BECKHOFF New Automation Technology

Documentation | EN

EL600x, EL602x

Serial Interface Terminals



2026-11-09 | Version: 6.2.0

Table of contents

1 Overview Serial Interface Terminals 5

2 Foreword 6

2.1 Notes on the documentation 6
2.2 Guide through documentation 7
2.3 Safety instructions 8
2.4 Documentation Issue Status 9

3 Product description 10

3.1 EL6001, EL6021 14
3.1.1 Introduction 14
3.1.2 Technical data 15
3.2 EL6021-0021 16
3.2.1 Introduction 16
3.2.2 Technical data 17
3.3 EL6002, EL6022 18
3.3.1 Introduction 18
3.3.2 Technical data 19
3.4 Start up 20

4 Basics communication 21

4.1 EtherCAT basics 21
4.2 EtherCAT cabling – wire-bound 21
4.3 General notes for setting the watchdog 23
4.4 EtherCAT State Machine 25
4.5 CoE Interface 27
4.6 Distributed Clock 33

5 Mounting and Wiring 34

5.1 Notes on ESD protection 34
5.2 Explosion protection 35
5.2.1 ATEX - Special conditions (standard temperature range) 35
5.2.2 ATEX - Special conditions (extended temperature range) 36
5.2.3 IECEx - Special conditions 37
5.2.4 Continuative documentation for ATEX and IECEx 38
5.2.5 cFMus - Special conditions 39
5.2.6 Continuative documentation for cFMus 40
5.3 UL notice 41
5.4 Positioning of passive terminals 42
5.5 Installation instructions for enhanced mechanical load capacity 43
5.6 Installation positions 44
5.7 Note on power supply 47
5.8 EL6001, EL6021-00xx 48
5.8.1 Mounting rail installation 48
5.8.2 Connection 51
5.8.3 LEDs and connector assignments 55
5.9 EL6002, EL6022 59

5.9.1 Mounting and demounting - top front unlocking ... 59
5.9.2 LEDs and pin assignment ... 61
5.10 Disposal ... 65

6 Commissioning ... 66
6.1 Operating modes and process data ... 66
6.2 Hints regarding TcVirtualComDriver ... 74
6.3 Communication features ... 76
6.4 LIN Master Feature EL6001 ... 79
6.5 Example programs ... 82
6.5.1 Sample program 1 ... 82
6.5.2 Sample program 2 ... 85
6.5.3 Sample program 3 (LIN) ... 87

7 Overview of CoE objects EL6001, EL6021 ... 90
7.1 Control and status word ... 90
7.2 Object description and parameterization ... 93
7.2.1 Objects for commissioning ... 93
7.2.2 Standard objects (0x1000-0x1FFF) ... 95
7.2.3 Profile-specific objects (0x6000-0xFFFF) [from hardware version 03] ... 111

8 Overview CoE objects EL6002, EL6022 ... 115
8.1 Object description and parameterization ... 115
8.1.1 Objects for commissioning ... 115
8.1.2 Standard objects (0x1000-0x1FFF) ... 116
8.1.3 Profile-specific objects (0x6000-0xFFFF) [from hardware version 03] ... 128
8.2 Control and status data ... 132

9 Appendix ... 134
9.1 EtherCAT AL Status Codes ... 134
9.2 Firmware compatibility ... 135
9.3 Support and Service ... 137

1 Overview Serial Interface Terminals

EL6001 [▶ 14] (1 channel Serial Interface Terminal, RS232C)
EL6021 [▶ 14] (1 channel Serial Interface Terminal, RS422/RS485)
EL6021-0021 [▶ 16] (1 channel Serial Interface Terminal, RS422/RS485), line device
EL6002 [▶ 18] (2 channel Serial Interface Terminal, RS232C)
EL6022 [▶ 18] (2 channel Serial Interface Terminal, RS422/RS485)

2 Foreword

2.1 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.
The qualified personnel is obliged to always use the currently valid documentation.

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®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® 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.

EtherCAT® and EtherCAT P® are registered trademarks and patented technologies, licensed by Beckhoff Automation GmbH, Germany.

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

Third-party brands

Trademarks of third parties may be used in this documentation. You can find the trademark notices here:
https://www.beckhoff.com/trademarks

2.2 Guide through documentation

NOTICE



Further components of documentation

This documentation describes device-specific content. It is part of the modular documentation concept for Beckhoff I/O components. For the use and safe operation of the devices described in this documentation, additional cross-product descriptions are required, which can be found in the following table.

Title	Description
EtherCAT System Documentation (PDF)	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • EtherCAT devices configuration
Explosion Protection for Terminal Systems (PDF)	Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx
Control Drawing I/O, CX, CPX (PDF)	Connection diagrams and Ex markings (conform to cFMus)
Infrastructure for EtherCAT/Ethernet (PDF)	Technical recommendations and notes for design, implementation and testing
Software Declarations I/O (PDF)	Open source software declarations for Beckhoff I/O components

The documentations can be viewed at and downloaded from the Beckhoff website (www.beckhoff.com) via:

- the “Documentation and Download” area of the respective product page,
- the Download finder,
- the Beckhoff Information System.

If you have any suggestions or proposals for our documentation, please send us an e-mail stating the documentation title and version number to: documentation@beckhoff.com

2.3 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.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product



This information includes, for example:
recommendations for action, assistance or further information on the product.

2.4 Documentation Issue Status

Version	Comment
6.2.0	• Update chapter "LEDs and connection" • Update chapter "Process data" • Update structure • Update revision status
6.1.0	• Update chapter "Product description" • Update structure • Update revision status
6.0.0	• Update chapter "Product description" • Update chapter "Technical data" • Update structure • Update revision status
5.9.0	• Update chapter "Commissioning" • Update structure
5.8.0	• Update chapter "LED and connection" • Update structure
5.7	• Update chapter "LED and connection" • Update structure
5.6	• Update chapter "Object description and parameterization" • Update structure • Update revision status
5.5	• EL6002, EL6022 Chapter “Recommended mounting rails” removed • Update chapter "Technical data" • Chapter “ATEX - Special conditions (standard temperature range)” added • Update revision status
5.4	• EL6021-0021 added • Update chapter "Product description" • Update structure • Update revision status
5.3	• Update chapter "Commissioning" • Update structure • Update revision status
5.2	• Update chapter "Object description and parameterization" • Update structure • Update revision status
5.1	• Update chapter "LIN Master Feature EL6001” • Update structure • Update revision status
0.1 – 5.0	*archived*

3 Product description

Technology

The EL600x and EL602x serial interface terminals enable the connection of devices with an RS232 (or RS485 / RS422) interface. In the case of the EL600x, the data is exchanged with the controller in full duplex mode; in the case of the EL602x, half duplex mode is additionally possible. The terminal has one receive buffer an one transmit buffer per channel, see technical data. Data transfer between the terminal and the controller takes place via a handshake.

The factory setting of the terminals is:

- 9600 baud
- 8N1: 8 data bits, 1 stop bit, no parity
- in the EL600x the RTS/CTS control is active
- the EL602x operates in full duplex mode with deactivated point-to-point connection.

Basic principles

During transfer of several bytes of data, the data (x bytes or 8*x bits in total) are sent in individual telegrams containing 7 or 8 bits, based on the coding specification (e.g. 7E2 or 8N1). A telegram consists of:

- Start bit
- Data bits (7 or 8, starting with the LSB [least significant bit])
- Optional: parity bit
  - "E" EVEN: The parity bit is set by the sender such that the parity is even
  - "O" ODD: The parity bit is set by the sender such that the parity is odd
  - "N" NOT: no parity bit
  - "M" MARK: The parity bit is set to 1 by the sender
  - "S" SPACE: The parity bit is set to 0 by the sender
- Stop bit (1 or 2)

Accordingly, the coding specification 8N1 means: 8 data bits, no parity bit, 1 stop bit.

If 7-bit coding is selected, of each data byte that is transferred from the PLC to the terminal via the cyclic process data, only the lower 7 bits are sent. In other words, if 10 bytes of data (consisting of 8 bits) are sent to the EL60xx, 10 telegrams of 7 bits each are sent.

If 9-bit coding is selected, the 16-bit process data interface must be used. The terminal then expects the 9 useful bits in the lower 9 bits of the 16-bit word.

Frequency

The frequency of the data transfer must be known in the sender and receiver and match within a few percent, in order to ensure that the receiver can correctly detect any changes in level on the line.

Handshake

An additional handshake between the sender and the receiver can be used so that the receiver can indicate that it is ready to receive. The EL60xx supports two types of handshake:

- via special RTS/CTS data cables
  - This features must be activated in the CoE.
  - Only possible with EL6001/EL6002.
- via special data telegrams
  - This features must be activated in the CoE.

RS Standards

The term RS232 (RS-232) is used in this document briefly for the ANSI/EIA/TIA-232-F standard.

The term RS422 (RS-422) is used briefly in this document for the ITU-T V.11 standard according to ANSI/TIA/EIA-422-B-1994.

The term RS485 (RS-485) is used in this document briefly for the ANSI/TIA/EIA-485-A-98 standard.

Level interfaces

The EL6001/6002 devices operate at an RS232 level with reference to GND, the EL6021/6022 devices with a differential RS485/422 level.

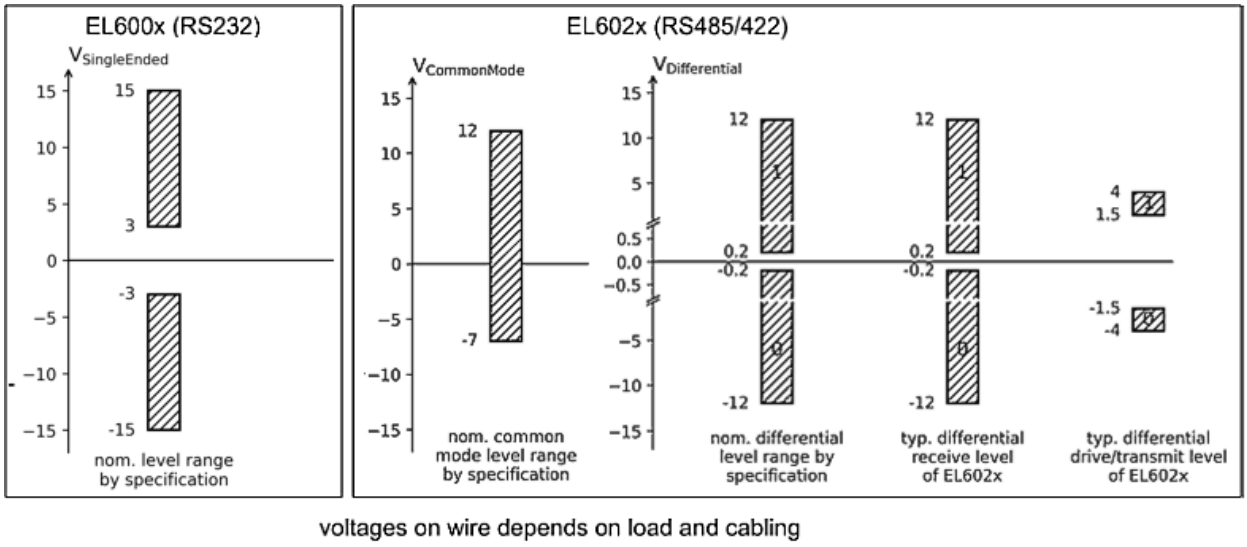


Fig. 1: Level interfaces RS232, RS485/422

Termination and topology

The serial RS422 and RS485 communication technologies operate with voltage levels on a 2-wire line. Reflections at high-resistance line ends can lead to signal distortion. For this reason termination resistors are required at the receiver. For RS422/485 these are 120 Ω resistors, which together with the line resistance result in a voltage drop over the transmission link.

Permitted cable length

The line resistance together with the termination resistor results in an overall voltage drop over the transmission link. An unacceptably high number of termination resistors would result in excessive attenuation of the signal.

The system design should ensure that the voltage does not drop below 200 mV at the receiver (see Fig.), which is the minimum voltage required.

In RS422 mode each line must be terminated with 120 Ω at the receiver.

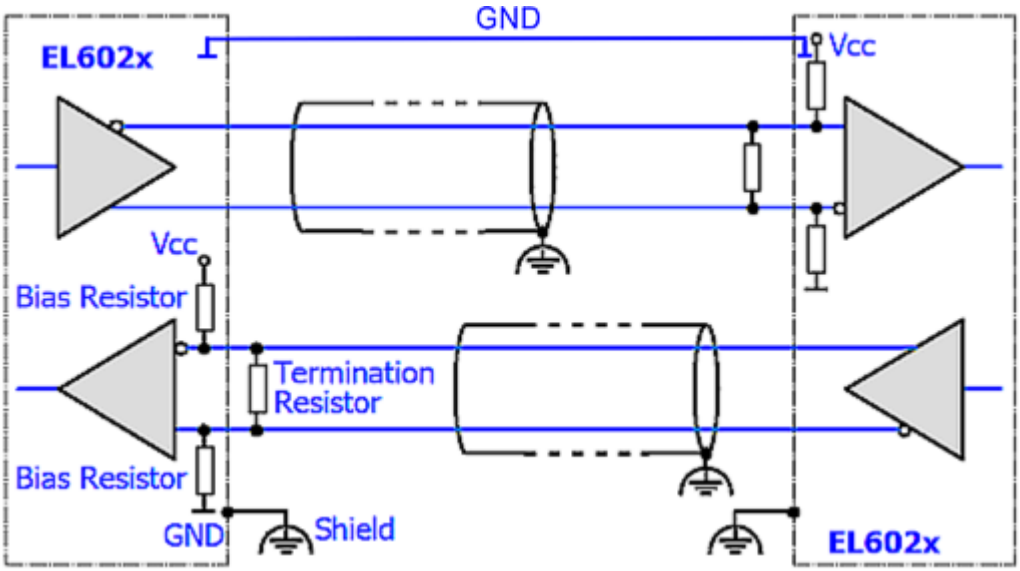


Fig. 2: RS422 termination

In RS485 mode with several devices, termination resistors are only used at the two end devices.

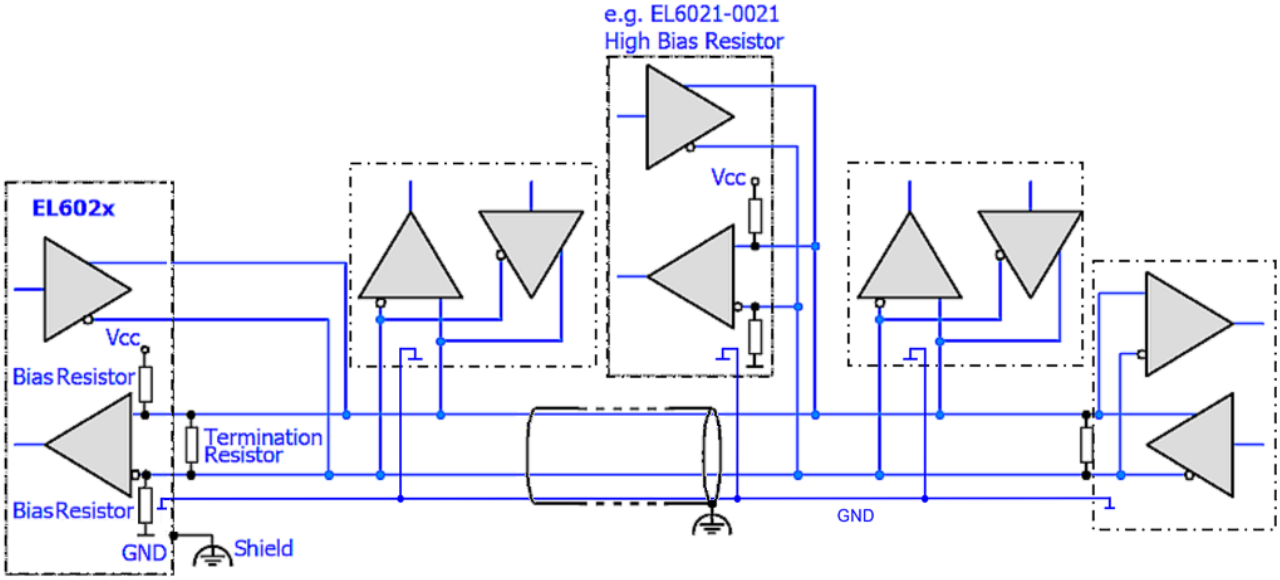


Fig. 3: RS485 termination

The background is the different design of RS422/EIA-422 and RS485/EIA-485:

- RS422: 1 Tx → Rx n (maximum 10 receivers)
- RS485: n Tx → Rx m (maximum 32/128 devices, depending on the resulting bus loading)

Components for RS485 usually have a higher input impedance, resulting in lower bus load.

Termination with EL602x and BIAS resistors

The EL602x devices do not have integrated termination resistors, in order to enable operation in bus mode. Any termination that may be required must be connected outside the terminal.
The EL602x devices feature integrated bias resistors < 1 kOhm, which bring the bus lines to defined levels, even if the line is disconnected.
If several EL602x devices are connected in a bus, the parallel bias resistors may hamper the data communication. In this case the EL6021-0021, which has significantly higher bias resistors, should be used as an intermediate device.

Topology

The termination and the bias resistors generate a load on the bus. However, they are essential for unambiguous bus levels and therefore have to be positioned with diligence. Ideally the RS422/485 bus should be configured as a daisy chain or a simple chain, see Fig. The following topologies may be problematic:

- Star topologies: each end point should ideally be terminated, but this can lead to excessive bus loading and ambiguous signal levels. Other potential issues are reflections and runtime variations.
- Intermeshed topologies: no clear end points, which means reflections and circulating currents are possible.

Shielding/shield

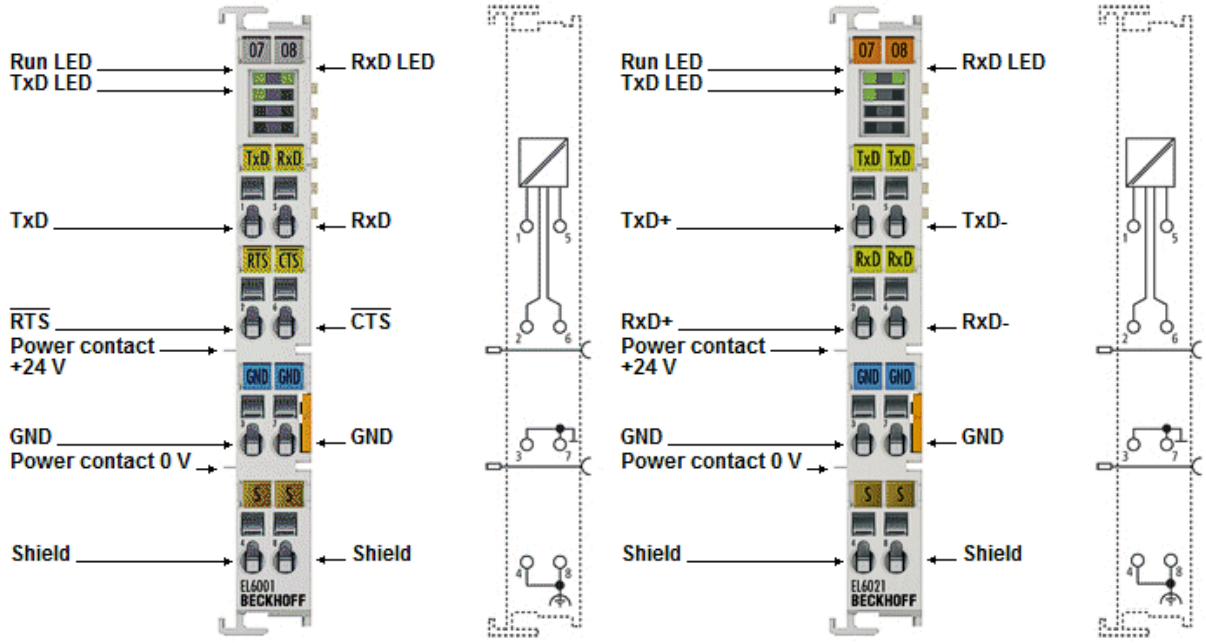
NOTICE

Do not use functional earth for discharge of residual currents or potential differences!

The EL60xx units offer a shielded connection for discharging EMC interference via the cable shield (FE, functional earth). The shield must not be misused for discharging residual currents or potential differences.

3.1 EL6001, EL6021

3.1.1 Introduction



Serial Interface Terminal (RS232C/RS422/RS485), 1 channel

The EL6001 and EL6021 serial interfaces enable the connection of devices with RS-232 or RS422/RS485 interface. The EL6001 operates in conformity with the CCITT V.28/DIN 66 259-1 standards.
The device connected to the EL6001 EtherCAT Terminal communicates with the automation device via the coupler. The active communication channel operates independently of the higher-level bus system in full duplex mode or selectable half duplex mode (EL6021) at up to 115.2 kbaud.
The RS232 interface guarantees high immunity to interference through electrically isolated signals, which is additionally guaranteed for the EL6021 through differential signal transmission.

In conjunction with the TwinCAT Virtual Serial COM Driver (see TwinCAT Supplements - Communication) the EL6001/EL6021 can be used as a normal Windows COM interface.

Quick links

- EtherCAT basics [▶ 21]
- Technology Serial Interface Terminals [▶ 10]
- Commissioning [▶ 66]
- Process data, general notes [▶ 66]
- CoE object description and parameterization EL60x1 [▶ 90]
- Control and status data EL60x1 [▶ 90]

3.1.2 Technical data

Technical data	EL6001	EL6021
Data transfer channels	TxD and RxD, full duplex	TxD and RxD, full/half duplex
Data transfer rate	2400...115200 baud, default: 9600 baud, 8 data bits, no parity, 1 stop bit from firmware 07 [▶ 135]: also 12000 baud and 14400 baud from firmware 11 [▶ 135]: any integer baud rate 1000…115200 Baud	2400...115200 baud, default: 9600 baud, 8 data bits, no parity, 1 stop bit
Data buffer	864 byte receive buffer, 128 byte transmit buffer EL6001 from FW08: 250 byte transmit buffer	
Bit transfer	-	with differential signal
Level interface	RS232	RS485/422
Bit distortion	< 3 %	-
Cable length	max. 15 m	max. 1000 m (Twisted Pair)
Line impedance	-	120 Ω
Providing external supply	-	-
Diagnosis	Status LEDs	
Power supply	via the E-bus	
Current consumption via E-bus	typ. 120 mA	typ. 170 mA (in case of short circuit: typ. 250 mA)
Electrical isolation	500 V (E-bus/RS232C)	500 V (E-bus/RS422, E-bus/RS485)
Bit width in process image	1 x 8 bit Control/Status, Inputs/Outputs: 3 x 8 bit user data or 1 x 8 bit Control/Status, Inputs/Outputs: 5 x 8 bit user data or 1 x 16 bit Control/Status, Inputs/Outputs: 22 x 8 bit user data (configurable)	
Configuration	no address setting required configuration via TwinCAT System Manager	
Weight	approx. 55 g	
Permissible ambient temperature range during operation	-25°C ... +60°C	
Permissible ambient temperature range during storage	-40°C ... +85°C	
Permissible relative humidity	95%, no condensation	
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)	
Mounting [▶ 48]	on 35 mm mounting rail conforms to EN 60715	
Enhanced mechanical load capacity	yes, see also installation instructions [▶ 43] for enhanced mechanical load capacity	
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	IP20	
Installation position	variable	
Approvals / markings*	CE, cULus [▶ 41], UKCA, EAC, ATEX [▶ 36], IECEx [▶ 37], DNV	CE, cULus [▶ 41], UKCA, EAC, ATEX [▶ 36], IECEx [▶ 37], cFMus [▶ 39], DNV

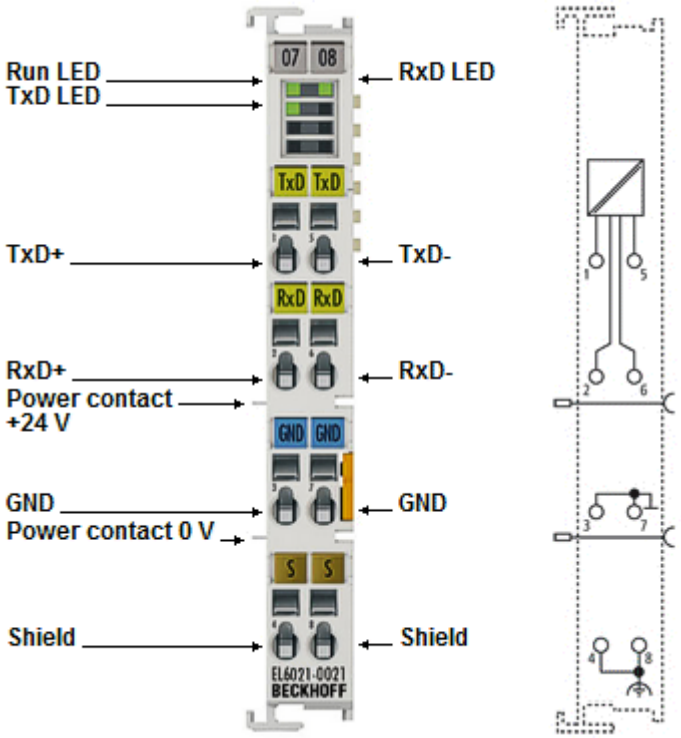
*) Real applicable approvals/markings see type plate on the side (product marking).

Ex markings

Standard	Marking
ATEX	II 3 G Ex ec IIC T4 Gc
IECEx	Ex ec IIC T4 Gc
cFMus	Class I, Division 2, Groups A, B, C, D Class I, Zone 2, AEx/Ex ec IIC T4 Gc

3.2 EL6021-0021

3.2.1 Introduction



Serial Interface Terminal (RS232C/RS422/RS485), 1 channel, line device

The EL6021-0021 serial interface enables the connection of devices with RS422 or RS485 interface. These devices communicate with the automation device via the coupler. The active communication channel operates independently of the higher-level bus system in full or half duplex mode at up to 115.2 kbaud. The transmission of differential signals according to RS422 guarantees high immunity to interference through electrically isolated signals.

If several EL602x are interconnected in a bus, the EL6021-0021 with its enlarged bias resistors [▶ 11] can be used for intermediate devices.

In conjunction with the TwinCAT Virtual Serial COM Driver (see TwinCAT Supplements – Communication) the EL6021-0021 can be used as a normal Windows COM interface.

Quick links

- EtherCAT basics [▶ 21]
- Technology Serial Interface Terminals [▶ 10]
- Commissioning [▶ 66]
- Process data, general notes [▶ 66]
- CoE object description and parameterization EL60x1 [▶ 90]
- Control and status data EL60x1 [▶ 90]

3.2.2 Technical data

Technical data	EL6021-0021
Data transfer channels	TxD and RxD, full/half duplex
Data transfer rate	2400...115200 baud, default: 9600 baud, 8 data bits, no parity, 1 stop bit
Data buffer	864 byte receive buffer, 128 byte transmit buffer EL6001 from FW08: 250 byte transmit buffer
Special features	enlarged bias resistors [▶ 11]
Bit transfer	with differential signal
Level interface	RS485/422
Bit distortion	-
Cable length	max. 1000 m (Twisted Pair)
Line impedance	120 Ω
Providing external supply	-
Diagnosis	Status LEDs
Power supply	via the E-bus
Current consumption via E-bus	typ. 170 mA (in case of short circuit: typ. 250 mA)
Electrical isolation	500 V (E-bus/RS422, E-bus/RS485)
Bit width in process image	1 x 8 bit Control/Status, Inputs/Outputs: 3 x 8 bit user data or 1 x 8 bit Control/Status, Inputs/Outputs: 5 x 8 bit user data or 1 x 16 bit Control/Status, Inputs/Outputs: 22 x 8 bit user data (configurable)
Configuration	no address setting required configuration via TwinCAT System Manager
Weight	approx. 55 g
Permissible ambient temperature range during operation	-25°C ... +60°C
Permissible ambient temperature range during storage	-40°C ... +85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting [▶ 48]	on 35 mm mounting rail conforms to EN 60715
Enhanced mechanical load capacity	Yes, see installation instructions [▶ 43] for enhanced mechanical load capacity
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	IP20
Installation position	variable
Approvals / markings*	CE, cULus [▶ 41], UKCA, EAC, ATEX [▶ 35], IECEx [▶ 37]

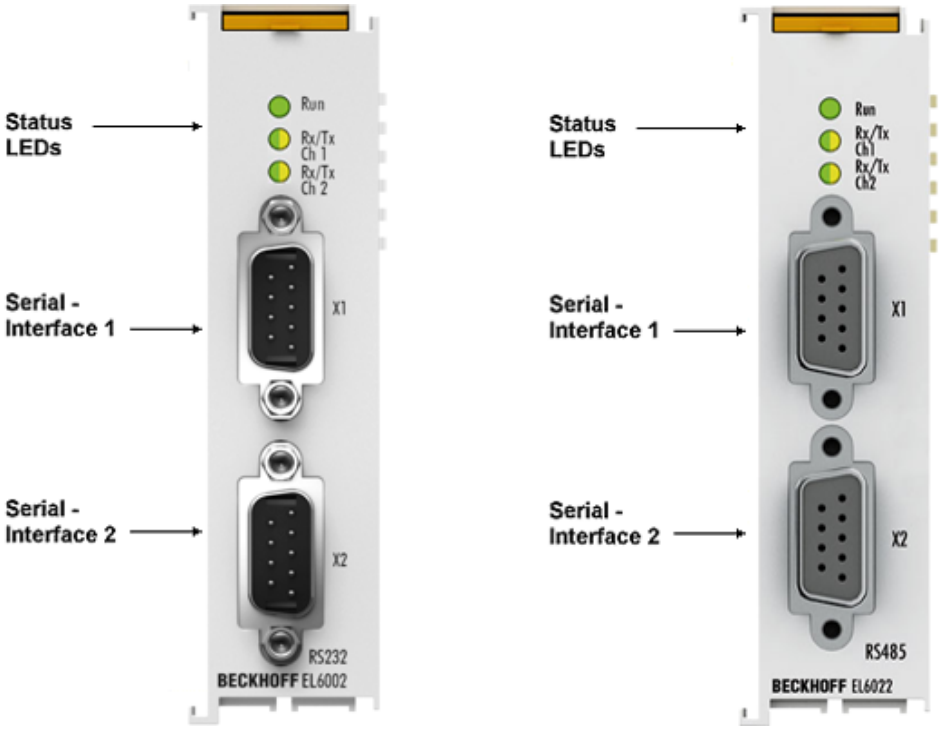
*) Real applicable approvals/markings see type plate on the side (product marking).

Ex markings

Standard	Marking
ATEX	II 3 G Ex ec IIC T4 Gc
IECEx	Ex ec IIC T4 Gc

3.3 EL6002, EL6022

3.3.1 Introduction



Serial Interface Terminal (RS232C/RS422/RS485), 2 channel

The EL6002 and EL6022 serial interfaces enable the connection of devices with two RS232 or two RS422/RS485 interfaces each with one D-Sub connector (9 pin). The interfaces are electrically isolated from each other and from the EtherCAT.
The devices connected to the EL6002/EL6022 EtherCAT Terminals communicate with the automation device via the Coupler. The active communication channel operates independently of the higher-level EtherCAT system in full duplex mode with 300 baud up to 115.2 kbaud.
The RS232/RS422/RS485 interfaces guarantee high interference immunity through electrically isolated signals. The EL6022 can provide 2 x 5 V/20 mA from the E-bus supply (electrically isolated, short-circuit-proof) as supply for external devices.

In conjunction with the TwinCAT Virtual Serial COM Driver, the EL60xx can be used as a normal Windows COM interface.

Quick links

- EtherCAT basics [▶ 21]
- Technology Serial Interface Terminals [▶ 10]
- Commissioning [▶ 66]
- Process data, general notes [▶ 66]
- CoE object description and parameterization EL60x1 [▶ 115]
- Control and status data EL60x1 [▶ 132]

3.3.2 Technical data

Technical data	EL6002	EL6022
Data transfer channels	2, TxD and RxD, full duplex	2, TXD and RXD, full/half duplex
Connection	2 x D-sub connector (DE9), 9-pin	2 x D-sub connector (DE9), 9-pin
Data transfer rate	300...115200 baud default: 9600 baud, 8 data bits, no parity, 1 stop bit	
Data buffer	864 byte receive buffer, 128 byte transmit buffer per channel	
Level interface	RS232	RS485/422
Cable length	max. 15 m	max. 1000 m (Twisted Pair)
Providing external supply	-	2x typ. 5 V (± 20%), from E-bus supply (electrically isolated), max. 20 mA, short-circuit-proof
Diagnosis	Status LEDs	
Power supply	via the E-bus	
Current consumption via E-bus	typ. 170 mA	typ. 250 mA (in case of short circuit: typ. 250 mA)
Electrical isolation	500 V (E-bus/RS232C)	500 V (E-bus/RS422, E-bus/RS485)
Bit width in process image	1 x 16 bit Control/Status, Inputs/Outputs: 22 x 8 bit user data	
Configuration	no address setting required configuration via TwinCAT System Manager	
Permissible ambient temperature range during operation	-25°C ... +60°C	
Permissible ambient temperature range during storage	-40°C ... +85°C	
Permissible relative humidity	95%, no condensation	
Weight	approx. 70 g	
Dimensions (W x H x D)	approx. 26 mm x 100 mm x 52 mm (width aligned: 23 mm)	
Mounting [▶ 59]	on 35 mm mounting rail conforms to EN 60715	
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	IP20	
Installation position	variable	
Approvals / markings*	CE, cULus [▶ 41], UKCA, EAC, ATEX [▶ 36], IECEx, [▶ 37]	

*) Real applicable approvals/markings see type plate on the side (product marking).

Ex markings

Standard	Marking
ATEX	II 3 G Ex ec IIC T4 Gc
IECEx	Ex ec IIC T4 Gc

3.4 Start up

Start

For commissioning:

- mount the EL600x / EL602x as described in the chapter Mounting and wiring [▶ 34]
- configure the EL600x / EL602x in TwinCAT as described in the chapter Commissioning [▶ 66]

4 Basics communication

4.1 EtherCAT basics

Please refer to the EtherCAT System Documentation for the EtherCAT fieldbus basics.

4.2 EtherCAT cabling – wire-bound

The cable length between two EtherCAT devices must not exceed 100 m. This results from the FastEthernet technology, which, above all for reasons of signal attenuation over the length of the cable, allows a maximum link length of 5 + 90 + 5 m if cables with appropriate properties are used. See also the Design recommendations for the infrastructure for EtherCAT/Ethernet.

Cables and connectors

For connecting EtherCAT devices only Ethernet connections (cables + plugs) that meet the requirements of at least category 5 (CAt5) according to EN 50173 or ISO/IEC 11801 should be used. EtherCAT uses 4 wires for signal transfer.

EtherCAT uses RJ45 plug connectors, for example. The pin assignment is compatible with the Ethernet standard (ISO/IEC 8802-3).

Pin	Color of conductor	Signal	Description
1	yellow	TD +	Transmission Data +
2	orange	TD -	Transmission Data -
3	white	RD +	Receiver Data +
6	blue	RD -	Receiver Data -

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

Recommended cables

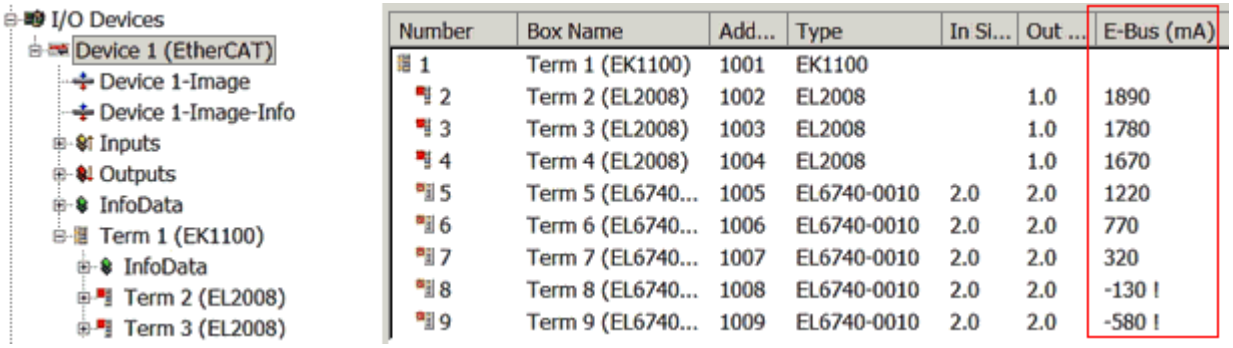
It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
- RJ45 connector, field assembly ZS1090-0005
- EtherCAT cable, field assembly ZB9010, ZB9020

Suitable cables for the connection of EtherCAT devices can be found on the Beckhoff website!

E-Bus supply

A bus coupler can supply the EL terminals added to it with the E-bus system voltage of 5 V; a coupler is thereby loadable up to 2 A as a rule (see details in respective device documentation).
Information on how much current each EL terminal requires from the E-bus supply is available online and in the catalogue. If the added terminals require more current than the coupler can supply, then power feed terminals (e.g. EL9410) must be inserted at appropriate places in the terminal strand.

The pre-calculated theoretical maximum E-Bus current is displayed in the TwinCAT System Manager. A shortfall is marked by a negative total amount and an exclamation mark; a power feed terminal is to be placed before such a position.



Number	Box Name	Add...	Type	In Si...	Out ...	E-Bus (mA)
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL2008)	1002	EL2008		1.0	1890
3	Term 3 (EL2008)	1003	EL2008		1.0	1780
4	Term 4 (EL2008)	1004	EL2008		1.0	1670
5	Term 5 (EL6740...	1005	EL6740-0010	2.0	2.0	1220
6	Term 6 (EL6740...	1006	EL6740-0010	2.0	2.0	770
7	Term 7 (EL6740...	1007	EL6740-0010	2.0	2.0	320
8	Term 8 (EL6740...	1008	EL6740-0010	2.0	2.0	-130 !
9	Term 9 (EL6740...	1009	EL6740-0010	2.0	2.0	-580 !

Fig. 4: System manager current calculation

NOTICE

Malfunction possible!

The same ground potential must be used for the E-Bus supply of all EtherCAT terminals in a terminal block!

4.3 General notes for setting the watchdog

The EtherCAT terminals are equipped with a safety device (watchdog) which, e. g. in the event of interrupted process data traffic, switches the outputs (if present) to a presettable state after a presettable time, depending on the device and setting, e. g. to FALSE (off) or an output value.

The EtherCAT slave controller features two watchdogs:

- Sync Manager (SM) watchdog (default: 100 ms)
- Process Data (PDI) watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:

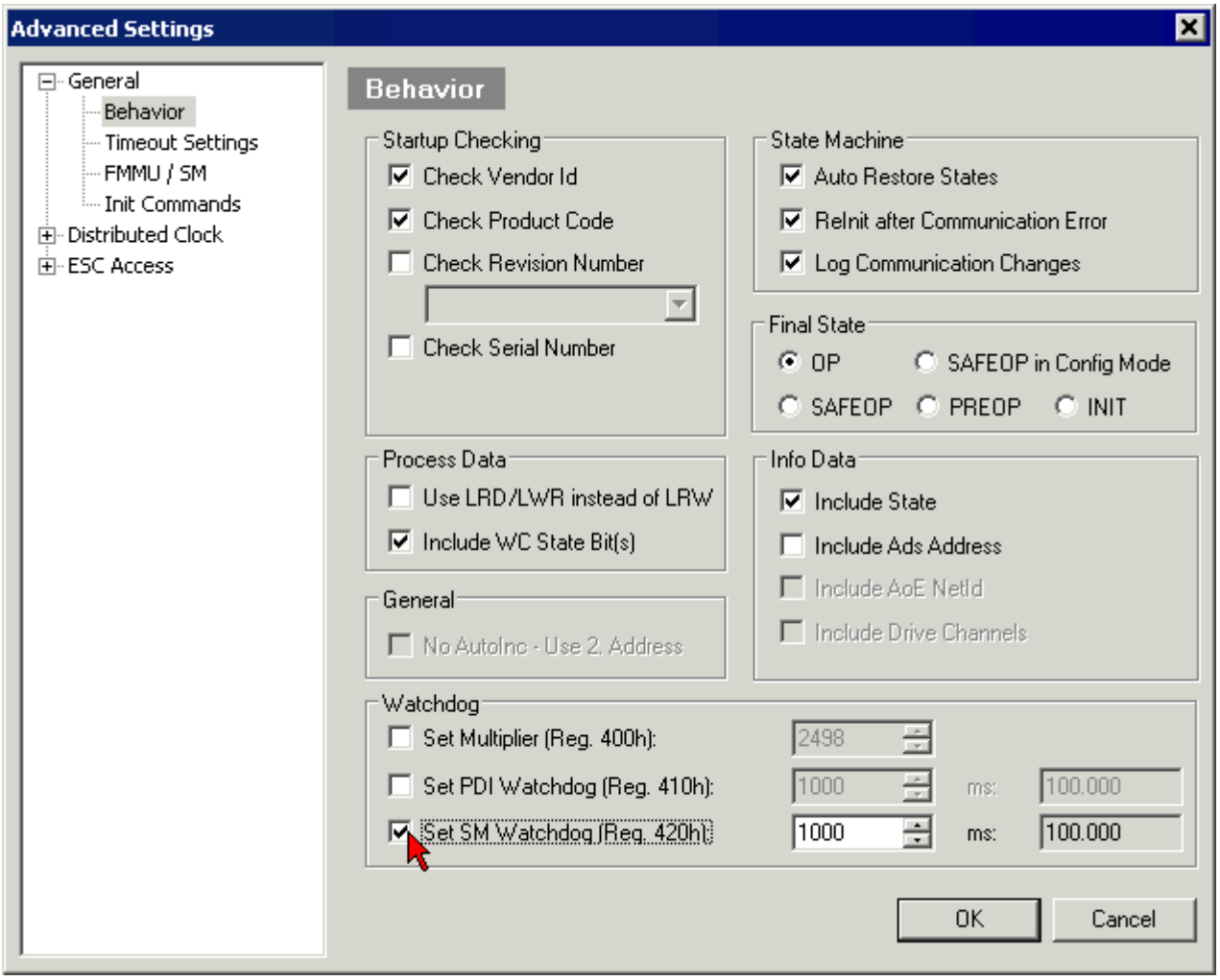


Fig. 5: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. 0x0400) is valid for both watchdogs.
- each watchdog has its own timer setting 410h or 420h, which together with the Multiplier results in a resulting time.
- important: the Multiplier/Timer setting is only loaded into the slave at EtherCAT startup if the checkbox in front of it is activated.
- if it is not checked, nothing is downloaded and the setting located in the ESC remains unchanged.
- the downloaded values can be seen in the ESC registers 400h, 410h and 420h: ESC Access -> Memory

SM watchdog (Sync Manager Watchdog)

The Sync Manager watchdog is reset with each successful EtherCAT process data communication with the terminal. If, for example, no EtherCAT process data communication with the terminal takes place for longer than the set and activated SM watchdog time due to a line interruption, the watchdog is triggered. The status of the terminal (usually OP) remains unaffected. The watchdog is only reset again by a successful EtherCAT process data access.

The Sync Manager watchdog is therefore a monitoring for correct and timely process data communication with the ESC from the EtherCAT side.

The maximum possible watchdog time depends on the device. For example, for "simple" EtherCAT slaves (without firmware) with watchdog execution in the ESC it is usually up to 170 seconds. For complex EtherCAT slaves (with firmware) the SM watchdog function is usually parameterized via register 400h/420h but executed by the microcontroller (µC) and can be significantly lower. In addition, the execution may then be subject to a certain time uncertainty. Since the TwinCAT dialog may allow inputs up to 65535, a test of the desired watchdog time is recommended.

PDI watchdog (Process Data Watchdog)

If there is no PDI communication with the ESC for longer than the set and activated Process Data Interface (PDI) watchdog time, this watchdog is triggered.

The PDI is the internal interface of the ESC, e.g. to local processors in the EtherCAT slave. With the PDI watchdog this communication can be monitored for failure.

The PDI watchdog is therefore a monitoring for correct and timely process data communication with the ESC, but viewed from the application side.

Calculation

Watchdog time = [1/25 MHz * (Watchdog multiplier + 2)] * SM/PDI watchdog

Example: default setting Multiplier = 2498, SM watchdog = 1000 => 100 ms

The value in “Watchdog multiplier + 2” in the formula above corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ CAUTION

Undefined state possible!

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

⚠ CAUTION

Damage of devices and undefined state possible!

If the SM watchdog is activated and a value of 0 is entered the watchdog switches off completely. This is the deactivation of the watchdog! Set outputs are NOT set in a safe state if the communication is interrupted.

4.4 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
- Operational
- Bootstrap

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

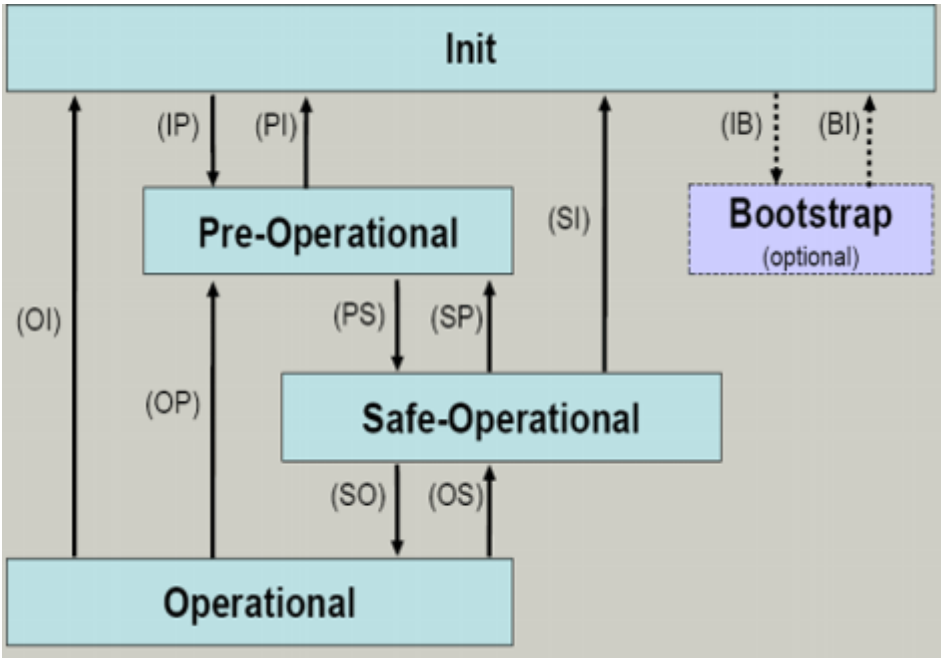


Fig. 6: States of the EtherCAT State Machine

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 Fieldbus Memory Management Unit (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 Dual Port (DP)-RAM areas of the ESC.

In *Safe-Op* state mailbox and process data communication is possible, although the slave keeps its outputs in a safe state, while the input data are updated cyclically.

Outputs in SAFEOP state

The default set watchdog monitoring sets the outputs of the ESC module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

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.

4.5 CoE Interface

General description

The CoE interface (CAN application protocol over EtherCAT interface) 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 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 attributes.

Different CoE data 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.

The order is specified in two levels via hexadecimal numbering: (main)index, followed by subindex.

The value ranges are

- Index: 0x0000 …0xFFFF ($0...65535_{dec}$)
- Subindex: 0x00…0xFF ($0...255_{dec}$)

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: here are the channel parameters for some EtherCAT devices. Historically, this was the first parameter area before the 0x8000 area was introduced. EtherCAT devices that were previously equipped with parameters in 0x4000 and changed to 0x8000 support both ranges for compatibility reasons and mirror internally.
- 0x6000: Input PDOs (“inputs” from the perspective of the EtherCAT master)
- 0x7000: Output PDOs (“outputs” 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:

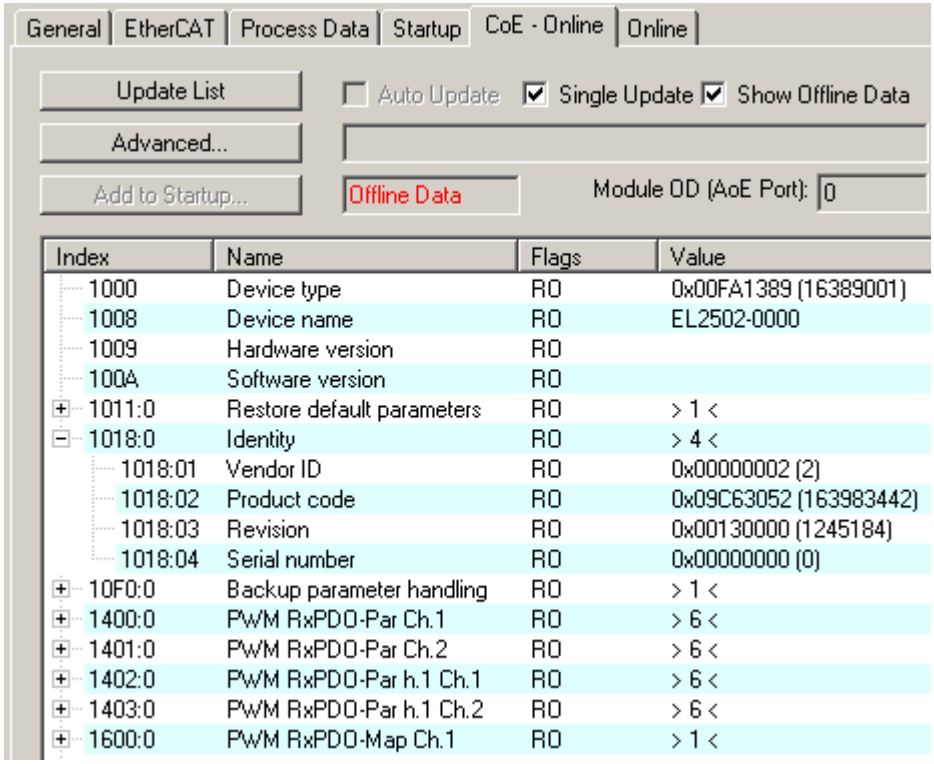


Fig. 7: “CoE Online” tab

The figure “’CoE Online’ tab” shows the CoE objects available in device “EL2502”, ranging from 0x1000 to 0x1600. The subindices for 0x1018 are expanded.

NOTICE

Changes in the CoE directory (CAN over EtherCAT directory), program access

When using/manipulating the CoE parameters observe the general CoE notes in chapter "CoE interface" of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the Beckhoff website),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see TwinCAT 3 | PLC Library: "Tc2_EtherCAT" and Example program R/W CoE)

Data management and function “NoCoeStorage”

Some parameters, particularly the setting parameters of the slave, are configurable and writeable,

- via the System Manager (Fig. “CoE Online” tab) by clicking.
  This is useful for commissioning of the system or slaves. Click on the row of the index to be parameterized and enter a value in the “SetValue” dialog.
- from the control system or PLC via ADS, e.g. through 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 slave CoE parameters are modified online, Beckhoff devices store any changes in a fail-safe manner in the EEPROM, i.e. the modified CoE parameters are still available after a restart.
The situation may be different with other manufacturers.

An EEPROM is subject to a limited lifetime with respect to write operations. From typically 100,000 write operations onwards it can no longer be guaranteed that new (changed) data are reliably saved or are still readable. This is irrelevant for normal commissioning. However, if CoE parameters are continuously changed via ADS at machine runtime, it is quite possible for the lifetime limit to be reached. Support for the NoCoeStorage function, which suppresses the saving of changed CoE values, depends on the firmware version.
Please refer to the technical data in this documentation as to whether this applies to the respective device.

- If the function is supported: the function is activated by entering the code word 0x12345678 once in CoE index 0xF008 and remains active as long as the code word is not changed. After switching the device on it is then inactive. Changed CoE values are not saved in the EEPROM and can thus be changed any number of times.
- If the function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

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 default 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.

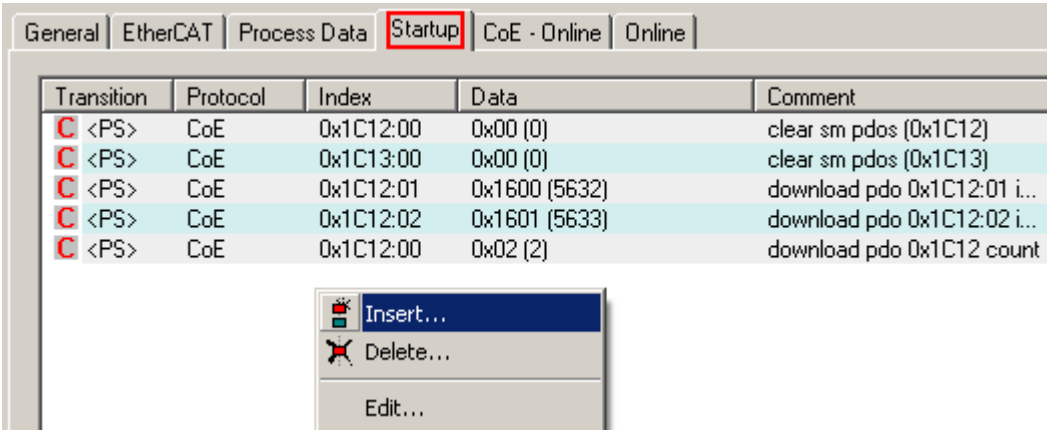


Fig. 8: Startup list in the TwinCAT System Manager

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

Online / offline list

When working with the TwinCAT System Manager, a distinction must be made as to whether the EtherCAT device is currently "available", i.e. switched on and connected via EtherCAT - i.e. **online** - or whether a configuration is created **offline** without slaves being connected.

In both cases a CoE list as shown in Fig. “CoE online tab” is displayed. The connectivity is shown 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 Data** is shown in red.

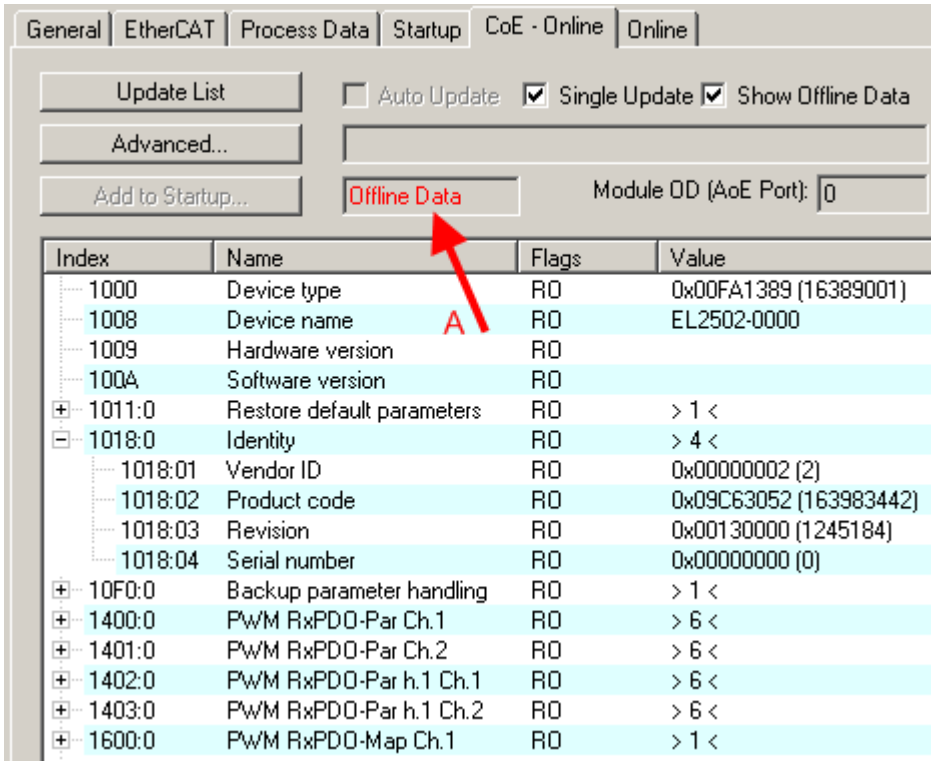


Fig. 9: Offline list

- If the slave is online:
  - The actual current slave list is read. This may take several seconds, depending on the size and cycle time.
  - The actual identity is displayed.
  - The firmware and hardware status of the device is displayed in the CoE.
  - **Online Data** is shown in green.

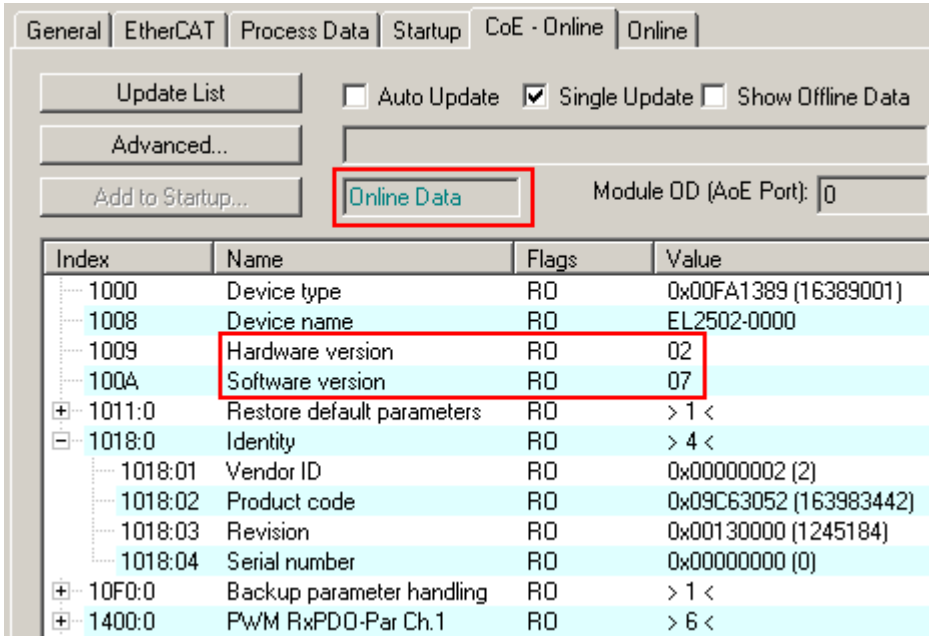


Fig. 10: Online list

Channel-based order

The CoE list is available in EtherCAT devices that usually feature several functionally equivalent channels, for example, a 4-channel analog input terminal also has four logical channels and therefore four 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} or 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.

BackUp objects and checksum 0x10F0:01

The following object types, among others, are defined for the CoE parameter/object directory with regard to information retention, but they do not all have to appear in every device at the same time:

- Vendor objects
  - Are stored persistently (fail-safe) in the device.
  - Technically with the property ReadWrite (RW).
  - Can only be modified/deleted if the corresponding vendor password is known.
  - Used for vendor-specific adjustment or identity data.
- BackUp objects
  - These are objects that are stored persistently in the device, even after changes.
  - Technically with the property ReadWrite (RW).
  - They can be modified/deleted at any time via CoE access from the EtherCAT master

- In particular, they are reset to their initial state, which is stored permanently in the firmware, by selecting "Restore Default Parameters" (see chapter "Restoring the delivery state"). Since this reset to a previous value resembles the restoration of a backup, these are referred to as “BackUp objects”.
- They are used for regular function parameters of the device that determine the behavior.

- BackUp objects with write protection option.
  - The same applies as for the BackUp objects.
  - In addition, the user can activate write protection for these objects using a code word in xF009, thereby preventing accidental changes. Details can be found in the device documentation of the devices that contain these objects.
- Volatile objects
  - These are objects that are not stored persistently in the device.
  - They are used to display internal information (process data, states, temperatures ...) and are available as ReadOnly (RO) or as function parameters (ReadWrite). However, the latter must be written by the EtherCAT master for each PowerOn if they are to have a value other than the default value.

The device indicates a 16-bit CRC in the 32-bit object 0x10F0:01 CheckSum, subindex 01 of Backup Parameter Handling by means of the current state of the so-called BackUp objects:

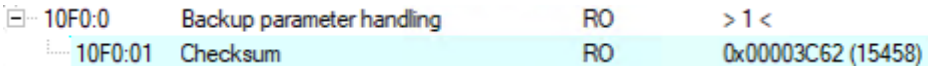


Fig. 11: CoE index 10F0

If a BackUp object is changed, the firmware calculates a new checksum accordingly. This can be used to detect changes to the BackUp objects.

Note: The initial value of the checksum may vary depending on the firmware version, as function extensions can add additional objects to the CRC’s detection range.

4.6 Distributed Clock

The distributed clock represents a local clock in the EtherCAT slave controller (ESC) with the following characteristics:

- Unit *1 ns*
- Zero point *1.1.2000 00:00*
- Size *64 bit* (sufficient for the next 584 years; however, some EtherCAT slaves only offer 32-bit support, i.e. the variable overflows after approx. 4.2 seconds)
- The EtherCAT master automatically synchronizes the local clock with the master clock in the EtherCAT bus with a precision of < 100 ns.

For detailed information please refer to the EtherCAT system description.

5 Mounting and Wiring

5.1 Notes on ESD protection

NOTICE

Devices can be destroyed by electrostatic charging!

The devices contain electrostatically sensitive components which can be damaged by improper handling.

- When handling the components, take care to avoid electrostatic discharge; also, prevent direct contact with the spring-loaded contacts (see illustration).
- Contact with highly insulating materials (synthetic fibers, plastic films, etc.) should be avoided when simultaneously handling components.
- When handling the components, ensure proper earthing of the environment (workplace, packaging, and persons).
- Each bus station must be terminated with the EL9011 or EL9012 end cap on its right side to guarantee the protection rating and ESD protection.

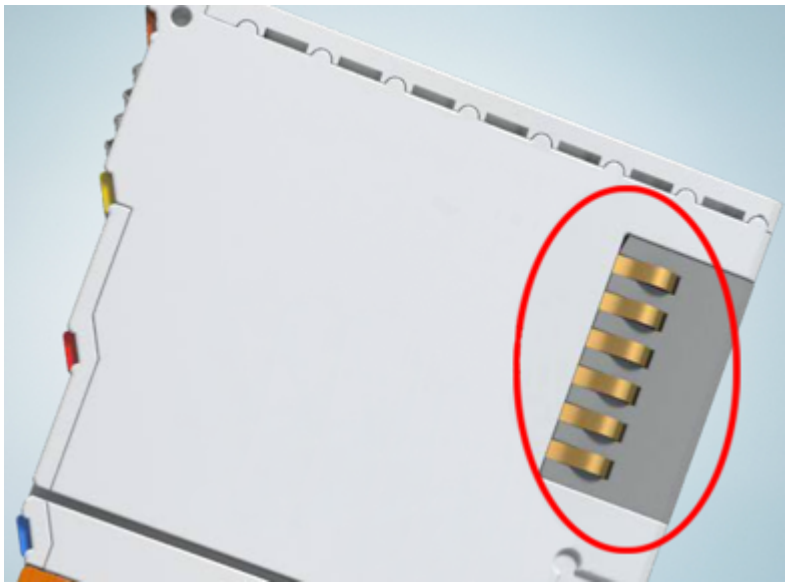


Fig. 12: Spring contacts of Beckhoff I/O components

5.2 Explosion protection

5.2.1 ATEX - Special conditions (standard temperature range)

⚠ WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components with standard temperature range in potentially explosive areas!

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60079-7! The environmental conditions during use are thereby to be taken into account!
- 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!
- Observe the permissible ambient temperature range of 0 to 55°C for the use of Beckhoff fieldbus components standard temperature range 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 individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- 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!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted 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 IEC 60079-0:2018
- EN 60079-7:2015+A1:2018

Marking

The Beckhoff fieldbus components with standard temperature range certified according to regulations for potentially explosive areas bear the following marking:



II 3 G Ex ec IIC T4 Gc
KEMA 10 ATEX0075 X
DEKRA 22UKEX6024X
Ta: 0 ... + 55°C

5.2.2 ATEX - Special conditions (extended temperature range)

⚠ WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components with extended temperature range (ET) in potentially explosive areas!

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60079-7! The environmental conditions during use are thereby to be taken into account!
- 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!
- Observe the permissible ambient temperature range of -25 to 60°C for the use of Beckhoff fieldbus components with extended temperature range (ET) 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 individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- 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!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted 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 IEC 60079-0:2018
- EN 60079-7:2015+A1:2018

Marking

The Beckhoff fieldbus components with extended temperature range (ET) certified according to regulations for potentially explosive areas bear the following marking:



II 3 G Ex ec IIC T4 Gc
KEMA 10 ATEX0075 X
DEKRA 22UKEX6024X
Ta: -25 ... + 60°C

5.2.3 IECEx - Special conditions

⚠ WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components in potentially explosive areas!

- For gas: The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to IEC 60079-7, taking into account the environmental conditions under which the equipment is used!
- The equipment shall only be used in an area of at least pollution degree 2, as defined in IEC 60664-1!
- Provisions shall be made to prevent the rated voltage from being exceeded by transient disturbances of more than 119 V!
- 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!
- Observe the permissible ambient temperature range for the use of Beckhoff fieldbus components in potentially explosive areas!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- 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!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The front hatch of certified units may only be opened if the supply voltage has been switched off or a non-explosive atmosphere is ensured!

Standards

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

- IEC 60079-0:2017 (Edition 7.0)
- IEC 60079-7:2017 (Edition 5.1)

Marking

Beckhoff fieldbus components that are certified in accordance with IECEx for use in areas subject to an explosion hazard bear the following marking:

IECEx DEK 16.0078 X

Ex ec IIC T4 Gc

5.2.4 Continuative documentation for ATEX and IECEx

NOTICE	
	**Continuative documentation about explosion protection according to ATEX and IECEx** Pay also attention to the continuative documentation **Ex. Protection for Terminal Systems** Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx, that is available for download within the download area of your product on the Beckhoff homepage www.beckhoff.com!

5.2.5 cFMus - Special conditions

⚠ WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components in potentially explosive areas!

- The equipment shall be installed within an enclosure that provides a minimum ingress protection of IP54 in accordance with ANSI/UL 60079-0 (US) or CSA C22.2 No. 60079-0 (Canada).
- The equipment shall only be used in an area of at least pollution degree 2, as defined in IEC 60664-1.
- Transient protection shall be provided that is set at a level not exceeding 140% of the peak rated voltage value at the supply terminals to the equipment.
- The circuits shall be limited to overvoltage Category II as defined in IEC 60664-1.
- The Fieldbus Components may only be removed or inserted when the system supply and the field supply are switched off, or when the location is known to be non-hazardous.
- The Fieldbus Components may only be disconnected or connected when the system supply is switched off, or when the location is known to be non-hazardous.

Standards

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

M20US0111X (US):

- FM Class 3600:2018
- FM Class 3611:2018
- FM Class 3810:2018
- ANSI/UL 121201:2019
- ANSI/ISA 61010-1:2012
- ANSI/UL 60079-0:2020
- ANSI/UL 60079-7:2017

FM20CA0053X (Canada):

- CAN/CSA C22.2 No. 213-17:2017
- CSA C22.2 No. 60079-0:2019
- CAN/CSA C22.2 No. 60079-7:2016
- CAN/CSA C22.2 No.61010-1:2012

Marking

Beckhoff fieldbus components that are certified in accordance with cFMus for use in areas subject to an explosion hazard bear the following markings:

FM20US0111X (US):	**Class I, Division 2, Groups A, B, C, D** **Class I, Zone 2, AEx ec IIC T4 Gc**
FM20CA0053X (Canada):	**Class I, Division 2, Groups A, B, C, D** **Ex ec T4 Gc**

5.2.6 Continuative documentation for cFMus

NOTICE	
	**Continuative documentation about explosion protection according to cFMus** Pay also attention to the continuative documentation **Control Drawing I/O, CX, CPX** Connection diagrams and Ex markings, that is available for download within the download area of your product on the Beckhoff homepage www.beckhoff.com!

5.3 UL notice

⚠ CAUTION
Application Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.

⚠ CAUTION
Examination For cULus examination, the Beckhoff I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).

⚠ CAUTION
For devices with Ethernet connectors Not for connection to telecommunication circuits.

Basic principles

UL certification according to UL508. Devices with this kind of certification are marked by this sign:



5.4 Positioning of passive terminals

Note on the placement of passive terminals in the bus terminal block

EtherCAT Terminals that do not take part in active data exchange within the bus terminal block are referred to as passive terminals. These terminals can be recognized because they do not consume power from the E-bus. To ensure optimal data exchange, no more than two passive terminals may be connected directly to each other!

Examples of the placement of passive terminals (highlighted in light color)

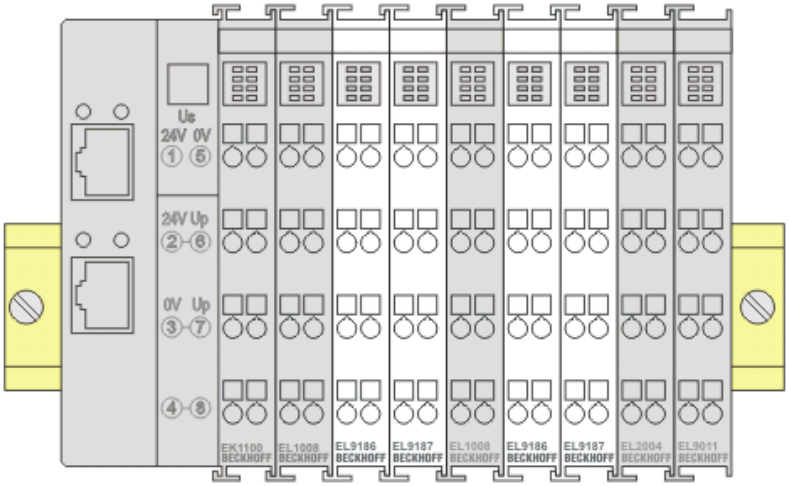


Fig. 13: Proper positioning

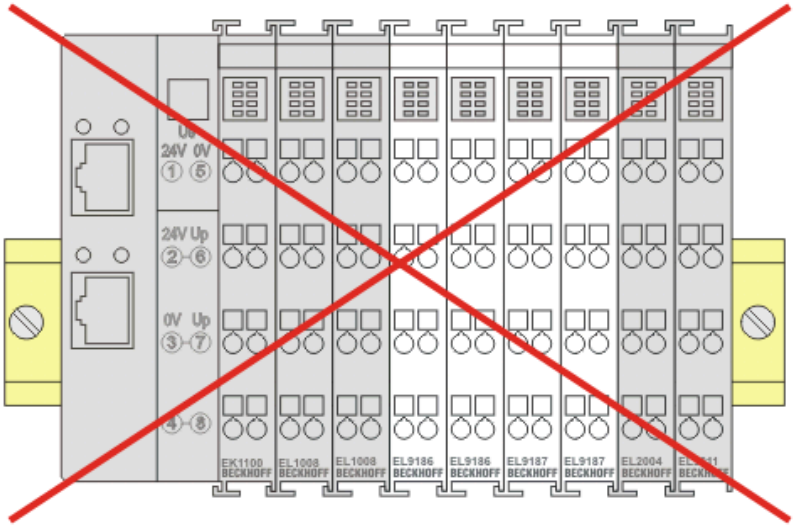


Fig. 14: Incorrect positioning

5.5 Installation instructions for enhanced mechanical load capacity

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the Bus Terminal system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals!

Additional checks

The terminals have undergone the following additional tests:

Verification	Explanation
Vibration	10 frequency runs in 3 axes
	6 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
	25 *g*, 6 ms

Additional installation instructions and notes

For terminals with enhanced mechanical load capacity, the following additional installation instructions and notes apply:

- The enhanced mechanical load capacity is valid for all permissible installation positions.
- Use a mounting rail according to EN 60715 TH35-15.
- Fix the terminal segment on both sides of the mounting rail with a mechanical fixture, e.g. an earth terminal or reinforced end clamp.
- The maximum total extension of the terminal segment (without coupler) is:
  64 terminals (12 mm mounting width) or 32 terminals (24 mm mounting width)
- Avoid deformation, twisting, crushing and bending of the mounting rail during edging and installation of the rail.
- The mounting points of the mounting rail must be set at 5 cm intervals.
- Use countersunk head screws to fasten the mounting rail.
- The free length between the strain relief and the wire connection should be kept as short as possible. A distance of approx. 10 cm should be maintained to the cable duct.

5.6 Installation positions

NOTICE

Constraints regarding installation position and operating temperature range

- Please refer to the technical data of the device to ascertain whether any restrictions regarding the installation position and/or the operating temperature range have been specified.
- When installing devices with increased heat dissipation, ensure that there is sufficient space above and below the devices during operation to guarantee adequate ventilation!

The installation positions and their names for mounting devices on mounting rails are specified below. The illustration of the devices in the following figures is an example.
The following applies to all installation positions: the reference direction “down” (see arrow) is the direction of the earth's gravitational force.

Horizontal installation (standard installation)

The mounting rail is mounted horizontally on a vertical mounting plate.
The connection level of the devices points to the front.

The devices are ventilated from below, which enables optimum cooling of the electronics through convection. This is therefore also the recommended installation position.

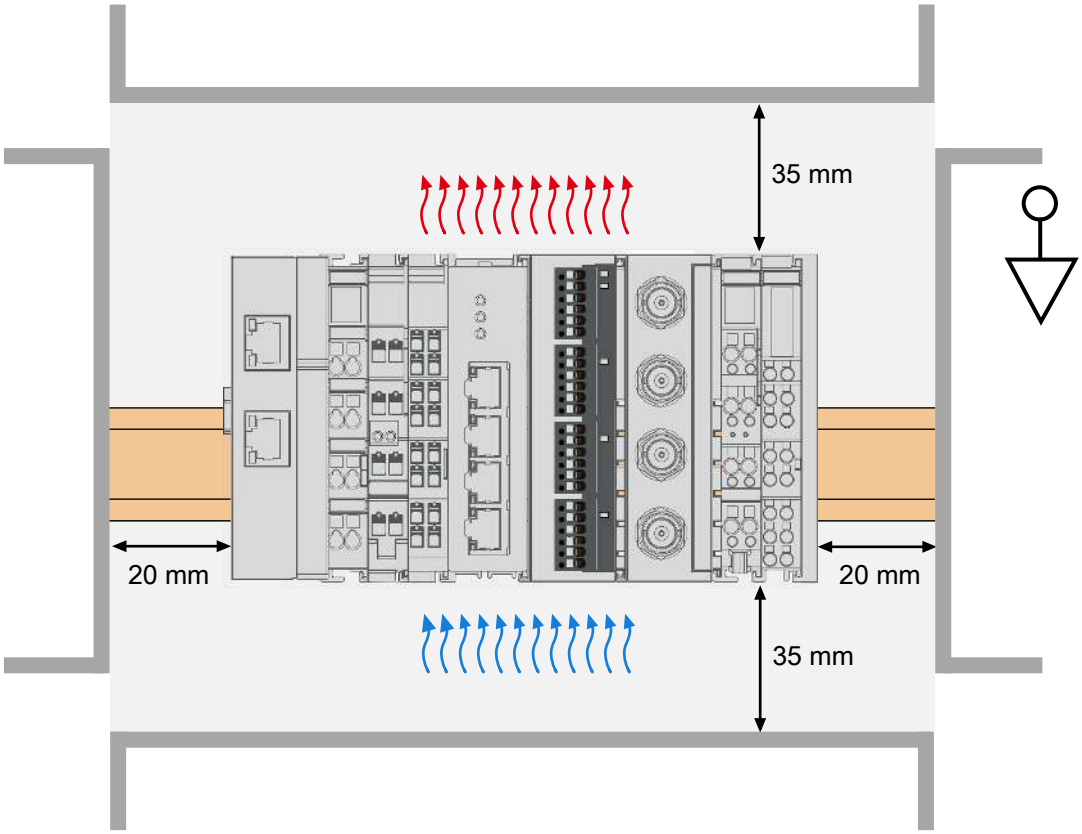


Fig. 15: Recommended minimum distances for standard installation position

NOTICE

Compliance with the minimum distances

Compliance with the minimum distances shown in the figure “Recommended minimum distances for standard installation position” is strongly recommended in all installation positions.

Vertical installation

The mounting rail is mounted vertically on a vertical mounting plate.
The connection level of the devices faces the front.
The devices can be arranged as follows:

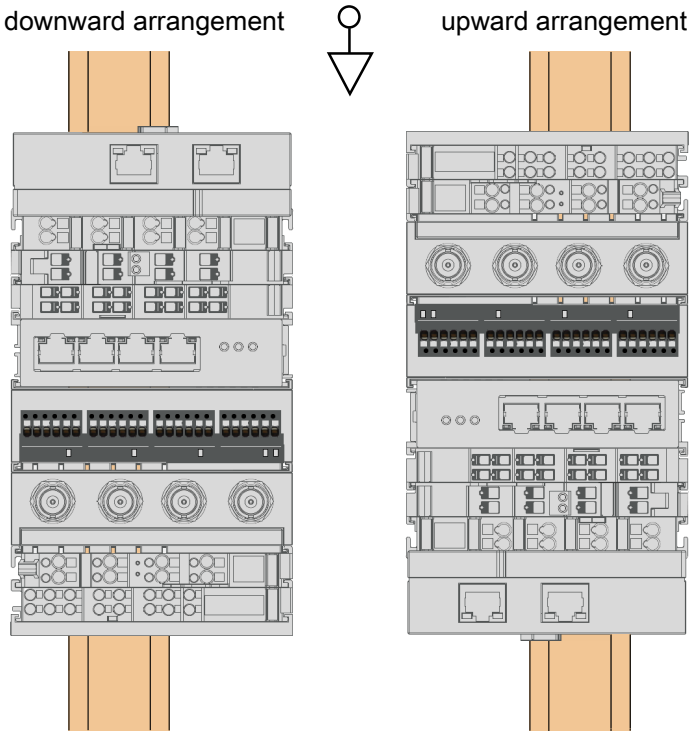


Fig. 16: Installation position: Vertical, downward arrangement (left)/upward arrangement (right)

Flat installation

The mounting rail is mounted on a horizontal mounting plate.
The connection level of the devices points upwards.

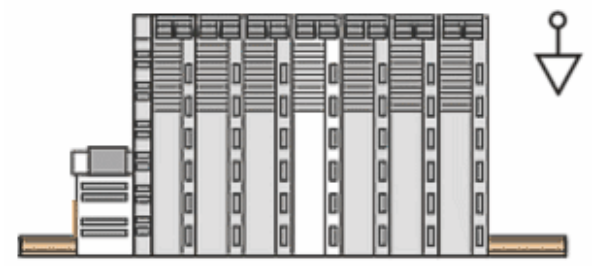


Fig. 17: Flat installation

NOTICE

Danger of sliding off the mounting rail

Particularly in the “vertical” installation position, but also in other installation positions under corresponding mechanical load, the terminal segment may move on the mounting rail. These can lead to undesirable malfunctions.

- If this risk exists, secure the terminal segment with appropriate locking devices, e.g. by clamping it to the mounting rail.

NOTICE

Compliance with the minimum distances

Compliance with the minimum distances shown in the figure “Recommended minimum distances for standard installation position” is strongly recommended in all installation positions.

Installation positions with ZB8610 fan cartridge

If the cooling should or must be increased for the intended application, the ZB8610 fan cartridge can be mounted on the underside of the device. In the horizontal installation position, the devices are ventilated from bottom to top by the fan cartridge. This optimum cooling is further enhanced by convection ventilation (see following figure).

The fan cartridge can be used in any installation position.

Further information on operation with and without a fan can be found in the technical data for the device (e.g. derating, information on installation positions, etc.).

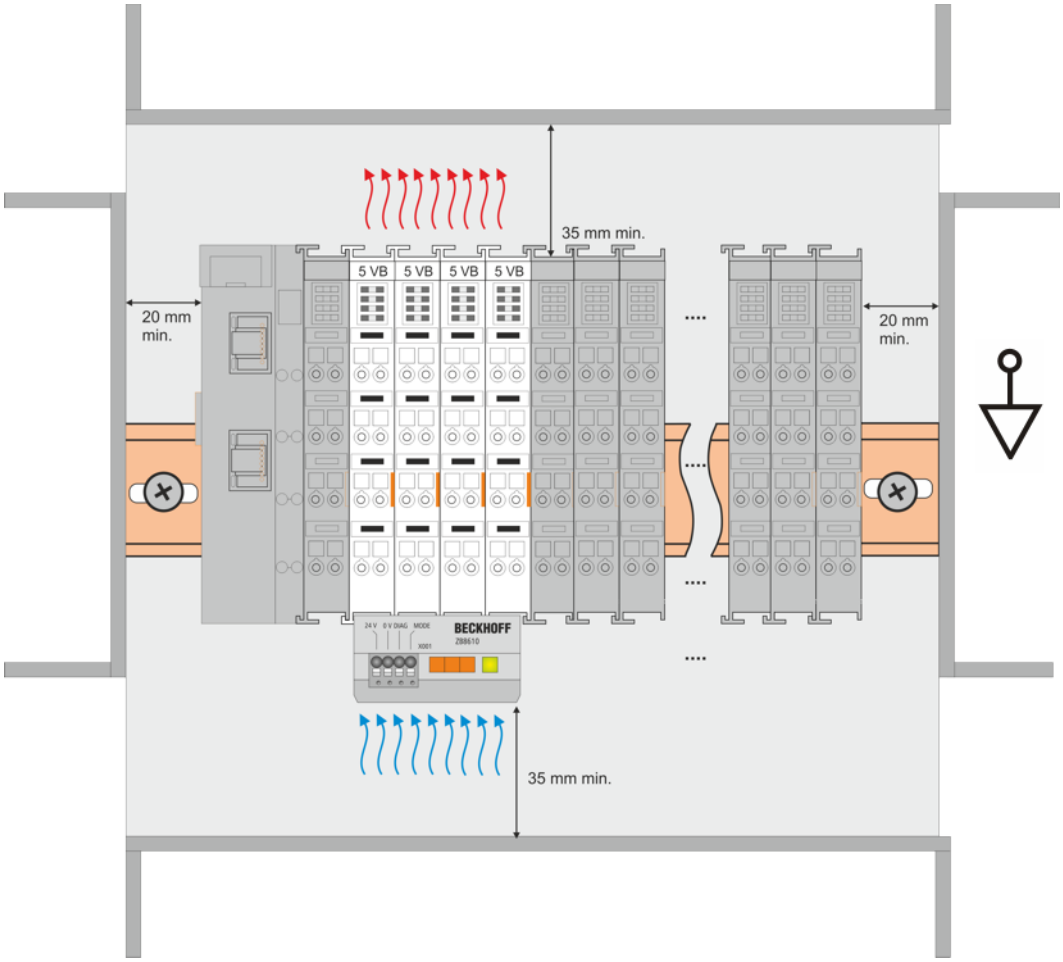


Fig. 18: Recommended minimum distances for operation with fan, using horizontal installation position as an example

NOTICE

Compliance with the minimum distances

Compliance with the minimum distances as shown in the figure “Recommended minimum distances for operation with fan” is strongly recommended.

5.7 Note on power supply

⚠ WARNING

Power supply from SELV / PELV power supply unit!

SELV / PELV circuits (safety extra-low voltage / protective extra-low voltage) according to IEC 61010-2-201 must be used to supply this device.

Notes:

- SELV / PELV circuits may give rise to further requirements from standards such as IEC 60204-1 et al, for example with regard to cable spacing and insulation.
- A SELV supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV supply also requires a safe connection to the protective conductor.

5.8 EL6001, EL6021-00xx

5.8.1 Mounting rail installation

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

The bus terminal system is designed for installation in a control cabinet or junction box.

Installation

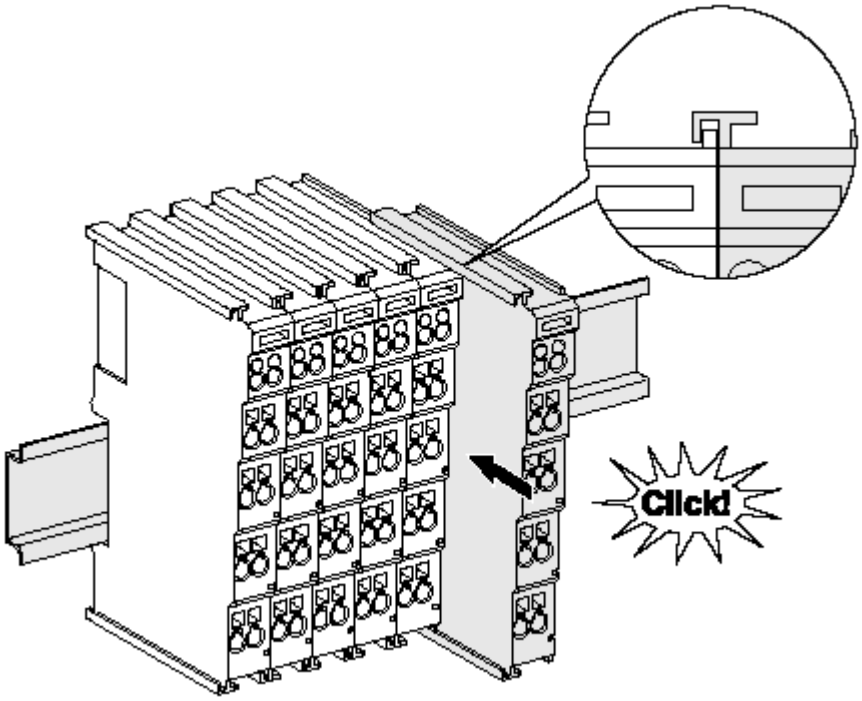


Fig. 19: Installation on the mounting rail

The bus couplers and bus terminals are attached to commercially available 35 mm mounting rails (DIN rail according to EN 60715) by applying slight pressure:

1. First attach the fieldbus coupler to the mounting rail.
2. The bus terminals are now attached on the right-hand side of the fieldbus coupler. To do this, connect the components with the slot and key and push the terminals against the mounting rail until the latch audibly engages on the mounting rail.
   If you snap the terminals onto the mounting rail first and then push them next to each other without the slot and key engaging, no functional connection will be established! When correctly assembled, no significant gap should be visible between the housing units.

Fastening the mounting rails

The locking mechanism of the terminals and couplers protrudes into the profile of the mounting rail. When installing the components, make sure that the locking mechanism doesn't come into conflict with the fixing bolts of the mounting rail. When fastening mounting rails with a height of 7.5 mm under the terminals and couplers, use flat fastening components such as countersunk head screws or blind rivets.

NOTICE

Earth the mounting rail!

Make sure the mounting rail is properly earthed.

Connections within a bus terminal block

The electric connections between the bus coupler and the bus terminals are made automatically by joining the components:

- The six spring-loaded contacts of the E-bus/K-bus deal with the data transfer and the supply of the bus terminal electronics.
- The power contacts deal with the supply for the field electronics and thus represent a supply rail within the bus terminal block. The power contacts are supplied via terminal points on the bus coupler (up to 24 V) or via power supply terminals for higher voltages.

Power contacts

When configuring a bus terminal block, note the contact assignments of the individual bus terminals, as some types

- do not loop through the power contacts or do not loop them through completely (e. g. analog bus terminals or digital 4-channel bus terminals),
- the power contacts are disconnected, thus marking the start of a new supply rail (power supply terminals).

Power contact ⏚

The power contact marked ⏚ (ground connection in accordance with IEC 60417-5017) can be used as a ground. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

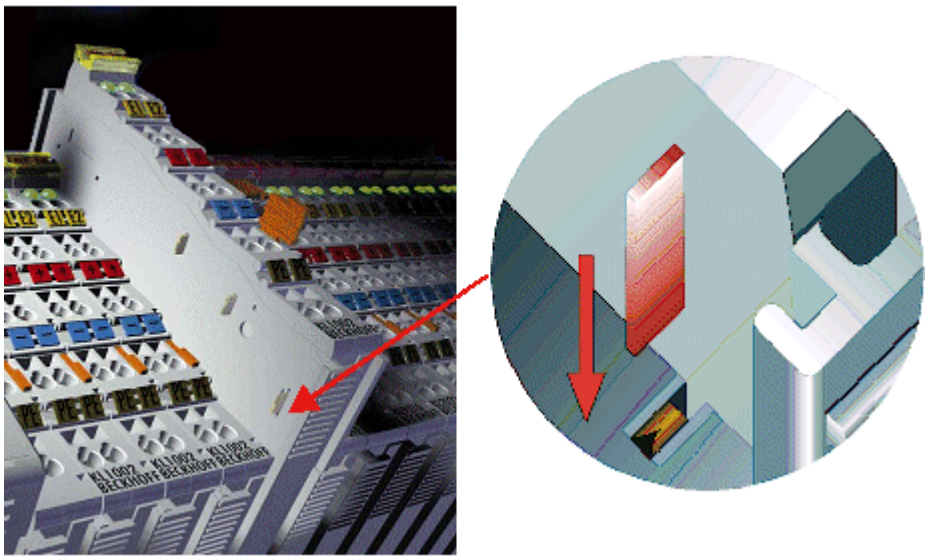


Fig. 20: Power contact on the left

⚠ WARNING

Risk of injury from electric shock!

The power contact marked ⏚ may not be used for other potentials!

NOTICE

Risk of damage to the device

Note that, for electromagnetic compatibility, the earth contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage to the terminal (e.g. disruptive discharge to the earth line during insulation testing of a consumer with a rated voltage of 230 V). For insulation testing, disconnect the earth supply line at the bus coupler or the power supply terminal! In order to decouple further supply points for testing, these power supply terminals can be released and pulled at least 10 mm from the group of terminals.

Disassembly

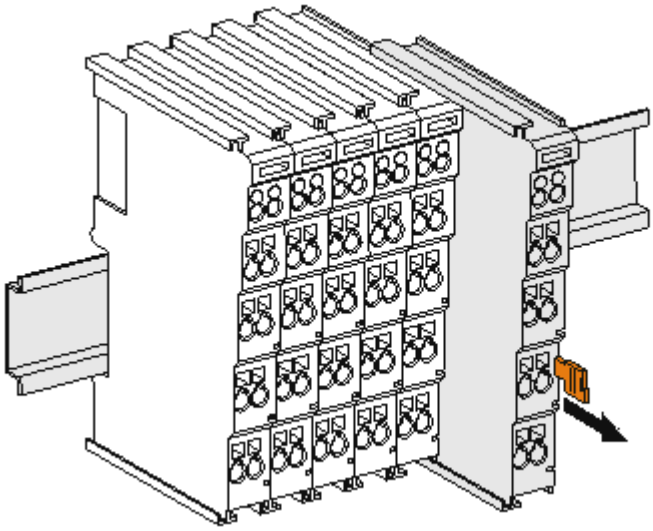


Fig. 21: Removal from mounting rail

Each terminal is secured by a lock on the mounting rail, which must be released for disassembly:

1. Pull down the terminal at its orange-colored straps from the mounting rail by approx. 1 cm. The rail locking of this terminal is automatically released, and you can now pull the terminal out of the bus terminal block with little effort.
2. To do this, grasp the unlocked terminal simultaneously at the top and bottom of the housing surfaces with your thumb and index finger and pull it out of the bus terminal block.

5.8.2 Connection

5.8.2.1 Connection system

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

Overview

The bus terminal system offers different connection options for optimum adaptation to the respective application:

- The terminals of ELxxxx and KLxxxx series with standard wiring include electronics and connection level in a single enclosure.
- The terminals of ESxxxx and KSxxxx series feature a pluggable connection level and enable steady wiring while replacing.
- The High Density Terminals (HD Terminals) include electronics and connection level in a single enclosure and have advanced packaging density.

Standard wiring (ELxxxx / KLxxxx)



Fig. 22: Standard wiring

The terminals of the ELxxxx and KLxxxx series integrate screwless spring-cage technology for quick and easy wiring.

Pluggable wiring (ESxxxx / KSxxxx)

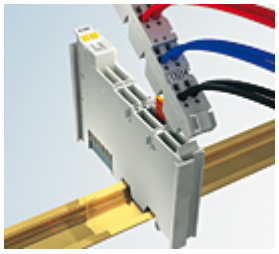


Fig. 23: Pluggable wiring

The terminals of ESxxxx and KSxxxx series feature a pluggable connection level.
The assembly and wiring procedure is the same as for the ELxxxx and KLxxxx series.
The pluggable connection level enables the complete wiring to be removed as a plug connector from the top of the housing for servicing.
The lower section can be removed from the terminal block by pulling the unlocking tab.
Insert the new component and plug in the connector with the wiring. This reduces the installation time and eliminates the risk of wires being mixed up.

The familiar dimensions of the terminal only had to be changed slightly. The new connector adds about 3 mm. The maximum height of the terminal remains unchanged.

A tab for strain relief of the cable simplifies assembly in many applications and prevents tangling of individual connection wires when the connector is removed.

Conductor cross sections between 0.08 mm^2 and 2.5 mm^2 can continue to be used with the proven spring force technology.

The overview and nomenclature of the product names for ESxxxx and KSxxxx series has been retained as known from ELxxxx and KLxxxx series.

High Density Terminals (HD Terminals)



Fig. 24: High Density Terminals

The terminals from these series with 16/32 terminal points are distinguished by a particularly compact design, as the packaging density is twice as large as that of the standard 12 mm bus terminals. Massive conductors and conductors with a wire end sleeve can be inserted directly into the spring loaded terminal point without tools.

Wiring HD Terminals

The High Density Terminals of the ELx8xx and KLx8xx series doesn't support pluggable wiring.

Ultrasonically compacted (ultrasonically welded) strands

Ultrasonically compacted (ultrasonically welded) strands

Ultrasonically compacted (ultrasonically welded) strands can also be connected to the standard and high-density terminals. In this case, please note the tables concerning the wire-size width [▶ 54]!

5.8.2.2 Wiring

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

Terminals for standard wiring ELxxxx/KLxxxx and for pluggable wiring ESxxxx/KSxxxx

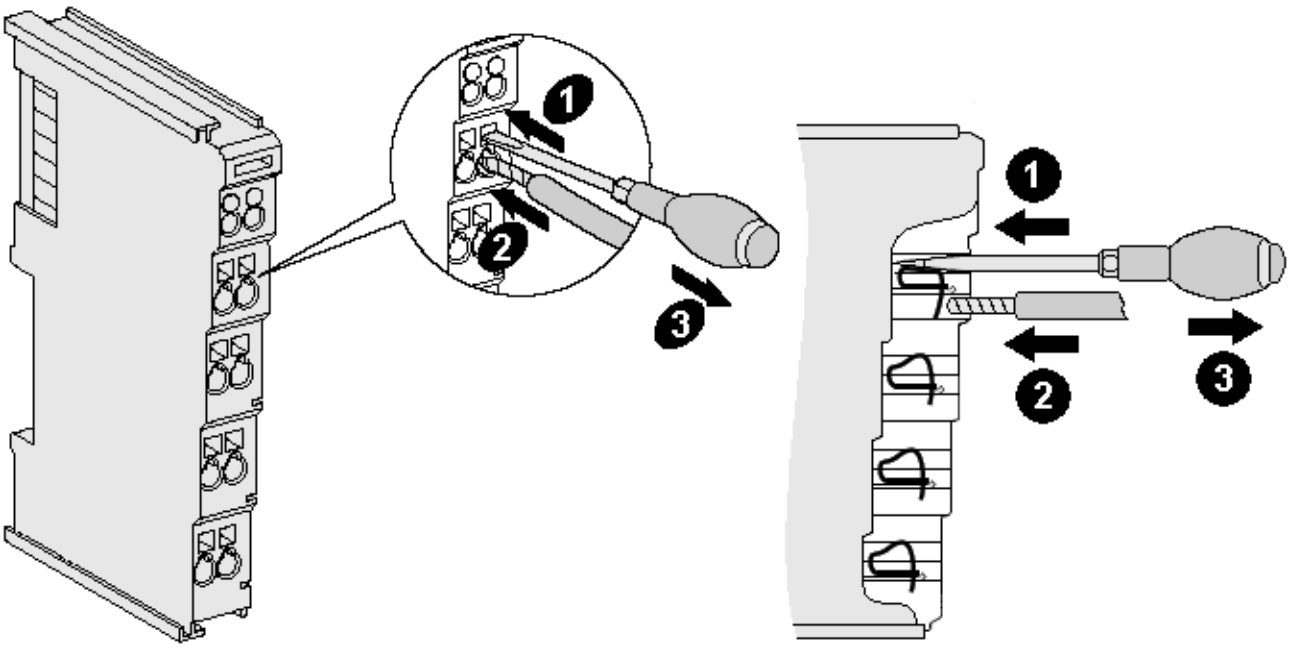


Fig. 25: Connecting a cable on a terminal point

Up to eight terminal points enable the connection of solid or finely stranded cables to the bus terminal. The terminal points are implemented in spring force technology. Connect the cables as follows (see fig. “Connecting a cable on a terminal point”:

1. Open a terminal point by pushing a screwdriver straight against the stop into the square opening above the terminal point. Do not turn the screwdriver or move it alternately (don't toggle).
2. The wire can now be inserted into the round terminal opening without any force.
3. When the screwdriver is removed, the terminal point closes automatically and holds the wire securely and permanently in place

See the following table for the suitable wire size width:

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width (single core wires)	0.08 ... 2.5 mm²	0.08 ... 2.5 mm²
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm²	0.08 ... 2.5 mm²
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm²	0.14 ... 1.5 mm²
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals (HD Terminals [▶ 52]) with 16/32 terminal points

The conductors of the HD Terminals are connected without tools for single-wire conductors using the direct plug-in technique, i.e. after stripping the wire is simply plugged into the terminal point. The cables are released, as usual, using the contact release with the aid of a screwdriver. See the following table for the suitable wire size width.

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 mm²
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm²
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm²
Wire size width (ultrasonically compacted [ultrasonically welded] strands)	only 1.5 mm² (see notice [▶ 52])
Wire stripping length	8 ... 9 mm

5.8.2.3 Shielding

Shielding

Encoders, analog sensors, and actuators should always be connected with shielded, pair-wise twisted cables.

5.8.3 LEDs and connector assignments

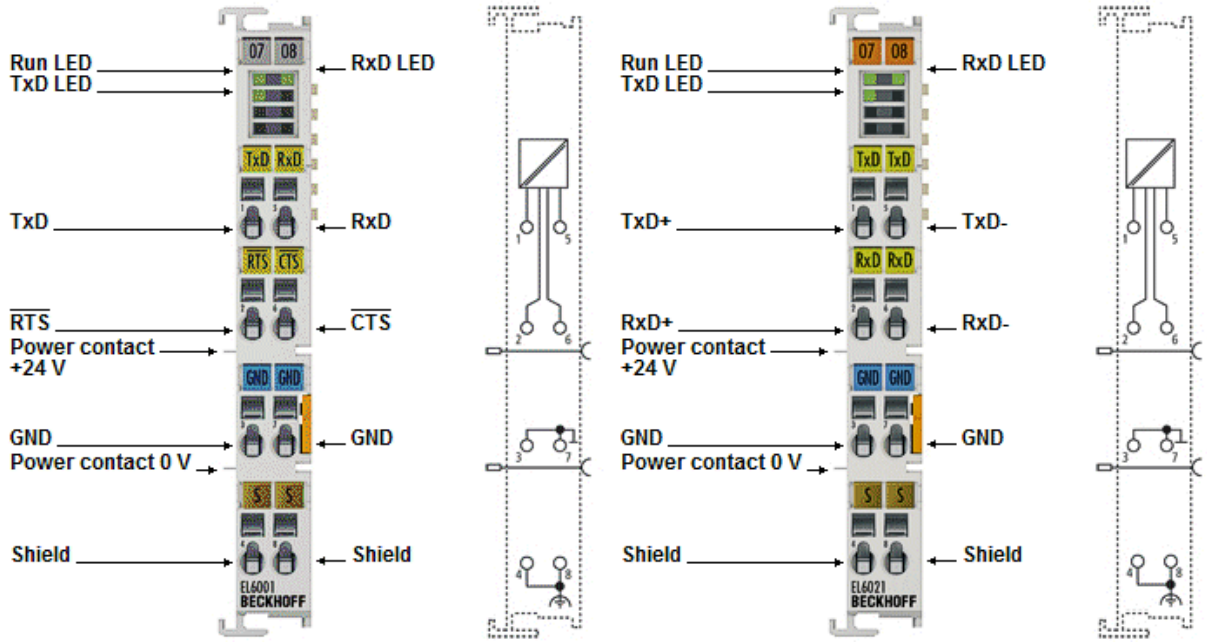


Fig. 26: EL6001, EL6021-00xx - LEDs and connector assignments

LEDs

LED	Color	Meaning	
RUN	Green	This LED indicates the terminal's operating state:	
		Off	State of the EtherCAT State Machine [▶ 25]: INIT = initialization of the terminal or BOOTSTRAP = function for firmware updates of the terminal
		flashing	State of the EtherCAT state machine: PREOP = function for mailbox communication and variant standard settings
		Single flash	State of the EtherCAT state machine: SAFEOP = verification of the Sync Manager channels and the distributed clocks. Outputs remain in safe state
		On	State of the EtherCAT State Machine: OP = normal operating state; mailbox and process data communication is possible
TxD	Green		State of the transmit signal line (on: HI signal level on transmit line)
RxD	Green		State of the receive signal line (on: HI signal level on receive line)

EL6001 terminal connector assignments

Terminal point	Name	Direction	Signal
1	TxD	→	Signal line (Transmit Data)
5	RxD	←	Signal line (Receive Data)
2	RTS	→	Control line (Request To Send)
6	CTS	←	Control line (Clear To Send)
3	GND	-	Ground (internally bridged with terminal 7)
7	GND	-	Ground (internally bridged with terminal 3)
4	Shield	-	Shield (internally bridged with terminal 8)
8	Shield	-	Shield (internally bridged with terminal 4)

EL6021-00xx terminal connector assignments

Terminal point	Name	Direction	Signal
1	TxD+	→	Signal line + (Transmit Data)
5	TxD-	→	Signal line - (Transmit Data)
2	RxD+	←	Signal line + (Receive Data)
6	RxD-	←	Signal line - (Receive Data)
3	GND	-	Ground (internally bridged with terminal 7)
7	GND	-	Ground (internally bridged with terminal 3)
4	Shield	-	Shield (internally bridged with terminal 8)
8	Shield	-	Shield (internally bridged with terminal 4)

Connection for RS422 transfer

In RS422 mode, data can be transferred in full duplex mode.

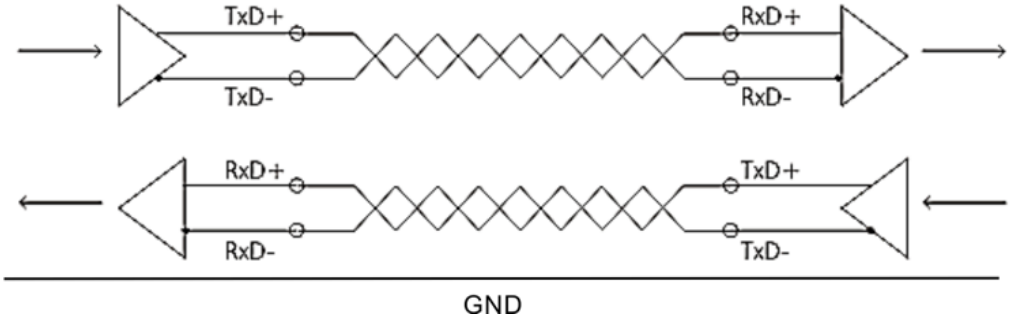


Fig. 27: Connection for RS422 transfer

One receiver is recommended for RS422, but up to 10 receivers are common. The more complex the network becomes (subscribers, branches, plug connections), the more complex the commissioning becomes, because the appropriate parameters must be determined for reliable communication: Termination (where and at what height), baud rate, line lengths if necessary.

A clear line topology with short branches (MultiDrop) is strongly recommended, if necessary terminated at both end points with the greatest distance.

Connection for RS485 transfer

In the RS485 operating mode, the data is exchanged in half-duplex transmission. In this operating mode, a bus structure consisting of several stations communicating with each other can be set up.

A clear line structure with short branches is recommended.



Fig. 28: Connection for RS485 transfer

The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own transmitted data. This can be suppressed with the index 0x8000:06 “Enable half duplex” in the Settings object.

In operating mode RS485, the reception of new data is only possible if transmission is complete.

“Enable half duplex”	“Enable point to point connection (RS422)”	Mode
0	0	RS485: The terminal receives its own data and the data from other devices
0	1	RS422: Normal operating mode; the terminal operates in full duplex mode.
1	0	RS485: The terminal only receives data from other devices
1	1	RS422: The receiver is only enabled after the last data has been transmitted.

5.9 EL6002, EL6022

5.9.1 Mounting and demounting - top front unlocking

The terminal modules are fastened to the assembly surface with the aid of a 35 mm mounting rail (e. g. mounting rail TH 35-15).

Fixing of mounting rails

The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the recommended mounting rails under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

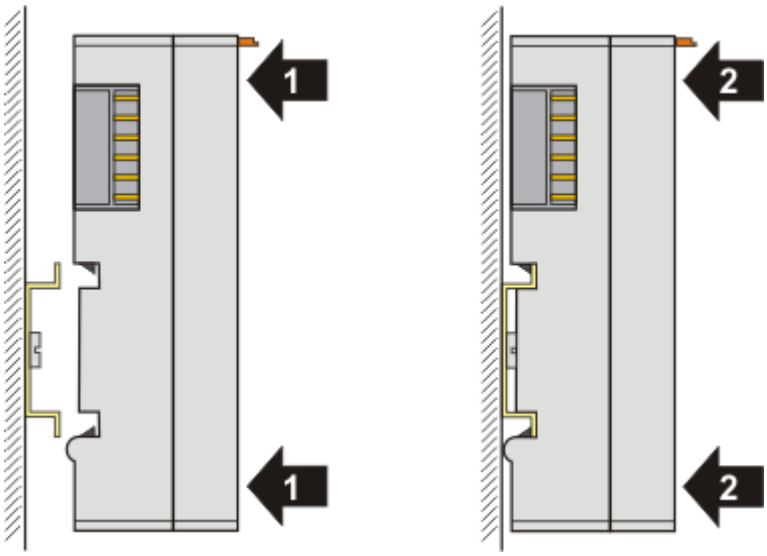
⚠ WARNING

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Mounting

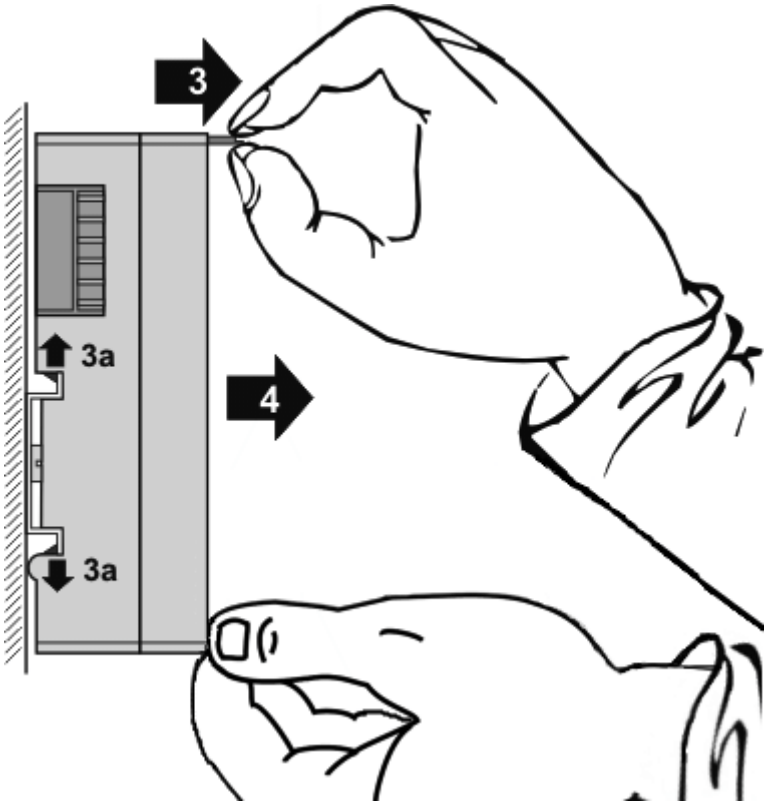
- Fit the mounting rail to the planned assembly location.



  and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).
- Attach the cables.

Demounting

- Remove all the cables.
- Lever the unlatching hook back with thumb and forefinger (3). An internal mechanism pulls the two latching lugs (3a) from the top hat rail back into the terminal module.



- Pull (4) the terminal module away from the mounting surface.
  Avoid canting of the module; you should stabilize the module with the other hand, if required.

5.9.2 LEDs and pin assignment

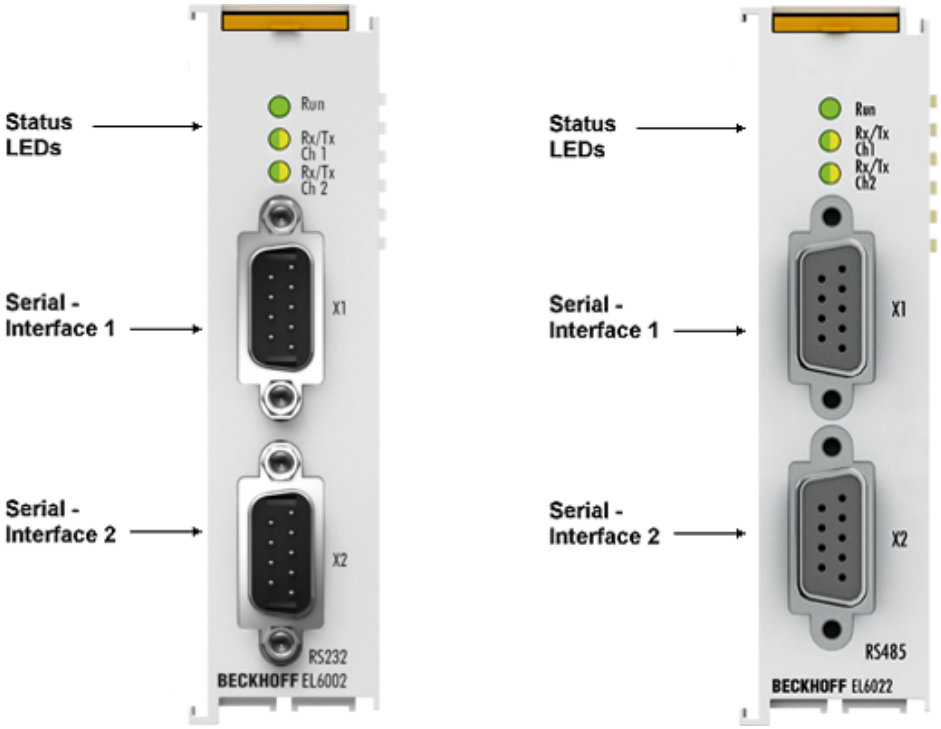


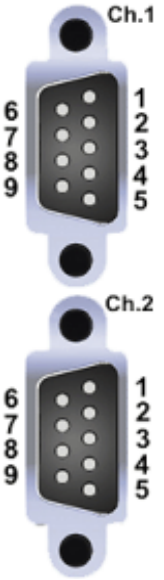
Fig. 29: El6002, EL6022 - LEDs

LEDs

LED	Color	Meaning	
RUN	Green	This LED indicates the terminal's operating state:	
		Off	State of the EtherCAT State Machine [▶ 25]: INIT = initialization of the terminal or BOOTSTRAP = function for firmware updates of the terminal
		flashing	State of the EtherCAT State Machine: PREOP = function for mailbox communication and variant standard settings
		Single flash	State of the EtherCAT State Machine: SAFEOP = verification of the Sync Manager channels and the distributed clocks. Outputs remain in safe state
		On	State of the EtherCAT State Machine: OP = normal operating state; mailbox and process data communication is possible
TxCh. 1	Orange		Serial port on this port sends data (channel 1)
RxCh. 1	Green		Serial port on this connector receives data (channel 1)
TxCh. 2	Orange		Serial port on this port sends data (channel 2)
RxCh. 2	Green		Serial port on this port receives data (channel 2)

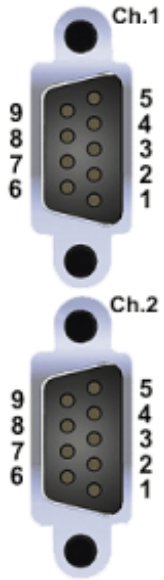
EL6002 pin assignment

2 x D-Sub connection male 9-pin

D-Sub connector, male (top view)	Pin assignment channel 1 and 2			
	**Pin**	**RS232**	**Direction**	**Note**
	1	DCD	→	internally bridged with DTR, DSR
	2	Rx	←	
	3	Tx	→	
	4	DTR	→	internally bridged with DCD, DSR
	5	GND	-	
	6	DSR	←	internally bridged with DCD, DTR
	7	RTS	→	
	8	CTS	←	
	9	-	-	-

EL6022 pin assignment

2 x D-Sub connector female, 9-pin

D-Sub connector, male (top view)	Pin assignment channel 1 and 2		
	**Pin**	**RS485/RS422**	**Direction**
	1	-	-
	2	Tx+	→
	3	Rx+	←
	4	-	-
	5	GND	-
	6	+5 V	-
	7	Tx-	→
	8	Rx-	←
	9	-	-

Screw-in Depth of D-Sub Connectors

The D-Sub connectors on the EL6002 / EL6022 feature 4-40 UNC threads.
The maximum allowable screw-in depth is 8 mm.

Connector housing (shield)

The housing of the D-Sub connectors for both channels is connected to the shield contact spring of the mounting rail latch via a high-impedance RC combination.

Connection for RS422 transfer

In RS422 mode, data can be transferred in full duplex mode.

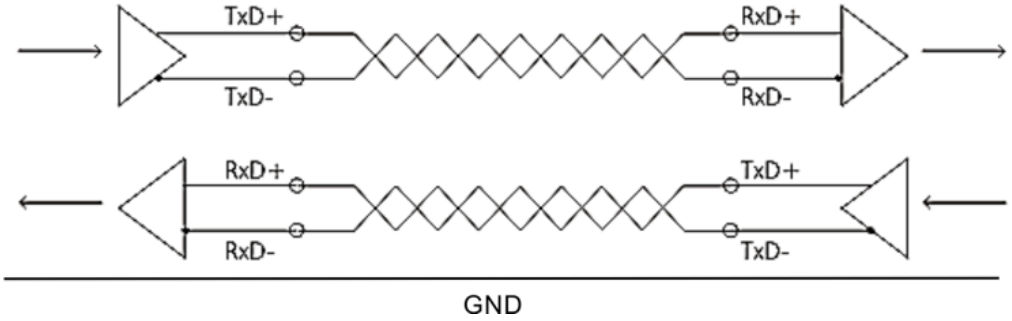


Fig. 30: Connection for RS422 transfer

One receiver is recommended for RS422, but up to 10 receivers are common. The more complex the network becomes (subscribers, branches, plug connections), the more complex the commissioning becomes, because the appropriate parameters must be determined for reliable communication: Termination (where and at what height), baud rate, line lengths if necessary.

A clear line topology with short branches (MultiDrop) is strongly recommended, if necessary terminated at both end points with the greatest distance.

Connection for RS485 transfer

In the RS485 operating mode, the data is exchanged in half-duplex transmission. In this operating mode, a bus structure consisting of several stations communicating with each other can be set up.

A clear line structure with short branches is recommended.



Fig. 31: Connection for RS485 transfer

The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own transmitted data. This can be suppressed with the index 0x8000:06 “Enable half duplex” in the Settings object.

In operating mode RS485, the reception of new data is only possible if transmission is complete.

“Enable half duplex”	“Enable point to point connection (RS422)”	Mode
0	0	RS485: The terminal receives its own data and the data from other devices
0	1	RS422: Normal operating mode; the terminal operates in full duplex mode.
1	0	RS485: The terminal only receives data from other devices
1	1	RS422: The receiver is only enabled after the last data has been transmitted.

5.10 Disposal



Products marked with a crossed-out wheeled bin shall not be discarded with the normal waste stream. The device is considered as waste electrical and electronic equipment. The national regulations for the disposal of waste electrical and electronic equipment must be observed.

6 Commissioning

NOTICE



Further information

For information on commissioning with TwinCAT and EtherCAT basics, please refer to the "EtherCAT system manual".

6.1 Operating modes and process data

Version notes

The single-channel EL60x1 has been developed and enhanced over a number of years. Through further development of the EL6001/EL6021 the following functional extensions have been implemented:

- from firmware (FW) 05 / hardware (HW) 03, (EL6001); firmware (FW) 04 / hardware (HW) 03, (EL6021) the objects for status monitoring and parameterization are also available from index 0x6000 (profile-specific objects) and can be parameterized in the TwinCAT System Manager, depending on the hardware.
- from firmware (FW) 06 / hardware (HW) 03, (EL6021) the command mode [▶ 72] is supported
- from firmware 08 (EL6001) the 16-bit process data interface for sending/receiving of > 8 bit is supported
- from firmware 11 (EL6001) all integer baud rates 1000…115200 baud can be used.
-

Older versions of the EL6001/EL6021 do not support this functionality!

from FW/HW	ESI	Control/status/parameterizing objects area
EL6001 01/01 EL6021 01/01	from EL6001-0000-0000 from EL6021-0000-0000	Index 0x300n:01 (Control-Word) Index 0x310n:01 (Status-Word) Index 0x4070 (Data bytes in send buffer) Index 0x4071 (Data bytes in receive buffer) Index 0x4072 (Diagnosis) Index 0x4073 (Baud rate) Index 0x4074 (Data frame) Index 0x4075 (Feature bits)
EL6001 05/03 EL6021 04/03	from EL6001-0000-0016 from EL6021-0000-0016	in addition to the above described objects: Index 0x6000 (COM inputs) Index 0x7000 (COM outputs) Index 0xA000 (COM Diag data) Index 0x8000 (COM settings)
EL6021 06/03	from EL6021-0000-0018	in addition to the above described objects: Index 0xB000 (Command)
EL6001 08/03	from EL6001-0000-0019	in addition to the above described objects: Objects for 16-bit PDO
EL6001 11/11	from EL6001-0000-0020	All baud rates 1000…115200 via explicitBaudrate implemented

The EL6002/EL6022 devices with revision -0016 already feature objects in the 6000, 7000 and 8000 range from the first release.

Compatibility in the case of service

Example: An EL6001/EL6021 employed and projected with Hardware 03 or higher cannot be replaced by an EL6001/EL6021 with an older hardware version (< 03)! The reverse case can be implemented without problem!

Process data monitoring

- **WcState**: if ≠ 0, this EtherCAT device does not take part in the process data traffic
- **State**: if ≠ 8, the EtherCAT device is not in OP (operational) status
- **TxPDO state, SyncError:** if ≠ 0, then no valid process data are available, e. g. caused by broken wire
- **TxPDO Toggle**: if this bit is toggling, a new set of process data is available

StartUp entries (hardware < 03)

StartUp list entry

For the EL6001/EL6021 with hardware < 03, the startUp entries can only be set in the transition from SafeOP to OP (S -> O). The default setting is PreOP -> SafeOP (P -> S). When creating StartUp entries, ensure that the checkbox "S -> O" is ticked (see Fig.)!

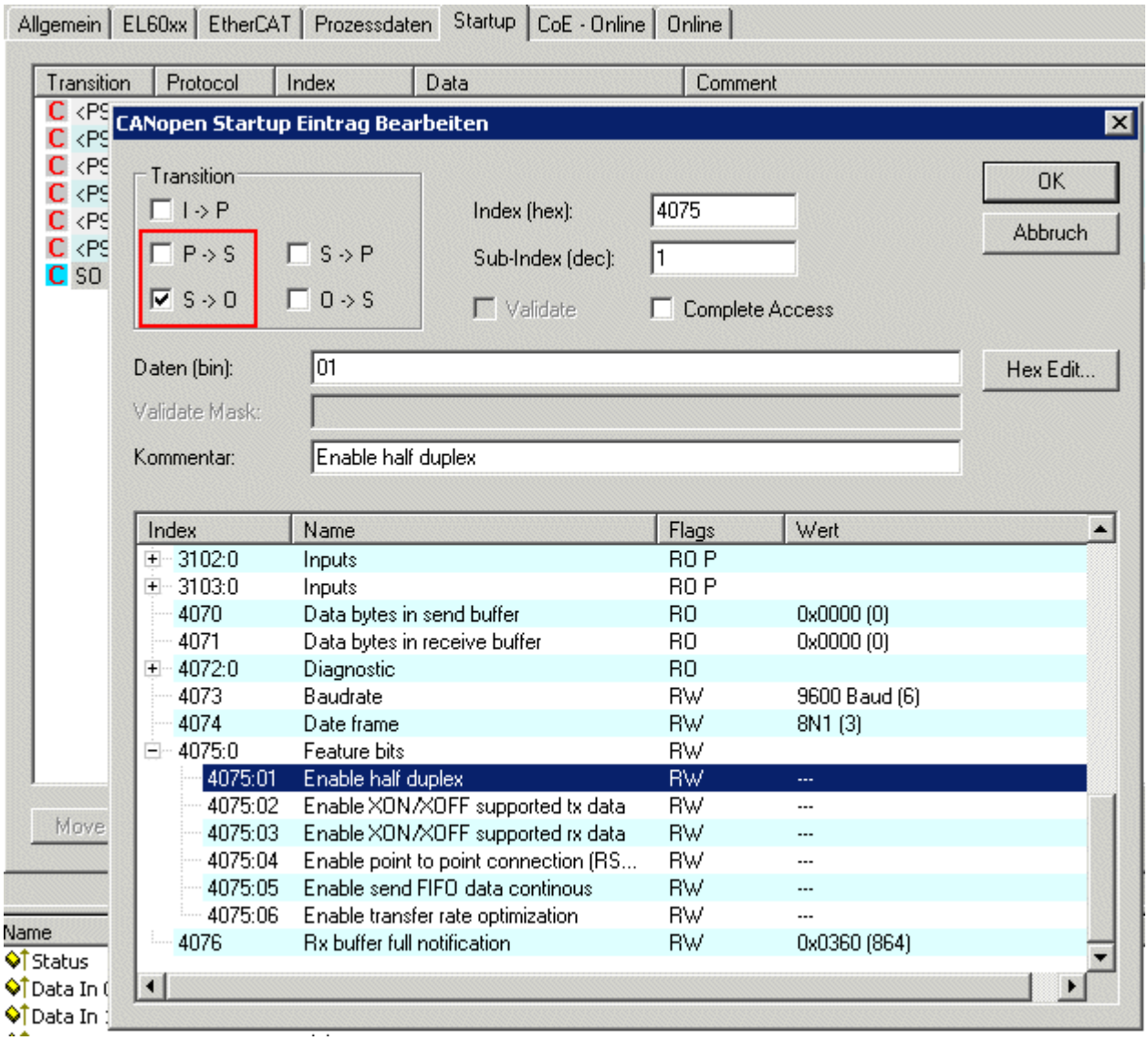


Fig. 32: StartUp entry with transition S -> O

Process data

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

The process data are generated from CoE objects 0x6000 (Inputs) and 0x7000 (Outputs) and are described in chapter Object description and parameterization [▶ 115] in detail.

Enlarged process image 50x 16-bit

An enlarged process data interface is necessary for communication with > 8 data bits. If the terminal supports this (see Communication features [▶ 76]) a 50-word interface can be set as an alternative to the 22-byte interface (PDOs 0x1605 and 0x1A05). This can be used with every encryption regulation (7xx, 8xx), but only makes sense with a regulation > 8 bits, e. g. 9N1. In each case the least significant bits must be occupied by the data bits, i.e. in the case of 9N1 the 9 least significant bits in the data words.

Process data of the EL60x2 from revision -0016

EL60x2 from revision -0016

Sync Manager (SM) – PDO Assignment

SM2, PDO assignment 0x1C12				
Index	**Index of excluded PDOs**	**Size (byte.bit)**	**Name**	**PDO content**
0x1600	-	24.0	COM Outputs Channel1	Index 0x7000:01 - Ctrl__Transmit request
				Index 0x7000:02 - Ctrl__Receive accepted
				Index 0x7000:03 - Ctrl__Init request
				Index 0x7000:04 - Ctrl__Send continous
				Index 0x7000:09 - Ctrl__Output length
				Index 0x7000:11 - Data out 0
				--
				Index 0x7000:26 - Data out 21
0x1601	-	24.0	COM Outputs Channel2	Index 0x7010:01 - Ctrl__Transmit request
				Index 0x7010:02 - Ctrl__Receive accepted
				Index 0x7010:03 - Ctrl__Init request
				Index 0x7010:04 - Ctrl__Send continuous
				Index 0x7010:09 - Ctrl__Output length
				Index 0x7010:11 - Data out 0
				--
				Index 0x7010:26 - Data out 21
0x1604 (default)	-	24.0	COM Outputs Channel1	Index 0x7001:01 - Ctrl
				Index 0x7000:11 - Data out 0
				--
				Index 0x7000:26 - Data out 21
0x1605 (default)	-	24.0	COM Outputs Channel2	Index 0x7011:01 - Ctrl
				Index 0x7010:11 - Data out 0
				--
				Index 0x7010:26 - Data out 21

SM3, PDO assignment 0x1C13				
Index	**Index of excluded PDOs**	**Size (byte.bit)**	**Name**	**PDO content**
0x1A00	-	24.0	COM Inputs Channel1	Index 0x6000:01 - Status__Transmit accepted
				Index 0x6000:02 - Status__Receive request
				Index 0x6000:03 - Status__Init accepted
				Index 0x6000:04 - Status__Buffer full
				Index 0x6000:05 - Status__Parity error
				Index 0x6000:06 - Status__Framing error
				Index 0x6000:07 - Status__Overrun error
				Index 0x6000:09 - Status__Input length
				Index 0x6000:11 - Data in 0
				--
				Index 0x6000:26 - Data in 21
0x1A01	-	24.0	COM Inputs Channel2	Index 0x6010:01 - Status__Transmit accepted
				Index 0x6010:02 - Status__Receive request
				Index 0x6010:03 - Status__Init accepted
				Index 0x6010:04 - Status__Buffer full
				Index 0x6010:05 - Status__Parity Error
				Index 0x6010:06 - Status__Framing error
				Index 0x6010:07 - Status__Overrun error
				Index 0x6010:09 - Status__Input length
				Index 0x6010:11 - Data in 0
				--
				Index 0x6000:26 - Data in 21
0x1A04 (default)	-	24.0	COM Inputs Channel1	Index 0x6001:01 - Status
				Index 0x6000:11 - Data in 0
				--
				Index 0x6000:26 - Data in 21
0x1A05 (default)	-	24.0	COM Inputs Channel2	Index 0x6011:01 - Status
				Index 0x6010:11 - Data in 0
				--
				Index 0x6010:26 - Data in 21

Features and application notes

Optional features are generally set in the CoE directory (CAN over EtherCAT, index 0x80n0).

Parameterization via the CoE list (CAN over EtherCAT)

Please note the following general CoE information when using/manipulating the CoE parameters: - Keep a startup list if components have to be replaced - Differentiate between online/offline dictionary, ensure existence of current XML description - use “CoE reload” for resetting changes Please note especially for the EL60xx, if any changes are made to the communication settings (baud rate, data frame, feature bits), an InitRequest via the control word is required to apply the changes.

The following CoE settings are possible from object 0x8000 and are shown below in their default settings:

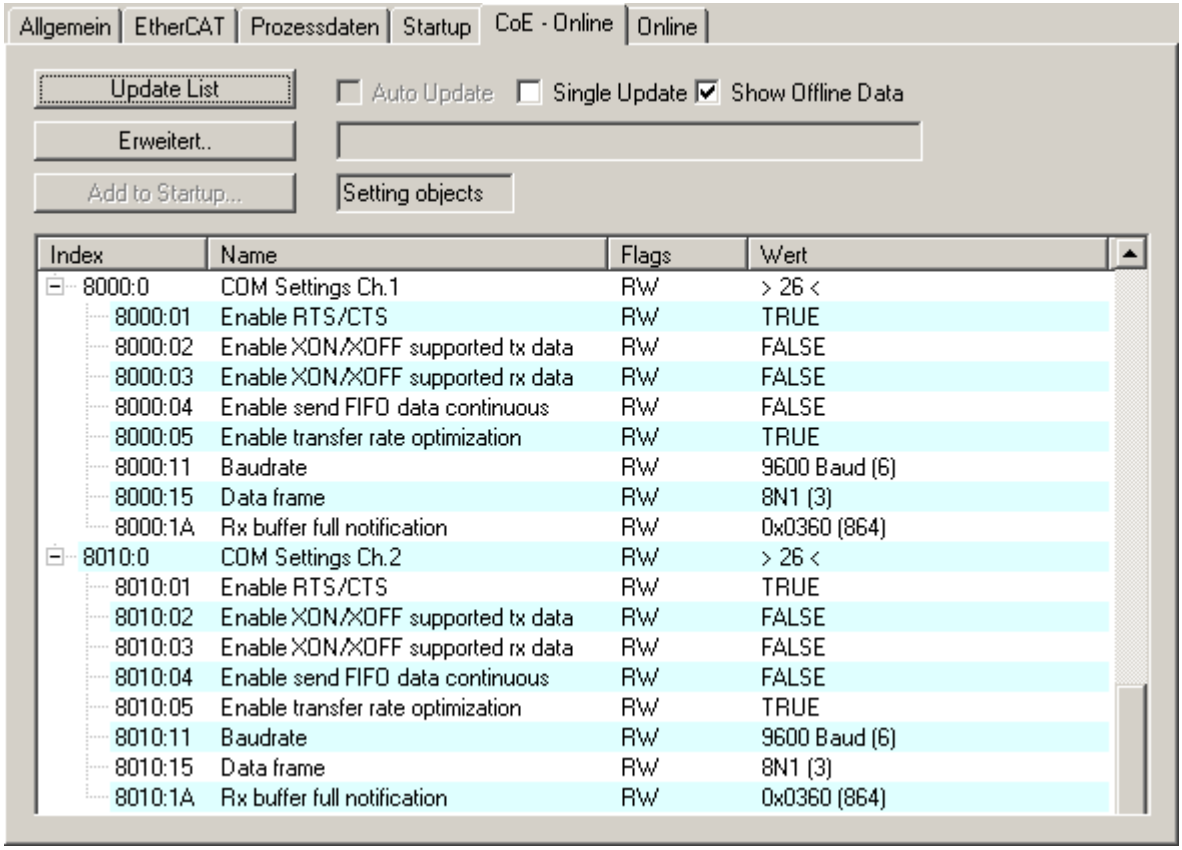


Fig. 33: “CoE - Online, EL60x2 terminals” tab

Transfer rates

The terminal has 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 terminal 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]*22[Byte]*10[Bit] = 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.

From firmware 11 the CoE 0x8000:1B supports with *explicitBaudrate* all baud rates 1000…115200 baud. When the baud rate is selected, the compatibility of the remote station to the set baud rate must be taken into account.
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 fetch the data quickly enough from the terminal, the data will be stored intermediately in the internal buffer of the terminal. The buffer for received data has a size of 864 bytes. If this is exhausted, all further data will be lost.

Another scenario: the controller transfers significantly more data to the terminal than the latter can transmit. For a ‘baud rate’ setting of 300 and a ‘data frame’ setting of 8N1, the terminal can only transmit 30 bytes per second. However, if more than these 30 bytes per second are received, a 128 byte transmit buffer will be written to first in this case also. Once this is full, all further data will be lost.

Transfer rate optimization: Automatic summary of received data

In serial communication, a data record is usually sent as a continuous stream of bytes. There is a pause between two data records. Based on the pauses, the terminal can recognize when a data record starts and when it ends. In this way, it can combine the bytes of a data record and pass them on to the controller in a coherent manner.

Mode of operation: Several bytes that are received in quick succession are first collected in the receive buffer. If a pause is detected after a byte (or the buffer is full), the collected receive data is transferred from the receive buffer to the process image. The "Receive Request" bit is inverted to indicate that new receive data is available.

Activate / deactivate

The automatic summary of receive data is activated in the factory setting in the CoE object 0x80n0:05 "Enable transfer rate optimization". It may be useful to deactivate it if the data that a terminal device sends is to be received as quickly as possible in the controller. Or if the end device sends continuously without pauses. If the automatic summary of received data is deactivated, each received byte is immediately transferred to the process image.

Adjustable length of the pause between two data sets

Note: This function is available in

- EL6001 from firmware 12
- EL6021, EL6021-0021 from firmware 10.

In order to change the standard time for the transfer rate optimization, the waiting time in bit times 1...254 must be entered in the CoE object 0xB000:01 in the hi-byte 0x24 and in the low-byte

Example: 0x10 (16_{dec}) for a waiting time of 16 bit times:

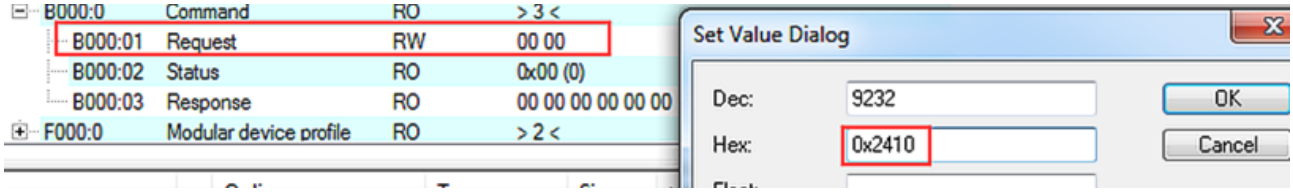


Fig. 34: Example for setting the waiting time of 16 bit times

At 115.2 kBaud this corresponds to 139 µs: (1/115200 [bit/s]) * 16 [bit]. After a gap of this length, the data is automatically copied to the process image.

Continuous transmission of data

Usually the EL60xx terminal automatically decides when to send the data bytes contained in the buffer. For many applications it is helpful to have a continuous data stream. For this purpose, the Beckhoff EL60xx terminals feature the ‘Enable send FIFO data continuous’ setting in the Settings object. If this switch is set.

- The internal send buffer (128 bytes) must be filled first. To this end, data will be sent from the controller to the terminal as in a normal data transfer.
- Data transfer from the buffer commences with a rising edge of the bit “Send continuous”
- If the data has been transferred, the terminal informs the controller by setting the “InitAccepted” bit. “Init accepted” is cleared with “Send continuous”.

This setting enables up to 128 bytes to be transferred without long delays, even with slower EtherCAT cycle times and high baud rates.

The terminal tries to leave as little distance as possible between the telegrams by seamlessly following each stop bit with the next start bit. Nevertheless, in two-channel mode high baud rates may result in short pauses between telegrams. A 20-byte data block may be sent in two blocks of 5 and 15 or 7 and 13 telegrams, for example, or with a different distribution. In a two-channel terminal channel 1 is given priority.

Prioritization

Since received data normally cannot be repeated, it has a higher priority 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.

Command mode

From firmware 06 / revision -0018 the EL6021 supports the so-called command mode. Certain terminal functions can be used and controlled through optional combination and/or sequence of commands. The following functions are currently supported:

- Multi-data frame feature: (from firmware 06) change of coding during ongoing data communication (sending)
  To this end the send buffer of the terminal should be filled with the bytes to be sent (note the maximum buffer size). As soon as the send process is started via the control word, the first n bytes are sent based on coding A, the remaining bytes in the buffer are sent with coding B. The buffer can then be refilled and sent accordingly. Example: The first byte is sent with mark parity, the remaining bytes with space parity.
  Sequence:
  - After each startup/restart of the EL60xx EtherCAT slave the parameterization has to be repeated. The command function is not stored in the event of a power failure.
  - Activation of SendContinuous mode through writing of 0x1 after 0x8000:04
  - Activation of the multi-data frame feature through writing of 0x2001 after request 0xB000:01
    Check that 0xB000:02 = 0
  - Specify coding A through writing of 0x2100 + [date frame code] after 0xB000:01
    Example: 8E2 = 0x12 --> write value 0x2112
    Check that 0xB000:02 = 0
  - Specify n bytes to be sent in coding A through writing of 0x2200 + [n] after 0xB000:01
    Check that 0xB000:02 = 0
  - Specify coding B through writing of 0x2300 + [date frame code] after 0xB000:01
    Check that 0xB000:02 = 0
  - Start sending through rising edge of Ctrl *SendContinuous*
  - Terminal signals completed transfer through *InitAccepted* = 1. *InitAccepted* is reset with *SendContinuous*.

  CmdNameMeaning0x2000 + ControlByteCmdControlByteBit 0: Enable MultiDataFrame feature Bit 1..7: do not use0x2100 + Valuefirst data frameselect preferred value from data frame table0x2200 + NoOfBytesNo of bytesNo of bytes transfered with first data frame0x2300 + Valuesecond data frameselect prefered value from data frame table

  Note: In this mode coding 8M1 and 8S1 is also supported, although this coding cannot be selected via 0x8000:15.
- Further features can be implemented on request

Data transfer examples

Initialization

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

Note: x80n0:01 "Enable RTS/CTS" is TRUE by default, the firmware expects signals here. If these lines are not occupied, this setting must be set to FALSE for this example.

Procedure:

1. Set “Init request” to 1
2. The terminal confirms successful initialization by setting “Init accepted”
3. Reset “Init request”
4. The terminal sets “Init accepted” to 0

The terminal is now ready for data exchange.

Data transmission from the controller to the terminal (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 terminal acknowledges receipt of the data by changing the state of the “Transmit accepted” bit.

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

1. The terminal 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”

6.2 Hints regarding TcVirtualComDriver

In case the EL60xx is to cooperate with the TwinCAT Virtual Serial COM Driver, here are a few hints:

Usually the customer-specific higher-level Windows application sets the desired feature of the COM interface according to the application, e. g. 2400 baud and 7N2 coding. Therefore, the customer-specific entries made *before* parameterization in the CoE StartUp entries, the Settings dialogue, the VirtualComPort or the device CoE are usually irrelevant.

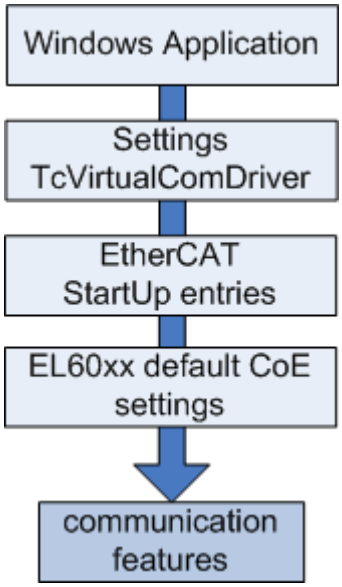


Fig. 35: Each higher level (if available or activated) dictates the communication features to the level below it.

NOTICE

TcVirtualComDriver version

Check whether the desired Baud rate/coding combination of the TcVirtualComDriver version and EL/EP terminal used by you is also supported. If this is not the case:

- it may not be possible to open the COM port
- an ADS error message may appear in the ADS logger in the System Manager
- a CoE error message may appear in the ADS logger

The TcVirtualComDriver supports ExplicitBaudrate (CoE 0x80n0:1B) from version 1.18.

If the COM application has then set the features (such as baud rate) during the runtime, the correct setting can be checked against it in the Online CoE, see fig.

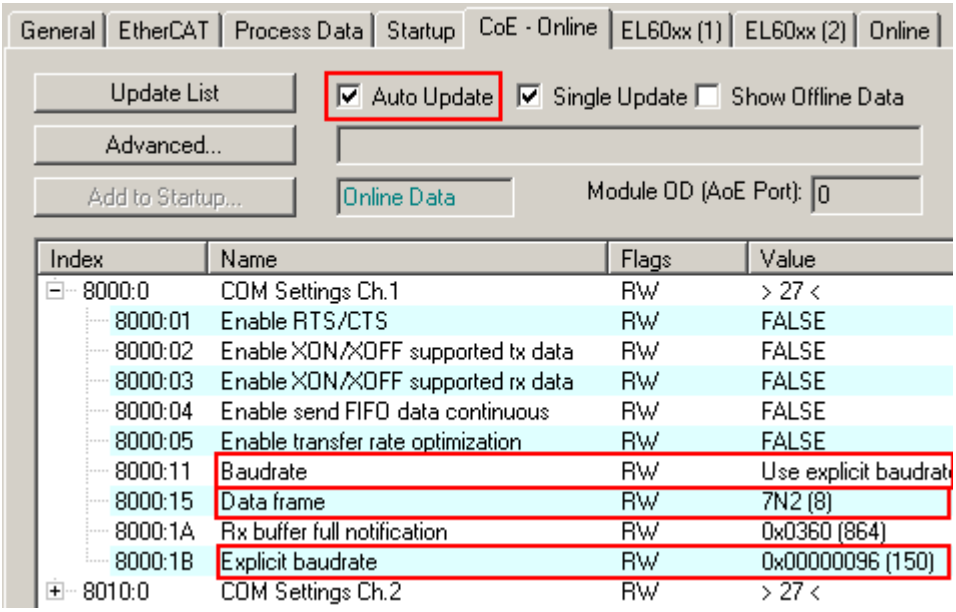


Fig. 36: Checking the settings desired by the COM application in the CoE

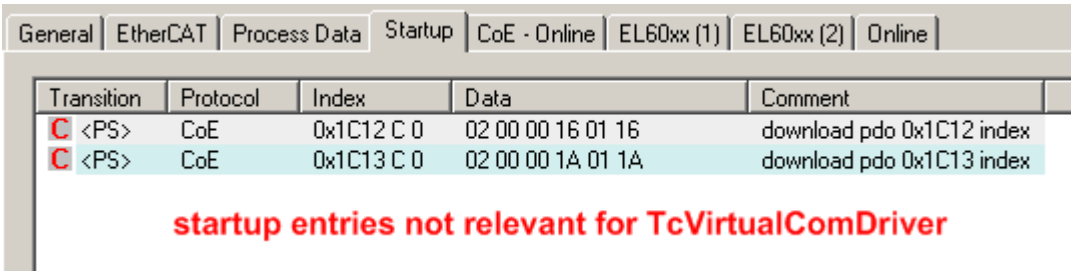


Fig. 37: Default startup entries of a EL6002 (example) – only the default entries (in this case 0x1C12 and 0x1C13) are required

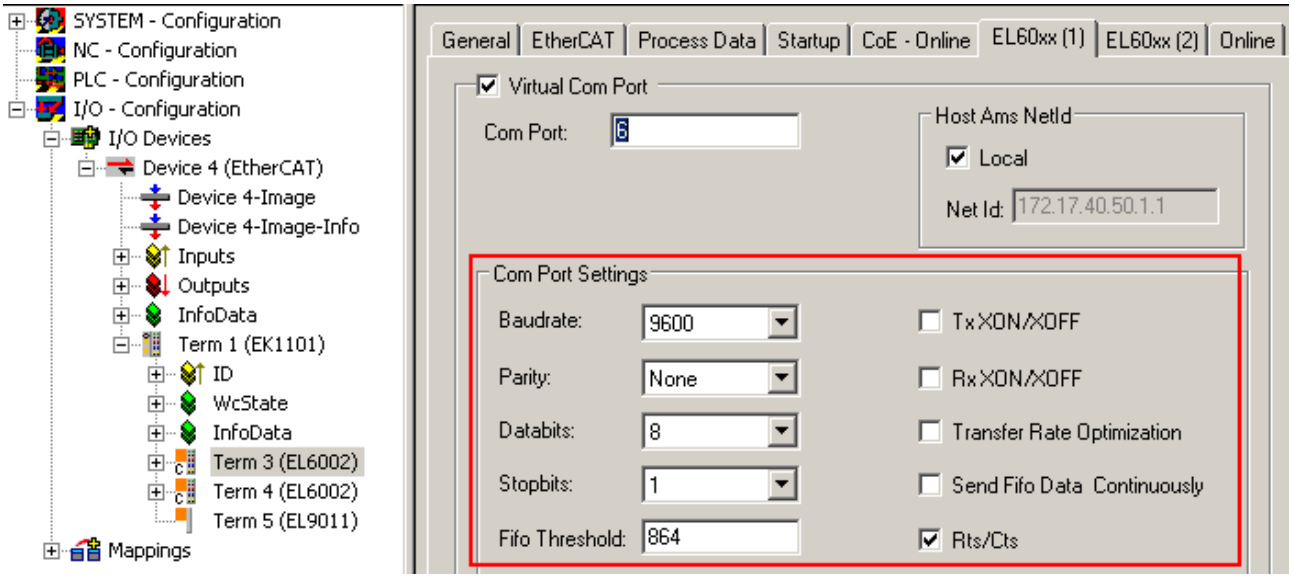


Fig. 38: VirtualComDriver Settings (example)

The settings are relevant only if no specification of the COM features is made or can be made from the application. The COM port is opened with these default settings.

Exception: *TransferRateOptimization, SendFifoDataContinously* and *FiFoThreshold* are to be set only from TwinCAT.

6.3 Communication features

Central features of the EL60xx devices for serial communication are baud rate and data telegram structure/ coding on the field side. The devices differ in the available features. For a clearer overview, the devices are listed here with the parameters available according to the firmware/hardware revision.

Check on the Beckhoff web page whether more up-to-date documentation is available.

EL6001								
Baud rate [0x80n0:11 [▶ 95]]			**Data frame [0x80n0:15 [▶ 95]]**			**Extended data frame [0x80n0:1C [▶ 95]]**		
Baud rate [baud]	**CoE Value**	**from FW/HW**	**Coding**	**CoE Value**	**from FW/ HW**	**Coding**	**CoE Value**	**from FW/HW**
2400	4		7E1	1		7E1	1	
4800	5		7O1	2		7O1	2	
9600	6		8N1	3		8N1	3	
12000	14	FW07	8E1	4		8E1	4	
14400	15	FW07	8O1	5		8O1	5	
19200	7		7N2	8		7N2	8	
38400	8		7E2	9		7E2	9	
57600	9		7O2	10		7O2	10	
115200	10		8N2	11		8N2	11	
104167	Explicit Baud rate [▶ 74] [0x8000:1B]	FW08	8E2	12		8E2	12	
			8O2	13		8O2	13	
1000… 115200 *)	Explicit Baud rate [▶ 74] [0x8000:1B]	FW11	Use extended data frame (Subindex 0x1C)	15	FW08	9N1	32	FW08

*) Es ist jede ganzzahlige Baud rate als int32 Wert einsetzbar. Der Frequenzfehler über alle Baud raten ist < 0,6 %.

EL6002					
Baud rate [0x80n0:11 [▶ 116]]			**Data frame [0x80n0:15 [▶ 116]]**		
Baud rate [baud]	**CoE Value**	**from FW/HW**	**Coding**	**CoE Value**	**from FW/HW**
110	Explicit Baud rate [▶ 74] [0x80n0:1B]	FW03	7E1	1	
150	Explicit Baud rate [▶ 74] [0x80n0:1B]	FW03	7O1	2	
300	1		8N1	3	
600	2		8E1	4	
1200	3		8O1	5	
2400	4		7N2	8	FW03
4800	5		7E2	9	
9600	6		7O2	10	
19200	7		8N2	11	
38400	8		8E2	12	
57600	9		8O2	13	
115200	10				

EL6021								
Baud rate [0x80n0:11 [▶ 95]]			**Data frame [0x80n0:15 [▶ 95]]**			**Extended data frame [0x80n0:1C [▶ 95]]**		
Baud rate [baud]	**CoE Value**	**from FW/HW**	**Coding**	**CoE Value**	**from FW/HW**	**Coding**	**CoE Value**	**from FW/HW**
2400	4		7E1	1		7E1	1	
4800	5		7O1	2		7O1	2	
9600	6		8N1	3		8N1	3	
12000	14	FW07	8E1	4		8E1	4	
14400	15	FW07	8O1	5		8O1	5	
19200	7		7N2	8		7N2	8	
38400	8		7E2	9		7E2	9	
57600	9		7O2	10		7O2	10	
115200	10		8N2	11		8N2	11	
104167	Explicit Baud rate [▶ 74] [0x8000:1B]	FW08	8E2	12		8E2	12	
			8O2	13		8O2	13	
1000… 115200 *)	Explicit Baud rate [▶ 74] [0x8000:1B]	FW11	Use extended data frame (Subindex 0x1C)	15	FW08	9N1	32	FW08

*) any integer baud rate can be set as an int32 value. The frequency error across all baud rates is < 0.6%.

EL6022					
Baud rate [0x80n0:11 [▶ 116]]			**Data frame [0x80n0:15 [▶ 116]]**		
Baud rate [baud]	**CoE Value**	**from FW/HW**	**Coding**	**CoE Value**	**from FW/HW**
300	1		7E1	1	
600	2		7O1	2	
1200	3		8N1	3	
2400	4		8E1	4	
4800	5		8O1	5	
9600	6		7E2	9	
19200	7		7O2	10	
38400	8		8N2	11	
57600	9		8E2	12	
115200	10		8O2	13	

6.4 LIN Master Feature EL6001

Description of “LIN master support” (from FW10)

The EL6001 realizes a bus interface for RS232 levels. From software version 10, the EL6001 features an auxiliary function for sending and receiving LIN telegrams. This function does not cover the formal treatment of the protocol layers for LIN (“protocol stack”) or a physical LIN connection. Instead, the EL6001 can deal with the prolonged start/end mark of a LIN telegram in write and read direction.

”Above” the terminal a LIN stack is required that is implemented in the PLC, for example, “below” the terminal an RS232 <-> 5/12 V level converter e. g. “RS232-LIN” is required.

Before using this function, please check the suitability in your system, since the EL6001 does not have a fully comprehensive LIN implementation!

A complete LIN frame results from a master query with directly following slave response. The master sends the master frame and receives the slave data. With activated LIN functionality of the EL6001 a “sync break” and a “sync byte” is output before the “protected identifier” (PID).
In fact the EL6001 terminal is therefore able to operate under LIN as follows

- EL6001 operating as LIN master, sends data to the Slave and receives the response.
- EL6001 operating as LIN slave: the EL6001 terminal cannot respond to the master telegram in time! Therefore
  - receive data from the master: yes
  - send data to the master: no

Activation

The addition of a “sync break” and “sync field” for sending, which is required for LIN, can be activated via the “Command” object. To this end, enter the value 0x3000 in the CoE object 0xB000:01 [▶ 113]. The response can be read in CoE object 0xB000:03 with the values 0x01 0x00 0x00 0x4C 0x49 0x4E (4C 49 4E = ASCII “LIN”) as confirmation of the activation of this function. The terminal is then ready for receiving frames on the LIN bus containing the 13-bit “sync break” and the “sync field” (0x55), and to make only the following information in the receive data and to output these IDs during sending.

This setting is not permanently stored in the terminal and has to be written again after each new startup. The startup list can be used to circumvent this.

User-specific baud rate

Any already available baud rate and or the baud rate 10417 can be entered in the object “Explicit baudrate” (0x8000:1B).

Further parameters

The following settings within CoE object 0x8000 [▶ 95] are necessary to ensure correct function in LIN mode:

Index:Subindex	Name	Value
0x8000:01	Enable RTS/CTS	FALSE
0x8000:02	Enable XON/XOFF supported tx data	FALSE
0x8000:03	Enable XON/XOFF supported rx data	FALSE
0x8000:04	Enable send FIFO data continuous	FALSE
0x8000:05	Enable transfer rate optimization	TRUE
0x8000:15	Dataframe	3 (8N1)
0x8000:1A	Rx buffer full notification	See objects for commissioning [▶ 95].

Operation

The application of the control and status words remains unchanged in LIN mode.

In LIN mode, the EL6001 automatically precedes the user data with the “Break field” and the “Sync byte” field. During receiving these two fields are automatically removed.

The process image should be used as follows:

Process data	Contents
Ctrl.Output length	Number of user data bytes (n) + 2
Data Out 0	Protected ID field (PID = Protected identifier field)
Data Out 1	Data byte 1
Data Out n	Data byte n
Data Out n+1	Checksum

EL6001 LIN example

The following LIN communication example is intended to illustrate that participation of a PLC controller via the EL6001 terminal in a LIN cluster is possible by activating the complementary LIN functionality via the 0xB000 CoE “command” object in combination with a physical conversion of RS-232 to the LIN bus.

Hardware connection based on a UART-LIN converter

The implementation of the physical layer as a basic prerequisite for LIN communication includes the provision of one-wire bus with corresponding signal levels 0V / 12V. This is best done with an RS232-LIN converter, which is connected to a sub-D connector and on the other side provides three poles for ground, power supply and the electrical connection to the one-wire LIN-bus.

The use of an RS232-LIN converter requires a sub-D 9-pin connector to be connected to the EL6001 terminal, as shown in the diagram:

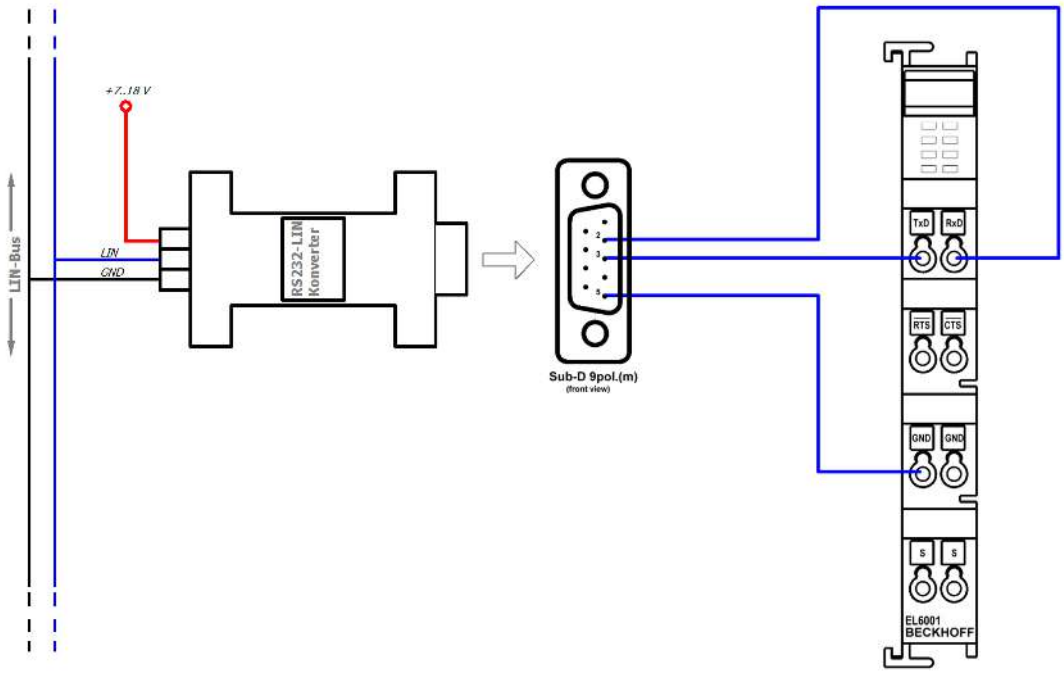


Fig. 39: RS232-LIN sub-D connector connected to the EL6001 terminal

Electrical connections between the EL6001 terminal and the sub-D 9-pin connector:

- RxD →pin 2
- TxD →pin 3
- GND → pin 5

Programming in ST

A LIN communication can be represented by a PLC “master task”, whereby a master sends an “unconditional frame”, as usual with a LIN bus, and only outputs the PID (protocol identifier) of a known slave node on the LIN bus, thereby generally requesting data from the respective slave. The chapter on commissioning contains a relevant programming example [▶ 82].

In this example the slave has the ID 0x07; the master therefore sends the ID 0x07 on the bus, including the calculated parity, resulting in the PID field 0x47. The message on the LIN bus then looks as follows:

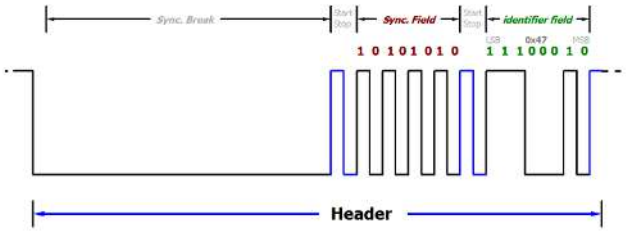


Fig. 40: LIN frame example: Query from master to node with ID 0x07

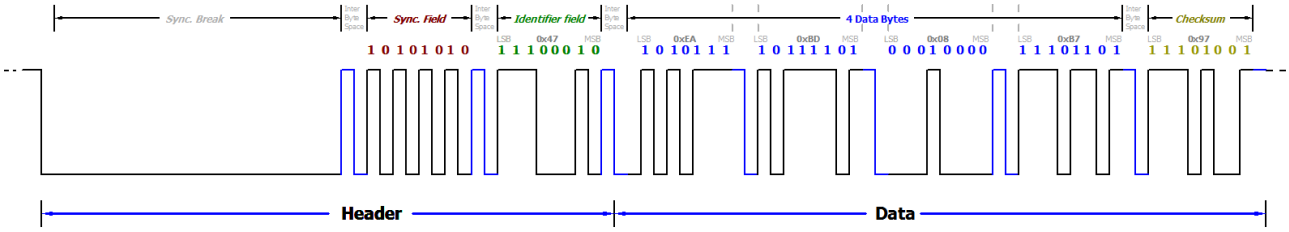


Fig. 41: LIN frame example: ID0x07 with data 0xEA,0xBD,0x08,0xB7 + checksum 0x97

Oscilloscope recordings LIN frame with ID 0x07:

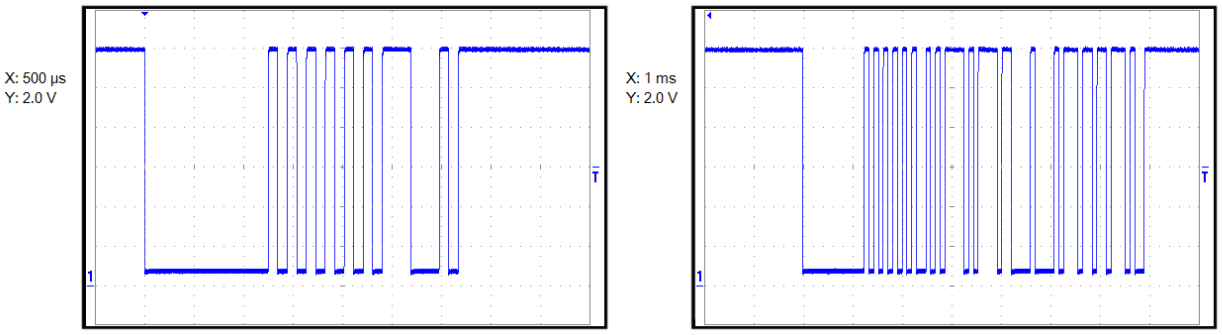


Fig. 42: Left: Query on the LIN bus with PID 0x47, right: LIN frame with the same PID and data, including checksum

6.5 Example programs

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.

6.5.1 Sample program 1

https://infosys.beckhoff.com/content/1033/el600x_el602x/Resources/1719190795.zip

Connection of a serial bar code scanner

In this example a bar code reader is connected to the EL6001. 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
- Barcode 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 example program

The application examples 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 EK1100 EtherCAT coupler, EL6001 terminals
- Serial barcode scanner with 9-pole sub-D connector

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 in accordance with fig. 1 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 one) under System configuration, I/O configuration, I/O devices, Device (EtherCAT); on the “Adapter” tab choose “Search...”, select the appropriate adapter and confirm (see Figures)

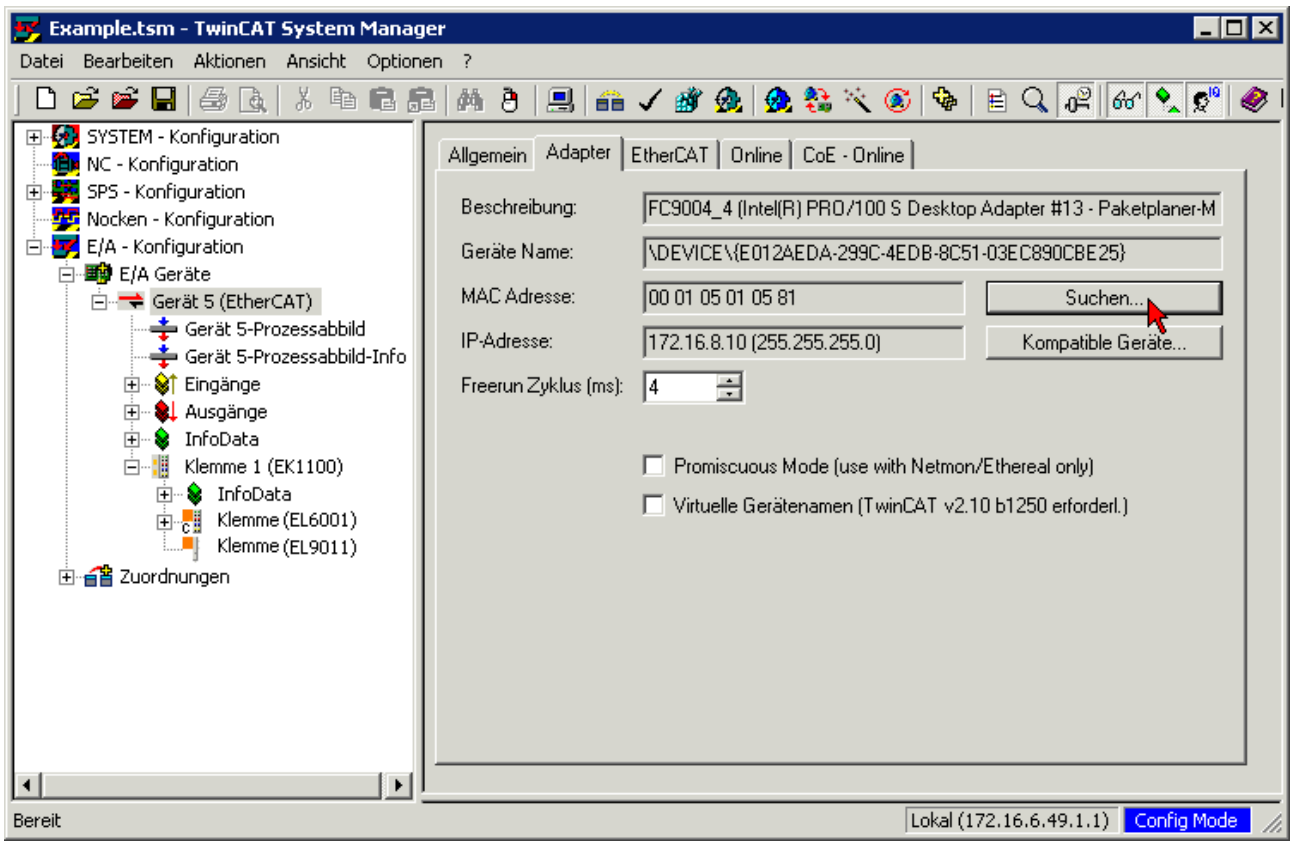


Fig. 43: Searching the Ethernet adapter

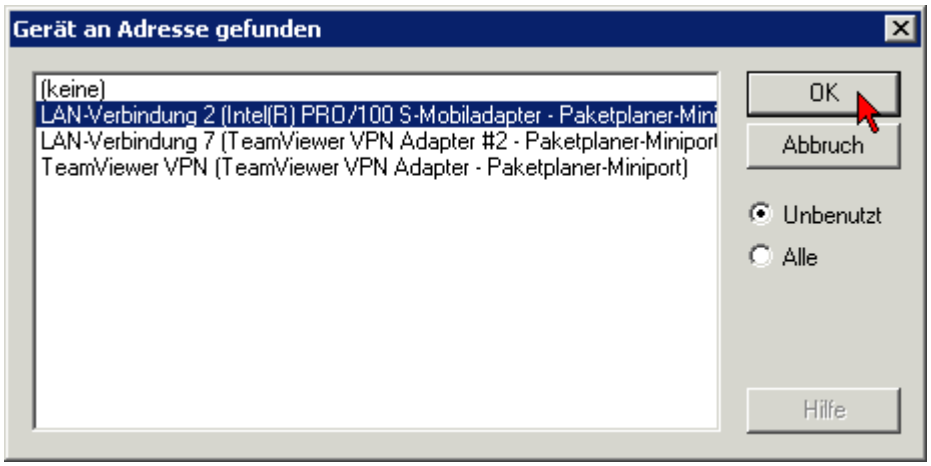


Fig. 44: Selection and confirmation of the Ethernet adapter

Activation of the configuration and confirmation:



Fig. 45: Activation of the configuration



Fig. 46: Confirming the activation of the configuration

- Confirming new variable mapping, restart in RUN mode:

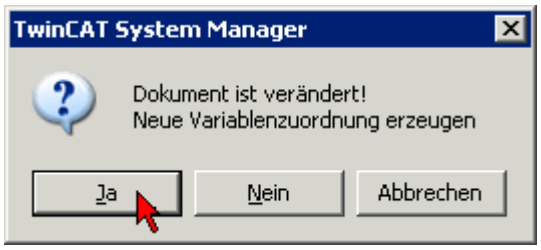


Fig. 47: Generating variable mapping



Fig. 48: Restarting TwinCAT in RUN mode

- In TwinCAT PLC, under the “Project” menu, select “Rebuild all” to compile the project:

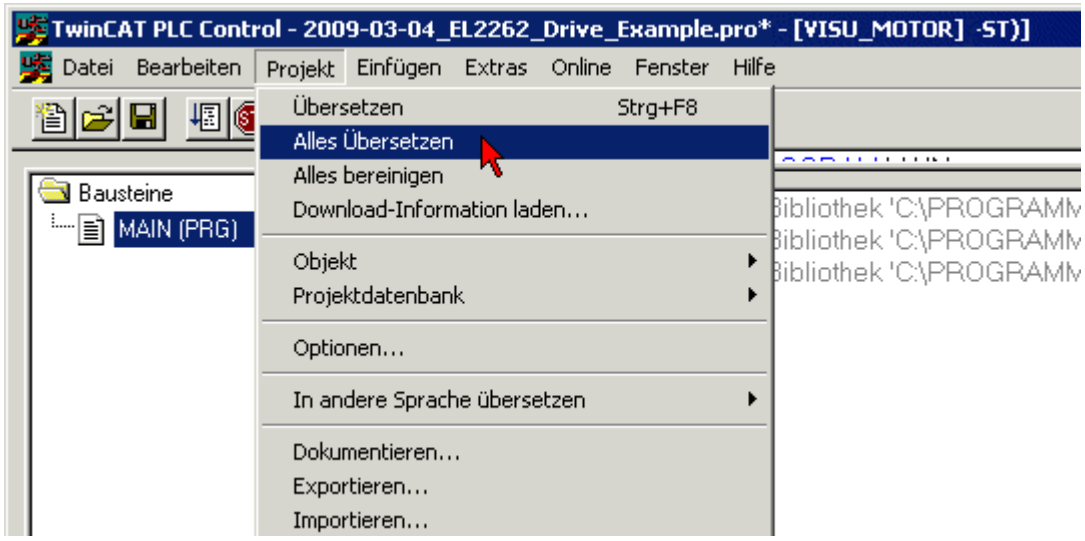


Fig. 49: Compile project

- In TwinCAT PLC: log in with the “F11” button, confirm loading the program, run the program with the “F5” button:

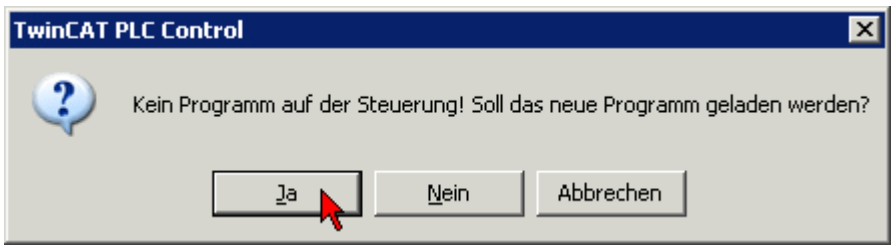


Fig. 50: Confirming program start

- After the character 0x13 has been received, the barcode is stored in “LastBarcode”

```
0001    Receive(
0002        Suffix:= '$0D',              (* RETURN*)
0003        Timeout:= T#1s,
0004        ReceivedString:= Barcode,
0005        RXbuffer:= RxBuffer,
0006        StringReceived=> BarcodeReceived,
0007        Busy=> ReceiveBusy,
0008        Error=> ReceiveErrorID,
0009        RxTimeout=> ReceiveTimeout );
0010
0011    NewBarcode(CLK  := Receive.StringReceived );
0012    IF NewBarcode.Q THEN
0013        LastBarcode := Barcode;
0014    END_IF
```

```
Receive.StringRec... = FALSE    Receive.Busy = FALSE


Barcode = '544900017888$R'

BarcodeReceived = FALSE
ReceiveBusy = FALSE
ReceiveErrorID = COMERRO...
ReceiveTimeout = FALSE

Receive.StringRec... = FALSE
NewBarcode.Q = FALSE
LastBarcode = '544900017888...   Barcode = '544900017888$R'
```

Fig. 51: Received barcode

6.5.2 Sample program 2

https://infosys.beckhoff.com/content/1033/el600x_el602x/Resources/1719192971.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 bit stream 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 by a Beckhoff EL/KL60xx serial data exchange terminal.

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

6.5.3 Sample program 3 (LIN)

Download (TwinCAT 3 example program):

https://infosys.beckhoff.com/content/1033/el600x_el602x/Resources/1805853195.zip

Global variables for LIN master example program

```
VAR_GLOBAL
nSetBaudrate : UINT := 10417; // Für Rx-Delay-Berechnung
(* I/O variables for EL6001 terminal acting as Master*)
COMin_EL6001_MASTER AT %I* : EL6inData22B;(* linked to the EL6001 in the TwinCAT System Manager *)
COMout_EL6001_MASTER AT %Q*: EL6outData22B;(* linked to the EL6001 in the TwinCAT System Manager *)
RxBuffer_MASTER : ComBuffer;(* Receive data buffer; used with all receive function blocks *)
TxBuffer_MASTER : ComBuffer;(* Transmit data buffer; used with all receive function blocks *)
END_VAR
```

Data types for LIN master-slave example program

```
TYPE tDataFrame : ARRAY[0..8] OF BYTE; END_TYPE // Datentyp für ein LIN-Frame
```

Function for adding data length information and parity bits:

Declaration part:

```
FUNCTION F_ADD_LIN_NODE_PARITY : BYTE
VAR_INPUT
nNodeID:BYTE; // Inputvariable: node-Id
nReqLen:BYTE; // Inputvariable: length-identification: 2,4,8 Byte-Frame
END_VAR
VAR
bParity0:BYTE; // Internal intermediate value parity 0
bParity1:BYTE; // Internal intermediate value parity 1
nPrepId:BYTE; // Intermediate value for PID
END_VAR
```

Execution part:

```
nPrepId := nNodeID OR SHL(nReqLen,4);

bParity0 :=
(nNodeID AND 2#0001)
XOR (SHR((nNodeID AND 2#0010), 1))
XOR (SHR((nNodeID AND 2#0100), 2))
XOR (SHR((nNodeID AND 2#0001_0000), 4));

bParity1 := 16#01 AND (NOT(
SHR((nNodeID AND 2#0010), 1)
XOR (SHR((nNodeID AND 2#1000),3))
XOR (SHR((nNodeID AND 2#0001_0000), 4))
XOR (SHR((nNodeID AND 2#0010_0000), 5))));

F_ADD_LIN_NODE_PARITY := nPrepId OR SHL(bParity0,6) OR SHL(bParity1,7);
```

Function for calculating the checksum (conventional method)

Declaration part:

```
FUNCTION F_CALC_LIN_CHKSUM : BYTE
VAR_INPUT
pData:POINTER TO ARRAY[0..10] OF BYTE; // Pointer to source datafield
nLen:BYTE; // Number of bytes used for calculating the checksum
END_VAR
VAR
```

```
i: BYTE; // Counter variable
wResult:WORD; // Resulting output value
END_VAR
```

Execution part:

```
wResult := BYTE_TO_WORD(pData^[0]);
FOR i := 1 TO (nLen-1) DO
wResult := wResult + BYTE_TO_WORD(pData^[i]);
IF wResult > 255 THEN
wResult := wResult - 255;
END_IF
END_FOR
F_CALC_LIN_CHKSUM := WORD_TO_BYTE(NOT(wResult));
```

This master program part should be called from a corresponding separate task. Here, a node in the LIN bus with the ID 0x07 is queried every 200 ms.

Declaration part:

```
PROGRAM EL6001_MASTER
VAR
Timer: TON; // Timer for periodical requests by the master
Send: SendData; // Functionblock of TC2_SerialCom
SendBusy: BOOL; // Flag copy of SendData.Busy
SendErrorID: ComError_t; // Flag-Copy of Error-ID
aDataTX:tDataFrame; // Data frame being send
Receive: ReceiveData; // Functionblock of TC2_SerialCom
LastReceivedDataBytes: tDataFrame; // Copy (Latch) of received data
DataReceived: BOOL; // Flag copy of receive confirmation
ReceiveBusy: BOOL; // Flag copy of reception not ready
ReceiveError: BOOL; // Flag copy of receive error
ReceiveErrorID: ComError_t;// Error-ID copy
ReceiveTimeout: BOOL; // Flag copy of receive-timeout
ReceiveCounter: UDINT := 0; // Number of received frames
aDataRX:tDataFrame; // Receiving data frame buffer
nDataLen:BYTE := 4; // Fixed data length
nState:BYTE := 0; // Initial state (start)
bNodeId_SL1: BYTE := 16#07; // ID of slave node
bReqLen_SL1: BYTE := 0; // Optional entry for 4 Data bytes Value 2
nRxChecksum: BYTE; // Storage of received checksum
nCalcChecksum:BYTE; // Storage of calculated checksum
T_ReceiveDelay:TIME; // Storage of calculated delay time
END_VAR
```

Execution part:

```
(*=====================================================================
Receive data
*)
CASE nState OF
0:
Timer(IN:=TRUE, PT:=T#0.5S); // Call timer for periodical master requests
IF Timer.Q THEN
// Put ID into Tranceive data:
aDataTX[0] := F_ADD_LIN_NODE_PARITY(bNodeId_SL1, bReqLen_SL1);
LastReceivedDataBytes[0] := aDataTX[0];
// Send request to Slave 1 (get Data)
Send(pSendData:= ADR(aDataTX), Length:= 1,
TXbuffer:= TxBuffer_MASTER,
Busy => SendBusy, Error => SendErrorID);
Timer(IN:=FALSE); (* reset timer *)
IF NOT SendBusy THEN // Wait until sending ends
nState := nState + 1;
END_IF
END_IF
1:
// Delaytime by 1/Tbaud * Number of Bytes * (8 Databits + 2 Bit:start-Stop) * 1000ms
T_ReceiveDelay := REAL_TO_TIME((1/DINT_TO_REAL(nSetBaudrate)) * 33 * 1000); // .. for 1 Byte
Timer(IN:=TRUE, PT:=T_ReceiveDelay);
IF Timer.Q THEN
// Wait until ID is send
nState := nState + 1;
Timer(IN:=FALSE); (* reset timer *)
END_IF
2:
Receive(
pReceiveData:= ADR(aDataRX),
SizeReceiveData:= (nDataLen + 1),
RXbuffer:= RxBuffer_MASTER,
```

```
Timeout:= T#1S,
DataReceived=> DataReceived,
busy=> ReceiveBusy,
Error=> ReceiveErrorID,
RxTimeout=> ReceiveTimeout );
IF DataReceived THEN
// DataReceived := FALSE;
ReceiveCounter := ReceiveCounter + 1;
IF NOT ReceiveBusy THEN
// Compare checksum
nRxChecksum := aDataRX[nDataLen];
nCalcChecksum := F_CALC_LIN_CHKSUM(pData := ADR(aDataRX), nLen := nDataLen);
IF(nRxChecksum = nCalcChecksum) THEN
// Response received - clear databuffer:
memset(ADR(LastReceivedDataBytes[1]), 0, (SIZEOF(aDataRX)-1));
// Take-over data when checksum OK:
memcpy(ADR(LastReceivedDataBytes[1]), ADR(aDataRX), (nDataLen +1));
END_IF
nState := 0;
END_IF
ELSE
Timer(IN:=TRUE, PT:=T#0.1S); // Receive-Timeout 100 ms: no data
IF Timer.Q THEN
nState := 0;
END_IF
END_IF
END_CASE
(*=====================================================================
*)
```

Program fast task / RS232 background communication

In this example, third task should be created with as few “cycle ticks” as possible, which deals with the background communication with the EL6001 terminal (as master).

Declaration part:

```
PROGRAM FAST
VAR
(* background communication with the EL6001 as Master device *)
COMportControl_MASTER: SerialLineControl;
COMportControlError_MASTER: BOOL;
COMportControlErrorID_MASTER: ComError_t;
END_VAR
```

Execution part:

```
COMportControl_MASTER(
Mode:= SERIALLINEMODE_EL6_22B,
pComIn:= ADR(COMin_EL6001_MASTER),(* I/O data; see global variables *)
pComOut:= ADR(COMout_EL6001_MASTER),(* I/O data; see global variables *)
SizeComIn:= SIZEOF(COMin_EL6001_MASTER),(* I/O data; see global variables *)
TxBuffer:= TxBuffer_MASTER,(* transmit buffer; see global variables *)
RxBuffer:= RxBuffer_MASTER,(* receive buffer; see global variables *)
Error=> COMportControlError_MASTER,(* indicating an error *)
ErrorID=> COMportControlErrorID_MASTER );(* contains the error-ID *)
```

7 Overview of CoE objects EL6001, EL6021

7.1 Control and status word

Control word

The control word (CW) is located in the output process image, and is transmitted from the controller to the terminal.

Bit	CW.15	CW.14	CW.13	CW.12	CW.11	CW.10	CW.9	CW.8	CW.7	CW.6	CW.5	CW.4	CW.3	CW.2	CW.1	CW.0
Name	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	-	OL2*	OL1*	OL0*	SC	IR	RA	TR

Bits CW.15 to CW.8 are only shown if the large process image is used.
If the small or medium process image is used, only bits CW.7 to CW.0 are shown as control bytes! The number of output bytes available for transfer from the controller to the terminal is displayed by bits OL2* ... OL0*.

Legend

Bit	Name	Description	
CW.15 ... CW8	OL7 ... OL0 (OutLenght)	1_{dec}... 22_{dec}	If the large process image is used: Number of output bytes available for transfer from the controller to the terminal.
		-	If the small/medium process image is used: not shown
CW.7	-	0_{bin}	reserved
CW.6 ... CW.4	OL2* ... OL0* (OutLenght*)	0	If the large process image is used: reserved
		1 ... 6	If the medium process image is used: Number of output bytes available for transfer from the controller to the terminal.
		1 ... 4	If the small process image is used: Number of output bytes available for transfer from the controller to the terminal.
CW.3	SC (SendContinuous)	rise	Continuous sending of data from the FIFO. The send buffer is filled (up to 128 bytes) by the controller. The buffer content is sent with rising edge of bit SC. The terminal acknowledges the data transfer to the controller through setting of bit SW.2. SW.2 is cancelled with CW.3.
CW.2	IR (InitRequest)	1_{bin}	The controller requests terminal for initialization. The send and receive functions are blocked, the FIFO pointers are reset, and the interface is initialized with the values of the responsible objects (baud rate 4073 [▶ 93], data frame 4074 [▶ 93], feature bits 4075 [▶ 94]). The terminal acknowledges completion of the initialization via bit SW.2 [▶ 91] (IA).
		0_{bin}	The controller once again requests the terminal to prepare for serial data exchange.
CW.1	RA (ReceiveAccepted)	toggle	The controller acknowledges receipt of data by changing the state of this bit. Only then new data can be transferred from the terminal to the controller.
CW.0	TR (TransmitRequest)	toggle	Via a change of state of this bit the controller notifies the terminal that the DataOut bytes contain the number of bytes indicated via the OL bits. The terminal acknowledges receipt of the data in the status byte via a change of state of bit SW.0 [▶ 91] (TA). Only now new data can be transferred from the controller to the terminal.

Status word

The status word (SW) is located in the input process image, and is transmitted from terminal to the controller.

Bit	SW.15	SW.14	SW.13	SW.12	SW.11	SW.10	SW.9	SW.8	SW.7	SW.6	SW.5	SW.4	SW.3	SW.2	SW.1	SW.0
Name (small and medium process image)	-	-	-	-	-	-	-	-	-	IL2*	IL1*	IL0*	BUF_F	IA	RR	TA
Name (large process image)	IL7	IL6	IL5	IL4	IL3	IL2	IL1	IL0	-	OVERRUN ERR	FRAMING ERR	PARITY ERR	BUF_F	IA	RR	TA

If the small or medium process image is used, only bits SW.7 to SW.0 are shown as status bytes! The number of input bytes available for transfer from the terminal to the controller is displayed by bits IL2* ... IL0*.

Legend

Bit	Name	Description	
SW.15 ... SW8	IL7 ... IL0 (InLenght*)	1_{dec} ... 22_{dec}	If the large process image is used: Number of input bytes available for transfer from the terminal to the controller.
		-	If the small/medium process image is used: not shown
SW.7	-	0_{bin}	reserved
SW.6	IL2* (InLenght*) OVERRUN ERR	0 ... 1	If the large process image is used: An overrun error has occurred. The affected data item is not loaded into the receive FIFO of the terminal and is lost.
		1 ... 6	If the medium process image is used: Number of input bytes available for transfer from the terminal to the controller.
		1 ... 4	If the small process image is used: Number of input bytes available for transfer from the terminal to the controller.
SW.5	IL1* (InLenght*) FRAMING ERR	0 ... 1	If the large process image is used: A framing error has occurred. The affected data item is not loaded into the receive FIFO of the terminal and is lost.
		1 ... 6	If the medium process image is used: Number of input bytes available for transfer from the terminal to the controller.
		1 ... 4	If the small process image is used: Number of input bytes available for transfer from the terminal to the controller.
SW.4	IL0* (InLenght*) PARITY ERR	0 ... 1	If the large process image is used: A parity error has occurred. The affected data item is not loaded into the receive FIFO of the terminal and is lost
		1 ... 6	If the medium process image is used: Number of input bytes available for transfer from the terminal to the controller.
		1 ... 4	If the small process image is used: Number of input bytes available for transfer from the terminal to the controller.
SW.3	BUF_F	1_{bin}	The reception FIFO is full. All further incoming data will be lost!
SW.2	IA (InitAccepted-Bit)	1_{bin}	Initialization was completed by the terminal.
		0_{bin}	The terminal is ready again for serial data exchange.
SW.1	RR (ReceiveRequest)	toggle	Via a change of state of this bit the terminal notifies the controller that the DataIn bytes contain the number of bytes indicated via the IL bits. The controller has to acknowledge receipt of the data in the control byte via a change of state of bit CW.1 [▶ 90] (RA). Only then new data can be transferred from the terminal to the controller.
SW.0	TA (TransmitAccepted)	toggle	The terminal acknowledges receipt of data by changing the state of this bit. Only now new data can be transferred from the controller to the terminal.

Data transfer examples

The examples use the large process image.

- Initialization:

Control word		Status word		Comment
CW.15 ... CW.8	CW.7 ... CW.0	SW.15 ... SW.8	SW.7 ... SW.0	
xxxx xxxx$_{bin}$	xxxx xxxx$_{bin}$	xxxx xxxx$_{bin}$	xxxx xxxx$_{bin}$	Start of data transmission
0000 0000$_{bin}$	0000 0**1**00$_{bin}$	0000 0000$_{bin}$	0000 0000$_{bin}$	The controller requests terminal initialization
0000 0000$_{bin}$	0000 0100$_{bin}$	0000 0000$_{bin}$	0000 0**1**00$_{bin}$	Command is executed: Terminal initialization is complete
0000 0000$_{bin}$	0000 0**0**00$_{bin}$	0000 0000$_{bin}$	0000 0100$_{bin}$	The controller requests the terminal to prepare for serial data exchange
0000 0000$_{bin}$	0000 0000$_{bin}$	0000 0000$_{bin}$	0000 0**0**00$_{bin}$	Command is executed: Terminal is ready for serial data exchange

- Data transfer from the controller to the terminal:

Control word		Status word		Comment
CW.15 ... CW.8	CW.7 ... CW.0	SW.15 ... SW.8	SW.7 ... SW.0	
0000 0000$_{bin}$	0000 0000$_{bin}$	xxxx xxxx$_{bin}$	0000 x0x0$_{bin}$	Start of data transmission
0000 00**10**$_{bin}$	0000 000**1**$_{bin}$	xxxx xxxx$_{bin}$	0000 x0x0$_{bin}$	The controller requests transmission of **2** bytes by the terminal
0000 0010$_{bin}$	0000 0001$_{bin}$	xxxx xxxx$_{bin}$	0000 x0x**1**$_{bin}$	Command is executed: Terminal has loaded **2** bytes into the transmission FIFO
000**1 0000**$_{bin}$	0000 000**0**$_{bin}$	xxxx xxxx$_{bin}$	0000 x0x1$_{bin}$	The controller requests transmission of **16** bytes by the terminal
0001 0000$_{bin}$	0000 0000$_{bin}$	xxxx xxxx$_{bin}$	0000 x0x**0**$_{bin}$	Command is executed: Terminal has loaded 16 bytes into the transmission FIFO

- Data transfer from the terminal to the controller:

Control word		Status word		Comment
CW.15 ... CW.8	CW.7 ... CW.0	SW.15 ... SW.8	SW.7 ... SW.0	
xxxx xxxx$_{bin}$	0000 000x$_{bin}$	0000 0000$_{bin}$	0000 000x$_{bin}$	Start of data transmission
xxxx xxxx$_{bin}$	0000 000x$_{bin}$	0000 00**11**$_{bin}$	0000 001x$_{bin}$	The terminal requests transfer of **3** bytes from the controller
xxxx xxxx$_{bin}$	0000 001x$_{bin}$	0000 0011$_{bin}$	0000 001x$_{bin}$	Acknowledgement: Controller has received 3 bytes from the reception FIFO
xxxx xxxx$_{bin}$	0000 001x$_{bin}$	000**1 0110**$_{bin}$	0000 00**0**x$_{bin}$	The terminal requests transfer of **22** bytes from the controller
xxxx xxxx$_{bin}$	0000 00**0**x$_{bin}$	0001 0110$_{bin}$	0000 000x$_{bin}$	Acknowledgement: Controller has received 22 bytes from the reception FIFO

7.2 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

The terminal is parameterized via the CoE - Online tab (with a double-click on the respective object, see below).

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization during commissioning
- Objects for indicating internal settings (may be fixed)
- Profile specific objects, which represent the status displays of the inputs and outputs (from hardware version 03 [▶ 135])

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

7.2.1 Objects for commissioning

Index 0x1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default parameters	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 0x001	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})

Use 0x8000 COM Settings for new implementations

For new implementations, it is recommended to use the settings listed below in 0x8000 COM Settings instead of the Settings objects 0x407_.

Index 0x4073 Baud rate

Index (hex)	Name	Meaning	Data type	Flags	Default
4073:0	Baud rate	Detailed information can be found in chapter Communication features [▶ 76]	UINT16	RW	0x0006 (6_{dec})

Index 0x4074 Data frame

Index (hex)	Name	Meaning	Data type	Flags	Default
4074:0	Data frame	Detailed information can be found in chapter Communication features [▶ 76]	UINT16	RW	0x0003 (3_{dec})

Index 0x4075 Feature bits

Index (hex)	Name	Meaning	Data type	Flags	Default
4075:0	Feature bits	Length of this object	UINT8	RO	0x06 (6_{dec})
4075:01	**EL6001:** Enable RTS/CTS	0: RTS/CTS not enabled 1: RTS/CTS enabled	BOOLEAN	RW	0x00 (0_{dec})
	EL6021: Enable half duplex	0: Full-duplex mode 1: Half-duplex mode			
4075:02	Enable XON/ XOFF supported tx data	0: XON/XOFF is not supported for send data 1: XON/XOFF is supported for send data	BOOLEAN	RW	0x00 (0_{dec})
4075:03	Enable XON/ XOFF supported rx data	0: XON/XOFF is not supported for receive data 1: XON/XOFF is supported for receive data	BOOLEAN	RW	0x00 (0_{dec})
4075:04	**EL6001:** Enable send FIFO data continous	0: No continuous sending of data from the FIFO 1: The send buffer is filled (up to 128 bytes) by the controller. The buffer content is sent with rising edge of bit CW.3 [▶ 90] in the control word. The terminal acknowledges the data transfer to the controller through setting of bit SW.2 [▶ 91] in the status word. SW.2 is reset with CW.3.	BOOLEAN	RO	0x00 (0_{dec})
	EL6021: Enable point to point connection (RS422)	0: Point-to-point connection disabled 1: Point-to-point connection enabled	BOOLEAN	RO	0x00 (0_{dec})
4075:05	**EL6001:** Enable transfer rate optimization	0: Transfer rate optimization switched off 1: Transfer rate optimization switched on: The content of the input buffer is automatically transferred into the process image if • no further byte was received for approx. 16 bit times (i.e. the time it would have taken to receive 2 bytes) after data were received; • the input buffer is full	BOOLEAN	RO	0x01 (1_{dec})
	EL6021: Enable send FIFO data continuous	0: No continuous sending of data from the FIFO 1: The send buffer is filled (up to 128 bytes) by the controller. The buffer content is sent with rising edge of bit CW.3 [▶ 90] in the control word. The terminal acknowledges the data transfer to the controller through setting of bit SW.2 [▶ 91] in the status word. SW.2 is reset with CW.3.	BOOLEAN	RO	0x00 (0_{dec})
4075:06	**EL6021 only:** Enable transfer rate optimization	0: Transfer rate optimization switched off 1: Transfer rate optimization switched on: The content of the input buffer is automatically transferred into the process image if • no further byte was received for approx. 16 bit times (i.e. the time it would have taken to receive 2 bytes) after data were received; • the process image is filled	BOOLEAN	RW	0x01 (1_{dec})

Index 0x4076 Rx buffer full notification

Index (hex)	Name	Meaning	Data type	Flags	Default
4076:0	Rx buffer full notification	The value determines the number of data in the receive FIFO from which bit SW.3 [▶ 91] (BUF_F) is set in the status byte.	UINT16	RW	0x0000 (0_{dec})

Index 0x8000 COM settings [from hardware version 03]

Index (hex)	Name	Meaning		Data type	Flags	Default
8000:0	COM Settings	Max. SubIndex (hex)		UINT8	RO	0x26 (38_{dec})
8000:01**	Enable RTS/CTS	FALSE	RTS/CTS not enabled	BOOLEAN	RW	0x01 (1_{dec})
		TRUE	RTS/CTS enabled			
8000:02	Enable XON/ XOFF supported tx data	FALSE	XON/XOFF is not supported for send data	BOOLEAN	RW	0x00 (0_{dec})
		TRUE	XON/XOFF is supported for send data			
8000:03	Enable XON/ XOFF supported rx data	FALSE	XON/XOFF is not supported for receive data	BOOLEAN	RW	0x00 (0_{dec})
		TRUE	XON/XOFF is supported for receive data			
8000:04	Enable send FIFO data continuous	FALSE	No continuous sending of data from the FIFO	BOOLEAN	RW	0x00 (0_{dec})
		TRUE	Continuous sending of data from the FIFO enabled: The send buffer is filled (up to 128 bytes) by the controller. The filled buffer content is sent on rising edge of bit “SendContinious [▶ 112]”. The terminal acknowledges the data transfer to the controller through setting of bit "InitAccepted". “InitAccepted [▶ 111]” is reset with “SendContinuous”.			
8000:05	Enable transfer rate optimization	FALSE	Transfer rate optimization switched off	BOOLEAN	RW	0x01 (1_{dec})
		TRUE	Transfer rate optimization switched on: The content of the input buffer is automatically transferred into the process image if • no further byte was received for approx. 16 bit times (i.e. the time it would have taken to receive 2 bytes) after data were received; • the process image is filled			
8000:06***	Enable half duplex	FALSE	Full-duplex mode	BOOLEAN	RW	0x00 (0_{dec})
		TRUE	Half-duplex mode			
8000:07***	Enable point to point connection (RS422)	FALSE	Point-to-point connection disabled	BOOLEAN	RW	0x00 (0_{dec})
		TRUE	Point-to-point connection enabled			
8000:11	Baud rate	Detailed information can be found in chapter Communication features [▶ 76]		BIT4	RW	0x06 (6_{dec})
8000:15	Data frame	Detailed information can be found in chapter Communication features [▶ 76]		BIT4	RW	0x03 (3_{dec})
8000:1A	Rx buffer full notification	The value specifies the number of data in the receive FIFO, from which the bit “buffer full [▶ 113]” 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. Only the baud rates specified in Communication features [▶ 76] are supported. Changes to this object are also adopted into the objects 0x8000:11 and 4073		UINT32	RW	0x00000384 (9600_{dec})
8000:1C**	Extended data frame	In this object special formats can also be selected in addition to the usual data frames (e.g. 9N1). Changes to this object are also adopted in the objects 0x8000:15 and 0x4074.		ENUM16	RW	0x0003 (3_{dec})

**) EL6001 from FW11, EL6021 from FW09
***) only EL6021

7.2.2 Standard objects (0x1000-0x1FFF)

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

Index 0x1000 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 0x1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL6001 EL6021

Index 0x1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	04

Index 0x100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	05

Index 0x1018 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	0x17853052 (394604626_{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	0x00100000 (1048576_{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 0x10F0 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 0x1400 RxPDO-Par Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1400:0	RxPDO-Par Outputs	PDO Parameter RxPDO 1	UINT8	RO	0x06 (6_{dec})
1400:06	Exclude RxPDOs	Specifies the RxPDOs (index 0x of RxPDO mapping objects) that must not be transferred together with RxPDO 1	OCTET-STRING[6]	RO	01 16 02 16 04 16

Index 0x1401 RxPDO-Par Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1401:0	RxPDO-Par Outputs	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1401:06	Exclude RxPDOs	Specifies the RxPDOs (index 0x of RxPDO mapping objects) that must not be transferred together with RxPDO 2	OCTET-STRING[6]	RO	00 16 02 16 04 16

Index 0x1402 RxPDO-Par Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1402:0	RxPDO-Par Outputs	PDO Parameter RxPDO 3	UINT8	RO	0x06 (6_{dec})
1402:06	Exclude RxPDOs	Specifies the RxPDOs (index 0x of RxPDO mapping objects) that must not be transferred together with RxPDO 3	OCTET-STRING[6]	RO	00 16 01 16 04 16

Index 0x1600 RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	RxPDO-Map Outputs	PDO Mapping RxPDO 1	UINT8	RO	0x04 (4_{dec})
1600:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3001 (Outputs), entry 0x01 (Ctrl))	UINT32	RO	0x3001:01, 8
1600:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3001 (Outputs), entry 0x02 (Data Out 0))	UINT32	RO	0x3001:02, 8
1600:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3001 (Outputs), entry 0x03 (Data Out 1))	UINT32	RO	0x3001:03, 8
1600:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3001 (Outputs), entry 0x04 (Data Out 2))	UINT32	RO	0x3001:04, 8

Index 0x1601 RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	RxPDO-Map Outputs	PDO Mapping RxPDO 2	UINT8	RO	0x06 (6_{dec})
1601:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3002 (Outputs), entry 0x01 (Ctrl))	UINT32	RO	0x3002:01, 8
1601:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3002 (Outputs), entry 0x02 (Data Out 0))	UINT32	RO	0x3002:02, 8
1601:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3002 (Outputs), entry 0x03 (Data Out 1))	UINT32	RO	0x3002:03, 8
1601:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3002 (Outputs), entry 0x04 (Data Out 2))	UINT32	RO	0x3002:04, 8
1601:05	SubIndex 0x005	5. PDO Mapping entry (object 0x3002 (Outputs), entry 0x05 (Data Out 3))	UINT32	RO	0x3002:05, 8
1601:06	SubIndex 0x006	6. PDO Mapping entry (object 0x3002 (Outputs), entry 0x06 (Data Out 4))	UINT32	RO	0x3002:06, 8

Index 0x1602 RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	RxPDO-Map Outputs	PDO Mapping RxPDO 3	UINT8	RO	0x17 (23_{dec})
1602:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3003 (Outputs), entry 0x01 (Ctrl))	UINT32	RO	0x3003:01, 16
1602:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3003 (Outputs), entry 0x02 (Data Out 0))	UINT32	RO	0x3003:02, 8
1602:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3003 (Outputs), entry 0x03 (Data Out 1))	UINT32	RO	0x3003:03, 8
1602:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3003 (Outputs), entry 0x04 (Data Out 2))	UINT32	RO	0x3003:04, 8
1602:05	SubIndex 0x005	5. PDO Mapping entry (object 0x3003 (Outputs), entry 0x05 (Data Out 3))	UINT32	RO	0x3003:05, 8
1602:06	SubIndex 0x006	6. PDO Mapping entry (object 0x3003 (Outputs), entry 0x06 (Data Out 4))	UINT32	RO	0x3003:06, 8
1602:07	SubIndex 0x007	7. PDO Mapping entry (object 0x3003 (Outputs), entry 0x07 (Data Out 5))	UINT32	RO	0x3003:07, 8
1602:08	SubIndex 0x008	8. PDO Mapping entry (object 0x3003 (Outputs), entry 0x08 (Data Out 6))	UINT32	RO	0x3003:08, 8
1602:09	SubIndex 0x009	9. PDO Mapping entry (object 0x3003 (Outputs), entry 0x09 (Data Out 7))	UINT32	RO	0x3003:09, 8
1602:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0A (Data Out 8))	UINT32	RO	0x3003:0A, 8
1602:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0B (Data Out 9))	UINT32	RO	0x3003:0B, 8
1602:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0C (Data Out 10))	UINT32	RO	0x3003:0C, 8
1602:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0D (Data Out 11))	UINT32	RO	0x3003:0D, 8
1602:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0E (Data Out 12))	UINT32	RO	0x3003:0E, 8
1602:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0F (Data Out 13))	UINT32	RO	0x3003:0F, 8
1602:10	SubIndex 0x016	16. PDO Mapping entry (object 0x3003 (Outputs), entry 0x10 (Data Out 14))	UINT32	RO	0x3003:10, 8
1602:11	SubIndex 0x017	17. PDO Mapping entry (object 0x3003 (Outputs), entry 0x11 (Data Out 15))	UINT32	RO	0x3003:11, 8
1602:12	SubIndex 0x018	18. PDO Mapping entry (object 0x3003 (Outputs), entry 0x12 (Data Out 16))	UINT32	RO	0x3003:12, 8
1602:13	SubIndex 0x019	19. PDO Mapping entry (object 0x3003 (Outputs), entry 0x13 (Data Out 17))	UINT32	RO	0x3003:13, 8
1602:14	SubIndex 0x020	20. PDO Mapping entry (object 0x3003 (Outputs), entry 0x14 (Data Out 18))	UINT32	RO	0x3003:14, 8
1602:15	SubIndex 0x021	21. PDO Mapping entry (object 0x3003 (Outputs), entry 0x15 (Data Out 19))	UINT32	RO	0x3003:15, 8
1602:16	SubIndex 0x022	22. PDO Mapping entry (object 0x3003 (Outputs), entry 0x16 (Data Out 20))	UINT32	RO	0x3003:16, 8
1602:17	SubIndex 0x023	23. PDO Mapping entry (object 0x3003 (Outputs), entry 0x17 (Data Out 21))	UINT32	RO	0x3003:17, 8

Index 0x1604 COM RxPDO-Map Outputs [ab Hardwarestand 03]

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	COM RxPDO-Map Outputs	PDO Mapping RxPDO 5	UINT8	RO	0x1C (28_{dec})
1604:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x01 (Transmit request))	UINT32	RO	0x7000:01, 1
1604:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x02 (Receive accepted))	UINT32	RO	0x7000:02, 1
1604:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x03 (Init request))	UINT32	RO	0x7000:03, 1
1604:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x04 (Send continues))	UINT32	RO	0x7000:04, 1
1604:05	SubIndex 0x005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1604:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x09 (Output length))	UINT32	RO	0x7000:09, 8
1604:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x11 (Data Out 0))	UINT32	RO	0x7000:11, 8
1604:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x12 (Data Out 1))	UINT32	RO	0x7000:12, 8
1604:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x13 (Data Out 2))	UINT32	RO	0x7000:13, 8
1604:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x14 (Data Out 3))	UINT32	RO	0x7000:14, 8
1604:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x15 (Data Out 4))	UINT32	RO	0x7000:15, 8
1604:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x16 (Data Out 5))	UINT32	RO	0x7000:16, 8
1604:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x17 (Data Out 6))	UINT32	RO	0x7000:17, 8
1604:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x18 (Data Out 7))	UINT32	RO	0x7000:18, 8
1604:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x19 (Data Out 8))	UINT32	RO	0x7000:19, 8
1604:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1A (Data Out 9))	UINT32	RO	0x7000:1A, 8
1604:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1B (Data Out 10))	UINT32	RO	0x7000:1B, 8
1604:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1C (Data Out 11))	UINT32	RO	0x7000:1C, 8
1604:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1D (Data Out 12))	UINT32	RO	0x7000:1D, 8
1604:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1E (Data Out 13))	UINT32	RO	0x7000:1E, 8
1604:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1F (Data Out 14))	UINT32	RO	0x7000:1F, 8
1604:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x20 (Data Out 15))	UINT32	RO	0x7000:20, 8
1604:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x21 (Data Out 16))	UINT32	RO	0x7000:21, 8
1604:18	SubIndex 0x024	24. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x22 (Data Out 17))	UINT32	RO	0x7000:22, 8
1604:19	SubIndex 0x025	25. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x23 (Data Out 18))	UINT32	RO	0x7000:23, 8
1604:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x24 (Data Out 19))	UINT32	RO	0x7000:24, 8
1604:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x25 (Data Out 20))	UINT32	RO	0x7000:25, 8
1604:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x26 (Data Out 21))	UINT32	RO	0x7000:26, 8

Index 0x1605 COM ext. outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	COM ext. outputs	PDO Mapping RxPDO 6	UINT8	RO	0x38 (56_{dec})
1605:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x01 (Transmit request))	UINT32	RO	0x7001:01, 1
1605:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x02 (Receive accepted))	UINT32	RO	0x7001:02, 1
1605:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x03 (Init request))	UINT32	RO	0x7001:03, 1
1605:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x04 (Send continues))	UINT32	RO	0x7001:04, 1
1605:05	SubIndex 0x005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1605:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x09 (Output length))	UINT32	RO	0x7001:09, 8
1605:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x11 (Data Out 0))	UINT32	RO	0x7001:11, 16
1605:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x12 (Data Out 1))	UINT32	RO	0x7001:12, 16
1605:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x13 (Data Out 2))	UINT32	RO	0x7001:13, 16
1605:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x14 (Data Out 3))	UINT32	RO	0x7001:14, 16
1605:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x15 (Data Out 4))	UINT32	RO	0x7001:15, 16
1605:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x16 (Data Out 5))	UINT32	RO	0x7001:16, 16
1605:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x17 (Data Out 6))	UINT32	RO	0x7001:17, 16
1605:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x18 (Data Out 7))	UINT32	RO	0x7001:18, 16
1605:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x19 (Data Out 8))	UINT32	RO	0x7001:19, 16
1605:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1A (Data Out 9))	UINT32	RO	0x7001:1A, 16
1605:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1B (Data Out 10))	UINT32	RO	0x7001:1B, 16
1605:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1C (Data Out 11))	UINT32	RO	0x7001:1C, 16
1605:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1D (Data Out 12))	UINT32	RO	0x7001:1D, 16
1605:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1E (Data Out 13))	UINT32	RO	0x7001:1E, 16
1605:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1F (Data Out 14))	UINT32	RO	0x7001:1F, 16
1605:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x20 (Data Out 15))	UINT32	RO	0x7001:20, 16
1605:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x21 (Data Out 16))	UINT32	RO	0x7001:21, 16
1605:18	SubIndex 0x024	24. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x22 (Data Out 17))	UINT32	RO	0x7001:22, 16
1605:19	SubIndex 0x025	25. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x23 (Data Out 18))	UINT32	RO	0x7001:23, 16
1605:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x24 (Data Out 19))	UINT32	RO	0x7001:24, 16
1605:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x25 (Data Out 20))	UINT32	RO	0x7001:25, 16
1605:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x26 (Data Out 21))	UINT32	RO	0x7001:26, 16
1605:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x27 (Data Out 22))	UINT32	RO	0x7001:27, 16
1605:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x28 (Data Out 23))	UINT32	RO	0x7001:28, 16
1605:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x29 (Data Out 24))	UINT32	RO	0x7001:29, 16

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:20	SubIndex 0x032	32. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2A (Data Out 25))	UINT32	RO	0x7001:2A, 16
1605:21	SubIndex 0x033	33. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2B (Data Out 26))	UINT32	RO	0x7001:2B, 16
1605:22	SubIndex 0x034	34. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2C (Data Out 27))	UINT32	RO	0x7001:2C, 16
1605:23	SubIndex 0x035	35. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2D (Data Out 28))	UINT32	RO	0x7001:2D, 16
1605:24	SubIndex 0x036	36. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2E (Data Out 29))	UINT32	RO	0x7001:2E, 16
1605:25	SubIndex 0x037	37. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2F (Data Out 30))	UINT32	RO	0x7001:2F, 16
1605:26	SubIndex 0x038	38. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x30 (Data Out 31))	UINT32	RO	0x7001:30, 16
1605:27	SubIndex 0x039	39. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x31 (Data Out 32))	UINT32	RO	0x7001:31, 16
1605:28	SubIndex 0x040	40. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x32 (Data Out 33))	UINT32	RO	0x7001:32, 16
1605:29	SubIndex 0x041	41. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x33 (Data Out 34))	UINT32	RO	0x7001:33, 16
1605:2A	SubIndex 0x042	42. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x34 (Data Out 35))	UINT32	RO	0x7001:34, 16
1605:2B	SubIndex 0x043	43. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x35 (Data Out 36))	UINT32	RO	0x7001:35, 16
1605:2C	SubIndex 0x044	44. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x36 (Data Out 37))	UINT32	RO	0x7001:36, 16
1605:2D	SubIndex 0x045	45. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x37 (Data Out 38))	UINT32	RO	0x7001:37, 16
1605:2E	SubIndex 0x046	46. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x38 (Data Out 39))	UINT32	RO	0x7001:38, 16
1605:2F	SubIndex 0x047	47. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x39 (Data Out 40))	UINT32	RO	0x7001:39, 16
1605:30	SubIndex 0x048	48. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3A (Data Out 41))	UINT32	RO	0x7001:3A, 16
1605:31	SubIndex 0x049	49. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3B (Data Out 42))	UINT32	RO	0x7001:3B, 16
1605:32	SubIndex 0x050	50. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3C (Data Out 43))	UINT32	RO	0x7001:3C, 16
1605:33	SubIndex 0x051	51. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3D (Data Out 44))	UINT32	RO	0x7001:3D, 16
1605:34	SubIndex 0x052	52. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3E (Data Out 45))	UINT32	RO	0x7001:3E, 16
1605:35	SubIndex 0x053	53. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3F (Data Out 46))	UINT32	RO	0x7001:3F, 16
1605:36	SubIndex 0x054	54. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x40 (Data Out 47))	UINT32	RO	0x7001:40, 16
1605:37	SubIndex 0x055	55. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x41 (Data Out 48))	UINT32	RO	0x7001:41, 16
1605:38	SubIndex 0x056	56. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x42 (Data Out 49))	UINT32	RO	0x7001:42, 16

Index 0x1800 TxPDO-Par Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1800:0	TxPDO-Par Inputs	PDO Parameter TxPDO 1	UINT8	RO	0x06 (6_{dec})
1800:06	Exclude TxPDOs	Specifies the TxPDOs (index 0x of TxPDO mapping objects) that must not be transferred together with TxPDO 1	OCTET-STRING[6]	RO	01 1A 02 1A 04 1A

Index 0x1801 TxPDO-Par Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1801:0	TxPDO-Par Inputs	PDO Parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1801:06	Exclude TxPDOs	Specifies the TxPDOs (index 0x of TxPDO mapping objects) that must not be transferred together with TxPDO 2	OCTET-STRING[6]	RO	00 1A 02 1A 04 1A

Index 0x1802 TxPDO-Par Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1802:0	TxPDO-Par Inputs	PDO Parameter TxPDO 3	UINT8	RO	0x06 (6_{dec})
1802:06	Exclude TxPDOs	Specifies the TxPDOs (index 0x of TxPDO mapping objects) that must not be transferred together with TxPDO 3	OCTET-STRING[6]	RO	00 1A 01 1A 04 1A

Index 0x1804 COM TxPDO-Par Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1804:0	COM TxPDO-Par Inputs	PDO Parameter TxPDO 5	UINT8	RO	0x06 (6_{dec})
1804:06	Exclude TxPDOs	Specifies the TxPDOs (index 0x of TxPDO mapping objects) that must not be transferred together with TxPDO 5	OCTET-STRING[8]	RO	00 1A 01 1A 02 1A 05 1A

Index 0x1805 COM ext. inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1805:0	COM ext. inputs	PDO Parameter TxPDO 6	UINT8	RO	0x06 (6_{dec})
1805:06	Exclude TxPDOs	Specifies the TxPDOs (index 0x of TxPDO mapping objects) that must not be transferred together with TxPDO 6	OCTET-STRING[8]	RO	00 1A 01 1A 02 1A 04 1A

Index 0x1A00 TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	TxPDO-Map Inputs	PDO Mapping TxPDO 1	UINT8	RO	0x04 (4_{dec})
1A00:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3101 (Inputs), entry 0x01 (Status))	UINT32	RO	0x3101:01, 8
1A00:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3101 (Inputs), entry 0x02 (Data In 0))	UINT32	RO	0x3101:02, 8
1A00:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3101 (Inputs), entry 0x03 (Data In 1))	UINT32	RO	0x3101:03, 8
1A00:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3101 (Inputs), entry 0x04 (Data In 2))	UINT32	RO	0x3101:04, 8

Index 0x1A01 TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	TxPDO-Map Inputs	PDO Mapping TxPDO 2	UINT8	RO	0x06 (6_{dec})
1A01:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3102 (Inputs), entry 0x01 (Status))	UINT32	RO	0x3102:01, 8
1A01:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3102 (Inputs), entry 0x02 (Data In 0))	UINT32	RO	0x3102:02, 8
1A01:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3102 (Inputs), entry 0x03 (Data In 1))	UINT32	RO	0x3102:03, 8
1A01:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3102 (Inputs), entry 0x04 (Data In 2))	UINT32	RO	0x3102:04, 8
1A01:05	SubIndex 0x005	5. PDO Mapping entry (object 0x3102 (Inputs), entry 0x05 (Data In 3))	UINT32	RO	0x3102:05, 8
1A01:06	SubIndex 0x006	6. PDO Mapping entry (object 0x3102 (Inputs), entry 0x06 (Data In 4))	UINT32	RO	0x3102:06, 8

Index 0x1A02 TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	TxPDO-Map Inputs	PDO Mapping TxPDO 3	UINT8	RO	0x17 (23_{dec})
1A02:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3103 (Inputs), entry 0x01 (Status))	UINT32	RO	0x3103:01, 16
1A02:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3103 (Inputs), entry 0x02 (Data In 0))	UINT32	RO	0x3103:02, 8
1A02:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3103 (Inputs), entry 0x03 (Data In 1))	UINT32	RO	0x3103:03, 8
1A02:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3103 (Inputs), entry 0x04 (Data In 2))	UINT32	RO	0x3103:04, 8
1A02:05	SubIndex 0x005	5. PDO Mapping entry (object 0x3103 (Inputs), entry 0x05 (Data In 3))	UINT32	RO	0x3103:05, 8
1A02:06	SubIndex 0x006	6. PDO Mapping entry (object 0x3103 (Inputs), entry 0x06 (Data In 4))	UINT32	RO	0x3103:06, 8
1A02:07	SubIndex 0x007	7. PDO Mapping entry (object 0x3103 (Inputs), entry 0x07 (Data In 5))	UINT32	RO	0x3103:07, 8
1A02:08	SubIndex 0x008	8. PDO Mapping entry (object 0x3103 (Inputs), entry 0x08 (Data In 6))	UINT32	RO	0x3103:08, 8
1A02:09	SubIndex 0x009	9. PDO Mapping entry (object 0x3103 (Inputs), entry 0x09 (Data In 7))	UINT32	RO	0x3103:09, 8
1A02:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0A (Data In 8))	UINT32	RO	0x3103:0A, 8
1A02:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0B (Data In 9))	UINT32	RO	0x3103:0B, 8
1A02:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0C (Data In 10))	UINT32	RO	0x3103:0C, 8
1A02:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0D (Data In 11))	UINT32	RO	0x3103:0D, 8
1A02:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0E (Data In 12))	UINT32	RO	0x3103:0E, 8
1A02:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0F (Data In 13))	UINT32	RO	0x3103:0F, 8
1A02:10	SubIndex 0x016	16. PDO Mapping entry (object 0x3103 (Inputs), entry 0x10 (Data In 14))	UINT32	RO	0x3103:10, 8
1A02:11	SubIndex 0x017	17. PDO Mapping entry (object 0x3103 (Inputs), entry 0x11 (Data In 15))	UINT32	RO	0x3103:11, 8
1A02:12	SubIndex 0x018	18. PDO Mapping entry (object 0x3103 (Inputs), entry 0x12 (Data In 16))	UINT32	RO	0x3103:12, 8
1A02:13	SubIndex 0x019	19. PDO Mapping entry (object 0x3103 (Inputs), entry 0x13 (Data In 17))	UINT32	RO	0x3103:13, 8
1A02:14	SubIndex 0x020	20. PDO Mapping entry (object 0x3103 (Inputs), entry 0x14 (Data In 18))	UINT32	RO	0x3103:14, 8
1A02:15	SubIndex 0x021	21. PDO Mapping entry (object 0x3103 (Inputs), entry 0x15 (Data In 19))	UINT32	RO	0x3103:15, 8
1A02:16	SubIndex 0x022	22. PDO Mapping entry (object 0x3103 (Inputs), entry 0x16 (Data In 20))	UINT32	RO	0x3103:16, 8
1A02:17	SubIndex 0x023	23. PDO Mapping entry (object 0x3103 (Inputs), entry 0x17 (Data In 21))	UINT32	RO	0x3103:17, 8

Index 0x1A04 COM TxPDO-Map Inputs [ab Hardwarestand 03]

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	COM TxPDO-Map Inputs	PDO Mapping TxPDO 5	UINT8	RO	0x1F (31_{dec})
1A04:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x01 (Transmit accepted))	UINT32	RO	0x6000:01, 1
1A04:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x02 (Receive request))	UINT32	RO	0x6000:02, 1
1A04:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x03 (Init accepted))	UINT32	RO	0x6000:03, 1
1A04:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x04 (Buffer full))	UINT32	RO	0x6000:04, 1
1A04:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x05 (Parity error))	UINT32	RO	0x6000:05, 1
1A04:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x06 (Framing error))	UINT32	RO	0x6000:06, 1
1A04:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x07 (Overrun error))	UINT32	RO	0x6000:07, 1
1A04:08	SubIndex 0x008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A04:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x09 (Input length))	UINT32	RO	0x6000:09, 8
1A04:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x11 (Data In 0))	UINT32	RO	0x6000:11, 8
1A04:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x12 (Data In 1))	UINT32	RO	0x6000:12, 8
1A04:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x13 (Data In 2))	UINT32	RO	0x6000:13, 8
1A04:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x14 (Data In 3))	UINT32	RO	0x6000:14, 8
1A04:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x15 (Data In 4))	UINT32	RO	0x6000:15, 8
1A04:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x16 (Data In 5))	UINT32	RO	0x6000:16, 8
1A04:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x17 (Data In 6))	UINT32	RO	0x6000:17, 8
1A04:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x18 (Data In 7))	UINT32	RO	0x6000:18, 8
1A04:12	SubIndex 0x018	18. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x19 (Data In 8))	UINT32	RO	0x6000:19, 8
1A04:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1A (Data In 9))	UINT32	RO	0x6000:1A, 8
1A04:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1B (Data In 10))	UINT32	RO	0x6000:1B, 8
1A04:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1C (Data In 11))	UINT32	RO	0x6000:1C, 8
1A04:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1D (Data In 12))	UINT32	RO	0x6000:1D, 8
1A04:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1E (Data In 13))	UINT32	RO	0x6000:1E, 8
1A04:18	SubIndex 0x024	24. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1F (Data In 14))	UINT32	RO	0x6000:1F, 8
1A04:19	SubIndex 0x025	25. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x20 (Data In 15))	UINT32	RO	0x6000:20, 8
1A04:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x21 (Data In 16))	UINT32	RO	0x6000:21, 8
1A04:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x22 (Data In 17))	UINT32	RO	0x6000:22, 8
1A04:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x23 (Data In 18))	UINT32	RO	0x6000:23, 8
1A04:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x24 (Data In 19))	UINT32	RO	0x6000:24, 8
1A04:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x25 (Data In 20))	UINT32	RO	0x6000:25, 8
1A04:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x26 (Data In 21))	UINT32	RO	0x6000:26, 8

Index 0x1A05 COM ext. inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	COM ext. inputs	PDO Mapping TxPDO 6	UINT8	RO	0x3B (59_{dec})
1A05:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x01 (Transmit accepted))	UINT32	RO	0x6001:01, 1
1A05:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x02 (Receive request))	UINT32	RO	0x6001:02, 1
1A05:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x03 (Init accepted))	UINT32	RO	0x6001:03, 1
1A05:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x04 (Buffer full))	UINT32	RO	0x6001:04, 1
1A05:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x05 (Parity error))	UINT32	RO	0x6001:05, 1
1A05:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x06 (Framing error))	UINT32	RO	0x6001:06, 1
1A05:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x07 (Overrun error))	UINT32	RO	0x6001:07, 1
1A05:08	SubIndex 0x008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A05:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x09 (Input length))	UINT32	RO	0x6001:09, 8
1A05:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x11 (Data In 0))	UINT32	RO	0x6001:11, 16
1A05:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x12 (Data In 1))	UINT32	RO	0x6001:12, 16
1A05:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x13 (Data In 2))	UINT32	RO	0x6001:13, 16
1A05:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x14 (Data In 3))	UINT32	RO	0x6001:14, 16
1A05:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x15 (Data In 4))	UINT32	RO	0x6001:15, 16
1A05:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x16 (Data In 5))	UINT32	RO	0x6001:16, 16
1A05:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x17 (Data In 6))	UINT32	RO	0x6001:17, 16
1A05:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x18 (Data In 7))	UINT32	RO	0x6001:18, 16
1A05:12	SubIndex 0x018	18. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x19 (Data In 8))	UINT32	RO	0x6001:19, 16
1A05:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1A (Data In 9))	UINT32	RO	0x6001:1A, 16
1A05:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1B (Data In 10))	UINT32	RO	0x6001:1B, 16
1A05:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1C (Data In 11))	UINT32	RO	0x6001:1C, 16
1A05:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1D (Data In 12))	UINT32	RO	0x6001:1D, 16
1A05:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1E (Data In 13))	UINT32	RO	0x6001:1E, 16
1A05:18	SubIndex 0x024	24. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1F (Data In 14))	UINT32	RO	0x6001:1F, 16
1A05:19	SubIndex 0x025	25. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x20 (Data In 15))	UINT32	RO	0x6001:20, 16
1A05:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x21 (Data In 16))	UINT32	RO	0x6001:21, 16
1A05:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x22 (Data In 17))	UINT32	RO	0x6001:22, 16
1A05:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x23 (Data In 18))	UINT32	RO	0x6001:23, 16
1A05:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x24 (Data In 19))	UINT32	RO	0x6001:24, 16
1A05:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x25 (Data In 20))	UINT32	RO	0x6001:25, 16
1A05:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x26 (Data In 21))	UINT32	RO	0x6001:26, 16

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:20	SubIndex 0x032	32. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x27 (Data In 22))	UINT32	RO	0x6001:27, 16
1A05:21	SubIndex 0x033	33. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x28 (Data In 23))	UINT32	RO	0x6001:28, 16
1A05:22	SubIndex 0x034	34. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x29 (Data In 24))	UINT32	RO	0x6001:29, 16
1A05:23	SubIndex 0x035	35. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2A (Data In 25))	UINT32	RO	0x6001:2A, 16
1A05:24	SubIndex 0x036	36. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2B (Data In 26))	UINT32	RO	0x6001:2B, 16
1A05:25	SubIndex 0x037	37. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2C (Data In 27))	UINT32	RO	0x6001:2C, 16
1A05:26	SubIndex 0x038	38. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2D (Data In 28))	UINT32	RO	0x6001:2D, 16
1A05:27	SubIndex 0x039	39. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2E (Data In 29))	UINT32	RO	0x6001:2E, 16
1A05:28	SubIndex 0x040	40. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2F (Data In 30))	UINT32	RO	0x6001:2F, 16
1A05:29	SubIndex 0x041	41. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x30 (Data In 31))	UINT32	RO	0x6001:30, 16
1A05:2A	SubIndex 0x042	42. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x31 (Data In 32))	UINT32	RO	0x6001:31, 16
1A05:2B	SubIndex 0x043	43. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x32 (Data In 33))	UINT32	RO	0x6001:32, 16
1A05:2C	SubIndex 0x044	44. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x33 (Data In 34))	UINT32	RO	0x6001:33, 16
1A05:2D	SubIndex 0x045	45. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x34 (Data In 35))	UINT32	RO	0x6001:34, 16
1A05:2E	SubIndex 0x046	46. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x35 (Data In 36))	UINT32	RO	0x6001:35, 16
1A05:2F	SubIndex 0x047	47. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x36 (Data In 37))	UINT32	RO	0x6001:36, 16
1A05:30	SubIndex 0x048	48. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x37 (Data In 38))	UINT32	RO	0x6001:37, 16
1A05:31	SubIndex 0x049	49. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x38 (Data In 39))	UINT32	RO	0x6001:38, 16
1A05:32	SubIndex 0x050	50. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x39 (Data In 40))	UINT32	RO	0x6001:39, 16
1A05:33	SubIndex 0x051	51. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3A (Data In 41))	UINT32	RO	0x6001:3A, 16
1A05:34	SubIndex 0x052	52. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3B (Data In 42))	UINT32	RO	0x6001:3B, 16
1A05:35	SubIndex 0x053	53. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3C (Data In 43))	UINT32	RO	0x6001:3C, 16
1A05:36	SubIndex 0x054	54. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3D (Data In 44))	UINT32	RO	0x6001:3D, 16
1A05:37	SubIndex 0x055	55. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3E (Data In 45))	UINT32	RO	0x6001:3E, 16
1A05:38	SubIndex 0x056	56. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3F (Data In 46))	UINT32	RO	0x6001:3F, 16
1A05:39	SubIndex 0x057	57. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x40 (Data In 47))	UINT32	RO	0x6001:40, 16
1A05:3A	SubIndex 0x058	58. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x41 (Data In 48))	UINT32	RO	0x6001:41, 16
1A05:3B	SubIndex 0x059	59. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x42 (Data In 49))	UINT32	RO	0x6001:42, 16

Index 0x1C00 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 0x001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1_{dec})
1C00:02	SubIndex 0x002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2_{dec})
1C00:03	SubIndex 0x003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3_{dec})
1C00:04	SubIndex 0x004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4_{dec})

Index 0x1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x01 (1_{dec})
1C12:01	SubIndex 0x001	1st allocated RxPDO (contains the index 0x of the associated RxPDO mapping object)	UINT16	RW	0x1602 (5634_{dec})
1C12:02	SubIndex 0x002	2nd allocated RxPDO (contains the index 0x of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 0x1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x01 (1_{dec})
1C13:01	SubIndex 0x001	1st allocated TxPDO (contains the index 0x of the associated TxPDO mapping object)	UINT16	RW	0x1A02 (6658_{dec})
1C13:02	SubIndex 0x002	2nd allocated TxPDO (contains the index 0x of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 0x1C32 SM output parameter [from hardware version 03]

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	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0007A120 (500000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchronous 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 [▶ 107])	UINT16	RO	0xC007 (49159_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00004E20 (20000_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{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, 1C32:05, 1C32:06, 1C32:09 [▶ 107], 1C33:03, 1C33:06, 1C33:09 [▶ 108] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{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 0x1C33 SM input parameter [from hardware version 03]

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: Synchronous with SM 3 event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchronous with SM 2 event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	as 1C32:02 [▶ 107]	UINT32	RW	0x0007A120 (500000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: synchronous with SM 2 event is supported (outputs available) • Bit 1: synchronous 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 [▶ 107] or 1C33:08 [▶ 108])	UINT16	RO	0xC007 (49159_{dec})
1C33:05	Minimum cycle time	as 1C32:05 [▶ 107]	UINT32	RO	0x00004E20 (20000_{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})
1C33:08	Command	as 1C32:08 [▶ 107]	UINT16	RW	0x0000 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:09	Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:0B	SM event missed counter	as 1C32:11 [▶ 107]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 1C32:12 [▶ 107]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 1C32:13 [▶ 107]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 1C32:32 [▶ 107]	BOOLEAN	RO	0x00 (0_{dec})

Index 0x3001 Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
3001:0	Outputs	Length of this object	UINT8	RO	0x04 (4_{dec})
3001:01	Ctrl	Control-Byte [▶ 90]	UINT8	RO	0x00 (0_{dec})
3001:02	Data Out 0	Output byte 0	UINT8	RO	0x00 (0_{dec})
3001:03	Data Out 1	Output byte 1	UINT8	RO	0x00 (0_{dec})
3001:04	Data Out 2	Output byte 2	UINT8	RO	0x00 (0_{dec})

Index 0x3002 Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
3002:0	Outputs	Length of this object	UINT8	RO	0x06 (6_{dec})
3002:01	Ctrl	Control-Byte [▶ 90]	UINT8	RO	0x00 (0_{dec})
3002:02	Data Out 0	Output byte 0	UINT8	RO	0x00 (0_{dec})
3002:03	Data Out 1	Output byte 1	UINT8	RO	0x00 (0_{dec})
3002:04	Data Out 2	Output byte 2	UINT8	RO	0x00 (0_{dec})
3002:05	Data Out 3	Output byte 3	UINT8	RO	0x00 (0_{dec})
3002:06	Data Out 4	Output byte 4	UINT8	RO	0x00 (0_{dec})

Index 0x3003 Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
3003:0	Outputs	Length of this object	UINT8	RO	0x17 (23_{dec})
3003:01	Ctrl	Control word [▶ 90]	UINT16	RO	0x0000 (0_{dec})
3003:02	Data Out 0	Output byte 0	UINT8	RO	0x00 (0_{dec})
3003:03	Data Out 1	Output byte 1	UINT8	RO	0x00 (0_{dec})
3003:04	Data Out 2	Output byte 2	UINT8	RO	0x00 (0_{dec})
3003:05	Data Out 3	Output byte 3	UINT8	RO	0x00 (0_{dec})
3003:06	Data Out 4	Output byte 4	UINT8	RO	0x00 (0_{dec})
3003:07	Data Out 5	Output byte 5	UINT8	RO	0x00 (0_{dec})
3003:08	Data Out 6	Output byte 6	UINT8	RO	0x00 (0_{dec})
3003:09	Data Out 7	Output byte 7	UINT8	RO	0x00 (0_{dec})
3003:0A	Data Out 8	Output byte 8	UINT8	RO	0x00 (0_{dec})
3003:0B	Data Out 9	Output byte 9	UINT8	RO	0x00 (0_{dec})
3003:0C	Data Out 10	Output byte 10	UINT8	RO	0x00 (0_{dec})
3003:0D	Data Out 11	Output byte 11	UINT8	RO	0x00 (0_{dec})
3003:0E	Data Out 12	Output byte 12	UINT8	RO	0x00 (0_{dec})
3003:0F	Data Out 13	Output byte 13	UINT8	RO	0x00 (0_{dec})
3003:10	Data Out 14	Output byte 14	UINT8	RO	0x00 (0_{dec})
3003:11	Data Out 15	Output byte 15	UINT8	RO	0x00 (0_{dec})
3003:12	Data Out 16	Output byte 16	UINT8	RO	0x00 (0_{dec})
3003:13	Data Out 17	Output byte 17	UINT8	RO	0x00 (0_{dec})
3003:14	Data Out 18	Output byte 18	UINT8	RO	0x00 (0_{dec})
3003:15	Data Out 19	Output byte 19	UINT8	RO	0x00 (0_{dec})
3003:16	Data Out 20	Output byte 20	UINT8	RO	0x00 (0_{dec})
3003:17	Data Out 21	Output byte 21	UINT8	RO	0x00 (0_{dec})

Index 0x3101 Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
3101:0	Inputs	Length of this object	UINT8	RO	0x04 (4_{dec})
3101:01	Status	Status byte [▶ 90]	UINT8	RO	0x00 (0_{dec})
3101:02	Data In 0	Input byte 0	UINT8	RO	0x00 (0_{dec})
3101:03	Data In 1	Input byte 1	UINT8	RO	0x00 (0_{dec})
3101:04	Data In 2	Input byte 2	UINT8	RO	0x00 (0_{dec})

Index 0x3102 Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
3102:0	Inputs	Length of this object	UINT8	RO	0x06 (6_{dec})
3102:01	Status	Status byte [▶ 90]	UINT8	RO	0x00 (0_{dec})
3102:02	Data In 0	Input byte 0	UINT8	RO	0x00 (0_{dec})
3102:03	Data In 1	Input byte 1	UINT8	RO	0x00 (0_{dec})
3102:04	Data In 2	Input byte 2	UINT8	RO	0x00 (0_{dec})
3102:05	Data In 3	Input byte 3	UINT8	RO	0x00 (0_{dec})
3102:06	Data In 4	Input byte 4	UINT8	RO	0x00 (0_{dec})

Index 0x3103 Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
3103:0	Inputs	Length of this object	UINT8	RO	0x17 (23_{dec})
3103:01	Status	Status word [▶ 90]	UINT16	RO	0x0000 (0_{dec})
3103:02	Data In 0	Input byte 0	UINT8	RO	0x00 (0_{dec})
3103:03	Data In 1	Input byte 1	UINT8	RO	0x00 (0_{dec})
3103:04	Data In 2	Input byte 2	UINT8	RO	0x00 (0_{dec})
3103:05	Data In 3	Input byte 3	UINT8	RO	0x00 (0_{dec})
3103:06	Data In 4	Input byte 4	UINT8	RO	0x00 (0_{dec})
3103:07	Data In 5	Input byte 5	UINT8	RO	0x00 (0_{dec})
3103:08	Data In 6	Input byte 6	UINT8	RO	0x00 (0_{dec})
3103:09	Data In 7	Input byte 7	UINT8	RO	0x00 (0_{dec})
3103:0A	Data In 8	Input byte 8	UINT8	RO	0x00 (0_{dec})
3103:0B	Data In 9	Input byte 9	UINT8	RO	0x00 (0_{dec})
3103:0C	Data In 10	Input byte 10	UINT8	RO	0x00 (0_{dec})
3103:0D	Data In 11	Input byte 11	UINT8	RO	0x00 (0_{dec})
3103:0E	Data In 12	Input byte 12	UINT8	RO	0x00 (0_{dec})
3103:0F	Data In 13	Input byte 13	UINT8	RO	0x00 (0_{dec})
3103:10	Data In 14	Input byte 14	UINT8	RO	0x00 (0_{dec})
3103:11	Data In 15	Input byte 15	UINT8	RO	0x00 (0_{dec})
3103:12	Data In 16	Input byte 16	UINT8	RO	0x00 (0_{dec})
3103:13	Data In 17	Input byte 17	UINT8	RO	0x00 (0_{dec})
3103:14	Data In 18	Input byte 18	UINT8	RO	0x00 (0_{dec})
3103:15	Data In 19	Input byte 19	UINT8	RO	0x00 (0_{dec})
3103:16	Data In 20	Input byte 20	UINT8	RO	0x00 (0_{dec})
3103:17	Data In 21	Input byte 21	UINT8	RO	0x00 (0_{dec})

Index 0x4070 Data bytes in send buffer

Index (hex)	Name	Meaning	Data type	Flags	Default
4070:0	Data bytes in send buffer	Number of data bytes in the send FIFO	UINT16	RO	0x0000 (0_{dec})

Index 0x4071 Data bytes in receive buffer

Index (hex)	Name	Meaning	Data type	Flags	Default
4071:0	Data bytes in receive buffer	Number of data bytes in the receive FIFO	UINT16	RO	0x0000 (0_{dec})

Index 0x4072 Diagnostic

Index (hex)	Name	Meaning	Data type	Flags	Default
4072:0	Diagnostic	Length of this object	UINT8	RO	0x05 (5_{dec})
4072:01	Buffer overflow	A buffer overflow has occurred.	BOOLEAN	RO	0x00 (0_{dec})
4072:02	Parity error	A parity error has occurred. The affected data item is not loaded into the receive FIFO of the terminal and is lost.	BOOLEAN	RO	0x00 (0_{dec})
4072:03	Framing error	A framing error has occurred. The affected data item is not loaded into the receive FIFO of the terminal and is lost.	BOOLEAN	RO	0x00 (0_{dec})
4072:04	Overrun error	An overrun error has occurred. The affected data item is not loaded into the receive FIFO of the terminal and is lost.	BOOLEAN	RO	0x00 (0_{dec})
4072:05	Buffer full	The reception FIFO is full. All further incoming data will be lost!	BOOLEAN	RO	0x00 (0_{dec})

7.2.3 Profile-specific objects (0x6000-0xFFFF) [from hardware version 03]

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

Index 0x6000 COM Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	COM Inputs	Max. SubIndex (hex)	UINT8	RO	0x26 (38_{dec})
6000:01	Transmit accepted	The terminal acknowledges receipt of data by changing the state of this bit. Only now new data can be transferred from the controller to the terminal.	BOOLEAN	RO	0x00 (0_{dec})
6000:02	Receive request	By changing the state of this bit, the terminal informs the controller that the DataIn bytes contain the number of bytes displayed in "Input length [▶ 111]". The controller must acknowledge receipt of the data by changing the state of the ReceiveAccepted [▶ 112] bit. Only then new data can be transferred from the terminal to the controller.	BOOLEAN	RO	0x00 (0_{dec})
6000:03	Init accepted	0 The terminal is ready again for serial data exchange. 1 Initialization was completed by the terminal.	BOOLEAN	RO	0x00 (0_{dec})
6000:04	Buffer full	The reception FIFO is full. All incoming data will be lost from this point on!	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 byte 0	UINT8	RO	0x00 (0_{dec})
...	...	....	...	...	...
6000:26	Data In 21	Input byte 21	UINT8	RO	0x00 (0_{dec})

Index 0x6001 COM ext. inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6001:0	COM ext. inputs		UINT8	RO	0x00 (0_{dec})
6001:01	Transmit accepted	identical to index 0x6000	BOOLEAN	RO	0x00 (0_{dec})
6001:02	Receive request		BOOLEAN	RO	0x00 (0_{dec})
6001:03	Init accepted		BOOLEAN	RO	0x00 (0_{dec})
6001:04	Buffer full		BOOLEAN	RO	0x00 (0_{dec})
6001:05	Parity error		BOOLEAN	RO	0x00 (0_{dec})
6001:06	Framing error		BOOLEAN	RO	0x00 (0_{dec})
6001:07	Overrun error		BOOLEAN	RO	0x00 (0_{dec})
6001:09	Input length		UINT8	RO	0x00 (0_{dec})
6001:11	Data In 0		UINT16	RO	0x0000 (0_{dec})
	....				
6001:42	Data In 49		UINT16	RO	0x0000 (0_{dec})

Index 0x7000 COM Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	COM Outputs	Max. SubIndex (hex)	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 [▶ 112]”. The terminal acknowledges receipt of the data by changing the state of the “TransmitAccepted [▶ 111]” bit. Only now new data can be transferred from the controller to the terminal.	BOOLEAN	RO	0x00 (0_{dec})
7000:02	Receive accepted	The controller acknowledges receipt of data by changing the state of this bit. Only then new data can be transferred from the terminal to the controller.	BOOLEAN	RO	0x00 (0_{dec})
7000:03	Init request	0: The controller once again requests the terminal to prepare for serial data exchange. 1: The controller requests terminal for initialization. The transmit and receive functions will be blocked, the FIFO pointer will be reset and the interface will be initialized with the values of the responsible Settings object. The execution of the initialization will be acknowledged by the terminal with the ‘Init accepted [▶ 111]’ bit.	BOOLEAN	RO	0x00 (0_{dec})
7000:04	Send continuous	Continuous sending of data from the FIFO. The send buffer is filled (up to 128 bytes) by the controller. The filled buffer contents will be sent on the rising edge of the bit. If the data has been transmitted, the terminal informs the controller by setting the “Init accepted [▶ 111]” bit. “Init accepted [▶ 111]” is cleared with “SendContinuous [▶ 112]”.	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 byte 0	UINT8	RO	0x00 (0_{dec})
...	...	....	...	...	...
7000:26	Data Out 21	Output byte 21	UINT8	RO	0x00 (0_{dec})

Index 0x7001 COM ext. outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7001:0	COM ext. outputs		UINT8	RO	0x00 (0_{dec})
7001:01	Transmit request	identical to index 0x7000	BOOLEAN	RO	0x00 (0_{dec})
7001:02	Receive accepted		BOOLEAN	RO	0x00 (0_{dec})
7001:03	Init request		BOOLEAN	RO	0x00 (0_{dec})
7001:04	Send continuous		BOOLEAN	RO	0x00 (0_{dec})
7001:09	Output length		UINT8	RO	0x00 (0_{dec})
7001:11	Data Out 0		UINT16	RO	0x0000 (0_{dec})
	…		<	<	<
7001:42	Data Out 49		UINT16	RO	0x0000 (0_{dec})

Index 0xA000 COM Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
A000:0	COM Diag data	Max. SubIndex (hex)	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 reception FIFO is full. All incoming data will be lost from this point on!	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 0xB000 Command

Only EL6021 since FW06 and EL6001 since FW08.

Index (hex)	Name	Meaning	Data type	Flags	Default
B000:0	Command	Max. SubIndex (hex)	UINT8	RO	0x03 (3_{dec})
B000:01	Request	Commands can be sent to the terminal via the request object see command mode [▶ 69]	OCTET-STRING[2]	RW	{0}
B000:02	Status	Status of the command currently being executed 0: Command executed without error. 255: Command is being executed	UINT8	RO	0x00 (0_{dec})
B000:03	Response	Optional response value of the command Byte 0: see B000:02 Byte 1: not used 2-n: Service response Data	OCTET-STRING[6]	RO	{0}

Index 0xF000 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 0xdistance	Index (hex) interval 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 0xF008 Code word

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

Index 0xF010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Max. SubIndex (hex)	UINT8	RW	0x02 (2_{dec})
F010:01	SubIndex 0x001	-	UINT32	RW	0x00000000 (0_{dec})
F010:02	SubIndex 0x002	-	UINT32	RW	0x00000258 (600_{dec})

8 Overview CoE objects EL6002, EL6022

8.1 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

The terminal is parameterized via the CoE - Online tab (with a double-click on the respective object, see below).

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization [▶ 115] during commissioning
- Objects for indicating internal settings [▶ 116] (may be fixed)
- Profile specific objects [▶ 128], which represent the status displays of the inputs and outputs.

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

8.1.1 Objects for commissioning

Index 0x1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default parameters	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 0x001	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 0x80n0 COM Settings Ch. 1 (n = 0), Ch. 2 (n = 1)

Index (hex)	Name	Meaning		Data type	Flags	Default
80n0:0	COM Settings Ch. 1 + Ch. 2	Max. SubIndex (hex)		UINT8	RO	0x1A (26$_{dec}$)
80n0:01**	Enable RTS/CTS	FALSE	RTS/CTS not enabled	BOOLEAN	RW	0x01 (1$_{dec}$)
		TRUE	RTS/CTS enabled			
80n0:02	Enable XON/XOFF supported tx data	FALSE	XON/XOFF is not supported for send data	BOOLEAN	RW	0x00 (0$_{dec}$)
		TRUE	XON/XOFF is supported for send data			
80n0:03	Enable XON/XOFF supported rx data	FALSE	XON/XOFF is not supported for receive data	BOOLEAN	RW	0x00 (0$_{dec}$)
		TRUE	XON/XOFF is supported for receive data			
80n0:04	Enable send FIFO data continuous	FALSE	No continuous sending of data from the FIFO	BOOLEAN	RW	0x00 (0$_{dec}$)
		TRUE	Continuous sending of data from the FIFO enabled: The send buffer is filled (up to 128 bytes) by the controller. The filled buffer content is sent on rising edge of bit “SendContinious”. The terminal acknowledges the data transfer to the controller through setting of bit “InitAccepted”. “InitAccepted” is reset with “SendContinuous”.			
80n0:05	Enable transfer rate optimization	FALSE	Transfer rate optimization switched off	BOOLEAN	RW	0x01 (1$_{dec}$)
		TRUE	Transfer rate optimization switched on: The content of the input buffer is automatically transferred into the process image if • no further byte was received for approx. 16 bit times (i.e. the time it would have taken to receive 2 bytes) after data were received; • the process image is filled			
80n0:06***	Enable half duplex	FALSE	Full-duplex mode	BOOLEAN	RW	0x00 (0$_{dec}$)
		TRUE	Half-duplex mode			
80n0:07***	Enable point to point connection (RS422)	FALSE	Point-to-point connection disabled	BOOLEAN	RW	0x00 (0$_{dec}$)
		TRUE	Point-to-point connection enabled			
80n0:11	Baud rate	Detailed information can be found in chapter Communication features [▶ 76]		BIT4	RW	0x06 (6$_{dec}$)
80n0:15	Data frame	Detailed information can be found in chapter Communication features [▶ 76]		BIT4	RW	0x03 (3$_{dec}$)
80n0:1A	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$_{dec}$)
80n0:1B****	Explicit baudrate	More detailed information can be found in chapters TcVirtualComDriver [▶ 74] and Communication features [▶ 76]		UINT32	RW	0x00002580 (9600$_{dec}$)

**) only EL6002
***) only EL6022
****) only EL6002 from firmware 03 [▶ 135]

8.1.2 Standard objects (0x1000-0x1FFF)

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

Index 0x1000 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 0x1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL60xx

Index 0x1009 Hardware version

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

Index 0x100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	01

Index 0x1018 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	EL6002: 0x17723052 (393359442_{dec}) EL6022: 0x17863052 (394670162_{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	0x00100000 (1048576_{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 0x10F0 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 0x1600 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 0x001	1. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x01 (Transmit request))	BOOLEAN	RO	0x7000:01, 1
1600:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x02 (Receive accepted))	BOOLEAN	RO	0x7000:02, 1
1600:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x03 (Init request))	BOOLEAN	RO	0x7000:03, 1
1600:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x04 (Send continuous))	BOOLEAN	RO	0x7000:04, 1
1600:05	SubIndex 0x005	5. PDO Mapping entry (4 bits align)	Align4	RO	0x0000:00, 4
1600:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x09 (Output length))	UINT8	RO	0x7000:09, 8
1600:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT8	RO	0x7000:11, 8
1600:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT8	RO	0x7000:12, 8
1600:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT8	RO	0x7000:13, 8
1600:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT8	RO	0x7000:14, 8
1600:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT8	RO	0x7000:15, 8
1600:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT8	RO	0x7000:16, 8
1600:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT8	RO	0x7000:17, 8
1600:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT8	RO	0x7000:18, 8
1600:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT8	RO	0x7000:19, 8
1600:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT8	RO	0x7000:1A, 8
1600:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT8	RO	0x7000:1B, 8
1600:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT8	RO	0x7000:1C, 8
1600:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT8	RO	0x7000:1D, 8
1600:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT8	RO	0x7000:1E, 8
1600:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT8	RO	0x7000:1F, 8
1600:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT8	RO	0x7000:20, 8
1600:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT8	RO	0x7000:21, 8
1600:18	SubIndex 0x024	24. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT8	RO	0x7000:22, 8
1600:19	SubIndex 0x025	25. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT8	RO	0x7000:23, 8
1600:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT8	RO	0x7000:24, 8
1600:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT8	RO	0x7000:25, 8
1600:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT8	RO	0x7000:26, 8

Index 0x1601 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 0x001	1. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x01 (Transmit request))	BOOLEAN	RO	0x7010:01, 1
1601:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x02 (Receive accepted))	BOOLEAN	RO	0x7010:02, 1
1601:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x03 (Init request))	BOOLEAN	RO	0x7010:03, 1
1601:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x04 (Send continuous))	BOOLEAN	RO	0x7010:04, 1
1601:05	SubIndex 0x005	5. PDO Mapping entry (4 bits align)	Align4	RO	0x0000:00, 4
1601:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x09 (Output length))	UINT8	RO	0x7010:09, 8
1601:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x11 (Data Out 0))	UINT8	RO	0x7010:11, 8
1601:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x12 (Data Out 1))	UINT8	RO	0x7010:12, 8
1601:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x13 (Data Out 2))	UINT8	RO	0x7010:13, 8
1601:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x14 (Data Out 3))	UINT8	RO	0x7010:14, 8
1601:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x15 (Data Out 4))	UINT8	RO	0x7010:15, 8
1601:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x16 (Data Out 5))	UINT8	RO	0x7010:16, 8
1601:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x17 (Data Out 6))	UINT8	RO	0x7010:17, 8
1601:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x18 (Data Out 7))	UINT8	RO	0x7010:18, 8
1601:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x19 (Data Out 8))	UINT8	RO	0x7010:19, 8
1601:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1A (Data Out 9))	UINT8	RO	0x7010:1A, 8
1601:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1B (Data Out 10))	UINT8	RO	0x7010:1B, 8
1601:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1C (Data Out 11))	UINT8	RO	0x7010:1C, 8
1601:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1D (Data Out 12))	UINT8	RO	0x7010:1D, 8
1601:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1E (Data Out 13))	UINT8	RO	0x7010:1E, 8
1601:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1F (Data Out 14))	UINT8	RO	0x7010:1F, 8
1601:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x20 (Data Out 15))	UINT8	RO	0x7010:20, 8
1601:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x21 (Data Out 16))	UINT8	RO	0x7010:21, 8
1601:18	SubIndex 0x024	24. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x22 (Data Out 17))	UINT8	RO	0x7010:22, 8
1601:19	SubIndex 0x025	25. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x23 (Data Out 18))	UINT8	RO	0x7010:23, 8
1601:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x24 (Data Out 19))	UINT8	RO	0x7010:24, 8
1601:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x25 (Data Out 20))	UINT8	RO	0x7010:25, 8
1601:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x26 (Data Out 21))	UINT8	RO	0x7010:26, 8

Index 0x1604 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 1	UINT8	RO	0x17 (23_{dec})
1604:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7001 (Ctrl Ch.1), entry 0x01 (Ctrl))	UINT16	RO	0x7001:01, 16
1604:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT8	RO	0x7000:11, 8
1604:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT8	RO	0x7000:12, 8
1604:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT8	RO	0x7000:13, 8
1604:05	SubIndex 0x005	5. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT8	RO	0x7000:14, 8
1604:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT8	RO	0x7000:15, 8
1604:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT8	RO	0x7000:16, 8
1604:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT8	RO	0x7000:17, 8
1604:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT8	RO	0x7000:18, 8
1604:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT8	RO	0x7000:19, 8
1604:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT8	RO	0x7000:1A, 8
1604:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT8	RO	0x7000:1B, 8
1604:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT8	RO	0x7000:1C, 8
1604:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT8	RO	0x7000:1D, 8
1604:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT8	RO	0x7000:1E, 8
1604:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT8	RO	0x7000:1F, 8
1604:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT8	RO	0x7000:20, 8
1604:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT8	RO	0x7000:21, 8
1604:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT8	RO	0x7000:22, 8
1604:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT8	RO	0x7000:23, 8
1604:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT8	RO	0x7000:24, 8
1604:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT8	RO	0x7000:25, 8
1604:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT8	RO	0x7000:26, 8

Index 0x1605 COM RxPDO-Map Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	COM RxPDO-Map Outputs Ch.1	PDO Mapping RxPDO 1	UINT8	RO	0x17 (23_{dec})
1605:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7011 (Ctrl Ch.2), entry 0x01 (Ctrl))	UINT16	RO	0x7011:01, 16
1605:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x11 (Data Out 0))	UINT8	RO	0x7010:11, 8
1605:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x12 (Data Out 1))	UINT8	RO	0x7010:12, 8
1605:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x13 (Data Out 2))	UINT8	RO	0x7010:13, 8
1605:05	SubIndex 0x005	5. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x14 (Data Out 3))	UINT8	RO	0x7010:14, 8
1605:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x15 (Data Out 4))	UINT8	RO	0x7010:15, 8
1605:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x16 (Data Out 5))	UINT8	RO	0x7010:16, 8
1605:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x17 (Data Out 6))	UINT8	RO	0x7010:17, 8
1605:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x18 (Data Out 7))	UINT8	RO	0x7010:18, 8
1605:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x19 (Data Out 8))	UINT8	RO	0x7010:19, 8
1605:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1A (Data Out 9))	UINT8	RO	0x7010:1A, 8
1605:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1B (Data Out 10))	UINT8	RO	0x7010:1B, 8
1605:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1C (Data Out 11))	UINT8	RO	0x7010:1C, 8
1605:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1D (Data Out 12))	UINT8	RO	0x7010:1D, 8
1605:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1E (Data Out 13))	UINT8	RO	0x7010:1E, 8
1605:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1F (Data Out 14))	UINT8	RO	0x7010:1F, 8
1605:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x20 (Data Out 15))	UINT8	RO	0x7010:20, 8
1605:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x21 (Data Out 16))	UINT8	RO	0x7010:21, 8
1605:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x22 (Data Out 17))	UINT8	RO	0x7010:22, 8
1605:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x23 (Data Out 18))	UINT8	RO	0x7010:23, 8
1605:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x24 (Data Out 19))	UINT8	RO	0x7010:24, 8
1605:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x25 (Data Out 20))	UINT8	RO	0x7010:25, 8
1605:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x26 (Data Out 21))	UINT8	RO	0x7010:26, 8

Index 0x1A00 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 0x001	1. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x01 (Transmit accepted))	BOOLEAN	RO	0x6000:01, 1
1A00:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x02 (Receive request))	BOOLEAN	RO	0x6000:02, 1
1A00:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x03 (Init accepted))	BOOLEAN	RO	0x6000:03, 1
1A00:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x04 (Buffer full))	BOOLEAN	RO	0x6000:04, 1
1A00:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x05 (Parity error))	BOOLEAN	RO	0x6000:05, 1
1A00:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x06 (Framing error))	BOOLEAN	RO	0x6000:06, 1
1A00:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x07 (Overrun error))	BOOLEAN	RO	0x6000:07, 1
1A00:08	SubIndex 0x008	8. PDO Mapping entry (1 bits align)	Align1	RO	0x0000:00, 1
1A00:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x09 (Input length))	UINT8	RO	0x6000:09, 8
1A00:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT8	RO	0x6000:11, 8
1A00:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT8	RO	0x6000:12, 8
1A00:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT8	RO	0x6000:13, 8
1A00:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT8	RO	0x6000:14, 8
1A00:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT8	RO	0x6000:15, 8
1A00:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT8	RO	0x6000:16, 8
1A00:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT8	RO	0x6000:17, 8
1A00:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT8	RO	0x6000:18, 8
1A00:12	SubIndex 0x018	18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT8	RO	0x6000:19, 8
1A00:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT8	RO	0x6000:1A, 8
1A00:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT8	RO	0x6000:1B, 8
1A00:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT8	RO	0x6000:1C, 8
1A00:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT8	RO	0x6000:1D, 8
1A00:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT8	RO	0x6000:1E, 8
1A00:18	SubIndex 0x024	24. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT8	RO	0x6000:1F, 8
1A00:19	SubIndex 0x025	25. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT8	RO	0x6000:20, 8
1A00:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT8	RO	0x6000:21, 8
1A00:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT8	RO	0x6000:22, 8
1A00:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT8	RO	0x6000:23, 8
1A00:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT8	RO	0x6000:24, 8
1A00:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT8	RO	0x6000:25, 8
1A00:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT8	RO	0x6000:26, 8

Index 0x1A01 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 0x001	1. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x01 (Transmit accepted))	BOOLEAN	RO	0x6010:01, 1
1A01:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x02 (Receive request))	BOOLEAN	RO	0x6010:02, 1
1A01:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x03 (Init accepted))	BOOLEAN	RO	0x6010:03, 1
1A01:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x04 (Buffer full))	BOOLEAN	RO	0x6010:04, 1
1A01:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x05 (Parity error))	BOOLEAN	RO	0x6010:05, 1
1A01:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x06 (Framing error))	BOOLEAN	RO	0x6010:06, 1
1A01:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x07 (Overrun error))	BOOLEAN	RO	0x6010:07, 1
1A01:08	SubIndex 0x008	8. PDO Mapping entry (1 bits align)	Align1	RO	0x0000:00, 1
1A01:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x09 (Input length))	UINT8	RO	0x6010:09, 8
1A01:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x11 (Data In 0))	UINT8	RO	0x6010:11, 8
1A01:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x12 (Data In 1))	UINT8	RO	0x6010:12, 8
1A01:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x13 (Data In 2))	UINT8	RO	0x6010:13, 8
1A01:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x14 (Data In 3))	UINT8	RO	0x6010:14, 8
1A01:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x15 (Data In 4))	UINT8	RO	0x6010:15, 8
1A01:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x16 (Data In 5))	UINT8	RO	0x6010:16, 8
1A01:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x17 (Data In 6))	UINT8	RO	0x6010:17, 8
1A01:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x18 (Data In 7))	UINT8	RO	0x6010:18, 8
1A01:12	SubIndex 0x018	18. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x19 (Data In 8))	UINT8	RO	0x6010:19, 8
1A01:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1A (Data In 9))	UINT8	RO	0x6010:1A, 8
1A01:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1B (Data In 10))	UINT8	RO	0x6010:1B, 8
1A01:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1C (Data In 11))	UINT8	RO	0x6010:1C, 8
1A01:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1D (Data In 12))	UINT8	RO	0x6010:1D, 8
1A01:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1E (Data In 13))	UINT8	RO	0x6010:1E, 8
1A01:18	SubIndex 0x024	24. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1F (Data In 14))	UINT8	RO	0x6010:1F, 8
1A01:19	SubIndex 0x025	25. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x20 (Data In 15))	UINT8	RO	0x6010:20, 8
1A01:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x21 (Data In 16))	UINT8	RO	0x6010:21, 8
1A01:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x22 (Data In 17))	UINT8	RO	0x6010:22, 8
1A01:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x23 (Data In 18))	UINT8	RO	0x6010:23, 8
1A01:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x24 (Data In 19))	UINT8	RO	0x6010:24, 8
1A01:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x25 (Data In 20))	UINT8	RO	0x6010:25, 8
1A01:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x26 (Data In 21))	UINT8	RO	0x6010:26, 8

Index 0x1A04 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 1	UINT8	RO	0x17 (23_{dec})
1A04:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6001 (Status Ch.1), entry 0x01 (Status))	UINT16	RO	0x6001:01, 16
1A04:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT8	RO	0x6000:11, 8
1A04:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT8	RO	0x6000:12, 8
1A04:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT8	RO	0x6000:13, 8
1A04:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT8	RO	0x6000:14, 8
1A04:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT8	RO	0x6000:15, 8
1A04:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT8	RO	0x6000:16, 8
1A04:08	SubIndex 0x008	8. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT8	RO	0x6000:17, 8
1A04:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT8	RO	0x6000:18, 8
1A04:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT8	RO	0x6000:19, 8
1A04:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT8	RO	0x6000:1A, 8
1A04:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT8	RO	0x6000:1B, 8
1A04:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT8	RO	0x6000:1C, 8
1A04:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT8	RO	0x6000:1D, 8
1A04:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT8	RO	0x6000:1E, 8
1A04:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT8	RO	0x6000:1F, 8
1A04:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT8	RO	0x6000:20, 8
1A04:12	SubIndex 0x018	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT8	RO	0x6000:21, 8
1A04:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT8	RO	0x6000:22, 8
1A04:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT8	RO	0x6000:23, 8
1A04:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT8	RO	0x6000:24, 8
1A04:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT8	RO	0x6000:25, 8
1A04:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT8	RO	0x6000:26, 8

Index 0x1A05 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 1	UINT8	RO	0x17 (23_{dec})
1A05:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6011 (Status Ch.2), entry 0x01 (Status))	UINT16	RO	0x6011:01, 16
1A05:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x11 (Data In 0))	UINT8	RO	0x6010:11, 8
1A05:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x12 (Data In 1))	UINT8	RO	0x6010:12, 8
1A05:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x13 (Data In 2))	UINT8	RO	0x6010:13, 8
1A05:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x14 (Data In 3))	UINT8	RO	0x6010:14, 8
1A05:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x15 (Data In 4))	UINT8	RO	0x6010:15, 8
1A05:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x16 (Data In 5))	UINT8	RO	0x6010:16, 8
1A05:08	SubIndex 0x008	8. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x17 (Data In 6))	UINT8	RO	0x6010:17, 8
1A05:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x18 (Data In 7))	UINT8	RO	0x6010:18, 8
1A05:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x19 (Data In 8))	UINT8	RO	0x6010:19, 8
1A05:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1A (Data In 9))	UINT8	RO	0x6010:1A, 8
1A05:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1B (Data In 10))	UINT8	RO	0x6010:1B, 8
1A05:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1C (Data In 11))	UINT8	RO	0x6010:1C, 8
1A05:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1D (Data In 12))	UINT8	RO	0x6010:1D, 8
1A05:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1E (Data In 13))	UINT8	RO	0x6010:1E, 8
1A05:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1F (Data In 14))	UINT8	RO	0x6010:1F, 8
1A05:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x20 (Data In 15))	UINT8	RO	0x6010:20, 8
1A05:12	SubIndex 0x018	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x21 (Data In 16))	UINT8	RO	0x6010:21, 8
1A05:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x22 (Data In 17))	UINT8	RO	0x6010:22, 8
1A05:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x23 (Data In 18))	UINT8	RO	0x6010:23, 8
1A05:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x24 (Data In 19))	UINT8	RO	0x6010:24, 8
1A05:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x25 (Data In 20))	UINT8	RO	0x6010:25, 8
1A05:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x26 (Data In 21))	UINT8	RO	0x6010:26, 8

Index 0x1C00 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 0x001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1_{dec})
1C00:02	SubIndex 0x002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2_{dec})
1C00:03	SubIndex 0x003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3_{dec})
1C00:04	SubIndex 0x004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4_{dec})

Index 0x1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RO	0x04 (4_{dec})
1C12:01	SubIndex 0x001	1st allocated RxPDO (contains the index 0x of the associated RxPDO mapping object)	UINT16	RO	0x1600 (5632_{dec})
1C12:02	SubIndex 0x002	2nd allocated RxPDO (contains the index 0x of the associated RxPDO mapping object)	UINT16	RO	0x1601 (5633_{dec})

Index 0x1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RO	0x04 (4_{dec})
1C13:01	SubIndex 0x001	1st allocated TxPDO (contains the index 0x of the associated TxPDO mapping object)	UINT16	RO	0x1A00 (6656_{dec})
1C13:02	SubIndex 0x002	2nd allocated TxPDO (contains the index 0x of the associated TxPDO mapping object)	UINT16	RO	0x1A01 (6657_{dec})

Index 0x1C32 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 • Synchronous 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	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchronous 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)	UINT16	RO	0xC007 (49159_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00004E20 (20000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 1C32:03, 1C32:05, 1C32:06, 1C32:09, 1C33:03, 1C33:06, and 1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{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 0x1C33 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: Synchronous with SM 3 event (no outputs available) • 2: DC - Synchronous with SYNC0 Event • 3: DC - Synchronous with SYNC1 Event • 34: Synchronous with SM 2 event (outputs available)	UINT16	RW	0x0000 (0_{dec})
1C33:02	Cycle time	as 1C32:02	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	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: synchronous with SM 2 event is supported (outputs available) • Bit 1: synchronous 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 or 1C33:08)	UINT16	RO	0xC007 (49159_{dec})
1C33:05	Minimum cycle time	as 1C32:05	UINT32	RO	0x00004E20 (20000_{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})
1C33:08	Command	as 1C32:08	UINT16	RW	0x0000 (0_{dec})
1C33:09	Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:0B	SM event missed counter	as 1C32:11	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 1C32:12	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 1C32:13	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 1C32:32	BOOLEAN	RO	0x00 (0_{dec})

8.1.3 Profile-specific objects (0x6000-0xFFFF) [from hardware version 03]

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

Index 0x60n0 COM Inputs Ch. 1 (n = 0), Ch. 2 (n = 1)

Index (hex)	Name	Meaning		Data type	Flags	Default
60n0:0	COM Inputs Ch. 1 + Ch. 2	Max. SubIndex (hex)		UINT8	RO	0x26 (38_{dec})
60n0:01	Transmit accepted	The terminal acknowledges receipt of data by changing the state of this bit. Only now new data can be transferred from the controller to the terminal.		BOOLEAN	RO	0x00 (0_{dec})
60n0:02	Receive request	By changing the state of this bit, the terminal informs the controller that the DataIn bytes contain the number of bytes displayed in "Input length [▶ 129]". The controller must acknowledge receipt of the data by changing the state of the ReceiveAccepted [▶ 130] bit. Only then new data can be transferred from the terminal to the controller.		BOOLEAN	RO	0x00 (0_{dec})
60n0:03	Init accepted	0	The terminal is ready again for serial data exchange.	BOOLEAN	RO	0x00 (0_{dec})
		1	Initialization was completed by the terminal.			
60n0:04	Buffer full	The reception FIFO is full. All incoming data will be lost from this point on!		BOOLEAN	RO	0x00 (0_{dec})
60n0:05	Parity error	A parity error has occurred.		BOOLEAN	RO	0x00 (0_{dec})
60n0:06	Framing error	A framing error has occurred.		BOOLEAN	RO	0x00 (0_{dec})
60n0:07	Overrun error	An overrun error has occurred.		BOOLEAN	RO	0x00 (0_{dec})
60n0:09	Input length	Number of input bytes available for transfer from the terminal to the controller.		UINT8	RO	0x00 (0_{dec})
60n0:11	Data In 0	Input byte 0		UINT8	RO	0x00 (0_{dec})
...	...	....		...	...	...
60n0:26	Data In 21	Input byte 21		UINT8	RO	0x00 (0_{dec})

Index 0x60n1 Status Ch. 1 (n = 0), Ch. 2 (n = 1)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n1:0	Status Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x01 (01_{dec})
60n1:01	Status	Status word	UINT16	RO	0x00 (0_{dec})

Index 0x70n0 COM Outputs Ch. 1 (n = 0), Ch. 2 (n = 1)

Index (hex)	Name	Meaning	Data type	Flags	Default
70n0:0	COM Outputs Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x26 (38_{dec})
70n0:01	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 [▶ 130]". The terminal acknowledges receipt of the data by changing the state of the "TransmitAccepted [▶ 129]" bit. Only now new data can be transferred from the controller to the terminal.	BOOLEAN	RO	0x00 (0_{dec})
70n0:02	Receive accepted	The controller acknowledges receipt of data by changing the state of this bit. Only then new data can be transferred from the terminal to the controller.	BOOLEAN	RO	0x00 (0_{dec})
70n0:03	Init request	0 The controller once again requests the terminal to prepare for serial data exchange. 1 The controller requests terminal for initialization. The transmit and receive functions will be blocked, the FIFO pointer will be reset and the interface will be initialized with the values of the responsible Settings object. The execution of the initialization will be acknowledged by the terminal with the 'Init accepted [▶ 129]' bit.	BOOLEAN	RO	0x00 (0_{dec})
70n0:04	Send continuous	Continuous sending of data from the FIFO. The send buffer is filled (up to 128 bytes) by the controller. The filled buffer contents will be sent on the rising edge of the bit. If the data has been transmitted, the terminal informs the controller by setting the "Init accepted [▶ 129]" bit. "Init accepted [▶ 129]" is cleared with "SendContinuous [▶ 130]".	BOOLEAN	RO	0x00 (0_{dec})
70n0:09	Output length	Number of output bytes available for transfer from the controller to the terminal.	UINT8	RO	0x00 (0_{dec})
70n0:11	Data Out 0	Output byte 0	UINT8	RO	0x00 (0_{dec})
...	...	....	...	...	...
70n0:26	Data Out 21	Output byte 21	UINT8	RO	0x00 (0_{dec})

Index 0x70n1 Ctrl Ch. 1 (n = 0), Ch. 2 (n = 1)

Index (hex)	Name	Meaning	Data type	Flags	Default
70n1:0	Crtl Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x01 (01_{dec})
70n1:01	Status	Control word	UINT16	RO	0x00 (0_{dec})

Index 0xA0n0 COM Diag data Ch. 1 (n = 0), Ch. 2 (n = 1)

Index (hex)	Name	Meaning	Data type	Flags	Default
A0n0:0	COM Diag data Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x12 (18_{dec})
A0n0:01	Buffer overflow	A buffer overflow has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A0n0:02	Parity error	A parity error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A0n0:03	Framing error	A framing error has occurred	BOOLEAN	RO	0x00 (0_{dec})
A0n0:04	Overrun error	An overrun error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A0n0:05	Buffer full	The reception FIFO is full. All incoming data will be lost from this point on!	BOOLEAN	RO	0x00 (0_{dec})
A0n0:11	Data bytes in send buffer	Number of data bytes in the send FIFO	UINT16	RO	0x0000 (0_{dec})
A0n0:12	Data bytes in receive buffer	Number of data bytes in the receive FIFO	UINT16	RO	0x0000 (0_{dec})

Index 0xF000 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 0xdistance	Index (hex) interval of the objects of the individual channels	UINT16	RO	0x0010 (16_{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0004 (4_{dec})

Index 0xF008 Code word

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

Index 0xF010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Max. SubIndex (hex)	UINT8	RW	0x04 (4_{dec})
F010:01	SubIndex 0x001	-	UINT32	RW	0x00000258 (600_{dec})
F010:02	SubIndex 0x002	-	UINT32	RW	0x00000258 (600_{dec})

8.2 Control and status data

The control and status data are located in the first 16 bits of the input and output process image. Communication between the terminal and the controller is controlled via this data.

Status data

Bit position	Name	Meaning	Data type
0	Transmit accepted	The terminal acknowledges receipt of data by changing the state of this bit. Only now new data can be transferred from the controller to the terminal.	BOOLEAN
1	Receive request	By changing the state of this bit, the terminal informs the controller that the DataIn bytes contain the number of bytes displayed in “Input length”. The controller must acknowledge receipt of the data by changing the state of the ReceiveAccepted bit. Only then new data can be transferred from the terminal to the controller.	BOOLEAN
2	Init accepted	0 The terminal is ready again for serial data exchange. 1 Initialization was completed by the terminal.	BOOLEAN
3	Buffer full	The reception FIFO is full. All incoming data will be lost from this point on!	BOOLEAN
4	Parity error	A parity error has occurred.	BOOLEAN
5	Framing error	A framing error has occurred.	BOOLEAN
6	Overrun error	An overrun error has occurred.	BOOLEAN
7	-		
8...15	Input length	Number of input bytes available for transfer from the terminal to the controller.	UINT8

Control data

Bit position	Name	Meaning	Data type
0	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”. The terminal acknowledges receipt of the data by changing the state of the “TransmitAccepted” bit. Only now new data can be transferred from the controller to the terminal.	BOOLEAN
1	Receive accepted	The controller acknowledges receipt of data by changing the state of this bit. Only then new data can be transferred from the terminal to the controller.	BOOLEAN
2	Init request	0 The controller once again requests the terminal to prepare for serial data exchange. 1 The controller requests terminal for initialization. The transmit and receive functions will be blocked, the FIFO pointer will be reset and the interface will be initialized with the values of the responsible Settings object. The execution of the initialization will be acknowledged by the terminal with the ‘Init accepted’ bit.	BOOLEAN
3	Send continuous	Continuous sending of data from the FIFO. The send buffer is filled (up to 128 bytes) by the controller. The filled buffer contents will be sent on the rising edge of the bit. If the data has been transmitted, the terminal informs the controller by setting the “Init accepted” bit. “Init accepted” is cleared with “SendContinuous”.	BOOLEAN
4...7	-		
8...15	Output length	Number of output bytes available for transfer from the controller to the terminal.	UINT8

PDO Assignment

The terminal makes two input/output process images available for each channel. These differ only in the representation of the control/status data.

In the case of bitwise representation, the data is made available as shown in the tables above. In the case of word-wise representation, the first 16 bits of the process data are combined in one word.

The process data objects begin as standard with a control/status word. This PDO assignment is required in order to use the ‘TwinCAT PLC Serial Communication’ library.

Status Inputs	Control Outputs	Representation
0x1A00	0x1600	bitwise Ch. 1
0x1A01	0x1601	bitwise Ch. 2
0x1A02	0x1602	bitwise Ch. 3
0x1A03	0x1603	bitwise Ch. 4
0x1A04	0x1604	word-wise Ch 1
0x1A05	0x1605	word-wise Ch 2
0x1A06	0x1606	word-wise Ch 3
0x1A07	0x1607	word-wise Ch 4

9 Appendix

9.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

9.2 Firmware compatibility

Beckhoff EtherCAT devices are delivered with the latest available firmware version. Firmware and hardware must be compatible; not every combination ensures compatibility. The overview below shows the hardware versions on which a firmware can be operated.

Note

- The newest possible firmware for the respective hardware should be implemented
- Beckhoff is not under any obligation to provide customers with free firmware updates for delivered products.

NOTICE

Risk of damage to the device!

Observe the notes on the firmware update in the EtherCAT System Manual in chapter “Firmware Update”.
If a device is set to BOOTSTRAP mode for firmware update, it may not check whether the new firmware is suitable during download.
This may result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

EL6001			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 - 02	01		05/2006
	02		12/2006
	03		04/2008
	04		04/2008
03 - 18*	05	EL6001-0000-0016	12/2009
	06		04/2010
	07	EL6001-0000-0017	03/2011
		EL6001-0000-0018	10/2012
	08	EL6001-0000-0019	05/2014
	09	EL6001-0000-0020	08/2014
	10		05/2015
	11		06/2017
	12	EL6001-0000-0021	01/2023
	13		05/2023
	14		07/2023
	15	EL6001-0000-0022	01/2024
	16*		02/2024

EL6002			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 - 18*	01	EL6002-0000-0016	12/2009
	02		06/2010
	03	EL6002-0000-0017	11/2012
		EL6002-0000-0018	08/2013
	04*	EL6002-0000-0019	05/2014

EL6021			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 - 02	01		05/2005
	02		12/2006
	03		04/2008
03 - 20*	04	EL6021-0000-0016	11/2009
	05		04/2010
		EL6021-0000-0017	10/2012
	06	EL6021-0000-0018	08/2013
	07	EL6021-0000-0019	06/2014
	08	EL6021-0000-0020	10/2014
	09	EL6021-0000-0021	01/2020
	10*	EL6021-0000-0022	01/2023

EL6021-0021			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 – 05*	06	EL6021-0021-0018	09/2013
	07	EL6021-0021-0019	06/2014
	08	EL6021-0021-0020	10/2014
	09	EL6021-0021-0021	01/2020
	10*	EL6021-0021-0022	01/2023

EL6022			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 - 19*	01	EL6022-0000-0016	01/2010
	02		06/2010
	03*		09/2011
		EL6022-0000-0017	08/2012
		EL6022-0000-0018	08/2013
		EL6022-0000-0019	03/2015

*) This is the current compatible firmware/hardware version at the time of the preparing this documentation. Check on the Beckhoff website whether more up-to-date documentation is available.

9.3 Support and Service

Beckhoff Support and Service provides assistance with questions regarding Beckhoff products and system solutions, as well as with their planning, commissioning, and operation.

Beckhoff subsidiaries and representative offices

Please contact your Beckhoff subsidiary or your representative office for local support and service for Beckhoff products. You can find the addresses of Beckhoff subsidiaries and representative offices at: www.beckhoff.com/worldwide.

For Beckhoff products and technologies, we offer Training programs, tutorials, and webinars.

Beckhoff Support

Beckhoff Support provides assistance with technical questions regarding the use, planning, programming, and commissioning of Beckhoff products.

☎ Hotline:	+49 5246 963-157
✉ Email:	support@beckhoff.com, Support form
Web page:	www.beckhoff.com/support

Beckhoff Service

Our service specialists are here to assist you with any questions you may have regarding after-sales service processes, such as replacement parts, repairs, etc.

☎ Hotline:	+49 5246 963-460
✉ Email:	service@beckhoff.com, Service form
Web page:	www.beckhoff.com/service

Download finder

The Download finder serves as a central source for downloadable documents and files related to Beckhoff products, such as application reports, technical documentation, drawings, and configuration files.

Beckhoff Information System

The Beckhoff Information System is the central online documentation resource for Beckhoff products and contains documentation on products and automation topics, as well as sample code.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl, Germany

☎ Phone:	+49 5246 963-0
✉ Email:	info@beckhoff.com
Web page:	www.beckhoff.com

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

Excel, IntelliSense, Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

More Information:
www.beckhoff.com/EL6xxx

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com