}, 10 µs/0.5 A
Distributed clocks	-
Data transfer rates	300...115,200 Baud; 9,600 Baud (8-bit, no parity, 1 stop bit) preset
Signal connection [▶ 34]	M12 sockets, screwable for RS232, RS422/485 or digital I/O
Bit distortion	< 3 %
Cable length	RS232: max. 15 m; RS422/RS485: approx. 1,000 m
Data buffer	864-byte receive buffer, 128-byte send buffer
Sensor supply	+ 5 V_{DC}, 1 A
Process image per channel	22 x 8-bit input, 22 x 8-bit output, 16-bit control, 16-bit status
Supply of the module electronics	from the control voltage Us
Current consumption of the module electronics	typical 130 mA + load, e.g. 130 mA + 2 x 20 mA = 170 mA
Power supply connection	Power supply: 1 x M8 connector, 4-pin; downstream connection: 1 x M8 socket, 4-pin
Electrical isolation	500 V
Special features	integrated supply for the end devices 5 V_{DC}/1 A
Permissible ambient temperature during operation	-25…+60 °C 0 °C ... +55 °C (according to cULus, see UL requirements)
Permissible ambient temperature during storage	-40…+85 °C
Vibration / shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2/EN 61000-6-4
Protection class	IP 65/66/67 (conforms to EN 60529)
Weight	app. 165 g
Installation position	variable
Approvals [▶ 42]	CE, UL

2.3.3 EP6001-0002 - Process image

The TwinCAT System Manager displays the EP6001-0002 data in a tree structure.

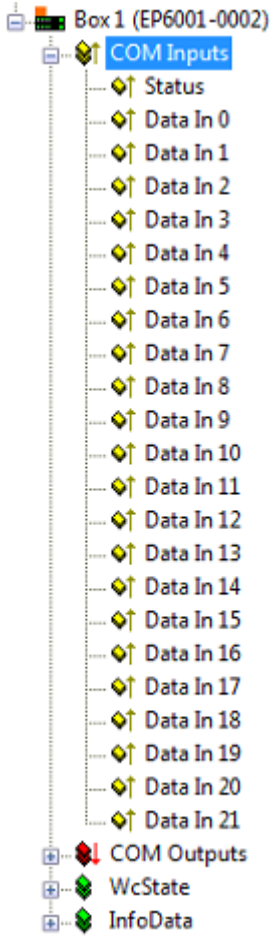


The tree shows:

COM inputs: Channel input data

COM outputs: Channel output data

COM Inputs

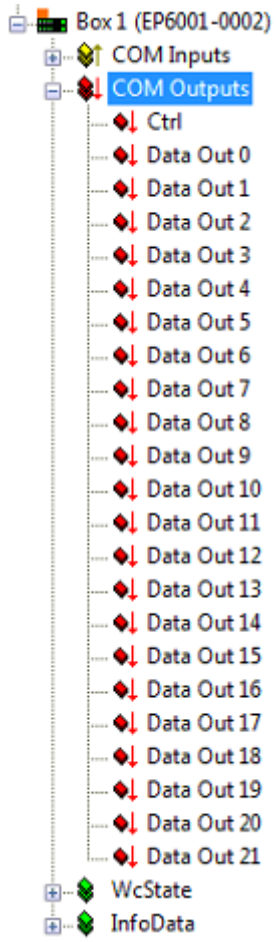


The tree shows:

Status: Status bits of the channel

Data In 0 to Data In 21: Channel input data

COM outputs



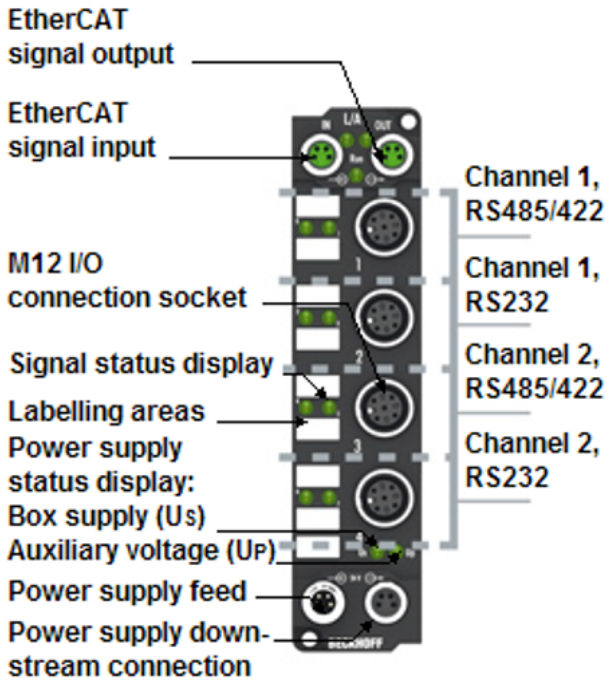
The tree shows:

Status: Control bits of the channel

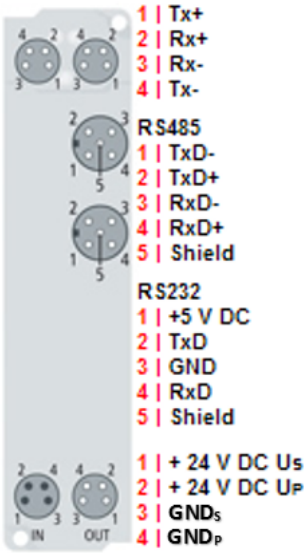
Data Out 0 to Data Out 21: Channel output data

2.4 EP6002-0002

2.4.1 EP6002-0002 - Introduction



EP6002-0002



Connector assignment

Dual-channel serial interface module: RS232 or RS422/RS485

The EP6002-0002 2-channel serial interface module enables the connection of devices with an RS232 or RS422/RS485 interface. The module transmits the data in a fully transparent manner to the higher-level automation device. The data is transferred via the fieldbus using a simple handshake protocol. This does not have any effect on the protocol of the serial interface. The active serial communication channel functions independently of the higher-level bus system in full duplex mode at up to 115,200 baud, while a 864 bytes receive buffer and a 128 byte send buffer are available.

In connection with TwinCAT's virtual serial COM driver (see TwinCAT supplements – communication), the EP6002 can be used as a normal Windows COM interface.

The choice of connection depends on the interface type. In the TwinCAT System Manager [▶ 74] you can select either the RS232 connection **or** the RS422/RS485 connection for each channel.

For

- RS422/RS485 use the M12 connections 1 and 3
- RS232 use the M12 connections 2 and 4

Interface modes / operation modes

the following settings for the interfaces can be made via the CoE objects:

- RS232: point-to-point connection to an RS232 device
- RS422: 4-wire point-to-point connection to an RS422 device
- RS485: 2-wire connection in bus structure to RS485 device(s)
- RS485: 2-wire connection with external bridge in bus structure to RS485 device(s), monitoring of the transmitted data

Quick links

Installation [▶ 21]

Interface modes [▶ 56]

Configuration [▶ 63]

Sample programs [▶ 75]

UL requirements [▶ 42]

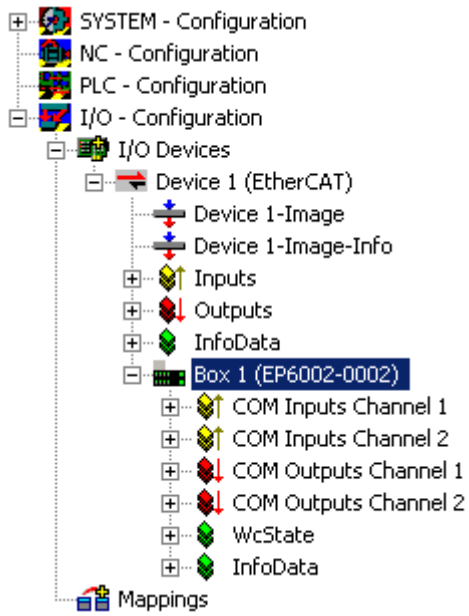
ATEX - Special conditions [▶ 43]

2.4.2 EP6002-0002 – Technical data

Technical data	EP6002-0002
Fieldbus [▶ 46]	EtherCAT
Fieldbus connection [▶ 25]	2 x M8 socket (green)
Data transfer channels	2 (1/1), TxD and RxD, full duplex
Data transfer rates	300...115,200 Baud; 9,600 Baud (8-bit, no parity, 1 stop bit) preset
Signal connection [▶ 34]	M12 sockets for RS232 or RS422/485
Bit distortion	< 3 %
Cable length	RS232: max. 15 m; RS422/RS485: approx. 1,000 m
Data buffer	864-byte receive buffer, 128-byte send buffer
Process image per channel	22 x 8-bit input, 22 x 8-bit output, 16-bit control, 16-bit status
Supply of the module electronics	from the control voltage Us
Current consumption of the module electronics	typical 130 mA + load, e.g. 130 mA + 2 x 20 mA = 170 mA
Power supply connection	Power supply: 1 x M8 plug, 4-pole Onward connection: 1 x M8 socket, 4-pole
Electrical isolation	500 V
Permissible ambient temperature during operation	-25°C ... +60°C 0°C ... +55°C (conforms to ATEX, see special conditions) 0 °C ... +55 °C (according to cULus, see UL requirements)
Permissible ambient temperature during storage	-40°C ... +85°C
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP65, IP66, IP67 (conforms to EN 60529)
Weight	app. 165 g
Installation position	variable
Approvals [▶ 42]	CE, cULus, ATEX

2.4.3 EP6002-0002 - Process image

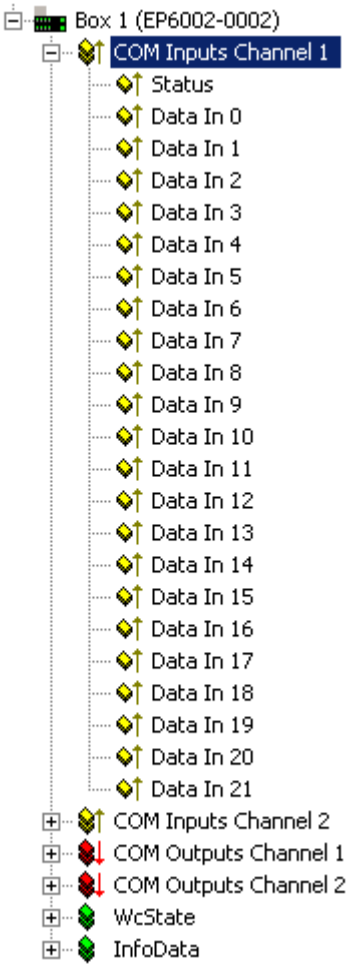
The TwinCAT System Manager displays the EP6002-0002 data in a tree structure.



The tree shows:

- COM Inputs Channel 1: input data of the 1st channel
- COM Inputs Channel 2: input data of the 2nd channel
- COM Outputs Channel 1: output data of the 1st channel
- COM Outputs Channel 2: output data of the 2nd channel

Table 1: COM Inputs Channel 1



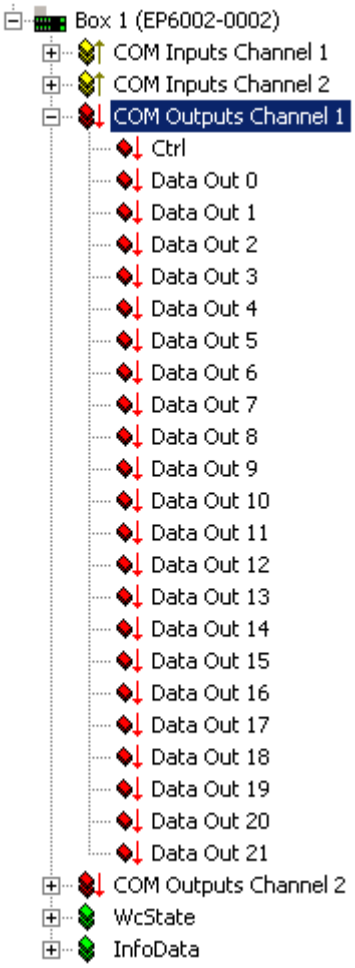
The tree shows:

- Status: status bits of the 1st channel
- Data In 0 to Data In 21: input data of the 1st channel

COM Inputs Channel 2

The data of the 2nd channel are structured identically to those of the 1st channel.

Table 2: COM Outputs Channel 1



The tree shows:

- Status: Control bits of the 1st channel
- Data Out 0 to Data Out 21: output data of the 1st channel

COM Outputs Channel 2

The data of the 2nd channel are structured identically to those of the 1st channel.

3 Mounting and connection

3.1 Mounting

3.1.1 Dimensions

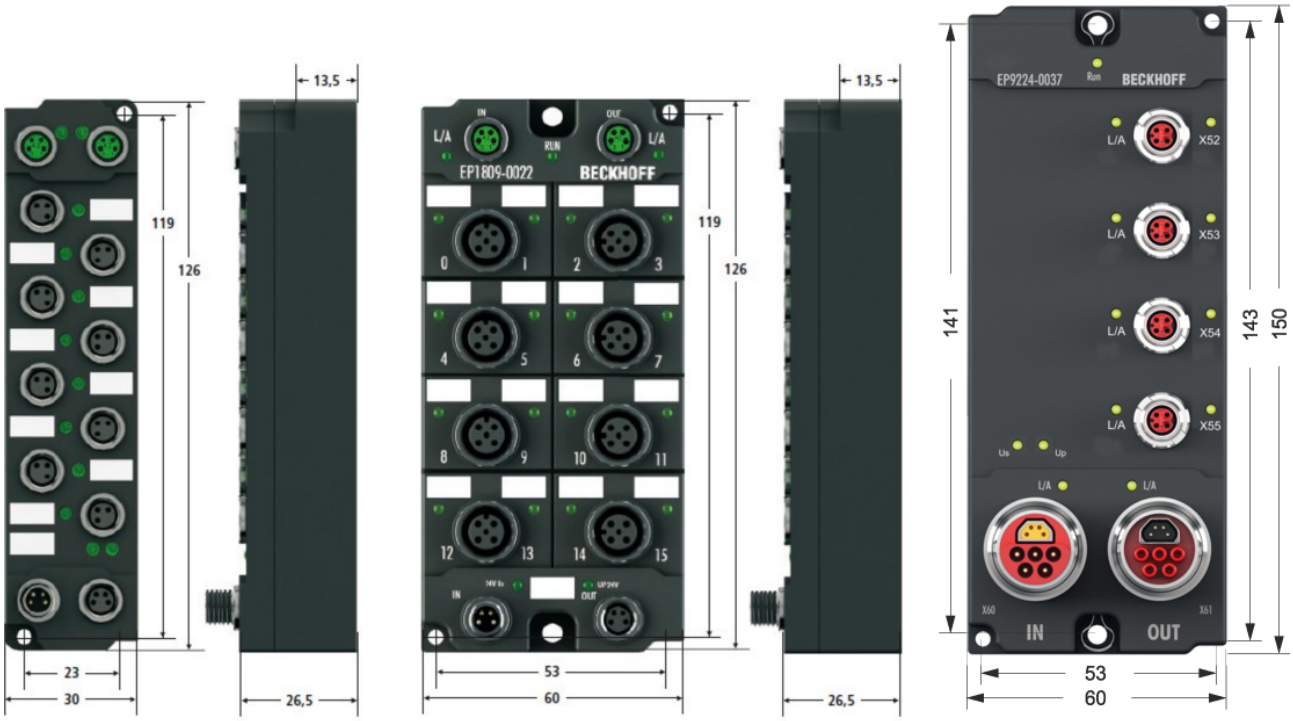


Fig. 5: Dimensions of the EtherCAT Box Modules

All dimensions are given in millimeters.

Housing properties

EtherCAT Box	lean body	wide bodies
Housing material	PA6 (polyamide)	
Casting compound	Polyurethane	
Mounting	two fastening holes Ø 3 mm for M3	two fastening holes Ø 3 mm for M3 two fastening holes Ø 4.5 mm for M4
Metal parts	Brass, nickel-plated	
Contacts	CuZn, gold-plated	
Power feed through	max. 4 A (M8) max. 16 A (7/8“) max. 15.5 A (B17 5G 1.5 mm^2)	
Installation position	variable	
Protection class	IP65, IP66, IP67 (conforms to EN 60529) when screwed together	
Dimensions (H x W x D)	app. 126 x 30 x 26.5 mm	app. 126 x 60 x 26,5 mm app. 150 x 60 x 26.5 mm (without 7/8", B17)

3.1.2 Fixing

Note or pointer

While mounting the modules, protect all connectors, especially the IP-Link, against contamination! Only with connected cables or plugs the protection class IP67 is guaranteed! Unused connectors have to be protected with the right plugs! See for plug sets in the catalogue.

Modules with narrow housing are mounted with two M3 bolts.
Modules with wide housing are mounted with two M3 bolts to the fixing holes located at the corners or mounted with two M4 bolts to the fixing holes located centrally.

The bolts must be longer than 15 mm. The fixing holes of the modules are not threaded.

When assembling, remember that the fieldbus connectors increases the overall height. See chapter accessories.

Mounting Rail ZS5300-0001

The mounting rail ZS5300-0001 (500 mm x 129 mm) allows the time saving assembly of modules.

The rail is made of stainless steel, 1.5 mm thick, with already pre-made M3 threads for the modules. The rail has got 5.3 mm slots to mount it via M5 screws to the machine.

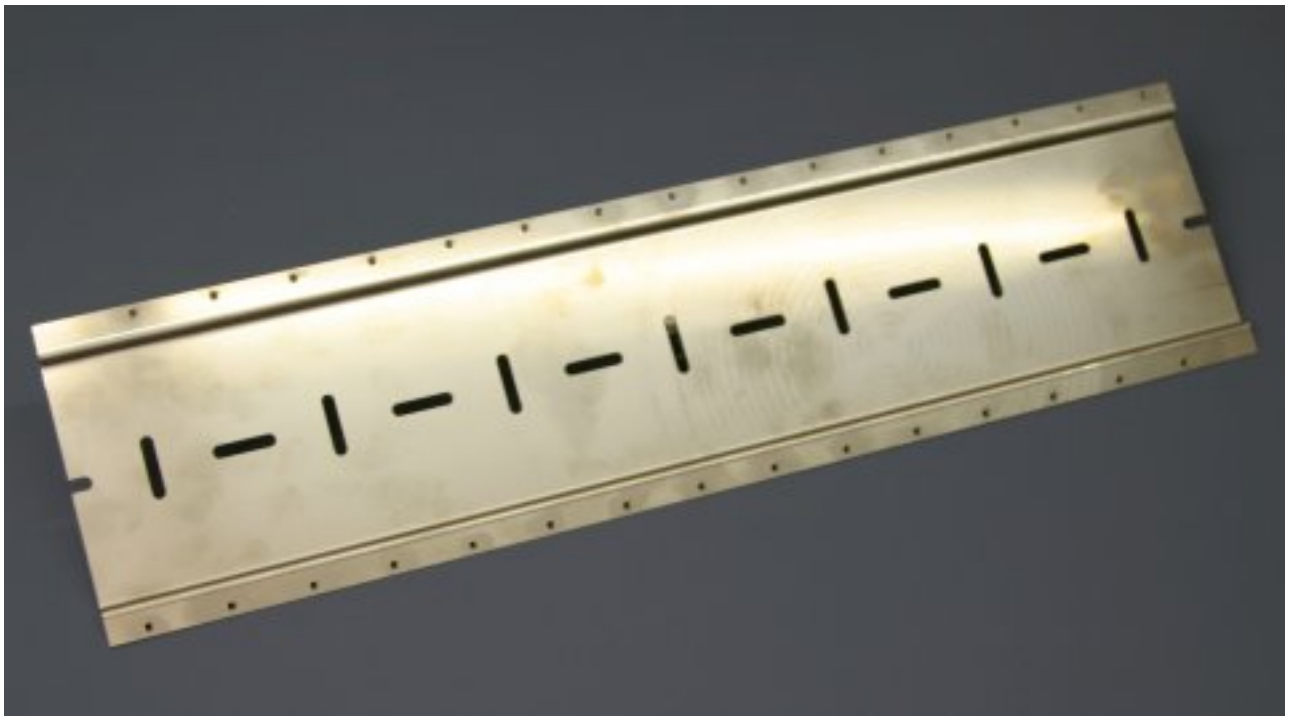


Fig. 6: Mounting Rail ZS5300-000

The mounting rail is 500 mm long, that way 15 narrow modules can be mounted with a distance of 2 mm between two modules. The rail can be cut to length for the application.

Mounting Rail ZS5300-0011

The mounting rail ZS5300-0011 (500 mm x 129 mm) has in addition to the M3 treads also pre-made M4 treads to fix 60 mm wide modules via their middle holes.

Up to 14 narrow or 7 wide modules may be mixed mounted.

3.1.3 Nut torque for connectors

M8 connectors

It is recommended to pull the M8 connectors tight with a nut torque of **0.4 Nm**. When using the torque control screwdriver ZB8800 is also a max. torque of **0.5 Nm** permissible.

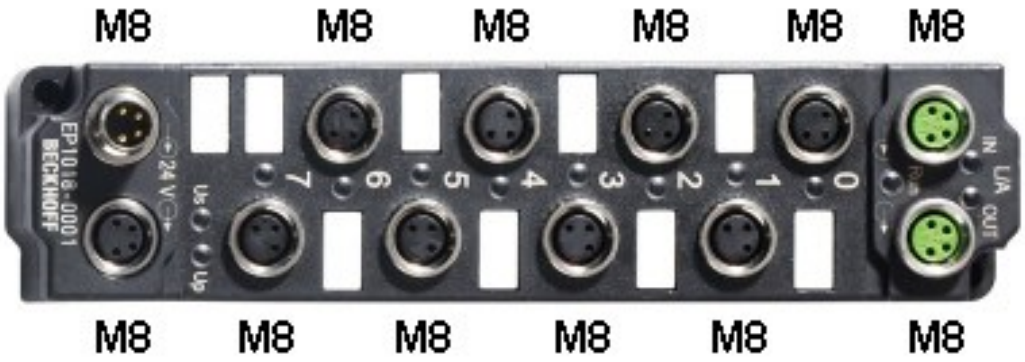


Fig. 7: EtherCAT Box with M8 connectors

M12 connectors

It is recommended to pull the M12 connectors tight with a nut torque of **0.6 Nm**.

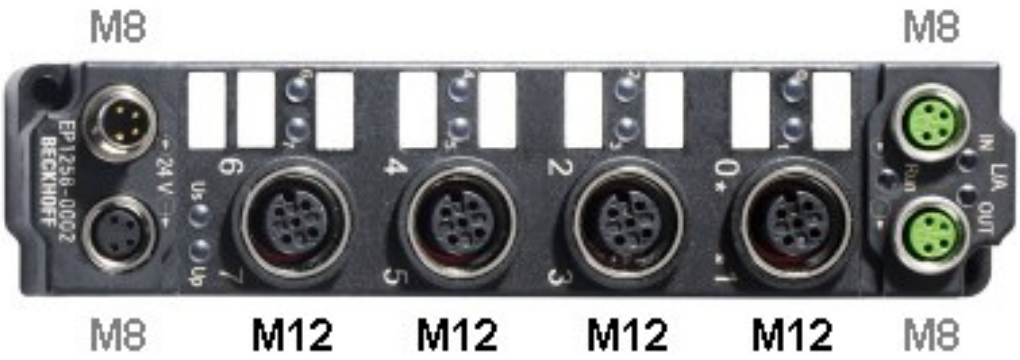


Fig. 8: EtherCAT Box with M8 and M12 connectors

7/8" plug connectors

We recommend fastening the 7/8" plug connectors with a torque of **1.5 Nm**.

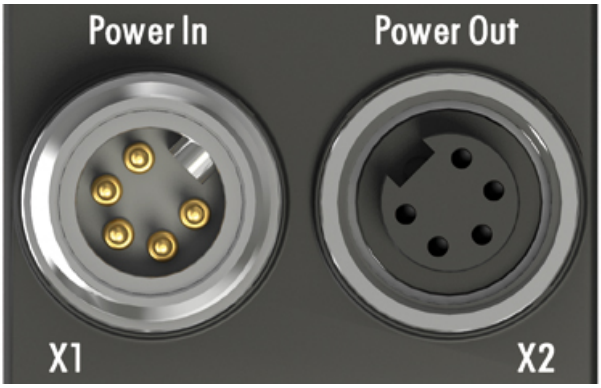


Fig. 9: 7/8" plug connectors

Torque socket wrenches

Fig. 10: ZB8801 torque socket wrench

Ensure the right torque

Use the torque socket wrenches available by Beckhoff to pull the connectors tight (ZB8800, ZB8801-0000)!

3.1.4 Additional checks

The boxes have undergone the following additional tests:

Verification	Explanation
Vibration	10 frequency runs in 3 axes
	5 Hz < f < 60 Hz displacement 0.35 mm, constant amplitude
	60.1 Hz < f < 500 Hz acceleration 5 g, constant amplitude
Shocks	1000 shocks in each direction, in 3 axes
	35 g, 11 ms

3.2 EtherCAT

3.2.1 EtherCAT connection

For the incoming and ongoing EtherCAT connection,

- the EtherCAT Box (EPxxxx) has two M8 sockets, marked in **green**
- the Coupler Box (FBB-x110) has two M12 sockets

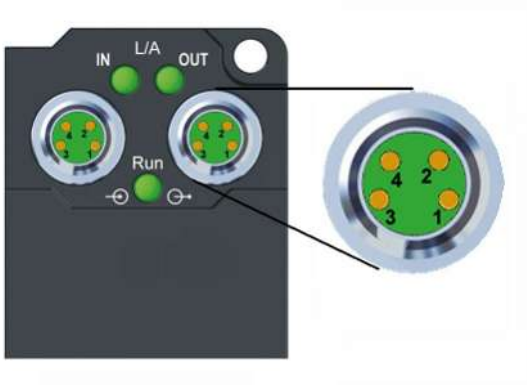


Fig. 11: EtherCAT Box: M8, 30 mm housing

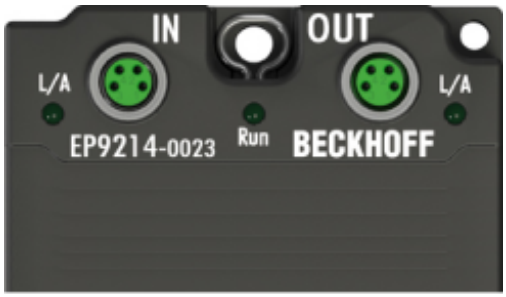


Fig. 12: EtherCAT Box: M860 mm housing (example: EP9214)

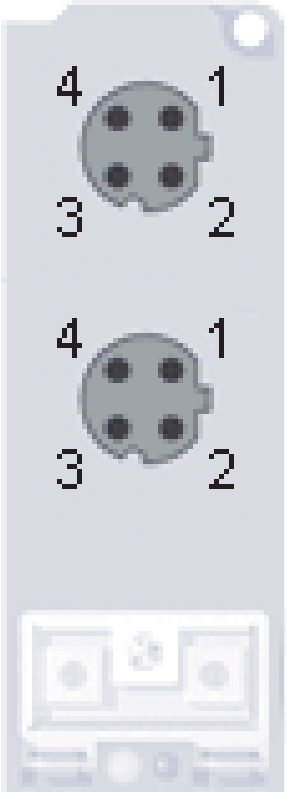


Fig. 13: Coupler Box: M12

Assignment

There are various different standards for the assignment and colors of connectors and cables for Ethernet/ EtherCAT.

Ethernet/EtherCAT		Plug connector			Cable		Standard
Signal	**Description**	**M8**	**M12**	**RJ45[1]**	**ZB9010, ZB9020, ZB9030, ZB9032, ZK1090-6292, ZK1090-3xxx-xxxx**	**ZB9031 and old versions of ZB9030, ZB9032, ZK1090-3xxx-xxxx**	**TIA-568B**
Tx +	Transmit Data+	Pin 1	Pin 1	Pin 1	yellow[2]	orange/white[3]	white/orange
Tx -	Transmit Data-	Pin 4	Pin 3	Pin 2	orange[2]	orange[3]	orange
Rx +	Receive Data+	Pin 2	Pin 2	Pin 3	white[2]	blue/white[3]	white/green
Rx -	Receive Data-	Pin 3	Pin 4	Pin 6	blue[2]	blue[3]	green
Shield	Shield	Housing		Shroud	Screen	Screen	Screen

[1]) colored markings according to EN 61918 in the four-pin RJ45 connector ZS1090-0003
[2]) wire colors according to EN 61918
[3]) wire colors

Assimilation of color coding for cable ZB9030, ZB9032 and ZK1090-3xxxx-xxxx (with M8 connectors)

For unification the prevalent cables ZB9030, ZB9032 and ZK1090-3xxx-xxxx this means the pre assembled cables with M8 connectors were changed to the colors of EN61918 (yellow, orange, white, blue).So different color coding exists. But the electrical properties are absolutely identical.

EtherCAT connector

The following connectors can be supplied for use in Beckhoff EtherCAT systems.

Name	Connector	Comment
ZS1090-0003	RJ45	four-pole, IP20, field-configurable
ZS1090-0004	M12, male	four-pin, IP67, for field assembly
ZS1090-0005	RJ45	eight-pole, IP20, field-configurable, suitable for gigabit Ethernet
ZS1090-0006	M8 plug connector	four-pole, IP67, field-configurable, for cable type ZB903x
ZS1090-0007	M8 socket	four-pole, IP67, field-configurable, for cable type ZB903x
ZS1090-1006	M8 plug connector	four-pole, IP67, field-configurable up to OD = 6.5 mm
ZS1090-1007	M8 socket	four-pole, IP67, field-configurable up to OD = 6.5 mm

3.2.2 EtherCAT - Fieldbus LEDs

Fig. 14: EtherCAT-LEDs

LED display

LED	Display	Meaning
IN L/A	off	no connection to the preceding EtherCAT module
	Lit	LINK: connection to the preceding EtherCAT module
	flashing	ACT: Communication with the preceding EtherCAT module
OUT L/A	off	no connection to the following EtherCAT module
	Lit	LINK: connection to the following EtherCAT module
	flashing	ACT: Communication with the following EtherCAT module
Run	off	Status of the EtherCAT module is Init
	flashes quickly	Status of the EtherCAT module is pre-operational
	flashes slowly	Status of the EtherCAT module is safe-operational
	Lit	Status of the EtherCAT module is operational

EtherCAT statuses

The various statuses in which an EtherCAT module may be found are described in the Basic System Documentation for EtherCAT, which is available for download from our website (www.beckhoff.com) under Downloads.

3.3 Power supply

3.3.1 Power Connection

The feeding and forwarding of supply voltages is done via two M8 connectors at the bottom end of the modules:

- IN: left M8 connector for feeding the supply voltages
- OUT: right M8 connector for forwarding the supply voltages

Fig. 15: EtherCAT Box, Connectors for power supply

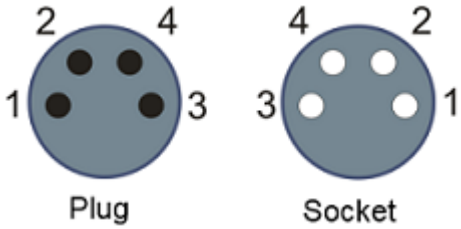


Fig. 16: Pin assignment M8, Power In and Power Out

Table 3: PIN assignment

Pin	Voltage	
1	Control voltage Us, +24 V_{DC}	
2	Auxiliary voltage Up, +24 V_{DC}	
3	GNDs*	*) may be connected internally to each other depending on the module: see specific module descriptions
4	GNDp*	

The pins M8 connectors carry a maximum current of 4 A.

Two LEDs display the status of the supply voltages.

NOTE

Don't confuse the power connectors with the EtherCAT connectors!

Never connect the power cables (M8, 24 V_{DC}) with the green marked EtherCAT sockets of the EtherCAT Box Modules! This can damage the modules!

Control voltage Us: 24 V_{DC}

Power is supplied to the fieldbus, the processor logic, the inputs and the sensors from the 24 V_{DC} control voltage Us. The control voltage is electrically isolated from the fieldbus circuitry.

Auxiliary voltage Up 24 V_{DC}

The Auxiliary voltage Up supplies the digital outputs; it can be brought in separately. If the load voltage is switched off, the fieldbus functions and the power supply and functionality of the inputs are retained.

Redirection of the supply voltages

The IN and OUT power connections are bridged in the module (not IP204x-Bxxx and IE204x). The supply voltages Us and Up can thus easily be transferred from EtherCAT Box to EtherCAT Box.

NOTE

Pay attention to the maximum permissible current!

Pay attention also for the redirection of the supply voltages Us and Up, the maximum permissible current for M8 connectors of 4 A must not be exceeded!

Supply via EP92x4-0023 PowerBox modules

If the machine requires higher current or if the EtherCAT Box Modules are installed far away from the control cabinet with included power supply, the usage of four cannel power distribution modules EP9214 or EP9224 (with integrated data logging, see www.beckhoff.com/EP9224) is recommended.

With these modules intelligent power distribution concepts with up to 2 x 16 A and a maximum of 2.5 mm² cable cross-section can be realized.

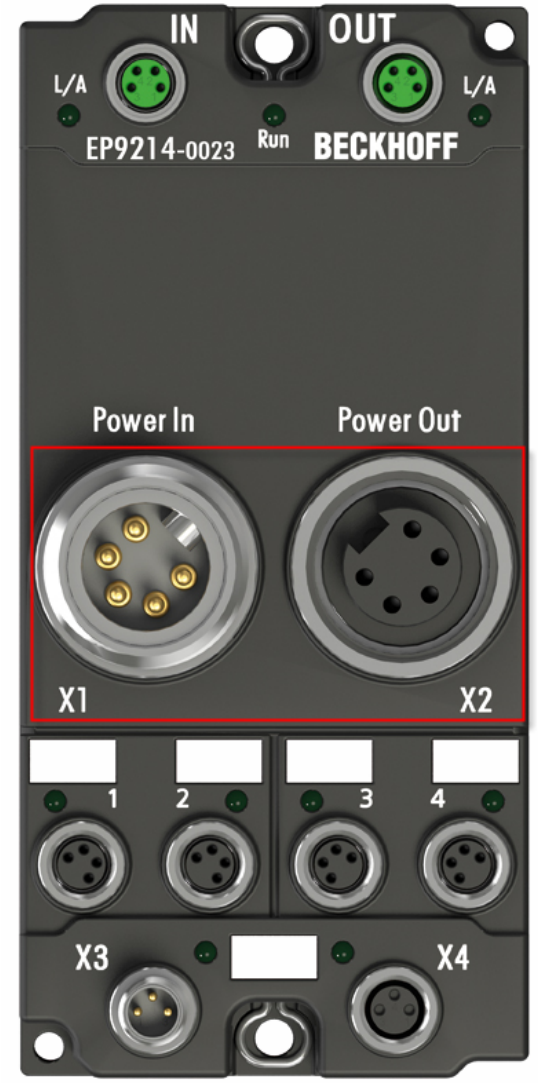


Fig. 17: EP92x4-0023, Connectors for Power In and Power Out

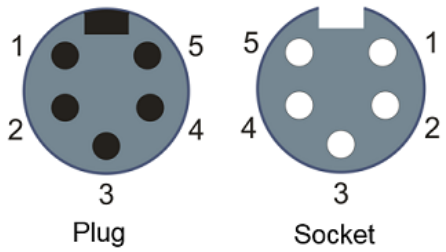


Fig. 18: Pin assignment 7/8”, Power In and Power Out

Electrical isolation

Digital modules

In the digital input/output modules, the grounds of the control voltage (GNDs) and the auxiliary voltage (GNDp) are connected to each other!

Check this at the documentation of each used EtherCAT Box.

Analog modules

In the analog input/output modules the grounds of the control voltage (GNDs) and the auxiliary voltage (GNDp) are separated from each other in order to ensure electrical isolation of the analog signals from the control voltage.

In some of the analog modules the sensors or actuators are supplied by Up - this means, for instance, that in the case of 0...10 V inputs, any reference voltage (0...30 V) may be connected to Up; this is then available to the sensors (e.g. smoothed 10 V for measuring potentiometers).

Details of the power supply may be taken from the specific module descriptions.

NOTE

Electrical isolation may be cancelled!

If digital and analog fieldbus boxes are connected directly via four-core power leads, the analog signals in the fieldbus boxes may be no longer electrically isolated from the control voltage!

3.3.2 Status LEDs for power supply

Fig. 19: Status LEDs for power supply

LED display

LED	Display	Meaning
Us (Control voltage)	off	The power supply voltage Us is not present
	green illuminated	The power supply voltage Us is present
	red illuminated	Because of overload (current > 0.5 A) the sensor supply generated from power supply voltage Us was switched off for all sensors fed from this.
Up (Auxiliary voltage)	off	The power supply voltage Up is not present
	green illuminated	The power supply voltage Up is present

3.3.3 Power cable conductor losses M8

The ZK2020-xxxx-yyyy power cables should not exceed the total length of 15 m at 4 A (with continuation). When planning the cabling, note that at 24 V nominal voltage, the functionality of the module can no longer be assured if the voltage drop reaches 6 V. Variations in the output voltage from the power supply unit must also be taken into account.

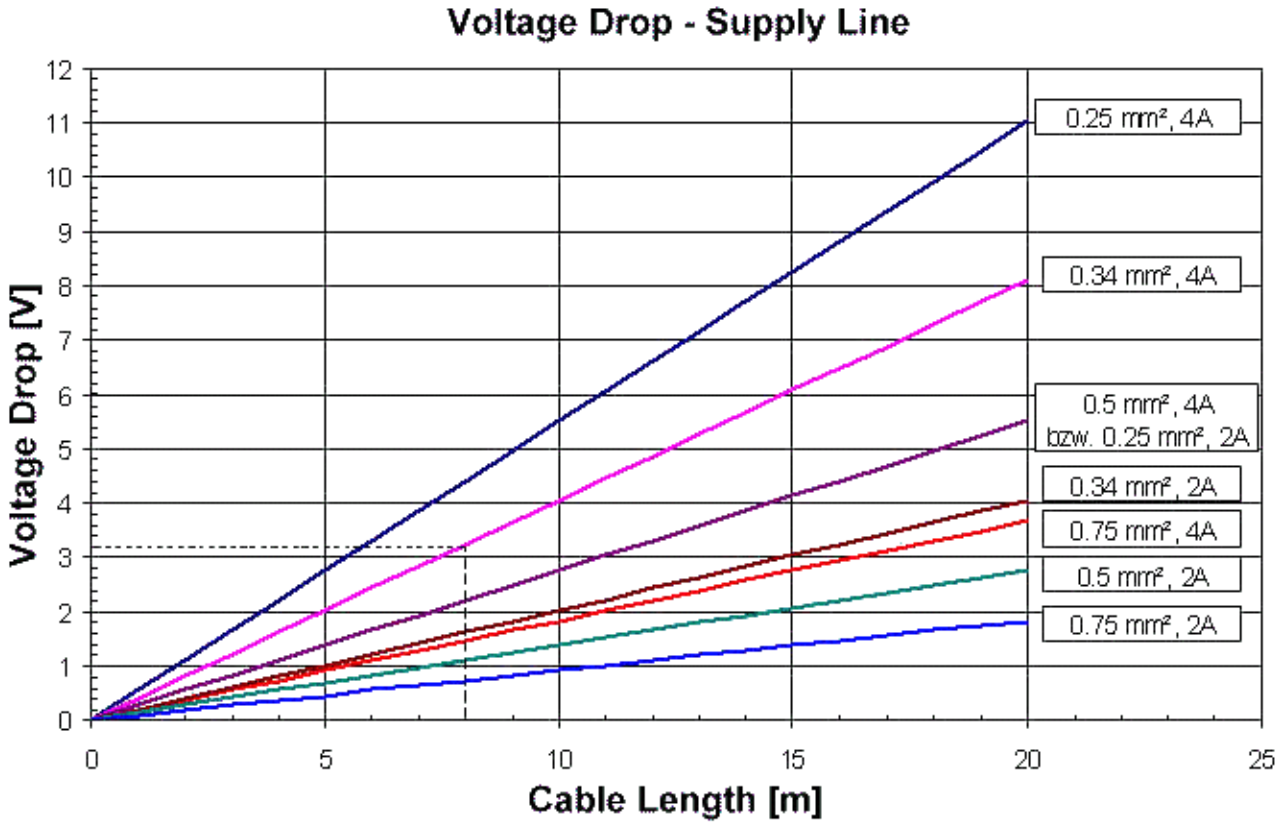


Fig. 20: Power cable conductor losses

Example

8 m power cable with 0.34 mm² cross-section has a voltage drop of 3.2 V at 4 A.

EP92x4 Power Distribution Modules

With EP9214 and EP9224 Power Distribution Modules intelligent concepts for voltage supply are available. Further information may be found under www.beckhoff.com/EP9224.

3.3.4 Conductor losses 7/8"

In the case of the power cables ZK2030-xxxx-yyy, a total length of 15 m should not be exceeded at 16 A. When wiring, note that with a rated voltage of 24 V the function of the modules can no longer be guaranteed from a voltage drop of 6 V. Variations in the output voltage from the power supply unit must also be taken into account.

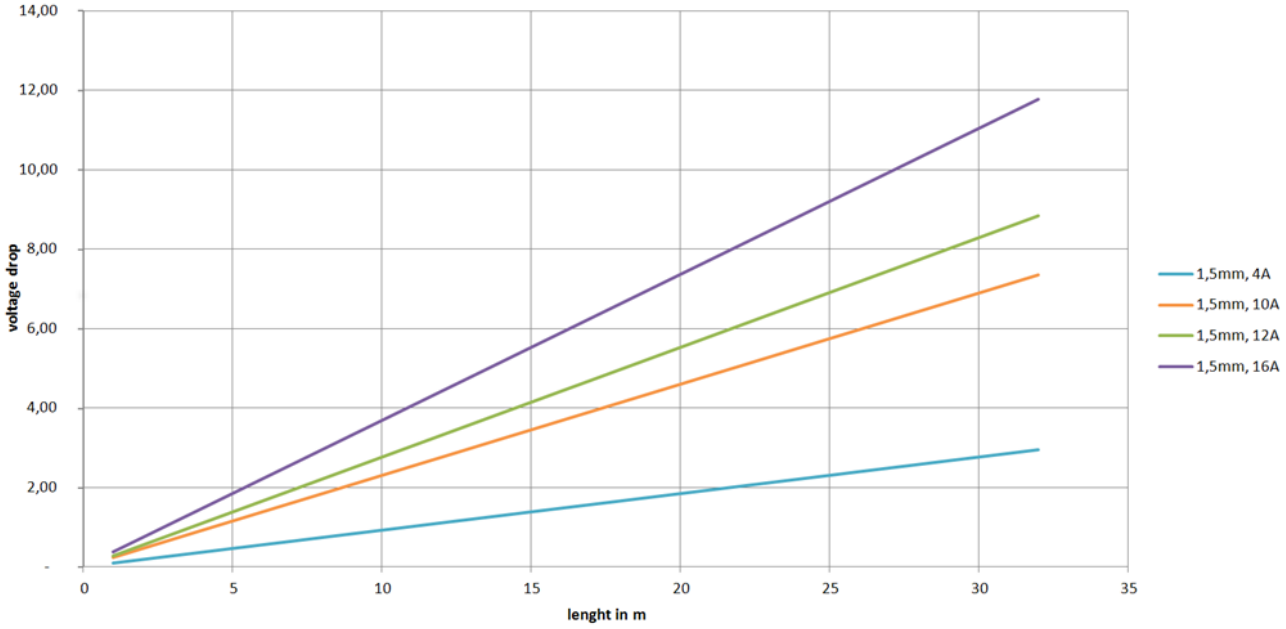


Fig. 21: ZK2030-xxxx-yyy - Conductor losses

Alternatively, larger cable cross-section can be used, e.g. 2.5 mm^2.

3.4 Signal connection

3.4.1 Signal connection EP6001-0002

RS232

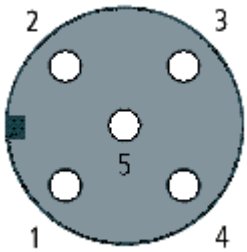


Fig. 22: M12 socket, A-coded

Pin	Signal	Connection for
1	5 V_{DC}	auxiliary voltage 5 V_{DC} (20 mA, short-circuit proof)
2	TxD	send data
3	GND	ground
4	RxD	receive data
5	SHLD	shield

RS485/RS422TX

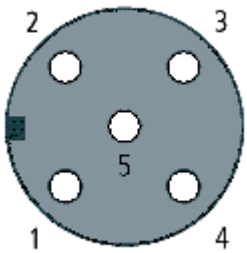


Fig. 23: M12 socket, A-coded

Pin	Signal	Connection for
1	Tx-	send data
2	Tx+	send data
3	GND	ground
4	Vcc	auxiliary voltage 5 V_{DC} (20 mA, short-circuit proof)
5	Shield	shield

RS485 - half duplex connection

For half duplex connection under RS485 only one pair of wires is needed for data transmission. Connect this core pair to Tx-/Tx+.

RS485 bus structure - use termination resistors

A linear bus with more than two devices can be set-up in RS485 mode. To prevent reflections during the data transmission, it is necessary to terminate the line ends of the bus cable with resistors (120 Ω).

RS422RX

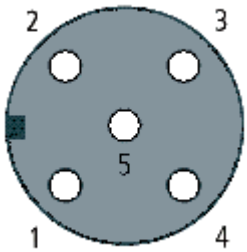


Fig. 24: M12 socket, A-coded

Pin	Signal	Connection for
1	Rx-	receive data
2	Rx+	receive data
3	GND	ground
4	Vcc	auxiliary voltage 5 V_{DC} (20 mA, short-circuit proof)
5	Shield	shield

Digital inputs M8 and M12

The digital input modules acquire the binary control signals from the process level and transmit them to the higher-level automation device.

The signals are connected via M8 connectors (EPxxxx-0001) or M12 connectors (EPxxxx-0002).

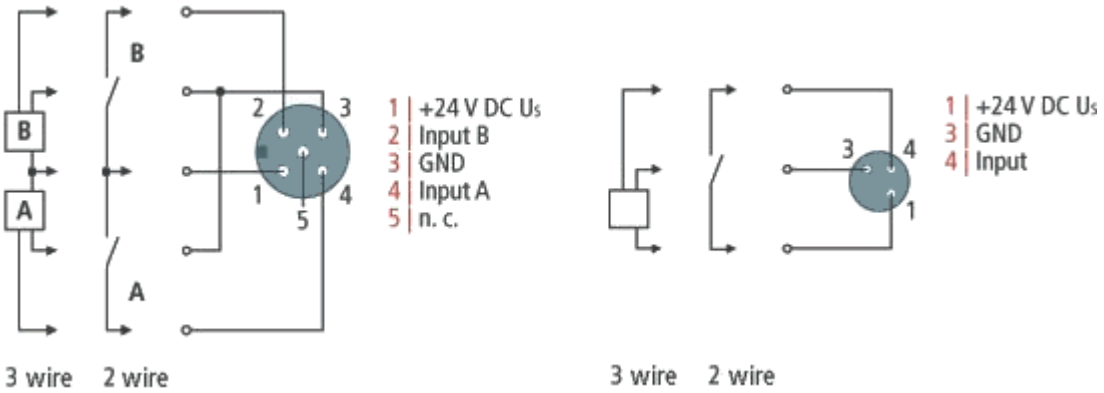


Fig. 25: Digital inputs M8 and M12

The sensors are supplied with a common maximum current of 0.5 A from the control voltage Us.

Light emitting diodes indicate the signal state of the inputs.

Digital outputs M8 and M12

The digital output modules connect the binary control signals from the automation unit on to the actuators at the process level.

The signals are connected via M8 connectors (EP2xxx-0001) or M12 connectors (EP2xxx-0002).

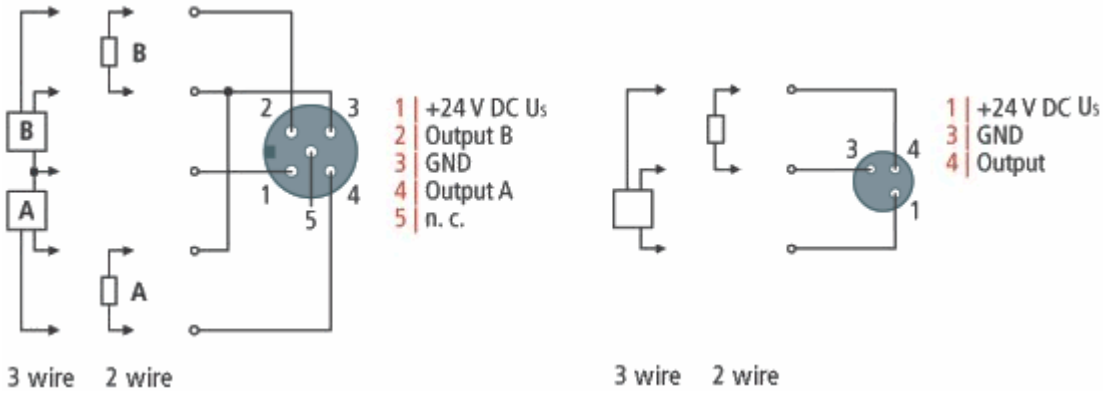


Fig. 26: Digital outputs M8 and M12

The outputs are short-circuit proof and protected against inverse connection.

LEDs indicate the signal state of the outputs.

3.4.2 Signal connection EP6002-0002

RS485/422

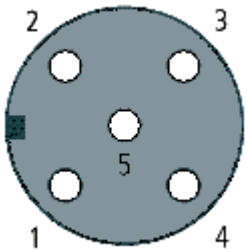


Fig. 27: M12 socket, A-coded

Pin	Signal	Connection for
1	TxD-	send data
2	TxD+	send data
3	RxD-	receive data
4	RxD+	receive data
5	SHLD	shield

RS485 - Half Duplex Connection

For half duplex connection under RS485 only one pair of wires is needed for data transmission. Connect this pair of wires to Tx-/Tx+.

RS485 bus set up - use terminating resistors

A linear bus with more than two devices can be set-up in RS485 mode. To prevent reflections during the data transmission, it is necessary to terminate the line ends of the bus cable with resistors (120 Ω).

RS232

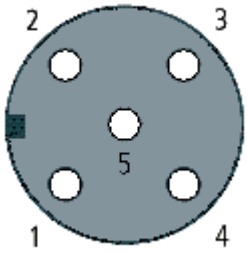


Fig. 28: M12 socket, A-coded

Pin	Signal	Connection for
1	5 V_{DC}	auxiliary voltage 5 V_{DC} (20 mA, short-circuit proof)
2	TxD	send data
3	GND	ground
4	RxD	receive data
5	SHLD	shield

3.5 EP6001-0002 - Status LEDs

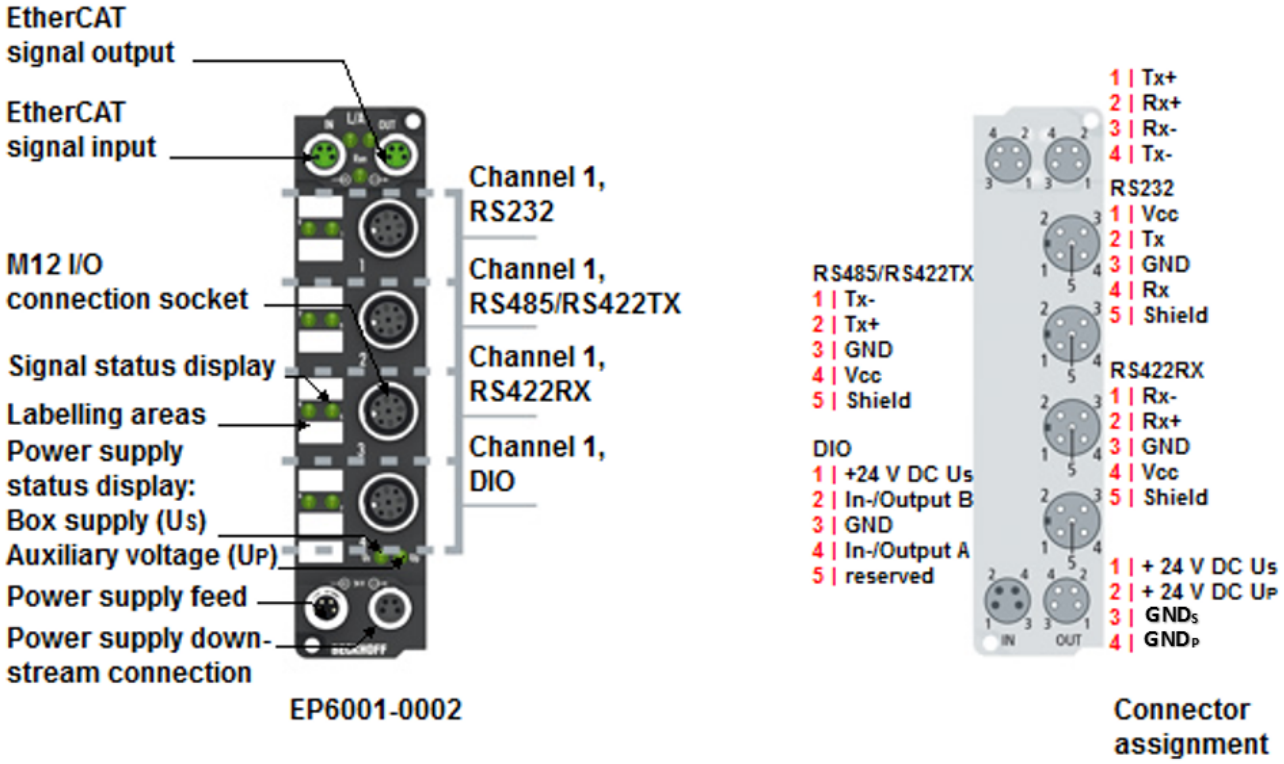


Fig. 29: EP6001-0002 - Status LEDs

Status LEDs at the M12 connectors

Connector	LED	Display	Meaning
M12 socket no. 1-3	RX left	off	Serial port on this M12 connector not ready to receive data
		green illuminated	Serial port on this M12 connector ready to receive data
		orange illuminated	Serial port on this M12 connector receiving data
	TX right	off	Serial port on this M12 connector not ready to transmit data
		green illuminated	Serial port on this M12 connector ready to transmit data
		orange illuminated	Serial port on this M12 connector transmitting data
M12 socket no. 4	left LED (6)	off	Output A / Input A inactive
		green illuminated	Output A / Input A active
	right LED (7)	off	Output B / Input B inactive
		green illuminated	Output B / Input B active

Power supply

LED	Display	Meaning
Us	off	The supply voltage, Us, is not present
	green illuminated	The supply voltage, Us, is present
Up	off	The supply voltage, Up, is not present
	green illuminated	The supply voltage, Up, is present

3.6 EP6002-0002 - Status LEDs

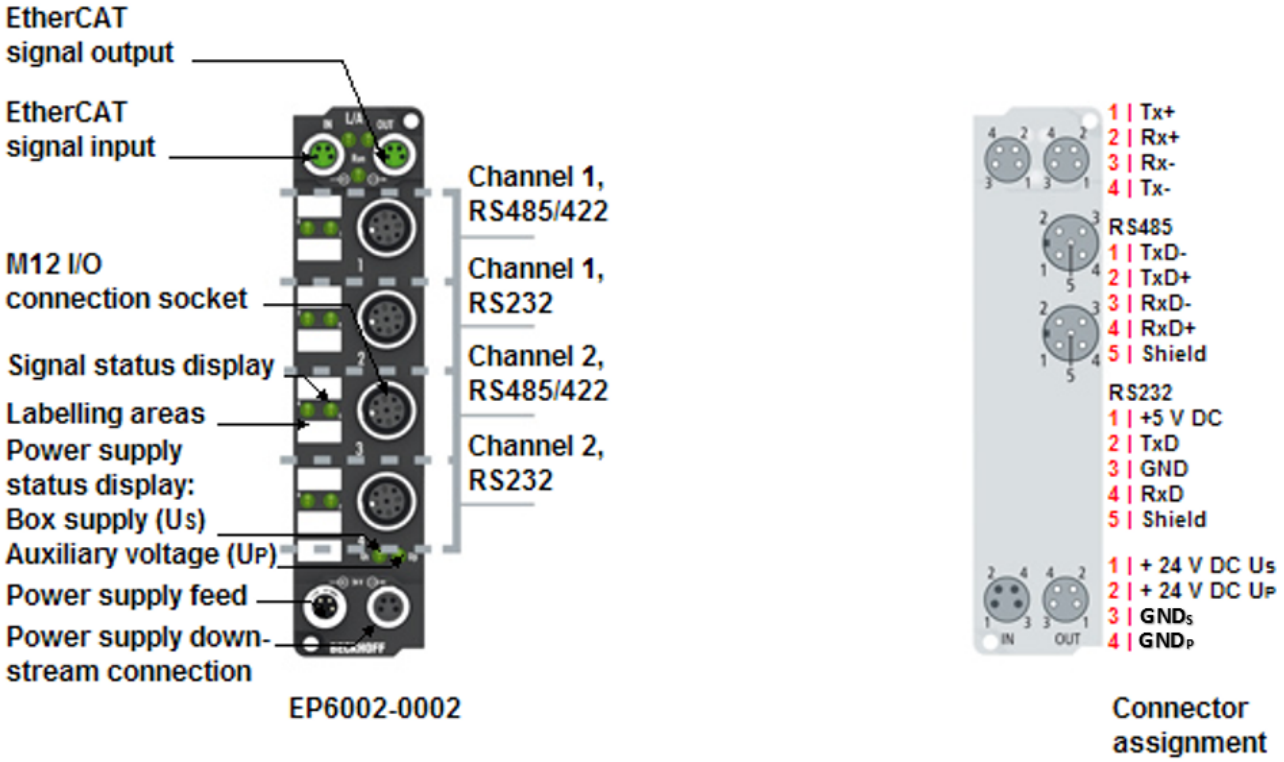


Fig. 30: EP6002-0002 - Status LEDs

Status LEDs at the M12 connectors

Connector	LED	Display	Meaning
M12 socket no. 1-4	RX left	off	Serial port on this M12 connector not ready to receive data
		green	Serial port on this M12 connector ready to receive data
		orange	Serial port on this M12 connector receiving data
	TX right	off	Serial port on this M12 connector not ready to transmit data
		green	Serial port on this M12 connector ready to transmit data
		orange	Serial port on this M12 connector transmitting data

Power supply

LED	Display	Meaning
Us	off	The supply voltage, Us, is not present
	green illuminated	The supply voltage, Us, is present
Up	off	The supply voltage, Up, is not present
	green illuminated	The supply voltage, Up, is present

3.7 Cabling

A list of EtherCAT cables, power cables, sensor cables, Ethernet/EtherCAT connectors and field-configurable connectors can be found under the following link: https://beckhoff.de/english/fieldbus_box/ethercat_box_accessories_overview.htm?id=25525466903389

The corresponding data sheets can be found under the following link:

https://beckhoff.de/english/ethercat-box/ethercat_box_cables.htm?id=690338951657421

EtherCAT cables

Fig. 31: ZK1090-3131-0xxx

For connecting EtherCAT devices, only use shielded Ethernet cables with a minimum specification of **category 5 (CAT5) according to EN 50173 or ISO/IEC 11801**.

Wiring recommendations

Detailed recommendations for EtherCAT cabling can be found in the documentation "Design recommendations for EtherCAT/Ethernet infrastructure", which is available for download from www.beckhoff.de.

EtherCAT uses four cable wires for signal transmission.
Due to automatic cable detection (auto-crossing) symmetric (1:1) or cross-over cables can be used between EtherCAT devices from Beckhoff.

Power cable

Fig. 32: ZK2020-3132-0xxx

Sensor cables

Fig. 33: Selection of Beckhoff sensor cables

3.8 UL Requirements

The installation of the EtherCAT Box Modules certified by UL has to meet the following requirements.

Supply voltage

⚠ CAUTION

CAUTION!

This UL requirements are valid for all supply voltages of all marked EtherCAT Box Modules!
For the compliance of the UL requirements the EtherCAT Box Modules should only be supplied

- by a 24 V_{DC} supply voltage, supplied by an isolating source and protected by means of a fuse (in accordance with UL248), rated maximum 4 Amp, or
- by a 24 V_{DC} power source, that has to satisfy *NEC class 2*.
  A *NEC class 2* power supply shall not be connected in series or parallel with another (class 2) power source!

⚠ CAUTION

CAUTION!

To meet the UL requirements, the EtherCAT Box Modules must not be connected to unlimited power sources!

Networks

⚠ CAUTION

CAUTION!

To meet the UL requirements, EtherCAT Box Modules must not be connected to telecommunication networks!

Ambient temperature range

⚠ CAUTION

CAUTION!

To meet the UL requirements, EtherCAT Box Modules has to be operated only at an ambient temperature range of 0 to 55°C!

Marking for UL

All EtherCAT Box Modules certified by UL (Underwriters Laboratories) are marked with the following label.

Fig. 34: UL label

3.9 ATEX notes

3.9.1 ATEX - Special conditions

⚠ WARNING

Observe the special conditions for the intended use of EtherCAT Box modules in potentially explosive areas – directive 94/9/EU.

- The certified components are to be installed in the BG2000-0000 protection enclosure [▶ 44] that guarantees a protection against mechanical hazards!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observethe permissible ambient temperature range of 0 - 55°C for the use of EtherCAT Box modules in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

Standards

The fundamental health and safety requirements are fulfilled by compliance with the following standards:

- EN 60079-0: 2006
- EN 60079-15: 2005

Marking

The EtherCAT Box modules certified for potentially explosive areas bear the following marking:

II 3 G Ex nA II T4 DEKRA 11ATEX0080 X Ta: 0 - 55°C

or

II 3 G Ex nA nC IIC T4 DEKRA 11ATEX0080 X Ta: 0 - 55°C

Batch number (D number)

The EtherCAT Box modules bear a batch number (D number) that is structured as follows:

D: WW YY FF HH

WW - week of production (calendar week)
YY - year of production
FF - firmware version
HH - hardware version

Beispiel mit Ser. Nr.: 29 10 02 01:

29 - week of production 29
10 - year of production 2010
02 - firmware version 02
01 - hardware version 01

3.9.2 BG2000-0000 - EtherCAT Box protection enclosure

⚠ WARNING

Risk of electric shock and damage of device!

Bring the EtherCAT system into a safe, powered down state before starting installation, disassembly or wiring of the modules!

ATEX

The BG2000-0000 protection enclosure has to be mounted over a single EtherCAT Box to fulfill the special conditions according to ATEX [▶ 43].

Installation

Put the cables for EtherCAT, power supply and sensors/actuators through the hole of the BG2000-0000 protection enclosure.

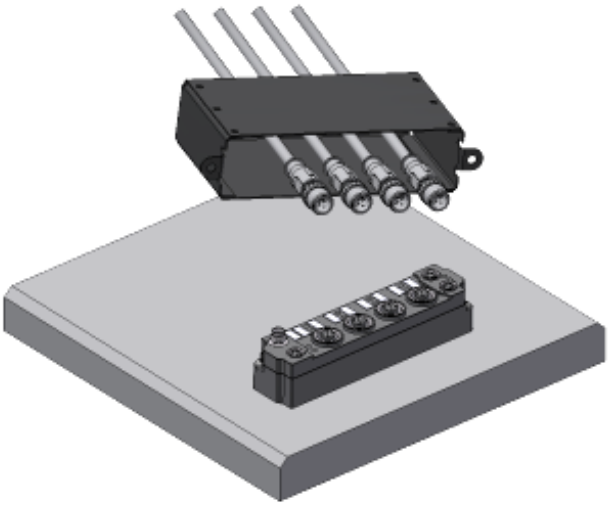


Fig. 35: BG2000-0000, putting the cables

Fix the wires for EtherCAT, power supply and sensors/actuators to the EtherCAT Box.

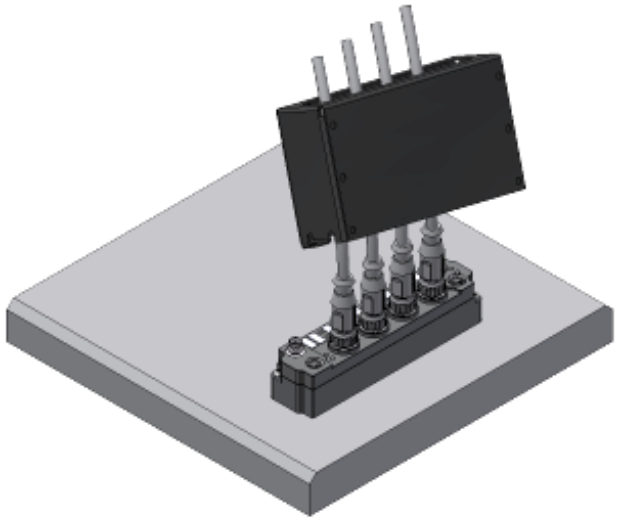


Fig. 36: BG2000-0000, fixing the cables

Mount the BG2000-0000 protection enclosure over the EtherCAT Box.

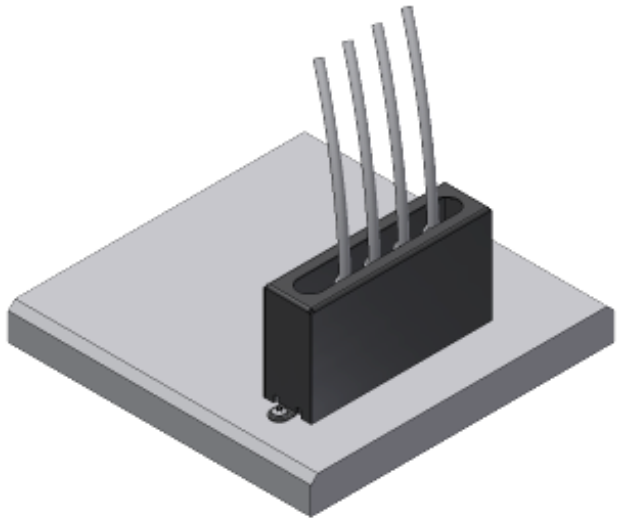


Fig. 37: BG2000-0000, mounting the protection enclosure

3.9.3 ATEX Documentation

Notes about operation of EtherCAT Box Modules (EPxxxx-xxxx) in potentially explosive areas (ATEX)

Pay also attention to the continuative documentationNotes about operation of EtherCAT Box Modules (EPxxxx-xxxx) in potentially explosive areas (ATEX) that is available in the download area of the Beckhoff homepage http:\\www.beckhoff.com!

4 Basics communication - EtherCAT

4.1 EtherCAT basics

Basic information on the EtherCAT fieldbus can be found in the EtherCAT system documentation.

4.2 Watchdog setting

General information on watchdog settings

The ELxxxx Terminals and EPxxxx Box Modules are equipped with a safety device (watchdog) that switches the outputs to a safe state after a time that can be preset, for example in the case of interrupted process data traffic, or to OFF, for example depending on device and setting.

The EtherCAT Slave Controller (ESC) has two watchdogs:

- SM watchdog (default: 100 ms)
- PDI watchdog (default: 100 ms)

SM Watchdog (SyncManagerWatchdog)

The SyncManager watchdog is reset after each successful EtherCAT process data communication with the terminal/box. If no EtherCAT process data communication takes place with the terminal/box for longer than the set and activated SM watchdog time, e.g. in the event of a line interruption, the watchdog is triggered and the outputs are set to FALSE. The OP status of the terminal/box is unaffected by this. The watchdog is only reset after a successful EtherCAT process data access. Set the monitoring time as specified below.

The SyncManager watchdog monitors correct and timely process data communication with the ESC from the EtherCAT side.

PDI watchdog (process data watchdog)

If no PDI communication with the EtherCAT slave controller (ESC) takes place for longer than the set and activated PDI watchdog time, this watchdog is triggered.

PDI (Process Data Interface) is the internal interface between the ESC and local processors in the EtherCAT slave, for example. The PDI watchdog can be used to monitor this communication for failure.

The PDI watchdog monitors correct and timely process data communication with the ESC but from the application side.

The SM and PDI watchdogs should be set separately for each slave in the TwinCAT System Manager:

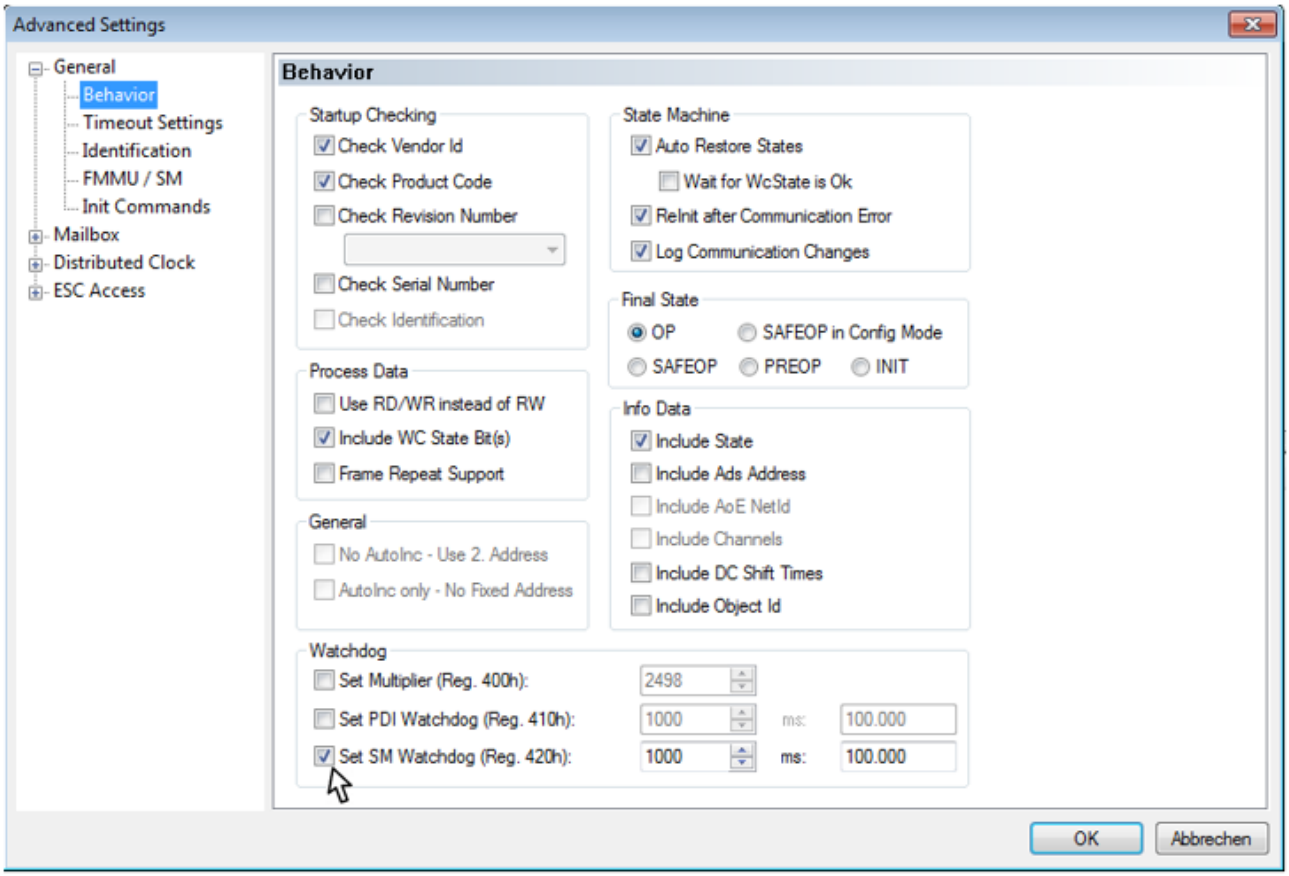


Fig. 38: EtherCAT tab --> Advanced settings --> Behavior --> Watchdog

Comments:

- The multiplier applies to both watchdogs.
- Each watchdog has its own timer setting, which together with the multiplier results in a time.
- Important: The multiplier/timer setting is loaded into the slave on start-up, if the corresponding checkbox is ticked. If the checkbox is not ticked, no download takes place, and the ESC setting remains unchanged.

Multiplier

Both watchdogs receive their pulses from the local terminal/box clock, divided by the watchdog multiplier.

1/25 MHz * (watchdog multiplier + 2) = 100 µs (for default setting of 2498 for the multiplier)

The standard setting of 1000 for the SM watchdog corresponds to a release time of 100 ms.

The value in multiplier + 2 corresponds to the number of basic 40 ns ticks representing a watchdog tick.

The multiplier can be modified in order to adjust the watchdog time over a larger range.

Example "Set SM watchdog"

This checkbox enables manual setting of the watchdog times. If the outputs are set and the EtherCAT communication is interrupted, the SM watchdog is triggered after the set time and the outputs are deleted. This setting can be used for adapting a terminal to a slower EtherCAT master or long cycle times. The default SM watchdog setting is 100 ms. The setting range is from 0 to 65535. Together with a multiplier in a range from 1 to 65535, this covers a watchdog period of 0 to ~170 seconds.

Calculation

Multiplier = 2498 → watchdog base time = 1 / 25 MHz * (2498 + 2) = 0.0001 seconds = 100 µs

SM watchdog = 10000 → 10000 * 100 µs = 1 second watchdog monitoring time

⚠ CAUTION

Caution! Unintended behavior of the system is possible!

The function for switching off of the SM watchdog via SM watchdog = 0 is only implemented in terminals from version -0016. In previous versions this operating mode should not be used.

⚠ CAUTION

Caution! Damage to the equipment and unintended behavior of the system is possible!

If the SM watchdog is activated and a value of 0 is entered the watchdog switches off completely. This is watchdog deactivation! Outputs are then *NOT* set to a safe state, in the event of an interruption in communication!

Outputs in SAFEOP

Watchdog monitoring is activated by default. It sets the outputs in the module to a safe state (e.g. OFF), depending on the *SAFEOP* and *OP* settings, and depending on the device and its settings. If this is prevented due to deactivation of watchdog monitoring in the module, outputs can be switched or remain set in device state *SAFEOP*.

4.3 EtherCAT State Machine

The state of the EtherCAT slave is controlled via the EtherCAT State Machine (ESM). Depending upon the state, different functions are accessible or executable in the EtherCAT slave. Specific commands must be sent by the EtherCAT master to the device in each state, particularly during the bootup of the slave.

A distinction is made between the following states:

- Init
- Pre-Operational
- Safe-Operational and
- Operational
- Boot

The regular state of each EtherCAT slave after bootup is the OP state.

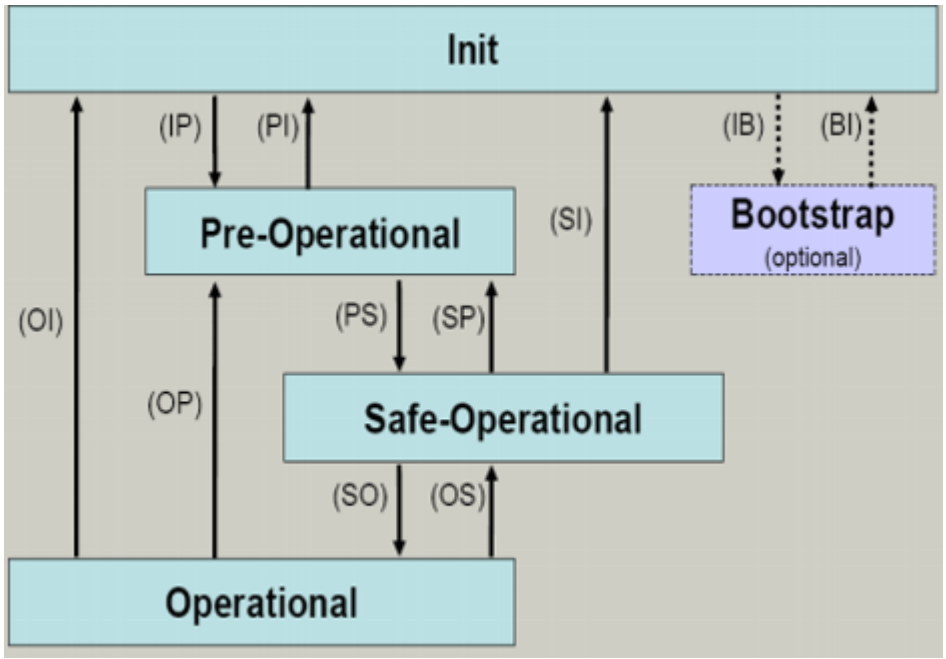


Fig. 39: EtherCAT State Machine

Init

After switch-on the EtherCAT slave in the *Init* state. No mailbox or process data communication is possible. The EtherCAT master initializes sync manager channels 0 and 1 for mailbox communication.

Pre-Operational (Pre-Op)

During the transition between *Init* and *Pre-Op* the EtherCAT slave checks whether the mailbox was initialized correctly.

In *Pre-Op* state mailbox communication is possible, but not process data communication. The EtherCAT master initializes the sync manager channels for process data (from sync manager channel 2), the FMMU channels and, if the slave supports configurable mapping, PDO mapping or the sync manager PDO assignment. In this state the settings for the process data transfer and perhaps terminal-specific parameters that may differ from the default settings are also transferred.

Safe-Operational (Safe-Op)

During transition between *Pre-Op* and *Safe-Op* the EtherCAT slave checks whether the sync manager channels for process data communication and, if required, the distributed clocks settings are correct. Before it acknowledges the change of state, the EtherCAT slave copies current input data into the associated DP-RAM areas of the EtherCAT slave controller (ECSC).

Mailbox and process data communication is possible in the *Safe-Op* state, but the slave keeps its outputs in the safe state. However, the input data are cyclically updated.

Operational (Op)

Before the EtherCAT master switches the EtherCAT slave from *Safe-Op* to *Op* it must transfer valid output data.

In the *Op* state the slave copies the output data of the masters to its outputs. Process data and mailbox communication is possible.

Boot

In the *Boot* state the slave firmware can be updated. The *Boot* state can only be reached via the *Init* state.

In the *Boot* state mailbox communication via the *file access over EtherCAT* (FoE) protocol is possible, but no other mailbox communication and no process data communication.

4.4 CoE interface

General description

The CoE interface (CANopen over EtherCAT) is used for parameter management of EtherCAT devices. EtherCAT slaves or the EtherCAT master manage fixed (read only) or variable parameters which they require for operation, diagnostics or commissioning.

CoE parameters are arranged in a table hierarchy. In principle, the user has read access via the fieldbus. The EtherCAT master (TwinCAT System Manager) can access the local CoE lists of the slaves via EtherCAT in read or write mode, depending on the properties.

Different CoE parameter types are possible, including string (text), integer numbers, Boolean values or larger byte fields. They can be used to describe a wide range of features. Examples of such parameters include manufacturer ID, serial number, process data settings, device name, calibration values for analog measurement or passwords.

Organization takes place on 2 levels by means of hexadecimal numbering: the (main) index is named first, then the subindex. The value ranges are:

- Index 0 to 65535
- Subindex: 0…255

A parameter localized in this way is normally written as 0x8010:07, with preceding "0x" to identify the hexadecimal numerical range and a colon between index and subindex.

The relevant ranges for EtherCAT fieldbus users are:

- 0x1000: This is where fixed identity information for the device is stored, including name, manufacturer, serial number etc., plus information about the current and available process data configurations.
- 0x8000: This is where the operational and functional parameters for all channels are stored, such as filter settings or output frequency.

Other important ranges are:

- 0x4000: In some EtherCAT devices the channel parameters are stored here (as an alternative to the 0x8000 range).
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

Availability

Not every EtherCAT device must have a CoE list. Simple I/O modules without dedicated processor usually have no variable parameters and therefore no CoE list.

If a device has a CoE list, it is shown in the TwinCAT System Manager as a separate tab with a listing of the elements:

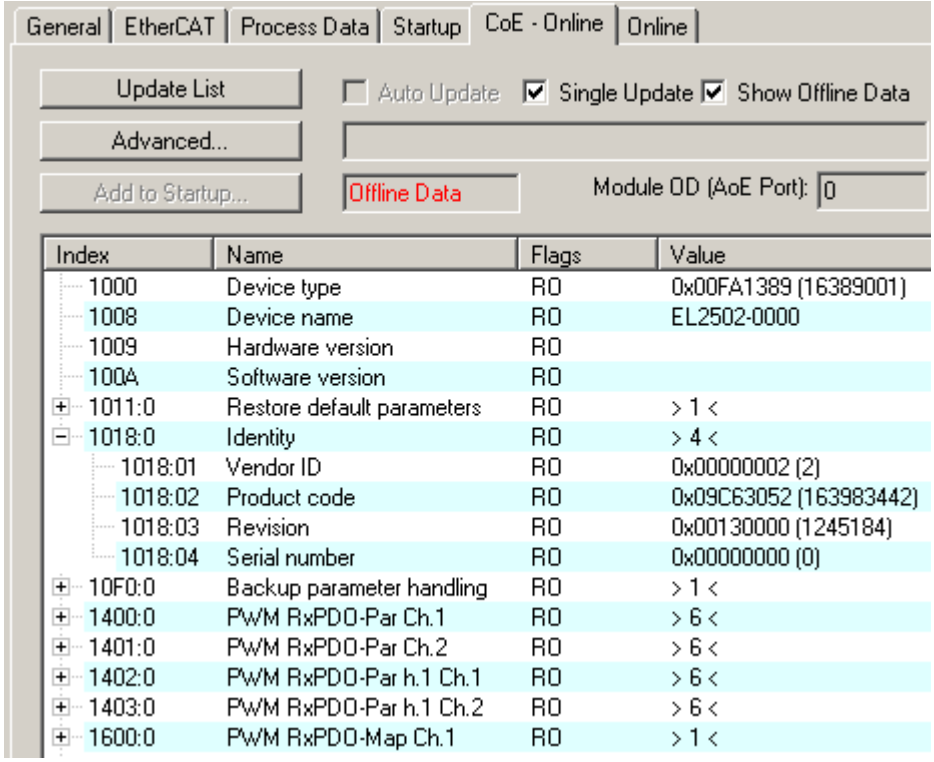


Fig. 40: CoE-Online tab

The CoE objects from 0x1000 to 0x1600, which are available in the example device "EL2502", can be seen in the above figure; the subindices from 0x1018 are expanded.

Data management

Some parameters, particularly the setting parameters of the slave, are configurable and writeable. This can be done in write or read mode

- via the System Manager (figure above) by clicking. This is useful for commissioning of the system/ slaves. Click on the row of the index to be parameterized and enter a value in the *SetValue* dialog.
- from the control system/PLC via ADS, e.g. through function blocks from the TcEtherCAT.lib library This is recommended for modifications while the system is running or if no System Manager or operating staff are available.

Data management

If CoE parameters on the slave are changed online, this is saved fail-safe in the device (EEPROM) in Beckhoff devices. This means that the changed CoE parameters are still retained after a restart. The situation may be different with other manufacturers.

Startup list

Startup list

Changes in the local CoE list of the terminal are lost **if the terminal is replaced**. If a terminal is replaced with a new Beckhoff terminal, it will have the factory settings. It is therefore advisable to link all changes in the CoE list of an EtherCAT slave with the **Startup list** of the slave, which is processed whenever the EtherCAT fieldbus is started. In this way a replacement EtherCAT slave can automatically be parameterized with the specifications of the user.

If EtherCAT slaves are used which are unable to store local CoE values permanently, the Startup list must be used.

Recommended approach for manual modification of CoE parameters

- Make the required change in the System Manager. The values are stored locally in the EtherCAT slave

- If the value is to be stored permanently, enter it in the Startup list. The order of the Startup entries is usually irrelevant.

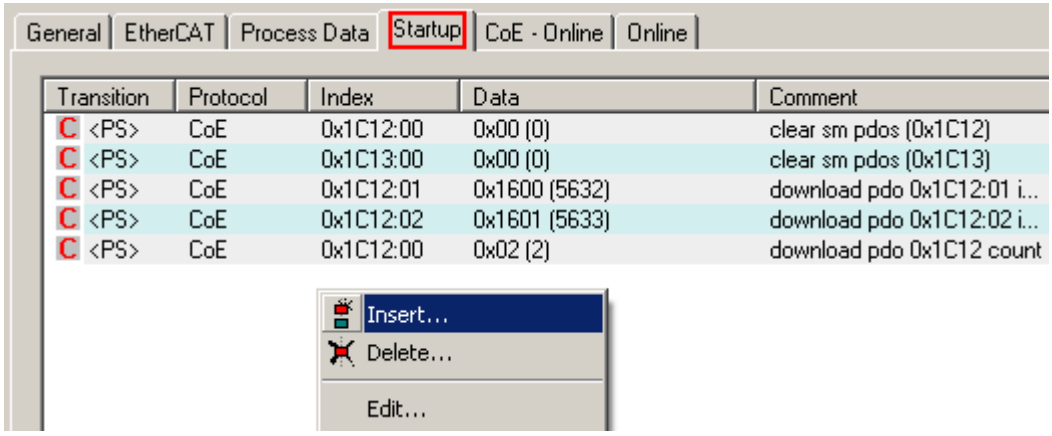


Fig. 41: Startup list in the TwinCAT System Manager

The Startup list may already contain values that were configured by the System Manager based on the ESI specifications. Additional application-specific entries can be created.

Online/offline directory

While working with the TwinCAT System Manager, a distinction has to be made whether the EtherCAT device is "available", i.e. switched on and linked via EtherCAT and therefore **online**, or whether a configuration is created **offline** without connected slaves.

In both cases a CoE directory is visible according to the figure "CoE-Online tab", but the connectivity is displayed as offline/online.

- If the slave is offline
  - the offline list from the ESI file is displayed. In this case modifications are not meaningful or possible.
  - the configured status is shown under Identity
  - no firmware or hardware version is displayed, since these are features of the physical device.
  - Offline is shown in red

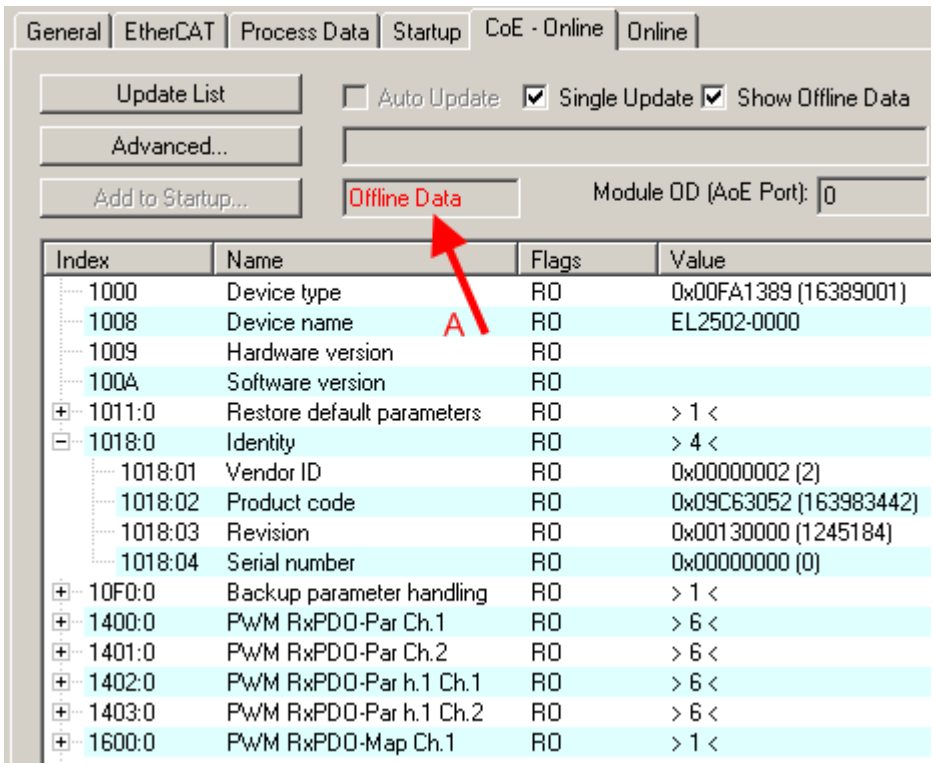


Fig. 42: Offline list

- If the slave is online
  - the actual current slave directory is read. This may take several seconds, depending on the size and cycle time.
  - the actual identity is displayed
  - the firmware and hardware version of the equipment according to the electronic information is displayed.
  - Online is shown in green

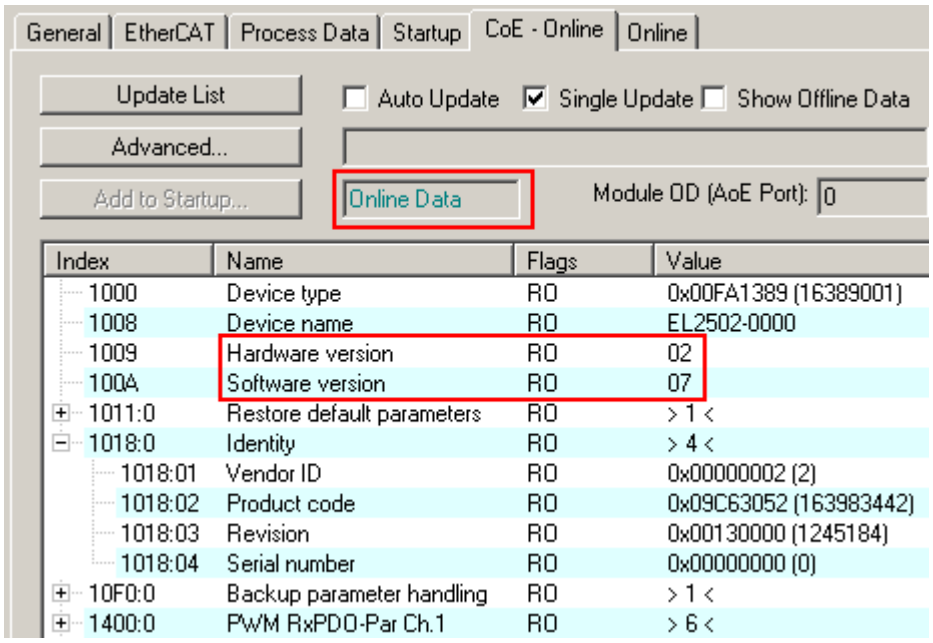


Fig. 43: Online list

Channel-based order

The CoE directory is located in EtherCAT devices that usually encompass several functionally equivalent channels. e.g. a 4-channel 0 – 10 V analog input terminal also has 4 logical channels and thus 4 identical sets of parameter data for the channels. In order to avoid having to list each channel in the documentation, the placeholder "n" tends to be used for the individual channel numbers.

In the CoE system 16 indices, each with 255 subindices, are generally sufficient for representing all channel parameters. The channel-based order is therefore arranged in $16_{dec}/10_{hex}$ steps. The parameter range 0x8000 exemplifies this:

- Channel 0: parameter range 0x8000:00 ... 0x800F:255
- Channel 1: parameter range 0x8010:00 ... 0x801F:255
- Channel 2: parameter range 0x8020:00 ... 0x802F:255
- …

This is generally written as 0x80n0. Detailed information on the CoE interface can be found in the EtherCAT system documentation on the Beckhoff website.

5 Commissioning/Configuration

5.1 EP600x-0002 - Interface modes

Via CoE objects the following settings can be done for the interfaces:

Parameterization

The module is parameterized via the CoE - Online tab (with a double-click on the corresponding object). Only the mandatory parameters for the respective interface mode are specified here. Further settings may be possible.

RS232: point-to-point connection to an RS232 device

Direct connection to an RS232 end device, full duplex data transmission (default setting).

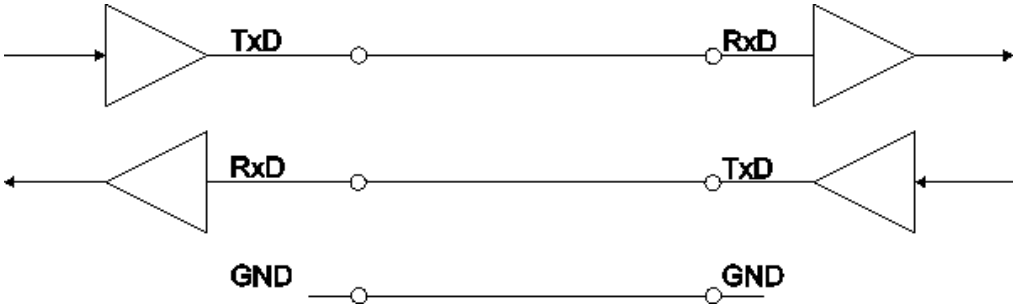


Fig. 44: Point-to-point connection to an RS232 device

The following CoE objects must be set

Index	Name	Meaning		Data type	Flags	Setting
F800:0n	Interface type Ch n	0x00	RS232	BIT1	RW	0x00 (0_{dec}) (default)
		0x01	RS485/422			

RS422: 4-wire point-to-point connection to an RS422 device

Direct connection to an RS422 end device, full duplex data transmission.
Data can be transmitted in full duplex in RS422 mode. Only point-to-point connections can be established.

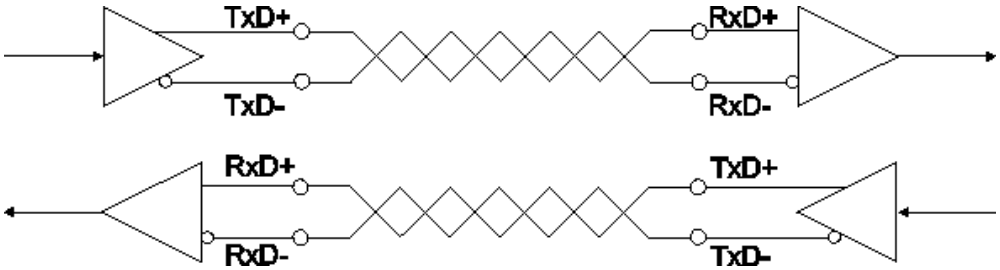


Fig. 45: 4-wire point-to-point connection to an RS422 device

The following CoE objects must be set

Index	Name	Meaning		Data type	Flags	Setting
F800:0n	Interface type Ch n	0x00	RS232	BIT1	RW	0x01 (1_{dec})
		0x01	RS485/422			

Index	Name	Meaning		Data type	Flags	Setting
80n0:07	Enable point-to-point connection (RS422) Channel n	0_{bin}	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	1_{bin}
		1_{bin}	The module is used for a point-to-point connection (RS422).			

RS485: 2-wire connection in bus structure to RS485 device(s)

Bus structure, half duplex data transmission

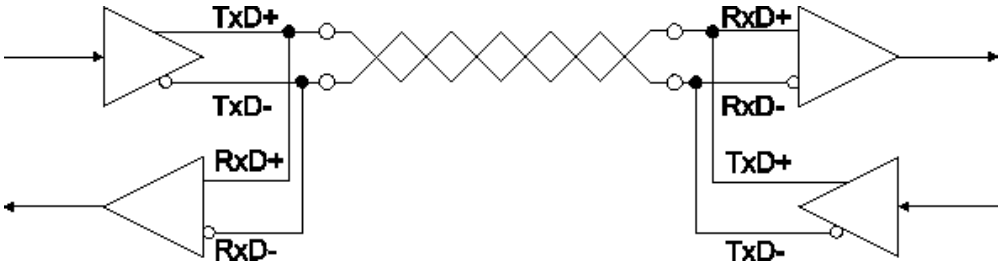


Fig. 46: 2-wire connection in bus structure to RS485 device(s)

The following CoE objects must be set

Index	Name	Meaning		Data type	Flags	Setting
F800:0n	Interface type Ch n	0x00	RS232	BIT1	RW	0x01 (1_{dec})
		0x01	RS485/422			

Index	Name	Meaning		Data type	Flags	Setting
80n0:06	Enable half duplex channel n	0_{bin}	Full duplex: transmitted data are monitored. The bit has no effect in RS232 and RS422 mode	BOOLEAN	RW	1_{bin} (default)
		1_{bin}	Half duplex: The reception of the data transmitted by the box itself is suppressed			

Index	Name	Meaning		Data type	Flags	Setting
80n0:07	Enable point-to-point connection (RS422) Channel n	0_{bin}	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0_{bin}
		1_{bin}	The module is used for a point-to-point connection (RS422).			

Deactivated receive driver

The receive driver is deactivated during the transmission procedure. The transmitted data are not monitored!

RS485: 2-wire connection with external bridge in bus structure to RS485 device(s)

Bus structure, half duplex data transmission with diagnosis of the transmitted data

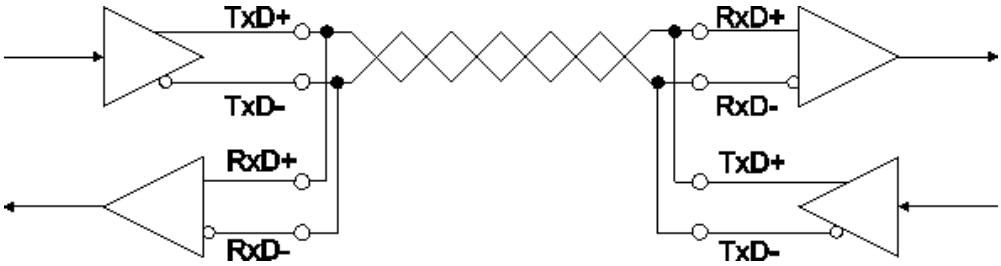


Fig. 47: 2-wire connection with external bridge in bus structure to RS485 device(s)

The following CoE objects must be set

Index	Name	Meaning		Data type	Flags	Setting
F800:0n	Interface type Ch n	0x00	RS232	BIT1	RW	0x01 (1_{dec})
		0x01	RS485/422			

Index	Name	Meaning		Data type	Flags	Setting
80n0:06	Enable half duplex channel n	0_{bin}	Full duplex: transmitted data are monitored. The bit has no effect in RS232 and RS422 mode	BOOLEAN	RW	0_{bin}
		1_{bin}	Half duplex: The reception of the data transmitted by the box itself is suppressed			

Index	Name	Meaning		Data type	Flags	Setting
80n0:07	Enable point-to-point connection (RS422) Channel n	0_{bin}	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0_{bin}
		1_{bin}	The module is used for a point-to-point connection (RS422).			

Activated receive driver (from firmware version 03)

The receive driver remains activated during the transmission procedure. The transmitted data are monitored! A conditional diagnosis of the line is thus possible. If there is a discrepancy between the transmitted data and the monitored data, it may be assumed that a further receiver also cannot receive these data flawlessly. In this case, check the bus line!

Also see about this

Configuration by means of the TwinCAT System Manager [▶ 70]

5.2 EP600x-0002 - Basic function principles

The EP6002-0002 2-channel serial interface module enables the connection of two devices with an RS232 or RS485/RS422 interface. The EP6001-0002 1-channel serial interface module enables the connection of one device with an RS232 or RS485/RS422 interface and in addition the connection of digital inputs/outputs.

EP6002-0002 - two configurable interfaces

The module has two physical interfaces, which can each be configured as RS232 or RS422/485.

Interface 1 of the EP6002-0002

- RS232 on M12 socket 1 **or**
- RS422/485 on M12 socket 2

Interface 2 of the EP6002-0002

- RS232 on M12 socket 3 **or**
- RS422/485 on M12 socket 4

The receive buffer has 864 bytes, the send buffer 128 bytes. The factory setting of the module is 9600 baud, 8 data bits, 1 stop bit, no parity.

No hardware flow control takes place; however, software flow control is possible via XON, XOFF.

EP6001-0002 - One configurable interface

The module has one physical interface, which can be configured as RS232 or RS422/485. In addition, the box allows the connection of digital inputs/outputs on M12 socket 4.

Interface 1 of the EP6001-0002

- RS232 on M12 socket 1 **or**
- RS485/422TX on M12 socket 2 **or**
- RS422RX on M12 socket 3

Communication between PLC and EP600x-0002

Communication takes place

- as with a COM port using the virtual serial COM driver or
- via control word and status word

Sending data

You can transmit up to 22 bytes of data to the module in one cycle via DataOut 0 ... DataOut 21.

- Set the *Output Length* parameter in the control byte to the number of bytes to be transmitted.
- Toggle the *Transmit Request* bit in the control byte.
- The module acknowledges the data transmission in the status byte via the *Transmit Accepted* parameter.

Receiving data

If the module toggles the *Receive Request* bit in the status byte, there are new receive data

- Read the *Input Length* parameter from the status byte. It contains the number of bytes to be received.
- The data are provided in DataOut 0 ... DataOut 21. The first datum is contained in DataIn 0.
- After reading the data, acknowledge this by toggling the *Receive Request* bit in the control byte. The module only makes new data available after that.

Interface level

The EP600x-0002 module operates at RS232 level with respect to GND or with differential RS485/422 level.

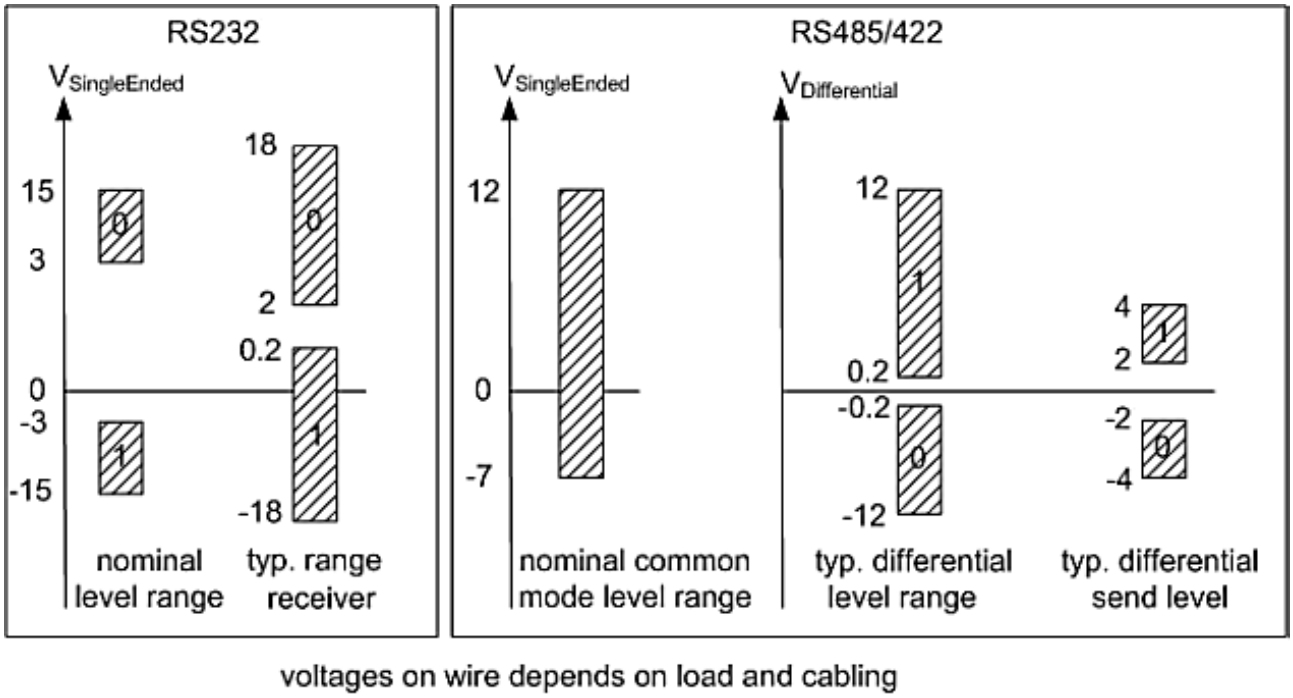


Fig. 48: Level of RS232, RS485/RS422 interfaces

Process data

As delivered, 22 bytes of user data and 1 control/status word are transferred.

Parameterization via CoE (index 0x80n0)

0x80n0 [▶ 91] Parameterization via CoE

The parameterization of the module can be set in the CoE (CAN over EtherCAT) list.

Parameterization via the CoE list (CAN over EtherCAT)

Please note the following general CoE notes when using/manipulating the CoE parameters: - Keep a startup list if components have to be replaced - Differentiation between online/offline dictionary, existence of current XML description - Use "CoE reload" for resetting changes

The following CoE settings are possible from object 0x8000 of the EP6002-0002 and are shown here in the default settings:

8000:0	COM Settings Ch.1	RW	> 26 <
8000:02	Enable XON/XOFF supported tx data	RW	FALSE
8000:03	Enable XON/XOFF supported rx data	RW	FALSE
8000:04	Enable send FIFO data continuous	RW	FALSE
8000:05	Enable transfer rate optimization	RW	TRUE
8000:06	Enable half duplex	RW	TRUE
8000:07	Enable point to point connection (RS...	RW	FALSE
8000:11	Baudrate	RW	9600 Baud (6)
8000:15	Data frame	RW	8N1 (3)
8000:1A	Rx buffer full notification	RW	0x0360 (864)
8010:0	COM Settings Ch.2	RW	> 26 <
8010:02	Enable XON/XOFF supported tx data	RW	FALSE
8010:03	Enable XON/XOFF supported rx data	RW	FALSE
8010:04	Enable send FIFO data continuous	RW	FALSE
8010:05	Enable transfer rate optimization	RW	TRUE
8010:06	Enable half duplex	RW	TRUE
8010:07	Enable point to point connection (RS...	RW	FALSE
8010:11	Baudrate	RW	9600 Baud (6)
8010:15	Data frame	RW	8N1 (3)
8010:1A	Rx buffer full notification	RW	0x0360 (864)

Fig. 49: EP6002-0002 - CoE settings on object 0x8000 (default)

Process data description

The process data are generated from CoE objects 0x6000 (Inputs) [▶ 107] and 0x7000 (Outputs) [▶ 109] and are described in chapter Object description and parameterization [▶ 90].

Transfer rates

The EP boxes have a process image of 22 bytes of user data. It possible to transmit or receive these 22 bytes every second cycle at the most.

The data is transferred from the EP box to the controller in the first cycle. In the second cycle, the controller must acknowledge that it has accepted the data.

Therefore, if the cycle time is 10 ms, 50 times 22 bytes can be transmitted per second.

With a set data frame of 8N1, each transmitted byte consists of a start bit, eight data bits and a stop bit. This is equivalent to 10 bits per user byte.

With the above mentioned settings, a **continuous** data transfer rate of:

- 50[1/s] x 22[bytes] x 10[bits] = 11000 bps

can be achieved.

The next lower baud rate is 9600 baud. Accordingly, continuous transfer at a maximum baud rate of 9600 can be secured with a cycle time of 10 ms.

If only low quantities of data are to be transmitted or received sporadically (e.g. bar code scanner) the baud rate can also be set higher, or the cycle time can be enlarged.

If the controller cannot collect the data quickly enough from the EP box, the data will be stored intermediately in the internal buffer of the EP box. The buffer for received data has a size of 864 bytes. If this is exhausted, all further data will be lost.

A buffer is also available for the transmit data. With a "baud rate" setting of 300 and a "data frame" setting of 8N1, the EP box can only transmit 30 bytes per second. However, if more than these 30 bytes per second are received, a 128 bytes transmit buffer will be written to first in this case also. Once this is full, all further data will be lost.

Optimization of transfer rates

In normal operating mode the data received will be adopted immediately into the process image. In order to enable a contiguous data stream, the "Enable transfer rate optimization" option in the Settings object is activated by default. Due to this switch, the data will first be stored intermediately in the receive buffer (864 bytes).

The data will only be copied into the process image if no further character is received for 16 bit times or if the buffer is full.

Continuous transmission of data

A continuous data stream is indispensable for many applications. For this purpose, the Beckhoff modules feature the "Enable send FIFO data continuous" setting in the Settings object. The internal transmit buffer (128 bytes) of the EP box can be filled first by setting this switch. After that the entire contents of the buffer can be transmitted without interruption. To this end, data will be sent from the controller to the EP box as in a normal transmission. The data from the buffer is only sent with a rising edge of the "Send continuous" bit. If the data has been transferred, the EP box informs the controller by setting the "Init accepted" bit. "Init accepted" is cleared with "SendContinuous".

Prioritization

Since received data normally cannot be repeated from the other transmitter, they have a higher priority in the module than data to be transmitted.
Furthermore, the priority decreases as the channel number increases. Hence, the reception of data on channel 1 has the highest priority.

Data transfer examples

Initialization

Initialization is performed prior to the first transmission/reception. The module is thereby parameterized with the data from the corresponding Settings object.

Procedure:

1. Set "Init request" to 1
2. The module confirms successful initialization by setting "Init accepted".
3. Reset "Init request"
4. The module sets "Init accepted" to 0.

The module is now ready for data exchange.

Data transmission from the controller to the module (send 2 characters)

1. Set "Output length" to 2
2. Fill "Data Out 0" and "Data Out 1" with user data
3. Change the state of "Transmit request"
4. The module acknowledges receipt by changing the state of the "Transmit accepted" bit.

Data transmission from the module to the controller (receive characters)

1. The module indicates that there is new data in the process image by changing the state of the "Receive request" bit.
2. The number of bytes received is written in "Input length"
3. The controller acknowledges acceptance of the bytes by changing the state of “Receive request”.

5.3 Insertion in the EtherCAT network

Installation of the latest XML device description

Please ensure that you have installed the corresponding latest XML device description in TwinCAT. This can be downloaded from the Beckhoff website (http://www.beckhoff.de/german/default.htm?download/elconfg.htm) and installed according to the installation instructions.

The configuration tree in the Beckhoff TwinCAT System Manager can be created in 2 ways:

- by scanning [▶ 63] of existing hardware (referred to as "online"), or
- by manually inserting/appending [▶ 63] fieldbus devices, couplers and slaves.

Automatic scanning of the module

- The EtherCAT system must be in a safe, de-energized state before you connect the EtherCAT modules to the EtherCAT network.
- After the operating voltage is switched on, open the TwinCAT System Manager (Config mode) and scan the devices (see fig. below). Acknowledge all dialogs with "*OK*", so that the configuration is in "FreeRun" mode.

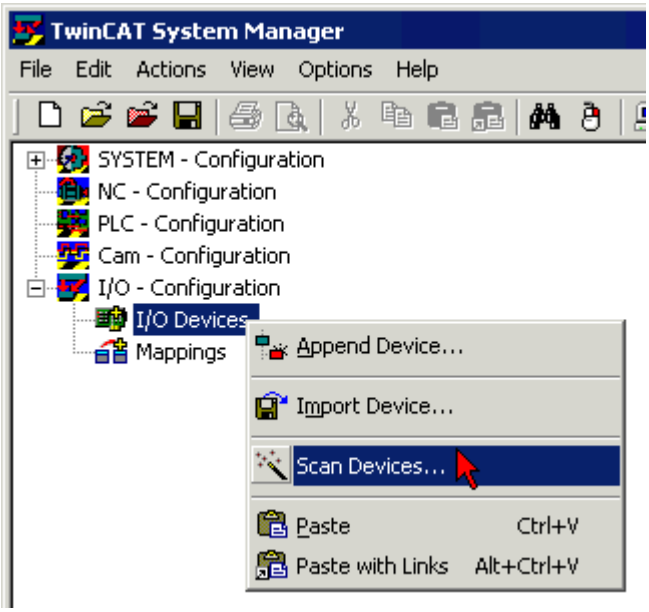


Fig. 50: Scanning in the EtherCAT configuration (I/O Devices-> right-click -> Scan Devices...

Appending a module manually

- The EtherCAT system must be in a safe, de-energized state before you connect the EtherCAT modules to the EtherCAT network.
- Switch on the operating voltage, open the TwinCAT System Manager (Config mode)
- Append a new I/O device (see fig. below).

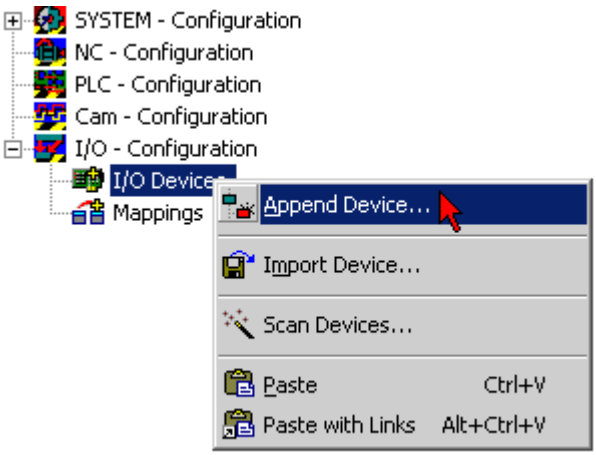


Fig. 51: *Appending a new I/O device (I/O Devices-> right-click -> Append Device...)*

In the following dialog, select the device *EtherCAT (Direct Mode)*, see following fig., and confirm with *OK*.

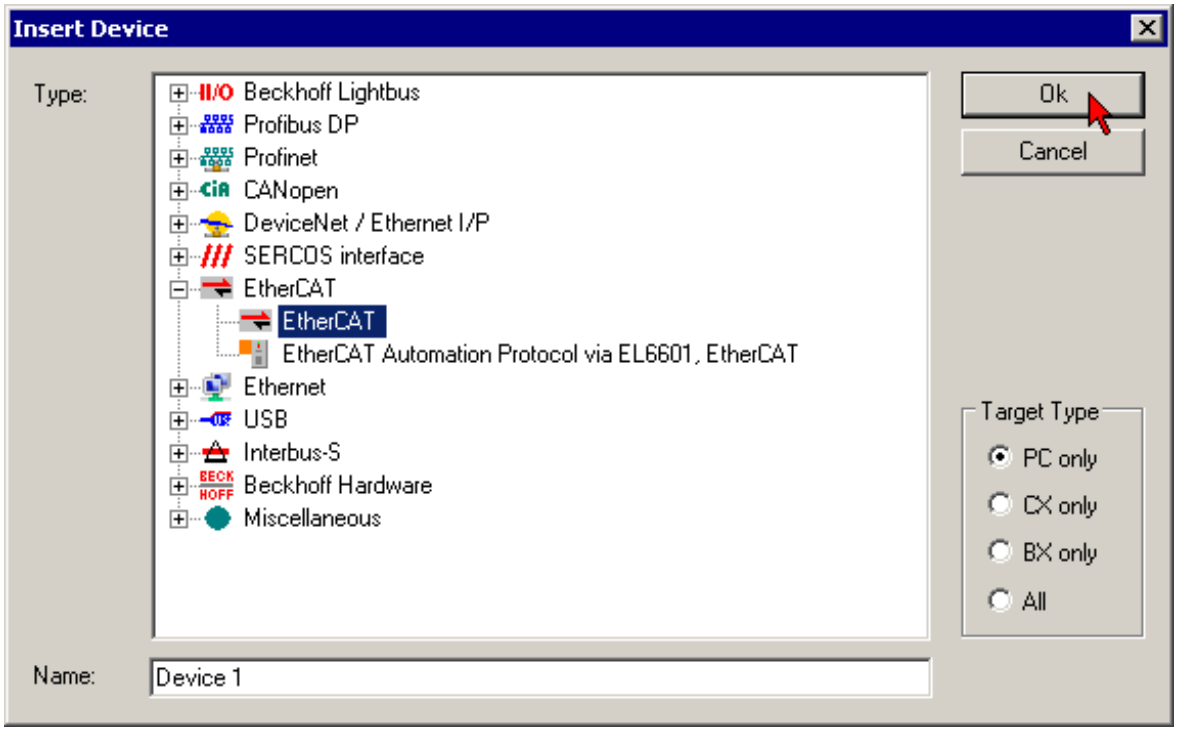


Fig. 52: *Selecting the device (EtherCAT)*

- Append a new box (see fig. below).

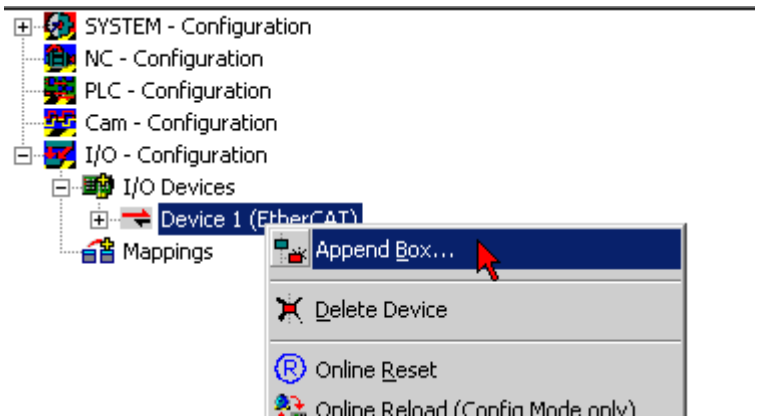


Fig. 53: *Appending a new box (Device -> right-click -> Append Box...))*

- In the dialog shown, select the desired box and confirm with *OK*.

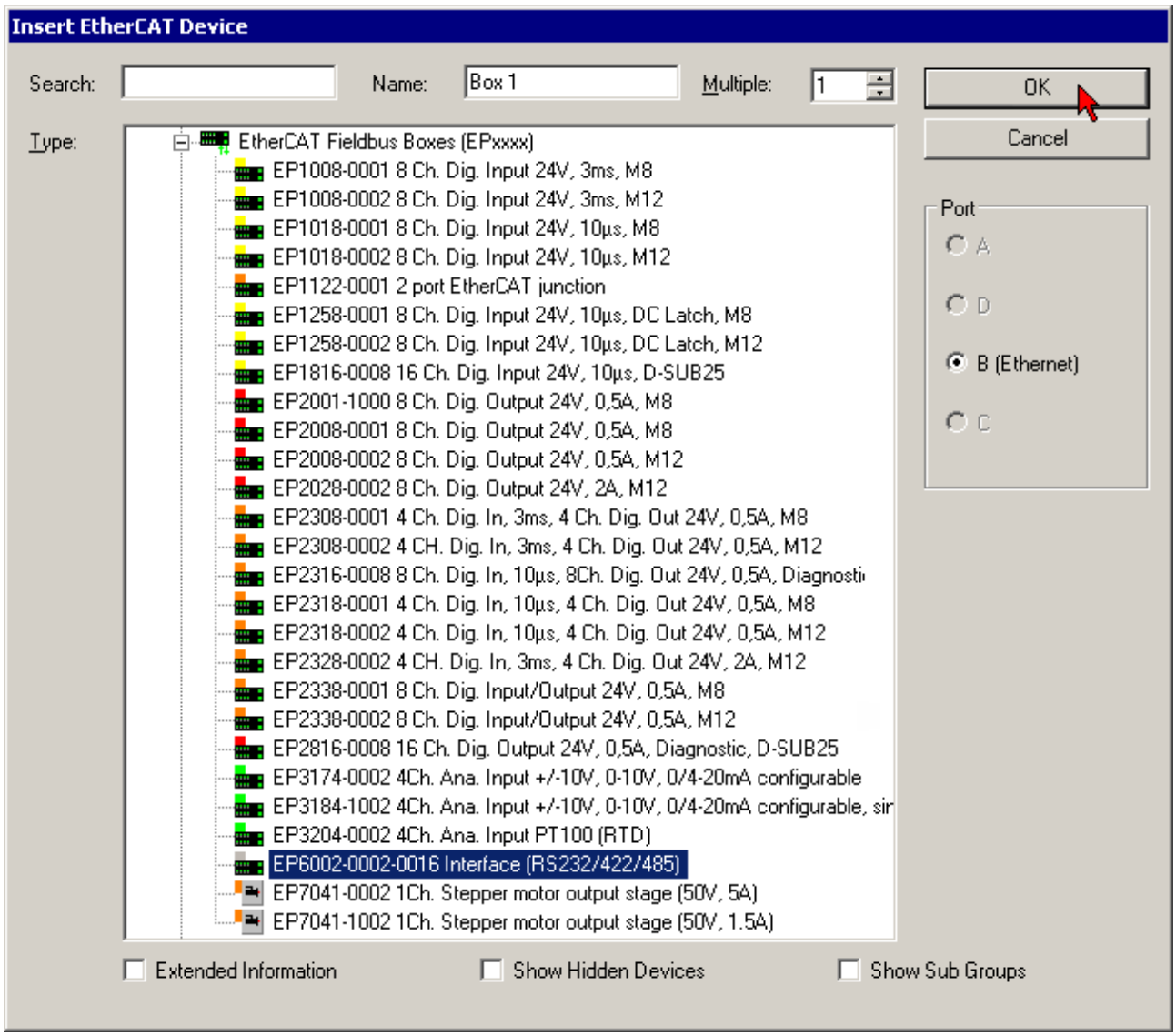


Fig. 54: *Selection of a box*

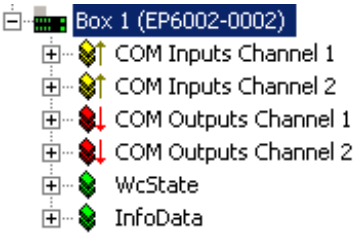


Fig. 55: *Appended box in the TwinCAT tree*

5.4 Configuration by means of the TwinCAT System Manager

In the left-hand window of the TwinCAT System Manager, click on the EtherCAT Box branch you wish to configure.

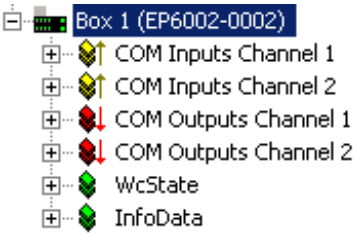


Fig. 56: TwinCAT System Manager - tree branch of the EtherCAT Box

In the right-hand window of the TwinCAT System Manager various tabs for configuring the EtherCAT Box are now available.

General tab

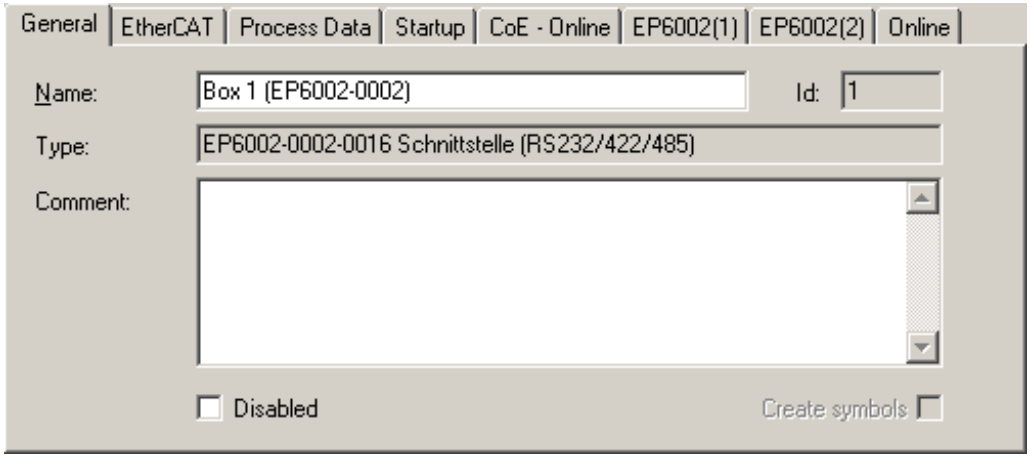


Fig. 57: General tab

Name	Name of the EtherCAT device
Id	Number of the EtherCAT device
Type	EtherCAT device type
Comment	Here you can add a comment (e.g. regarding the system).
Disabled	Here you can deactivate the EtherCAT device.
Create symbols	Access to this EtherCAT slave via ADS is only available if this checkbox is activated.

EtherCAT tab

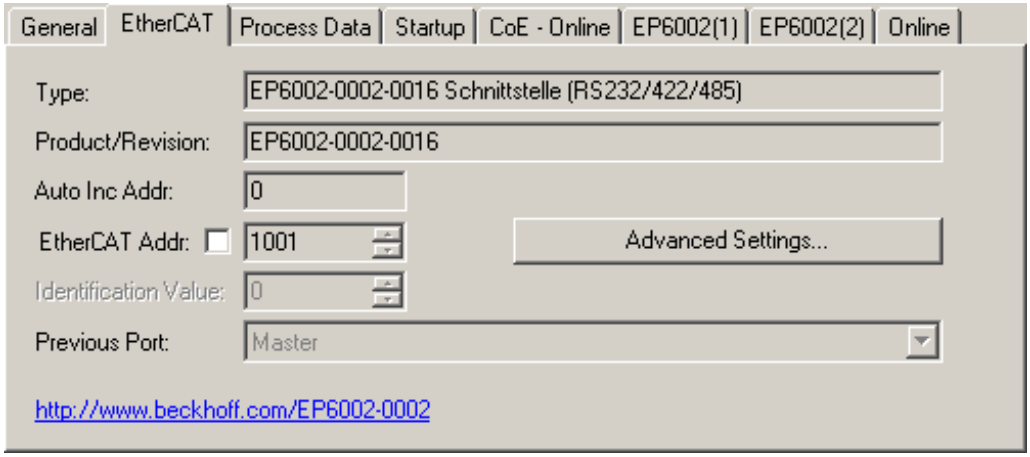


Fig. 58: EtherCAT tab

Type	EtherCAT device type
Product/Revision	Product and revision number of the EtherCAT device
Auto Inc Addr.	Auto increment address of the EtherCAT device. The auto increment address can be used for addressing each EtherCAT device in the communication ring through its physical position. Auto increment addressing is used during the start-up phase when the EtherCAT master allocates addresses to the EtherCAT devices. With auto increment addressing the first EtherCAT slave in the ring has the address 0000_{hex}. For each further slave the address is decremented by 1 ($FFFF_{hex}$, $FFFE_{hex}$ etc.).
EtherCAT Addr.	Fixed address of an EtherCAT slave. This address is allocated by the EtherCAT master during the start-up phase. Tick the checkbox to the left of the input field in order to modify the default value.
Previous Port	Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combobox is activated and the EtherCAT device to which this device is to be connected can be selected.
Advanced Settings	This button opens the dialogs for advanced settings.

The link at the bottom of the tab points to the product page for this EtherCAT device on the web.

Process Data tab

Indicates the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (PDO). The user can select a PDO via PDO assignment and modify the content of the individual PDO via this dialog, if the EtherCAT slave supports this function.

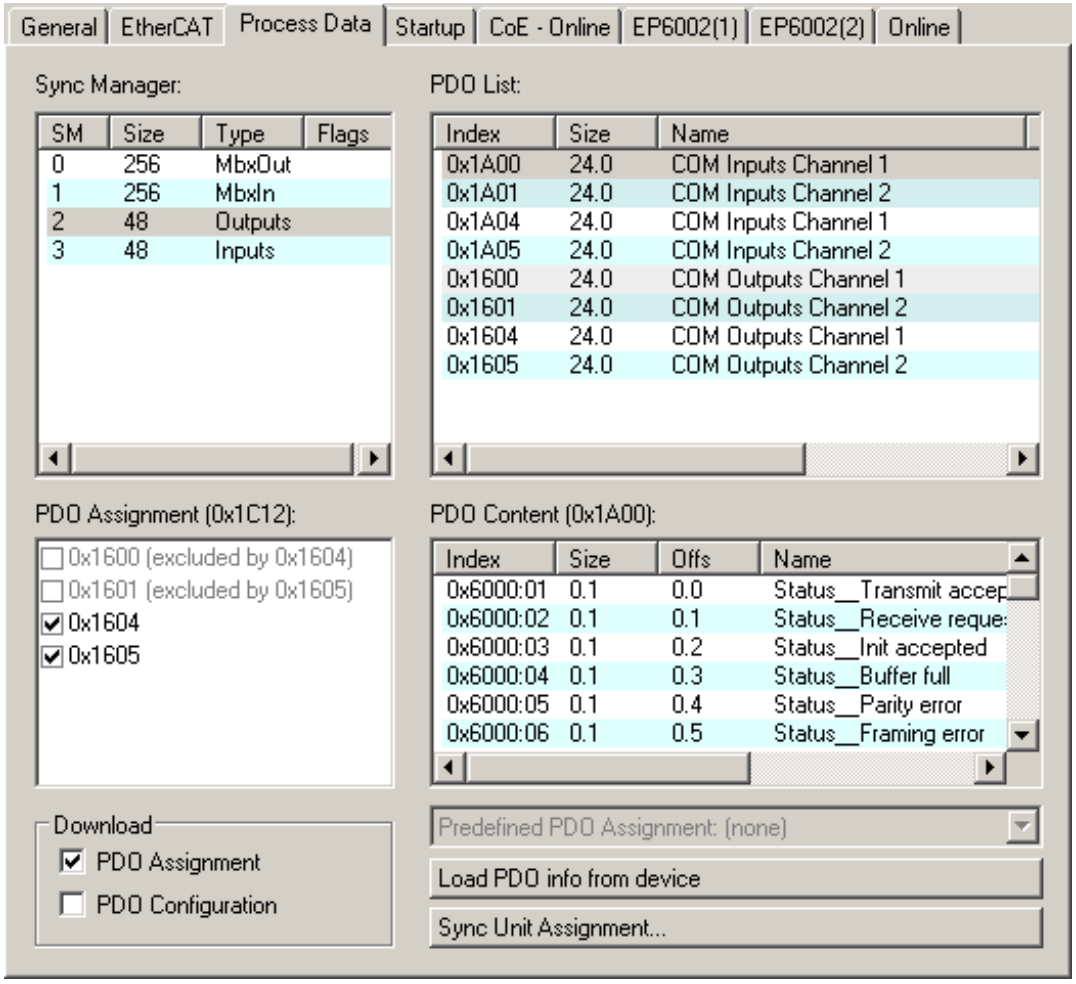


Fig. 59: Process Data tab

Sync-Manager

Lists the configuration of the Sync Manager (SM).
If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn).
SM2 is used for the output process data (outputs) and SM3 (inputs) for the input process data.

If an input is selected, the corresponding PDO assignment is displayed in the *PDO Assignment* list below.

PDO Assignment

PDO assignment of the selected Sync Manager. All PDOs defined for this Sync Manager type are listed here:

- If the output Sync Manager (outputs) is selected in the Sync Manager list, all RxPDOs are displayed.
- If the input Sync Manager (inputs) is selected in the Sync Manager list, all TxPDOs are displayed.

The selected entries are the PDOs involved in the process data transfer. In the tree diagram of the System Manager these PDOs are displayed as variables of the EtherCAT device. The name of the variable is identical to the *Name* parameter of the PDO, as displayed in the PDO list. If an entry in the PDO assignment list is deactivated (not selected and greyed out), this indicates that the input is excluded from the PDO assignment. In order to be able to select a greyed out PDO, the currently selected PDO has to be deselected first.

Activation of the PDO assignment

- the EtherCAT slave has to run through the PS state transition cycle (from pre-operational to safe-operational) once (see Online tab [▶ 73]),
- and the System Manager has to reload the EtherCAT slaves ( button)

PDO list

List of all PDOs supported by this EtherCAT device. The content of the selected PDOs is displayed in the *PDO Content* list. The PDO configuration can be modified by double-clicking on an entry.

Column	Description	
Index	PDO index.	
Size	Size of the PDO in bytes.	
Name	Name of the PDO. If this PDO is assigned to a Sync Manager, it appears as a variable of the slave with this parameter as the name.	
Flags	F	Fixed content: The content of this PDO is fixed and cannot be changed by the System Manager.
	M	Mandatory PDO. This PDO is mandatory and must therefore be assigned to a Sync Manager! Consequently, this PDO cannot be deleted from the *PDO Assignment* list
SM	Sync Manager to which this PDO is assigned. If this entry is empty, this PDO does not take part in the process data traffic.	
SU	Sync Unit to which this PDO is assigned.	

PDO Content

Indicates the content of the PDO. If flag F (fixed content) of the PDO is not set the content can be modified.

Download

If the device is intelligent and has a mailbox, the configuration of the PDO and the PDO assignments can be downloaded to the device. This is an optional feature that is not supported by all EtherCAT slaves.

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the Startup [▶ 69] tab.

PDO Configuration

If this check box is selected, the configuration of the respective PDOs (as shown in the PDO list and the PDO Content display) is downloaded to the EtherCAT slave.

Startup tab

The *Startup* tab is displayed if the EtherCAT slave has a mailbox and supports the *CANopen over EtherCAT* (CoE) or *Servo drive over EtherCAT* protocol. This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the slave in the same order as they are shown in the list.

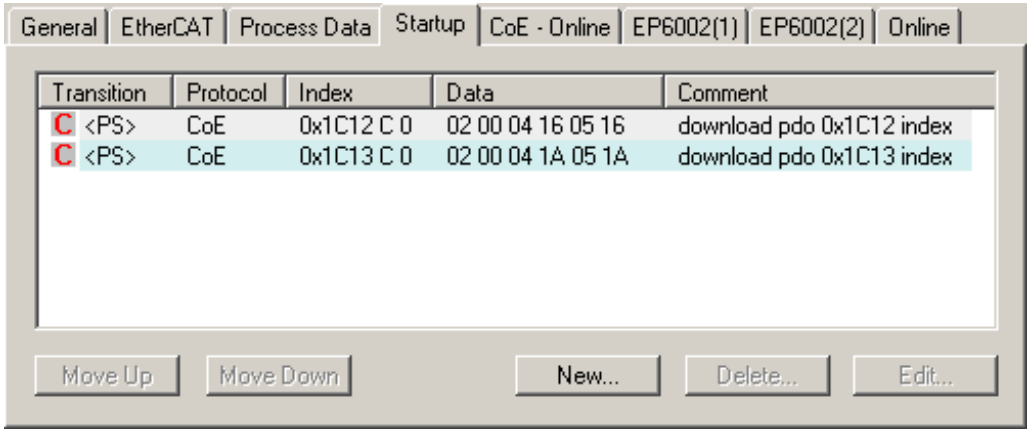


Fig. 60: Startup tab

Column	Description
Transition	Transition to which the request is sent. This can either be • the transition from pre-operational to safe-operational (PS), or • the transition from safe-operational to operational (SO). If the transition is enclosed in "<>" (e.g. <PS>), the mailbox request is fixed and cannot be modified or deleted by the user.
Protocol	Type of mailbox protocol
Index	Index of the object
Data	Date on which this object is to be downloaded.
Comment	Description of the request to be sent to the mailbox

Move Up	This button moves the selected request up by one position in the list.
Move Down	This button moves the selected request down by one position in the list.
New	This button adds a new mailbox download request to be sent during startup.
Delete	This button deletes the selected entry.
Edit	This button edits an existing request.

CoE - Online tab

The additional *CoE - Online* tab is displayed if the EtherCAT slave supports the *CANopen over EtherCAT* (CoE) protocol. This dialog lists the content of the object directory of the slave (SDO upload) and enables the user to modify the content of an object from this list. Details for the objects of the individual EtherCAT devices can be found in the device-specific object descriptions.

General | EtherCAT | Process Data | Startup | CoE - Online | EP6002 (1) | EP6002 (2) | Online

Update List | Auto Update | Single Update | Show Offline Data

Advanced...

Add to Startup... | Online Data | Module OD (AoE Port): 0

Index	Name	Flags	Value
1000	Device type	RO	0x02581389 (39326601)
1008	Device name	RO	EP6002-0002
1009	Hardware version	RO	00
100A	Software version	RO	00
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10F0:0	Backup parameter handling	RO	> 1 <
1400:0	COM RxPDO-Par Outputs Ch.1	RO	> 6 <
1401:0	COM RxPDO-Par Outputs Ch.2	RO	> 6 <
1404:0	COM RxPDO-Par Outputs Ch.1	RO	> 6 <
1405:0	COM RxPDO-Par Outputs Ch.2	RO	> 6 <
1600:0	COM RxPDO-Map Outputs Ch.1	RO	> 28 <
1601:0	COM RxPDO-Map Outputs Ch.2	RO	> 28 <
1604:0	COM RxPDO-Map Outputs Ch.1	RO	> 23 <
1605:0	COM RxPDO-Map Outputs Ch.2	RO	> 23 <
1800:0	COM TxPDO-Par Inputs Ch.1	RO	> 6 <
1801:0	COM TxPDO-Par Inputs Ch.2	RO	> 6 <
1804:0	COM TxPDO-Par Inputs Ch.1	RO	> 6 <
1805:0	COM TxPDO-Par Inputs Ch.2	RO	> 6 <
1A00:0	COM TxPDO-Map Inputs Ch.1	RO	> 31 <
1A01:0	COM TxPDO-Map Inputs Ch.2	RO	> 31 <
1A04:0	COM TxPDO-Map Inputs Ch.1	RO	> 23 <
1A05:0	COM TxPDO-Map Inputs Ch.2	RO	> 23 <
1C00:0	Sync manager type	RO	> 4 <
1C12:0	RxPDO assign	RW	> 2 <
1C13:0	TxPDO assign	RW	> 2 <
1C32:0	SM output parameter	RO	> 32 <
1C33:0	SM input parameter	RO	> 32 <
6000:0	COM Inputs Ch.1	RO	> 38 <
6001:0	Status Ch.1	RO	> 1 <
6010:0	COM Inputs Ch.2	RO	> 38 <
6011:0	Status Ch.2	RO	> 1 <
7000:0	COM Outputs Ch.1	RO	> 38 <
7001:0	Ctrl Ch.1	RO	> 1 <
7010:0	COM Outputs Ch.2	RO	> 38 <
7011:0	Ctrl Ch.2	RO	> 1 <
8000:0	COM Settings Ch.1	RW	> 26 <
8010:0	COM Settings Ch.2	RW	> 26 <
A000:0	COM Diag data Ch.1	RO	> 33 <
A010:0	COM Diag data Ch.2	RO	> 33 <
F000:0	Modular device profile	RO	> 2 <
F008	Code word	RW	0x00000000 (0)
F010:0	Module list	RW	> 2 <
F800:0	COM Settings	RW	> 3 <

Fig. 61: CoE - Online tab

Object list display

Column	Description	
Index	Index and subindex of the object	
Name	Name of the object	
Flags	RW	The object can be read, and data can be written to the object (read/write)
	RO	The object can be read, but no data can be written to the object (read only)
	P	An additional P identifies the object as a process data object.
Value	Value of the object	

Update List The *Update list* button updates all objects in the displayed list

Auto Update If this check box is selected, the content of the objects is updated automatically.

Advanced The *Advanced* button opens the *Advanced Settings* dialog. Here you can specify which objects are displayed in the list.

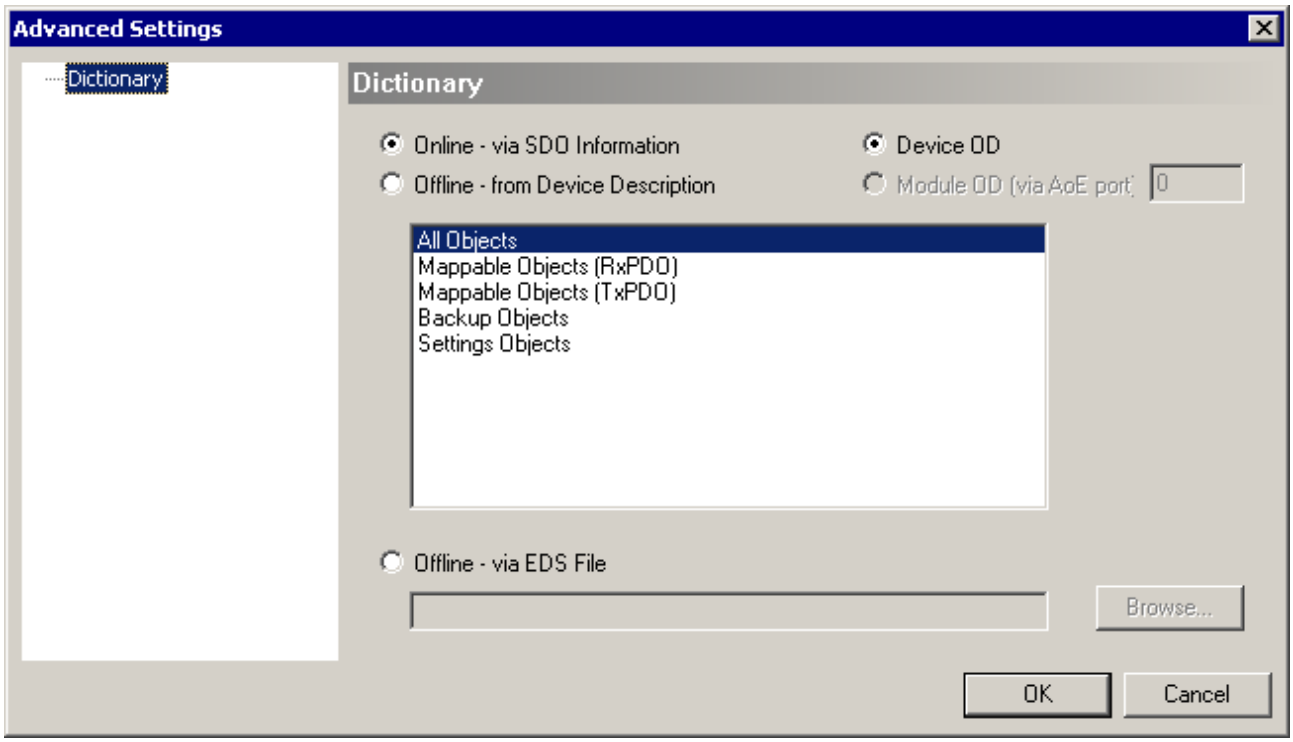


Fig. 62: Advanced Settings - Dictionary

Online - via SDO Information If this radio button is selected, the list of the objects included in the object directory of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.

Offline - via EDS File If this option button is selected, the list of the objects included in the object directory is read from an EDS file provided by the user.

Online tab

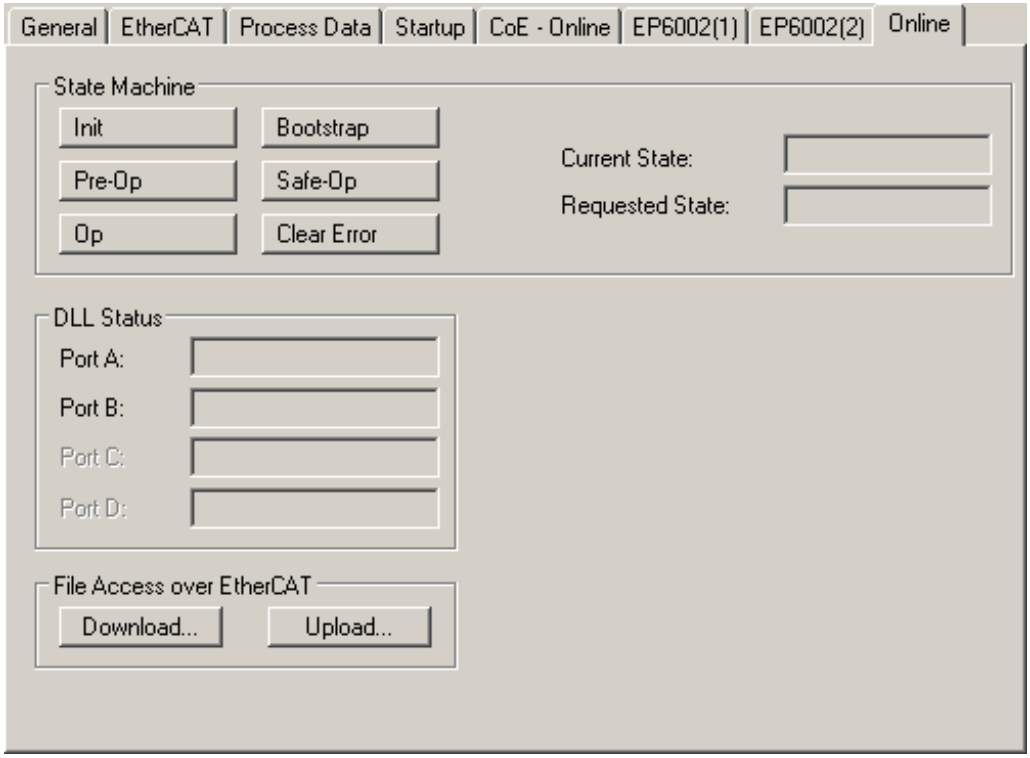


Fig. 63: Online tab

State Machine

Init	This button attempts to set the EtherCAT device to the *Init* state.
Pre-Op	This button attempts to set the EtherCAT device to the *pre-operational* state.
Op	This button attempts to set the EtherCAT device to the *operational* state.
Bootstrap	This button attempts to set the EtherCAT device to the *Bootstrap* state.
Safe-Op	This button attempts to set the EtherCAT device to the *safe-operational* state.
Clear Error	This button attempts to delete the fault display. If an EtherCAT slave fails during change of state it sets an error flag. Example: An EtherCAT slave is in PREOP state (pre-operational). The master now requests the SAFEOP state (safe-operational). If the slave fails during change of state it sets the error flag. The current state is now displayed as ERR PREOP. When the *Clear Error* button is pressed the error flag is cleared, and the current state is displayed as PREOP again.
Current State	Indicates the current state of the EtherCAT device.
Requested State	Indicates the state requested for the EtherCAT device.

DLL Status

Indicates the DLL status (data link layer status) of the individual ports of the EtherCAT slave. The DLL status can have four different states:

Status	Description
No Carrier / Open	No carrier signal is available at the port, but the port is open.
No Carrier / Closed	No carrier signal is available at the port, and the port is closed.
Carrier / Open	A carrier signal is available at the port, and the port is open.
Carrier / Closed	A carrier signal is available at the port, but the port is closed.

File Access over EtherCAT

Download	With this button a file can be written to the EtherCAT device.
Upload	With this button a file can be read from the EtherCAT device.

5.5 EP600x-0002 - Selection of the interface type

You can parameterize the serial interfaces under the object 0xF800:0 [▶ 92] on the CoE-Online tab in the TwinCAT System Manager.

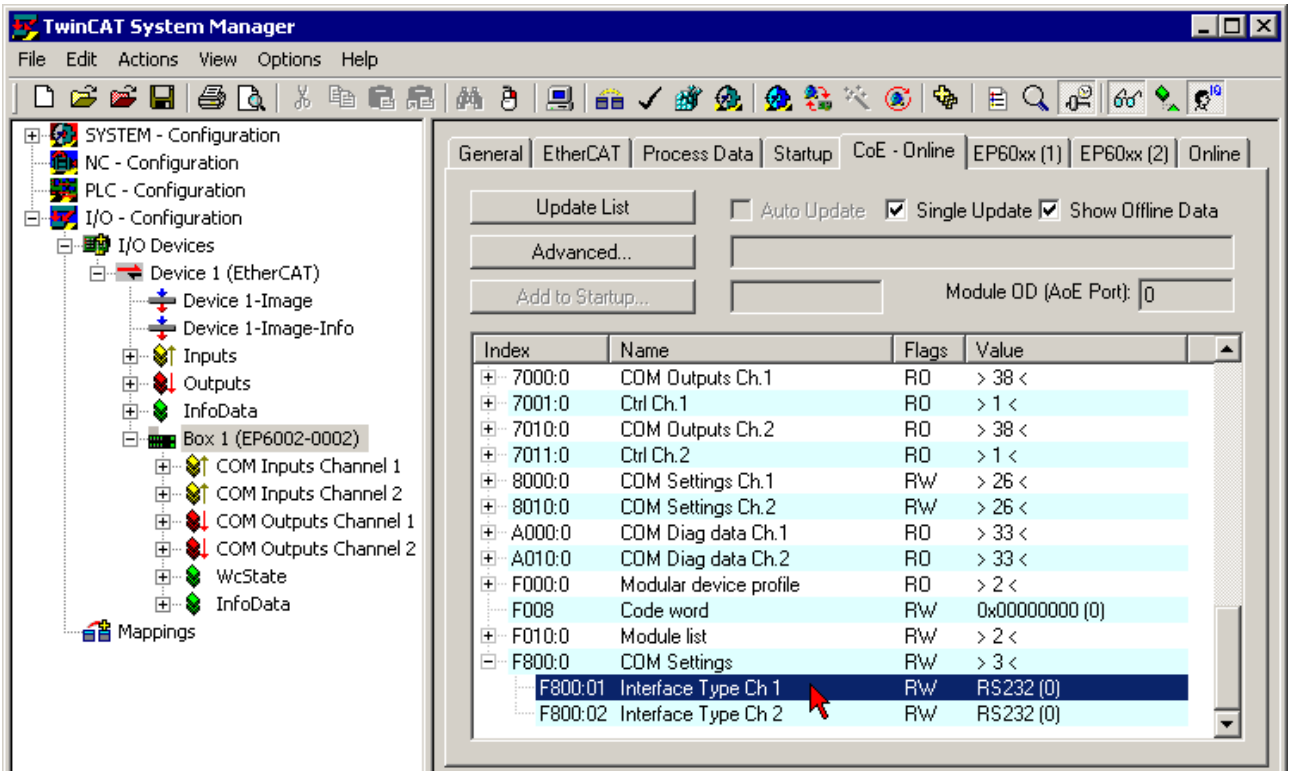


Fig. 64: CoE object 0xF800:0 COM Settings

Click on the objects 0xF800:01 [▶ 92] and 0xF800:02 [▶ 92] and select the interface type for both interfaces.

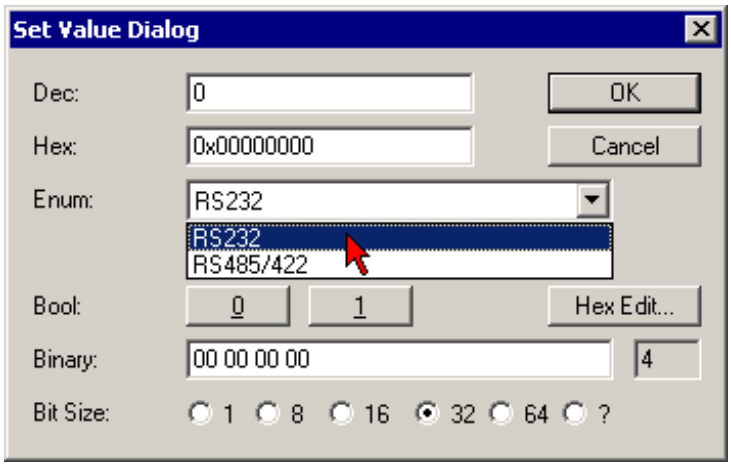


Fig. 65: Set Value Dialog Index 0xF800:01 Interface Type Ch. 1

EP6002-0002 - assignment of the connections

Depending on the interface type you must use the corresponding M12 connection.

Channel	selected interface type	use	Comment
Channel 1	RS485/RS422	M12 connection no. 1	M12 connection no. 2 has no function
	RS232	M12 connection no. 2	M12 connection no. 1 has no function
Channel 2	RS485/RS422	M12 connection no. 3	M12 connection no. 4 has no function
	RS232	M12 connection no. 4	M12 connection no. 3 has no function

EP6001-0002 - assignment of the connections

Depending on the interface type you must use the corresponding M12 connection.

Channel	selected interface type	use	Comment
Channel 1	RS232	M12 connection no. 1	M12 connection nos. 2 and 3 have no function
	RS485/422TX	M12 connection no. 2	M12 connection no. 1 has no function
	RS422RX	M12 connection no. 3	M12 connection no. 1 has no function

5.6 EP6002-0002 - sample program 1

Using the sample programs

This document contains sample applications of our products for certain areas of application. The application notes provided here are based on typical features of our products and only serve as examples. The notes provided with this documentation expressly make no reference to specific application cases. Therefore it is the customer's responsibility to check and decide whether the product is suitable for use in a certain application area. We accept no responsibility for the completeness and correctness of the source code contained in this document. We reserve the right to modify the content of this document at any time and accept no responsibility for errors and missing information.

Connection of a serial bar code scanner

https://infosys.beckhoff.com/content/1033/EP6001_EP6002/Resources/zip/2200499211.zip

In this example, a barcode reader will be connected to the EP6002-0002. Characters will be read by the reader until the ASCII character 0x0D (13_{dec}, CR) is received.

Data:

- Quick task for executing the serial communication: 1 ms cycle time
- Standard PLC task: 10 ms cycle time
- Bar code scanner on channel 1
- TwinCAT 2.11 required
- "TwinCAT PLC Serial Communication" supplement is required

A detailed description for the use of the serial communication library is stored in the Beckhoff Information System.

Beckhoff Information System -> TwinCAT -> TwinCAT PLC -> TwinCAT libraries for PC-based systems -> TwinCAT PLC Library: Serial communication

Starting the sample program

The application samples have been tested with a test configuration and are described accordingly.

Certain deviations when setting up actual applications are possible.

The following hardware and software were used for the test configuration:

- TwinCAT master PC with Windows XP Professional SP 3, TwinCAT version 2.11 (Build 1528) and INTEL PRO/100 VE Ethernet adapter
- Beckhoff EP6002-0002 EtherCAT Box
- Serial bar code scanner

Procedure for starting the program

- After clicking the Download button, save the zip file locally on your hard disk, and unzip the *.TSM (configuration) and the *.PRO (PLC program) files into a temporary working folder
- Run the *.TSM file and the *.PRO file; the TwinCAT System Manager and TwinCAT PLC will open

- Connect the hardware as suited and connect the Ethernet adapter of your PC to the EtherCAT coupler (further information on this can be found in the corresponding coupler manuals)
- Select the local Ethernet adapter (with real-time driver if applicable) under System Configuration, I/O Configuration, I/O Devices, Device (EtherCAT); then select the appropriate adapter on the "Adapter" tab, "Search..." and confirm (see the following two figures)

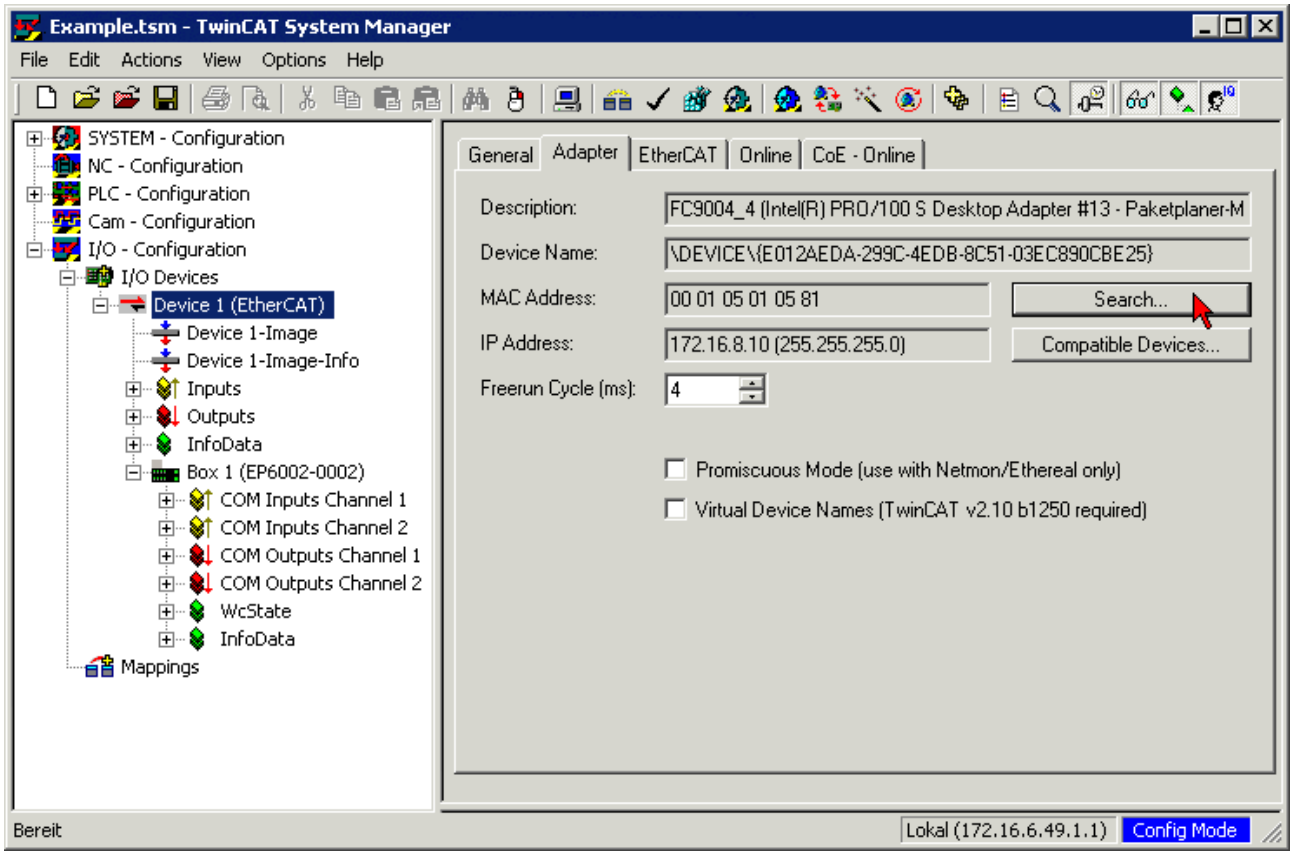


Fig. 66: Searching the Ethernet adapter

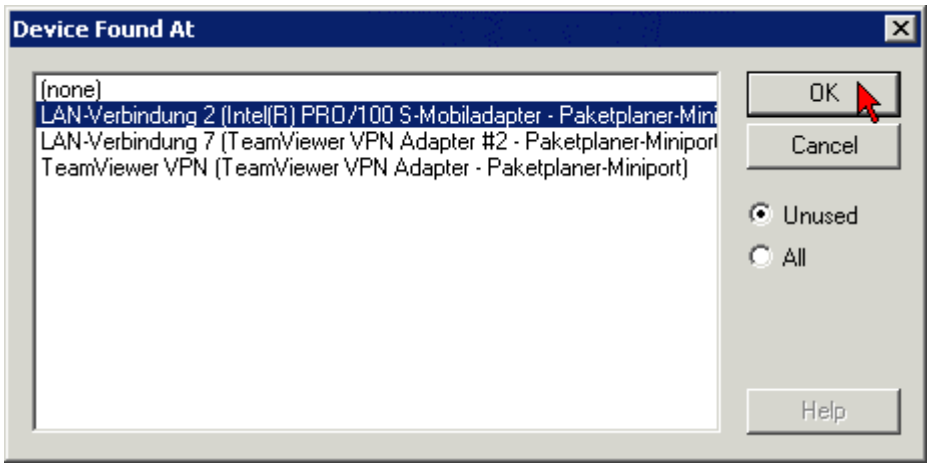


Fig. 67: Selection and confirmation of the Ethernet adapter

Activation of the configuration and confirmation (see the following two figures)

Fig. 68: Activation of the configuration

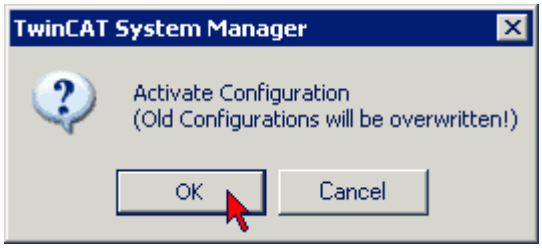


Fig. 69: Confirming the activation of the configuration

- Confirm new variable mapping, restart in RUN mode (see the following two figures)

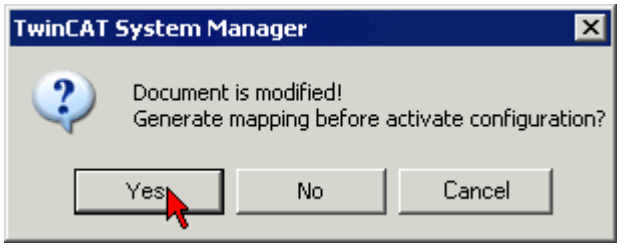


Fig. 70: Generating variable mapping

Fig. 71: Restarting TwinCAT in RUN mode

- In TwinCAT PLC under the "Project" menu, select "Rebuild all" to compile the project (see following figure)

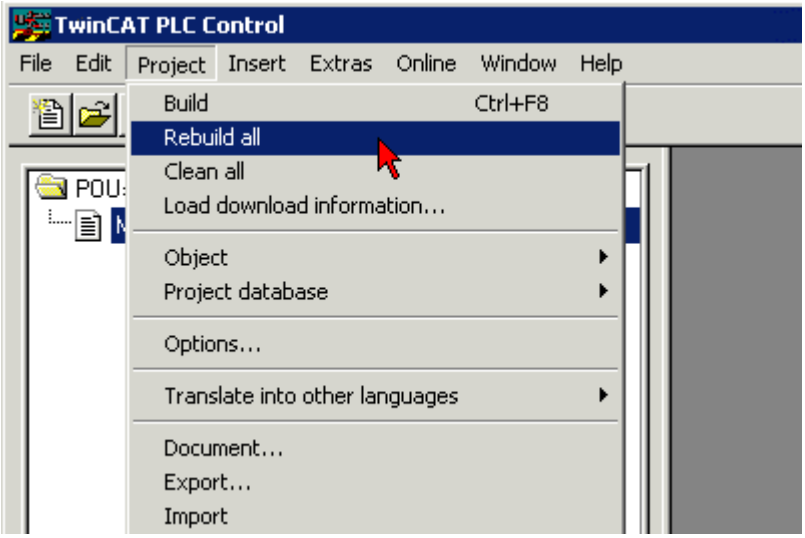


Fig. 72: Build project

- In TwinCAT PLC: log in with the "F11" button, confirm loading the program (see following figure), start the program with the "F5" key

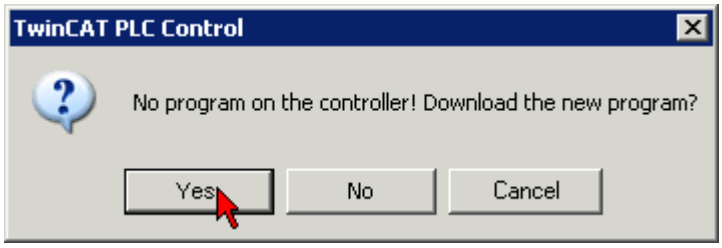


Fig. 73: Confirming program start

- After the character 0x13 has been received, the barcode is stored in "LastBarcode".

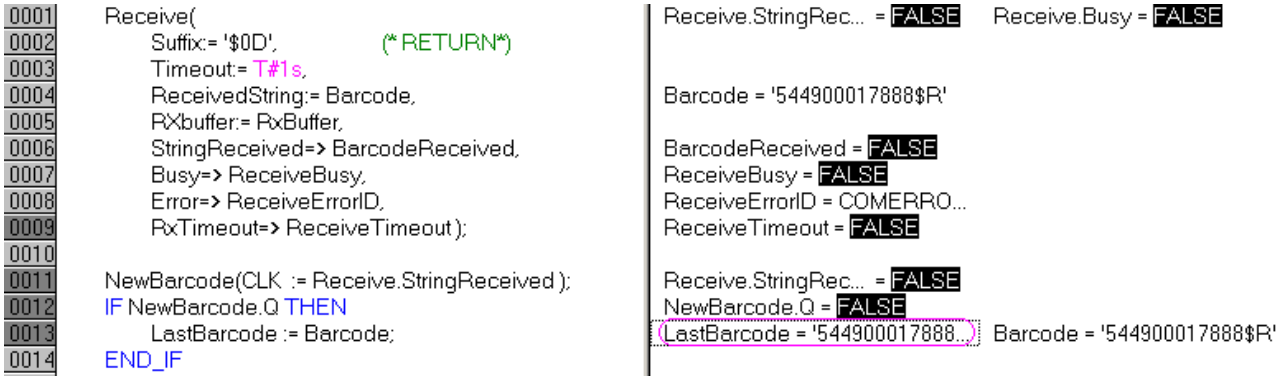


Fig. 74: Received barcode

5.7 EP6002-0002 - Sample program 2

Using the sample programs

This document contains sample applications of our products for certain areas of application. The application notes provided here are based on typical features of our products and only serve as examples. The notes contained in this document explicitly do not refer to specific applications. The customer is therefore responsible for assessing and deciding whether the product is suitable for a particular application. We accept no responsibility for the completeness and correctness of the source code contained in this document. We reserve the right to modify the content of this document at any time and accept no responsibility for errors and missing information.

Download sample program:

https://infosys.beckhoff.com/content/1033/EP6001_EP6002/Resources/zip/2213177355.zip

Reading and interpreting time telegrams

This example shows a way to process and interpret the most diverse serial time protocols in the PLC. To this end, IEC61131-PLC blocks will be presented that interpret the bitstream arriving at the PLC and, if necessary, extract the time/place information. This information can be used, for example, to synchronize controllers or record movements

In this example, it is assumed that the data is delivered via a 22-byte interface from an EP600x serial data exchange box.

Background information

Not only in maritime applications is time and place information transported via serial buses: In the majority of cases an existing source distributes its information to lower level listeners/listeners cyclically or after a trigger via RS232 / RS485, USB or Ethernet.

A very large number of telegram formats exists worldwide for the distribution of time and place information; these are also known as ‘sentences’. Such a telegram consists of n bytes of data and is characterized by:

- Start and end characters STX, ETX for telegram recognition, possibly more than one end character
- a defined and fixed length
- a defined structure
- checksum (not necessary)
- type designations in the sentence if necessary

The most diverse organizations and companies have developed open or proprietary formats for specific purposes of use. Therefore, two sample implementations will be presented in this example that can easily be adapted to other specific protocols. The telegram formats in the example are the Meinberg Standard and NMEA0183 v2.3 type RMC.

Sources of information

GPS or radio-controlled clock gateways are used as data transmitters in the serial sector. These devices receive the respective time signal (GPS via satellite or radio-controlled clock via long wave) and convert it to the serial, wire-bound transmission e.g. RS232 with 8N1. The gateway often contains a local clock source in order to be able to continue distributing reliable time information for a certain time in the event of a short-term failure of the reference signal (GPS, radio transmitter). In Central Europe, the German DCF77 transmitter can be used.

If necessary, further information from the reference signal can be used:

- GPS: location information (W/N and height), upcoming time adjustment DCF77: weather information, major incidents
- DCF77: weather information, major incidents

Synchronization of lower level time slaves

In general, lower level slaves should be adjusted to the time gateway, i.e. they should synchronize their time. The following are necessary for this:

- Offset : the absolute deviation of the slave’s own clock from the gateway reference time – this information can be transmitted, for example, acyclically and serially if it is known when the time information is to be considered valid. Gaining these offset-information is possible via the serial transport route from this example.
- the frequency ratio: a high-precision cyclic signal from the gateway to the slaves allows drift processes to be compensated and might state the time when the above absolute offset is valid. One example of such a signal is the widespread PPS (pulses per second).

If the serial telegram from the gateway is placed cyclically to the bus, the time of the first bit can often be interpreted as a PPS signal. In the function block in this example, this information would be lost; only the absolute time information is evaluated.

Time formats

Time telegrams conforming to the NMEA0183 standard are widespread. Please note:

- there are currently (as of 2009) 8 versions of NMEA0183 1.5 to 4.00 within the NMEA0183 standard – the structure of telegrams may have changed between the versions.
- 70 different formats are defined in NMEA0183 v2.30; device manufacturers can create their own formats in addition.
- The telegram is called a sentence.
- A TalkerID (2 characters) and a TypeID (3 characters) at the start define the type of sentence used.
- A checksum is calculated for the telegram.
- Information can be found online at www.nmea.org or elsewhere.

Furthermore, many proprietary formats exist, such as Meinberg Standard, Siemens SINEC H1 and SAT 1703, or military formats, such as the IRIG codes (USA).

Using the sample program

The PLC project contains 2 function blocks (FBs), which must be linked exclusively with an EL/KL600x. Then collect the received bytes from the terminal and interpret the contents as far as possible. The FBs cover:

- Meinberg Standard
- NMEA0183 v2.3 type RMC

For other time formats, you can create your own interpretation FBs on the basis of a known telegram structure; contact your device manufacturer regarding this.

Hardware used in the example: EK1100 and EL6001 (also EL600x, KL600x)
The FB to be tested must be linked with its process data to the terminal in the System Manager (22-byte process image and control/status word).

The NMEA block is linked in the example. The baud rate of the terminal must be set to match your transmitter.

References:

- www.beuth.de.de, IEC61162: based on NMEA2000
- www.gpsinformation.net/, private, via NMEA, many formats: http://www.gpsinformation.org/dale/nmea.htm
  http://www.nmea.de/nmea0183datensaetze.html
- https://www.meinberg.de/german/info/irig.htm, IRIG codes

Documents about this

el6001_nmea_demo.zip (Resources/zip/2200502155.zip)

5.8 EP6002-0002 - Object overview

EtherCAT XML Device Description

The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

Index (hex)		Name	Flags	Default value
1000 [▶ 93]		Device type	RO	0x02581389 (39326601_{dec})
1008 [▶ 93]		Device name	RO	EP6002-0002
1009 [▶ 93]		Hardware version	RO	00
100A [▶ 93]		Software version	RO	00
1011:0 [▶ 90]	**Subindex**	Restore default parameters	RO	0x01 (1_{dec})
	1011:01	SubIndex 001	RW	0x00000000 (0_{dec})
1018:0 [▶ 93]	**Subindex**	Identity	RO	0x04 (4_{dec})
	1018:01	Vendor ID	RO	0x00000002 (2_{dec})
	1018:02	Product code	RO	0x17724052 (393363538_{dec})
	1018:03	Revision	RO	0x00100002 (1048578_{dec})
	1018:04	Serial number	RO	0x00000000 (0_{dec})
10F0:0 [▶ 93]	**Subindex**	Backup parameter handling	RO	0x01 (1_{dec})
	10F0:01	Checksum	RO	0x00000000 (0_{dec})
1400:0 [▶ 93]	**Subindex**	COM RxPDO-Par Outputs Ch.1	RO	0x06 (6_{dec})
	1400:06	Exclude RxPDOs	RO	04 16
1401:0 [▶ 94]	**Subindex**	COM RxPDO-Par Outputs Ch.2	RO	0x06 (6_{dec})
	1401:06	Exclude RxPDOs	RO	05 16
1404:0 [▶ 94]	**Subindex**	COM RxPDO-Par Outputs Ch.1	RO	0x06 (6_{dec})
	1404:06	Exclude RxPDOs	RO	00 16
1405:0 [▶ 94]	**Subindex**	COM RxPDO-Par Outputs Ch.2	RO	0x06 (6_{dec})
	1405:06	Exclude RxPDOs	RO	01 16

Index (hex)		Name	Flags	Default value
1600:0 [▶ 95]	**Subindex**	COM RxPDO-Map Outputs Ch.1	RO	0x1C (28_{dec})
	1600:01	SubIndex 001	RO	0x7000:01, 1
	1600:02	SubIndex 002	RO	0x7000:02, 1
	1600:03	SubIndex 003	RO	0x7000:03, 1
	1600:04	SubIndex 004	RO	0x7000:04, 1
	1600:05	SubIndex 005	RO	0x0000:00, 4
	1600:06	SubIndex 006	RO	0x7000:09, 8
	1600:07	SubIndex 007	RO	0x7000:11, 8
	1600:08	SubIndex 008	RO	0x7000:12, 8
	1600:09	SubIndex 009	RO	0x7000:13, 8
	1600:0A	SubIndex 010	RO	0x7000:14, 8
	1600:0B	SubIndex 011	RO	0x7000:15, 8
	1600:0C	SubIndex 012	RO	0x7000:16, 8
	1600:0D	SubIndex 013	RO	0x7000:17, 8
	1600:0E	SubIndex 014	RO	0x7000:18, 8
	1600:0F	SubIndex 015	RO	0x7000:19, 8
	1600:10	SubIndex 016	RO	0x7000:1A, 8
	1600:11	SubIndex 017	RO	0x7000:1B, 8
	1600:12	SubIndex 018	RO	0x7000:1C, 8
	1600:13	SubIndex 019	RO	0x7000:1D, 8
	1600:14	SubIndex 020	RO	0x7000:1E, 8
	1600:15	SubIndex 021	RO	0x7000:1F, 8
	1600:16	SubIndex 022	RO	0x7000:20, 8
	1600:17	SubIndex 023	RO	0x7000:21, 8
	1600:18	SubIndex 024	RO	0x7000:22, 8
	1600:19	SubIndex 025	RO	0x7000:23, 8
	1600:1A	SubIndex 026	RO	0x7000:24, 8
	1600:1B	SubIndex 027	RO	0x7000:25, 8
	1600:1C	SubIndex 028	RO	0x7000:26, 8
1601:0 [▶ 96]	**Subindex**	COM RxPDO-Map Outputs Ch.2	RO	0x1C (28_{dec})
	1601:01	SubIndex 001	RO	0x7010:01, 1
	1601:02	SubIndex 002	RO	0x7010:02, 1
	1601:03	SubIndex 003	RO	0x7010:03, 1
	1601:04	SubIndex 004	RO	0x7010:04, 1
	1601:05	SubIndex 005	RO	0x0000:00, 4
	1601:06	SubIndex 006	RO	0x7010:09, 8
	1601:07	SubIndex 007	RO	0x7010:11, 8
	1601:08	SubIndex 008	RO	0x7010:12, 8
	1601:09	SubIndex 009	RO	0x7010:13, 8
	1601:0A	SubIndex 010	RO	0x7010:14, 8
	1601:0B	SubIndex 011	RO	0x7010:15, 8
	1601:0C	SubIndex 012	RO	0x7010:16, 8
	1601:0D	SubIndex 013	RO	0x7010:17, 8
	1601:0E	SubIndex 014	RO	0x7010:18, 8
	1601:0F	SubIndex 015	RO	0x7010:19, 8
	1601:10	SubIndex 016	RO	0x7010:1A, 8
	1601:11	SubIndex 017	RO	0x7010:1B, 8
	1601:12	SubIndex 018	RO	0x7010:1C, 8
	1601:13	SubIndex 019	RO	0x7010:1D, 8
	1601:14	SubIndex 020	RO	0x7010:1E, 8
	1601:15	SubIndex 021	RO	0x7010:1F, 8
	1601:16	SubIndex 022	RO	0x7010:20, 8
	1601:17	SubIndex 023	RO	0x7010:21, 8
	1601:18	SubIndex 024	RO	0x7010:22, 8
	1601:19	SubIndex 025	RO	0x7010:23, 8
	1601:1A	SubIndex 026	RO	0x7010:24, 8
	1601:1B	SubIndex 027	RO	0x7010:25, 8
	1601:1C	SubIndex 028	RO	0x7010:26, 8

Index (hex)		Name	Flags	Default value
1604:0 [▶ 97]	**Subindex**	COM RxPDO-Map Outputs Ch.1	RO	0x17 (23_{dec})
	1604:01	SubIndex 001	RO	0x7001:01, 16
	1604:02	SubIndex 002	RO	0x7000:11, 8
	1604:03	SubIndex 003	RO	0x7000:12, 8
	1604:04	SubIndex 004	RO	0x7000:13, 8
	1604:05	SubIndex 005	RO	0x7000:14, 8
	1604:06	SubIndex 006	RO	0x7000:15, 8
	1604:07	SubIndex 007	RO	0x7000:16, 8
	1604:08	SubIndex 008	RO	0x7000:17, 8
	1604:09	SubIndex 009	RO	0x7000:18, 8
	1604:0A	SubIndex 010	RO	0x7000:19, 8
	1604:0B	SubIndex 011	RO	0x7000:1A, 8
	1604:0C	SubIndex 012	RO	0x7000:1B, 8
	1604:0D	SubIndex 013	RO	0x7000:1C, 8
	1604:0E	SubIndex 014	RO	0x7000:1D, 8
	1604:0F	SubIndex 015	RO	0x7000:1E, 8
	1604:10	SubIndex 016	RO	0x7000:1F, 8
	1604:11	SubIndex 017	RO	0x7000:20, 8
	1604:12	SubIndex 018	RO	0x7000:21, 8
	1604:13	SubIndex 019	RO	0x7000:22, 8
	1604:14	SubIndex 020	RO	0x7000:23, 8
	1604:15	SubIndex 021	RO	0x7000:24, 8
	1604:16	SubIndex 022	RO	0x7000:25, 8
	1604:17	SubIndex 023	RO	0x7000:26, 8
1605:0 [▶ 98]	**Subindex**	COM RxPDO-Map Outputs Ch.2	RO	0x17 (23_{dec})
	1605:01	SubIndex 001	RO	0x7011:01, 16
	1605:02	SubIndex 002	RO	0x7010:11, 8
	1605:03	SubIndex 003	RO	0x7010:12, 8
	1605:04	SubIndex 004	RO	0x7010:13, 8
	1605:05	SubIndex 005	RO	0x7010:14, 8
	1605:06	SubIndex 006	RO	0x7010:15, 8
	1605:07	SubIndex 007	RO	0x7010:16, 8
	1605:08	SubIndex 008	RO	0x7010:17, 8
	1605:09	SubIndex 009	RO	0x7010:18, 8
	1605:0A	SubIndex 010	RO	0x7010:19, 8
	1605:0B	SubIndex 011	RO	0x7010:1A, 8
	1605:0C	SubIndex 012	RO	0x7010:1B, 8
	1605:0D	SubIndex 013	RO	0x7010:1C, 8
	1605:0E	SubIndex 014	RO	0x7010:1D, 8
	1605:0F	SubIndex 015	RO	0x7010:1E, 8
	1605:10	SubIndex 016	RO	0x7010:1F, 8
	1605:11	SubIndex 017	RO	0x7010:20, 8
	1605:12	SubIndex 018	RO	0x7010:21, 8
	1605:13	SubIndex 019	RO	0x7010:22, 8
	1605:14	SubIndex 020	RO	0x7010:23, 8
	1605:15	SubIndex 021	RO	0x7010:24, 8
	1605:16	SubIndex 022	RO	0x7010:25, 8
	1605:17	SubIndex 023	RO	0x7010:26, 8
1800:0 [▶ 98]	**Subindex**	COM TxPDO-Par Inputs Ch.1	RO	0x06 (6_{dec})
	1800:06	Exclude TxPDOs	RO	04 1A
1801:0 [▶ 99]	**Subindex**	COM TxPDO-Par Inputs Ch.2	RO	0x06 (6_{dec})
	1801:06	Exclude TxPDOs	RO	05 1A
1804:0 [▶ 99]	**Subindex**	COM TxPDO-Par Inputs Ch.1	RO	0x06 (6_{dec})
	1804:06	Exclude TxPDOs	RO	00 1A
1805:0 [▶ 99]	**Subindex**	COM TxPDO-Par Inputs Ch.2	RO	0x06 (6_{dec})
	1805:06	Exclude TxPDOs	RO	01 1A
1A00:0 [▶ 100]	**Subindex**	COM TxPDO-Map Inputs Ch.1	RO	0x1F (31_{dec})
	1A00:01	SubIndex 001	RO	0x6000:01, 1

Index (hex)		Name	Flags	Default value
	1A00:02	SubIndex 002	RO	0x6000:02, 1
	1A00:03	SubIndex 003	RO	0x6000:03, 1
	1A00:04	SubIndex 004	RO	0x6000:04, 1
	1A00:05	SubIndex 005	RO	0x6000:05, 1
	1A00:06	SubIndex 006	RO	0x6000:06, 1
	1A00:07	SubIndex 007	RO	0x6000:07, 1
	1A00:08	SubIndex 008	RO	0x0000:00, 1
	1A00:09	SubIndex 009	RO	0x6000:09, 8
	1A00:0A	SubIndex 010	RO	0x6000:11, 8
	1A00:0B	SubIndex 011	RO	0x6000:12, 8
	1A00:0C	SubIndex 012	RO	0x6000:13, 8
	1A00:0D	SubIndex 013	RO	0x6000:14, 8
	1A00:0E	SubIndex 014	RO	0x6000:15, 8
	1A00:0F	SubIndex 015	RO	0x6000:16, 8
	1A00:10	SubIndex 016	RO	0x6000:17, 8
	1A00:11	SubIndex 017	RO	0x6000:18, 8
	1A00:12	SubIndex 018	RO	0x6000:19, 8
	1A00:13	SubIndex 019	RO	0x6000:1A, 8
	1A00:14	SubIndex 020	RO	0x6000:1B, 8
	1A00:15	SubIndex 021	RO	0x6000:1C, 8
	1A00:16	SubIndex 022	RO	0x6000:1D, 8
	1A00:17	SubIndex 023	RO	0x6000:1E, 8
	1A00:18	SubIndex 024	RO	0x6000:1F, 8
	1A00:19	SubIndex 025	RO	0x6000:20, 8
	1A00:1A	SubIndex 026	RO	0x6000:21, 8
	1A00:1B	SubIndex 027	RO	0x6000:22, 8
	1A00:1C	SubIndex 028	RO	0x6000:23, 8
	1A00:1D	SubIndex 029	RO	0x6000:24, 8
	1A00:1E	SubIndex 030	RO	0x6000:25, 8
	1A00:1F	SubIndex 031	RO	0x6000:26, 8
1A01:0 [▶ 101]	**Subindex**	COM TxPDO-Map Inputs Ch.2	RO	0x1F (31_{dec})
	1A01:01	SubIndex 001	RO	0x6010:01, 1
	1A01:02	SubIndex 002	RO	0x6010:02, 1
	1A01:03	SubIndex 003	RO	0x6010:03, 1
	1A01:04	SubIndex 004	RO	0x6010:04, 1
	1A01:05	SubIndex 005	RO	0x6010:05, 1
	1A01:06	SubIndex 006	RO	0x6010:06, 1
	1A01:07	SubIndex 007	RO	0x6010:07, 1
	1A01:08	SubIndex 008	RO	0x0000:00, 1
	1A01:09	SubIndex 009	RO	0x6010:09, 8
	1A01:0A	SubIndex 010	RO	0x6010:11, 8
	1A01:0B	SubIndex 011	RO	0x6010:12, 8
	1A01:0C	SubIndex 012	RO	0x6010:13, 8
	1A01:0D	SubIndex 013	RO	0x6010:14, 8
	1A01:0E	SubIndex 014	RO	0x6010:15, 8
	1A01:0F	SubIndex 015	RO	0x6010:16, 8
	1A01:10	SubIndex 016	RO	0x6010:17, 8
	1A01:11	SubIndex 017	RO	0x6010:18, 8
	1A01:12	SubIndex 018	RO	0x6010:19, 8
	1A01:13	SubIndex 019	RO	0x6010:1A, 8
	1A01:14	SubIndex 020	RO	0x6010:1B, 8
	1A01:15	SubIndex 021	RO	0x6010:1C, 8
	1A01:16	SubIndex 022	RO	0x6010:1D, 8
	1A01:17	SubIndex 023	RO	0x6010:1E, 8
	1A01:18	SubIndex 024	RO	0x6010:1F, 8
	1A01:19	SubIndex 025	RO	0x6010:20, 8
	1A01:1A	SubIndex 026	RO	0x6010:21, 8
	1A01:1B	SubIndex 027	RO	0x6010:22, 8

Index (hex)		Name	Flags	Default value
	1A01:1C	SubIndex 028	RO	0x6010:23, 8
	1A01:1D	SubIndex 029	RO	0x6010:24, 8
	1A01:1E	SubIndex 030	RO	0x6010:25, 8
	1A01:1F	SubIndex 031	RO	0x6010:26, 8
1A04:0 [▶ 102]	**Subindex**	COM TxPDO-Map Inputs Ch.1	RO	0x17 (23_{dec})
	1A04:01	SubIndex 001	RO	0x6001:01, 16
	1A04:02	SubIndex 002	RO	0x6000:11, 8
	1A04:03	SubIndex 003	RO	0x6000:12, 8
	1A04:04	SubIndex 004	RO	0x6000:13, 8
	1A04:05	SubIndex 005	RO	0x6000:14, 8
	1A04:06	SubIndex 006	RO	0x6000:15, 8
	1A04:07	SubIndex 007	RO	0x6000:16, 8
	1A04:08	SubIndex 008	RO	0x6000:17, 8
	1A04:09	SubIndex 009	RO	0x6000:18, 8
	1A04:0A	SubIndex 010	RO	0x6000:19, 8
	1A04:0B	SubIndex 011	RO	0x6000:1A, 8
	1A04:0C	SubIndex 012	RO	0x6000:1B, 8
	1A04:0D	SubIndex 013	RO	0x6000:1C, 8
	1A04:0E	SubIndex 014	RO	0x6000:1D, 8
	1A04:0F	SubIndex 015	RO	0x6000:1E, 8
	1A04:10	SubIndex 016	RO	0x6000:1F, 8
	1A04:11	SubIndex 017	RO	0x6000:20, 8
	1A04:12	SubIndex 018	RO	0x6000:21, 8
	1A04:13	SubIndex 019	RO	0x6000:22, 8
	1A04:14	SubIndex 020	RO	0x6000:23, 8
	1A04:15	SubIndex 021	RO	0x6000:24, 8
	1A04:16	SubIndex 022	RO	0x6000:25, 8
	1A04:17	SubIndex 023	RO	0x6000:26, 8
1A05:0 [▶ 103]	**Subindex**	COM TxPDO-Map Inputs Ch.2	RO	0x17 (23_{dec})
	1A05:01	SubIndex 001	RO	0x6011:01, 16
	1A05:02	SubIndex 002	RO	0x6010:11, 8
	1A05:03	SubIndex 003	RO	0x6010:12, 8
	1A05:04	SubIndex 004	RO	0x6010:13, 8
	1A05:05	SubIndex 005	RO	0x6010:14, 8
	1A05:06	SubIndex 006	RO	0x6010:15, 8
	1A05:07	SubIndex 007	RO	0x6010:16, 8
	1A05:08	SubIndex 008	RO	0x6010:17, 8
	1A05:09	SubIndex 009	RO	0x6010:18, 8
	1A05:0A	SubIndex 010	RO	0x6010:19, 8
	1A05:0B	SubIndex 011	RO	0x6010:1A, 8
	1A05:0C	SubIndex 012	RO	0x6010:1B, 8
	1A05:0D	SubIndex 013	RO	0x6010:1C, 8
	1A05:0E	SubIndex 014	RO	0x6010:1D, 8
	1A05:0F	SubIndex 015	RO	0x6010:1E, 8
	1A05:10	SubIndex 016	RO	0x6010:1F, 8
	1A05:11	SubIndex 017	RO	0x6010:20, 8
	1A05:12	SubIndex 018	RO	0x6010:21, 8
	1A05:13	SubIndex 019	RO	0x6010:22, 8
	1A05:14	SubIndex 020	RO	0x6010:23, 8
	1A05:15	SubIndex 021	RO	0x6010:24, 8
	1A05:16	SubIndex 022	RO	0x6010:25, 8
	1A05:17	SubIndex 023	RO	0x6010:26, 8
1C00:0 [▶ 103]	**Subindex**	Sync manager type	RO	0x04 (4_{dec})
	1C00:01	SubIndex 001	RO	0x01 (1_{dec})
	1C00:02	SubIndex 002	RO	0x02 (2_{dec})
	1C00:03	SubIndex 003	RO	0x03 (3_{dec})
	1C00:04	SubIndex 004	RO	0x04 (4_{dec})
1C12:0 [▶ 104]	**Subindex**	RxPDO assign	RW	0x02 (2_{dec})

Index (hex)		Name	Flags	Default value
	1C12:01	SubIndex 001	RW	0x1604 (5636_{dec})
	1C12:02	SubIndex 002	RW	0x1605 (5637_{dec})
1C13:0 [▶ 104]	**Subindex**	TxPDO assign	RW	0x02 (2_{dec})
	1C13:01	SubIndex 001	RW	0x1A04 (6660_{dec})
	1C13:02	SubIndex 002	RW	0x1A05 (6661_{dec})
1C32 [▶ 105]:0	**Subindex**	SM output parameter	RO	0x20 (32_{dec})
	1C32:01	Sync mode	RW	0x0000 (0_{dec})
	1C32:02	Cycle time	RW	0x000F4240 (1000000_{dec})
	1C32:03	Shift time	RO	0x00000384 (900_{dec})
	1C32:04	Sync modes supported	RO	0xC007 (49159_{dec})
	1C32:05	Minimum cycle time	RO	0x00002710 (10000_{dec})
	1C32:06	Calc and copy time	RO	0x00000000 (0_{dec})
	1C32:07	Minimum delay time	RO	0x00000384 (900_{dec})
	1C32:08	Command	RW	0x0000 (0_{dec})
	1C32:09	Maximum Delay time	RO	0x00000384 (900_{dec})
	1C32:0B	SM event missed counter	RO	0x0000 (0_{dec})
	1C32:0C	Cycle exceeded counter	RO	0x0000 (0_{dec})
	1C32:0D	Shift too short counter	RO	0x0000 (0_{dec})
	1C32:20	Sync error	RO	0x00 (0_{dec})
1C33:0 [▶ 106]	**Subindex**	SM input parameter	RO	0x20 (32_{dec})
	1C33:01	Sync mode	RW	0x0000 (0_{dec})
	1C33:02	Cycle time	RW	0x000F4240 (1000000_{dec})
	1C33:03	Shift time	RO	0x00000384 (900_{dec})
	1C33:04	Sync modes supported	RO	0xC007 (49159_{dec})
	1C33:05	Minimum cycle time	RO	0x00002710 (10000_{dec})
	1C33:06	Calc and copy time	RO	0x00000000 (0_{dec})
	1C33:07	Minimum delay time	RO	0x00000384 (900_{dec})
	1C33:08	Command	RW	0x0000 (0_{dec})
	1C33:09	Maximum Delay time	RO	0x00000384 (900_{dec})
	1C33:0B	SM event missed counter	RO	0x0000 (0_{dec})
	1C33:0C	Cycle exceeded counter	RO	0x0000 (0_{dec})
	1C33:0D	Shift too short counter	RO	0x0000 (0_{dec})
	1C33:20	Sync error	RO	0x00 (0_{dec})
6000:0 [▶ 107]	**Subindex**	COM Inputs Ch.1	RO	0x26 (38_{dec})
	6000:01	Transmit accepted	RO	0x00 (0_{dec})
	6000:02	Receive request	RO	0x00 (0_{dec})
	6000:03	Init Accepted	RO	0x00 (0_{dec})
	6000:04	Buffer full	RO	0x00 (0_{dec})
	6000:05	Parity error	RO	0x00 (0_{dec})
	6000:06	Framing error	RO	0x00 (0_{dec})
	6000:07	Overrun error	RO	0x00 (0_{dec})
	6000:09	Input length	RO	0x00 (0_{dec})
	6000:11	Data In 0	RO	0x00 (0_{dec})
	6000:12	Data In 1	RO	0x00 (0_{dec})
	6000:13	Data In 2	RO	0x00 (0_{dec})
	6000:14	Data In 3	RO	0x00 (0_{dec})
	6000:15	Data In 4	RO	0x00 (0_{dec})
	6000:16	Data In 5	RO	0x00 (0_{dec})
	6000:17	Data In 6	RO	0x00 (0_{dec})
	6000:18	Data In 7	RO	0x00 (0_{dec})
	6000:19	Data In 8	RO	0x00 (0_{dec})
	6000:1A	Data In 9	RO	0x00 (0_{dec})
	6000:1B	Data In 10	RO	0x00 (0_{dec})
	6000:1C	Data In 11	RO	0x00 (0_{dec})
	6000:1D	Data In 12	RO	0x00 (0_{dec})
	6000:1E	Data In 13	RO	0x00 (0_{dec})
	6000:1F	Data In 14	RO	0x00 (0_{dec})
	6000:20	Data In 15	RO	0x00 (0_{dec})

Index (hex)		Name	Flags	Default value
	6000:21	Data In 16	RO	0x00 (0_{dec})
	6000:22	Data In 17	RO	0x00 (0_{dec})
	6000:23	Data In 18	RO	0x00 (0_{dec})
	6000:24	Data In 19	RO	0x00 (0_{dec})
	6000:25	Data In 20	RO	0x00 (0_{dec})
	6000:26	Data In 21	RO	0x00 (0_{dec})
6001:0 [▶ 107]	**Subindex**	Status Ch.1	RO	0x01 (1_{dec})
	6001:01	Status	RO	0x0000 (0_{dec})
6010:0 [▶ 108]	**Subindex**	COM Inputs Ch.2	RO	0x26 (38_{dec})
	6010:01	Transmit accepted	RO	0x00 (0_{dec})
	6010:02	Receive request	RO	0x00 (0_{dec})
	6010:03	Init Accepted	RO	0x00 (0_{dec})
	6010:04	Buffer full	RO	0x00 (0_{dec})
	6010:05	Parity error	RO	0x00 (0_{dec})
	6010:06	Framing error	RO	0x00 (0_{dec})
	6010:07	Overrun error	RO	0x00 (0_{dec})
	6010:09	Input length	RO	0x00 (0_{dec})
	6010:11	Data In 0	RO	0x00 (0_{dec})
	6010:12	Data In 1	RO	0x00 (0_{dec})
	6010:13	Data In 2	RO	0x00 (0_{dec})
	6010:14	Data In 3	RO	0x00 (0_{dec})
	6010:15	Data In 4	RO	0x00 (0_{dec})
	6010:16	Data In 5	RO	0x00 (0_{dec})
	6010:17	Data In 6	RO	0x00 (0_{dec})
	6010:18	Data In 7	RO	0x00 (0_{dec})
	6010:19	Data In 8	RO	0x00 (0_{dec})
	6010:1A	Data In 9	RO	0x00 (0_{dec})
	6010:1B	Data In 10	RO	0x00 (0_{dec})
	6010:1C	Data In 11	RO	0x00 (0_{dec})
	6010:1D	Data In 12	RO	0x00 (0_{dec})
	6010:1E	Data In 13	RO	0x00 (0_{dec})
	6010:1F	Data In 14	RO	0x00 (0_{dec})
	6010:20	Data In 15	RO	0x00 (0_{dec})
	6010:21	Data In 16	RO	0x00 (0_{dec})
	6010:22	Data In 17	RO	0x00 (0_{dec})
	6010:23	Data In 18	RO	0x00 (0_{dec})
	6010:24	Data In 19	RO	0x00 (0_{dec})
	6010:25	Data In 20	RO	0x00 (0_{dec})
	6010:26	Data In 21	RO	0x00 (0_{dec})
6011:0 [▶ 108]	**Subindex**	Status Ch.2	RO	0x01 (1_{dec})
	6011:01	Status	RO	0x0000 (0_{dec})
7000:0 [▶ 109]	**Subindex**	COM Outputs Ch.1	RO	0x26 (38_{dec})
	7000:01	Transmit request	RO	0x00 (0_{dec})
	7000:02	Receive accepted	RO	0x00 (0_{dec})
	7000:03	Init request	RO	0x00 (0_{dec})
	7000:04	Send continuous	RO	0x00 (0_{dec})
	7000:09	Output length	RO	0x00 (0_{dec})
	7000:11	Data Out 0	RO	0x00 (0_{dec})
	7000:12	Data Out 1	RO	0x00 (0_{dec})
	7000:13	Data Out 2	RO	0x00 (0_{dec})
	7000:14	Data Out 3	RO	0x00 (0_{dec})
	7000:15	Data Out 4	RO	0x00 (0_{dec})
	7000:16	Data Out 5	RO	0x00 (0_{dec})
	7000:17	Data Out 6	RO	0x00 (0_{dec})
	7000:18	Data Out 7	RO	0x00 (0_{dec})
	7000:19	Data Out 8	RO	0x00 (0_{dec})
	7000:1A	Data Out 9	RO	0x00 (0_{dec})
	7000:1B	Data Out 10	RO	0x00 (0_{dec})

Index (hex)		Name	Flags	Default value
	7000:1C	Data Out 11	RO	0x00 (0_{dec})
	7000:1D	Data Out 12	RO	0x00 (0_{dec})
	7000:1E	Data Out 13	RO	0x00 (0_{dec})
	7000:1F	Data Out 14	RO	0x00 (0_{dec})
	7000:20	Data Out 15	RO	0x00 (0_{dec})
	7000:21	Data Out 16	RO	0x00 (0_{dec})
	7000:22	Data Out 17	RO	0x00 (0_{dec})
	7000:23	Data Out 18	RO	0x00 (0_{dec})
	7000:24	Data Out 19	RO	0x00 (0_{dec})
	7000:25	Data Out 20	RO	0x00 (0_{dec})
	7000:26	Data Out 21	RO	0x00 (0_{dec})
7001:0 [▶ 109]	**Subindex**	Ctrl Ch.1	RO	0x01 (1_{dec})
	7001:01	Ctrl	RO	0x0000 (0_{dec})
7010:0 [▶ 110]	**Subindex**	COM Outputs Ch.2	RO	0x26 (38_{dec})
	7010:01	Transmit request	RO	0x00 (0_{dec})
	7010:02	Receive accepted	RO	0x00 (0_{dec})
	7010:03	Init request	RO	0x00 (0_{dec})
	7010:04	Send continuous	RO	0x00 (0_{dec})
	7010:09	Output length	RO	0x00 (0_{dec})
	7010:11	Data Out 0	RO	0x00 (0_{dec})
	7010:12	Data Out 1	RO	0x00 (0_{dec})
	7010:13	Data Out 2	RO	0x00 (0_{dec})
	7010:14	Data Out 3	RO	0x00 (0_{dec})
	7010:15	Data Out 4	RO	0x00 (0_{dec})
	7010:16	Data Out 5	RO	0x00 (0_{dec})
	7010:17	Data Out 6	RO	0x00 (0_{dec})
	7010:18	Data Out 7	RO	0x00 (0_{dec})
	7010:19	Data Out 8	RO	0x00 (0_{dec})
	7010:1A	Data Out 9	RO	0x00 (0_{dec})
	7010:1B	Data Out 10	RO	0x00 (0_{dec})
	7010:1C	Data Out 11	RO	0x00 (0_{dec})
	7010:1D	Data Out 12	RO	0x00 (0_{dec})
	7010:1E	Data Out 13	RO	0x00 (0_{dec})
	7010:1F	Data Out 14	RO	0x00 (0_{dec})
	7010:20	Data Out 15	RO	0x00 (0_{dec})
	7010:21	Data Out 16	RO	0x00 (0_{dec})
	7010:22	Data Out 17	RO	0x00 (0_{dec})
	7010:23	Data Out 18	RO	0x00 (0_{dec})
	7010:24	Data Out 19	RO	0x00 (0_{dec})
	7010:25	Data Out 20	RO	0x00 (0_{dec})
	7010:26	Data Out 21	RO	0x00 (0_{dec})
7011:0 [▶ 110]	**Subindex**	Ctrl Ch.2	RO	0x01 (1_{dec})
	7011:01	Ctrl	RO	0x0000 (0_{dec})
8000:0 [▶ 91]	**Subindex**	COM Settings Ch.1	RW	0x1A (26_{dec})
	8000:02	Enable XON/XOFF supported tx data	RW	0x00 (0_{dec})
	8000:03	Enable XON/XOFF supported rx data	RW	0x00 (0_{dec})
	8000:04	Enable send FIFO data continuous	RW	0x00 (0_{dec})
	8000:05	Enable transfer rate optimization	RW	0x01 (1_{dec})
	8000:07	Enable point to point connection (RS422)	RW	0x00 (0_{dec})
	8000:11	Baud rate	RW	0x06 (6_{dec})
	8000:15	Data frame	RW	0x03 (3_{dec})
	8000:1A	Rx buffer full notification	RW	0x0360 (864_{dec})
8010:0 [▶ 92]	**Subindex**	COM Settings Ch.2	RW	0x1A (26_{dec})
	8010:02	Enable XON/XOFF supported tx data	RW	0x00 (0_{dec})
	8010:03	Enable XON/XOFF supported rx data	RW	0x00 (0_{dec})
	8010:04	Enable send FIFO data continuous	RW	0x00 (0_{dec})
	8010:05	Enable transfer rate optimization	RW	0x01 (1_{dec})
	8010:07	Enable point to point connection (RS422)	RW	0x00 (0_{dec})

Index (hex)		Name	Flags	Default value
	8010:11	Baud rate	RW	0x06 (6_{dec})
	8010:15	Data frame	RW	0x03 (3_{dec})
	8010:1A	Rx buffer full notification	RW	0x0360 (864_{dec})
A000:0 [▶ 110]	**Subindex**	COM Diag data Ch.1	RO	0x21 (33_{dec})
	A000:01	Buffer overflow	RO	0x00 (0_{dec})
	A000:02	Parity error	RO	0x00 (0_{dec})
	A000:03	Framing error	RO	0x00 (0_{dec})
	A000:04	Overrun error	RO	0x00 (0_{dec})
	A000:05	Buffer full	RO	0x00 (0_{dec})
	A000:11	Data bytes in send buffer	RO	0x0000 (0_{dec})
	A000:21	Data bytes in receive buffer	RO	0x0000 (0_{dec})
A010:0 [▶ 111]	**Subindex**	COM Diag data Ch.2	RO	0x21 (33_{dec})
	A010:01	Buffer overflow	RO	0x00 (0_{dec})
	A010:02	Parity error	RO	0x00 (0_{dec})
	A010:03	Framing error	RO	0x00 (0_{dec})
	A010:04	Overrun error	RO	0x00 (0_{dec})
	A010:05	Buffer full	RO	0x00 (0_{dec})
	A010:11	Data bytes in send buffer	RO	0x0000 (0_{dec})
	A010:21	Data bytes in receive buffer	RO	0x0000 (0_{dec})
F000:0 [▶ 111]	**Subindex**	Modular device profile	RO	0x02 (2_{dec})
	F000:01	Module index distance	RO	0x0010 (16_{dec})
	F000:02	Maximum number of modules	RO	0x0002 (2_{dec})
F008 [▶ 111]		Code word	RW	0x00000000 (0_{dec})
F010:0 [▶ 111]	**Subindex**	Module list	RW	0x02 (2_{dec})
	F010:01	SubIndex 001	RW	0x00000258 (600_{dec})
	F010:02	SubIndex 002	RW	0x00000258 (600_{dec})
F800:0 [▶ 92]	**Subindex**	COM Settings	RW	0x03 (3_{dec})
	F800:01	Interface Type Ch 1	RW	0x00 (0_{dec})
	F800:02	Interface Type Ch 2	RW	0x00 (0_{dec})

Key

Flags:

RO (Read Only): This object can only be read.

RW (Read/Write): This object can be read and written to.

5.9 EP6002-0002 - Object description and parameterization

EtherCAT XML Device Description

The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE - Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE notes when using/manipulating the CoE parameters:
- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary, existence of current XML description
- use “CoE reload” for resetting changes

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization [▶ 90] during commissioning
- Objects required for the selection of the interface type [▶ 92]
- Objects intended for regular operation [▶ 92], e.g. through ADS access
- Objects for indicating internal settings [▶ 93] (may be fixed)
- Further profile-specific objects [▶ 106] indicating inputs, outputs and status information

The following section first describes the objects required for normal operation, followed by a complete overview of missing objects.

Objects to be parameterized during commissioning

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default settings	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If this object is set to **"0x64616F6C"** in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0_{dec})

Index 8000 COM Settings Ch.1

Index (hex)	Name	Meaning		Data type	Flags	Default
8000:0	COM Settings Ch.1			UINT8	RO	0x1A (26_{dec})
8000:02	Enable XON/XOFF supported tx data	XON/XOFF is supported for send data		BOOLEAN	RW	0x00 (0_{dec})
8000:03	Enable XON/XOFF supported rx data	XON/XOFF is supported for receive data		BOOLEAN	RW	0x00 (0_{dec})
8000:04	Enable send FIFO data continuous	Continuous sending of data from the FIFO enabled		BOOLEAN	RW	0x00 (0_{dec})
8000:05	Enable transfer rate optimization	Switch on the transfer rate optimization		BOOLEAN	RW	0x01 (1_{dec})
8000:06	Enable half duplex	Half duplex for RS485 mode (this bit is not evaluated in RS232 and RS422 mode)		BOOLEAN	RW	0x00 (0_{dec})
		0	Full duplex: The module monitors its transmitted data.			
		1	Half duplex: The module does not monitor the data that it has transmitted itself.			
8000:07	Enable point to point connection (RS422)	0	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0x00 (0_{dec})
		1	The module is used as a point-to-point connection (RS422)			
8000:11	Baud rate	Baud Rate		BIT4	RW	0x06 (6_{dec})
		0x01	300 baud			
		0x02	600 baud			
		0x03	1200 baud			
		0x04	2400 baud			
		0x05	4800 baud			
		0x06	9600 baud			
		0x07	19200 baud			
		0x08	38400 baud			
		0x09	57600 baud			
		0x0A	115200 baud			
8000:15	Data frame	Data frame / Stop bits		BIT4	RW	0x03 (3_{dec})
		0x01	7E1			
		0x02	7O1			
		0x03	8N1			
		0x04	8E1			
		0x05	8O1			
		0x09	7E2			
		0x0A	7O2			
		0x0B	8N2			
		0x0C	8E2			
		0x0D	8O2			
8000:1A	Rx buffer full notification	The value specifies the number of data in the receive FIFO, from which the bit "buffer full" is set.		UINT16	RW	0x0360 (864_{dec})

Index 8010 COM Settings Ch.2

Index (hex)	Name	Meaning		Data type	Flags	Default
8010:0	COM Settings Ch.2			UINT8	RO	0x1A (26_{dec})
8010:02	Enable XON/XOFF supported tx data	XON/XOFF is supported for send data		BOOLEAN	RW	0x00 (0_{dec})
8010:03	Enable XON/XOFF supported rx data	XON/XOFF is supported for receive data		BOOLEAN	RW	0x00 (0_{dec})
8010:04	Enable send FIFO data continuous	Continuous sending of data from the FIFO enabled		BOOLEAN	RW	0x00 (0_{dec})
8010:05	Enable transfer rate optimization	Switch on the transfer rate optimization		BOOLEAN	RW	0x01 (1_{dec})
8010:06	Enable half duplex	Half duplex for RS485 mode (this bit is not evaluated in RS232 and RS422 mode)		BOOLEAN	RW	0x00 (0_{dec})
		0	Full duplex: The module monitors its transmitted data.			
		1	Half duplex: The module does not monitor the data that it has transmitted itself.			
8010:07	Enable point to point connection (RS422)	0	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0x00 (0_{dec})
		1	The module is used as a point-to-point connection (RS422)			
8010:11	Baud rate	Baud Rate		BIT4	RW	0x06 (6_{dec})
		0x01	300 baud			
		0x02	600 baud			
		0x03	1200 baud			
		0x04	2400 baud			
		0x05	4800 baud			
		0x06	9600 baud			
		0x07	19200 baud			
		0x08	38400 baud			
		0x09	57600 baud			
		0x0A	115200 baud			
8010:15	Data frame	Data frame / Stop bits		BIT4	RW	0x03 (3_{dec})
		0x01	7E1			
		0x02	7O1			
		0x03	8N1			
		0x04	8E1			
		0x05	8O1			
		0x09	7E2			
		0x0A	7O2			
		0x0B	8N2			
		0x0C	8E2			
		0x0D	8O2			
8010:1A	Rx buffer full notification	The value specifies the number of data in the receive FIFO, from which the bit "buffer full" is set.		UINT16	RW	0x0360 (864_{dec})

Index F800 COM Settings

Index (hex)	Name	Meaning		Data type	Flags	Default
F800:0	COM Settings			UINT8	RO	0x03 (3_{dec})
F800:01	Interface Type Ch 1	0x00	RS232	BIT1	RW	0x00 (0_{dec})
		0x01	RS485/422			
F800:02	Interface Type Ch 2	0x00	RS232	BIT1	RW	0x00 (0_{dec})
		0x01	RS485/422			

Objects for regular operation

The EP6002 has no such objects.

Additional objects

Standard objects (0x1000-0x1FFF)

The standard objects have the same meaning for all EtherCAT slaves.

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x02581389 (39326601_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EP6002-0002

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	00

Index 100A Software Version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	00

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x17724052 (393363538_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description	UINT32	RO	0x00100002 (1048578_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

Index (hex)	Name	Meaning	Data type	Flags	Default
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries	UINT8	RO	0x01 (1_{dec})
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	0x00000000 (0_{dec})

Index 1400 COM RxPDO-Par Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1400:0	COM RxPDO-Par Outputs Ch.1	PDO Parameter RxPDO 1	UINT8	RO	0x06 (6_{dec})
1400:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 1	OCTET-STRING[2]	RO	04 16

Index 1401 COM RxPDO-Par Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1401:0	COM RxPDO-Par Outputs Ch.2	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1401:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 2	OCTET-STRING[2]	RO	05 16

Index 1404 COM RxPDO-Par Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1404:0	COM RxPDO-Par Outputs Ch.1	PDO Parameter RxPDO 5	UINT8	RO	0x06 (6_{dec})
1404:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 5	OCTET-STRING[2]	RO	00 16

Index 1405 COM RxPDO-Par Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1405:0	COM RxPDO-Par Outputs Ch.2	PDO Parameter RxPDO 6	UINT8	RO	0x06 (6_{dec})
1405:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 6	OCTET-STRING[2]	RO	01 16

Index 1600 COM RxPDO-Map Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	COM RxPDO-Map Outputs Ch.1	PDO Mapping RxPDO 1	UINT8	RO	0x1C (28_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x01 (Transmit request))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x02 (Receive accepted))	UINT32	RO	0x7000:02, 1
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x03 (Init request))	UINT32	RO	0x7000:03, 1
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x04 (Send continuous))	UINT32	RO	0x7000:04, 1
1600:05	SubIndex 005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1600:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x09 (Output length))	UINT32	RO	0x7000:09, 8
1600:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT32	RO	0x7000:11, 8
1600:08	SubIndex 008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT32	RO	0x7000:12, 8
1600:09	SubIndex 009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT32	RO	0x7000:13, 8
1600:0A	SubIndex 010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT32	RO	0x7000:14, 8
1600:0B	SubIndex 011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT32	RO	0x7000:15, 8
1600:0C	SubIndex 012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT32	RO	0x7000:16, 8
1600:0D	SubIndex 013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT32	RO	0x7000:17, 8
1600:0E	SubIndex 014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT32	RO	0x7000:18, 8
1600:0F	SubIndex 015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT32	RO	0x7000:19, 8
1600:10	SubIndex 016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT32	RO	0x7000:1A, 8
1600:11	SubIndex 017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT32	RO	0x7000:1B, 8
1600:12	SubIndex 018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT32	RO	0x7000:1C, 8
1600:13	SubIndex 019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT32	RO	0x7000:1D, 8
1600:14	SubIndex 020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT32	RO	0x7000:1E, 8
1600:15	SubIndex 021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT32	RO	0x7000:1F, 8
1600:16	SubIndex 022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT32	RO	0x7000:20, 8
1600:17	SubIndex 023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT32	RO	0x7000:21, 8
1600:18	SubIndex 024	24. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT32	RO	0x7000:22, 8
1600:19	SubIndex 025	25. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT32	RO	0x7000:23, 8
1600:1A	SubIndex 026	26. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT32	RO	0x7000:24, 8
1600:1B	SubIndex 027	27. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT32	RO	0x7000:25, 8
1600:1C	SubIndex 028	28. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT32	RO	0x7000:26, 8

Index 1601 COM RxPDO-Map Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	COM RxPDO-Map Outputs Ch.2	PDO Mapping RxPDO 2	UINT8	RO	0x1C (28_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x01 (Transmit request))	UINT32	RO	0x7010:01, 1
1601:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x02 (Receive accepted))	UINT32	RO	0x7010:02, 1
1601:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x03 (Init request))	UINT32	RO	0x7010:03, 1
1601:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x04 (Send continuous))	UINT32	RO	0x7010:04, 1
1601:05	SubIndex 005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1601:06	SubIndex 006	6. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x09 (Output length))	UINT32	RO	0x7010:09, 8
1601:07	SubIndex 007	7. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x11 (Data Out 0))	UINT32	RO	0x7010:11, 8
1601:08	SubIndex 008	8. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x12 (Data Out 1))	UINT32	RO	0x7010:12, 8
1601:09	SubIndex 009	9. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x13 (Data Out 2))	UINT32	RO	0x7010:13, 8
1601:0A	SubIndex 010	10. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x14 (Data Out 3))	UINT32	RO	0x7010:14, 8
1601:0B	SubIndex 011	11. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x15 (Data Out 4))	UINT32	RO	0x7010:15, 8
1601:0C	SubIndex 012	12. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x16 (Data Out 5))	UINT32	RO	0x7010:16, 8
1601:0D	SubIndex 013	13. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x17 (Data Out 6))	UINT32	RO	0x7010:17, 8
1601:0E	SubIndex 014	14. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x18 (Data Out 7))	UINT32	RO	0x7010:18, 8
1601:0F	SubIndex 015	15. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x19 (Data Out 8))	UINT32	RO	0x7010:19, 8
1601:10	SubIndex 016	16. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1A (Data Out 9))	UINT32	RO	0x7010:1A, 8
1601:11	SubIndex 017	17. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1B (Data Out 10))	UINT32	RO	0x7010:1B, 8
1601:12	SubIndex 018	18. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1C (Data Out 11))	UINT32	RO	0x7010:1C, 8
1601:13	SubIndex 019	19. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1D (Data Out 12))	UINT32	RO	0x7010:1D, 8
1601:14	SubIndex 020	20. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1E (Data Out 13))	UINT32	RO	0x7010:1E, 8
1601:15	SubIndex 021	21. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1F (Data Out 14))	UINT32	RO	0x7010:1F, 8
1601:16	SubIndex 022	22. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x20 (Data Out 15))	UINT32	RO	0x7010:20, 8
1601:17	SubIndex 023	23. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x21 (Data Out 16))	UINT32	RO	0x7010:21, 8
1601:18	SubIndex 024	24. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x22 (Data Out 17))	UINT32	RO	0x7010:22, 8
1601:19	SubIndex 025	25. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x23 (Data Out 18))	UINT32	RO	0x7010:23, 8
1601:1A	SubIndex 026	26. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x24 (Data Out 19))	UINT32	RO	0x7010:24, 8
1601:1B	SubIndex 027	27. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x25 (Data Out 20))	UINT32	RO	0x7010:25, 8
1601:1C	SubIndex 028	28. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x26 (Data Out 21))	UINT32	RO	0x7010:26, 8

Index 1604 COM RxPDO-Map Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	COM RxPDO-Map Outputs Ch.1	PDO Mapping RxPDO 5	UINT8	RO	0x17 (23_{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0x7001 (Ctrl Ch.1), entry 0x01 (Ctrl))	UINT32	RO	0x7001:01, 16
1604:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT32	RO	0x7000:11, 8
1604:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT32	RO	0x7000:12, 8
1604:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT32	RO	0x7000:13, 8
1604:05	SubIndex 005	5. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT32	RO	0x7000:14, 8
1604:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT32	RO	0x7000:15, 8
1604:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT32	RO	0x7000:16, 8
1604:08	SubIndex 008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT32	RO	0x7000:17, 8
1604:09	SubIndex 009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT32	RO	0x7000:18, 8
1604:0A	SubIndex 010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT32	RO	0x7000:19, 8
1604:0B	SubIndex 011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT32	RO	0x7000:1A, 8
1604:0C	SubIndex 012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT32	RO	0x7000:1B, 8
1604:0D	SubIndex 013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT32	RO	0x7000:1C, 8
1604:0E	SubIndex 014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT32	RO	0x7000:1D, 8
1604:0F	SubIndex 015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT32	RO	0x7000:1E, 8
1604:10	SubIndex 016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT32	RO	0x7000:1F, 8
1604:11	SubIndex 017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT32	RO	0x7000:20, 8
1604:12	SubIndex 018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT32	RO	0x7000:21, 8
1604:13	SubIndex 019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT32	RO	0x7000:22, 8
1604:14	SubIndex 020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT32	RO	0x7000:23, 8
1604:15	SubIndex 021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT32	RO	0x7000:24, 8
1604:16	SubIndex 022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT32	RO	0x7000:25, 8
1604:17	SubIndex 023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT32	RO	0x7000:26, 8

Index 1605 COM RxPDO-Map Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	COM RxPDO-Map Outputs Ch.2	PDO Mapping RxPDO 6	UINT8	RO	0x17 (23_{dec})
1605:01	SubIndex 001	1. PDO Mapping entry (object 0x7011 (Ctrl Ch.2), entry 0x01 (Ctrl))	UINT32	RO	0x7011:01, 16
1605:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x11 (Data Out 0))	UINT32	RO	0x7010:11, 8
1605:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x12 (Data Out 1))	UINT32	RO	0x7010:12, 8
1605:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x13 (Data Out 2))	UINT32	RO	0x7010:13, 8
1605:05	SubIndex 005	5. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x14 (Data Out 3))	UINT32	RO	0x7010:14, 8
1605:06	SubIndex 006	6. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x15 (Data Out 4))	UINT32	RO	0x7010:15, 8
1605:07	SubIndex 007	7. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x16 (Data Out 5))	UINT32	RO	0x7010:16, 8
1605:08	SubIndex 008	8. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x17 (Data Out 6))	UINT32	RO	0x7010:17, 8
1605:09	SubIndex 009	9. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x18 (Data Out 7))	UINT32	RO	0x7010:18, 8
1605:0A	SubIndex 010	10. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x19 (Data Out 8))	UINT32	RO	0x7010:19, 8
1605:0B	SubIndex 011	11. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1A (Data Out 9))	UINT32	RO	0x7010:1A, 8
1605:0C	SubIndex 012	12. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1B (Data Out 10))	UINT32	RO	0x7010:1B, 8
1605:0D	SubIndex 013	13. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1C (Data Out 11))	UINT32	RO	0x7010:1C, 8
1605:0E	SubIndex 014	14. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1D (Data Out 12))	UINT32	RO	0x7010:1D, 8
1605:0F	SubIndex 015	15. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1E (Data Out 13))	UINT32	RO	0x7010:1E, 8
1605:10	SubIndex 016	16. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1F (Data Out 14))	UINT32	RO	0x7010:1F, 8
1605:11	SubIndex 017	17. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x20 (Data Out 15))	UINT32	RO	0x7010:20, 8
1605:12	SubIndex 018	18. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x21 (Data Out 16))	UINT32	RO	0x7010:21, 8
1605:13	SubIndex 019	19. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x22 (Data Out 17))	UINT32	RO	0x7010:22, 8
1605:14	SubIndex 020	20. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x23 (Data Out 18))	UINT32	RO	0x7010:23, 8
1605:15	SubIndex 021	21. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x24 (Data Out 19))	UINT32	RO	0x7010:24, 8
1605:16	SubIndex 022	22. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x25 (Data Out 20))	UINT32	RO	0x7010:25, 8
1605:17	SubIndex 023	23. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x26 (Data Out 21))	UINT32	RO	0x7010:26, 8

Index 1800 COM TxPDO-Par Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1800:0	COM TxPDO-Par Inputs Ch.1	PDO parameter TxPDO 1	UINT8	RO	0x06 (6_{dec})
1800:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1	OCTET-STRING[2]	RO	04 1A

Index 1801 COM TxPDO-Par Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1801:0	COM TxPDO-Par Inputs Ch.2	PDO parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1801:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2	OCTET-STRING[2]	RO	05 1A

Index 1804 COM TxPDO-Par Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1804:0	COM TxPDO-Par Inputs Ch.1	PDO parameter TxPDO 5	UINT8	RO	0x06 (6_{dec})
1804:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 5	OCTET-STRING[2]	RO	00 1A

Index 1805 COM TxPDO-Par Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1805:0	COM TxPDO-Par Inputs Ch.2	PDO parameter TxPDO 6	UINT8	RO	0x06 (6_{dec})
1805:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 6	OCTET-STRING[2]	RO	01 1A

Index 1A00 COM TxPDO-Map Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	COM TxPDO-Map Inputs Ch.1	PDO Mapping TxPDO 1	UINT8	RO	0x1F (31_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x01 (Transmit accepted))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x02 (Receive request))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x03 (Init accepted))	UINT32	RO	0x6000:03, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x04 (Buffer full))	UINT32	RO	0x6000:04, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x05 (Parity error))	UINT32	RO	0x6000:05, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x06 (Framing error))	UINT32	RO	0x6000:06, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x07 (Overrun error))	UINT32	RO	0x6000:07, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x09 (Input length))	UINT32	RO	0x6000:09, 8
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT32	RO	0x6000:11, 8
1A00:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT32	RO	0x6000:12, 8
1A00:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT32	RO	0x6000:13, 8
1A00:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT32	RO	0x6000:14, 8
1A00:0E	SubIndex 014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT32	RO	0x6000:15, 8
1A00:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT32	RO	0x6000:16, 8
1A00:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT32	RO	0x6000:17, 8
1A00:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT32	RO	0x6000:18, 8
1A00:12	SubIndex 018	18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT32	RO	0x6000:19, 8
1A00:13	SubIndex 019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT32	RO	0x6000:1A, 8
1A00:14	SubIndex 020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT32	RO	0x6000:1B, 8
1A00:15	SubIndex 021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT32	RO	0x6000:1C, 8
1A00:16	SubIndex 022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT32	RO	0x6000:1D, 8
1A00:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT32	RO	0x6000:1E, 8
1A00:18	SubIndex 024	24. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT32	RO	0x6000:1F, 8
1A00:19	SubIndex 025	25. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT32	RO	0x6000:20, 8
1A00:1A	SubIndex 026	26. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT32	RO	0x6000:21, 8
1A00:1B	SubIndex 027	27. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT32	RO	0x6000:22, 8
1A00:1C	SubIndex 028	28. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT32	RO	0x6000:23, 8
1A00:1D	SubIndex 029	29. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT32	RO	0x6000:24, 8
1A00:1E	SubIndex 030	30. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT32	RO	0x6000:25, 8
1A00:1F	SubIndex 031	31. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT32	RO	0x6000:26, 8

Index 1A01 COM TxPDO-Map Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	COM TxPDO-Map Inputs Ch.2	PDO Mapping TxPDO 2	UINT8	RO	0x1F (31_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x01 (Transmit accepted))	UINT32	RO	0x6010:01, 1
1A01:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x02 (Receive request))	UINT32	RO	0x6010:02, 1
1A01:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x03 (Init accepted))	UINT32	RO	0x6010:03, 1
1A01:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x04 (Buffer full))	UINT32	RO	0x6010:04, 1
1A01:05	SubIndex 005	5. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x05 (Parity error))	UINT32	RO	0x6010:05, 1
1A01:06	SubIndex 006	6. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x06 (Framing error))	UINT32	RO	0x6010:06, 1
1A01:07	SubIndex 007	7. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x07 (Overrun error))	UINT32	RO	0x6010:07, 1
1A01:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A01:09	SubIndex 009	9. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x09 (Input length))	UINT32	RO	0x6010:09, 8
1A01:0A	SubIndex 010	10. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x11 (Data In 0))	UINT32	RO	0x6010:11, 8
1A01:0B	SubIndex 011	11. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x12 (Data In 1))	UINT32	RO	0x6010:12, 8
1A01:0C	SubIndex 012	12. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x13 (Data In 2))	UINT32	RO	0x6010:13, 8
1A01:0D	SubIndex 013	13. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x14 (Data In 3))	UINT32	RO	0x6010:14, 8
1A01:0E	SubIndex 014	14. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x15 (Data In 4))	UINT32	RO	0x6010:15, 8
1A01:0F	SubIndex 015	15. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x16 (Data In 5))	UINT32	RO	0x6010:16, 8
1A01:10	SubIndex 016	16. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x17 (Data In 6))	UINT32	RO	0x6010:17, 8
1A01:11	SubIndex 017	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x18 (Data In 7))	UINT32	RO	0x6010:18, 8
1A01:12	SubIndex 018	18. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x19 (Data In 8))	UINT32	RO	0x6010:19, 8
1A01:13	SubIndex 019	19. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1A (Data In 9))	UINT32	RO	0x6010:1A, 8
1A01:14	SubIndex 020	20. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1B (Data In 10))	UINT32	RO	0x6010:1B, 8
1A01:15	SubIndex 021	21. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1C (Data In 11))	UINT32	RO	0x6010:1C, 8
1A01:16	SubIndex 022	22. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1D (Data In 12))	UINT32	RO	0x6010:1D, 8
1A01:17	SubIndex 023	23. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1E (Data In 13))	UINT32	RO	0x6010:1E, 8
1A01:18	SubIndex 024	24. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1F (Data In 14))	UINT32	RO	0x6010:1F, 8
1A01:19	SubIndex 025	25. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x20 (Data In 15))	UINT32	RO	0x6010:20, 8
1A01:1A	SubIndex 026	26. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x21 (Data In 16))	UINT32	RO	0x6010:21, 8
1A01:1B	SubIndex 027	27. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x22 (Data In 17))	UINT32	RO	0x6010:22, 8
1A01:1C	SubIndex 028	28. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x23 (Data In 18))	UINT32	RO	0x6010:23, 8
1A01:1D	SubIndex 029	29. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x24 (Data In 19))	UINT32	RO	0x6010:24, 8
1A01:1E	SubIndex 030	30. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x25 (Data In 20))	UINT32	RO	0x6010:25, 8
1A01:1F	SubIndex 031	31. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x26 (Data In 21))	UINT32	RO	0x6010:26, 8

Index 1A04 COM TxPDO-Map Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	COM TxPDO-Map Inputs Ch.1	PDO Mapping TxPDO 5	UINT8	RO	0x17 (23_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (Status Ch.1), entry 0x01 (Status))	UINT32	RO	0x6001:01, 16
1A04:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT32	RO	0x6000:11, 8
1A04:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT32	RO	0x6000:12, 8
1A04:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT32	RO	0x6000:13, 8
1A04:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT32	RO	0x6000:14, 8
1A04:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT32	RO	0x6000:15, 8
1A04:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT32	RO	0x6000:16, 8
1A04:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT32	RO	0x6000:17, 8
1A04:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT32	RO	0x6000:18, 8
1A04:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT32	RO	0x6000:19, 8
1A04:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT32	RO	0x6000:1A, 8
1A04:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT32	RO	0x6000:1B, 8
1A04:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT32	RO	0x6000:1C, 8
1A04:0E	SubIndex 014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT32	RO	0x6000:1D, 8
1A04:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT32	RO	0x6000:1E, 8
1A04:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT32	RO	0x6000:1F, 8
1A04:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT32	RO	0x6000:20, 8
1A04:12	SubIndex 018	18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT32	RO	0x6000:21, 8
1A04:13	SubIndex 019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT32	RO	0x6000:22, 8
1A04:14	SubIndex 020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT32	RO	0x6000:23, 8
1A04:15	SubIndex 021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT32	RO	0x6000:24, 8
1A04:16	SubIndex 022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT32	RO	0x6000:25, 8
1A04:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT32	RO	0x6000:26, 8

Index 1A05 COM TxPDO-Map Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	COM TxPDO-Map Inputs Ch.2	PDO Mapping TxPDO 6	UINT8	RO	0x17 (23_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6011 (Status Ch.2), entry 0x01 (Status))	UINT32	RO	0x6011:01, 16
1A05:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x11 (Data In 0))	UINT32	RO	0x6010:11, 8
1A05:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x12 (Data In 1))	UINT32	RO	0x6010:12, 8
1A05:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x13 (Data In 2))	UINT32	RO	0x6010:13, 8
1A05:05	SubIndex 005	5. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x14 (Data In 3))	UINT32	RO	0x6010:14, 8
1A05:06	SubIndex 006	6. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x15 (Data In 4))	UINT32	RO	0x6010:15, 8
1A05:07	SubIndex 007	7. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x16 (Data In 5))	UINT32	RO	0x6010:16, 8
1A05:08	SubIndex 008	8. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x17 (Data In 6))	UINT32	RO	0x6010:17, 8
1A05:09	SubIndex 009	9. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x18 (Data In 7))	UINT32	RO	0x6010:18, 8
1A05:0A	SubIndex 010	10. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x19 (Data In 8))	UINT32	RO	0x6010:19, 8
1A05:0B	SubIndex 011	11. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1A (Data In 9))	UINT32	RO	0x6010:1A, 8
1A05:0C	SubIndex 012	12. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1B (Data In 10))	UINT32	RO	0x6010:1B, 8
1A05:0D	SubIndex 013	13. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1C (Data In 11))	UINT32	RO	0x6010:1C, 8
1A05:0E	SubIndex 014	14. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1D (Data In 12))	UINT32	RO	0x6010:1D, 8
1A05:0F	SubIndex 015	15. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1E (Data In 13))	UINT32	RO	0x6010:1E, 8
1A05:10	SubIndex 016	16. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1F (Data In 14))	UINT32	RO	0x6010:1F, 8
1A05:11	SubIndex 017	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x20 (Data In 15))	UINT32	RO	0x6010:20, 8
1A05:12	SubIndex 018	18. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x21 (Data In 16))	UINT32	RO	0x6010:21, 8
1A05:13	SubIndex 019	19. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x22 (Data In 17))	UINT32	RO	0x6010:22, 8
1A05:14	SubIndex 020	20. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x23 (Data In 18))	UINT32	RO	0x6010:23, 8
1A05:15	SubIndex 021	21. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x24 (Data In 19))	UINT32	RO	0x6010:24, 8
1A05:16	SubIndex 022	22. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x25 (Data In 20))	UINT32	RO	0x6010:25, 8
1A05:17	SubIndex 023	23. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x26 (Data In 21))	UINT32	RO	0x6010:26, 8

Index 1C00 Sync manager type

Index (hex)	Name	Meaning	Data type	Flags	Default
1C00:0	Sync manager type	Using the Sync Managers	UINT8	RO	0x04 (4_{dec})
1C00:01	SubIndex 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1_{dec})
1C00:02	SubIndex 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2_{dec})
1C00:03	SubIndex 003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3_{dec})
1C00:04	SubIndex 004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4_{dec})

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1604 (5636_{dec})
1C12:02	Subindex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1605 (5637_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x02 (2_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A04 (6660_{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A05 (6661_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0000 (0_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000384 (900_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 105])	UINT16	RO	0xC007 (49159_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00002710 (10000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 1C32:03 [▶ 105], 1C32:05 [▶ 105], 1C32:06 [▶ 105], 1C32:09 [▶ 105], 1C33:03 [▶ 106], 1C33:06 [▶ 105], 1C33:09 [▶ 106] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Maximum Delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000384 (900_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0000 (0_{dec})
1C33:02	Cycle time	as 1C32:02 [▶ 105]	UINT32	RW	0x000F4240 (1000000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 105] or 1C33:08 [▶ 106])	UINT16	RO	0xC007 (49159_{dec})
1C33:05	Minimum cycle time	as 1C32:05 [▶ 105]	UINT32	RO	0x00002710 (10000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C33:08	Command	as 1C32:08 [▶ 105]	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900_{dec})
1C33:0B	SM event missed counter	as 1C32:11 [▶ 105]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 1C32:12 [▶ 105]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 1C32:13 [▶ 105]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 1C32:32 [▶ 105]	BOOLEAN	RO	0x00 (0_{dec})

Profile-specific objects (0x6000-0xFFFF)

The profile-specific objects have the same meaning for all EtherCAT slaves that support the profile 5001.

Index 6000 COM Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	COM Inputs Ch.1		UINT8	RO	0x26 (38_{dec})
6000:01	Transmit accepted	The module acknowledges receipt of data by changing the state of this bit	BOOLEAN	RO	0x00 (0_{dec})
6000:02	Receive request	By changing the state of this bit, the module informs the controller that the DataIn bytes contain the number of bytes displayed in "Input length"	BOOLEAN	RO	0x00 (0_{dec})
6000:03	Init Accepted	The initialization is carried out from the terminal	BOOLEAN	RO	0x00 (0_{dec})
6000:04	Buffer full	The receive FIFO is full	BOOLEAN	RO	0x00 (0_{dec})
6000:05	Parity error	A parity error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6000:06	Framing error	A framing error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6000:07	Overrun error	An overrun error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6000:09	Input length	Number of input bytes available for transfer from the terminal to the controller	UINT8	RO	0x00 (0_{dec})
6000:11	Data In 0	Input data	UINT8	RO	0x00 (0_{dec})
6000:12	Data In 1	Input data	UINT8	RO	0x00 (0_{dec})
6000:13	Data In 2	Input data	UINT8	RO	0x00 (0_{dec})
6000:14	Data In 3	Input data	UINT8	RO	0x00 (0_{dec})
6000:15	Data In 4	Input data	UINT8	RO	0x00 (0_{dec})
6000:16	Data In 5	Input data	UINT8	RO	0x00 (0_{dec})
6000:17	Data In 6	Input data	UINT8	RO	0x00 (0_{dec})
6000:18	Data In 7	Input data	UINT8	RO	0x00 (0_{dec})
6000:19	Data In 8	Input data	UINT8	RO	0x00 (0_{dec})
6000:1A	Data In 9	Input data	UINT8	RO	0x00 (0_{dec})
6000:1B	Data In 10	Input data	UINT8	RO	0x00 (0_{dec})
6000:1C	Data In 11	Input data	UINT8	RO	0x00 (0_{dec})
6000:1D	Data In 12	Input data	UINT8	RO	0x00 (0_{dec})
6000:1E	Data In 13	Input data	UINT8	RO	0x00 (0_{dec})
6000:1F	Data In 14	Input data	UINT8	RO	0x00 (0_{dec})
6000:20	Data In 15	Input data	UINT8	RO	0x00 (0_{dec})
6000:21	Data In 16	Input data	UINT8	RO	0x00 (0_{dec})
6000:22	Data In 17	Input data	UINT8	RO	0x00 (0_{dec})
6000:23	Data In 18	Input data	UINT8	RO	0x00 (0_{dec})
6000:24	Data In 19	Input data	UINT8	RO	0x00 (0_{dec})
6000:25	Data In 20	Input data	UINT8	RO	0x00 (0_{dec})
6000:26	Data In 21	Input data	UINT8	RO	0x00 (0_{dec})

Index 6001 Status Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6001:0	Status Ch.1		UINT8	RO	0x01 (1_{dec})
6001:01	Status	Status word for compatible process image	UINT16	RO	0x0000 (0_{dec})

Index 6010 COM Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	COM Inputs Ch.2		UINT8	RO	0x26 (38_{dec})
6010:01	Transmit accepted	The module acknowledges receipt of data by changing the state of this bit	BOOLEAN	RO	0x00 (0_{dec})
6010:02	Receive request	By changing the state of this bit, the module informs the controller that the DataIn bytes contain the number of bytes displayed in "Input length"	BOOLEAN	RO	0x00 (0_{dec})
6010:03	Init Accepted	The initialization is carried out from the terminal	BOOLEAN	RO	0x00 (0_{dec})
6010:04	Buffer full	The receive FIFO is full	BOOLEAN	RO	0x00 (0_{dec})
6010:05	Parity error	A parity error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6010:06	Framing error	A framing error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6010:07	Overrun error	An overrun error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6010:09	Input length	Number of input bytes available for transfer from the terminal to the controller	UINT8	RO	0x00 (0_{dec})
6010:11	Data In 0	Input data	UINT8	RO	0x00 (0_{dec})
6010:12	Data In 1	Input data	UINT8	RO	0x00 (0_{dec})
6010:13	Data In 2	Input data	UINT8	RO	0x00 (0_{dec})
6010:14	Data In 3	Input data	UINT8	RO	0x00 (0_{dec})
6010:15	Data In 4	Input data	UINT8	RO	0x00 (0_{dec})
6010:16	Data In 5	Input data	UINT8	RO	0x00 (0_{dec})
6010:17	Data In 6	Input data	UINT8	RO	0x00 (0_{dec})
6010:18	Data In 7	Input data	UINT8	RO	0x00 (0_{dec})
6010:19	Data In 8	Input data	UINT8	RO	0x00 (0_{dec})
6010:1A	Data In 9	Input data	UINT8	RO	0x00 (0_{dec})
6010:1B	Data In 10	Input data	UINT8	RO	0x00 (0_{dec})
6010:1C	Data In 11	Input data	UINT8	RO	0x00 (0_{dec})
6010:1D	Data In 12	Input data	UINT8	RO	0x00 (0_{dec})
6010:1E	Data In 13	Input data	UINT8	RO	0x00 (0_{dec})
6010:1F	Data In 14	Input data	UINT8	RO	0x00 (0_{dec})
6010:20	Data In 15	Input data	UINT8	RO	0x00 (0_{dec})
6010:21	Data In 16	Input data	UINT8	RO	0x00 (0_{dec})
6010:22	Data In 17	Input data	UINT8	RO	0x00 (0_{dec})
6010:23	Data In 18	Input data	UINT8	RO	0x00 (0_{dec})
6010:24	Data In 19	Input data	UINT8	RO	0x00 (0_{dec})
6010:25	Data In 20	Input data	UINT8	RO	0x00 (0_{dec})
6010:26	Data In 21	Input data	UINT8	RO	0x00 (0_{dec})

Index 6011 Status Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
6011:0	Status Ch.2		UINT8	RO	0x01 (1_{dec})
6011:01	Status	Status word for compatible process image	UINT16	RO	0x0000 (0_{dec})

Index 7000 COM Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	COM Outputs Ch.1		UINT8	RO	0x26 (38_{dec})
7000:01	Transmit request	By changing the state of this bit, the controller informs the terminal that the DataOut bytes contain the number of bytes displayed in "Output length".	BOOLEAN	RO	0x00 (0_{dec})
7000:02	Receive accepted	The controller acknowledges receipt of data by changing the state of this bit.	BOOLEAN	RO	0x00 (0_{dec})
7000:03	Init request	The controller requests the module to initialize.	BOOLEAN	RO	0x00 (0_{dec})
7000:04	Send continuous	Continuous sending of data from the FIFO.	BOOLEAN	RO	0x00 (0_{dec})
7000:09	Output length	Number of output bytes available for transfer from the controller to the terminal.	UINT8	RO	0x00 (0_{dec})
7000:11	Data Out 0	Output data	UINT8	RO	0x00 (0_{dec})
7000:12	Data Out 1	Output data	UINT8	RO	0x00 (0_{dec})
7000:13	Data Out 2	Output data	UINT8	RO	0x00 (0_{dec})
7000:14	Data Out 3	Output data	UINT8	RO	0x00 (0_{dec})
7000:15	Data Out 4	Output data	UINT8	RO	0x00 (0_{dec})
7000:16	Data Out 5	Output data	UINT8	RO	0x00 (0_{dec})
7000:17	Data Out 6	Output data	UINT8	RO	0x00 (0_{dec})
7000:18	Data Out 7	Output data	UINT8	RO	0x00 (0_{dec})
7000:19	Data Out 8	Output data	UINT8	RO	0x00 (0_{dec})
7000:1A	Data Out 9	Output data	UINT8	RO	0x00 (0_{dec})
7000:1B	Data Out 10	Output data	UINT8	RO	0x00 (0_{dec})
7000:1C	Data Out 11	Output data	UINT8	RO	0x00 (0_{dec})
7000:1D	Data Out 12	Output data	UINT8	RO	0x00 (0_{dec})
7000:1E	Data Out 13	Output data	UINT8	RO	0x00 (0_{dec})
7000:1F	Data Out 14	Output data	UINT8	RO	0x00 (0_{dec})
7000:20	Data Out 15	Output data	UINT8	RO	0x00 (0_{dec})
7000:21	Data Out 16	Output data	UINT8	RO	0x00 (0_{dec})
7000:22	Data Out 17	Output data	UINT8	RO	0x00 (0_{dec})
7000:23	Data Out 18	Output data	UINT8	RO	0x00 (0_{dec})
7000:24	Data Out 19	Output data	UINT8	RO	0x00 (0_{dec})
7000:25	Data Out 20	Output data	UINT8	RO	0x00 (0_{dec})
7000:26	Data Out 21	Output data	UINT8	RO	0x00 (0_{dec})

Index 7001 Ctrl Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7001:0	Ctrl Ch.1		UINT8	RO	0x01 (1_{dec})
7001:01	Ctrl	Control word for compatible process image	UINT16	RO	0x0000 (0_{dec})

Index 7010 COM Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	COM Outputs Ch.2		UINT8	RO	0x26 (38_{dec})
7010:01	Transmit request	By changing the state of this bit, the controller informs the terminal that the DataOut bytes contain the number of bytes displayed in "Output length".	BOOLEAN	RO	0x00 (0_{dec})
7010:02	Receive accepted	The controller acknowledges receipt of data by changing the state of this bit.	BOOLEAN	RO	0x00 (0_{dec})
7010:03	Init request	The controller requests the module to initialize.	BOOLEAN	RO	0x00 (0_{dec})
7010:04	Send continuous	Continuous sending of data from the FIFO.	BOOLEAN	RO	0x00 (0_{dec})
7010:09	Output length	Number of output bytes available for transfer from the controller to the terminal.	UINT8	RO	0x00 (0_{dec})
7010:11	Data Out 0	Output data	UINT8	RO	0x00 (0_{dec})
7010:12	Data Out 1	Output data	UINT8	RO	0x00 (0_{dec})
7010:13	Data Out 2	Output data	UINT8	RO	0x00 (0_{dec})
7010:14	Data Out 3	Output data	UINT8	RO	0x00 (0_{dec})
7010:15	Data Out 4	Output data	UINT8	RO	0x00 (0_{dec})
7010:16	Data Out 5	Output data	UINT8	RO	0x00 (0_{dec})
7010:17	Data Out 6	Output data	UINT8	RO	0x00 (0_{dec})
7010:18	Data Out 7	Output data	UINT8	RO	0x00 (0_{dec})
7010:19	Data Out 8	Output data	UINT8	RO	0x00 (0_{dec})
7010:1A	Data Out 9	Output data	UINT8	RO	0x00 (0_{dec})
7010:1B	Data Out 10	Output data	UINT8	RO	0x00 (0_{dec})
7010:1C	Data Out 11	Output data	UINT8	RO	0x00 (0_{dec})
7010:1D	Data Out 12	Output data	UINT8	RO	0x00 (0_{dec})
7010:1E	Data Out 13	Output data	UINT8	RO	0x00 (0_{dec})
7010:1F	Data Out 14	Output data	UINT8	RO	0x00 (0_{dec})
7010:20	Data Out 15	Output data	UINT8	RO	0x00 (0_{dec})
7010:21	Data Out 16	Output data	UINT8	RO	0x00 (0_{dec})
7010:22	Data Out 17	Output data	UINT8	RO	0x00 (0_{dec})
7010:23	Data Out 18	Output data	UINT8	RO	0x00 (0_{dec})
7010:24	Data Out 19	Output data	UINT8	RO	0x00 (0_{dec})
7010:25	Data Out 20	Output data	UINT8	RO	0x00 (0_{dec})
7010:26	Data Out 21	Output data	UINT8	RO	0x00 (0_{dec})

Index 7011 Ctrl Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
7011:0	Ctrl Ch.2		UINT8	RO	0x01 (1_{dec})
7011:01	Ctrl	Control word for compatible process image	UINT16	RO	0x0000 (0_{dec})

Index A000 COM Diag data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
A000:0	COM Diag data Ch.1		UINT8	RO	0x21 (33_{dec})
A000:01	Buffer overflow	A buffer overflow has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:02	Parity error	A parity error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:03	Framing error	A framing error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:04	Overrun error	An overrun error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:05	Buffer full	The receive FIFO is full.	BOOLEAN	RO	0x00 (0_{dec})
A000:11	Data bytes in send buffer	Number of data bytes in the send FIFO	UINT16	RO	0x0000 (0_{dec})
A000:21	Data bytes in receive buffer	Number of data bytes in the receive FIFO	UINT16	RO	0x0000 (0_{dec})

Index A010 COM Diag data Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
A010:0	COM Diag data Ch.2		UINT8	RO	0x21 (33_{dec})
A010:01	Buffer overflow	A buffer overflow has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:02	Parity error	A parity error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:03	Framing error	A framing error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:04	Overrun error	An overrun error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:05	Buffer full	The receive FIFO is full.	BOOLEAN	RO	0x00 (0_{dec})
A010:11	Data bytes in send buffer	Number of data bytes in the send FIFO	UINT16	RO	0x0000 (0_{dec})
A010:21	Data bytes in receive buffer	Number of data bytes in the receive FIFO	UINT16	RO	0x0000 (0_{dec})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2_{dec})
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16_{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0002 (2_{dec})

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0	Code word		UINT32	RW	0x00000000 (0_{dec})

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list		UINT8	RW	0x02 (2_{dec})
F010:01	SubIndex 001		UINT32	RW	0x00000258 (600_{dec})
F010:02	SubIndex 002		UINT32	RW	0x00000258 (600_{dec})

5.10 EP6001-0002 - Object overview

EtherCAT XML Device Description

The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

Index (hex)		Name	Flags	Default value
1000 [▶ 119]		Device type	RO	0x00001389 (5001_{dec})
1008 [▶ 119]		Device name	RO	EP6001-0002
1009 [▶ 119]		Hardware version	RO	00
100A [▶ 119]		Software version	RO	00
1011:0 [▶ 117]	**Subindex**	Restore default parameters	RO	0x01 (1_{dec})
	1011:01	SubIndex 001	RW	0x00000000 (0_{dec})
1018:0 [▶ 119]	**Subindex**	Identity	RO	0x04 (4_{dec})
	1018:01	Vendor ID	RO	0x00000002 (2_{dec})
	1018:02	Product code	RO	0x17714052 (393298002_{dec})
	1018:03	Revision	RO	0x00000000 (0_{dec})
	1018:04	Serial number	RO	0x00000000 (0_{dec})
10F0:0 [▶ 119]	**Subindex**	Backup parameter handling	RO	0x01 (1_{dec})
	10F0:01	Checksum	RO	0x00000000 (0_{dec})
1400:0 [▶ 119]	**Subindex**	COM RxPDO-Par Outputs	RO	0x06 (6_{dec})
	1400:06	Exclude RxPDOs	RO	04 16
1404:0 [▶ 120]	**Subindex**	COM RxPDO-Par Outputs	RO	0x06 (6_{dec})
	1404:06	Exclude RxPDOs	RO	00 16
1600:0 [▶ 121]	**Subindex**	COM RxPDO-Map Outputs	RO	0x1C (28_{dec})
	1600:01	SubIndex 001	RO	0x7000:01, 1
	1600:02	SubIndex 002	RO	0x7000:02, 1
	1600:03	SubIndex 003	RO	0x7000:03, 1
	1600:04	SubIndex 004	RO	0x7000:04, 1
	1600:05	SubIndex 005	RO	0x0000:00, 4
	1600:06	SubIndex 006	RO	0x7000:09, 8
	1600:07	SubIndex 007	RO	0x7000:11, 8
	1600:08	SubIndex 008	RO	0x7000:12, 8
	1600:09	SubIndex 009	RO	0x7000:13, 8
	1600:0A	SubIndex 010	RO	0x7000:14, 8
	1600:0B	SubIndex 011	RO	0x7000:15, 8
	1600:0C	SubIndex 012	RO	0x7000:16, 8
	1600:0D	SubIndex 013	RO	0x7000:17, 8
	1600:0E	SubIndex 014	RO	0x7000:18, 8
	1600:0F	SubIndex 015	RO	0x7000:19, 8
	1600:10	SubIndex 016	RO	0x7000:1A, 8
	1600:11	SubIndex 017	RO	0x7000:1B, 8
	1600:12	SubIndex 018	RO	0x7000:1C, 8
	1600:13	SubIndex 019	RO	0x7000:1D, 8
	1600:14	SubIndex 020	RO	0x7000:1E, 8
	1600:15	SubIndex 021	RO	0x7000:1F, 8
	1600:16	SubIndex 022	RO	0x7000:20, 8
	1600:17	SubIndex 023	RO	0x7000:21, 8
	1600:18	SubIndex 024	RO	0x7000:22, 8
	1600:19	SubIndex 025	RO	0x7000:23, 8
	1600:1A	SubIndex 026	RO	0x7000:24, 8
	1600:1B	SubIndex 027	RO	0x7000:25, 8
	1600:1C	SubIndex 028	RO	0x7000:26, 8

Index (hex)		Name	Flags	Default value
1604:0 [▶ 122]	**Subindex**	COM RxPDO-Map Outputs	RO	0x17 (23_{dec})
	1604:01	SubIndex 001	RO	0x7001:01, 16
	1604:02	SubIndex 002	RO	0x7000:11, 8
	1604:03	SubIndex 003	RO	0x7000:12, 8
	1604:04	SubIndex 004	RO	0x7000:13, 8
	1604:05	SubIndex 005	RO	0x7000:14, 8
	1604:06	SubIndex 006	RO	0x7000:15, 8
	1604:07	SubIndex 007	RO	0x7000:16, 8
	1604:08	SubIndex 008	RO	0x7000:17, 8
	1604:09	SubIndex 009	RO	0x7000:18, 8
	1604:0A	SubIndex 010	RO	0x7000:19, 8
	1604:0B	SubIndex 011	RO	0x7000:1A, 8
	1604:0C	SubIndex 012	RO	0x7000:1B, 8
	1604:0D	SubIndex 013	RO	0x7000:1C, 8
	1604:0E	SubIndex 014	RO	0x7000:1D, 8
	1604:0F	SubIndex 015	RO	0x7000:1E, 8
	1604:10	SubIndex 016	RO	0x7000:1F, 8
	1604:11	SubIndex 017	RO	0x7000:20, 8
	1604:12	SubIndex 018	RO	0x7000:21, 8
	1604:13	SubIndex 019	RO	0x7000:22, 8
	1604:14	SubIndex 020	RO	0x7000:23, 8
	1604:15	SubIndex 021	RO	0x7000:24, 8
	1604:16	SubIndex 022	RO	0x7000:25, 8
	1604:17	SubIndex 023	RO	0x7000:26, 8
1608:0 [▶ 122]	**Subindex**	DIG RxPDO-Map Outputs	RO	0x17 (23_{dec})
	1608:01	SubIndex 001	RO	0x7010:01, 1
	1608:02	SubIndex 002	RO	0x7010:02, 1
	1608:03	SubIndex 003	RO	0x0000:00, 14
1800:0 [▶ 123]	**Subindex**	COM TxPDO-Par Inputs	RO	0x06 (6_{dec})
	1800:06	Exclude TxPDOs	RO	04 1A
1804:0 [▶ 123]	**Subindex**	COM TxPDO-Par Inputs	RO	0x06 (6_{dec})
	1804:06	Exclude TxPDOs	RO	00 1A
1A00:0 [▶ 124]	**Subindex**	COM TxPDO-Map Inputs	RO	0x1F (31_{dec})
	1A00:01	SubIndex 001	RO	0x6000:01, 1
	1A00:02	SubIndex 002	RO	0x6000:02, 1
	1A00:03	SubIndex 003	RO	0x6000:03, 1
	1A00:04	SubIndex 004	RO	0x6000:04, 1
	1A00:05	SubIndex 005	RO	0x6000:05, 1
	1A00:06	SubIndex 006	RO	0x6000:06, 1
	1A00:07	SubIndex 007	RO	0x6000:07, 1
	1A00:08	SubIndex 008	RO	0x0000:00, 1
	1A00:09	SubIndex 009	RO	0x6000:09, 8
	1A00:0A	SubIndex 010	RO	0x6000:11, 8
	1A00:0B	SubIndex 011	RO	0x6000:12, 8
	1A00:0C	SubIndex 012	RO	0x6000:13, 8
	1A00:0D	SubIndex 013	RO	0x6000:14, 8
	1A00:0E	SubIndex 014	RO	0x6000:15, 8
	1A00:0F	SubIndex 015	RO	0x6000:16, 8
	1A00:10	SubIndex 016	RO	0x6000:17, 8
	1A00:11	SubIndex 017	RO	0x6000:18, 8
	1A00:12	SubIndex 018	RO	0x6000:19, 8
	1A00:13	SubIndex 019	RO	0x6000:1A, 8
	1A00:14	SubIndex 020	RO	0x6000:1B, 8
	1A00:15	SubIndex 021	RO	0x6000:1C, 8
	1A00:16	SubIndex 022	RO	0x6000:1D, 8
	1A00:17	SubIndex 023	RO	0x6000:1E, 8
	1A00:18	SubIndex 024	RO	0x6000:1F, 8
	1A00:19	SubIndex 025	RO	0x6000:20, 8

Index (hex)		Name	Flags	Default value
	1A00:1A	SubIndex 026	RO	0x6000:21, 8
	1A00:1B	SubIndex 027	RO	0x6000:22, 8
	1A00:1C	SubIndex 028	RO	0x6000:23, 8
	1A00:1D	SubIndex 029	RO	0x6000:24, 8
	1A00:1E	SubIndex 030	RO	0x6000:25, 8
	1A00:1F	SubIndex 031	RO	0x6000:26, 8
1A04:0 [▶ 125]	**Subindex**	COM TxPDO-Map Inputs	RO	0x17 (23_{dec})
	1A04:01	SubIndex 001	RO	0x6001:01, 16
	1A04:02	SubIndex 002	RO	0x6000:11, 8
	1A04:03	SubIndex 003	RO	0x6000:12, 8
	1A04:04	SubIndex 004	RO	0x6000:13, 8
	1A04:05	SubIndex 005	RO	0x6000:14, 8
	1A04:06	SubIndex 006	RO	0x6000:15, 8
	1A04:07	SubIndex 007	RO	0x6000:16, 8
	1A04:08	SubIndex 008	RO	0x6000:17, 8
	1A04:09	SubIndex 009	RO	0x6000:18, 8
	1A04:0A	SubIndex 010	RO	0x6000:19, 8
	1A04:0B	SubIndex 011	RO	0x6000:1A, 8
	1A04:0C	SubIndex 012	RO	0x6000:1B, 8
	1A04:0D	SubIndex 013	RO	0x6000:1C, 8
	1A04:0E	SubIndex 014	RO	0x6000:1D, 8
	1A04:0F	SubIndex 015	RO	0x6000:1E, 8
	1A04:10	SubIndex 016	RO	0x6000:1F, 8
	1A04:11	SubIndex 017	RO	0x6000:20, 8
	1A04:12	SubIndex 018	RO	0x6000:21, 8
	1A04:13	SubIndex 019	RO	0x6000:22, 8
	1A04:14	SubIndex 020	RO	0x6000:23, 8
	1A04:15	SubIndex 021	RO	0x6000:24, 8
	1A04:16	SubIndex 022	RO	0x6000:25, 8
	1A04:17	SubIndex 023	RO	0x6000:26, 8
1A08:0 [▶ 125]	**Subindex**	DIG TxPDO-Map Inputs	RO	0x03 (3_{dec})
	1A08:01	SubIndex 001	RO	0x6010:01, 1
	1A08:02	SubIndex 002	RO	0x6010:02, 1
	1A05:03	SubIndex 003	RO	0x0000:00, 14
1C00:0 [▶ 126]	**Subindex**	Sync manager type	RO	0x04 (4_{dec})
	1C00:01	SubIndex 001	RO	0x01 (1_{dec})
	1C00:02	SubIndex 002	RO	0x02 (2_{dec})
	1C00:03	SubIndex 003	RO	0x03 (3_{dec})
	1C00:04	SubIndex 004	RO	0x04 (4_{dec})
1C12:0 [▶ 126]	**Subindex**	RxPDO assign	RW	0x01 (1_{dec})
	1C12:01	SubIndex 001	RW	0x1604 (5636_{dec})
	1C12:02	SubIndex 002	RW	--
1C13:0 [▶ 126]	**Subindex**	TxPDO assign	RW	0x01 (1_{dec})
	1C13:01	SubIndex 001	RW	0x1A04 (6660_{dec})
	1C13:02	SubIndex 002	RW	--
1C32:0 [▶ 127]	**Subindex**	SM output parameter	RO	0x20 (32_{dec})
	1C32:01	Sync mode	RW	0x0000 (0_{dec})
	1C32:02	Cycle time	RW	0x0003D090 (250000_{dec})
	1C32:03	Shift time	RO	0x00000384 (900_{dec})
	1C32:04	Sync modes supported	RO	0x0001 (1_{dec})
	1C32:05	Minimum cycle time	RO	0x0003D090 (250000_{dec})
	1C32:06	Calc and copy time	RO	0x00000000 (0_{dec})
	1C32:07	Minimum delay time	RO	0x00000384 (900_{dec})
	1C32:08	Command	RW	0x0000 (0_{dec})
	1C32:09	Maximum Delay time	RO	0x00000384 (900_{dec})
	1C32:0B	SM event missed counter	RO	0x0000 (0_{dec})
	1C32:0C	Cycle exceeded counter	RO	0x0000 (0_{dec})
	1C32:0D	Shift too short counter	RO	0x0000 (0_{dec})

Index (hex)		Name	Flags	Default value
	1C32:20	Sync error	RO	0x00 (0_{dec})
1C33:0 [▶ 128]	**Subindex**	SM input parameter	RO	0x20 (32_{dec})
	1C33:01	Sync mode	RW	0x0000 (0_{dec})
	1C33:02	Cycle time	RW	0x0003D090 (250000_{dec})
	1C33:03	Shift time	RO	0x00000384 (900_{dec})
	1C33:04	Sync modes supported	RO	0xC007 (49159_{dec})
	1C33:05	Minimum cycle time	RO	0x0003D090 (250000_{dec})
	1C33:06	Calc and copy time	RO	0x00000000 (0_{dec})
	1C33:07	Minimum delay time	RO	0x00000384 (900_{dec})
	1C33:08	Command	RW	0x0000 (0_{dec})
	1C33:09	Maximum Delay time	RO	0x00000384 (900_{dec})
	1C33:0B	SM event missed counter	RO	0x0000 (0_{dec})
	1C33:0C	Cycle exceeded counter	RO	0x0000 (0_{dec})
	1C33:0D	Shift too short counter	RO	0x0000 (0_{dec})
	1C33:20	Sync error	RO	0x00 (0_{dec})
6000:0 [▶ 129]	**Subindex**	COM Inputs Ch.1	RO	0x26 (38_{dec})
	6000:01	Transmit accepted	RO	0x00 (0_{dec})
	6000:02	Receive request	RO	0x00 (0_{dec})
	6000:03	Init Accepted	RO	0x00 (0_{dec})
	6000:04	Buffer full	RO	0x00 (0_{dec})
	6000:05	Parity error	RO	0x00 (0_{dec})
	6000:06	Framing error	RO	0x00 (0_{dec})
	6000:07	Overrun error	RO	0x00 (0_{dec})
	6000:09	Input length	RO	0x00 (0_{dec})
	6000:11	Data In 0	RO	0x00 (0_{dec})
	6000:12	Data In 1	RO	0x00 (0_{dec})
	6000:13	Data In 2	RO	0x00 (0_{dec})
	6000:14	Data In 3	RO	0x00 (0_{dec})
	6000:15	Data In 4	RO	0x00 (0_{dec})
	6000:16	Data In 5	RO	0x00 (0_{dec})
	6000:17	Data In 6	RO	0x00 (0_{dec})
	6000:18	Data In 7	RO	0x00 (0_{dec})
	6000:19	Data In 8	RO	0x00 (0_{dec})
	6000:1A	Data In 9	RO	0x00 (0_{dec})
	6000:1B	Data In 10	RO	0x00 (0_{dec})
	6000:1C	Data In 11	RO	0x00 (0_{dec})
	6000:1D	Data In 12	RO	0x00 (0_{dec})
	6000:1E	Data In 13	RO	0x00 (0_{dec})
	6000:1F	Data In 14	RO	0x00 (0_{dec})
	6000:20	Data In 15	RO	0x00 (0_{dec})
	6000:21	Data In 16	RO	0x00 (0_{dec})
	6000:22	Data In 17	RO	0x00 (0_{dec})
	6000:23	Data In 18	RO	0x00 (0_{dec})
	6000:24	Data In 19	RO	0x00 (0_{dec})
	6000:25	Data In 20	RO	0x00 (0_{dec})
	6000:26	Data In 21	RO	0x00 (0_{dec})
6001:0 [▶ 129]	**Subindex**	Status Ch.1	RO	0x01 (1_{dec})
	6001:01	Status	RO	0x0000 (0_{dec})
6010:0 [▶ 129]	**Subindex**	DIG Inputs	RO	0x02 (2_{dec})
	6010:01	Digital Input 1	RO	0x00 (0_{dec})
	6010:02	Digital Input 2	RO	0x00 (0_{dec})
7000:0 [▶ 130]	**Subindex**	COM Outputs Ch.1	RO	0x26 (38_{dec})
	7000:01	Transmit request	RO	0x00 (0_{dec})
	7000:02	Receive accepted	RO	0x00 (0_{dec})
	7000:03	Init request	RO	0x00 (0_{dec})
	7000:04	Send continuous	RO	0x00 (0_{dec})
	7000:09	Output length	RO	0x00 (0_{dec})
	7000:11	Data Out 0	RO	0x00 (0_{dec})

Index (hex)		Name	Flags	Default value
	7000:12	Data Out 1	RO	0x00 (0_{dec})
	7000:13	Data Out 2	RO	0x00 (0_{dec})
	7000:14	Data Out 3	RO	0x00 (0_{dec})
	7000:15	Data Out 4	RO	0x00 (0_{dec})
	7000:16	Data Out 5	RO	0x00 (0_{dec})
	7000:17	Data Out 6	RO	0x00 (0_{dec})
	7000:18	Data Out 7	RO	0x00 (0_{dec})
	7000:19	Data Out 8	RO	0x00 (0_{dec})
	7000:1A	Data Out 9	RO	0x00 (0_{dec})
	7000:1B	Data Out 10	RO	0x00 (0_{dec})
	7000:1C	Data Out 11	RO	0x00 (0_{dec})
	7000:1D	Data Out 12	RO	0x00 (0_{dec})
	7000:1E	Data Out 13	RO	0x00 (0_{dec})
	7000:1F	Data Out 14	RO	0x00 (0_{dec})
	7000:20	Data Out 15	RO	0x00 (0_{dec})
	7000:21	Data Out 16	RO	0x00 (0_{dec})
	7000:22	Data Out 17	RO	0x00 (0_{dec})
	7000:23	Data Out 18	RO	0x00 (0_{dec})
	7000:24	Data Out 19	RO	0x00 (0_{dec})
	7000:25	Data Out 20	RO	0x00 (0_{dec})
	7000:26	Data Out 21	RO	0x00 (0_{dec})
7001:0 [▶ 130]	**Subindex**	Ctrl Ch.1	RO	0x01 (1_{dec})
	7001:01	Ctrl	RO	0x0000 (0_{dec})
7010:0 [▶ 130]	**Subindex**	DIG Outputs	RO	0x02 (2_{dec})
	7010:01	Digital Output 1	RO	0x00 (0_{dec})
	7010:02	Digital Output 2	RO	0x00 (0_{dec})
8000:0 [▶ 118]	**Subindex**	COM Settings Ch.1	RW	0x1B (27_{dec})
	8000:02	Enable XON/XOFF supported tx data	RW	0x00 (0_{dec})
	8000:03	Enable XON/XOFF supported rx data	RW	0x00 (0_{dec})
	8000:04	Enable send FIFO data continuous	RW	0x00 (0_{dec})
	8000:05	Enable transfer rate optimization	RW	0x01 (1_{dec})
	8000:07	Enable point to point connection (RS422)	RW	0x00 (0_{dec})
	8000:11	Baud rate	RW	0x06 (6_{dec})
	8000:15	Data frame	RW	0x03 (3_{dec})
	8000:19	Sensor Power Supply Output	RW	0x01 (1_{dec})
	8000:1A	Rx buffer full notification	RW	0x0360 (864_{dec})
	8000:1B	Explicit baud rate	RW	0x00002580 (9600_{dec})
A000:0 [▶ 130]	**Subindex**	COM Diag data Ch.1	RO	0x21 (33_{dec})
	A000:01	Buffer overflow	RO	0x00 (0_{dec})
	A000:02	Parity error	RO	0x00 (0_{dec})
	A000:03	Framing error	RO	0x00 (0_{dec})
	A000:04	Overrun error	RO	0x00 (0_{dec})
	A000:05	Buffer full	RO	0x00 (0_{dec})
	A000:11	Data bytes in send buffer	RO	0x0000 (0_{dec})
	A000:21	Data bytes in receive buffer	RO	0x0000 (0_{dec})
F000:0 [▶ 131]	**Subindex**	Modular device profile	RO	0x02 (2_{dec})
	F000:01	Module index distance	RO	0x0010 (16_{dec})
	F000:02	Maximum number of modules	RO	0x0002 (2_{dec})
F008 [▶ 131]		Code word	RW	0x00000000 (0_{dec})
F010:0 [▶ 131]	**Subindex**	Module list	RW	0x02 (2_{dec})
	F010:01	SubIndex 001	RW	0x00000258 (600_{dec})
	F010:02	SubIndex 002	RW	0x00000118 (280_{dec})
F800:0 [▶ 118]	**Subindex**	COM Settings	RW	0x03 (3_{dec})
	F800:01	Interface Type Ch 1	RW	0x00 (0_{dec})

Key

Flags:

RO (Read Only):	This object can only be read.
RW (Read/Write):	This object can be read and written to.

5.11 EP6001-0002 - Object description and parameterization

EtherCAT XML Device Description

The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE - Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE notes when using/manipulating the CoE parameters:
- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary, existence of current XML description
- use “CoE reload” for resetting changes

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization [▶ 117] during commissioning
- Objects required for the selection of the interface type [▶ 118]
- Objects intended for regular operation [▶ 118], e.g. through ADS access
- Objects for indicating internal settings [▶ 119] (may be fixed)
- Further profile-specific objects [▶ 128] indicating inputs, outputs and status information

The following section first describes the objects required for normal operation, followed by a complete overview of missing objects.

Objects to be parameterized during commissioning

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default settings	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If this object is set to **"0x64616F6C"** in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0_{dec})

Index 8000 COM Settings Ch.1

Index (hex)	Name	Meaning		Data type	Flags	Default
8000:0	COM Settings Ch.1			UINT8	RO	0x1A (26_{dec})
8000:02	Enable XON/XOFF supported tx data	XON/XOFF is supported for send data		BOOLEAN	RW	0x00 (0_{dec})
8000:03	Enable XON/XOFF supported rx data	XON/XOFF is supported for receive data		BOOLEAN	RW	0x00 (0_{dec})
8000:04	Enable send FIFO data continuous	Continuous sending of data from the FIFO enabled		BOOLEAN	RW	0x00 (0_{dec})
8000:05	Enable transfer rate optimization	Switch on the transfer rate optimization		BOOLEAN	RW	0x01 (1_{dec})
8000:06	Enable half duplex	Half duplex for RS485 mode (this bit is not evaluated in RS232 and RS422 mode)		BOOLEAN	RW	0x00 (0_{dec})
		0	Full duplex: The module monitors its transmitted data.			
		1	Half duplex: The module does not monitor the data that it has transmitted itself.			
8000:07	Enable point to point connection (RS422)	0	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0x00 (0_{dec})
		1	The module is used as a point-to-point connection (RS422)			
8000:11	Baud rate	Baud Rate		BIT4	RW	0x06 (6_{dec})
		0x01	300 baud			
		0x02	600 baud			
		0x03	1200 baud			
		0x04	2400 baud			
		0x05	4800 baud			
		0x06	9600 baud			
		0x07	19200 baud			
		0x08	38400 baud			
		0x09	57600 baud			
		0x0A	115200 baud			
8000:15	Data frame	Data frame / Stop bits		BIT4	RW	0x03 (3_{dec})
		0x01	7E1			
		0x02	7O1			
		0x03	8N1			
		0x04	8E1			
		0x05	8O1			
		0x09	7E2			
		0x0A	7O2			
		0x0B	8N2			
		0x0C	8E2			
		0x0D	8O2			
8000:19	Sensor Power Supply Output	0: Off 1: 5 V		BIT4	RW	5 V (1)
8000:1A	Rx buffer full notification	The value specifies the number of data in the receive FIFO, from which the bit "buffer full" is set.		UINT16	RW	0x0360 (864_{dec})
8000:1B	Explicit baudrate	In this object the desired baud rate can be entered directly as a number.		UINT32	RW	0x00002580 (9600_{dec})

Index F800 COM Settings

Index (hex)	Name	Meaning		Data type	Flags	Default
F800:0	COM Settings			UINT8	RO	0x01 (1_{dec})
F800:01	Interface Type Ch 1	0x00	RS232	BIT1	RW	0x00 (0_{dec})

Objects for regular operation

The EP6002 has no such objects.

Additional objects

Standard objects (0x1000-0x1FFF)

The standard objects have the same meaning for all EtherCAT slaves.

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x00001389 (5001_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EP6001-0002

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	00

Index 100A Software Version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	00

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x17714052 (393298002_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description	UINT32	RO	0x00100002 (1048578_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

Index (hex)	Name	Meaning	Data type	Flags	Default
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries	UINT8	RO	0x01 (1_{dec})
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	0x00000000 (0_{dec})

Index 1400 COM RxPDO-Par Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1400:0	COM RxPDO-Par Outputs	PDO Parameter RxPDO 1	UINT8	RO	0x06 (6_{dec})
1400:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 1	OCTET-STRING[2]	RO	04 16

Index 1404 COM RxPDO-Par Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1404:0	COM RxPDO-Par Outputs	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1404:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 5	OCTET-STRING[2]	RO	00 16

Index 1600 COM RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	COM RxPDO-Map Outputs	PDO Mapping RxPDO 1	UINT8	RO	0x1C (28_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x01 (Transmit request))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x02 (Receive accepted))	UINT32	RO	0x7000:02, 1
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x03 (Init request))	UINT32	RO	0x7000:03, 1
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x04 (Send continuous))	UINT32	RO	0x7000:04, 1
1600:05	SubIndex 005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1600:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x09 (Output length))	UINT32	RO	0x7000:09, 8
1600:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT32	RO	0x7000:11, 8
1600:08	SubIndex 008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT32	RO	0x7000:12, 8
1600:09	SubIndex 009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT32	RO	0x7000:13, 8
1600:0A	SubIndex 010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT32	RO	0x7000:14, 8
1600:0B	SubIndex 011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT32	RO	0x7000:15, 8
1600:0C	SubIndex 012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT32	RO	0x7000:16, 8
1600:0D	SubIndex 013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT32	RO	0x7000:17, 8
1600:0E	SubIndex 014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT32	RO	0x7000:18, 8
1600:0F	SubIndex 015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT32	RO	0x7000:19, 8
1600:10	SubIndex 016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT32	RO	0x7000:1A, 8
1600:11	SubIndex 017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT32	RO	0x7000:1B, 8
1600:12	SubIndex 018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT32	RO	0x7000:1C, 8
1600:13	SubIndex 019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT32	RO	0x7000:1D, 8
1600:14	SubIndex 020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT32	RO	0x7000:1E, 8
1600:15	SubIndex 021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT32	RO	0x7000:1F, 8
1600:16	SubIndex 022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT32	RO	0x7000:20, 8
1600:17	SubIndex 023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT32	RO	0x7000:21, 8
1600:18	SubIndex 024	24. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT32	RO	0x7000:22, 8
1600:19	SubIndex 025	25. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT32	RO	0x7000:23, 8
1600:1A	SubIndex 026	26. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT32	RO	0x7000:24, 8
1600:1B	SubIndex 027	27. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT32	RO	0x7000:25, 8
1600:1C	SubIndex 028	28. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT32	RO	0x7000:26, 8

Index 1604 COM RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	COM RxPDO-Map Outputs	PDO Mapping RxPDO 2	UINT8	RO	0x17 (23_{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0x7001 (Ctrl Ch.1), entry 0x01 (Ctrl))	UINT32	RO	0x7001:01, 16
1604:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT32	RO	0x7000:11, 8
1604:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT32	RO	0x7000:12, 8
1604:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT32	RO	0x7000:13, 8
1604:05	SubIndex 005	5. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT32	RO	0x7000:14, 8
1604:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT32	RO	0x7000:15, 8
1604:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT32	RO	0x7000:16, 8
1604:08	SubIndex 008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT32	RO	0x7000:17, 8
1604:09	SubIndex 009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT32	RO	0x7000:18, 8
1604:0A	SubIndex 010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT32	RO	0x7000:19, 8
1604:0B	SubIndex 011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT32	RO	0x7000:1A, 8
1604:0C	SubIndex 012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT32	RO	0x7000:1B, 8
1604:0D	SubIndex 013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT32	RO	0x7000:1C, 8
1604:0E	SubIndex 014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT32	RO	0x7000:1D, 8
1604:0F	SubIndex 015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT32	RO	0x7000:1E, 8
1604:10	SubIndex 016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT32	RO	0x7000:1F, 8
1604:11	SubIndex 017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT32	RO	0x7000:20, 8
1604:12	SubIndex 018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT32	RO	0x7000:21, 8
1604:13	SubIndex 019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT32	RO	0x7000:22, 8
1604:14	SubIndex 020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT32	RO	0x7000:23, 8
1604:15	SubIndex 021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT32	RO	0x7000:24, 8
1604:16	SubIndex 022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT32	RO	0x7000:25, 8
1604:17	SubIndex 023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT32	RO	0x7000:26, 8

Index 1608 DIG RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1608:0	DIG RxPDO-Map Outputs	PDO Mapping RxPDO 3	UINT8	RO	0x03 (3_{dec})
1608:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DIG Outputs), entry 0x01 (Digital Output 1))	UINT32	RO	0x7001:01, 1
1608:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (DIG Outputs), entry 0x02 (Digital Output 2))	UINT32	RO	0x7010:02, 1
1608:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14

Index 1800 COM TxPDO-Par Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1800:0	COM TxPDO-Par Inputs	PDO parameter TxPDO 1	UINT8	RO	0x06 (6_{dec})
1800:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1	OCTET-STRING[2]	RO	04 1A

Index 1804 COM TxPDO-Par Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1804:0	COM TxPDO-Par Inputs	PDO parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1804:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2	OCTET-STRING[2]	RO	00 1A

Index 1A00 COM TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	COM TxPDO-Map Inputs	PDO Mapping TxPDO 1	UINT8	RO	0x1F (31_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x01 (Transmit accepted))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x02 (Receive request))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x03 (Init accepted))	UINT32	RO	0x6000:03, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x04 (Buffer full))	UINT32	RO	0x6000:04, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x05 (Parity error))	UINT32	RO	0x6000:05, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x06 (Framing error))	UINT32	RO	0x6000:06, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x07 (Overrun error))	UINT32	RO	0x6000:07, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x09 (Input length))	UINT32	RO	0x6000:09, 8
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT32	RO	0x6000:11, 8
1A00:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT32	RO	0x6000:12, 8
1A00:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT32	RO	0x6000:13, 8
1A00:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT32	RO	0x6000:14, 8
1A00:0E	SubIndex 014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT32	RO	0x6000:15, 8
1A00:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT32	RO	0x6000:16, 8
1A00:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT32	RO	0x6000:17, 8
1A00:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT32	RO	0x6000:18, 8
1A00:12	SubIndex 018	18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT32	RO	0x6000:19, 8
1A00:13	SubIndex 019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT32	RO	0x6000:1A, 8
1A00:14	SubIndex 020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT32	RO	0x6000:1B, 8
1A00:15	SubIndex 021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT32	RO	0x6000:1C, 8
1A00:16	SubIndex 022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT32	RO	0x6000:1D, 8
1A00:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT32	RO	0x6000:1E, 8
1A00:18	SubIndex 024	24. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT32	RO	0x6000:1F, 8
1A00:19	SubIndex 025	25. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT32	RO	0x6000:20, 8
1A00:1A	SubIndex 026	26. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT32	RO	0x6000:21, 8
1A00:1B	SubIndex 027	27. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT32	RO	0x6000:22, 8
1A00:1C	SubIndex 028	28. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT32	RO	0x6000:23, 8
1A00:1D	SubIndex 029	29. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT32	RO	0x6000:24, 8
1A00:1E	SubIndex 030	30. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT32	RO	0x6000:25, 8
1A00:1F	SubIndex 031	31. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT32	RO	0x6000:26, 8

Index 1A04 COM TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	COM TxPDO-Map Inputs	PDO Mapping TxPDO 2	UINT8	RO	0x17 (23_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (Status Ch.1), entry 0x01 (Status))	UINT32	RO	0x6001:01, 16
1A04:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT32	RO	0x6000:11, 8
1A04:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT32	RO	0x6000:12, 8
1A04:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT32	RO	0x6000:13, 8
1A04:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT32	RO	0x6000:14, 8
1A04:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT32	RO	0x6000:15, 8
1A04:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT32	RO	0x6000:16, 8
1A04:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT32	RO	0x6000:17, 8
1A04:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT32	RO	0x6000:18, 8
1A04:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT32	RO	0x6000:19, 8
1A04:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT32	RO	0x6000:1A, 8
1A04:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT32	RO	0x6000:1B, 8
1A04:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT32	RO	0x6000:1C, 8
1A04:0E	SubIndex 014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT32	RO	0x6000:1D, 8
1A04:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT32	RO	0x6000:1E, 8
1A04:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT32	RO	0x6000:1F, 8
1A04:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT32	RO	0x6000:20, 8
1A04:12	SubIndex 018	18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT32	RO	0x6000:21, 8
1A04:13	SubIndex 019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT32	RO	0x6000:22, 8
1A04:14	SubIndex 020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT32	RO	0x6000:23, 8
1A04:15	SubIndex 021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT32	RO	0x6000:24, 8
1A04:16	SubIndex 022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT32	RO	0x6000:25, 8
1A04:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT32	RO	0x6000:26, 8

Index 1A08 DIG TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	COM TxPDO-Map Inputs Ch.2	PDO Mapping TxPDO 6	UINT8	RO	0x03 (3_{dec})
1A08:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DIG Inputs), entry 0x01 (Digital Input 1))	UINT32	RO	0x6010:01, 1
1A08:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (DIG Inputs), entry 0x02 (Digital Input 2))	UINT32	RO	0x6010:02, 1
1A08:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14

Index 1C00 Sync manager type

Index (hex)	Name	Meaning	Data type	Flags	Default
1C00:0	Sync manager type	Using the Sync Managers	UINT8	RO	0x04 (4_{dec})
1C00:01	SubIndex 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1_{dec})
1C00:02	SubIndex 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2_{dec})
1C00:03	SubIndex 003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3_{dec})
1C00:04	SubIndex 004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4_{dec})

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1604 (5636_{dec})
1C12:02	Subindex 002	2. reserved	UINT16	RW	--

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x02 (2_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A04 (6660_{dec})
1C13:02	Subindex 002	2. reserved	UINT16	RW	--

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0000 (0_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0003D090 (250000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000384 (900_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 Event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 127])	UINT16	RO	0x0001 (1_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0003D090 (250000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 1C32:03 [▶ 127], 1C32:05 [▶ 127], 1C32:06 [▶ 127], 1C32:09 [▶ 127], 1C33:03 [▶ 128], 1C33:06 [▶ 127], 1C33:09 [▶ 128] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Maximum Delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000384 (900_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0000 (0_{dec})
1C33:02	Cycle time	as 1C32:02 [▶ 127]	UINT32	RW	0x0003D090 (250000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 Event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 127] or 1C33:08 [▶ 128])	UINT16	RO	0x0001 (1_{dec})
1C33:05	Minimum cycle time	as 1C32:05 [▶ 127]	UINT32	RO	0x0003D090 (250000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C33:08	Command	as 1C32:08 [▶ 127]	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900_{dec})
1C33:0B	SM event missed counter	as 1C32:11 [▶ 127]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 1C32:12 [▶ 127]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 1C32:13 [▶ 127]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 1C32:32 [▶ 127]	BOOLEAN	RO	0x00 (0_{dec})

Profile-specific objects (0x6000-0xFFFF)

The profile-specific objects have the same meaning for all EtherCAT slaves that support the profile 5001.

Index 6000 COM Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	COM Inputs Ch.1		UINT8	RO	0x26 (38_{dec})
6000:01	Transmit accepted	The module acknowledges receipt of data by changing the state of this bit	BOOLEAN	RO	0x00 (0_{dec})
6000:02	Receive request	By changing the state of this bit, the module informs the controller that the DataIn bytes contain the number of bytes displayed in "Input length"	BOOLEAN	RO	0x00 (0_{dec})
6000:03	Init Accepted	The initialization is carried out from the terminal	BOOLEAN	RO	0x00 (0_{dec})
6000:04	Buffer full	The receive FIFO is full	BOOLEAN	RO	0x00 (0_{dec})
6000:05	Parity error	A parity error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6000:06	Framing error	A framing error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6000:07	Overrun error	An overrun error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6000:09	Input length	Number of input bytes available for transfer from the terminal to the controller	UINT8	RO	0x00 (0_{dec})
6000:11	Data In 0	Input data	UINT8	RO	0x00 (0_{dec})
6000:12	Data In 1	Input data	UINT8	RO	0x00 (0_{dec})
6000:13	Data In 2	Input data	UINT8	RO	0x00 (0_{dec})
6000:14	Data In 3	Input data	UINT8	RO	0x00 (0_{dec})
6000:15	Data In 4	Input data	UINT8	RO	0x00 (0_{dec})
6000:16	Data In 5	Input data	UINT8	RO	0x00 (0_{dec})
6000:17	Data In 6	Input data	UINT8	RO	0x00 (0_{dec})
6000:18	Data In 7	Input data	UINT8	RO	0x00 (0_{dec})
6000:19	Data In 8	Input data	UINT8	RO	0x00 (0_{dec})
6000:1A	Data In 9	Input data	UINT8	RO	0x00 (0_{dec})
6000:1B	Data In 10	Input data	UINT8	RO	0x00 (0_{dec})
6000:1C	Data In 11	Input data	UINT8	RO	0x00 (0_{dec})
6000:1D	Data In 12	Input data	UINT8	RO	0x00 (0_{dec})
6000:1E	Data In 13	Input data	UINT8	RO	0x00 (0_{dec})
6000:1F	Data In 14	Input data	UINT8	RO	0x00 (0_{dec})
6000:20	Data In 15	Input data	UINT8	RO	0x00 (0_{dec})
6000:21	Data In 16	Input data	UINT8	RO	0x00 (0_{dec})
6000:22	Data In 17	Input data	UINT8	RO	0x00 (0_{dec})
6000:23	Data In 18	Input data	UINT8	RO	0x00 (0_{dec})
6000:24	Data In 19	Input data	UINT8	RO	0x00 (0_{dec})
6000:25	Data In 20	Input data	UINT8	RO	0x00 (0_{dec})
6000:26	Data In 21	Input data	UINT8	RO	0x00 (0_{dec})

Index 6001 Status Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6001:0	Status Ch.1		UINT8	RO	0x01 (1_{dec})
6001:01	Status	Status word for compatible process image	UINT16	RO	0x0000 (0_{dec})

Index 6010 DIG Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	DIG Inputs		UINT8	RO	0x02 (2_{dec})
6010:01	Digital Input 1		BOOLEAN	RO	0x00 (0_{dec})
6010:02	Digital Input 1		BOOLEAN	RO	0x00 (0_{dec})
6010:03	Init Accepted	The initialization is carried out from the terminal	BOOLEAN	RO	0x00 (0_{dec})

Index 7000 COM Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	COM Outputs Ch.1		UINT8	RO	0x26 (38_{dec})
7000:01	Transmit request	By changing the state of this bit, the controller informs the terminal that the DataOut bytes contain the number of bytes displayed in "Output length".	BOOLEAN	RO	0x00 (0_{dec})
7000:02	Receive accepted	The controller acknowledges receipt of data by changing the state of this bit.	BOOLEAN	RO	0x00 (0_{dec})
7000:03	Init request	The controller requests the module to initialize.	BOOLEAN	RO	0x00 (0_{dec})
7000:04	Send continuous	Continuous sending of data from the FIFO.	BOOLEAN	RO	0x00 (0_{dec})
7000:09	Output length	Number of output bytes available for transfer from the controller to the terminal.	UINT8	RO	0x00 (0_{dec})
7000:11	Data Out 0	Output data	UINT8	RO	0x00 (0_{dec})
7000:12	Data Out 1	Output data	UINT8	RO	0x00 (0_{dec})
7000:13	Data Out 2	Output data	UINT8	RO	0x00 (0_{dec})
7000:14	Data Out 3	Output data	UINT8	RO	0x00 (0_{dec})
7000:15	Data Out 4	Output data	UINT8	RO	0x00 (0_{dec})
7000:16	Data Out 5	Output data	UINT8	RO	0x00 (0_{dec})
7000:17	Data Out 6	Output data	UINT8	RO	0x00 (0_{dec})
7000:18	Data Out 7	Output data	UINT8	RO	0x00 (0_{dec})
7000:19	Data Out 8	Output data	UINT8	RO	0x00 (0_{dec})
7000:1A	Data Out 9	Output data	UINT8	RO	0x00 (0_{dec})
7000:1B	Data Out 10	Output data	UINT8	RO	0x00 (0_{dec})
7000:1C	Data Out 11	Output data	UINT8	RO	0x00 (0_{dec})
7000:1D	Data Out 12	Output data	UINT8	RO	0x00 (0_{dec})
7000:1E	Data Out 13	Output data	UINT8	RO	0x00 (0_{dec})
7000:1F	Data Out 14	Output data	UINT8	RO	0x00 (0_{dec})
7000:20	Data Out 15	Output data	UINT8	RO	0x00 (0_{dec})
7000:21	Data Out 16	Output data	UINT8	RO	0x00 (0_{dec})
7000:22	Data Out 17	Output data	UINT8	RO	0x00 (0_{dec})
7000:23	Data Out 18	Output data	UINT8	RO	0x00 (0_{dec})
7000:24	Data Out 19	Output data	UINT8	RO	0x00 (0_{dec})
7000:25	Data Out 20	Output data	UINT8	RO	0x00 (0_{dec})
7000:26	Data Out 21	Output data	UINT8	RO	0x00 (0_{dec})

Index 7001 Ctrl Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7001:0	Ctrl Ch.1		UINT8	RO	0x01 (1_{dec})
7001:01	Ctrl	Control word for compatible process image	UINT16	RO	0x0000 (0_{dec})

Index 7010 DIG Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	DIG Outputs		UINT8	RO	0x26 (38_{dec})
7010:01	Digital Output 1		BOOLEAN	RO	0x00 (0_{dec})
7010:02	Digital Output 2		BOOLEAN	RO	0x00 (0_{dec})

Index A000 COM Diag data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
A000:0	COM Diag data Ch.1		UINT8	RO	0x21 (33_{dec})
A000:01	Buffer overflow	A buffer overflow has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:02	Parity error	A parity error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:03	Framing error	A framing error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:04	Overrun error	An overrun error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:05	Buffer full	The receive FIFO is full.	BOOLEAN	RO	0x00 (0_{dec})
A000:11	Data bytes in send buffer	Number of data bytes in the send FIFO	UINT16	RO	0x0000 (0_{dec})
A000:21	Data bytes in receive buffer	Number of data bytes in the receive FIFO	UINT16	RO	0x0000 (0_{dec})

Index A010 COM Diag data Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
A010:0	COM Diag data Ch.2		UINT8	RO	0x21 (33_{dec})
A010:01	Buffer overflow	A buffer overflow has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:02	Parity error	A parity error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:03	Framing error	A framing error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:04	Overrun error	An overrun error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:05	Buffer full	The receive FIFO is full.	BOOLEAN	RO	0x00 (0_{dec})
A010:11	Data bytes in send buffer	Number of data bytes in the send FIFO	UINT16	RO	0x0000 (0_{dec})
A010:21	Data bytes in receive buffer	Number of data bytes in the receive FIFO	UINT16	RO	0x0000 (0_{dec})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2_{dec})
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16_{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0002 (2_{dec})

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0	Code word		UINT32	RW	0x00000000 (0_{dec})

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list		UINT8	RW	0x02 (2_{dec})
F010:01	SubIndex 001		UINT32	RW	0x00000258 (600_{dec})
F010:02	SubIndex 002		UINT32	RW	0x00000118 (280_{dec})

5.12 Restoring the delivery state

To restore the delivery state for backup objects in ELxxxx terminals / EPxxxx boxes, the CoE object *Restore default parameters, SubIndex 001* can be selected in the TwinCAT System Manager (Config mode).

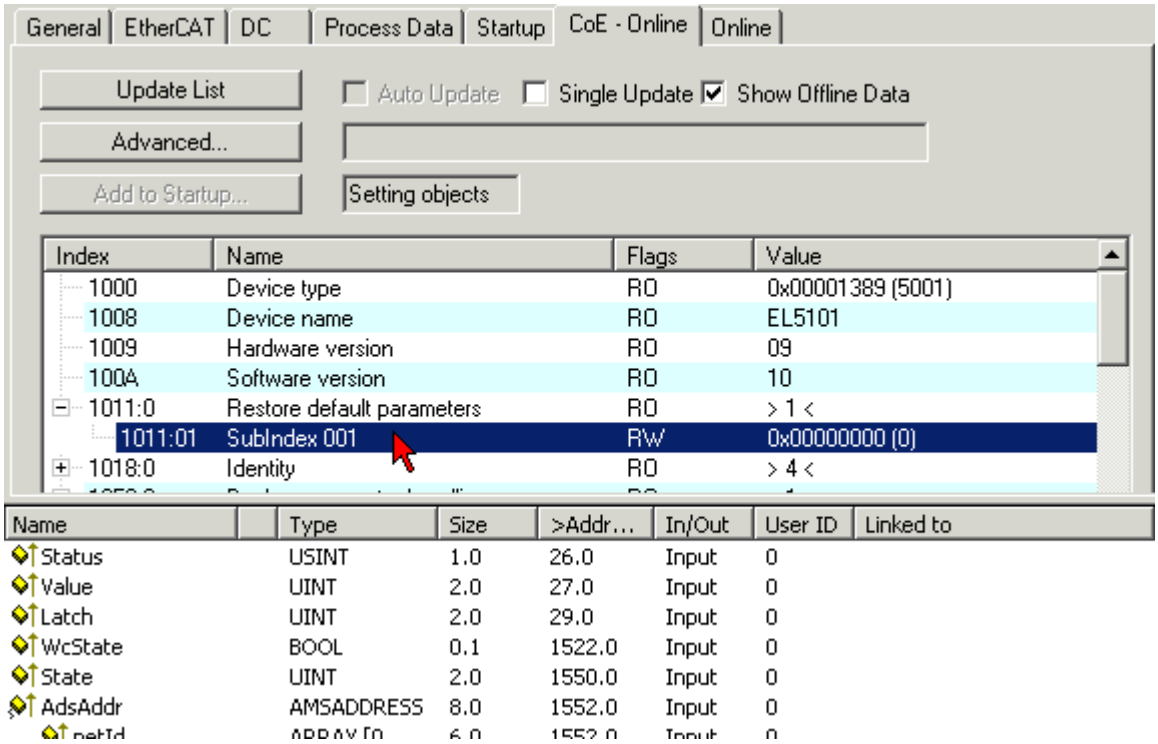


Fig. 75: Selecting the Restore default parameters PDO

Double-click on *SubIndex 001* to enter the Set Value dialog. Enter the value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* and confirm with OK.

All backup objects are reset to the delivery state.

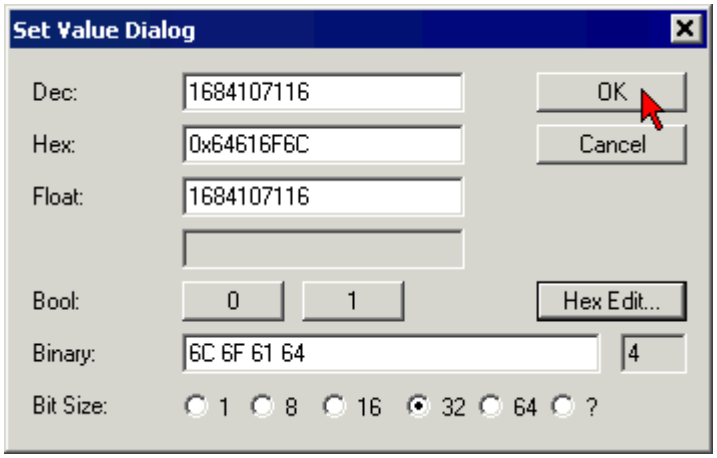


Fig. 76: Entering a restore value in the Set Value dialog

Alternative restore value

In some older terminals / boxes the backup objects can be switched with an alternative restore value:
Decimal value: 1819238756
Hexadecimal value: 0x6C6F6164

An incorrect entry for the restore value has no effect.

6 Appendix

6.1 General operating conditions

Protection degrees (IP-Code)

The standard IEC 60529 (DIN EN 60529) defines the degrees of protection in different classes.

1. Number: dust protection and touch guard	Definition
0	Non-protected
1	Protected against access to hazardous parts with the back of a hand. Protected against solid foreign objects of Ø 50 mm
2	Protected against access to hazardous parts with a finger. Protected against solid foreign objects of Ø 12.5 mm.
3	Protected against access to hazardous parts with a tool. Protected against solid foreign objects Ø 2.5 mm.
4	Protected against access to hazardous parts with a wire. Protected against solid foreign objects Ø 1 mm.
5	Protected against access to hazardous parts with a wire. Dust-protected. Intrusion of dust is not totally prevented, but dust shall not penetrate in a quantity to interfere with satisfactory operation of the device or to impair safety.
6	Protected against access to hazardous parts with a wire. Dust-tight. No intrusion of dust.

2. Number: water* protection	Definition
0	Non-protected
1	Protected against water drops
2	Protected against water drops when enclosure tilted up to 15°.
3	Protected against spraying water. Water sprayed at an angle up to 60° on either side of the vertical shall have no harmful effects.
4	Protected against splashing water. Water splashed against the disclosure from any direction shall have no harmful effects
5	Protected against water jets
6	Protected against powerful water jets
7	Protected against the effects of temporary immersion in water. Intrusion of water in quantities causing harmful effects shall not be possible when the enclosure is temporarily immersed in water for 30 min. in 1 m depth.

*) These protection classes define only protection against water!

Chemical Resistance

The Resistance relates to the Housing of the Fieldbus/EtherCAT Box and the used metal parts. In the table below you will find some typical resistance.

Character	Resistance
Steam	at temperatures >100°C: not resistant
Sodium base liquor (ph-Value > 12)	at room temperature: resistant > 40°C: not resistant
Acetic acid	not resistant
Argon (technical clean)	resistant

Key

- resistant: Lifetime several months
- non inherently resistant: Lifetime several weeks
- not resistant: Lifetime several hours resp. early decomposition

6.2 EtherCAT Box- / EtherCAT P Box - Accessories

Fixing

Ordering information	Description
ZS5300-0001	Mounting rail (500 mm x 129 mm)

Marking material, plugs

Ordering information	Description
ZS5000-0000	Fieldbus Box set M8 (contact labels, plugs)
ZS5000-0002	Fieldbus Box set M12 (contact labels, plugs)
ZS5000-0010	plugs M8, IP67 (50 pieces)
ZS5000-0020	plugs M12, IP67 (50 pieces)
ZS5100-0000	marking labels, not printed, 4 stripes at 10 pieces
ZS5100-xxxx	printed marking labels, on request

Tools

Ordering information	Description
ZB8800	torque wrench for M8 cables with knurl, incl. ratchet
ZB8800-0001	M12 ratchet for torque wrench ZB8800
ZB8800-0002	M8 ratchet (field assembly) for torque wrench ZB8800
ZB8801-0000	torque wrench for hexagonal plugs, adjustable
ZB8801-0001	torque cable key, M8/wrench size 9, for torque wrench ZB8801-0000
ZB8801-0002	torque cable key, M12/wrench size 13, for torque wrench ZB8801-0000
ZB8801-0003	torque cable key, M12 field assembly/wrench size 13, for torque wrench ZB8801-0000

Further accessories

Further accessories may be found at the price list for Beckhoff fieldbus components and at the internet under https://www.beckhoff.com

6.3 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages:
http://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49(0)5246/963-0
Fax:	+49(0)5246/963-198
e-mail:	info@beckhoff.com

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49(0)5246/963-157
Fax:	+49(0)5246/963-9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49(0)5246/963-460
Fax:	+49(0)5246/963-479
e-mail:	service@beckhoff.com

Table of Figures