BECKHOFF New Automation Technology

Documentation | EN

EL5131

1-Channel-Incremental Encoder Interface, 5 V with 2 parameterizable 24 V-DC Outputs



2025-07-31 | Version: 1.4.0

Table of contents

1 Foreword 7
1.1 Notes on the documentation 7
1.2 Safety instructions 8
1.3 Guide through documentation 9
1.4 Documentation issue status 10
1.5 Version identification of EtherCAT devices 11
1.5.1 General notes on marking 11
1.5.2 Version identification of EL terminals 12
1.5.3 Beckhoff Identification Code (BIC) 12
1.5.4 Electronic access to the BIC (eBIC) 14

2 EL5131 - Product description 16
2.1 Introduction 16
2.2 Technical data 17
2.3 Overview of functions 19
2.4 Technical properties 20
2.4.1 Supported encoders / signal types 20
2.4.2 Latch and Gate/Latch inputs 23
2.4.3 Status Input 24
2.4.4 Encoder operating voltage (supply voltage) 24
2.4.5 Outputs 25
2.5 Start 26

3 EL51xx series overview 27

4 Incremental encoder basics 28

5 Basics communication 29
5.1 EtherCAT basics 29
5.2 EtherCAT cabling – wire-bound 29
5.3 General notes for setting the watchdog 31
5.4 EtherCAT State Machine 32
5.5 CoE Interface 34
5.6 Distributed Clock 39

6 Mounting and wiring 40
6.1 Instructions for ESD protection 40
6.2 Installation on mounting rails 41
6.3 UL notice 44
6.4 Connection 45
6.4.1 Connection system 45
6.4.2 Wiring 47
6.4.3 Shielding 48
6.5 Note - power supply 48
6.6 Installation position for operation with or without fan 49
6.7 Positioning of passive Terminals 53
6.8 EL5131 – connection 54
6.8.1 RS422 mode 55

6.8.2 TTL mode ... 57
6.8.3 Open Collector mode ... 59
6.9 EL5131 - LEDs ... 61
6.10 Disposal ... 62

7 Commissioning ... 63
7.1 TwinCAT Quick Start ... 63
7.1.1 TwinCAT 2 ... 66
7.1.2 TwinCAT 3 ... 76
7.2 TwinCAT Development Environment ... 89
7.2.1 Installation of the TwinCAT real-time driver ... 90
7.2.2 Notes regarding ESI device description ... 96
7.2.3 TwinCAT ESI Updater ... 100
7.2.4 Distinction between Online and Offline ... 100
7.2.5 OFFLINE configuration creation ... 101
7.2.6 ONLINE configuration creation ... 106
7.2.7 EtherCAT subscriber configuration ... 114
7.2.8 NC configuration (motion) ... 123
7.3 General Commissioning Instructions for an EtherCAT Slave ... 127

8 EL5131 - Commissioning ... 135
8.1 Process data ... 135
8.1.1 Sync Manager (SM) ... 135
8.1.2 PDO assignment ... 136
8.1.3 Predefined PDO Assignment ... 141
8.1.4 Synchronicity mode ... 144
8.1.5 EtherCAT cycle time depending on Predefined PDO Assignment ... 144
8.1.6 Illustration of the process data and structural contents ... 145
8.2 Overview of functions ... 146
8.3 Basic functions ... 148
8.3.1 Counter value ... 148
8.3.2 Counter value reset ... 154
8.3.3 Set counter value ... 155
8.3.4 Detect counting direction ... 157
8.3.5 Save counter value ... 159
8.3.6 Lock counter value ... 163
8.4 Extended functionalities ... 164
8.4.1 Frequency measurement ... 164
8.4.2 Period value measurement ... 166
8.4.3 Velocity, speed calculation ... 167
8.4.4 Duty cycle evaluation ... 169
8.4.5 Micro-increments ... 170
8.4.6 Timestamp function ... 172
8.4.7 Adjustable interference pulse filters ... 173
8.4.8 Plausibility check ... 175
8.5 Output function ... 176
8.5.1 Set switching characteristics ... 178

8.5.2 Set state in case of bus error ... 179
8.5.3 Switching outputs via PLC variable ... 181
8.5.4 Switching outputs with preset values ... 182
8.5.5 Function block “MC_DigitalCamSwitch_EL5131” ... 193
8.6 Inputs ... 194
8.6.1 Zero pulse C input ... 195
8.6.2 Latch input (Latch extern) ... 197
8.6.3 Gate/Latch input ... 198
8.6.4 Input Status Input ... 200
8.7 Outputs ... 201

9 Diagnostics ... 202
9.1 Diagnostics - basic principles of diag messages ... 202
9.2 EL5131 - Diagnostics ... 212
9.2.1 Overflow and underflow of the counter limits ... 212
9.2.2 RS422 - wire break and short-circuit detection (open circuit) ... 213
9.2.3 Monitoring of the encoder operating voltage ... 214
9.2.4 Status Input diagnostics ... 214
9.2.5 Plausibility check ... 215
9.2.6 Exceeding the filter frequency ... 216
9.2.7 Short circuit/overload diagnosis of outputs ... 217

10 EL5131 - object description and parameterization ... 218
10.1 Restore object ... 218
10.2 Configuration data ... 219
10.3 Input data ... 223
10.4 Output data ... 225
10.5 Information / diagnostic data (channel specific) ... 229
10.6 Information / diagnostic data (device specific) ... 229
10.7 Standard objects ... 229

11 Appendix ... 244
11.1 EtherCAT AL Status Codes ... 244
11.2 Firmware compatibility ... 244
11.3 Firmware Update EL/ES/EM/ELM/EP/EPP/ERPxxxx ... 244
11.3.1 Device description ESI file/XML ... 246
11.3.2 Firmware explanation ... 249
11.3.3 Updating controller firmware *.efw ... 249
11.3.4 FPGA firmware *.rbf ... 251
11.3.5 Simultaneous updating of several EtherCAT devices ... 255
11.4 Restoring the delivery state ... 256
11.5 Support and Service ... 258

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
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® is registered trademark and patented technology, 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

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

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.

1.3 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 device / 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
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

1.4 Documentation issue status

Version	Comment
1.4.0	• Update chapter “Technical data” • Chapter “UL notice” added • Update revision status • Structural update
1.3	• Chapter “Function block ‘MC_DigitalCamSwitch_EL5131’” added
1.2	• Update chapter “Technical data” • Structural update
1.1	• Update chapter “Switching outputs with preset values”
1.0	• First release
0.1	• First preliminary documentation for EL5131

1.5 Version identification of EtherCAT devices

1.5.1 General notes on marking

Designation

A Beckhoff EtherCAT device has a 14-digit designation, made up of

- family key
- type
- version
- revision

Example	Family	Type	Version	Revision
EL3314-0000-0016	EL terminal 12 mm, non-pluggable connection level	3314 4-channel thermocouple terminal	0000 basic type	0016
ES3602-0010-0017	ES terminal 12 mm, pluggable connection level	3602 2-channel voltage measurement	0010 high-precision version	0017
CU2008-0000-0000	CU device	2008 8-port fast ethernet switch	0000 basic type	0000

Notes

- The elements mentioned above result in the **technical designation**. EL3314-0000-0016 is used in the example below.
- EL3314-0000 is the order identifier, in the case of “-0000” usually abbreviated to EL3314. “-0016” is the EtherCAT revision.
- The **order identifier** is made up of
  - family key (EL, EP, CU, ES, KL, CX, etc.)
  - type (3314)
  - version (-0000)
- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.
  From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. *“EL2872 with revision 0022 and serial number 01200815”*.
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

1.5.2 Version identification of EL terminals

The serial number/ data code for Beckhoff IO devices is usually the 8-digit number printed on the device or on a sticker. The serial number indicates the configuration in delivery state and therefore refers to a whole production batch, without distinguishing the individual modules of a batch.

Structure of the serial number: **KK YY FF HH**	Example with serial number 12 06 3A 02:
KK - week of production (CW, calendar week)	12 - production week 12
YY - year of production	06 - production year 2006
FF - firmware version	3A - firmware version 3A
HH - hardware version	02 - hardware version 02



Fig. 1: EL2872 with revision 0022 and serial number 01200815

1.5.3 Beckhoff Identification Code (BIC)

The Beckhoff Identification Code (BIC) is increasingly being applied to Beckhoff products to uniquely identify the product. The BIC is represented as a Data Matrix Code (DMC, code scheme ECC200), the content is based on the ANSI standard MH10.8.2-2016.

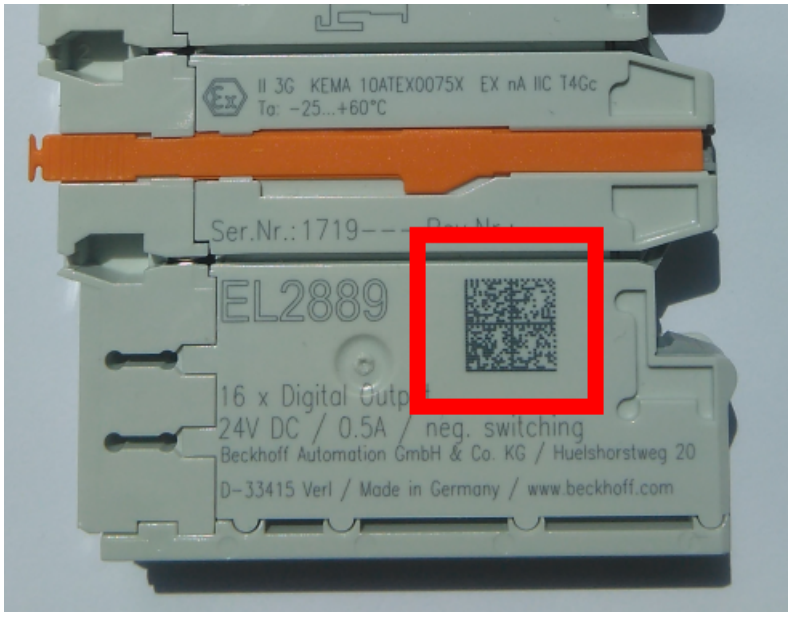


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

The BIC will be introduced step by step across all product groups.

Depending on the product, it can be found in the following places:

- on the packaging unit
- directly on the product (if space suffices)
- on the packaging unit and the product

The BIC is machine-readable and contains information that can also be used by the customer for handling and product management.

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, spaces are added to it.

Following information is possible, positions 1 to 4 are always present, the other according to need of production:

Position	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	**1P**072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	SBTN	12	**SBTN**k4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	**1K**EL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	**Q**1
5	Batch number	Optional: Year and week of production	2P	14	**2P**401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	**51S**678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	12	**30P**F971, 2*K183
...					

Further types of information and data identifiers are used by Beckhoff and serve internal processes.

Structure of the BIC

Example of composite information from positions 1 to 4 and with the above given example value on position 6. The data identifiers are highlighted in bold font:

1P072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:



Fig. 3: Example DMC **1P**072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, position 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTICE

This information has been carefully prepared. However, the procedure described is constantly being further developed. We reserve the right to revise and change procedures and documentation at any time and without prior notice. No claims for changes can be made from the information, illustrations and descriptions in this documentation.

1.5.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

The Beckhoff Identification Code (BIC) is applied to the outside of Beckhoff products in a visible place. If possible, it should also be electronically readable.

The interface that the product can be electronically addressed by is crucial for the electronic readout.

K-bus devices (IP20, IP67)

Currently, no electronic storage or readout is planned for these devices.

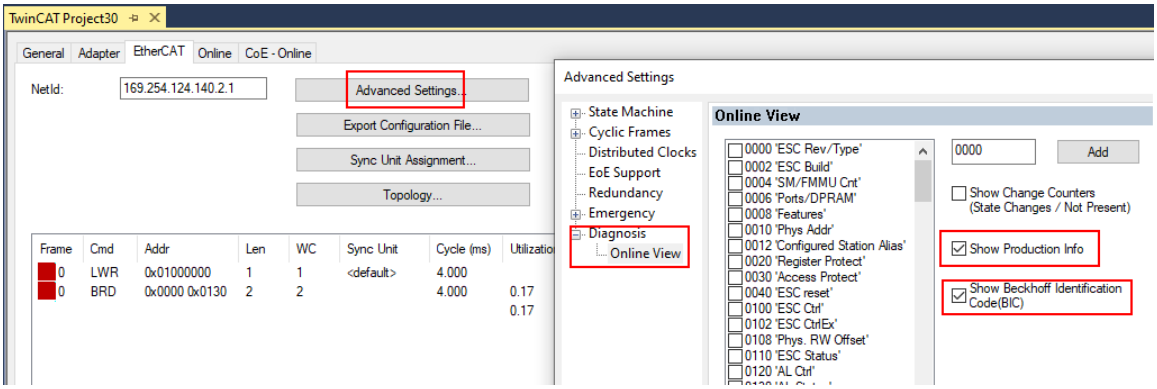
EtherCAT devices (IP20, IP67)

All Beckhoff EtherCAT devices have an ESI-EEPROM which contains the EtherCAT identity with the revision number. The EtherCAT slave information, also colloquially known as the ESI/XML configuration file for the EtherCAT master, is stored in it. See the corresponding chapter in the EtherCAT system manual (Link) for the relationships.

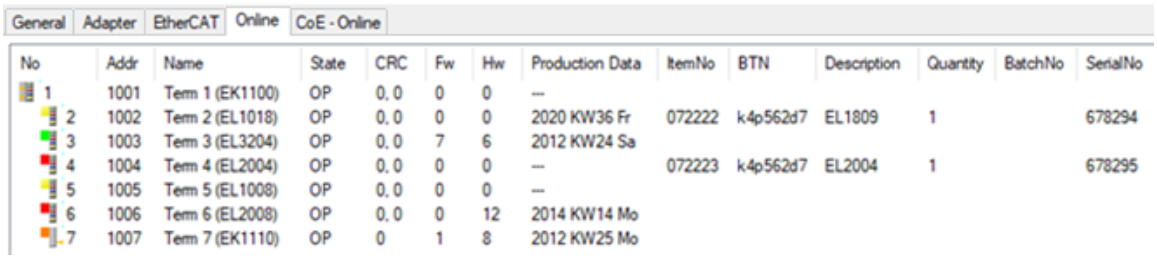
Beckhoff also stores the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

The user can electronically access the eBIC (if present) as follows:

- With all EtherCAT devices, the EtherCAT master (TwinCAT) can read the eBIC from the ESI-EEPROM
  - From TwinCAT 3.1 build 4024.11, the eBIC can be displayed in the online view.
  - To do this, check the "Show Beckhoff Identification Code (BIC)" checkbox under EtherCAT → Advanced Settings → Diagnostics:

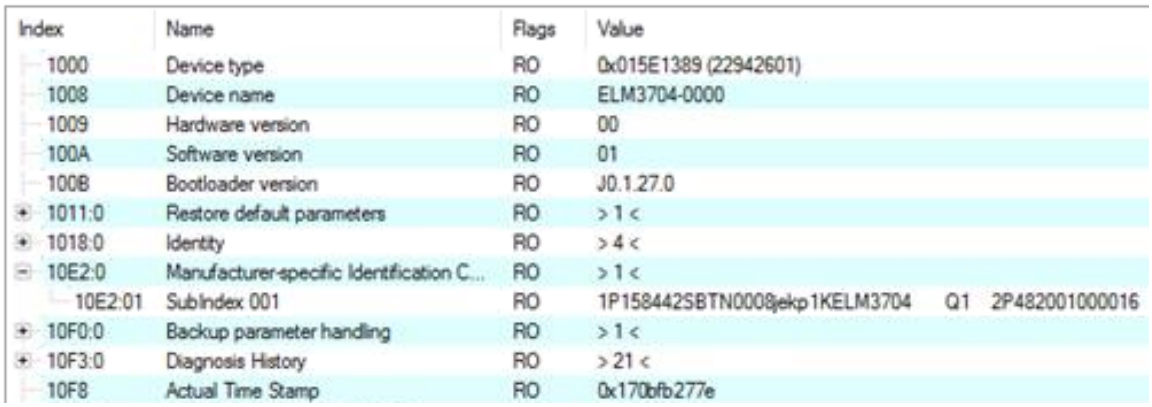


  - The BTN and its contents are then displayed:



  - Note: As shown in the figure, the production data HW version, FW version, and production date, which have been programmed since 2012, can also be displayed with "Show production info".
  - Access from the PLC: From TwinCAT 3.1. build 4024.24, the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0.
- EtherCAT devices with a CoE directory may also have the object 0x10E2:01 to display their own eBIC, which can also be easily accessed by the PLC:

- The device must be in PREOP/SAFEOP/OP for access:



Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	SubIndex 001	RO	1P158442SBTN0008jekp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

  - The object 0x10E2 will be preferentially introduced into stock products in the course of necessary firmware revision.
  - From TwinCAT 3.1. build 4024.24, the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0
- The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2_Utilities* as of TwinCAT 3.1 build 4024.24
  - F_SplitBIC: The function splits the Beckhoff Identification Code (BIC) sBICValue into its components using known identifiers and returns the recognized substrings in the ST_SplittedBIC structure as a return value
  - BIC_TO_BTN: The function extracts the BTN from the BIC and returns it as a return value
- Note: If there is further electronic processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
  The new BIC information is written as an additional category in the ESI-EEPROM during device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored using a category in accordance with the ETG.2010. ID 03 tells all EtherCAT masters that they may not overwrite these data in the event of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see here. The EEPROM therefore requires approx. 50..200 bytes of memory.
- Special cases
  - If multiple hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC information.
  - If multiple non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC information.
  - If the device consists of several sub-devices which each have their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

PROFIBUS, PROFINET, and DeviceNet devices

Currently, no electronic storage or readout is planned for these devices.

2 EL5131 - Product description

2.1 Introduction

1-channel encoder interface, incremental 5 V DC, 5 MHz, with 2 x digital output 24 V DC

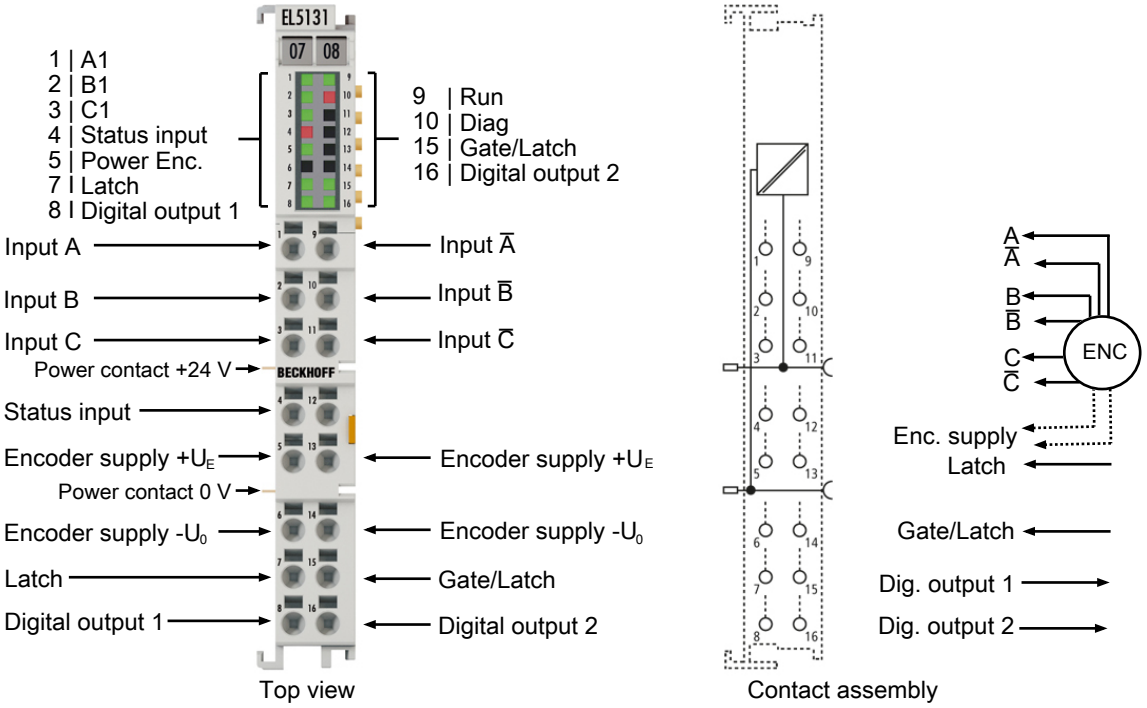


Fig. 4: EL5131

The EL5131 EtherCAT Terminal is an interface for the direct connection of incremental encoders with differential signals (RS422) or single-ended signals (TTL and Open Collector).

Two parameterizable 24 V_{DC} push-pull, tristate outputs are available. These are set after a specified count, frequency or velocity value is reached. This allows an exact reaction time independent of the fieldbus cycle.

Two additional 24 V digital inputs are available for saving, locking and setting the counter value. Furthermore, there is an option to set the counter value to a specified value or to lock the counter directly via the process data.

The alarm output of an encoder can be connected and evaluated via the Status Input. The encoders can be supplied with 5 V, 12 V or 24 V directly via the connection points of the terminal.

The EL5131 allows the measurement of a period, frequency and velocity with a resolution of 10 ns. Additionally a duty cycle measurement of the incoming signal is implemented.

Due to the optional interpolating microincrement function, the EL5131 can supply even more precise axis positions for dynamic axes. Furthermore, it supports synchronous reading of the encoder value together with other input data in the EtherCAT system via the high-precision EtherCAT Distributed Clocks (DC). In addition, timestamps for the last registered incremental edge, the edge at the latch input, the zero pulse track C and the output switching limits are available.

The use of encoder profiles allows the simple and fast linking of the process data to the Motion Control application.

Quick links

Basics communication [▶ 29] | LEDs [▶ 61] and connection [▶ 54] | Configuration data [▶ 219]
Creation of the TwinCAT configuration [▶ 106] | Process data [▶ 135] | EL5131 - Object description [▶ 218]

2.2 Technical data

Encoder	EL5131
Encoder type	Incremental, differential (RS422), single-ended (TTL, Open Collector), counter, pulse generator
Number of channels	1
Encoder connection	1 x A, B, C Differential inputs (RS422): A, $\overline{A}$, B, $\overline{B}$, C, $\overline{C}$ Single-ended connection (TTL, Open Collector): A, B, C Counter, pulse generator: A, B
Additional inputs	Latch, Gate/Latch (24 V_{DC}, t_{ON} > 1 µs), Status Input (max. 5 V_{DC}, negative switching, t_{ON} > 10 µs)
Encoder operating voltage	switchable, (generated from the 24 V_{DC} – power contacts) • 5 V_{DC} (preset) up to 0.3 A, • 12 V_{DC} up to 0.15 A, • 24 V_{DC} up to 0.3 A
Counter	32-bit (default) or 16/32-bit switchable
Cut-off frequency	RS422 mode: max. 20 million increments/s with 4-fold evaluation, equivalent to 5 MHz TTL mode: max. 4 million increments/s with 4-fold evaluation, equivalent to 1 MHz Open Collector: max. 400,000 increments/s with 4-fold evaluation, equivalent to 100 kHz
Quadrature decoder	Switchable: 4-fold evaluation (preset), 2-fold evaluation, 1-fold evaluation switchable

Digital outputs	EL5131
Additional outputs	2 x push-pull, high-resistance switchable (tristate)
Nominal voltage of the outputs	24 V_{DC} (-15 %/+20 %)
Load type	ohmic, inductive, lamp load
Output current	Push-operation: max. 0.5 A per channel pull-operation: 0.15 A per channel
Short circuit current	Push-operation: < 1 A typ. pull-operation: < 0.2 A typ.
Switching times	T_{ON}: 3 µs typ. (Push-operation) T_{OFF}:4 µs typ. (Pull-operation), 7 µs typ. (tristate-operation)

Function and communication	EL5131
Timestamp [▶ 141] resolution	1 ns
Distributed Clocks	Yes
Cycle time	min. 100 µs without output function 187.5 µs when using the output function
Microincrements resolution	1/256 bit microincrements
Wire break detection to encoder	Yes for RS422 encoder
Supports NoCoEStorage function [▶ 35]	Yes
Configuration	via TwinCAT System Manager [▶ 114]
Special features	• Wire break detection • Period value, frequency, velocity and duty cycle measurement • Microincrements • Timestamp on last incremental edge, zero pulse C, latch input, gate/latch 2 input and on the output switching limits

Supply and potentials	EL5131
Current consumption via E-bus	240 mA typ.
Current consumption from the power contacts	20 mA typ. + load
Electrical isolation	500 V (E-bus/field voltage)

Environmental conditions	EL5131
Permissible ambient temperature range during operation	0°C ... +55°C
Permissible ambient temperature range during storage	-25°C ... +85°C
Permissible relative air humidity	95 %, no condensation

General data	EL5131
Weight	approx. 50 g
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting [▶ 41]	on 35 mm mounting rail conforms to EN 60715
Installation position [▶ 49]	• without fan cartridge ZB8610: standard installing position • with fan cartridge ZB8610: standard installing position, other installing positions (example 1 and 2) see chapter Installation position for operation with fan [▶ 49]

Standards and approvals	EL5131
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
Approvals/markings*	CE, EAC, UKCA, cULus [▶ 44]

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

2.3 Overview of functions

EL5131 functions	
Set counter value via	PLC variable
	Zero pulse C
	Latch input
Reset counter value via	Zero pulse C
	Latch input
Save counter value via	Zero pulse C
	Latch input
	Gate/Latch input
Lock counter value via	PLC variable
	Gate/Latch input
Detect counting direction	
Detect reversion of rotation	
Frequency calculation	
Period duration calculation	
Duty cycle evaluation	
Microincrements	
Timestamp function	
Filter function	
Plausibility check	
Switch output via	PLC variable
	Automatic switching function: • when specified counter values are reached • when specified frequency or velocity values are reached

2.4 Technical properties

The EL51xx series incremental encoder interface terminals enable incremental encoders to be connected to bus couplers or the PLC. In addition to the encoder inputs A, B and optional zero pulse C, up to two additional 24 V_{DC} inputs are available (latch and gate/latch), which can be used for resetting, setting, blocking and storing the counter value. If the incremental encoder has a fault signal output, this can be connected to the Status Input (5 V_{DC}).

- The following inputs are available with the respective technical characteristics:
  - Encoder connection:
    differential signals according to RS422 and single-ended signals from TTL encoders and Open Collector encoders are supported.
  - Latch input and Gate/Latch input
  - Status Input
- The terminal provides a parameterizable encoder supply.
  - Encoder operating voltage
- Two parameterizable 24 V_{DC} push-pull, tristate outputs with short circuit/overload diagnostics are available. These are set via a PLC variable or after reaching specified count, frequency or velocity values.

NOTICE

Fast digital inputs – interference from interfering devices

Please note that the input wiring has very little filtering. It has been optimized for fast signal transmission from the input to the evaluation unit. In other words, rapid level changes/pulses in the µs range and/or high-frequency interference signals from devices (e.g. proportional valves, stepper motor or DC motor output stages) arrive at the evaluation unit almost unfiltered/unattenuated. These interferences can be incorrectly detected as a signal.

- To suppress interference, an additional input filter can be parameterized.
- Furthermore, EMC-compliant cabling and the use of separate power supply units for the terminal and the devices causing interference are recommended.

2.4.1 Supported encoders / signal types

Differential signals according to RS422 are intended as encoder connection. Single-ended signals from TTL encoders and also signals from Open Collector encoders are also options, through internal pull-up resistances.

The following signal types are supported:

Encoder	Signal type	Setting in Index 0x80n1:1D * "Counter mode"	Cut-off frequency	Comments
Encoder with or without zero pulse track C	RS422 (diff. input)	0	20 million increments/s with 4-fold evaluation, corresponds to 5 MHz per track	Signal levels according to RS422 are expected Detection of wire break and short circuit [▶ 21]
Counter/pulse generator with or without zero pulse track C		1		
Encoder with or without zero pulse track C	TTL (single ended)	2	4 million increments/s with 4-fold evaluation, corresponds to 1 MHz per track	A voltage level of nominally 2.0 V to 6.0 V with a current of 2.1 mA or higher is expected. No wire break detection
Counter/pulse generator with or without zero pulse track C		3		
Encoder with or without zero pulse track C	open collector	4	400,000 increments/s with 4-fold evaluation, corresponds to 100 kHz per track	No wire break detection
Counter/pulse generator with or without zero pulse track C		5		
*) depending on the number of channels (n = 0 for channel 1 and n = 1 for channel 2)				

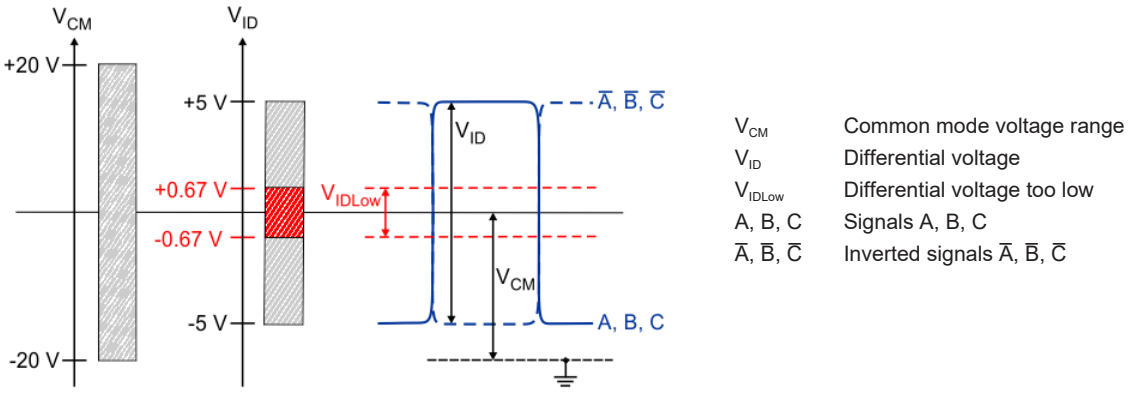
The correct wiring for the respective encoder can be found in chapter Connection [▶ 54].

2.4.1.1 Signal type RS422 (diff. input)

Differential signal levels according to RS422 are expected with the following settings in "Counter mode" (0x80n1:1D):

- 0: Encoder RS422 (diff. input)
- 1: Counter RS422 (diff. input)

A cut-off frequency of up to 20 million increments per second is permissible with 4-fold evaluation (corresponds to 5 MHz).



RS422 signal level

NOTICE

Exceeding of Common Mode range

Exceeding the Common Mode voltage range can lead to destruction of the device.

RS422 - wire break and short circuit detection (open circuit)

In the RS422 (differential input) modes it is possible to detect a wire break or short circuit at the individual encoder inputs.

- In case of wire break, e.g. between input A and input $\overline{A}$,
  - the differential voltage V_{ID} is almost 0 V,
  - which leads to an error with low differential voltage.
- In case of a short circuit, e.g. between input A and input $\overline{A}$, the error behavior is similar to a wire break and also leads to error detection.

Activation and diagnosis of a wire break or short circuit can be found in the chapter RS422 - Wire break and short circuit detection (Open circuit). [▶ 213]

Error bits not permanently set in case of wire break at an encoder input

If a wire break is only present at one encoder input (e.g. only track A), it may happen in individual cases that the differential voltage (V_{ID}) is above the limit range (V_{IDLow}) due to the applied common mode voltage (V_{CM}).
This means that the error is not clearly identified.

The corresponding error bits ("Open circuit" and "Error A") are not permanently present!

NOTICE

Differential and single-ended connection

The RS422 signal transmits a differential voltage, which makes the signal less sensitive to interference compared to a single-ended signal.

- If the encoder signal is to be transmitted over longer distances or at higher frequencies, an encoder with RS422 signals is recommended.
- Shielded and twisted pair cables should be used.

2.4.1.2 Signal type TTL (single-ended) and Open Collector

With the following settings in "Counter mode" (0x80n1:1D), a voltage level of nominally 2.0 V to 6.0 V with a current of 2.1 mA or higher is expected:

- 2: Encoder TTL (single ended)
- 3: Counter TTL (single ended)
- 4: Encoder open collector
- 5: Counter open collector

For TTL encoders a cut-off frequency of up to 4 million increments per second is permissible with 4-fold evaluation. This corresponds to 1 MHz.

For Open Collector encoders, a cut-off frequency of up to 400,000 increments per second is permissible with 4-fold evaluation. This corresponds to 100 kHz.

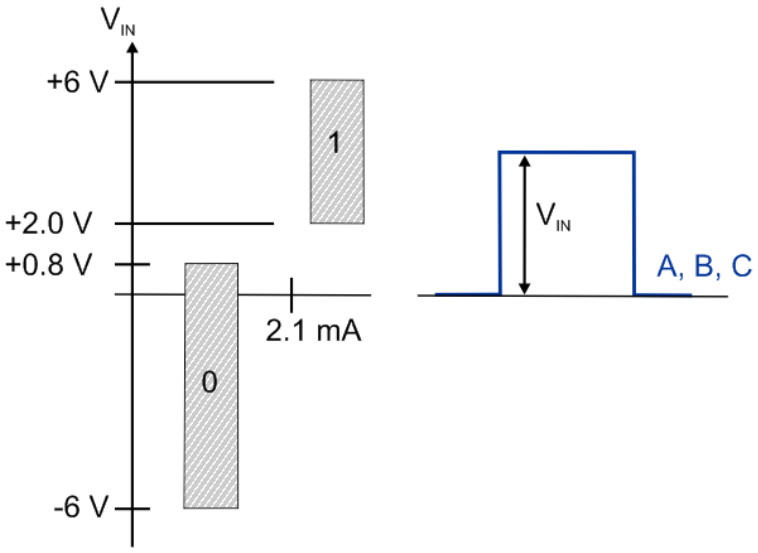


Fig. 5: TTL, Open Collector signal level (single-ended signal)

Key:

V_{IN}	Single-ended input voltage
A, B, C	Encoder signals A, B, C

NOTICE

Open circuit detection

Open circuit detection inherently does not work with single-ended lines: TTL and Open Collector encoders and counters/pulse generators.

NOTICE

Open Collector wiring

When selecting an Open Collector encoder under "Counter mode" (0x80n1:1D), the inputs A, B, C are connected to 5 V via pull-up resistors (1 kΩ).

2.4.2 Latch and Gate/Latch inputs

The terminal provides two digital 24 V_{DC} inputs. The function of these inputs is described in the respective chapter.

- Latch input [▶ 197] (Latch extern)
- Gate/latch input [▶ 198] (Latch extern 2)

Both inputs are type 3 inputs according to EN 61131-2, with a minimum pulse duration of t_{ON} > 1µs.

Digital input type 3, according to EN 61131-2	Voltage [V]	Input current [mA]
Signal voltage "0 - LOW"	-3 V ... +5 V typ.	0 mA ... 2.6 mA typ.
Signal voltage "1 - HIGH"	11 V ... 30 V typ.	typ. 2.5 mA

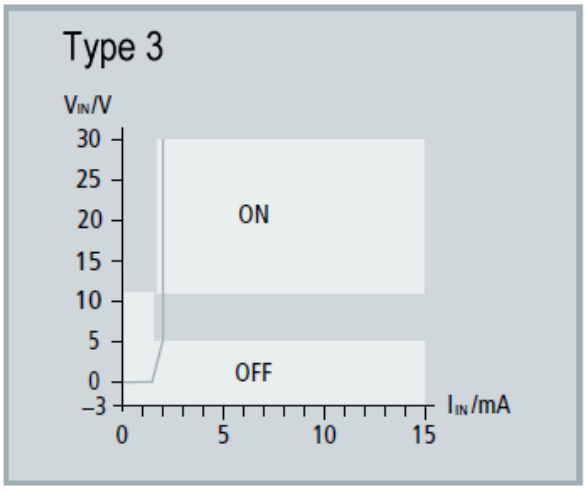


Fig. 6: Characteristic 24 V_{DC} Input type 3

NOTICE

Be aware of bouncing when using electromechanical switches and push buttons

When using electromechanical switches and push buttons, repeated closing and opening of the switch or push button can occur when the switch or push button is actuated, which is referred to as bouncing.

- If the function 0x80n0:22 "Enable continuous latch extern" or 0x80n0:23 "Enable continuous latch extern 2" is active, the stored value is overwritten several times due to the bouncing. As a result, parameter 0x60n0:12 "Latch value" or 0x60n0:22 "Latch value 2" contains the value that was saved last, not the value that was saved first.
- If the function is deactivated, only the first opening or closing of the switch or push button is detected and saved as a value in the corresponding parameter. No other transactions are taken into account.

2.4.3 Status Input

The terminal provides a Status Input. The function is described in chapter Status Input. [▶ 200]

The input is 5 V compatible.

Digital input, 5 V TTL input characteristic	Voltage [V]	Input current [mA]
Signal voltage "0 - LOW"	-6 V … + 0.8 V	typ. 5 mA
Signal voltage "1 - HIGH"	+2 V … +6 V	typ. 0 mA

NOTICE

Wiring of the Status Input

In the terminal the Status Input is internally connected to 5 V via a pull-up resistor (1 kΩ). The encoder output must actively pull the signal against GND. The resistance must be dimensioned so that it is less than 120 Ω.

External power supply is not recommended. If an external supply is used, the maximum permitted voltage is 5 V against GND.

2.4.4 Encoder operating voltage (supply voltage)

The encoder supply is generated internally from the 24 V of the power contacts. The encoder supply can be set in index 0x8001:17 [▶ 221] "Supply voltage". An operating voltage of 5 V_{DC} is preset. Voltage values of 5 V_{DC}, 12 V_{DC} and 24 V_{DC} can be selected.

The following tolerances apply

Voltage range	Tolerance
5 V_{DC}	+/- 5% (4.75 V … 5.25 V)
12 V_{DC}	+/- 10% (10.8 V … 13.2 V)
24 V_{DC}	-15 % to +20 % (20.4 V … 28.8 V)

NOTICE

Setting the encoder supply voltage

- Before switching to a higher voltage, make sure that the connected encoder supports the selected voltage range!
- Note the output power limitation (5 V_{DC} to 0.3 A, 12 V_{DC} to 0.15 A, and 24 V_{DC} to 0.3 A).
- To write to 0x8001:17 "Supply voltage" you have to set the value 0x72657375 (ASCII: "user") in index 0xF008 [▶ 243] "Code word".

2.4.5 Outputs

Two 24 V_{DC} - digital outputs are available, which can be parameterized and switched independently of each other.

The following options are available for parameterization:

- Set switching characteristics [▶ 178]:
  Push-Pull [▶ 26] or Push-Tristate [▶ 26] mode
- Set state in case of bus error [▶ 179]:

Furthermore, the following options are available for switching the outputs:

manual	Via PLC variable [▶ 181]	
Automatic switching function [▶ 182]	when specified counter values [▶ 184] are reached • 1 … 8 threshold values • "Output state 1" ... "Output state 8" freely selectable (TRUE, FALSE) • Timestamp function [▶ 172] available • Switching possible for a defined time [▶ 184] • Realization of a digital cam controller [▶ 183] possible	when reaching predefined frequency and velocity values [▶ 189] • Max. 2 threshold values • "Output state" fixed ("Output state 1": TRUE, "Output state 2": FALSE) • Timestamp function [▶ 172] available

The parameterization of the outputs is described in the chapter Output function [▶ 176].

Each output has a separate short circuit/overload diagnosis (see chapter Short circuit/overload diagnosis of the outputs [▶ 217]).

2.4.5.1 Push-pull mode

The output circuit is optimized for fast signal output, even at higher currents. To this end, each output has a push-pull output stage that actively switches to 24 V and 0 V. The pull stage reduces the output to 1.2 V. If a 0 V level is explicitly required, this must be implemented via an external load.
The push and pull stages are overcurrent and short circuit protected. The edge steepness of the output is 1 µs/24 V (typ.). The following table shows the output characteristics:

Technical data	Push mode	Pull mode
Switching times	T_{ON}: 3 µs typ.	T_{OFF}: 4 µs typ.
Output current per channel	0.5 A	0.15 A
Load type	ohmic, inductive, lamp load	

The power supply of the terminal should be dimensioned according to the power requirements of the connected actuators. The power supply lines, power supply unit and actuator lines should be suitably short or generous respectively.

NOTICE

Switching behavior of the output when switching with a PLC variable

If the output is switched via a PLC variable, the delays caused by the PLC cycle time must also be taken into account in addition to the T_{ON}/T_{OFF}.

2.4.5.2 Push-Tristate mode

In Push Tristate mode each output can be switched individually to 24 V and high-resistance.

The edge steepness of the output is 1 µs/24 V (typ.). The high-resistance tristate state ensures that the respective output behaves as if it was not connected and does not affect the outputs of other outputs/devices connected to it. The associated output takes on the same output voltage as the other active devices. The following table shows the output characteristics:

Technical data	Push mode	Tristate mode
Switching times	T_{ON}: 3 µs typ.	T_{OFF}: 7 µs typ.
Output current per channel	0.5 A	/
Load type	ohmic, inductive, lamp load	

2.5 Start

For commissioning:

- mount the EL5131 as described in the chapter Mounting and wiring [▶ 40]
- configure the EL5131 in TwinCAT as described in the chapter Commissioning [▶ 63].
- Parameterize the EL5131 as described in chapter EL5131 - Commissioning [▶ 135].

3 EL51xx series overview

Technical data		EL5102	EL5112		EL5122	EL5131
		2 x A, B, C	1 x A, B, C	2 x A, B	2 x A, B	1 x A, B, C
Number of channels		2	1	2	2	1
Encoder type, incremental	Differential RS422	Yes	Yes	Yes	No	Yes
	Single-ended TTL	Yes	Yes	Yes	Yes	Yes
	OpenCollector	Yes	Yes	Yes	Yes	Yes
	Counter / pulse generator	Yes	Yes	Yes	Yes	Yes
Number of digital inputs per channel		2	2	1	1	2
Number of digital outputs per channel		No	No	No	No	2
Encoder operating voltage 5 V_{DC}, 12 V_{DC}, 24 V_{DC} switchable		Yes	Yes	Yes	Yes	Yes
Encoder output current per channel		0.3 A	0.3 A	Sum current 0.3 A	Sum current 0.3 A	0.3 A
Cut-off frequency		20 million increments/s, corresponds to 5 MHz	20 million increments/s, corresponds to 5 MHz	20 million increments/s, corresponds to 5 MHz	4 million increments/s, corresponds to 1 MHz	20 million increments/s, corresponds to 5 MHz

Functions		EL5102	EL5112		EL5122	EL5131
		2 x A, B, C	1 x A, B, C	2 x A, B	2 x A, B	1 x A, B, C
Reset counter value via	Zero pulse C	Yes	Yes	No	No	Yes
	Latch input	Yes	Yes	Yes	Yes	Yes
Set counter value via	PLC variable	Yes	Yes	Yes	Yes	Yes
	Zero pulse C	Yes	Yes	No	No	Yes
	Latch input	Yes	Yes	Yes	No	Yes
	Gate/Latch input	No	No	Yes	Yes	No
Save counter value via	Zero pulse C	Yes	Yes	No	No	Yes
	Latch input	Yes	Yes	Yes	No	Yes
	Gate/Latch input	Yes	Yes	No	Yes	Yes
Lock counter value via	PLC variable	Yes	Yes	Yes	Yes	Yes
	Gate/Latch input	Yes	Yes	Yes	Yes	Yes
Switching at comparison values (Counter / Frequency / Period value)		No	No	No	No	Yes
Detect counting direction		Yes	Yes	No	No	Yes
Detect reversion of rotation		Yes	Yes	No	No	Yes
Frequency measurement		Yes	Yes	Yes	Yes	Yes
Period value measurement		Yes	Yes	Yes	Yes	Yes
Velocity, speed calculation		Yes	Yes	Yes	Yes	Yes
Duty cycle evaluation		Yes	Yes	No	No	Yes
Microincrements		Yes	Yes	No	No	Yes
Timestamp function		Yes	Yes	No	No	Yes
Adjustable interference pulse filters		Yes	Yes	Yes	Yes	Yes
Plausibility check		Yes	Yes	Yes	Yes	Yes

4 Incremental encoder basics

Incremental encoders divide a 360° rotation of the encoder axis into individual steps (increments) and mark a full revolution by means of a special mark (zero pulse). An RS422 encoder transmits the signal symmetrically as a differential line pair. TTL and Open Collector encoders use single signal lines (single-ended).

The module evaluates the 90° phase-shifted square wave signals of an incremental encoder on tracks A and B. The zero pulse is captured on track C. With a differential connection, the inverted signals ($\overline{A}$, $\overline{B}$, $\overline{C}$) are also recorded.

These signals are converted by means of the quadrature decoder and the 32-bit counter into a position value with optional quadruple, double or single evaluation. The digital inputs enable latch, reset and set functionalities and thus exact and speed-independent referencing and storage of the counter value.

Encoder type		Incremental signals
RS422 encoder	with zero pulse	A, $\overline{A}$, B, $\overline{B}$, C, $\overline{C}$
RS422 encoder	without zero pulse	A, $\overline{A}$, B, $\overline{B}$
RS422 counter or pulse generator	with zero pulse	A, $\overline{A}$, C, $\overline{C}$; counting direction specification via track B (B, $\overline{B}$)
RS422 counter or pulse generator	without zero pulse	A, $\overline{A}$; counting direction specification via track B (B, $\overline{B}$)
TTL, Open Collector encoder	with zero pulse	A, B, C
TTL, Open Collector encoder	without zero pulse	A, B
TTL, Open Collector counter or pulse generator	with zero pulse	A, C; counting direction specification via B
TTL, Open Collector counter or pulse generator	without zero pulse	A, counting direction specification via B

The phase position between the signals at track A and track B determines the counting direction.
Forward (cw): signal at track A is 90° leading with respect to track B
Backward (ccw): signal at track A is 90° lagging compared to track B.

With 1-fold evaluation the rising edges on track A are counted.
With 2-fold evaluation, the rising and falling edges on track A are counted.
With 4-fold evaluation, the rising and falling edges on track A and track B are counted.

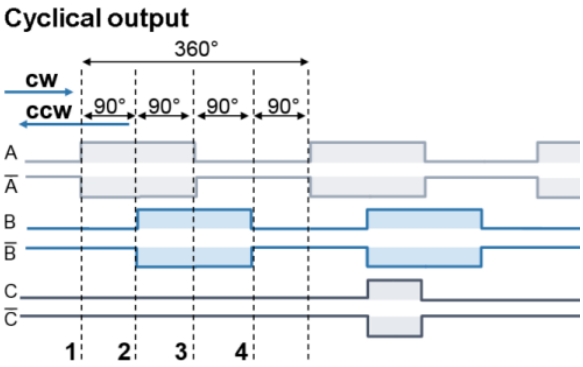


Fig. 7: Incremental signals

Absolute value encoders provide an absolute position value directly after switch-on, which is unambiguous over the entire travel path. With incremental encoders, a homing must be carried out after switching on in order to be able to determine a unique position.
Homing can be carried out, for example, with the aid of referencing cams or via the zero pulse of the encoder.

NOTICE

Differential and single-ended connection

The RS422 signal transmits a differential voltage, which makes the signal less sensitive to interference compared to a single-ended signal.

- If the encoder signal is to be transmitted over longer distances or at higher frequencies, an encoder with RS422 signals is recommended.
- Shielded and twisted pair cables should be used.

5 Basics communication

5.1 EtherCAT basics

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

5.2 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.

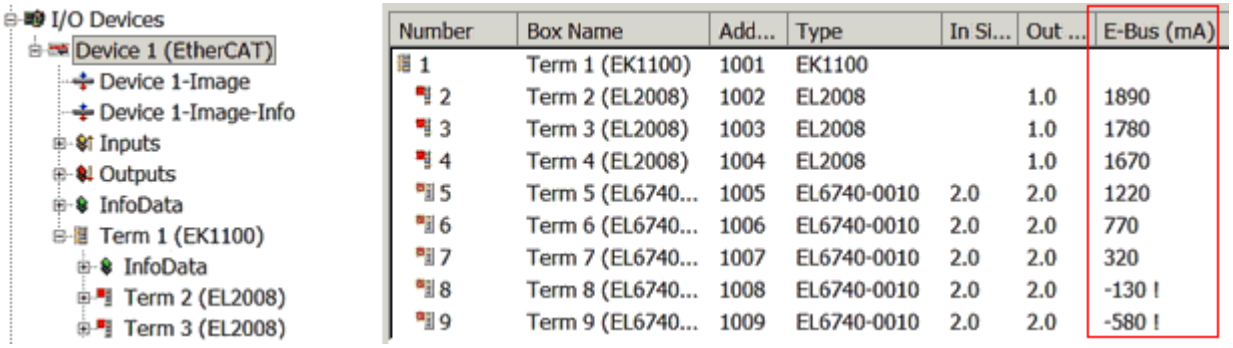


Fig. 8: System manager current calculation

NOTICE

Malfunction possible!

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

5.3 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:

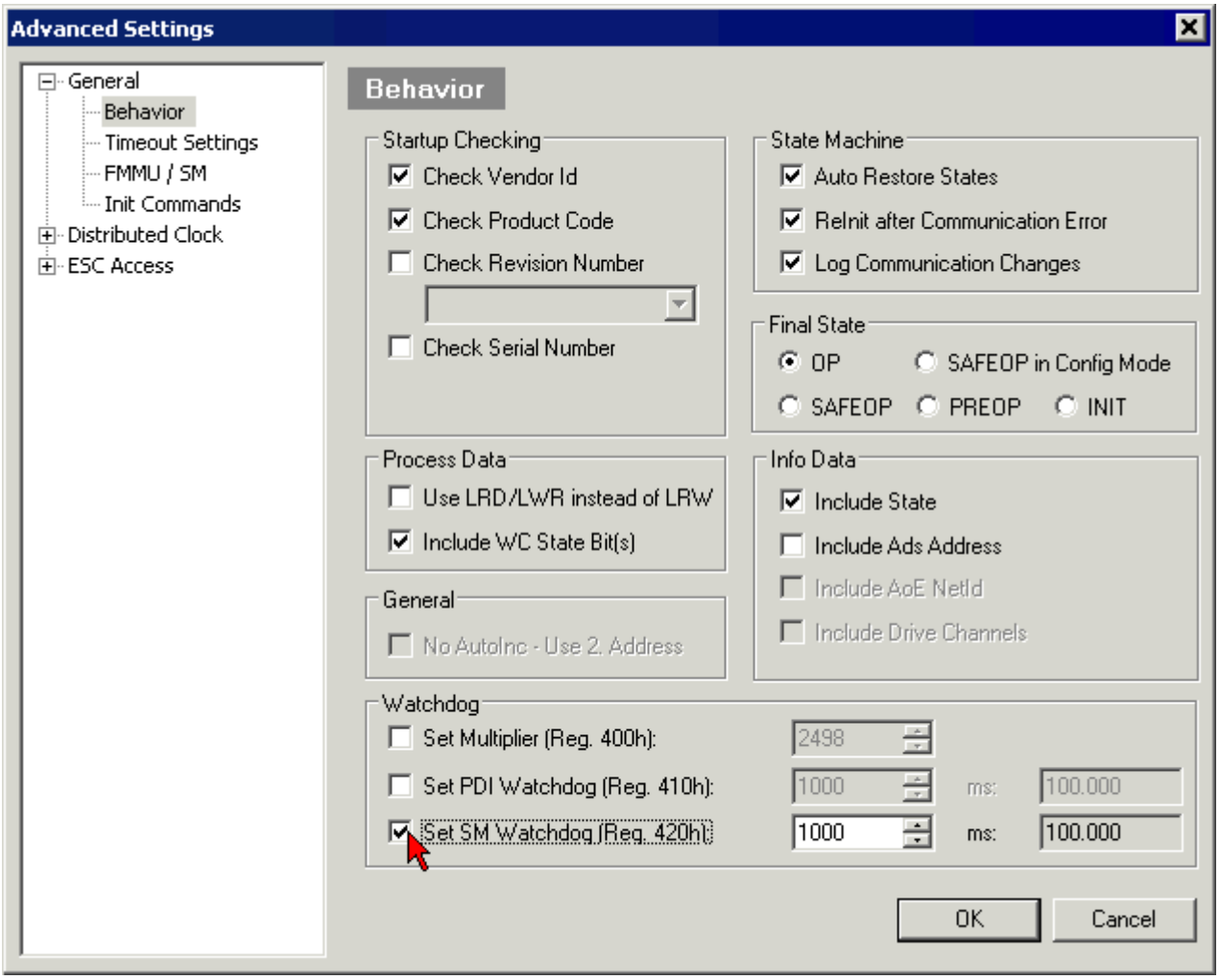


Fig. 9: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

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 (SyncManager Watchdog)

The SyncManager 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 SyncManager 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.

5.4 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.

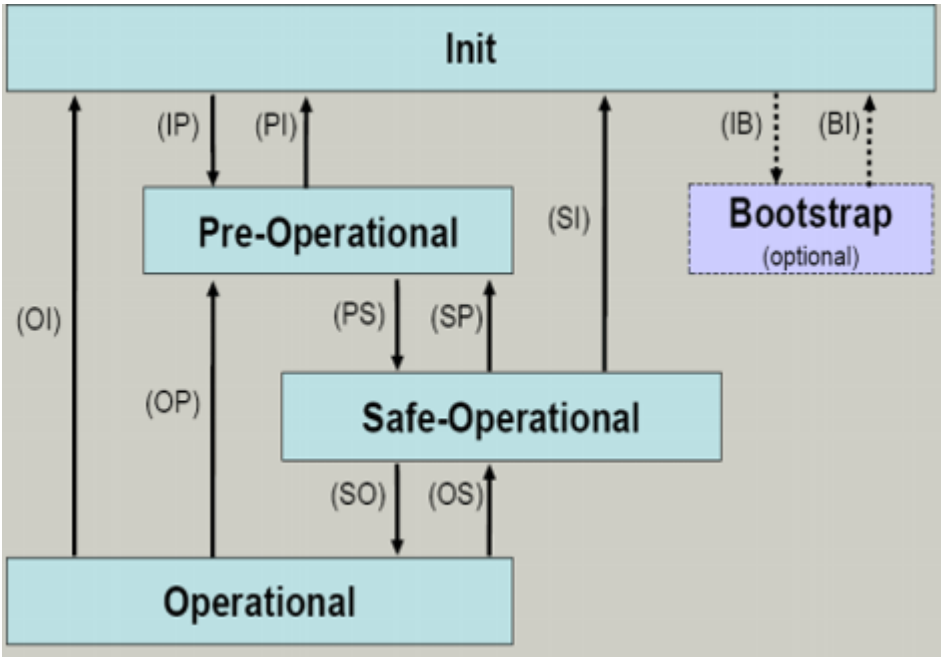


Fig. 10: States of the EtherCAT State Machine

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.

5.5 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:

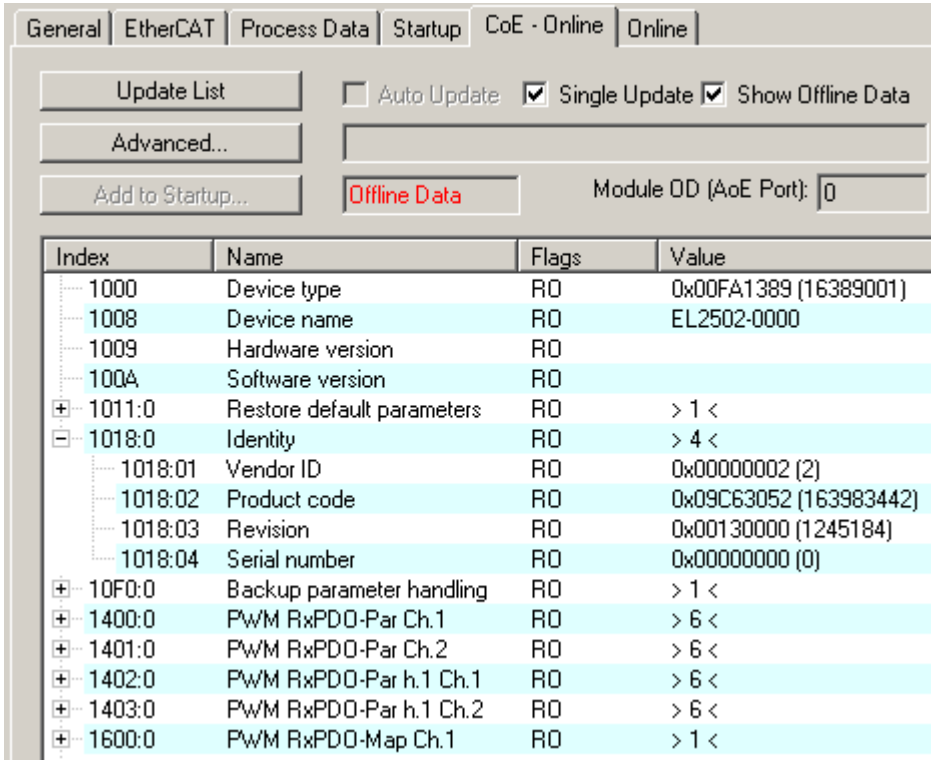


Fig. 11: “CoE Online” tab

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.

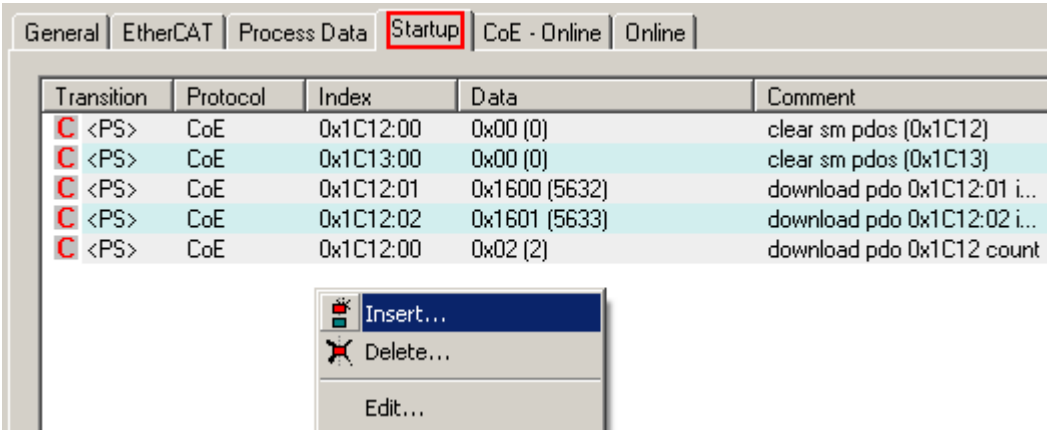


Fig. 12: Startup list in the TwinCAT System Manager

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.

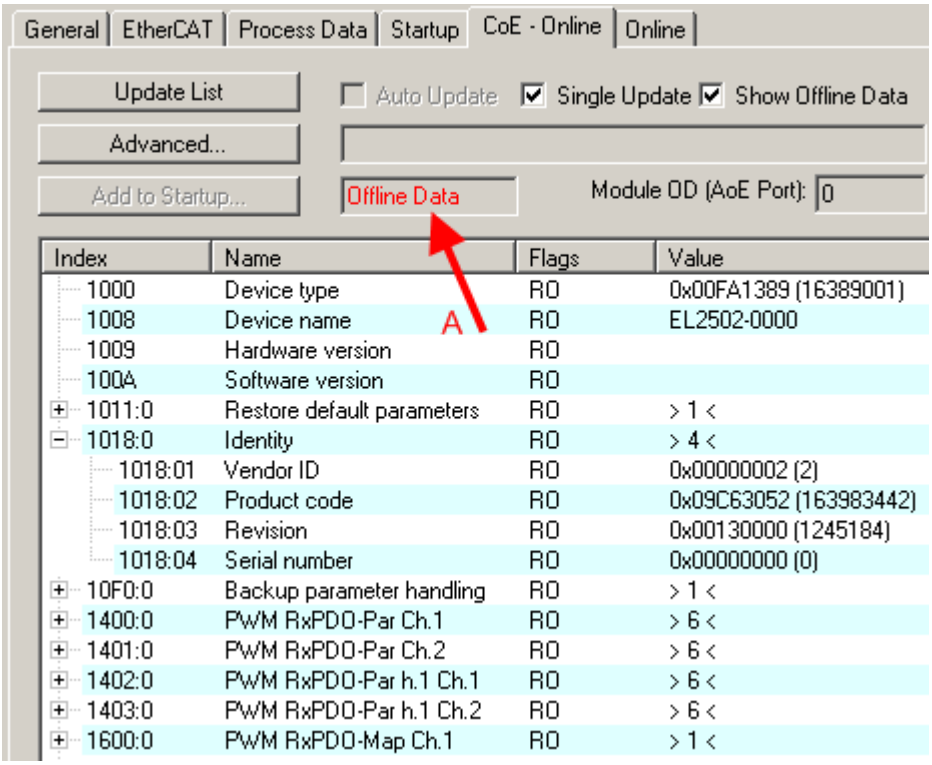


Fig. 13: Offline list

- 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.

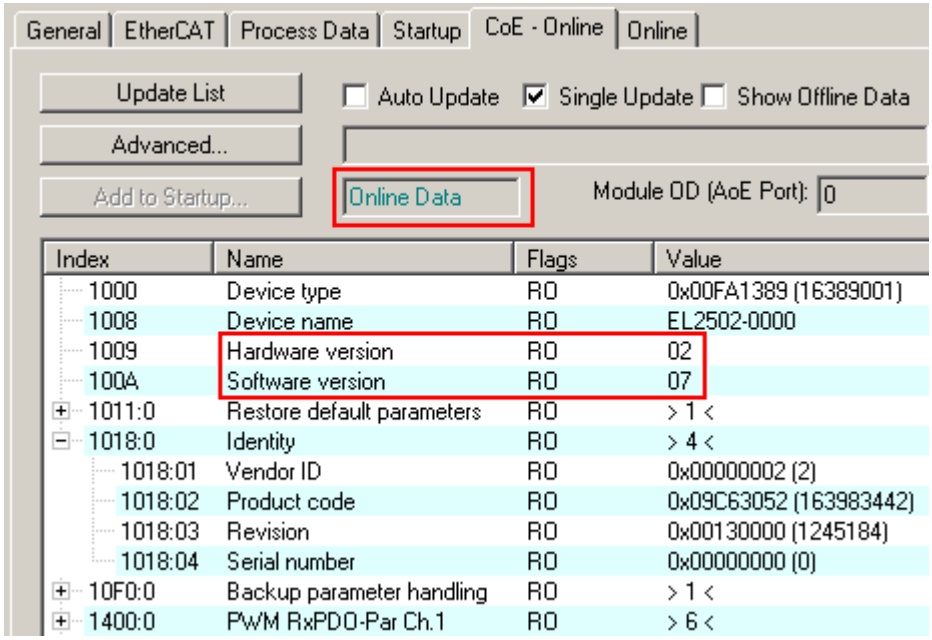


Fig. 14: Online list

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.

5.6 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.

6 Mounting and wiring

6.1 Instructions for ESD protection

NOTICE

Destruction of the devices by electrostatic discharge possible!

The devices contain components at risk from electrostatic discharge caused by improper handling.

- When handling the components, ensure that there is no electrostatic discharge; also avoid touching the spring contacts directly (see illustration).
- Contact with highly insulating materials (synthetic fibers, plastic films, etc.) should be avoided when handling components at the same time.
- When handling the components, ensure that the environment (workplace, packaging and persons) is properly earthed.
- Each bus station must be terminated on the right-hand side with the EL9011 or EL9012 end cap to ensure the degree of protection and ESD protection.

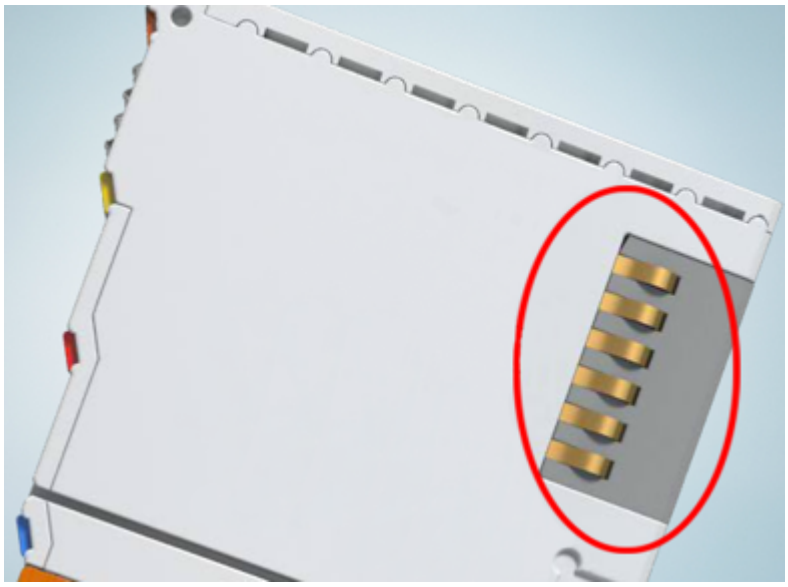


Fig. 15: Spring contacts of the Beckhoff I/O components

6.2 Installation on mounting rails

⚠ 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!

The Bus Terminal system and is designed for mounting in a control cabinet or terminal box.

Assembly

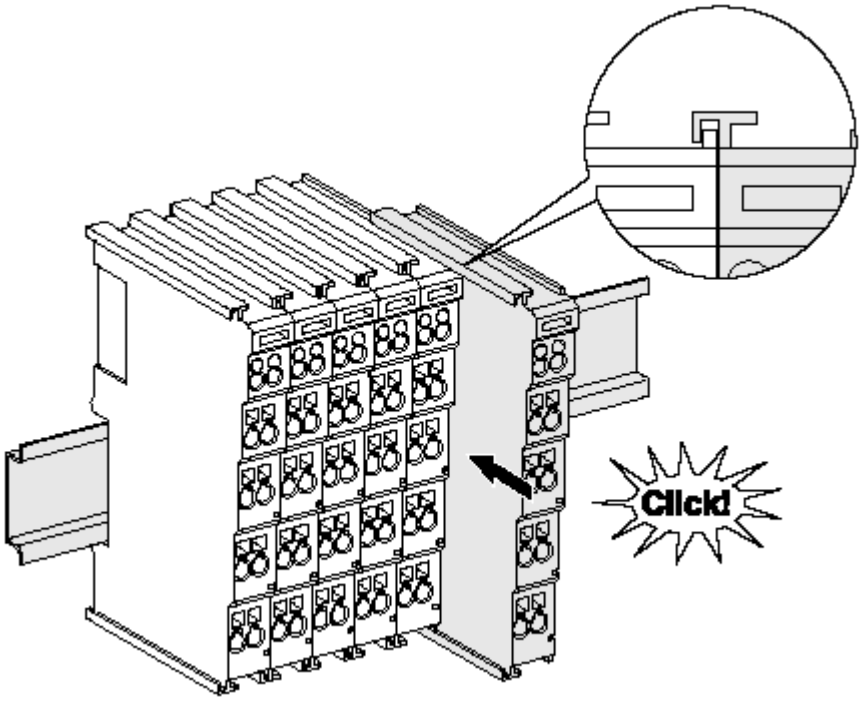


Fig. 16: Attaching on mounting rail

The bus coupler and bus terminals are attached to commercially available 35 mm mounting rails (DIN rails 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. Join the components with tongue and groove and push the terminals against the mounting rail, until the lock clicks onto the mounting rail.
   If the terminals are clipped onto the mounting rail first and then pushed together without tongue and groove, the connection will not be operational! When correctly assembled, no significant gap should be visible between the housings.

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 mounting rails with a height of 7.5 mm under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

NOTICE

Ground the mounting rail!

Ensure that the mounting rail is sufficiently earthed.

Connections within a bus terminal block

The electric connections between the Bus Coupler and the Bus Terminals are automatically realized by joining the components:

- The six spring contacts of the E-Bus/K-Bus deal with the transfer of the data 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 terminals points on the Bus Coupler (up to 24 V) or for higher voltages via power feed terminals.

Power Contacts

During the design of a bus terminal block, the pin assignment of the individual Bus Terminals must be taken account of, since some types (e.g. analog Bus Terminals or digital 4-channel Bus Terminals) do not or not fully loop through the power contacts. Power Feed Terminals (EL91xx, EL92xx or KL91xx, KL92xx) interrupt the power contacts and thus represent the start of a new supply rail.

Power contact ⏚

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

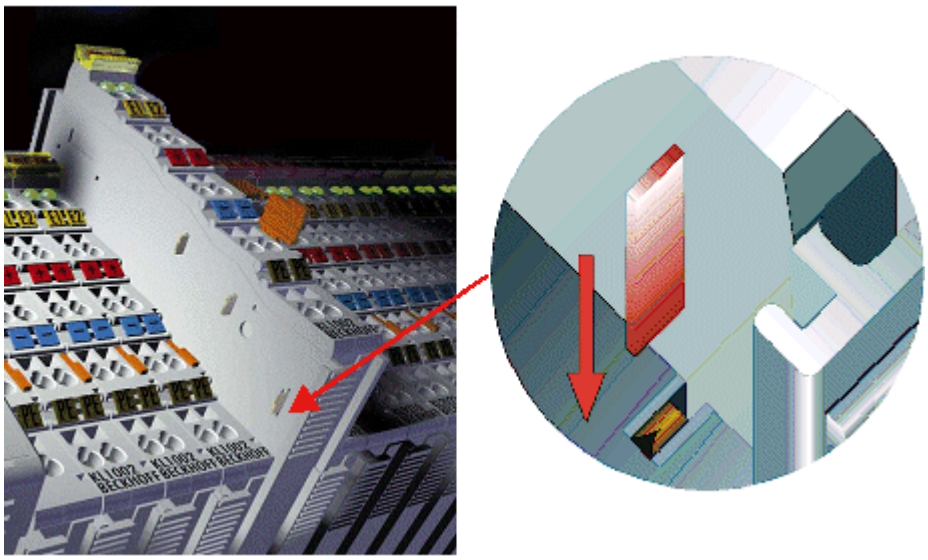


Fig. 17: Power contact on left side

⚠ WARNING

Risk of electric shock!

The power contact labeled ⏚ must not be used for other potentials!

NOTICE

Possible damage of the device

Note that, for reasons of electromagnetic compatibility, the earthing contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage on the terminal (e.g. disruptive discharge to the earthing line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the earthing supply line at the Bus Coupler or the Power Feed Terminal! In order to decouple further feed points for testing, these Power Feed Terminals can be released and pulled at least 10 mm from the group of terminals.

Disassembly

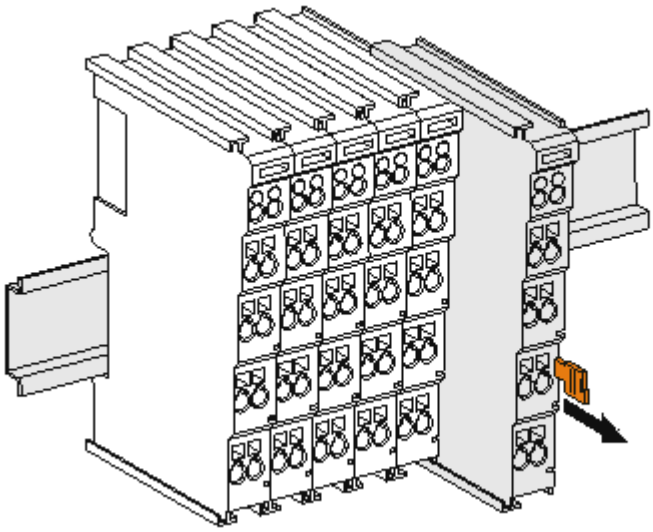


Fig. 18: Disassembling of terminal

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

1. Pull the terminal by its orange-colored lugs approximately 1 cm away from the mounting rail. In doing so for this terminal the mounting rail lock is released automatically and you can pull the terminal out of the bus terminal block easily without excessive force.
2. Grasp the released terminal with thumb and index finger simultaneous at the upper and lower grooved housing surfaces and pull the terminal out of the bus terminal block.

6.3 UL notice

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

⚠ CAUTION	
c UL us	**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	
c UL us	**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:



6.4 Connection

6.4.1 Connection system

⚠ 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!

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)

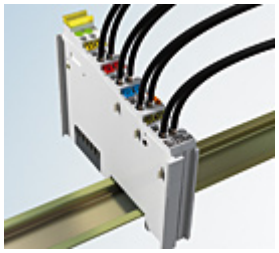


Fig. 19: Standard wiring

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

Pluggable wiring (ESxxxx / KSxxxx)

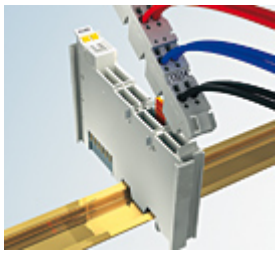


Fig. 20: Pluggable wiring

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. 21: 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 [▶ 48]!

6.4.2 Wiring

⚠ 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!

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

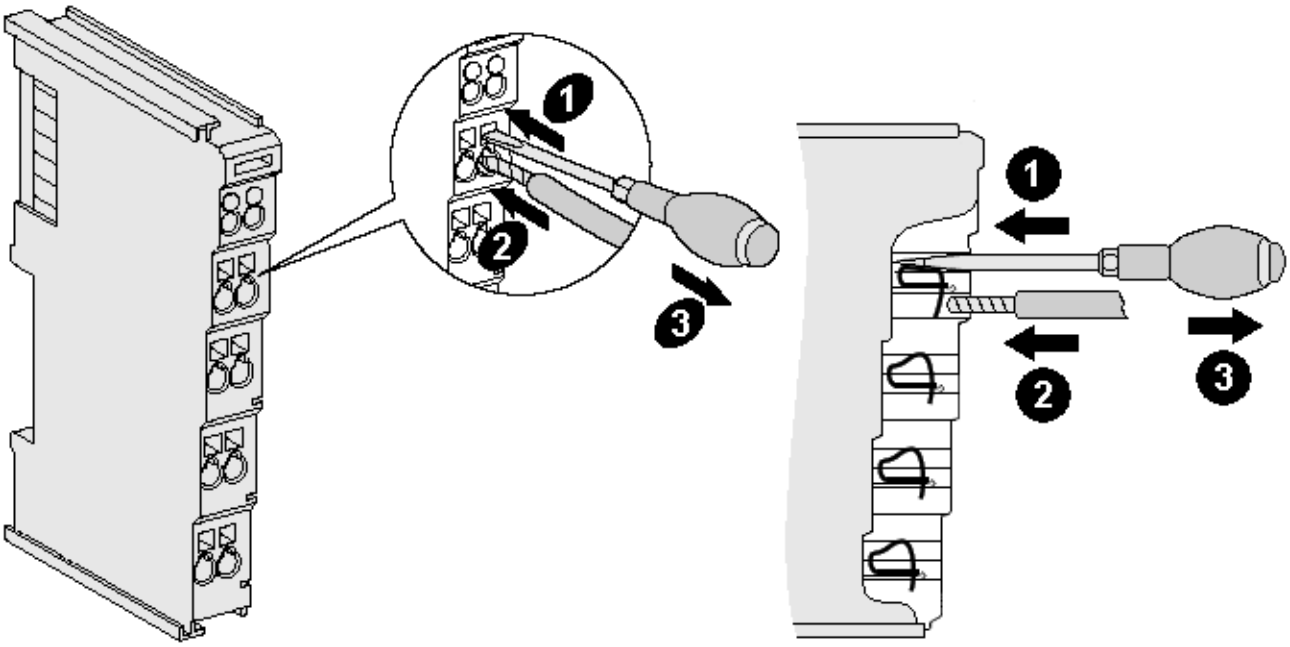


Fig. 22: Connecting a cable on a terminal point

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 [▶ 46]) 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 [▶ 46])
Wire stripping length	8 ... 9 mm

6.4.3 Shielding

Shielding

Encoder, analog sensors and actuators should always be connected with shielded, twisted paired wires.

6.5 Note - 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.

6.6 Installation position for operation with or without fan

NOTICE

Constraints regarding installation position and operating temperature range

When installing the terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

Prescribed installation position for operation without fan

The prescribed installation position requires the mounting rail to be installed horizontally and the connection surfaces of the EL/KL terminals to face forward (see Fig. "*Recommended distances of installation position for operating without fan*").

The terminals are ventilated from below, which enables optimum cooling of the electronics through convection.

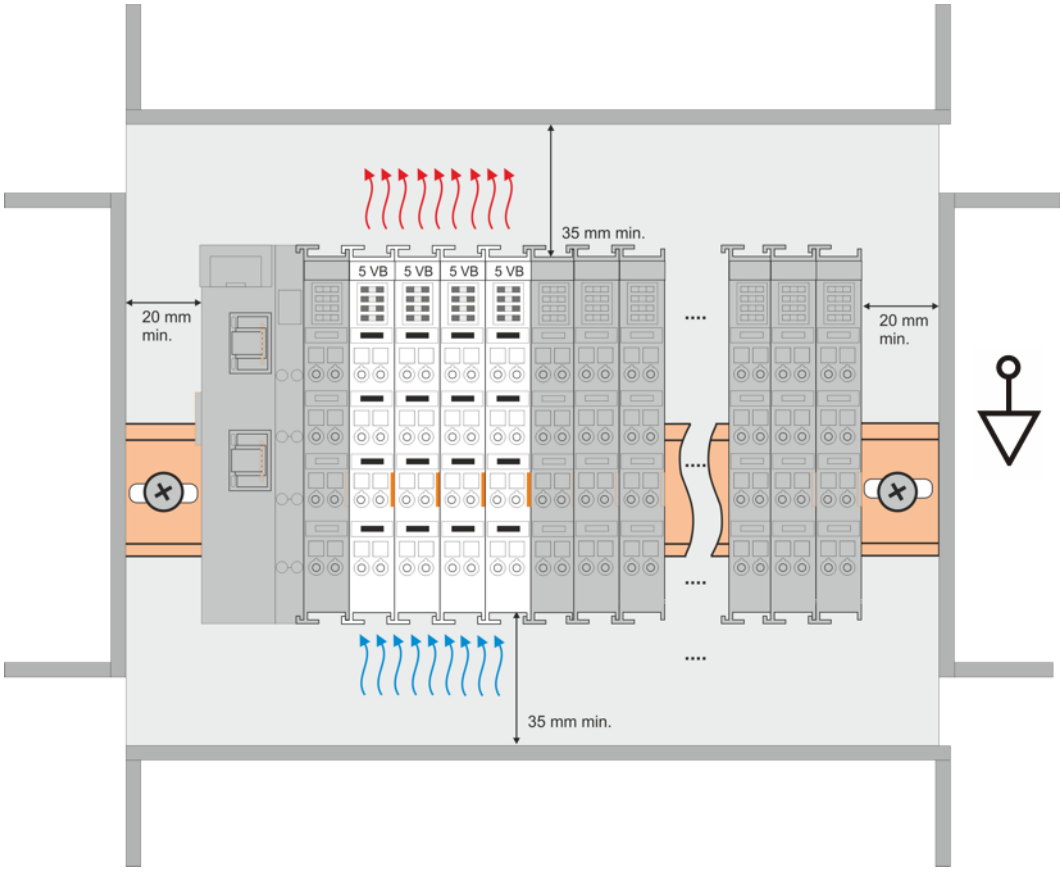


Fig. 23: Recommended distances of installation position for operating without fan

Compliance with the distances shown in Fig. "*Recommended distances of installation position for operating without fan*" is recommended.

For further information regarding the operation without fan refer to the Technical Data of the terminal.

Standard installation position for operation with fan

The standard installation position for operation with fan requires the mounting rail to be installed horizontally and the connection surfaces of the EL/KL terminals to face forward (see Fig. *Recommended distances for installation position for operation with fan*).
The terminals are ventilated fan supported (e.g. with ZB8610 fan cartridge) from below.

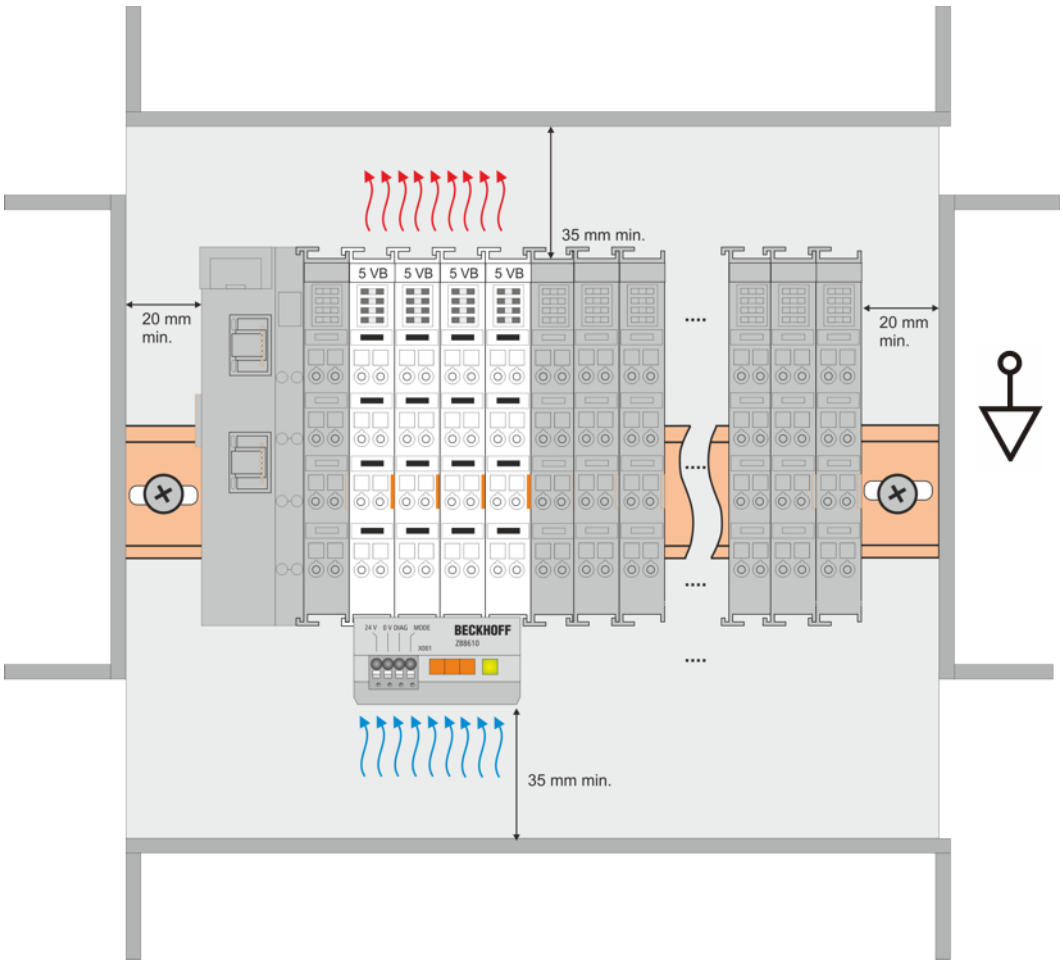


Fig. 24: Recommended distances for installation position for operation with fan

Other installation positions

Due to the enforced effect of the fan on the ventilation of the terminals, other installation positions (see Fig. "*Other installation positions, example 1 + 2*") may be permitted where appropriate.

See corresponding notes in the Technical Data of the terminal.

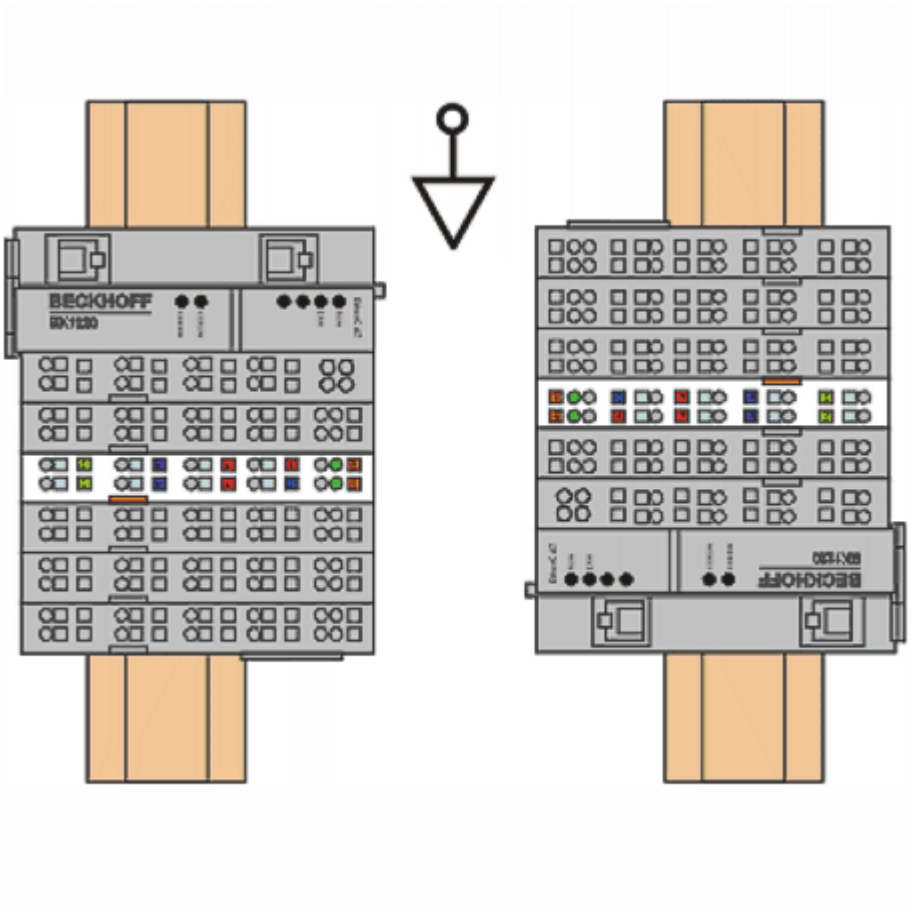


Fig. 25: Other installation positions, example 1

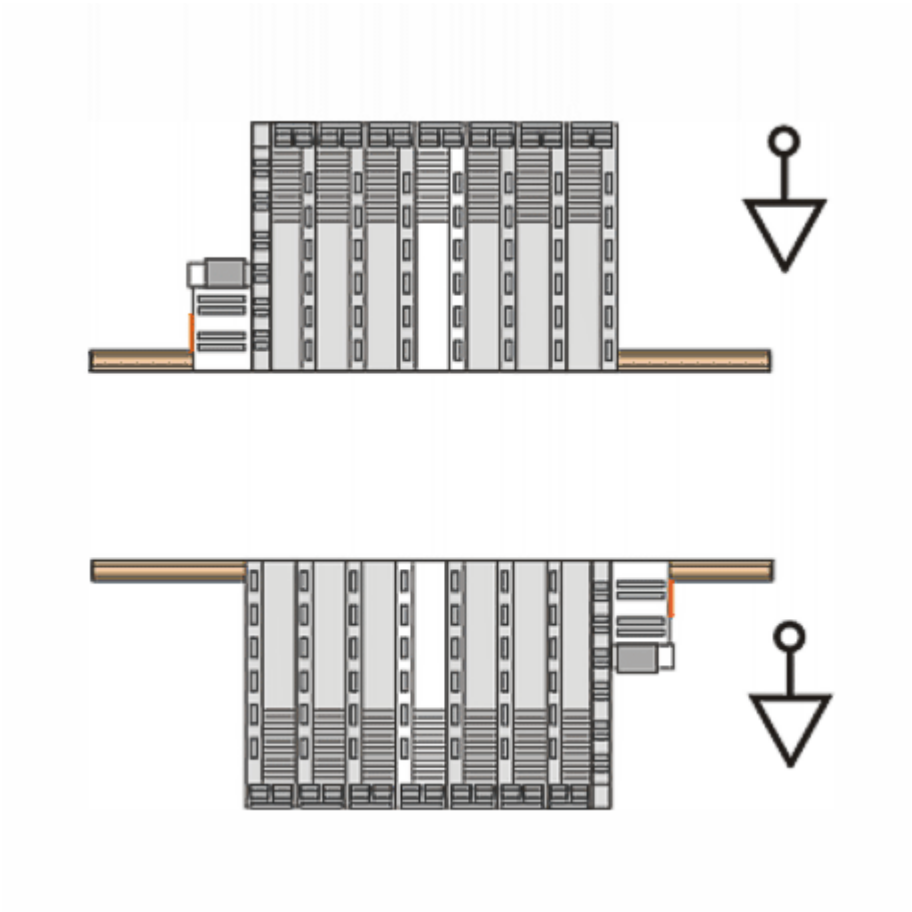


Fig. 26: Other installation positions, example 2

6.7 Positioning of passive Terminals

Hint for positioning of passive terminals in the bus terminal block

EtherCAT Terminals (ELxxxx / ESxxxx), which do not take an active part in data transfer within the bus terminal block are so called passive terminals. The passive terminals have no current consumption out of the E-Bus.
To ensure an optimal data transfer, you must not directly string together more than two passive terminals!

Examples for positioning of passive terminals (highlighted)

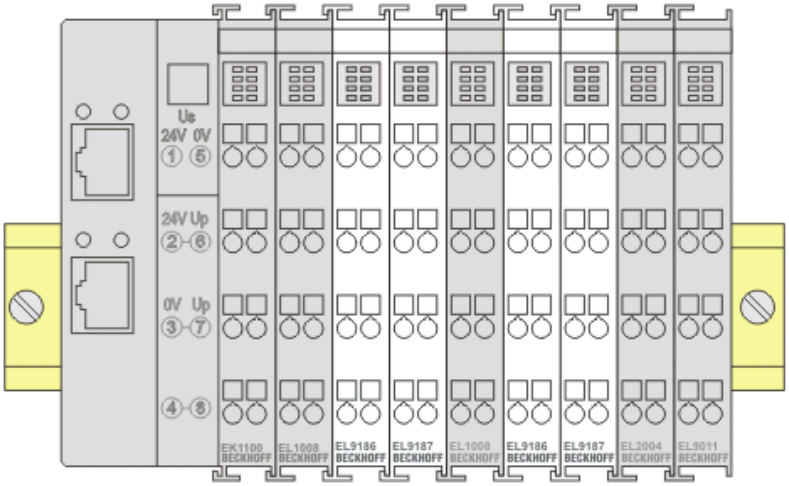


Fig. 27: Correct positioning

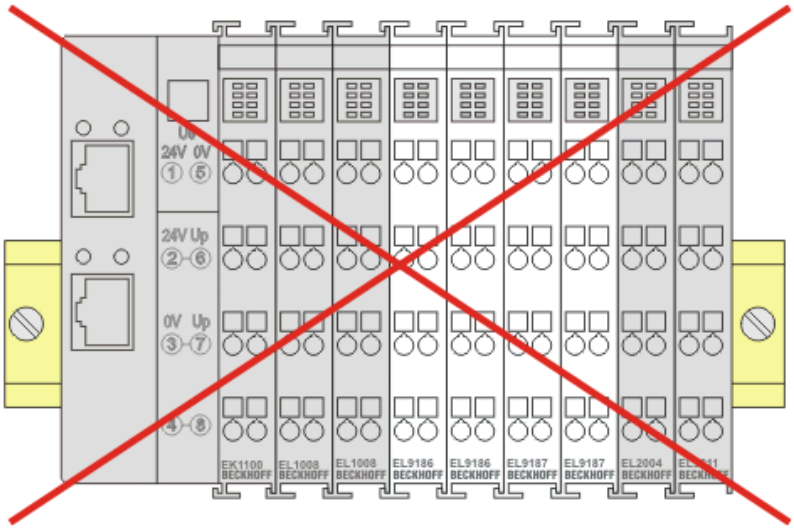


Fig. 28: Incorrect positioning

6.8 EL5131 – connection

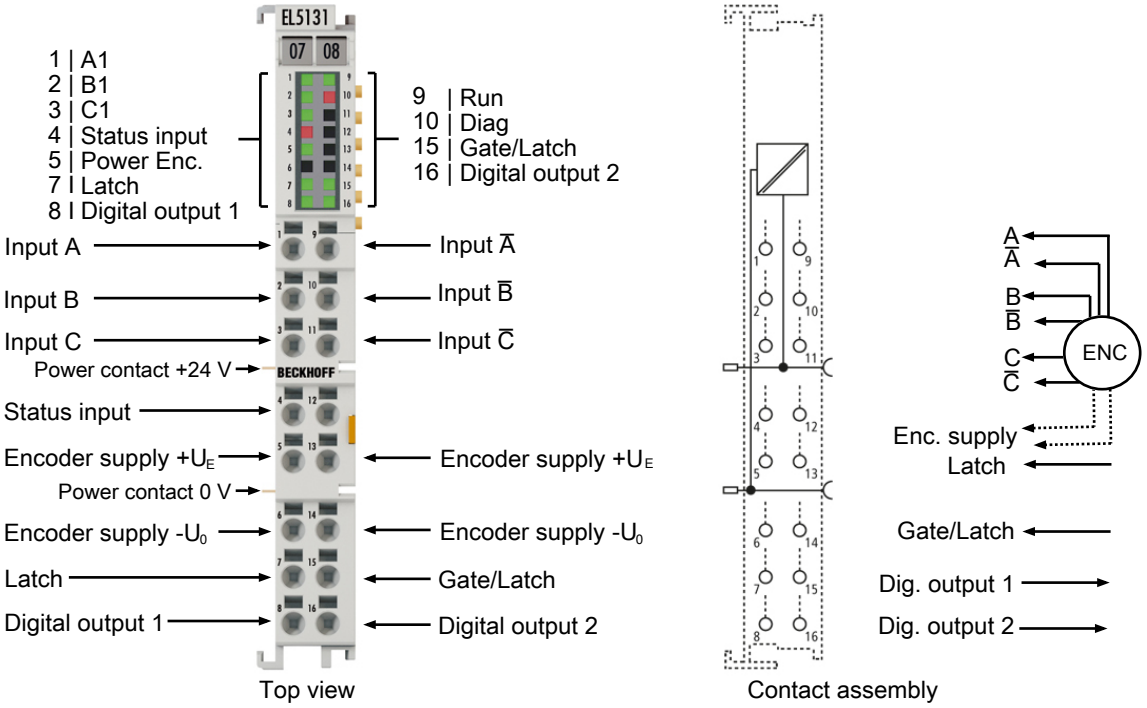


Fig. 29: EL5131 – connection

Terminal point	No.	Comment
A	1	Encoder input A
B	2	Encoder input B
C	3	Encoder input C
Status Input	4	Status input The Status Input is internally connected to 5 V via a pull-up resistor. The encoder fault signal output must actively pull the signal against GND. The external supply is not recommended. If an external supply is used, the maximum permitted voltage is 5 V against GND.
+ Ue	5	Encoder supply (5 V default, parameterizable 12 V, 24 V)
-Uo	6	0 V encoder supply
Latch	7	Latch extern input
Digital output 1	8	Digital output 1
Ā	9	Encoder input Ā
B̄	10	Encoder input B̄
C̄	11	Encoder input C̄
n.c.	12	Do not connect the terminal point
+ Ue	13	Encoder supply (5 V default, parameterizable 12 V, 24 V)
-Uo	14	0 V encoder supply
Gate/latch	15	The Gate input can also be used as Latch extern 2 input
Digital output 2	16	Digital output 2

NOTICE

Setting the encoder supply voltage

- Before switching to a higher voltage, make sure that the connected encoder supports the selected voltage range!
- Note the output power limitation (5 V_{DC} to 0.3 A, 12 V_{DC} to 0.15 A, and 24 V_{DC} to 0.3 A).
- To write to 0x8001:17 "Supply voltage" you have to set the value 0x72657375 (ASCII: "user") in index 0xF008 [▶ 243] "Code word".

6.8.1 RS422 mode

NOTICE

Differential and single-ended connection

The RS422 signal transmits a differential voltage, which makes the signal less sensitive to interference compared to a single-ended signal.

- If the encoder signal is to be transmitted over longer distances or at higher frequencies, an encoder with RS422 signals is recommended.
- Shielded and twisted pair cables should be used.

Connection of RS422 encoders with or without zero pulse

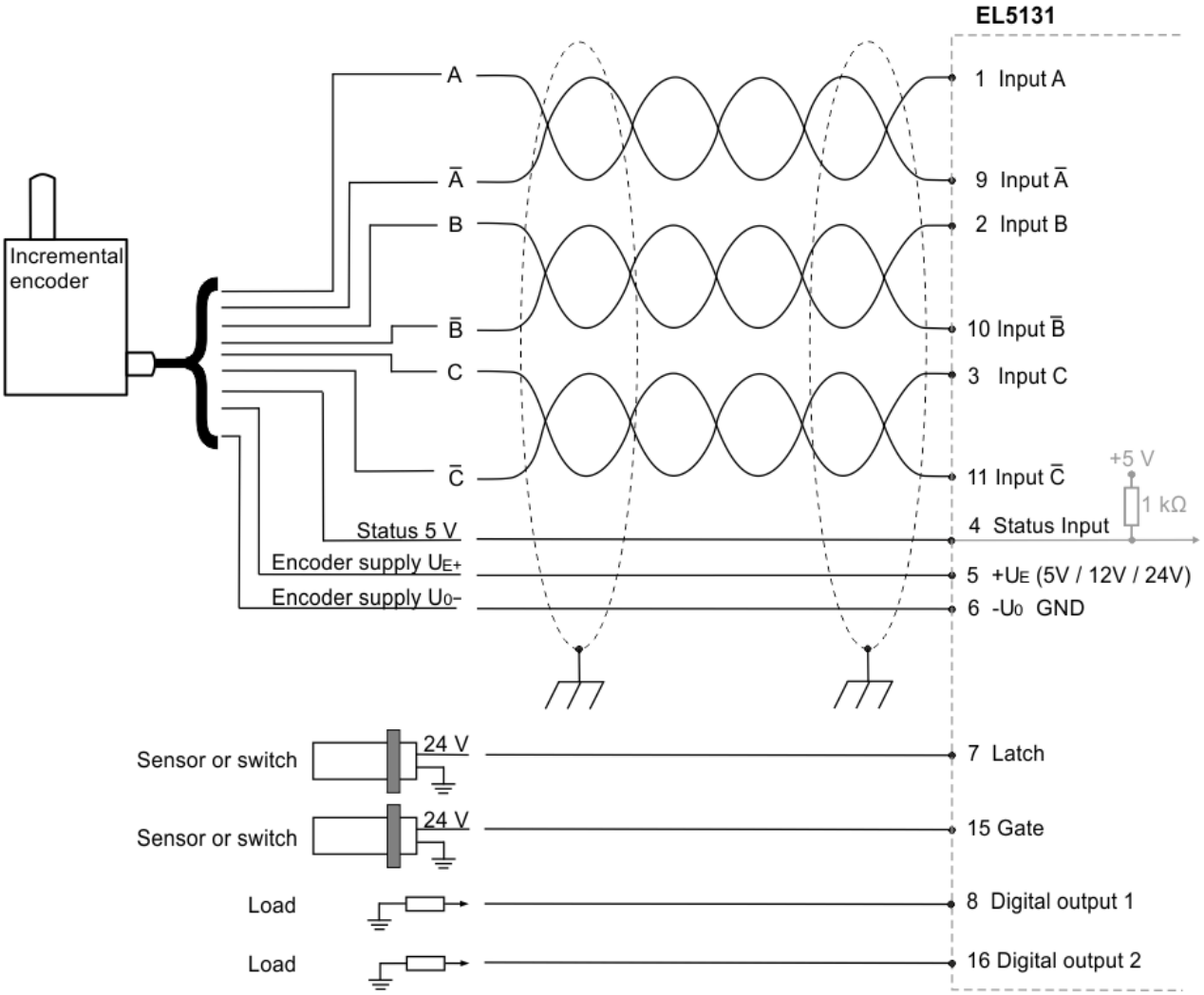


Fig. 30: Connection encoder RS422 mode with zero pulse

Connection instructions

- When using an encoder without zero pulse C, the terminal points for the C track are not connected.

Connection of RS422 counters / pulse generators with or without zero pulse

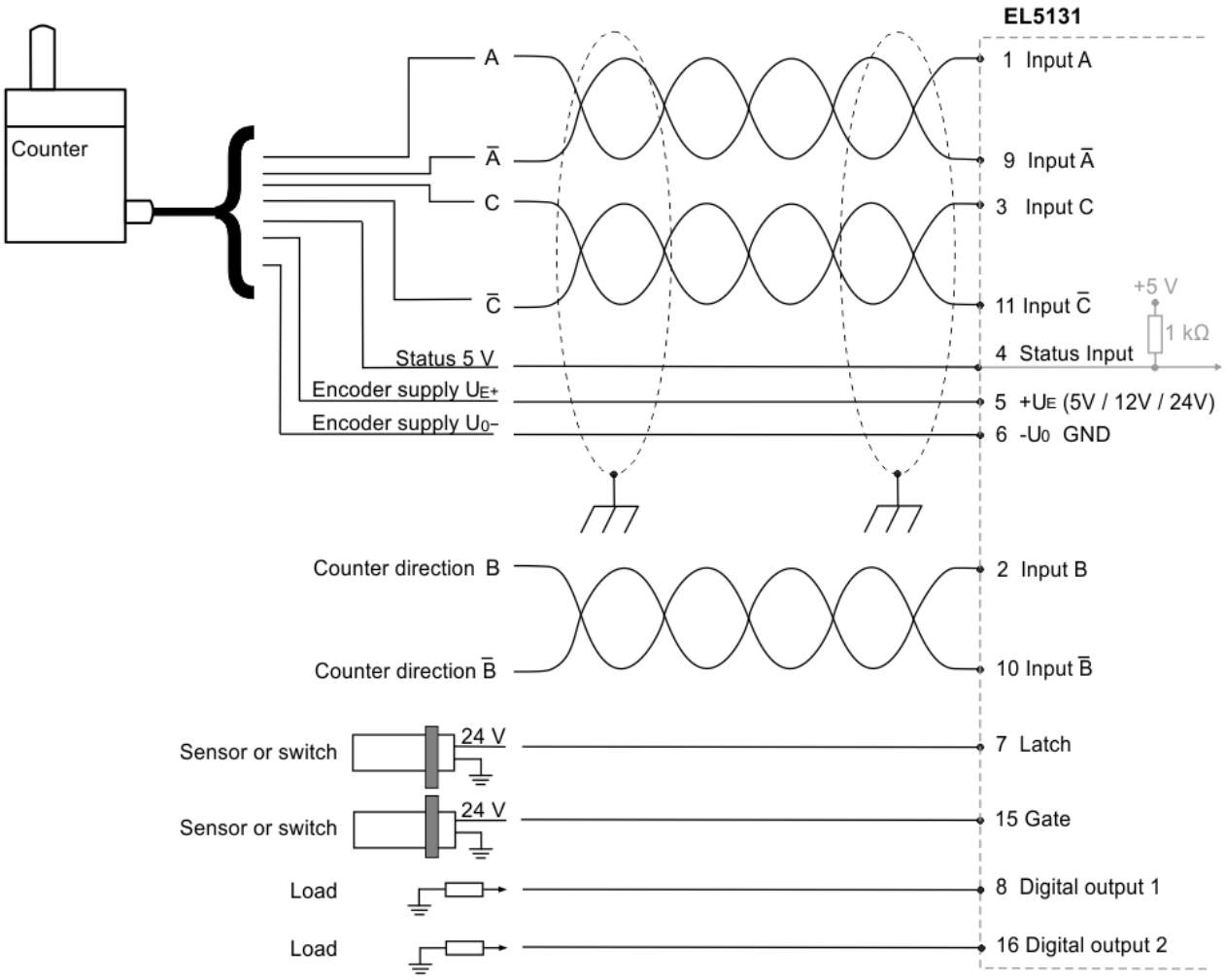


Fig. 31: Connection counter / pulse generator RS422 mode with zero pulse

Connection instructions

- When a counter/pulse generator is connected, the B track specifies the counting direction. In RS422 mode, a differential signal is expected on the B̄ track.
- When using a counter / pulse generator without zero pulse C, the terminal points for the C track are not connected.

6.8.2 TTL mode

Connection of TTL encoders with or without zero pulse

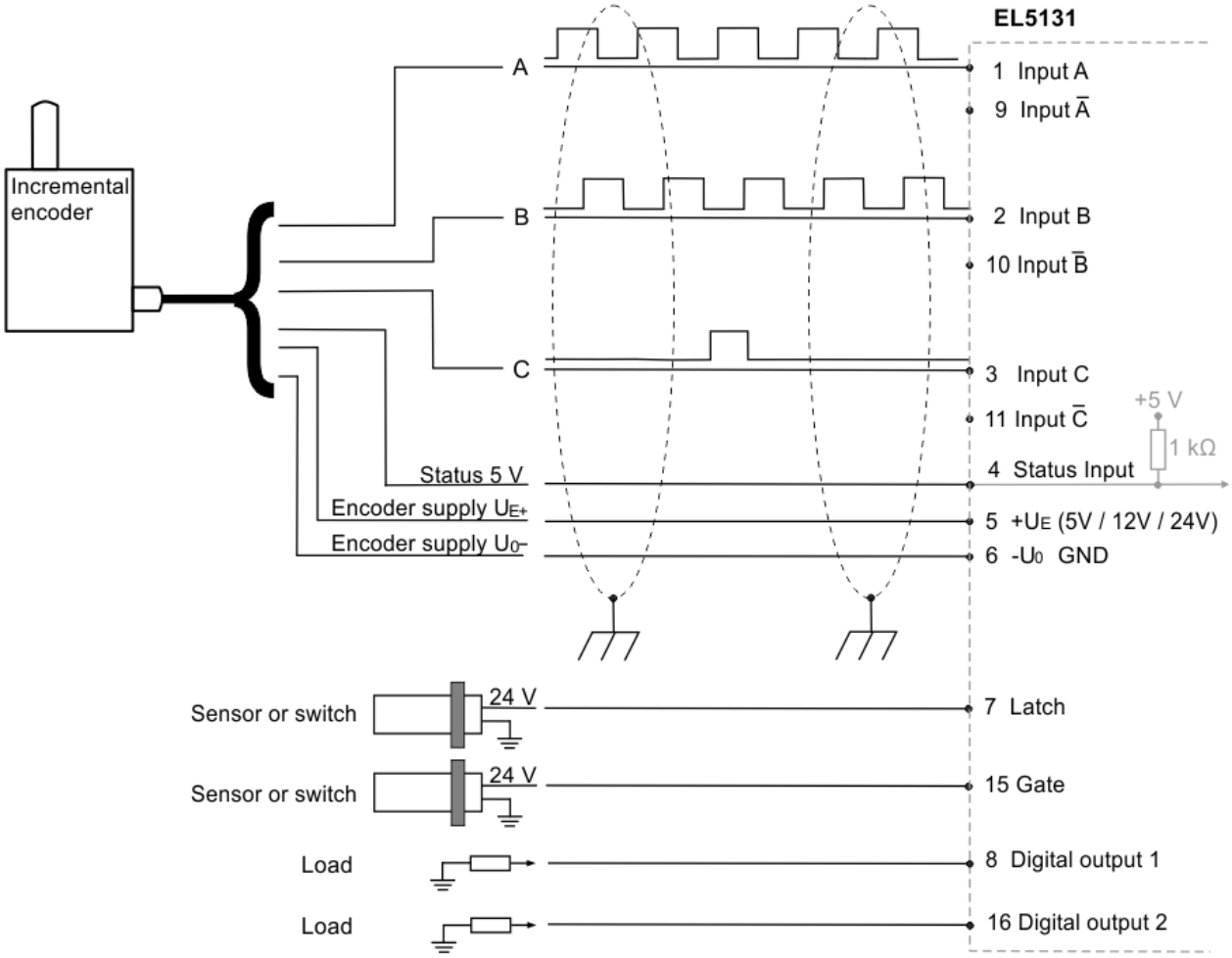


Fig. 32: Connection encoder TTL mode with zero pulse

Connection instructions

- In TTL mode the inverse inputs are not connected.
- When using an encoder without zero pulse C, the terminal points for the C track are not connected.

Connection of TTL counters / pulse generators with or without zero pulse

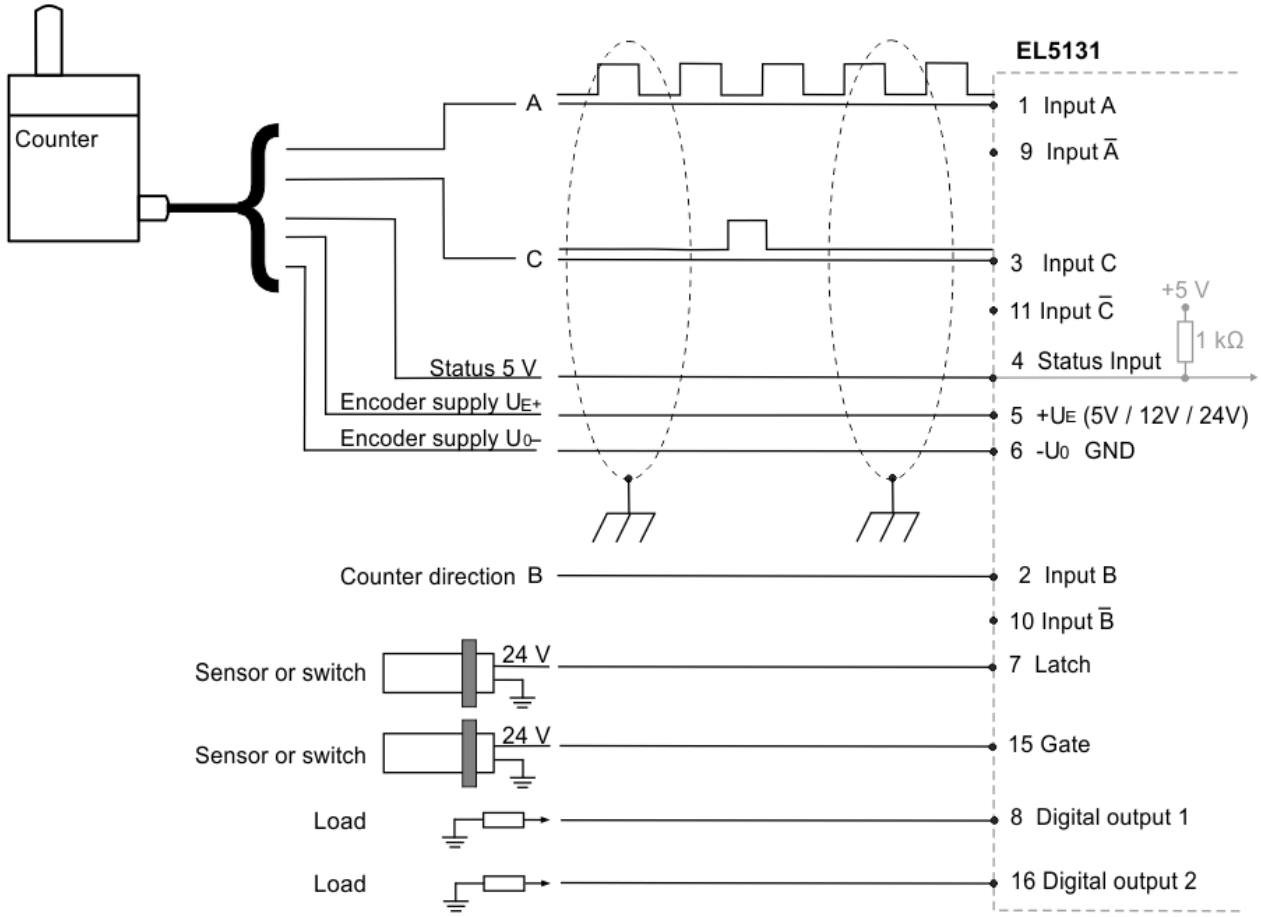


Fig. 33: Connection counter / pulse generator TTL mode with zero pulse

Connection instructions

- In TTL mode the inverse inputs are not connected.
- When a counter/pulse generator is connected, the B track specifies the counting direction. Input B̄ is not connected.
- When using a counter / pulse generator without zero pulse C, the terminal points for the C track are not connected.

6.8.3 Open Collector mode

Connection of Open Collector encoders with or without zero pulse

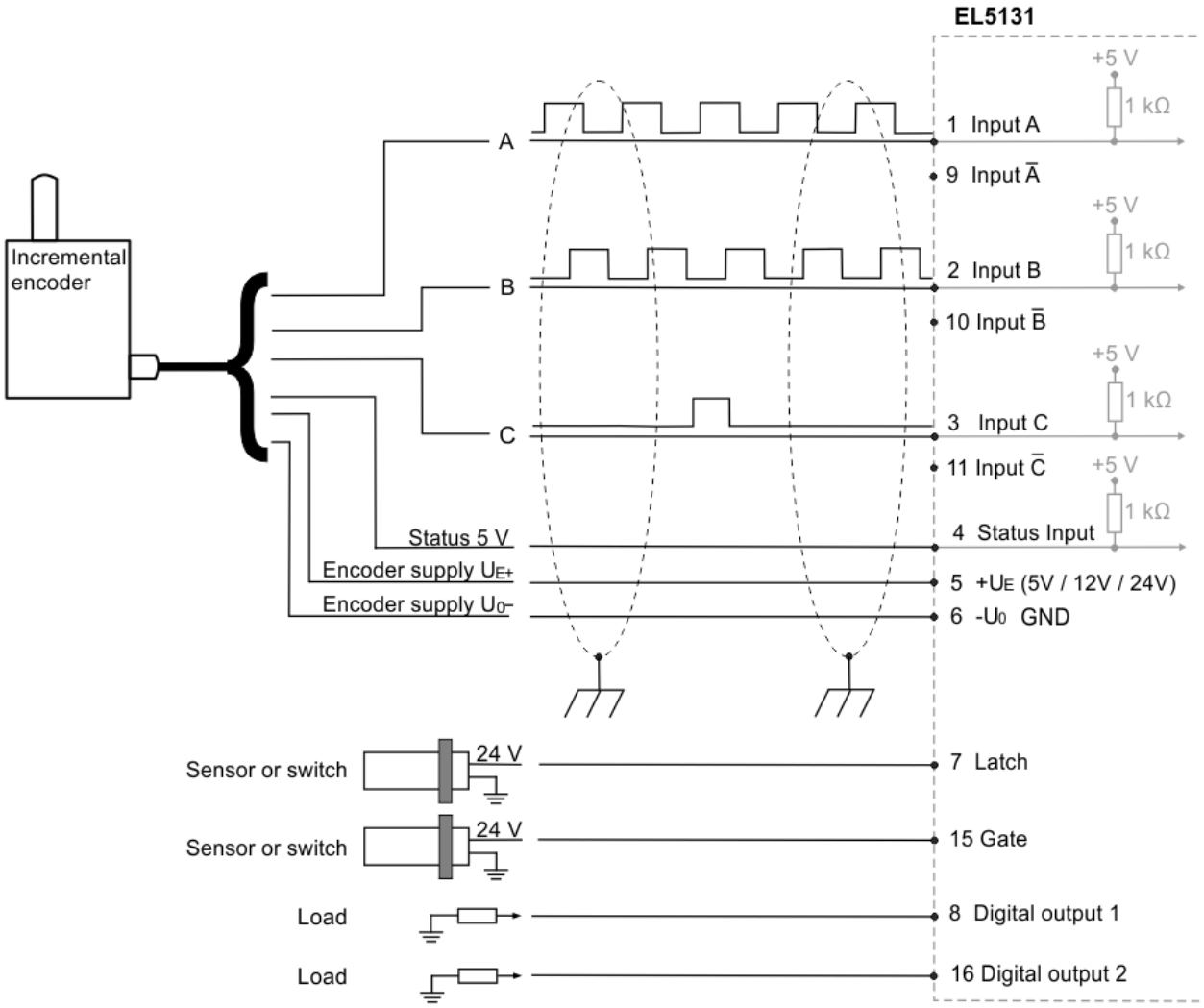


Fig. 34: Connection encoder Open Collector mode with zero pulse

Connection instructions

- In Open Collector mode the inverse inputs are not connected.
- When using an encoder without zero pulse C, the terminal points for the C track are not connected.

Connection of Open Collector counters / pulse generators with or without zero pulse

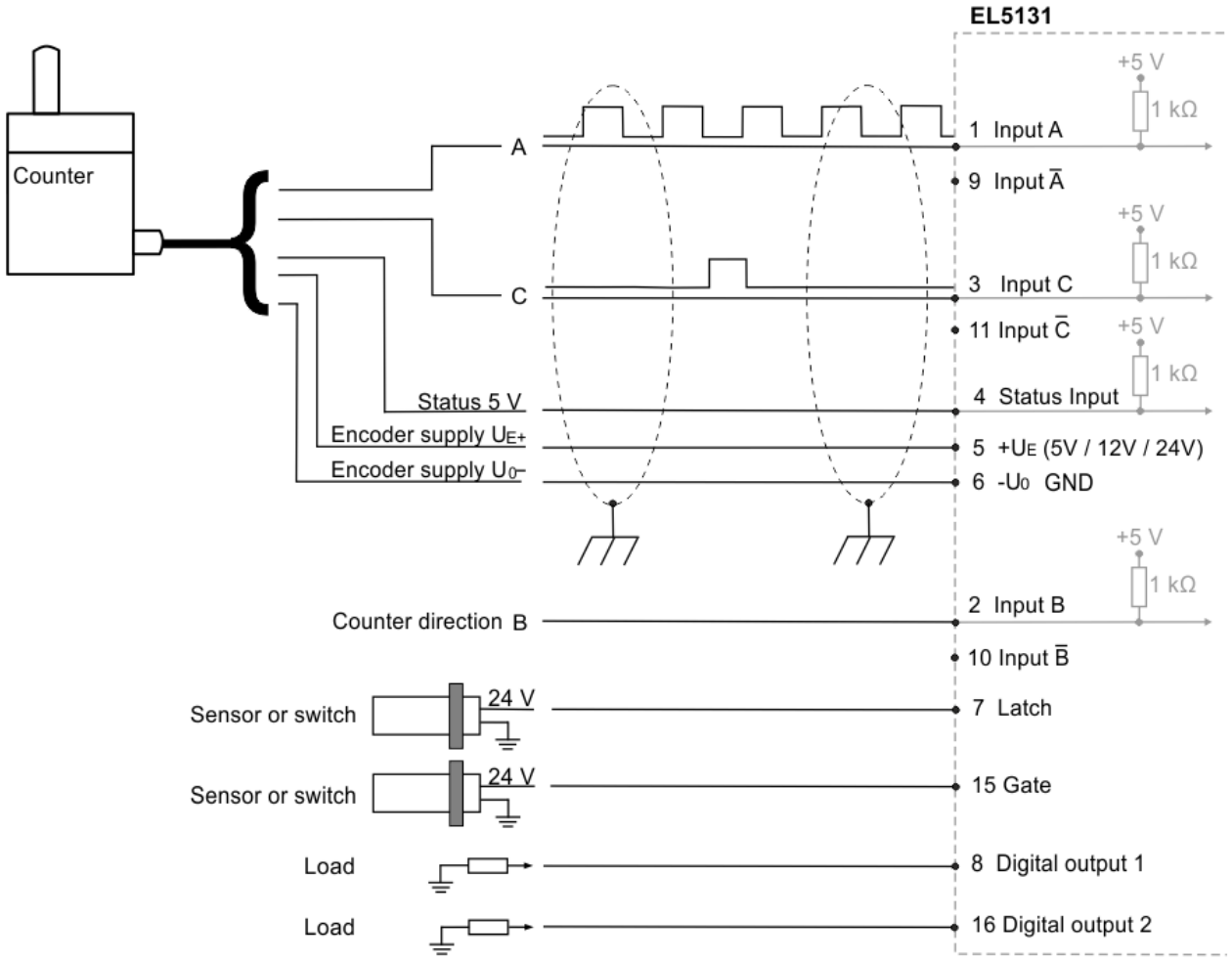


Fig. 35: Connection counter / pulse generator Open Collector mode with zero pulse

Connection instructions

- In Open Collector mode the inverse inputs are not connected.
- When a counter/pulse generator is connected, the B track specifies the counting direction. Input B̄ is not connected.
- When using a counter / pulse generator without zero pulse C, the terminal points for the C track are not connected.

6.9 EL5131 - LEDs

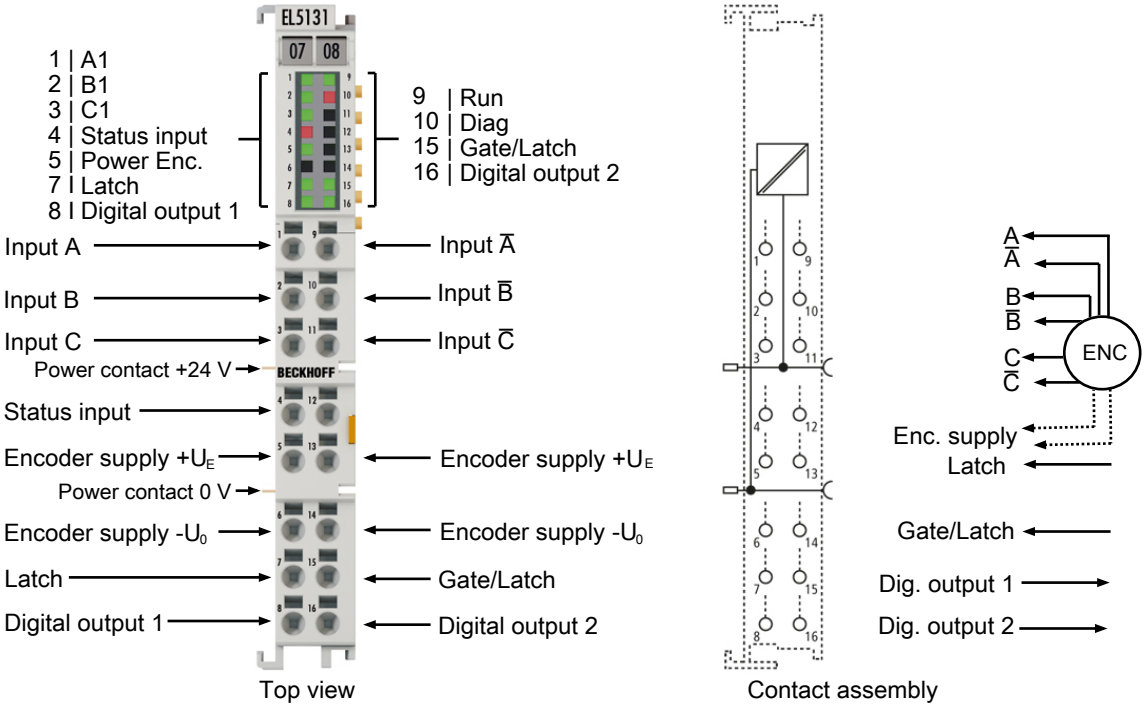


Fig. 36: EL5131 - LEDs

No.	Name	Color	Description	
1	A1	green, red	**green**: indicates TRUE level **red**: There is a wire break at the respective input. Diagnosis is only possible if the following three conditions are met. • The corresponding input is differentially wired, Corresponds to: "Counter mode" 0x8001:1D = 0 (Encoder RS422 (diff. input)) or "Counter mode" 0x8001:1D = 1 (Counter RS422 (diff. input)) • The "Error Detection" of the respective input is enabled: "Error Detection A" 0x8000:0B = TRUE "Error Detection B" 0x8000:0C = TRUE "Error Detection C" 0x8000:0D = TRUE • The differential voltage $V_{ID\ Low}$ is typically less than 0.67 V (subject to change).	
2	B1			
3	C1			
4	Status Input	red	Fault signal input from rotary encoder. Input is internally connected to 5 V via a pull-up resistance. The encoder output must actively pull the signal against GND.	
			on	Output active at the encoder, a fault message is present at the encoder
			off	Output not active at the encoder; no fault message is present
5	Power Enc.	green	Operating voltage display for incremental encoder power supply	
7	Latch	green	is lit, if a signal (+24 V) is connected to the latch input	
8	Digital output 1	green	is lit, if output 1 is set (+24 V_{DC})	
9	Run	green	This LED indicates the terminal's operating state:	
			off	State of the EtherCAT State Machine [▶ 32]: **INIT** = initialization of the terminal or BOOTSTRAP = function for terminal firmware updates [▶ 244]
			flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different default settings set
			single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager [▶ 114] 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
10	Diag	red	Initialization process active or state of the EtherCAT State Machine: **BOOT**	
15	Gate/Latch	green	is lit if a signal (+24 V) is present at the gate/latch input	
16	Digital output 2	green	is lit if output 2 is set (+24 V_{DC})	

6.10 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.

7 Commissioning

7.1 TwinCAT Quick Start

TwinCAT is a development environment for real-time control including a multi PLC system, NC axis control, programming and operation. The whole system is mapped through this environment and enables access to a programming environment (including compilation) for the controller. Individual digital or analog inputs or outputs can also be read or written directly, in order to verify their functionality, for example.

For further information, please refer to http://infosys.beckhoff.com:

- **EtherCAT System Manual:**
  Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- **TwinCAT 2** → TwinCAT System Manager → I/O Configuration
- In particular, for TwinCAT – driver installation:
  Fieldbus components → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation

Devices contain the relevant terminals for the actual configuration. All configuration data can be entered directly via editor functions (offline) or via the `scan function (online):

- **“offline”**: The configuration can be customized by adding and positioning individual components. These can be selected from a directory and configured.
  - The procedure for the offline mode can be found under http://infosys.beckhoff.com:
    TwinCAT 2 → TwinCAT System Manager → IO Configuration → Add an I/O device
- **“online”**: The existing hardware configuration is read
  - See also http://infosys.beckhoff.com:
    Fieldbus components → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation → Searching for devices

The following relationship is envisaged between the user PC and individual control elements:

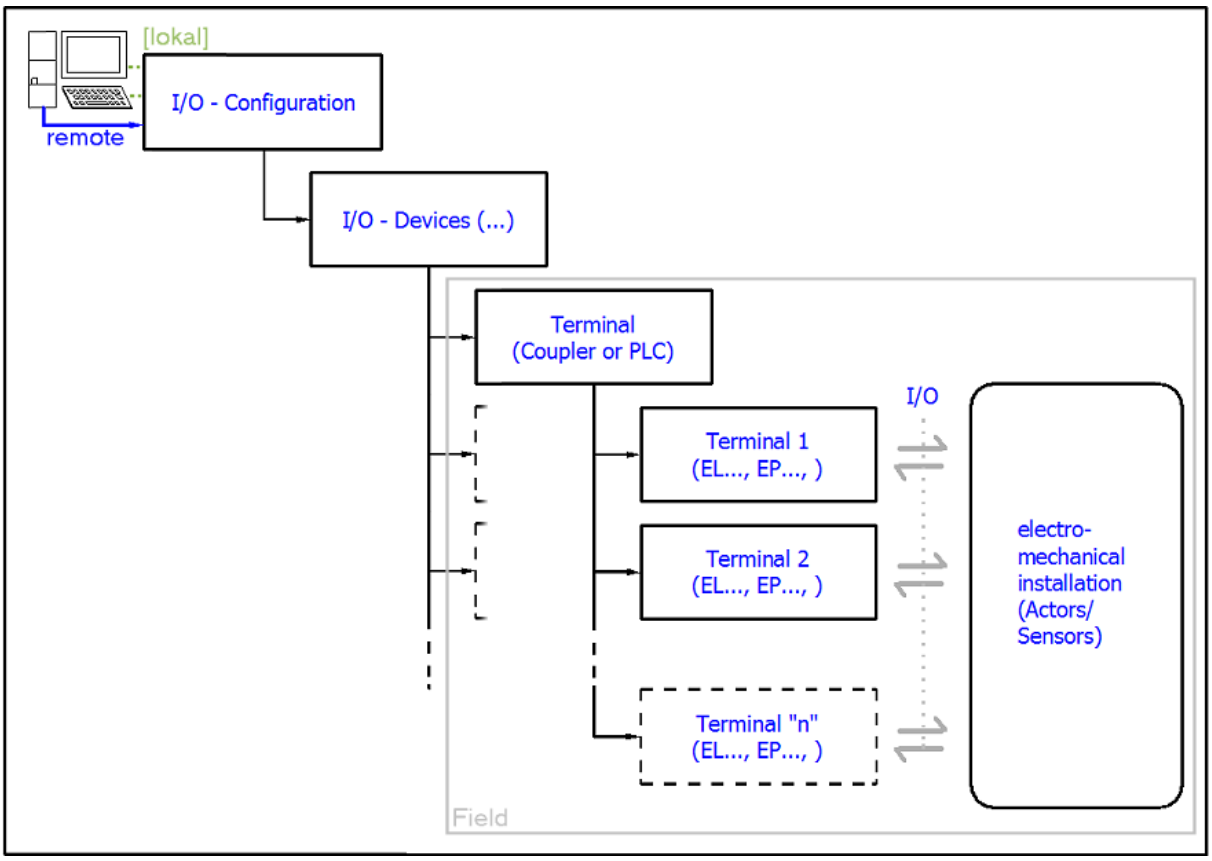


Fig. 37: Relationship between user side (commissioning) and installation

Insertion of certain components (I/O device, terminal, box...) by users functions the same way as in TwinCAT 2 and TwinCAT 3. The descriptions below relate solely to the online procedure.

Example configuration (actual configuration)

Based on the following example configuration, the subsequent subsections describe the procedure for TwinCAT 2 and TwinCAT 3:

- **CX2040** control system (PLC) including **CX2100-0004** power supply unit
- Connected to CX2040 on the right (E-bus):
  EL1004 (4-channel digital input terminal 24 V_{DC})
- Linked via the X001 port (RJ-45): **EK1100** EtherCAT Coupler
- Connected to the EK1100 EtherCAT Coupler on the right (E-bus):
  EL2008 (8-channel digital output terminal 24 V_{DC}; 0.5 A)
- (Optional via X000: a link to an external PC for the user interface)

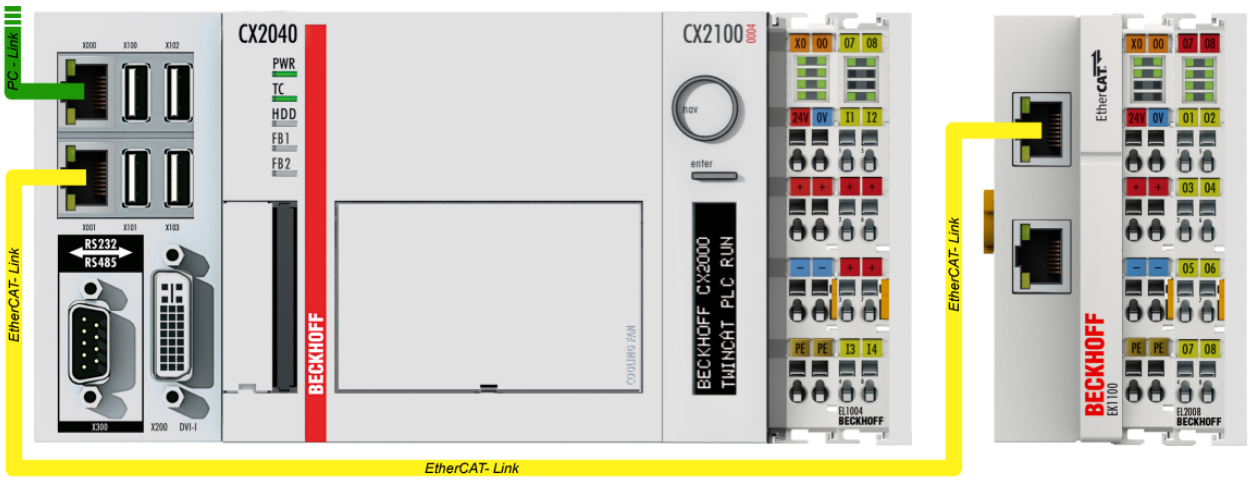


Fig. 38: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

Note that all combinations of a configuration are possible; for example, the EL1004 terminal could also be connected after the coupler, or the EL2008 terminal could additionally be connected to the CX2040 on the right, in which case the EK1100 coupler wouldn't be necessary.

7.1.1 TwinCAT 2

Startup

TwinCAT 2 basically uses two user interfaces: the TwinCAT System Manager for communication with the electromechanical components and TwinCAT PLC Control for the development and compilation of a controller. The starting point is the TwinCAT System Manager.

After successful installation of the TwinCAT system on the PC to be used for development, the TwinCAT 2 System Manager displays the following user interface after startup:

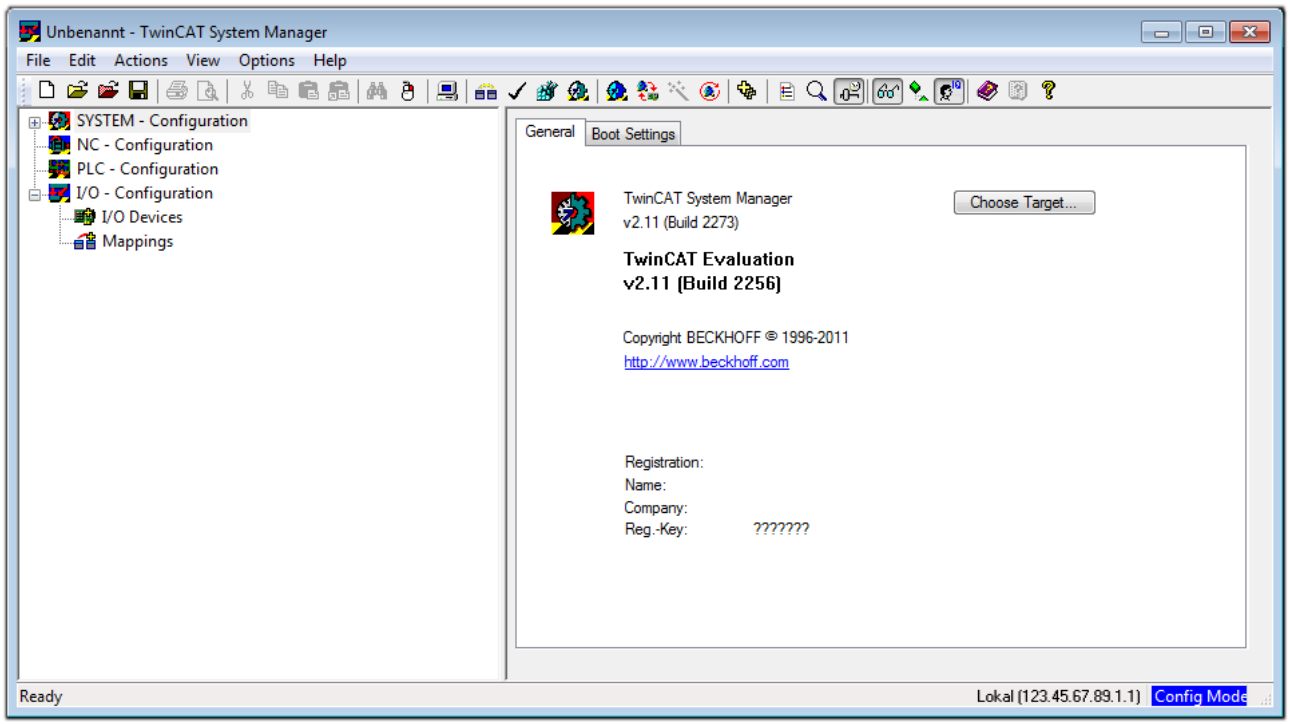


Fig. 39: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system, including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thus the next step is “Insert Device [▶ 68]”.

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. In the menu under “Actions” → “Choose Target System...”, the following window is opened for this via the symbol “  ” or the “F8” key:

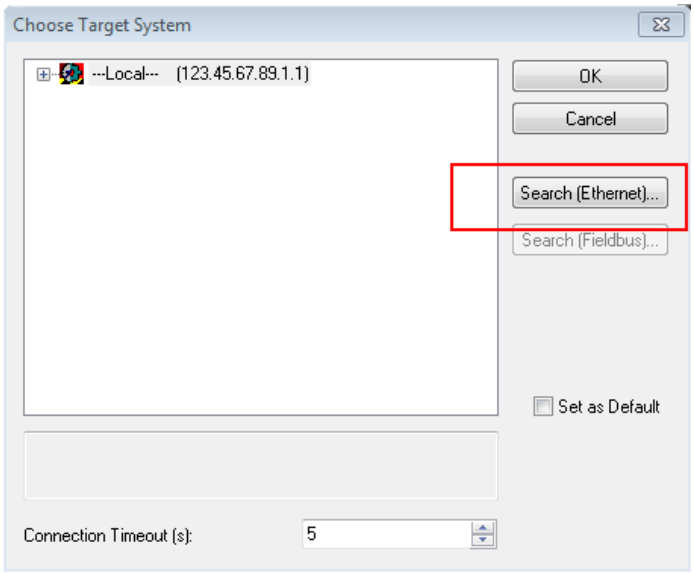


Fig. 40: Selection of the target system

Use “Search (Ethernet)...” to enter the target system. Thus another dialog opens to either:

- enter the known computer name after “Enter Host Name / IP:” (as shown in red)
- perform a “Broadcast Search” (if the exact computer name is not known)
- enter the known computer – IP or AmsNetID

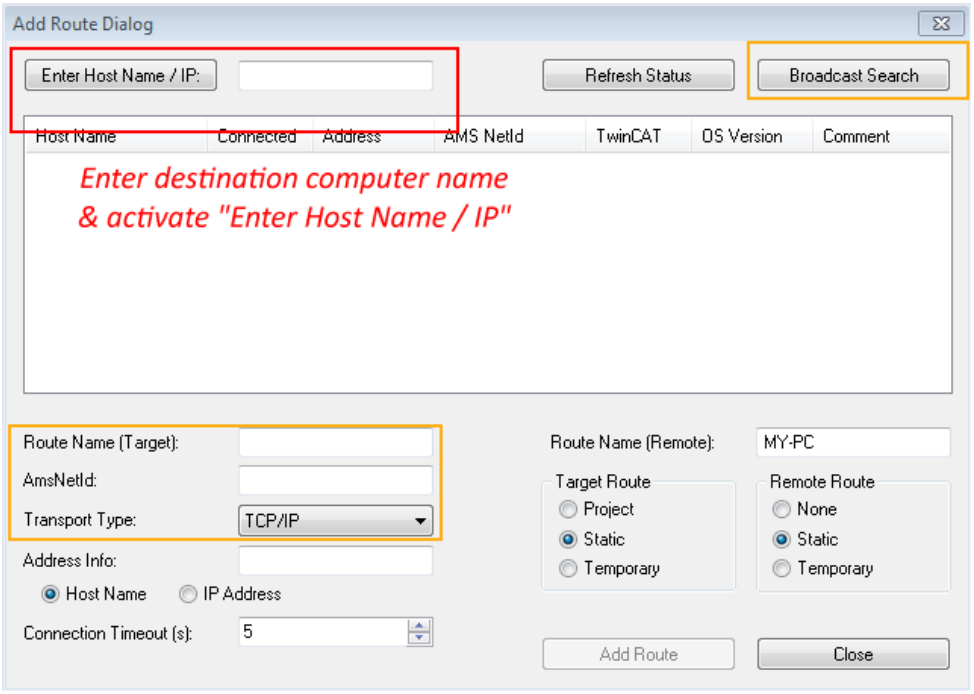
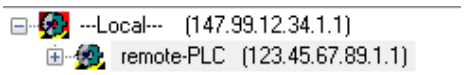


Fig. 41: specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (a correct password may have to be entered before this):



After confirmation with “OK”, the target system can be accessed via the System Manager.

Adding devices

In the configuration tree of the TwinCAT 2 System Manager user interface on the left, select “I/O Devices” and then right-click to open a context menu and select “Scan Devices…”, or start the action in the menu bar via  . The TwinCAT System Manager may first have to be set to “Config Mode” via  or via the menu
“Actions” → “Set/Reset TwinCAT to Config Mode…” (Shift + F4).

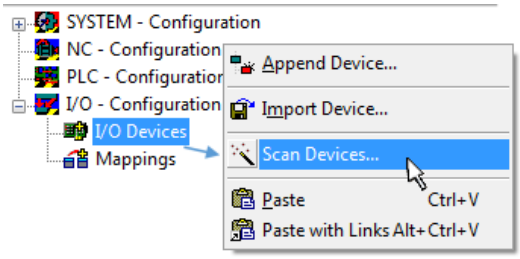


Fig. 42: Select “Scan Devices...”

Confirm the warning message, which follows, and select the “EtherCAT” devices in the dialog:

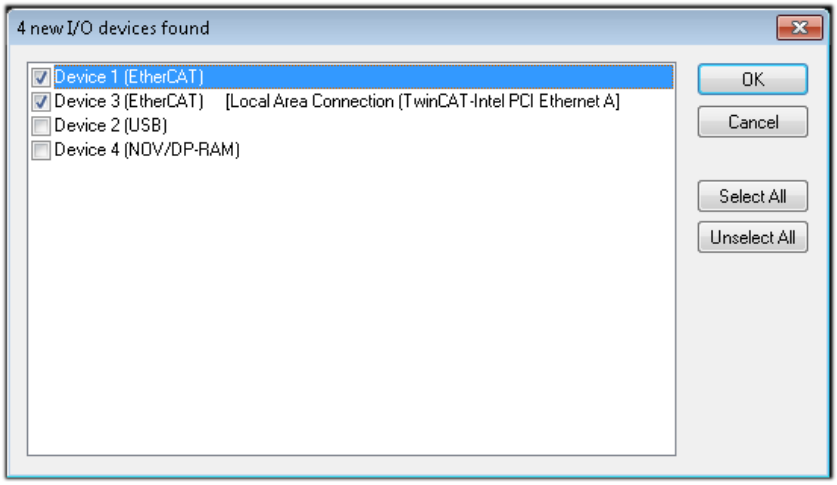


Fig. 43: Automatic detection of I/O devices: selection of the devices to be integrated

Confirm the message “Find new boxes”, in order to determine the terminals connected to the devices. “Free Run” enables manipulation of input and output values in “Config Mode” and should also be acknowledged.

Based on the example configuration [▶ 64] described at the beginning of this section, the result is as follows:

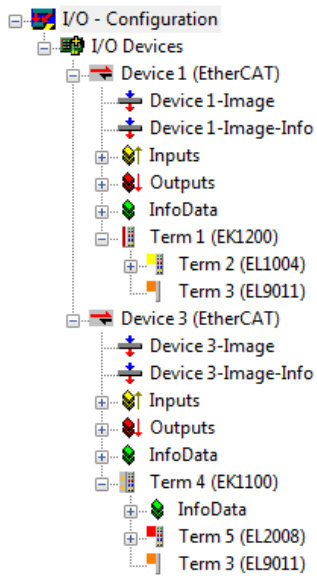


Fig. 44: Mapping of the configuration in the TwinCAT 2 System Manager

The whole process consists of two stages, which can also be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan (search function) can also be initiated by selecting “Device ...” from the context menu, which then only reads the elements below which are present in the configuration:

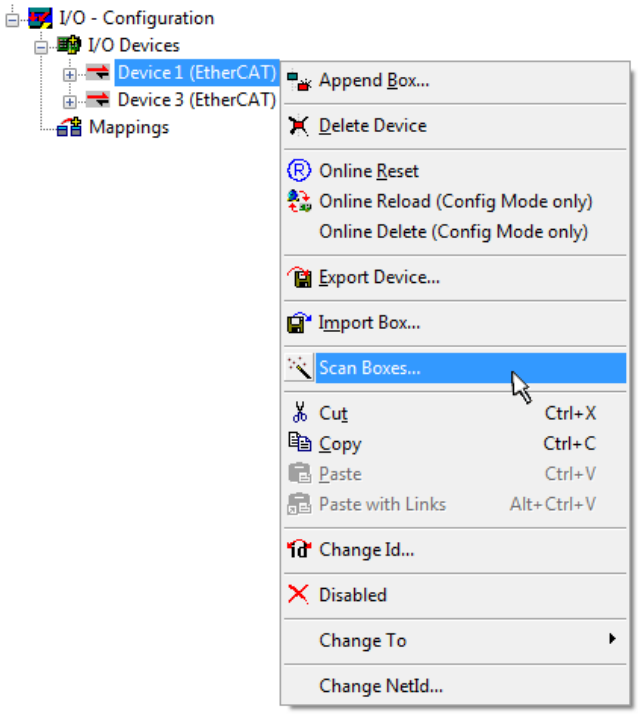


Fig. 45: Reading of individual terminals connected to a device

This functionality is useful if the actual configuration is modified at short notice.

Programming and integrating the PLC

TwinCAT PLC Control is the development environment for generating the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

- **Text-based languages**
  - Instruction List (IL)
  - Structured Text (ST)

- **Graphical languages**
  - Function Block Diagram (FBD)
  - Ladder Diagram (LD)
  - The Continuous Function Chart Editor (CFC)
  - Sequential Function Chart (SFC)

The following section refers solely to Structured Text (ST).

After starting TwinCAT PLC Control, the following user interface is shown for an initial project:

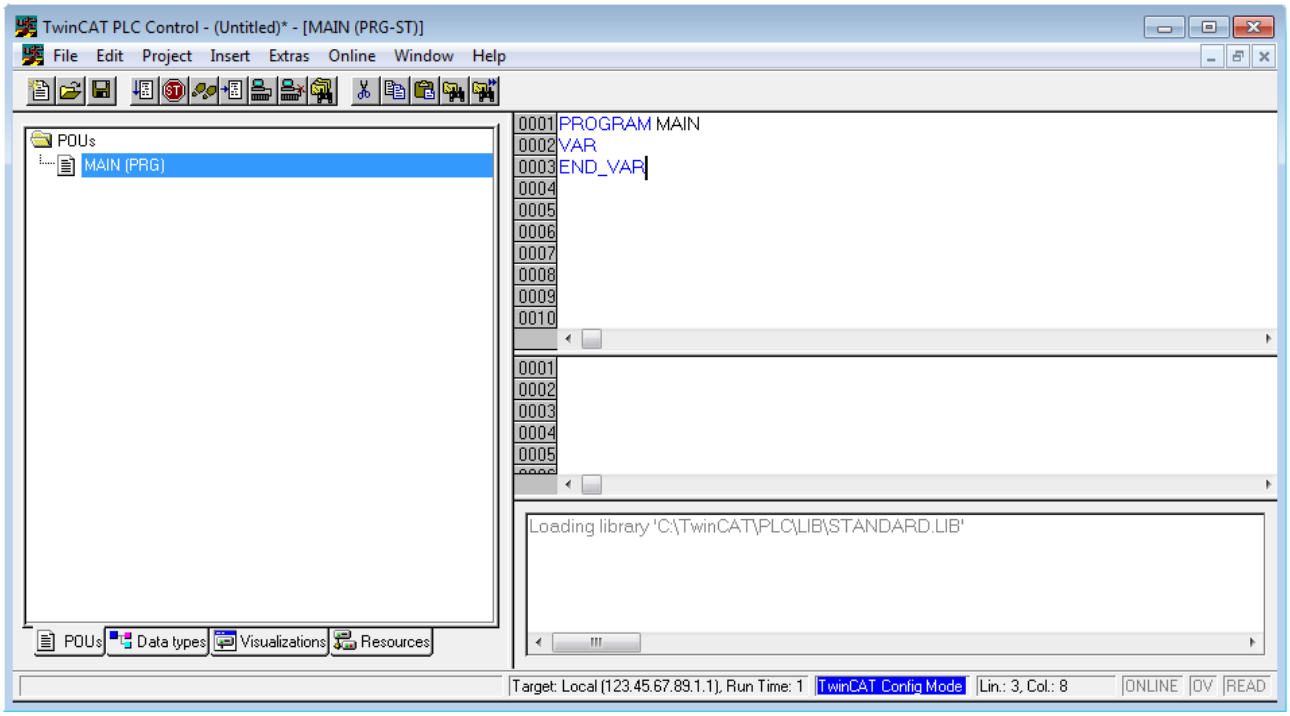


Fig. 46: TwinCAT PLC Control after startup

Example variables and an example program have been created and stored under the name “PLC_example.pro”:

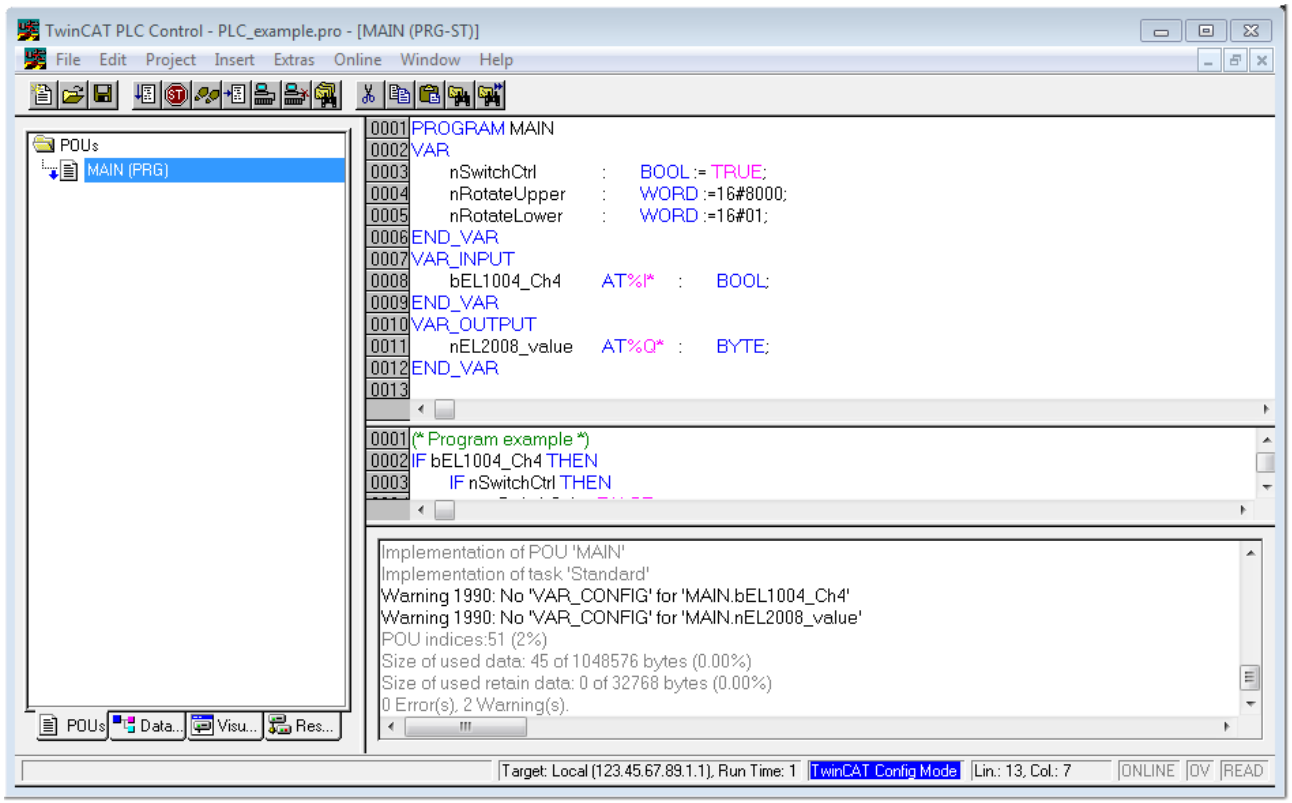


Fig. 47: Example program with variables after a compile process (without variable integration)

Warning 1990 (missing “VAR_CONFIG”) after a compile process indicates that the variables defined as external (with the ID “AT%I*” or “AT%Q*”) have not been assigned. After successful compilation, TwinCAT PLC Control creates a “*.tpy” file in the directory in which the project was stored. This file (“*.tpy”) contains variable assignments and is not known to the System Manager, hence the warning. Once the System Manager has been notified, the warning no longer appears.

First, integrate the TwinCAT PLC Control project in the **System Manager**. This is performed via the context menu of the PLC configuration (right-click) and selecting “Append PLC Project…”:

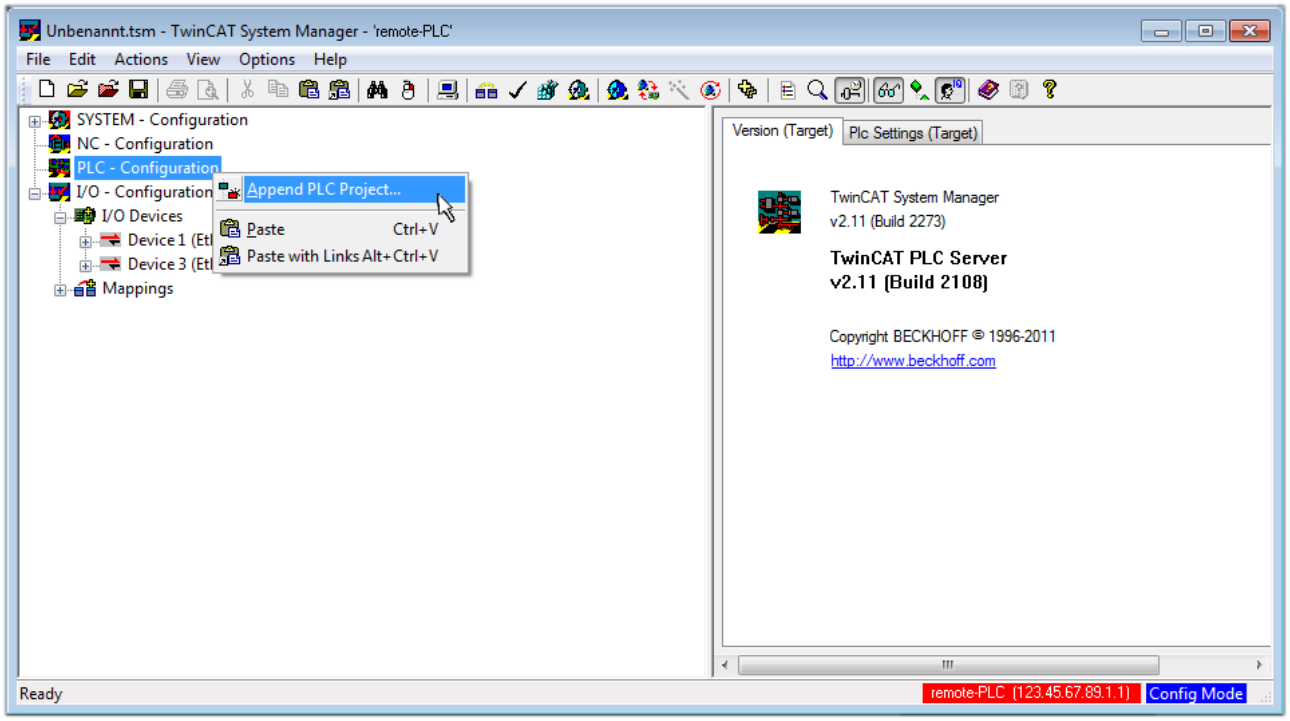


Fig. 48: Appending the TwinCAT PLC Control project

Select the PLC configuration “PLC_example.tpy” in the browser window that opens. The project including the two variables identified with “AT” are then integrated in the configuration tree of the System Manager:

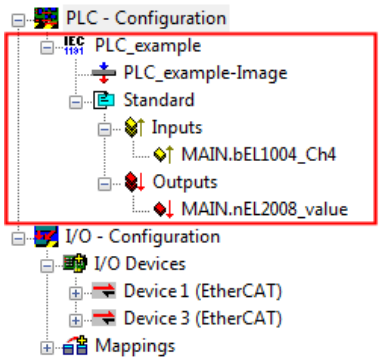


Fig. 49: PLC project integrated in the PLC configuration of the System Manager

The two variables “bEL1004_Ch4” and “nEL2008_value” can now be assigned to certain process objects of the I/O configuration.

Assigning variables

Open a window for selecting a suitable process object (PDO) via the context menu of a variable of the integrated project “PLC_example” and via “Modify Link...” “Standard”:

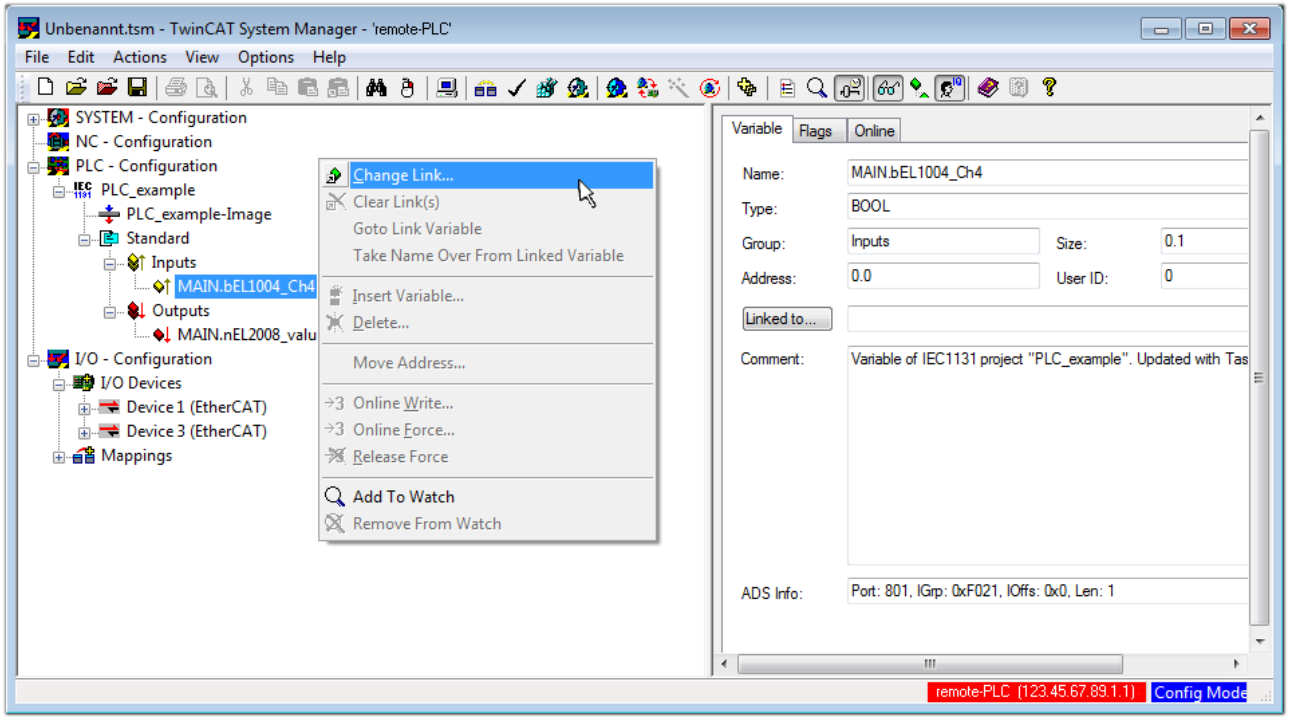


Fig. 50: Creating the links between PLC variables and process objects

In the window that opens, the process object for the “bEL1004_Ch4” BOOL-type variable can be selected from the PLC configuration tree:

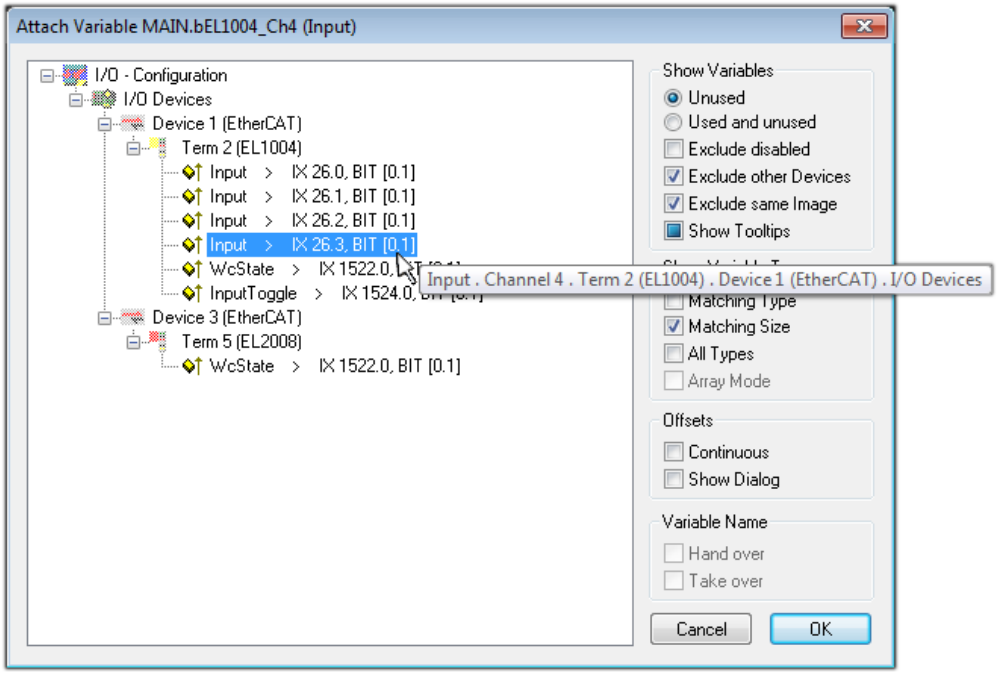


Fig. 51: Selecting BOOL-type PDO

According to the default setting, only certain PDO objects are now available for selection. In this example, the input of channel 4 of the EL1004 terminal is selected for linking. In contrast, the checkbox “All types” must be ticked to create the link for the output variables, in order to allocate a set of eight separate output bits to a byte variable in this case. The following diagram shows the whole process:

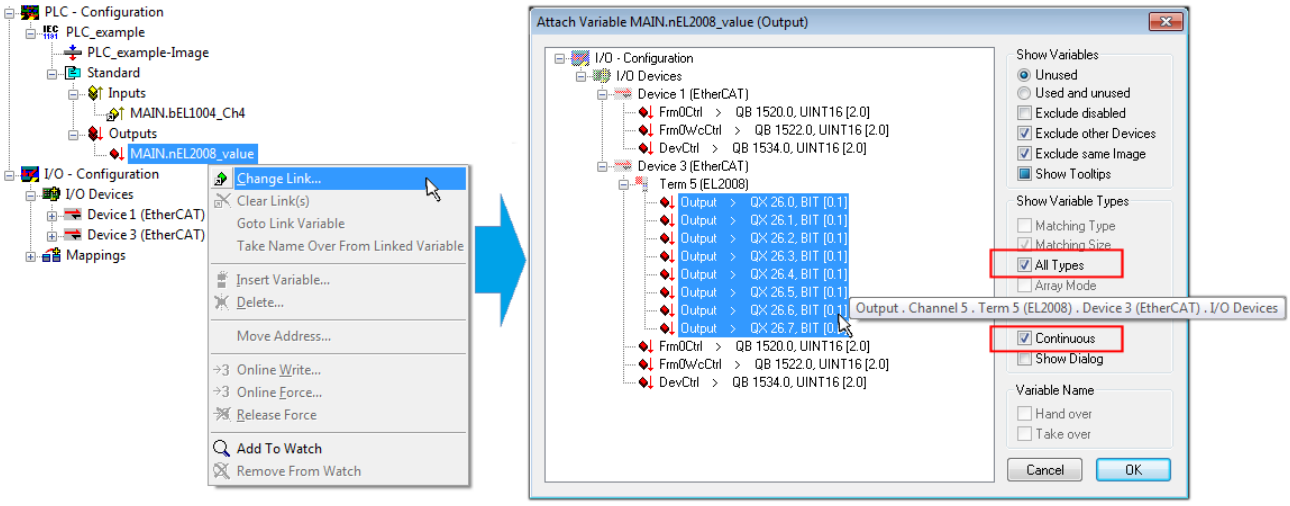


Fig. 52: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol ( ) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

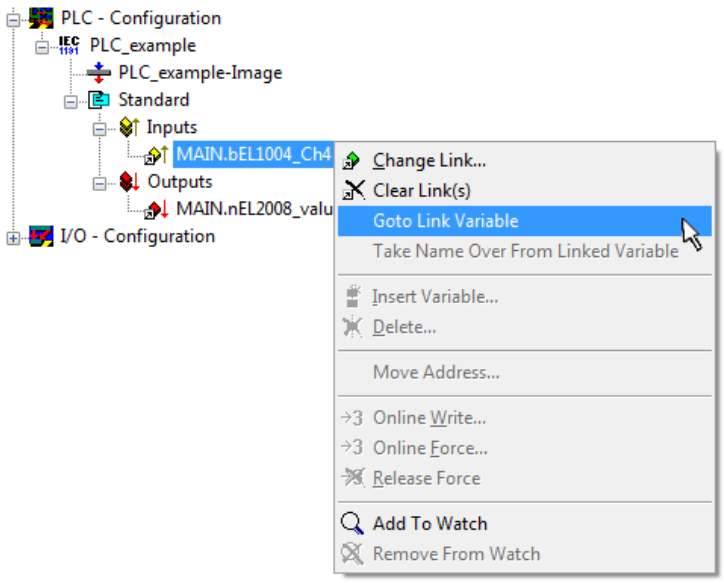
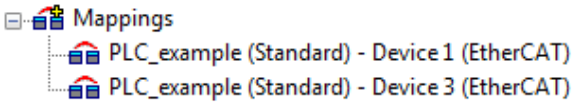


Fig. 53: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

The process of assigning variables to the PDO is completed via the menu option “Actions” → “Create assignment”, or via  .

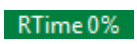
This can be visualized in the configuration:



The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word, integer or similar PDO, it is also possible to allocate this to a set of bit-standardized variables. Here, too, a “Goto Link Variable” can be executed in the other direction, so that the respective PLC instance can then be selected.

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated. First, the configuration can be verified via  (or via “Actions” → “Check Configuration”). If no error is present, the configuration can be activated via  (or via “Actions” → “Activate Configuration…”) to transfer the System Manager settings to the runtime system. Confirm the messages “Old configurations will be overwritten!” and “Restart TwinCAT system in Run mode” with “OK”.

A few seconds later, the real-time status 

is displayed at the bottom right in the System Manager. The PLC system can then be started as described below.

Starting the controller

Starting from a remote system, the PLC control has to be linked with the embedded PC over the Ethernet via “Online” → “Choose Runtime System…”:

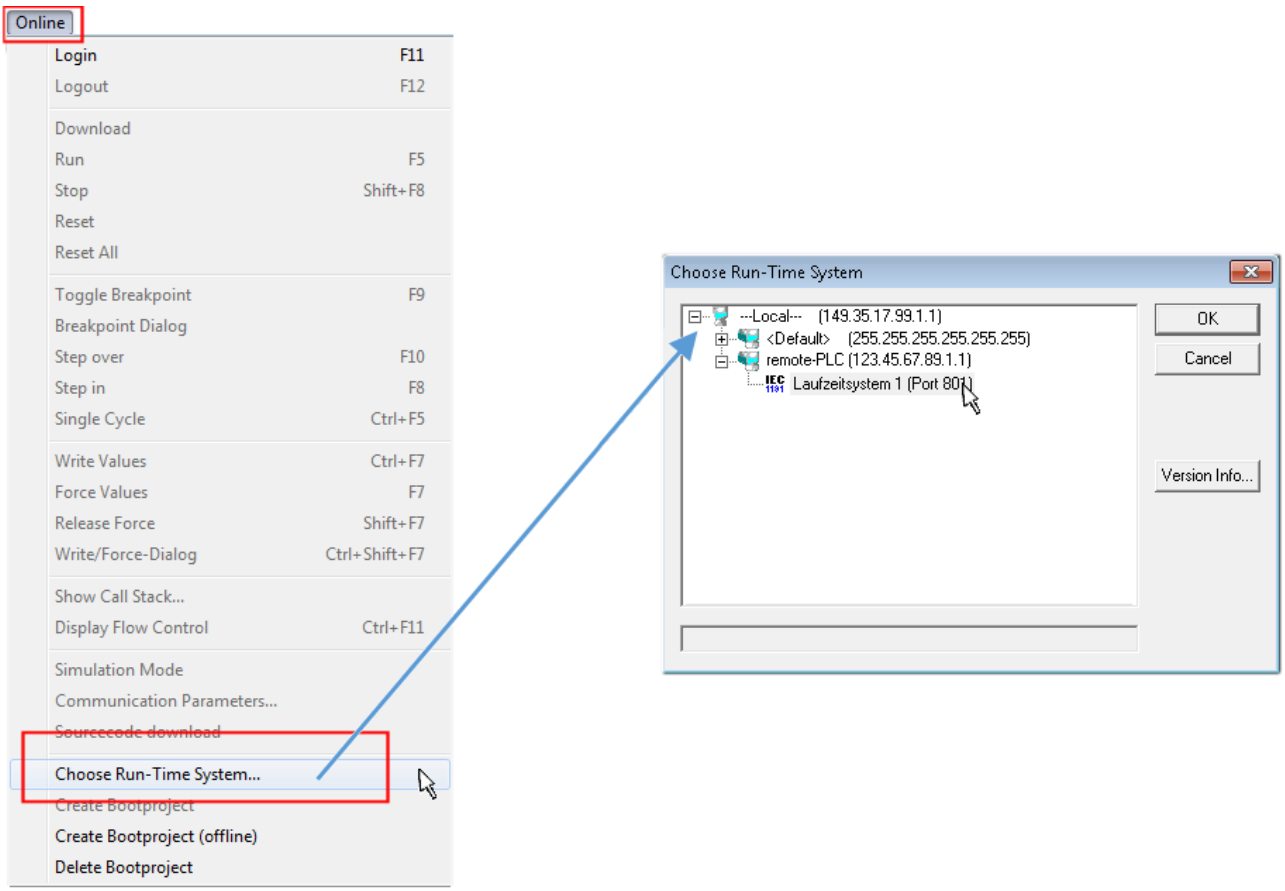


Fig. 54: Choose target system (remote)

In this example, “Runtime system 1 (port 801)” is selected and confirmed. Link the PLC with the real-time system via the menu option “Online” → “Login”, the F11 key or by clicking on the symbol . The control program can then be loaded for execution. This results in the message “No program on the controller! Should the new program be loaded?”, which should be confirmed with “Yes”. The runtime environment is ready for the program start:

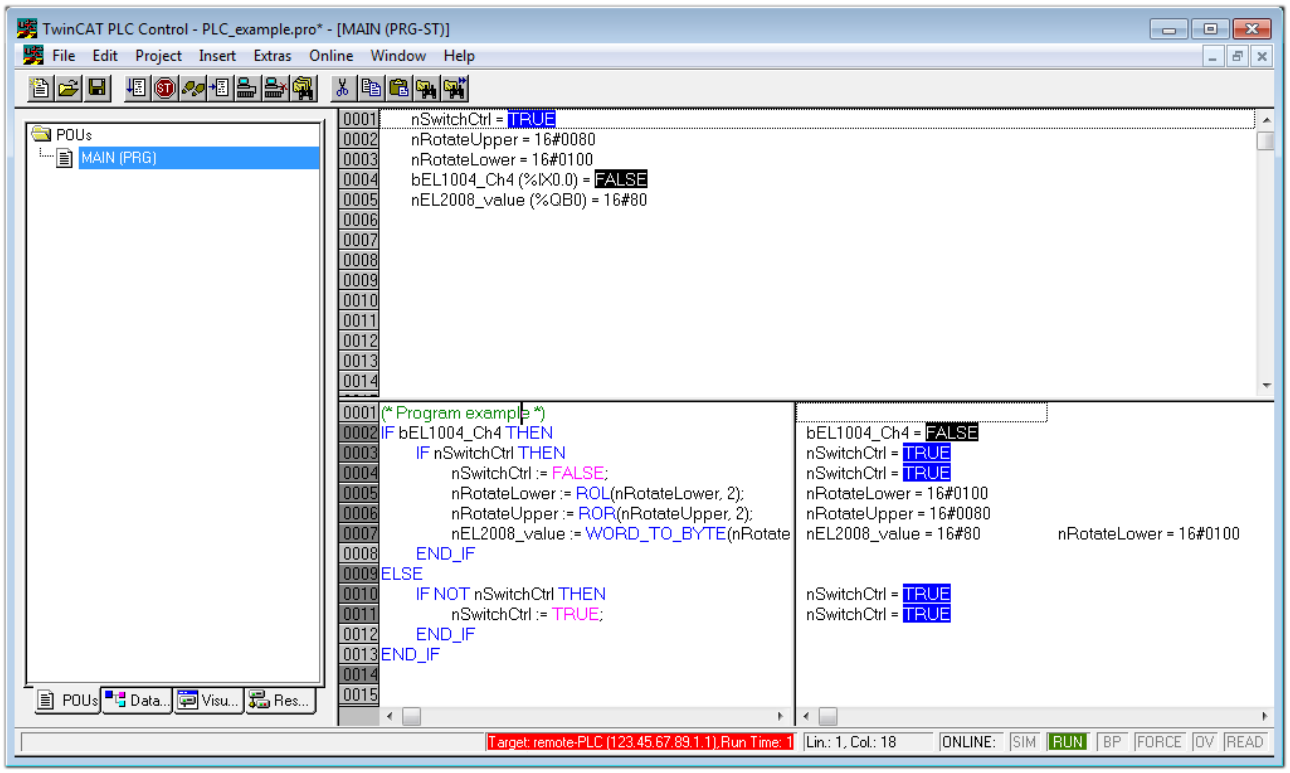


Fig. 55: PLC Control logged in, ready for program startup

The PLC can now be started via “Online” → “Run”, F5 key or .

7.1.2 TwinCAT 3

Startup

TwinCAT 3 makes the development environment areas available all together, with Microsoft Visual Studio: after startup, the project folder explorer appears on the left in the general window area (see “TwinCAT System Manager” of TwinCAT 2) for communication with the electromechanical components.

After successful installation of the TwinCAT system on the PC to be used for development, TwinCAT 3 (shell) displays the following user interface after startup:

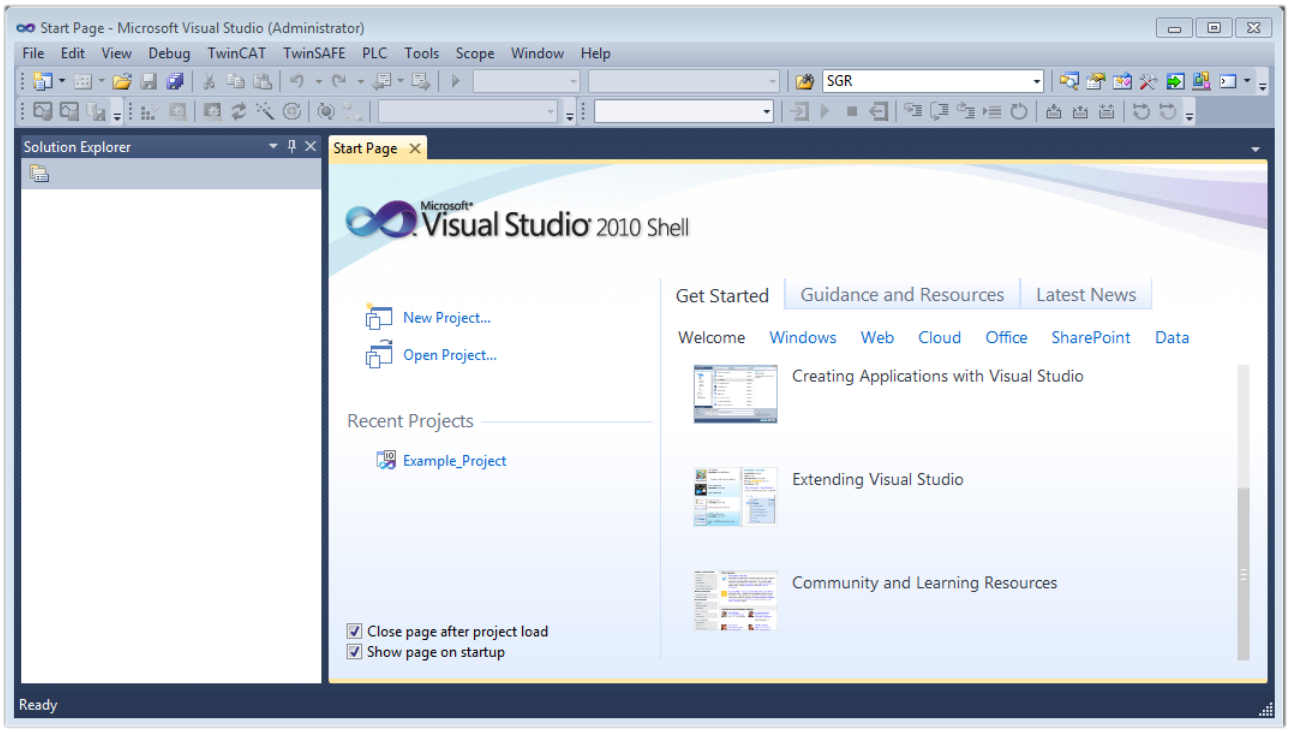


Fig. 56: Initial TwinCAT 3 user interface

First create a new project via  New TwinCAT Project... (or under “File”→“New”→ “Project…”). In the following dialog, make the corresponding entries as required (as shown in the diagram):

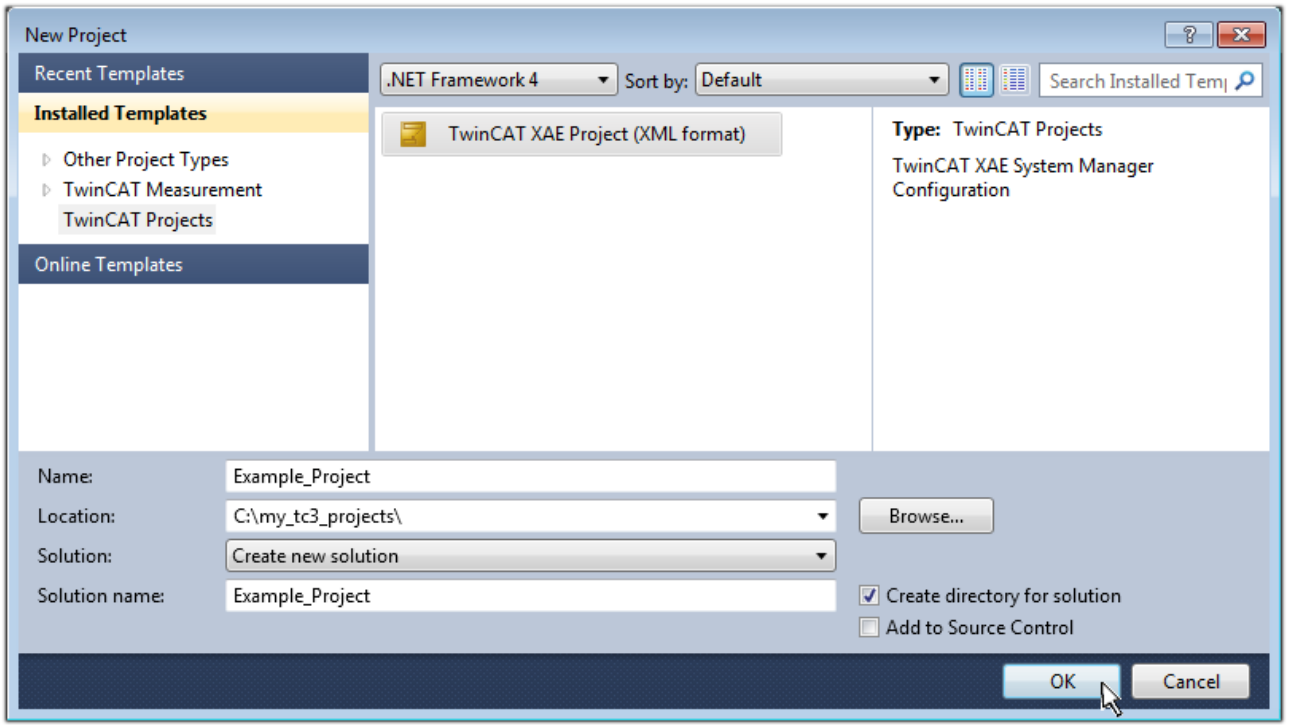


Fig. 57: Create new TwinCAT 3 project

The new project is then available in the project folder explorer:

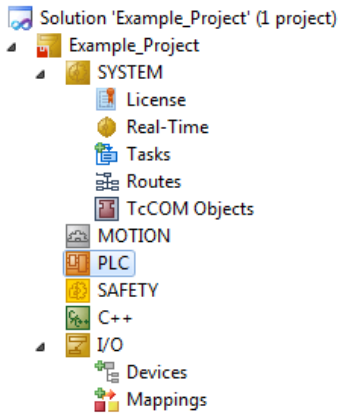
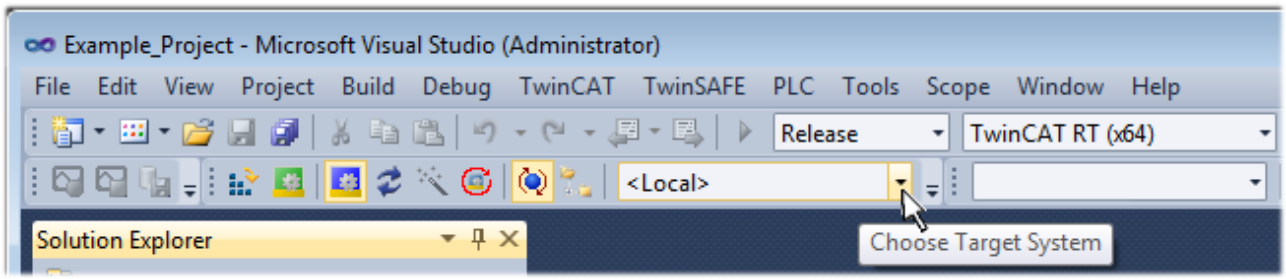


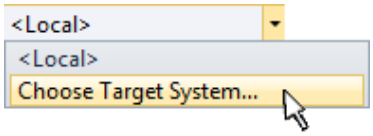
Fig. 58: New TwinCAT 3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC (locally), TwinCAT can be used in local mode and the process can be continued with the next step, "Insert Device [▶ 79]".

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. Via the symbol in the menu bar:



expand the pull-down menu:



and open the following window:

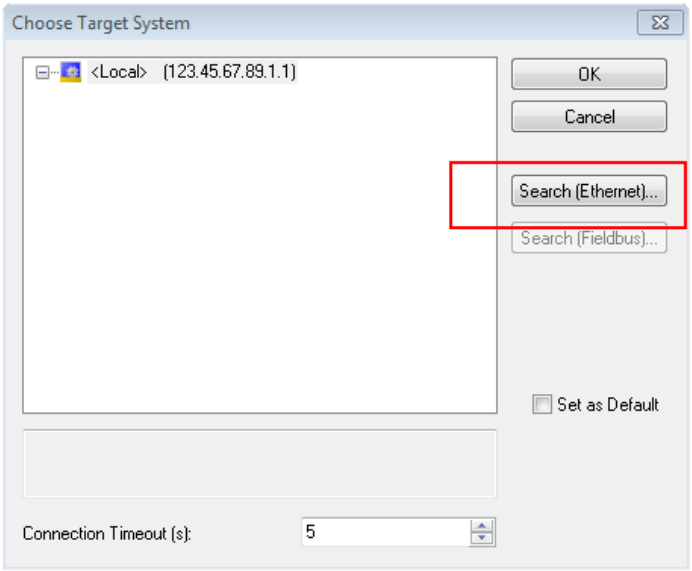


Fig. 59: Selection dialog: Choose the target system

Use “Search (Ethernet)...” to enter the target system. Thus another dialog opens to either:

- enter the known computer name after “Enter Host Name / IP:” (as shown in red)
- perform a “Broadcast Search” (if the exact computer name is not known)
- enter the known computer – IP or AmsNetID

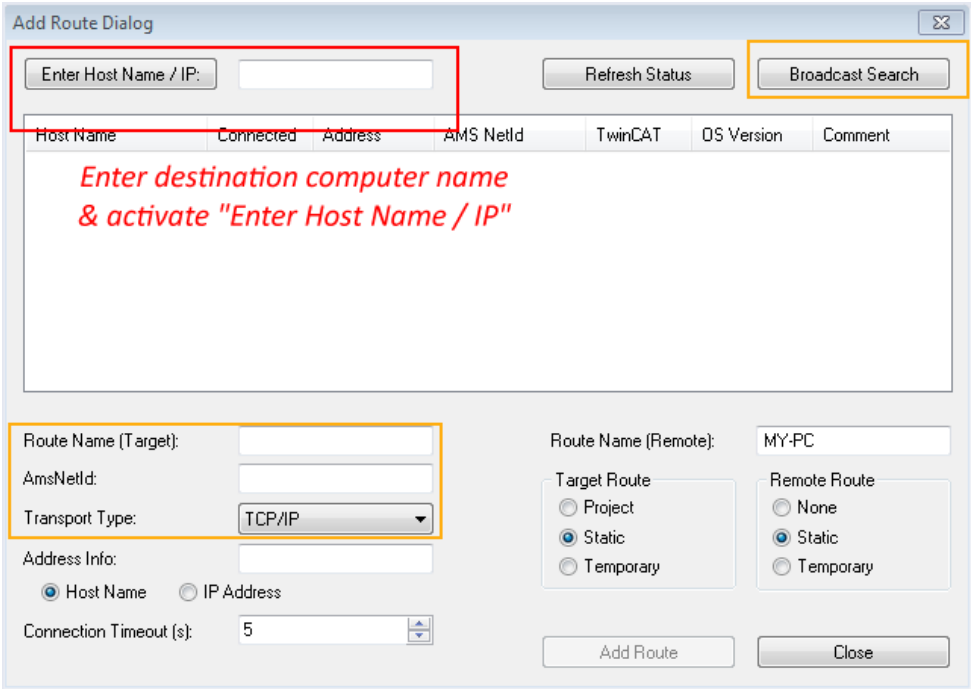
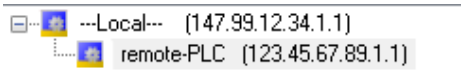


Fig. 60: specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (the correct password may have to be entered beforehand):



After confirmation with “OK” the target system can be accessed via the Visual Studio shell.

Adding devices

In the project folder explorer on the left of the Visual Studio shell user interface, select “Devices” within the element “I/O”, then right-click to open a context menu and select “Scan” or start the action via  in the menu bar. The TwinCAT System Manager may first have to be set to “Config mode” via  or via the menu “TwinCAT” → “Restart TwinCAT (Config Mode)”.

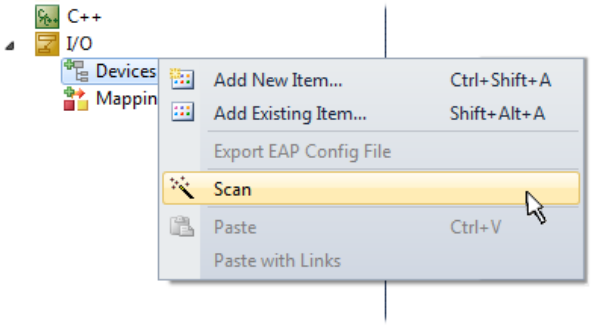


Fig. 61: Select “Scan”

Confirm the warning message, which follows, and select the “EtherCAT” devices in the dialog:

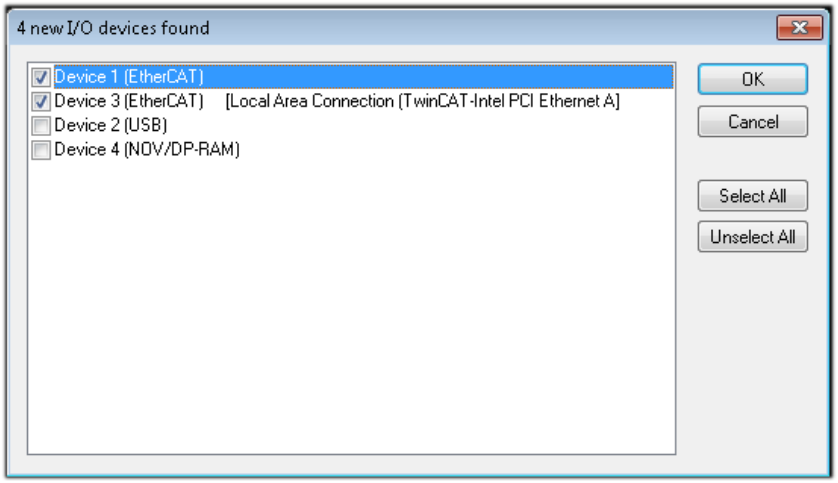


Fig. 62: Automatic detection of I/O devices: selection of the devices to be integrated

Confirm the message “Find new boxes”, in order to determine the terminals connected to the devices. “Free Run” enables manipulation of input and output values in “Config Mode” and should also be acknowledged.

Based on the example configuration [▶ 64] described at the beginning of this section, the result is as follows:

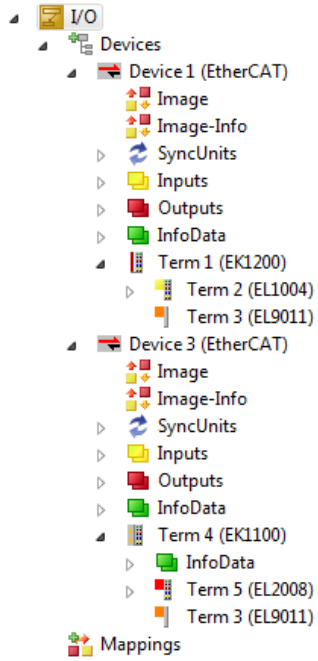


Fig. 63: Mapping of the configuration in VS shell of the TwinCAT 3 environment

The whole process consists of two stages, which can also be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan (search function) can also be initiated by selecting “Device ...” from the context menu, which then only reads the elements below which are present in the configuration:

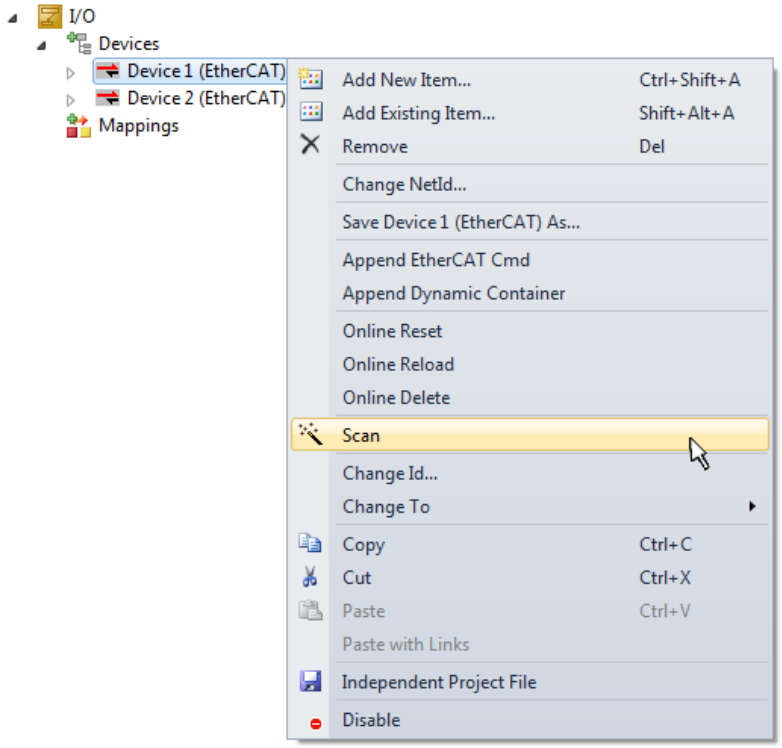


Fig. 64: Reading of individual terminals connected to a device

This functionality is useful if the actual configuration is modified at short notice.

Programming the PLC

TwinCAT PLC Control is the development environment for generating the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

- **Text-based languages**
  - Instruction List (IL)
  - Structured Text (ST)
- **Graphical languages**
  - Function Block Diagram (FBD)
  - Ladder Diagram (LD)
  - The Continuous Function Chart Editor (CFC)
  - Sequential Function Chart (SFC)

The following section refers solely to Structured Text (ST).

In order to create a programming environment, a PLC subproject is added to the example project via the context menu of the “PLC” in the project folder explorer by selecting “Add New Item….”:

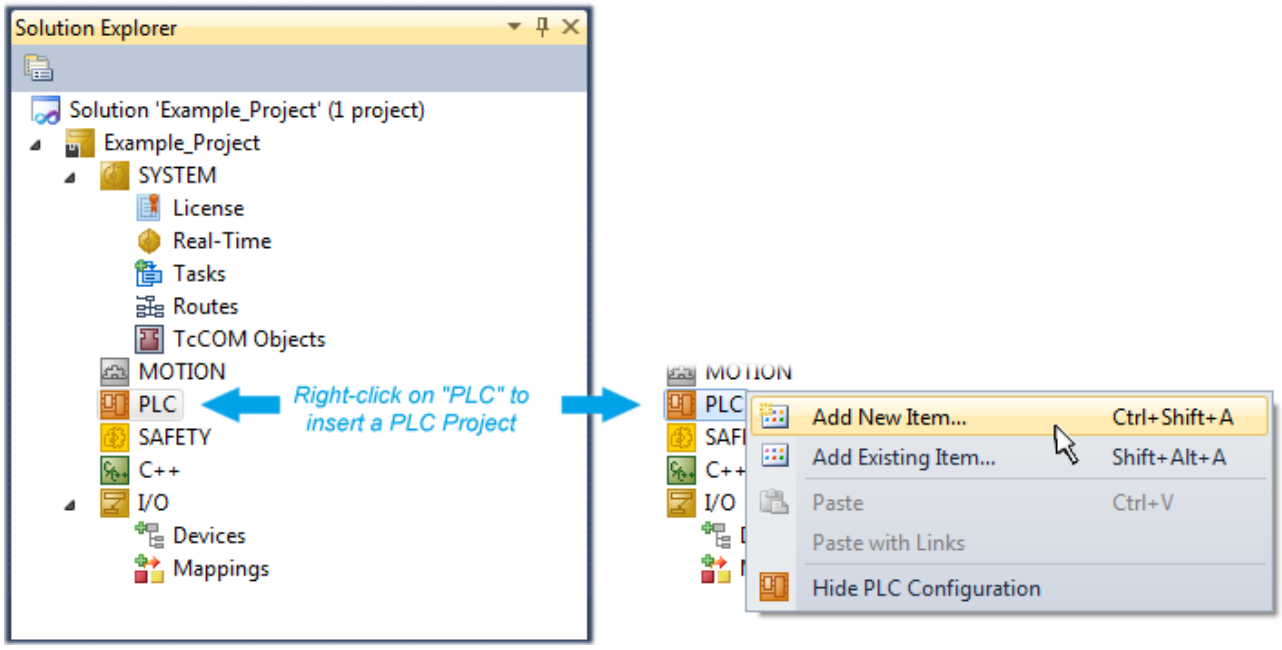


Fig. 65: Adding the programming environment in “PLC”

In the dialog that opens, select “Standard PLC project” and enter “PLC_example” as project name, for example, and select a corresponding directory:

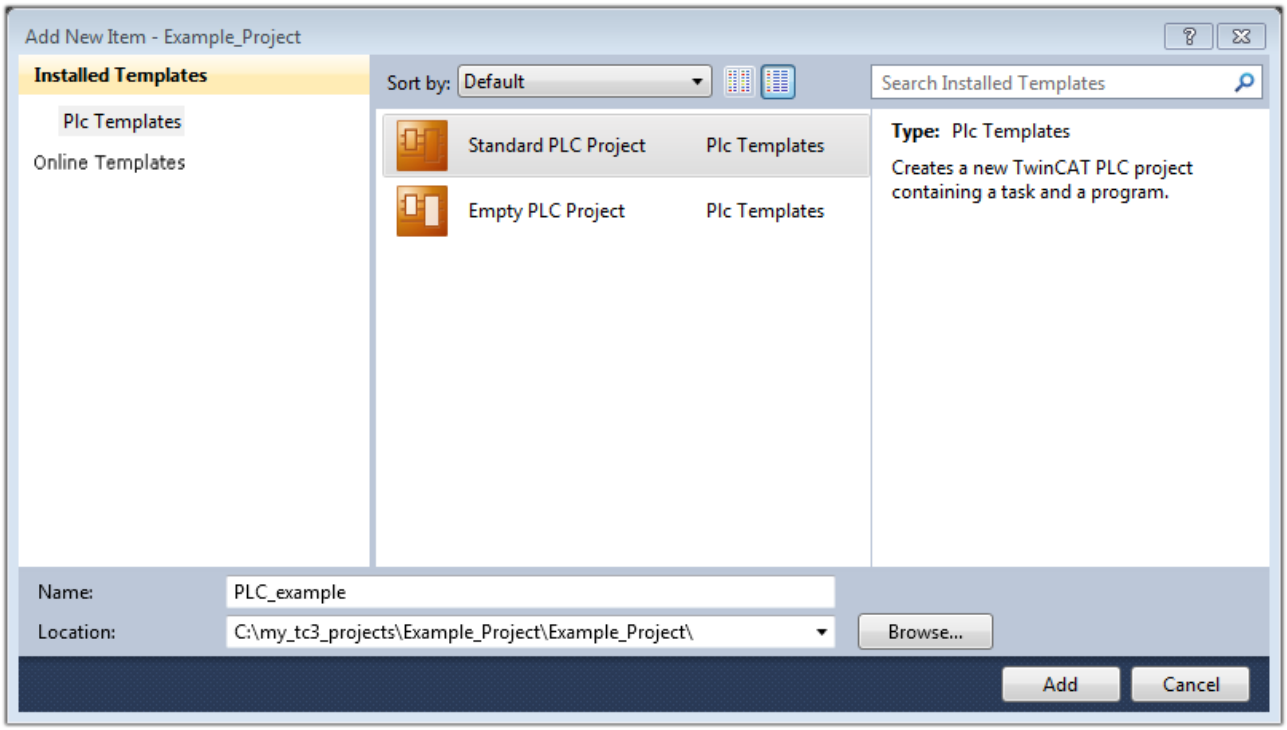


Fig. 66: Specifying the name and directory for the PLC programming environment

The “Main” program, which already exists due to selecting “Standard PLC project”, can be opened by double-clicking on “PLC_example_project” in “POUs”. The following user interface is shown for an initial project:

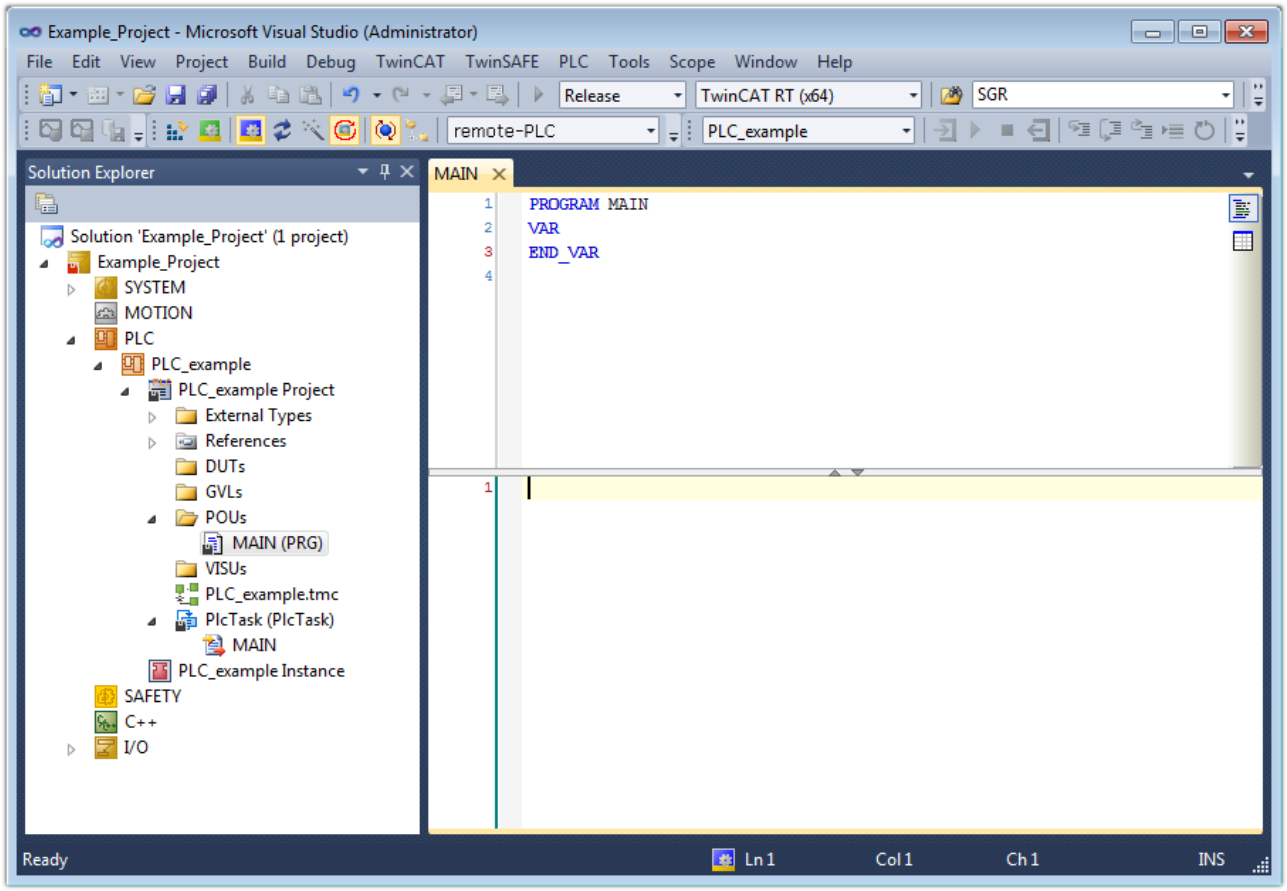


Fig. 67: Initial “Main” program for the standard PLC project

Now example variables and an example program have been created for the next stage of the process:

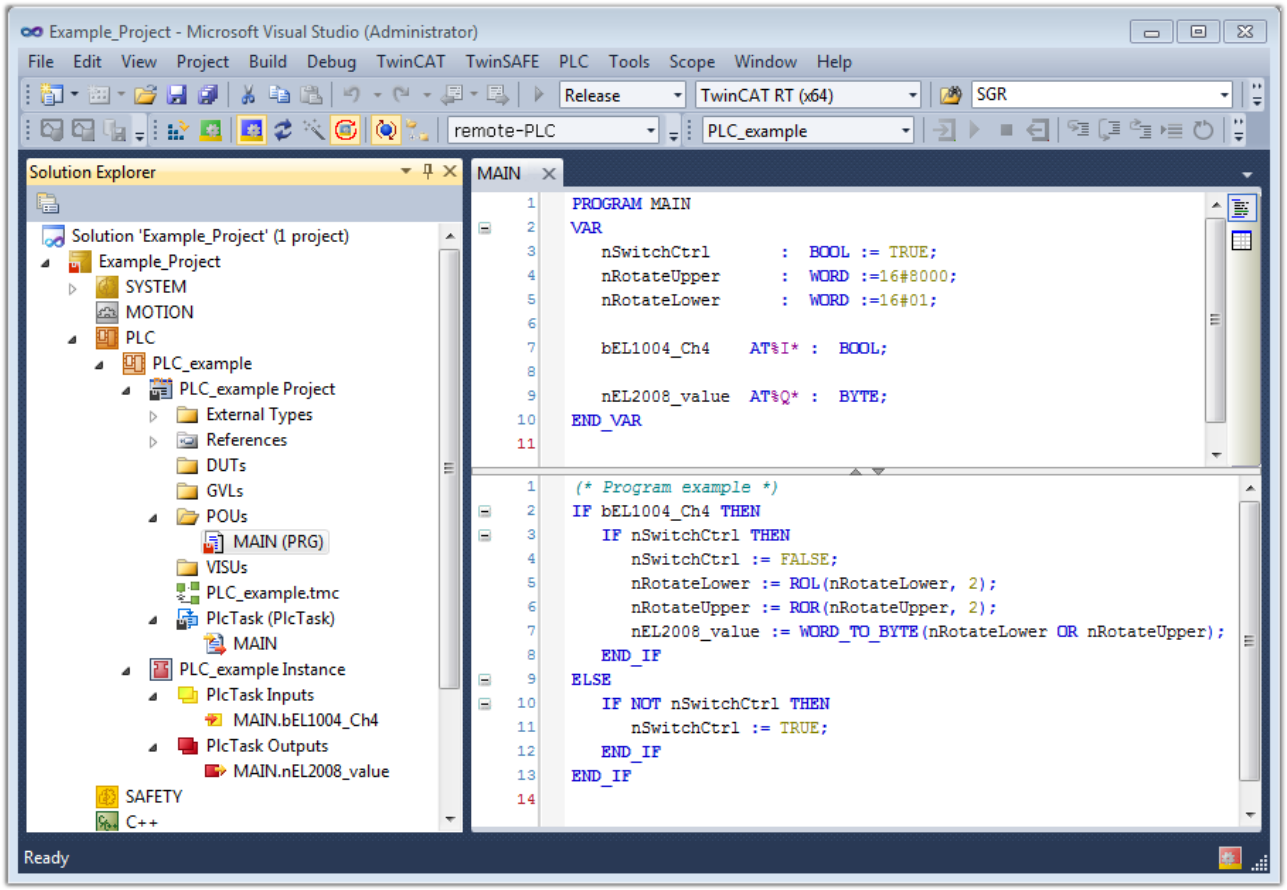


Fig. 68: Example program with variables after a compile process (without variable integration)

The control program is now created as a project folder, followed by the compile process:

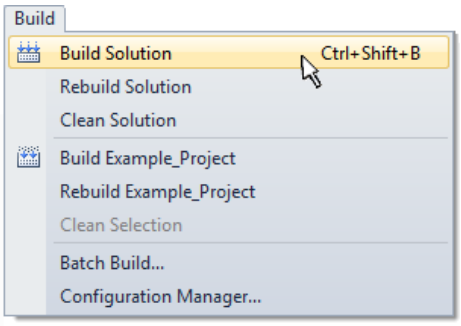
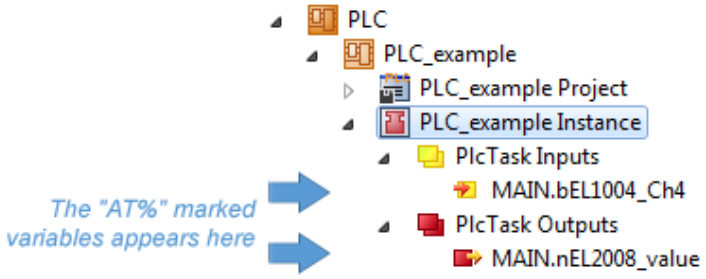


Fig. 69: Start program compilation

The following variables, identified in the ST/PLC program with “AT%”, are then available under “Assignments” in the project folder explorer:



Assigning variables

Via the menu of an instance – variables in the “PLC” context, use the “Modify Link…” option to open a window to select a suitable process object (PDO) for linking:

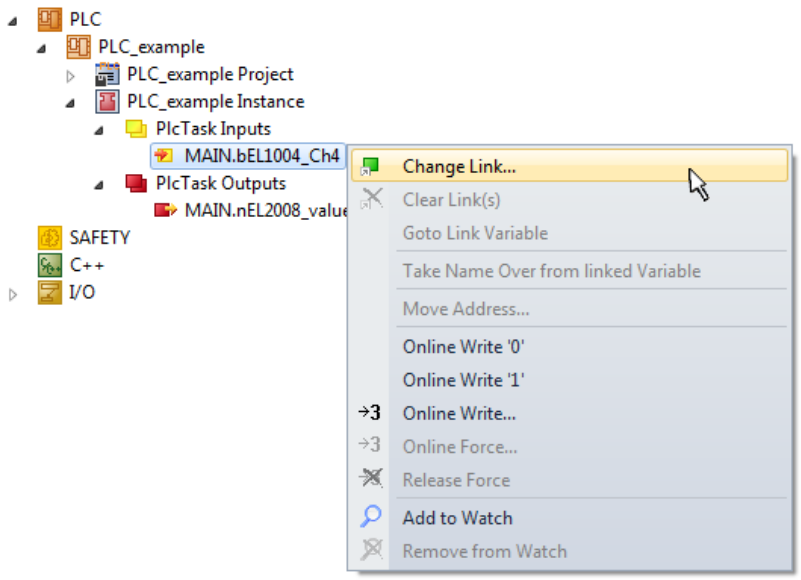


Fig. 70: Creating the links between PLC variables and process objects

In the window that opens, the process object for the “bEL1004_Ch4” BOOL-type variable can be selected from the PLC configuration tree:

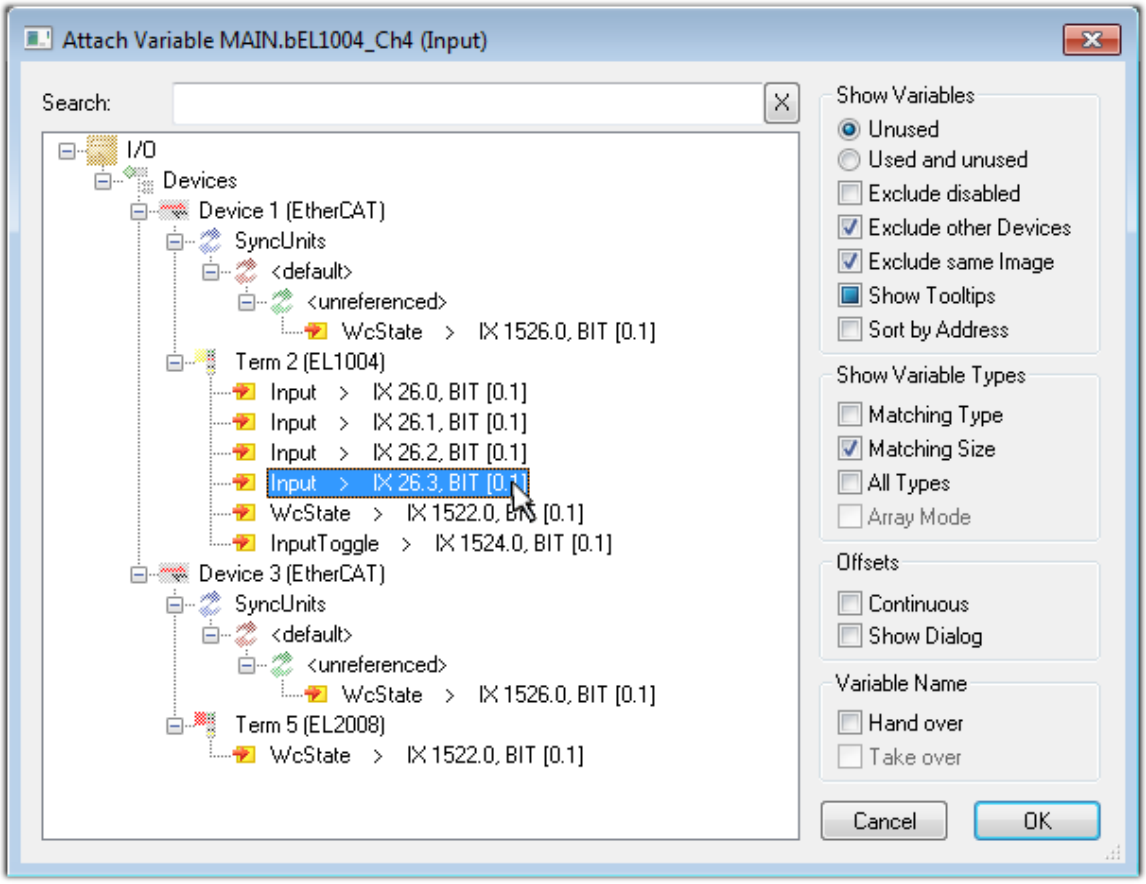


Fig. 71: Selecting BOOL-type PDO

According to the default setting, only certain PDO objects are now available for selection. In this example, the input of channel 4 of the EL1004 terminal is selected for linking. In contrast, the checkbox “All types” must be ticked to create the link for the output variables, in order to allocate a set of eight separate output bits to a byte variable in this case. The following diagram shows the whole process:

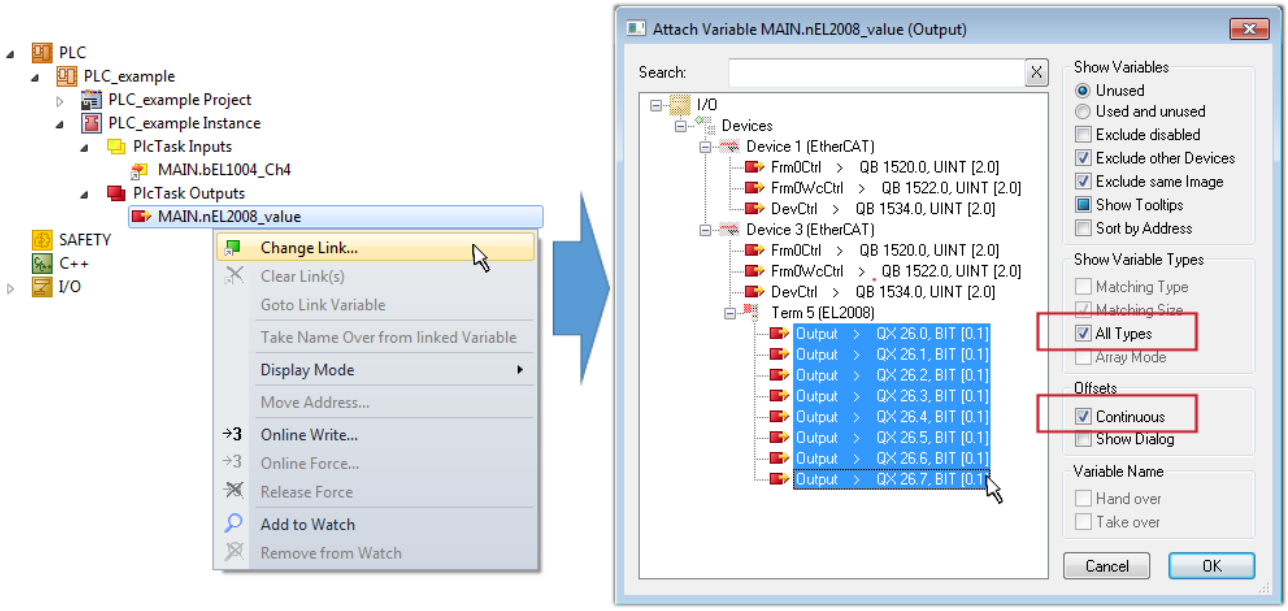


Fig. 72: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol ( ) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

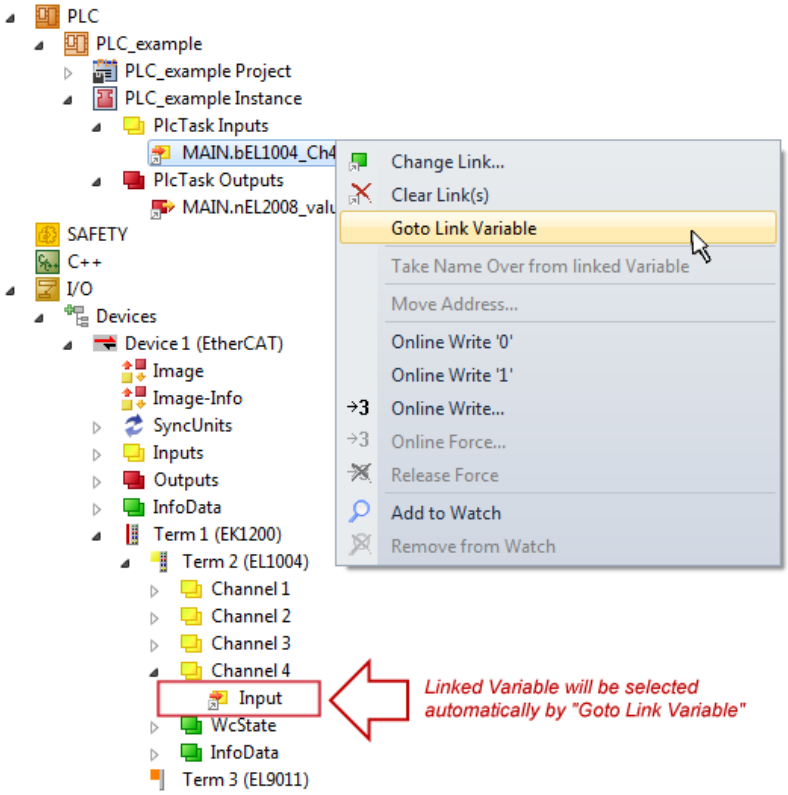


Fig. 73: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word,

integer or similar PDO, it is also possible to allocate this to a set of bit-standardized variables. Here, too, a “Goto Link Variable” can be executed in the other direction, so that the respective PLC instance can then be selected.

Note on type of variable assignment

The following type of variable assignment can only be used from TwinCAT version V3.1.4024.4 onwards and is only available for terminals with a microcontroller.

In TwinCAT, a structure can be created from the mapped process data of a terminal. An instance of this structure can then be created in the PLC, so it is possible to access the process data directly from the PLC without having to declare own variables.

The procedure for the EL3001 1-channel analog input terminal -10...+10 V is shown as an example.

1. First, the required process data must be selected in the “Process data” tab in TwinCAT.
2. After that, the PLC data type must be generated in the “PLC” tab via the check box.
3. The data type in the “Data Type” field can then be copied using the “Copy” button.

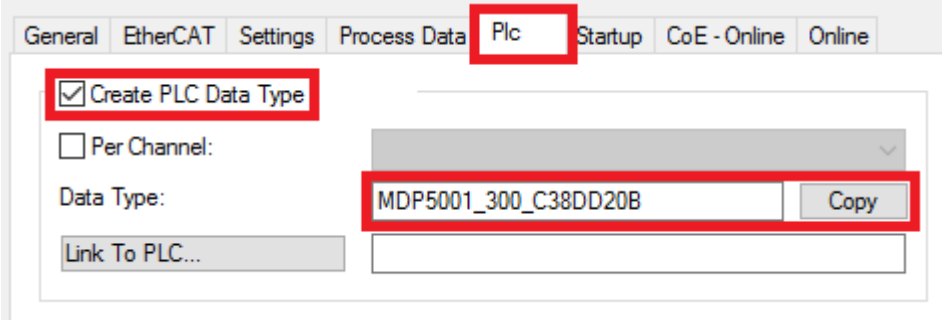


Fig. 74: Creating a PLC data type

4. An instance of the data structure of the copied data type must then be created in the PLC.

```
MAIN
1  PROGRAM MAIN
2  VAR
3      EL3001  : MDP5001_300_C38DD20B;
4  END_VAR
```

Fig. 75: Instance_of_struct

5. Then the project folder must be created. This can be done either via the key combination “CTRL + Shift + B” or via the “Build” tab in TwinCAT.
6. The structure in the “PLC” tab of the terminal must then be linked to the created instance.

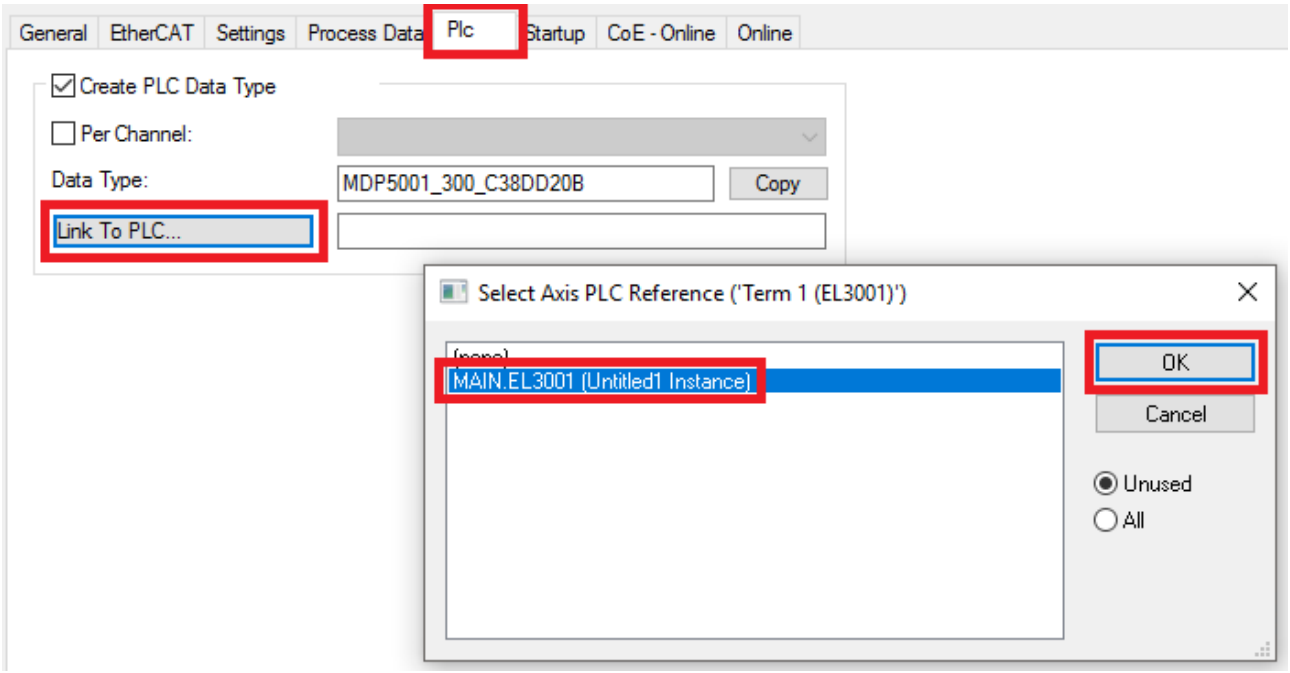


Fig. 76: Linking the structure

7. In the PLC, the process data can then be read or written via the structure in the program code.

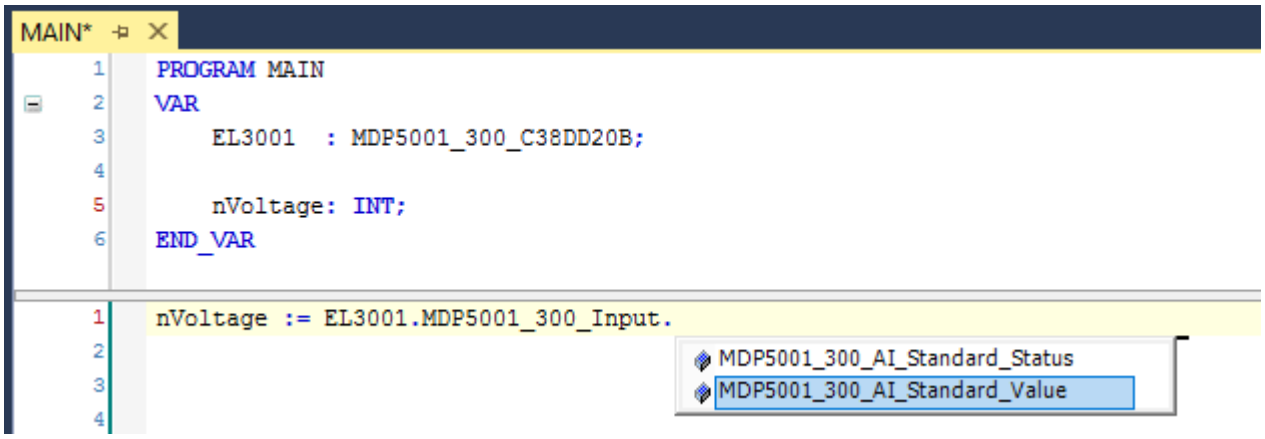
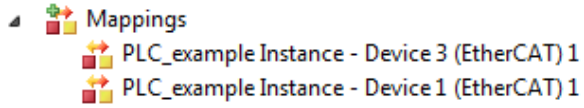


Fig. 77: Reading a variable from the structure of the process data

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated with  or via the menu under “TwinCAT” in order to transfer the settings of the development environment to the runtime system. Confirm the messages “Old configurations will be overwritten!” and “Restart TwinCAT system in Run mode” with “OK”. The corresponding assignments can be seen in the project folder explorer:



A few seconds later, the corresponding status of the Run mode is displayed in the form of a rotating symbol  at the bottom right of the VS shell development environment. The PLC system can then be started as described below.

Starting the controller

Select the menu option “PLC” → “Login” or click on  to link the PLC with the real-time system and load the control program for execution. This results in the message ”*No program on the controller! Should the new program be loaded?*”, which should be acknowledged with “Yes”. The runtime environment is ready for the program to be started by clicking on symbol , the “F5” key or via “PLC” in the menu, by selecting “Start”. The started programming environment shows the runtime values of individual variables:

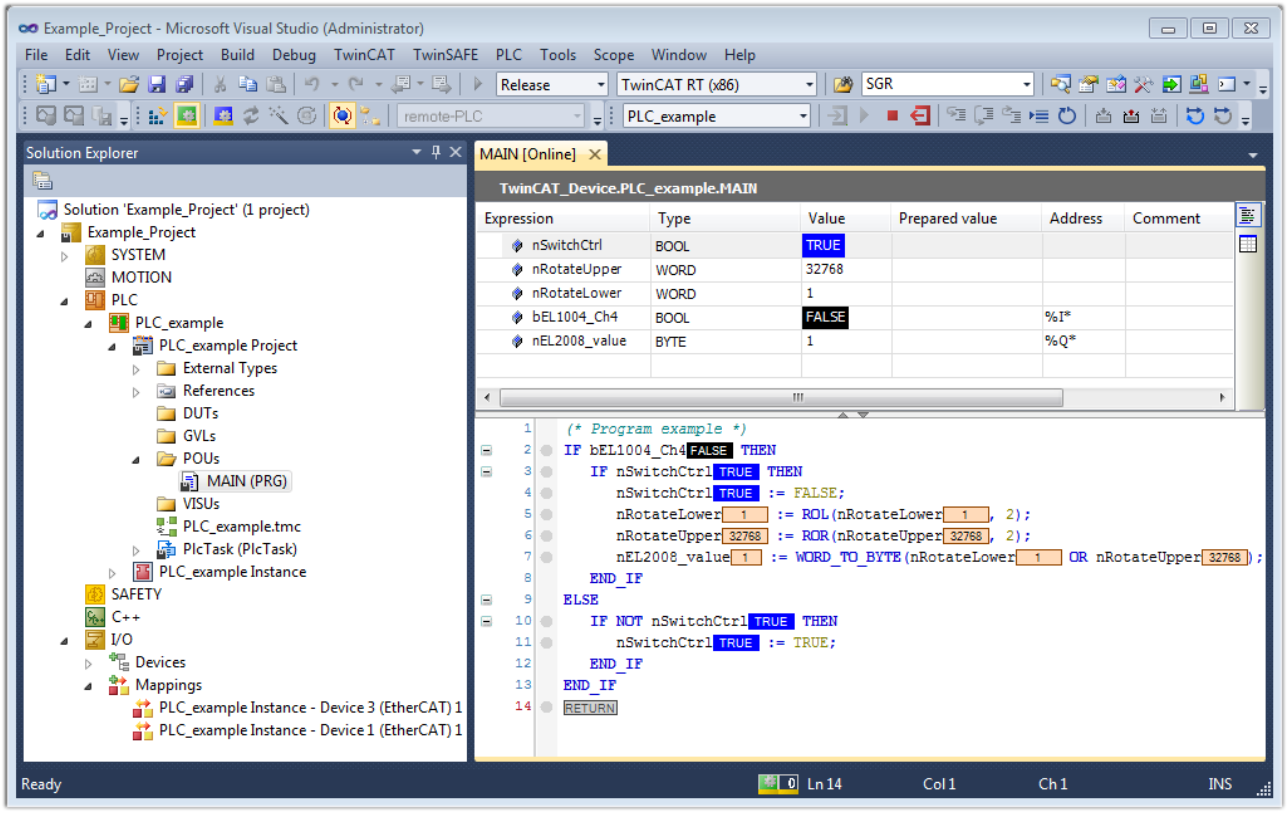


Fig. 78: TwinCAT 3 development environment (VS shell): logged-in, after program startup

The two operator control elements for stopping  and logout  result in the required action (also, “Shift + F5” can be used for stop, or both actions can be selected via the PLC menu).

7.2 TwinCAT Development Environment

The Software for automation TwinCAT (The Windows Control and Automation Technology) will be distinguished into:

- TwinCAT 2: System Manager (Configuration) & PLC Control (Programming)
- TwinCAT 3: Enhancement of TwinCAT 2 (Programming and Configuration takes place via a common Development Environment)

Details:

- **TwinCAT 2:**
  - Connects I/O devices to tasks in a variable-oriented manner
  - Connects tasks to tasks in a variable-oriented manner
  - Supports units at the bit level
  - Supports synchronous or asynchronous relationships
  - Exchange of consistent data areas and process images

- Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)
- Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/ 2000/XP/Vista, Windows 7, NT/XP Embedded, CE
- Interconnection to all common fieldbusses
- More...

Additional features:

- **TwinCAT 3** (eXtended Automation)**:**
  - Visual Studio® integration
  - Choice of the programming language
  - Supports object orientated extension of IEC 61131-3
  - Usage of C/C++ as programming language for real time applications
  - Connection to MATLAB®/Simulink®
  - Open interface for expandability
  - Flexible run-time environment
  - Active support of multi-core- and 64 bit operating system
  - Automatic code generation and project creation with the TwinCAT Automation Interface
  - More...

Within the following sections commissioning of the TwinCAT Development Environment on a PC System for the control and also the basically functions of unique control elements will be explained.

Please see further information to TwinCAT 2 and TwinCAT 3 at http://infosys.beckhoff.com.

7.2.1 Installation of the TwinCAT real-time driver

In order to assign real-time capability to a standard Ethernet port of an IPC controller, the Beckhoff real-time driver has to be installed on this port under Windows.

This can be done in several ways.

A: Via the TwinCAT Adapter dialog

In the System Manager call up the TwinCAT overview of the local network interfaces via Options → Show Real Time Ethernet Compatible Devices.

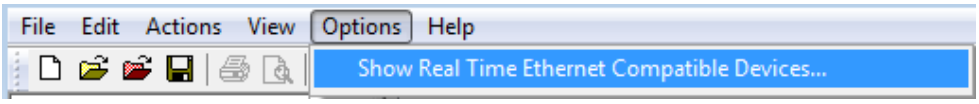


Fig. 79: System Manager “Options” (TwinCAT 2)

This have to be called up by the menu “TwinCAT” within the TwinCAT 3 environment:

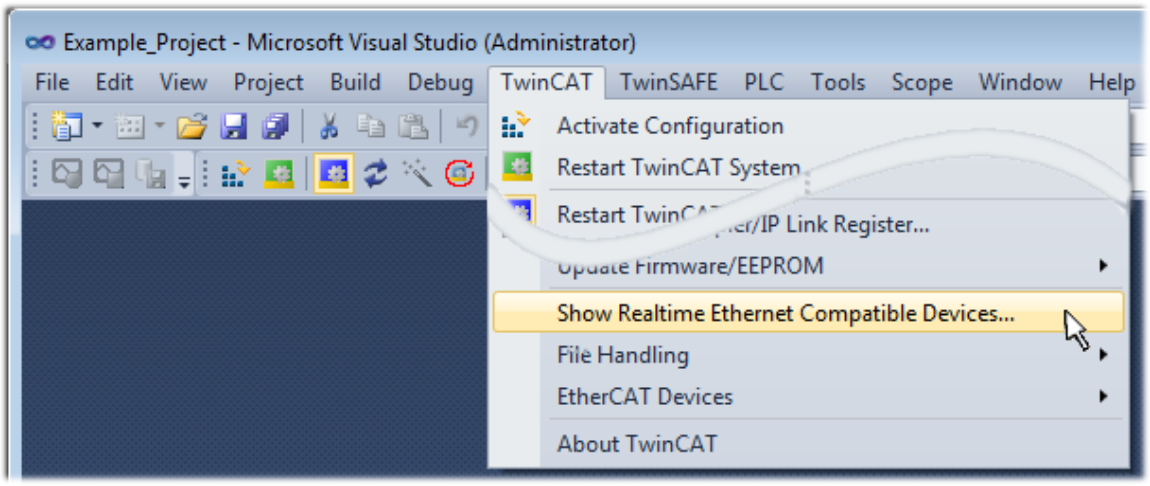


Fig. 80: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

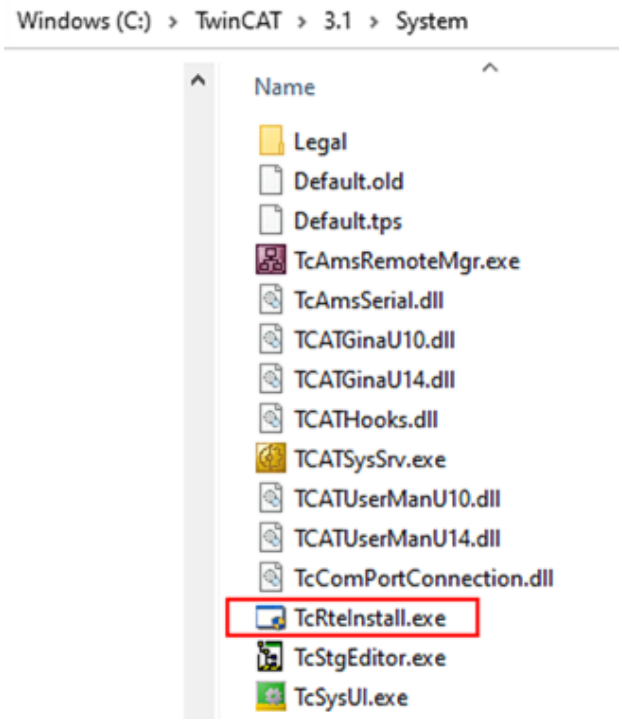


Fig. 81: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

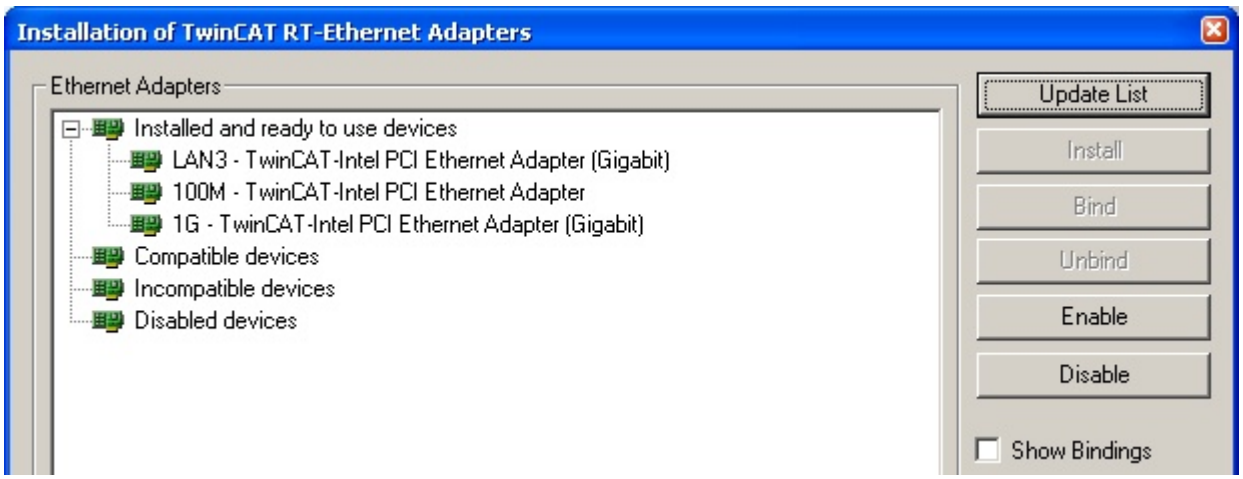


Fig. 82: Overview of network interfaces

Interfaces listed under “Compatible devices” can be assigned a driver via the “Install” button. A driver should only be installed on compatible devices.

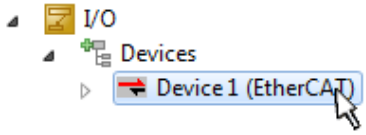
A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter Offline configuration creation, section “Creating the EtherCAT device” [▶ 101] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices…”):



Fig. 83: EtherCAT device properties (TwinCAT 2): click on “Compatible Devices…” of tab “Adapter”

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on “Device .. (EtherCAT)” within the Solution Explorer under “I/O”:



After the installation the driver appears activated in the Windows overview for the network interface (Windows Start → System Properties → Network)

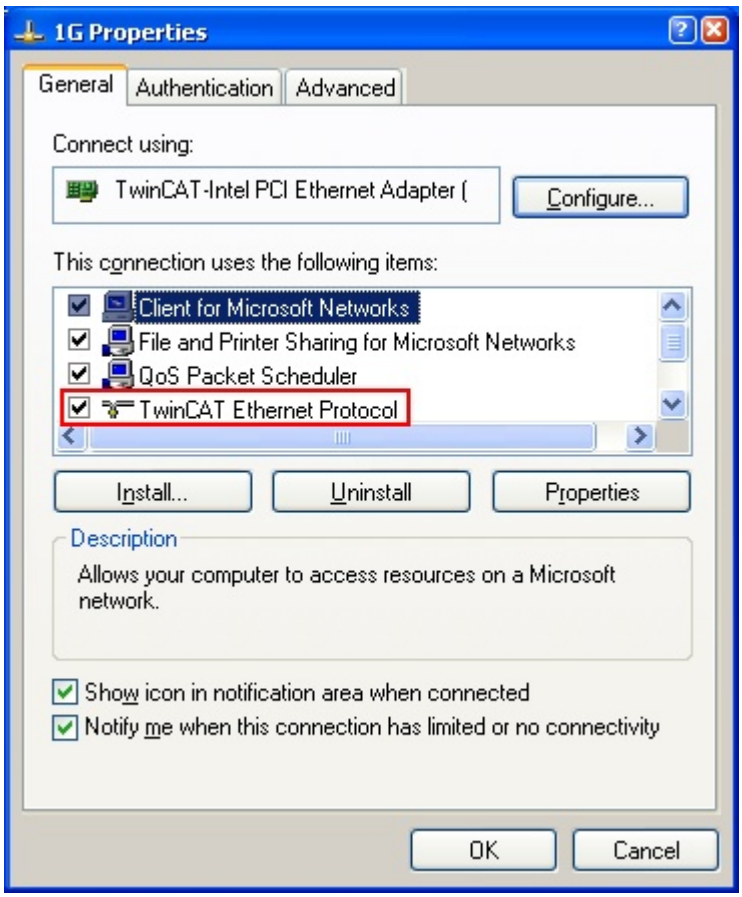


Fig. 84: Windows properties of the network interface

A correct setting of the driver could be:

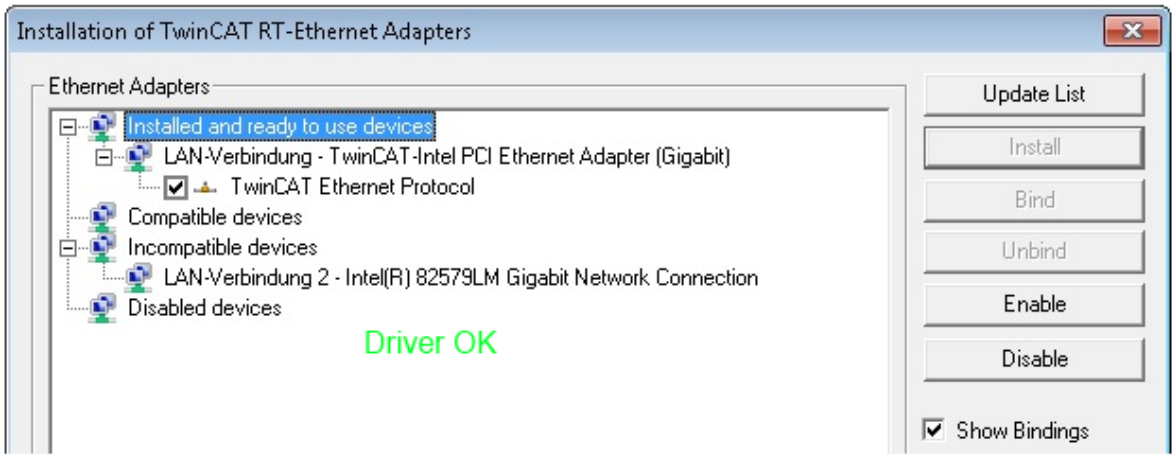


Fig. 85: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

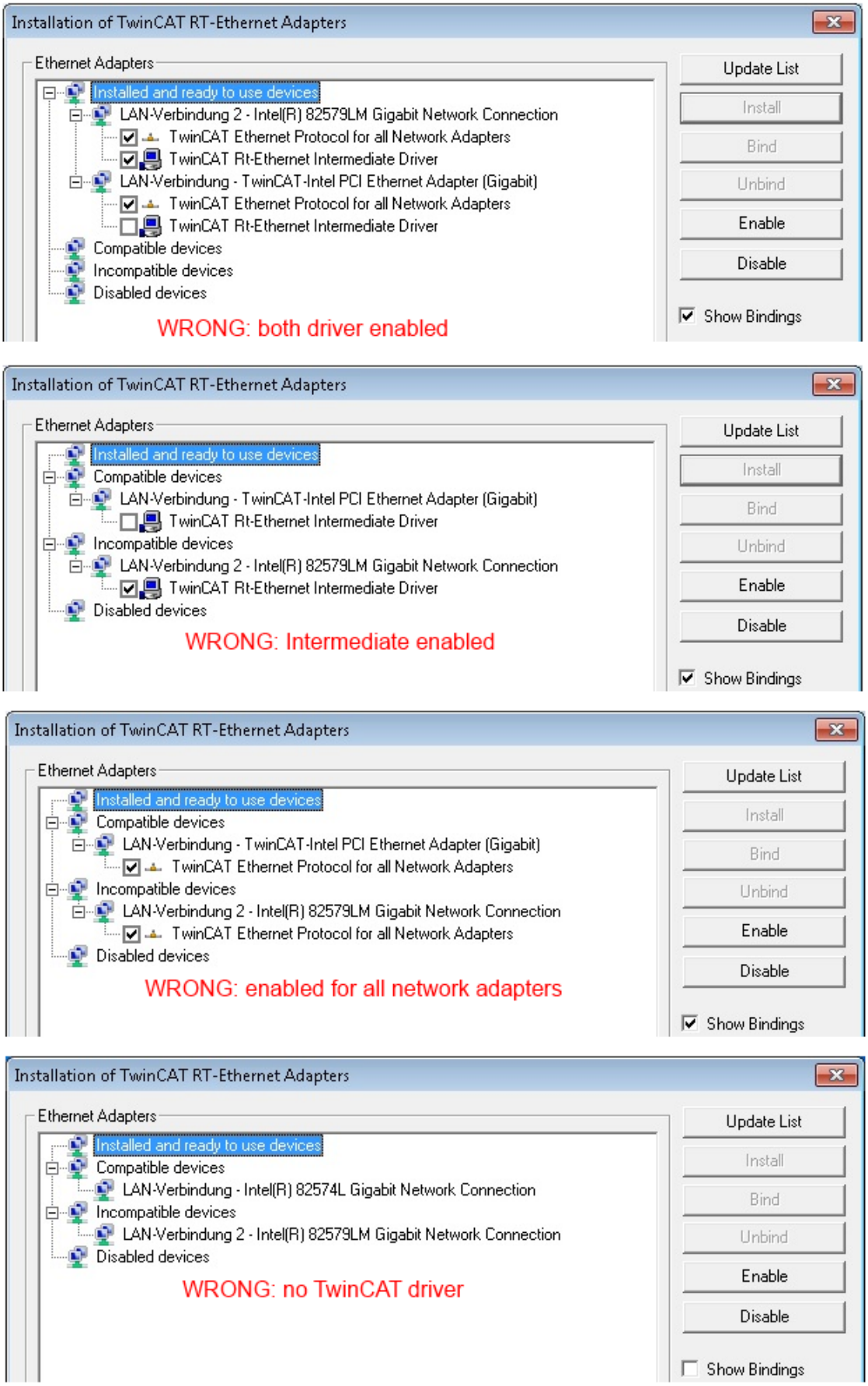


Fig. 86: Incorrect driver settings for the Ethernet port

IP address of the port used

IP address/DHCP

In most cases an Ethernet port that is configured as an EtherCAT device will not transport general IP packets. For this reason and in cases where an EL6601 or similar devices are used it is useful to specify a fixed IP address for this port via the “Internet Protocol TCP/IP” driver setting and to disable DHCP. In this way the delay associated with the DHCP client for the Ethernet port assigning itself a default IP address in the absence of a DHCP server is avoided. A suitable address space is 192.168.x.x, for example.

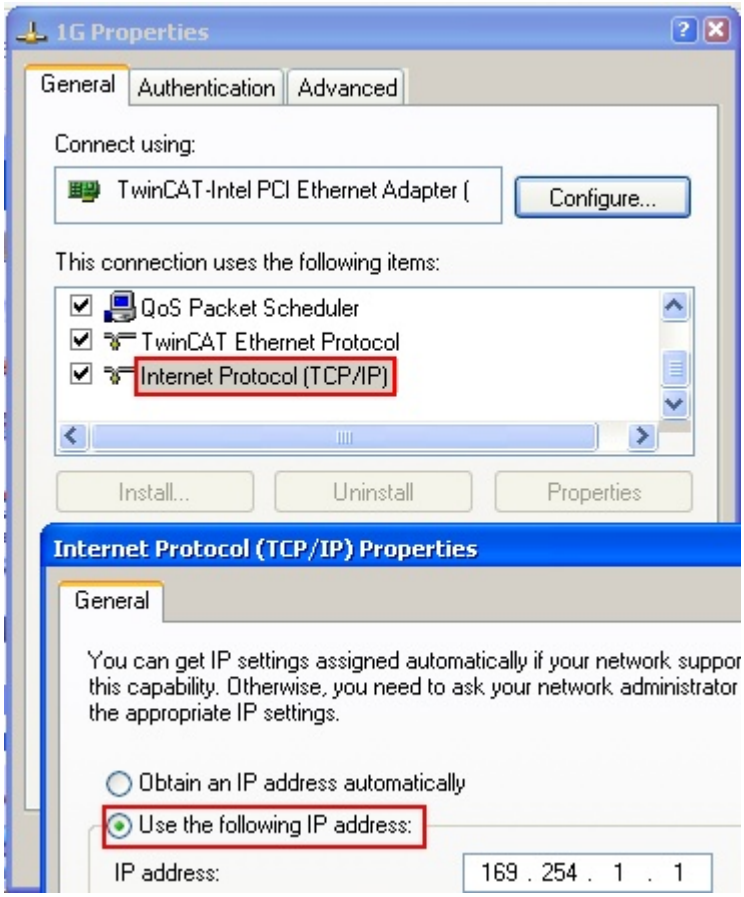


Fig. 87: TCP/IP setting for the Ethernet port

7.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

The TwinCAT EtherCAT master/System Manager needs the device description files for the devices to be used in order to generate the configuration in online or offline mode. The device descriptions are contained in the so-called ESI files (EtherCAT Slave Information) in XML format. These files can be requested from the respective manufacturer and are made available for download. An *.xml file may contain several device descriptions.

The ESI files for Beckhoff EtherCAT devices are available on the Beckhoff website.

The ESI files should be stored in the TwinCAT installation directory.

Default settings:

- **TwinCAT 2**: C:\TwinCAT\IO\EtherCAT
- **TwinCAT 3**: C:\TwinCAT\3.1\Config\Io\EtherCAT

The files are read (once) when a new System Manager window is opened, if they have changed since the last time the System Manager window was opened.

A TwinCAT installation includes the set of Beckhoff ESI files that was current at the time when the TwinCAT build was created.

For TwinCAT 2.11/TwinCAT 3 and higher, the ESI directory can be updated from the System Manager, if the programming PC is connected to the Internet; by

- **TwinCAT 2**: Option → “Update EtherCAT Device Descriptions”
- **TwinCAT 3**: TwinCAT → EtherCAT Devices → “Update Device Descriptions (via ETG Website)…”

The TwinCAT ESI Updater [▶ 100] is available for this purpose.

ESI

The *.xml files are associated with *.xsd files, which describe the structure of the ESI XML files. To update the ESI device descriptions, both file types should therefore be updated.

Device differentiation

EtherCAT devices/slaves are distinguished by four properties, which determine the full device identifier. For example, the device identifier EL2521-0025-1018 consists of:

- family key “EL”
- name “2521”
- type “0025”
- and revision “1018”

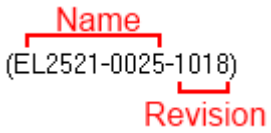


Fig. 88: Identifier structure

The order identifier consisting of name + type (here: EL2521-0025) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further notes [▶ 11].

Online description

If the EtherCAT configuration is created online through scanning of real devices (see section Online setup) and no ESI descriptions are available for a slave (specified by name and revision) that was found, the System Manager asks whether the description stored in the device should be used. In any case, the System Manager needs this information for setting up the cyclic and acyclic communication with the slave correctly.

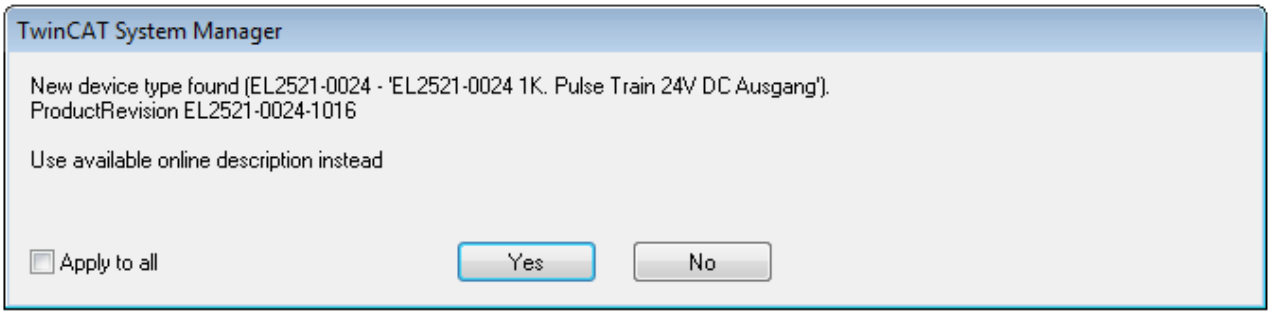


Fig. 89: OnlineDescription information window (TwinCAT 2)

In TwinCAT 3 a similar window appears, which also offers the Web update:

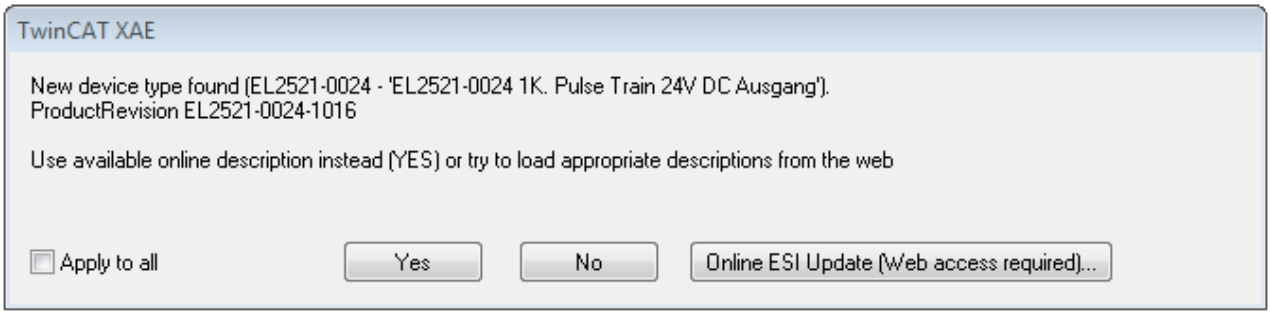


Fig. 90: Information window OnlineDescription (TwinCAT 3)

If possible, the *Yes* is to be rejected and the required ESI is to be requested from the device manufacturer. After installation of the XML/XSD file the configuration process should be repeated.

NOTICE

Changing the "usual" configuration through a scan

✓ If a scan discovers a device that is not yet known to TwinCAT, distinction has to be made between two cases. Taking the example here of the EL2521-0000 in the revision 1019

a) no ESI is present for the EL2521-0000 device at all, either for the revision 1019 or for an older revision. The ESI must then be requested from the manufacturer (in this case Beckhoff).

b) an ESI is present for the EL2521-0000 device, but only in an older revision, e.g. 1018 or 1017. In this case an in-house check should first be performed to determine whether the spare parts stock allows the integration of the increased revision into the configuration at all. A new/higher revision usually also brings along new features. If these are not to be used, work can continue without reservations with the previous revision 1018 in the configuration. This is also stated by the Beckhoff compatibility rule.

Refer in particular to the chapter "General notes on the use of Beckhoff EtherCAT IO components" and for manual configuration to the chapter "Offline configuration creation [▶ 101]".

If the OnlineDescription is used regardless, the System Manager reads a copy of the device description from the EEPROM in the EtherCAT slave. In complex slaves the size of the EEPROM may not be sufficient for the complete ESI, in which case the ESI would be *incomplete* in the configurator. Therefore it's recommended using an offline ESI file with priority in such a case.

The System Manager creates for online recorded device descriptions a new file "OnlineDescription0000...xml" in its ESI directory, which contains all ESI descriptions that were read online.

OnlineDescriptionCache00000002.xml

Fig. 91: File OnlineDescription.xml created by the System Manager

Is a slave desired to be added manually to the configuration at a later stage, online created slaves are indicated by a prepended symbol “>” in the selection list (see Figure *Indication of an online recorded ESI of EL2521 as an example)*.

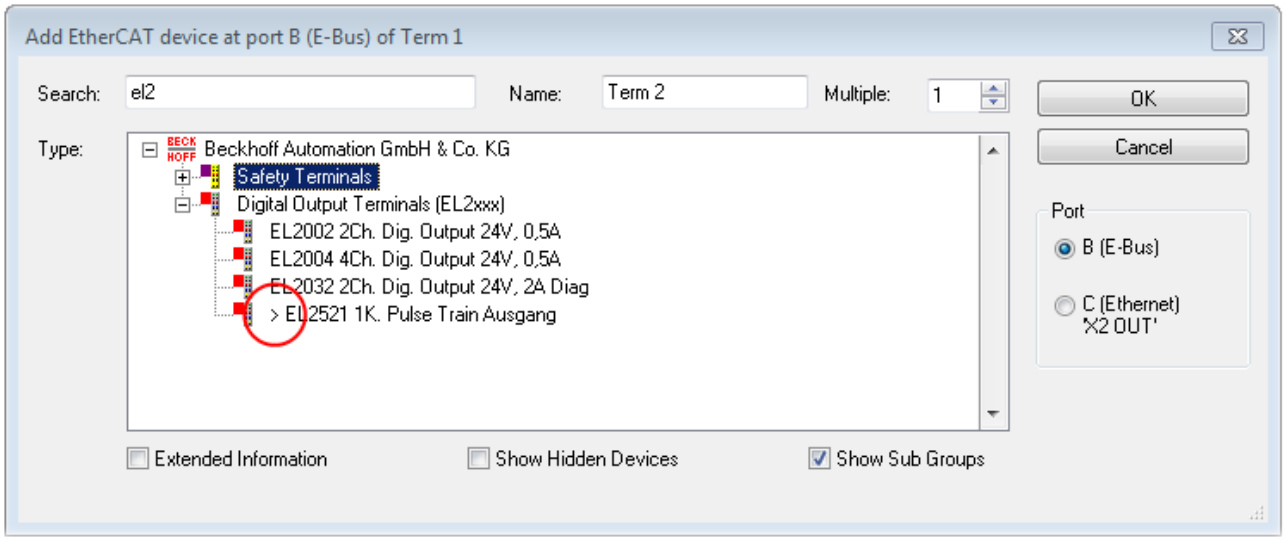


Fig. 92: Indication of an online recorded ESI of EL2521 as an example

If such ESI files are used and the manufacturer's files become available later, the file OnlineDescription.xml should be deleted as follows:

- close all System Manager windows
- restart TwinCAT in Config mode
- delete “OnlineDescription0000...xml”
- restart TwinCAT System Manager

This file should not be visible after this procedure, if necessary press <F5> to update

OnlineDescription for TwinCAT 3.x

In addition to the file described above “OnlineDescription0000...xml”, a so called EtherCAT cache with new discovered devices is created by TwinCAT 3.x, e.g. under Windows 7:

C:\User\[USERNAME]\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml

(Please note the language settings of the OS!)
You have to delete this file, too.

Faulty ESI file

If an ESI file is faulty and the System Manager is unable to read it, the System Manager brings up an information window.

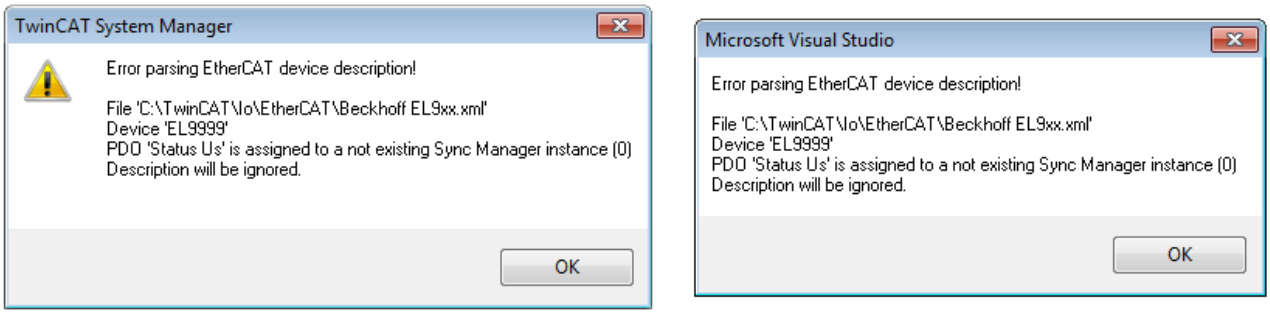


Fig. 93: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

- Structure of the *.xml does not correspond to the associated *.xsd file → check your schematics
- Contents cannot be translated into a device description → contact the file manufacturer

7.2.3 TwinCAT ESI Updater

For TwinCAT 2.11 and higher, the System Manager can search for current Beckhoff ESI files automatically, if an online connection is available:

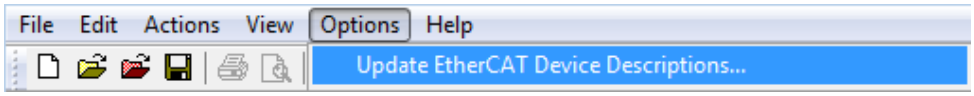


Fig. 94: Using the ESI Updater (>= TwinCAT 2.11)

The call up takes place under:
“Options” → “Update EtherCAT Device Descriptions”

Selection under TwinCAT 3:

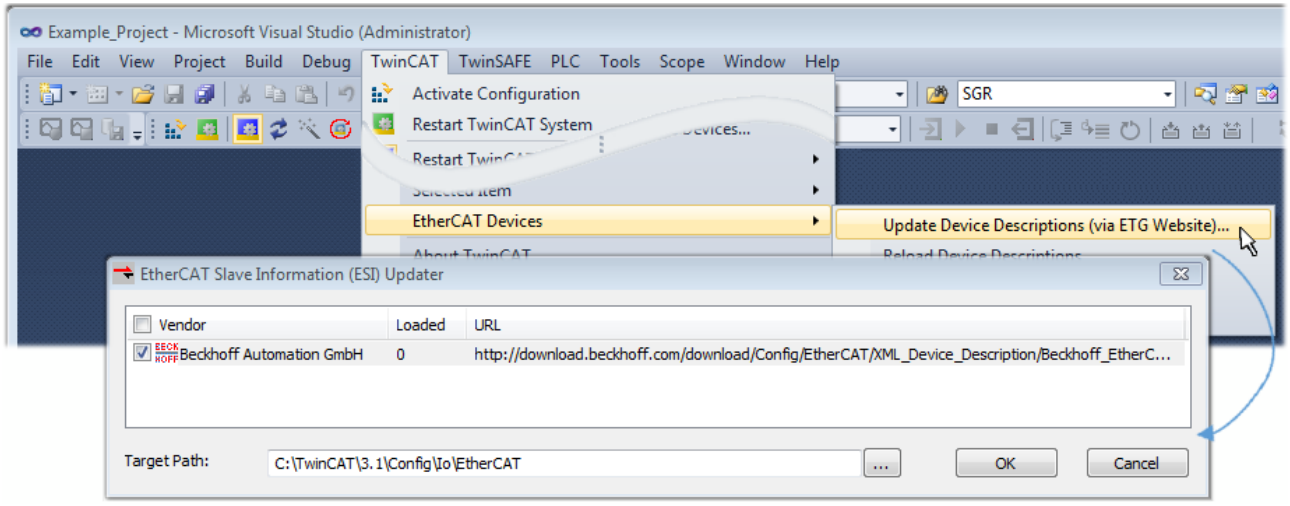


Fig. 95: Using the ESI Updater (TwinCAT 3)

The ESI Updater (TwinCAT 3) is a convenient option for automatic downloading of ESI data provided by EtherCAT manufacturers via the Internet into the TwinCAT directory (ESI = EtherCAT slave information). TwinCAT accesses the central ESI ULR directory list stored at ETG; the entries can then be viewed in the Updater dialog, although they cannot be changed there.

The call up takes place under:
“TwinCAT” → “EtherCAT Devices” → “Update Device Description (via ETG Website)…”.

7.2.4 Distinction between Online and Offline

The distinction between online and offline refers to the presence of the actual I/O environment (drives, terminals, EJ-modules). If the configuration is to be prepared in advance of the system configuration as a programming system, e.g. on a laptop, this is only possible in “Offline configuration” mode. In this case all components have to be entered manually in the configuration, e.g. based on the electrical design.

If the designed control system is already connected to the EtherCAT system and all components are energised and the infrastructure is ready for operation, the TwinCAT configuration can simply be generated through “scanning” from the runtime system. This is referred to as online configuration.

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note “Installation of the latest ESI-XML device description” [▶ 96].

For preparation of a configuration:

- the real EtherCAT hardware (devices, couplers, drives) must be present and installed
- the devices/modules must be connected via EtherCAT cables or in the terminal/ module strand in the same way as they are intended to be used later
- the devices/modules be connected to the power supply and ready for communication

- TwinCAT must be in CONFIG mode on the target system.

The online scan process consists of:

- detecting the EtherCAT device [▶ 106] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 107]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 110]

The scan with existing configuration [▶ 111] can also be carried out for comparison.

7.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

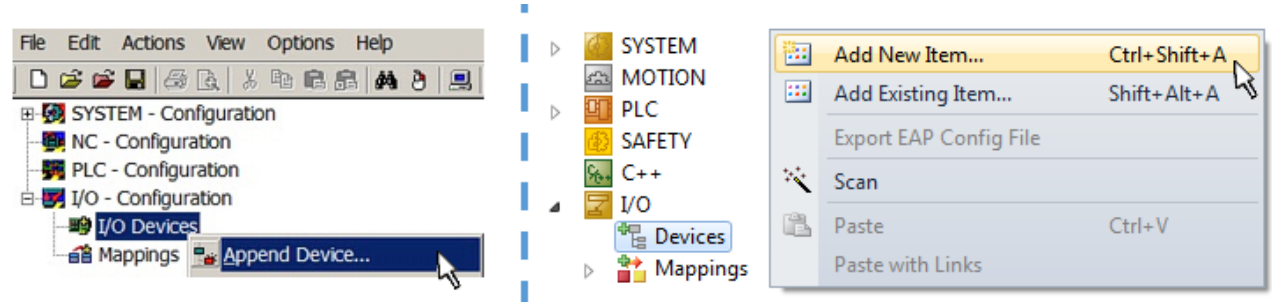


Fig. 96: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

Select type “EtherCAT” for an EtherCAT I/O application with EtherCAT slaves. For the present publisher/subscriber service in combination with an EL6601/EL6614 terminal select “EtherCAT Automation Protocol via EL6601”.

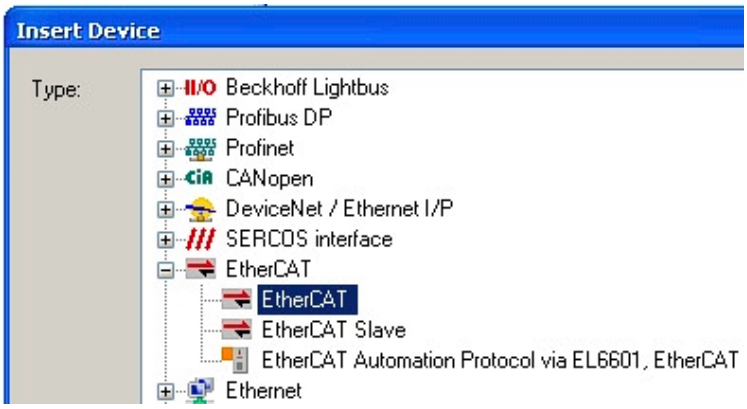


Fig. 97: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

Then assign a real Ethernet port to this virtual device in the runtime system.

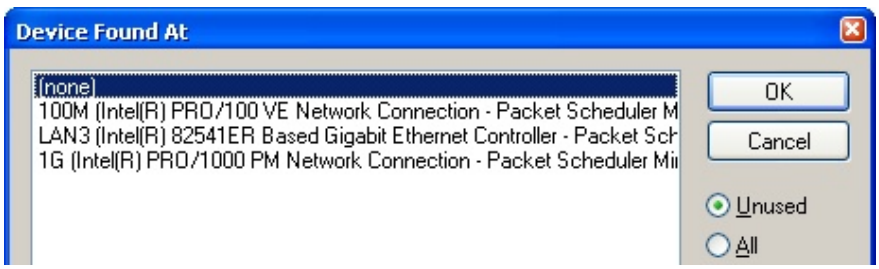


Fig. 98: Selecting the Ethernet port

This query may appear automatically when the EtherCAT device is created, or the assignment can be set/ modified later in the properties dialog; see Fig. “EtherCAT device properties (TwinCAT 2)”.

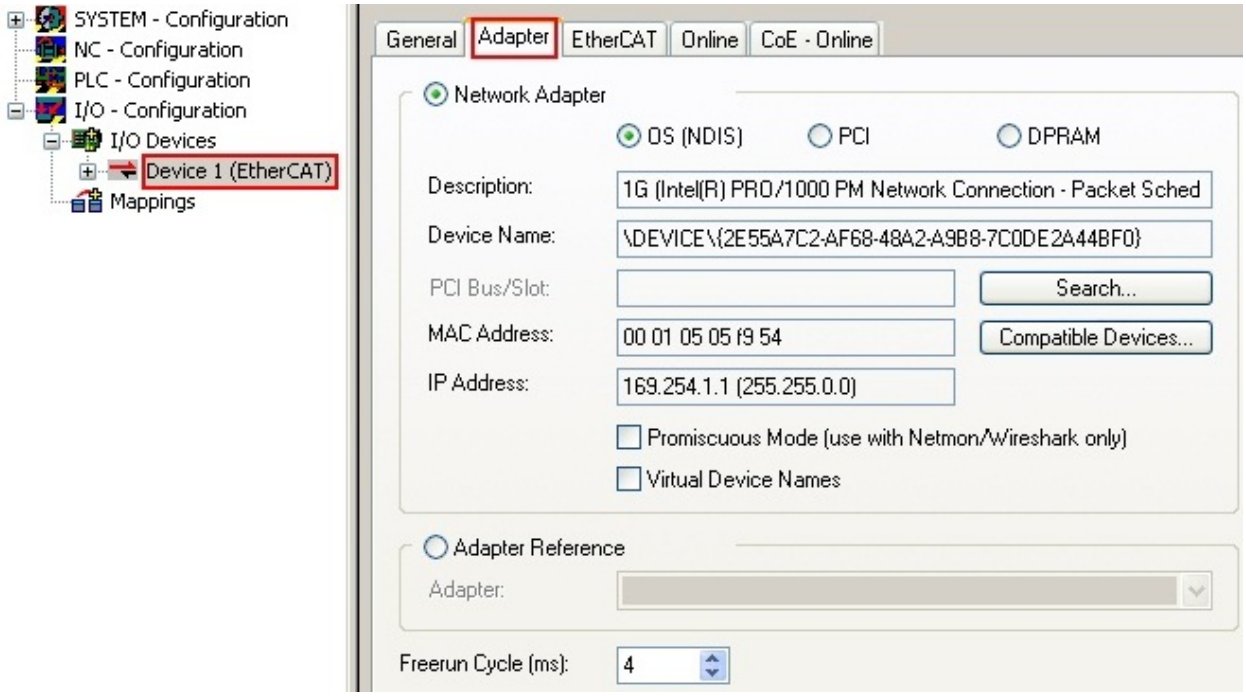
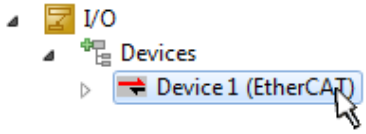


Fig. 99: EtherCAT device properties (TwinCAT 2)

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on “Device .. (EtherCAT)” within the Solution Explorer under “I/O”:



Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 90].

Defining EtherCAT slaves

Further devices can be appended by right-clicking on a device in the configuration tree.

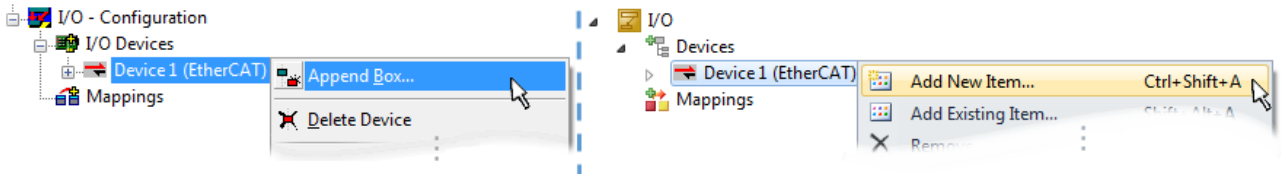


Fig. 100: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

The dialog for selecting a new device opens. Only devices for which ESI files are available are displayed.

Only devices are offered for selection that can be appended to the previously selected device. Therefore, the physical layer available for this port is also displayed (Fig. “Selection dialog for new EtherCAT device”, A). In the case of cable-based Fast-Ethernet physical layer with PHY transfer, then also only cable-based devices are available, as shown in Fig. “Selection dialog for new EtherCAT device”. If the preceding device has several free ports (e.g. EK1122 or EK1100), the required port can be selected on the right-hand side (A).

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: couplers, box modules, devices with RJ45/M8/M12 connector

- “E-Bus”: LVDS “terminal bus”, EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

The search field facilitates finding specific devices (since TwinCAT 2.11 or TwinCAT 3).

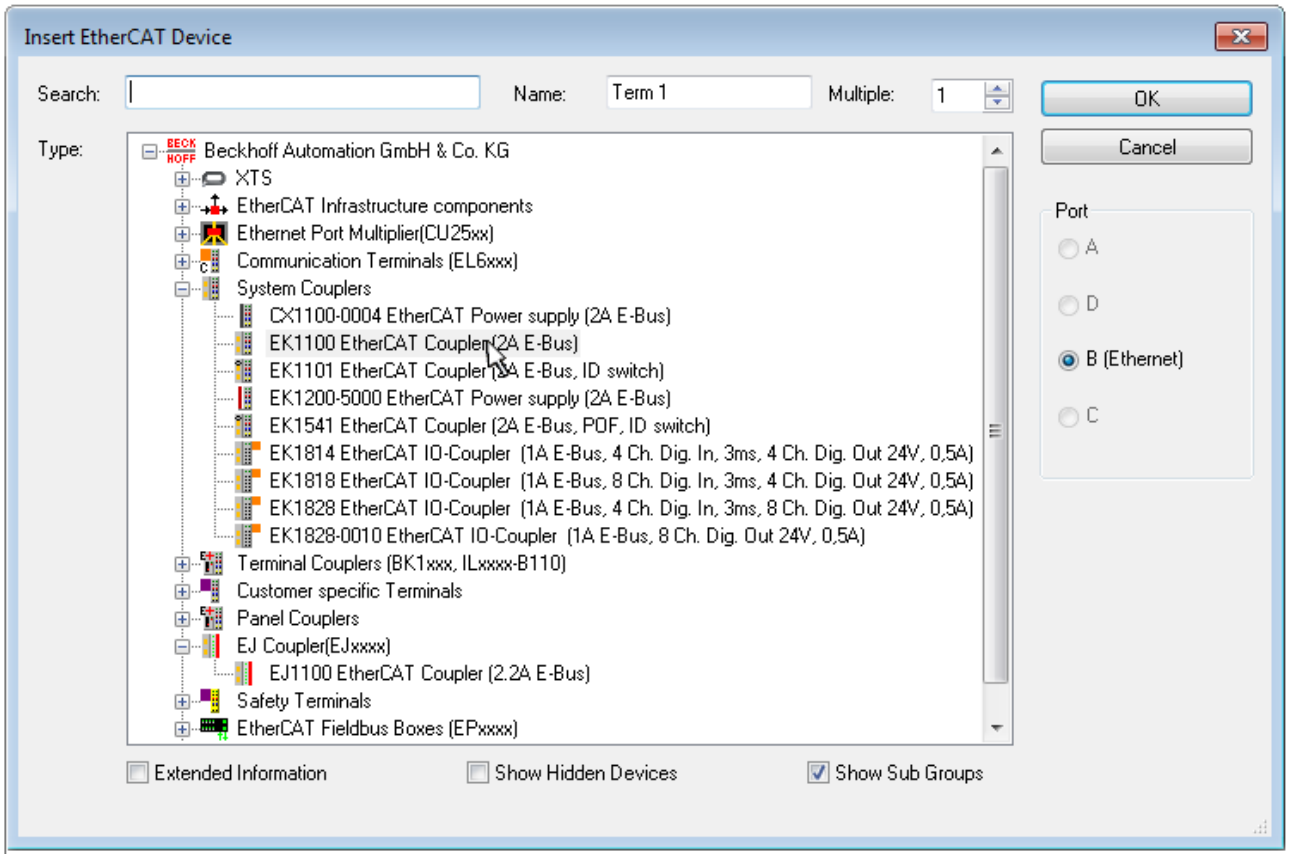


Fig. 101: Selection dialog for new EtherCAT device

By default, only the name/device type is used as selection criterion. For selecting a specific revision of the device, the revision can be displayed as “Extended Information”.

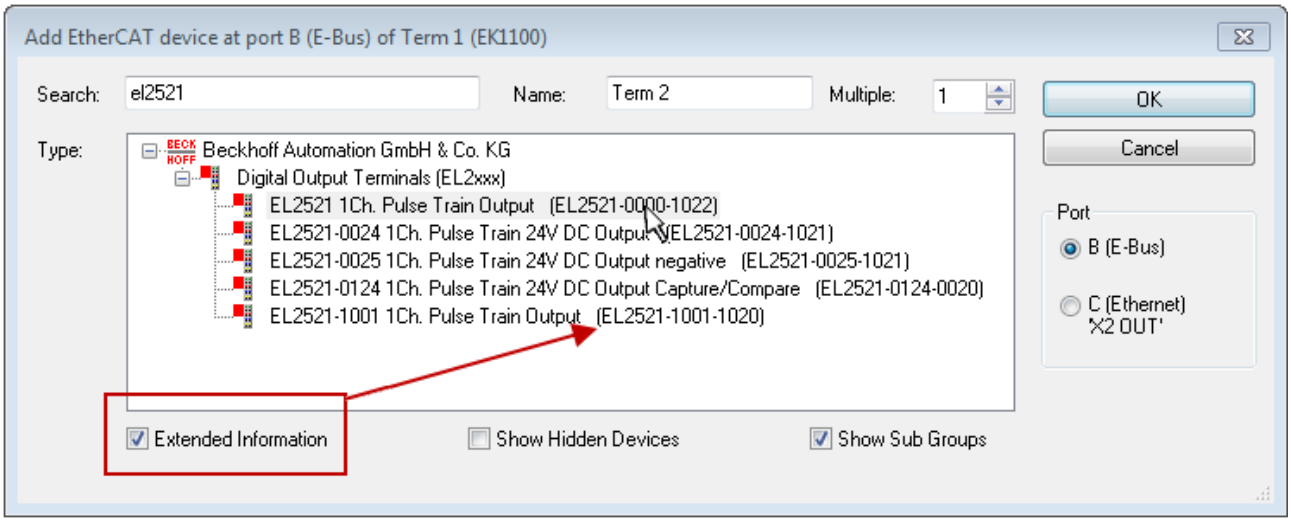


Fig. 102: Display of device revision

In many cases several device revisions were created for historic or functional reasons, e.g. through technological advancement. For simplification purposes (see Fig. “Selection dialog for new EtherCAT device”) only the last (i.e. highest) revision and therefore the latest state of production is displayed in the selection dialog for Beckhoff devices. To show all device revisions available in the system as ESI descriptions tick the “Show Hidden Devices” check box, see Fig. “Display of previous revisions”.

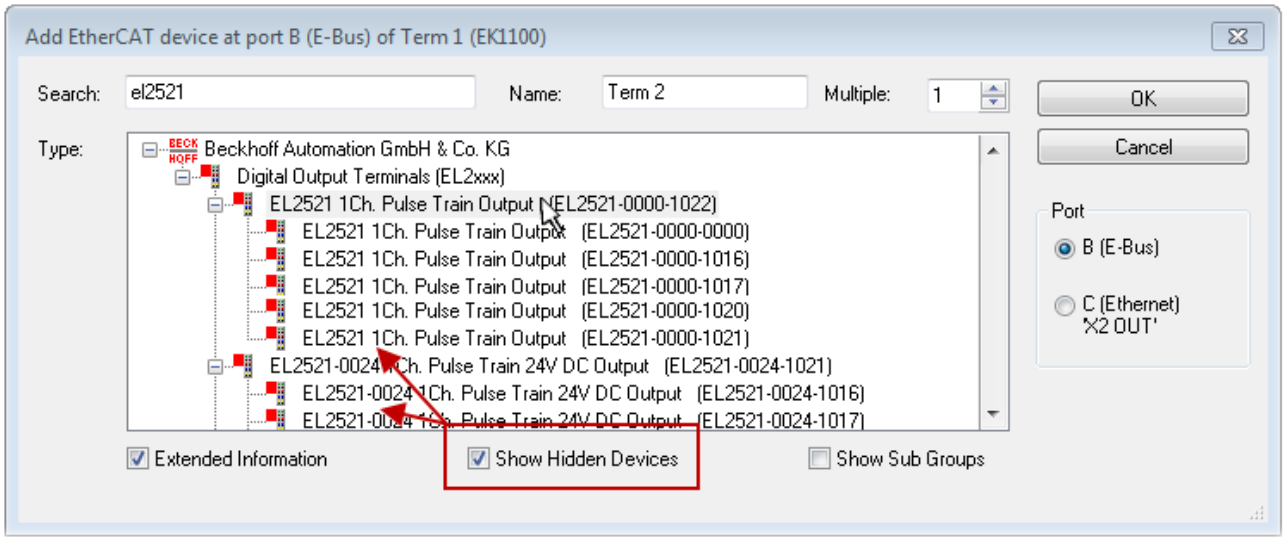


Fig. 103: Display of previous revisions

Device selection based on revision, compatibility

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The following compatibility rule of thumb is to be assumed for Beckhoff EtherCAT Terminals/ Boxes/ EJ-modules:

device revision in the system >= device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-**1018** is specified in the configuration, an EL2521-0025-**1018** or higher (-**1019**, -**1020**) can be used in practice.

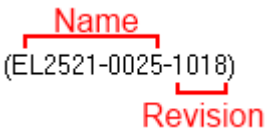


Fig. 104: Name/revision of the terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

In this case the process image of the device is shown in the configuration tree and can be parameterized as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

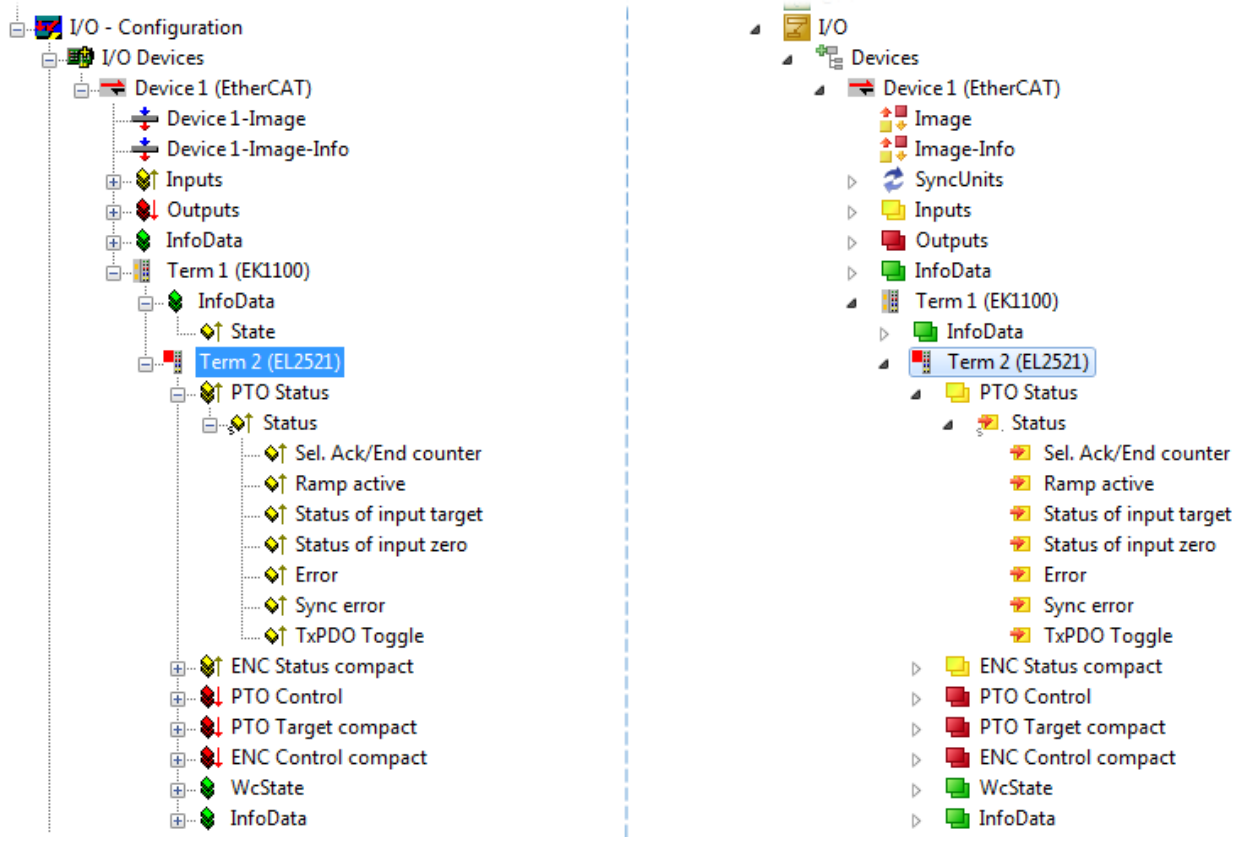


Fig. 105: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

7.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

The online device search can be used if the TwinCAT system is in CONFIG mode. This can be indicated by a symbol right below in the information bar:

- on TwinCAT 2 by a blue display “Config Mode” within the System Manager window: Config Mode .
- on TwinCAT 3 within the user interface of the development environment by a symbol .

TwinCAT can be set into this mode:

- TwinCAT 2: by selection of in the Menubar or by “Actions” → “Set/Reset TwinCAT to Config Mode…”
- TwinCAT 3: by selection of in the Menubar or by “TwinCAT” → “Restart TwinCAT (Config Mode)”

Online scanning in Config mode

The online search is not available in RUN mode (production operation). Note the differentiation between TwinCAT programming system and TwinCAT target system.

The TwinCAT 2 icon ( ) or TwinCAT 3 icon ( ) within the Windows-Taskbar always shows the TwinCAT mode of the local IPC. Compared to that, the System Manager window of TwinCAT 2 or the user interface of TwinCAT 3 indicates the state of the target system.

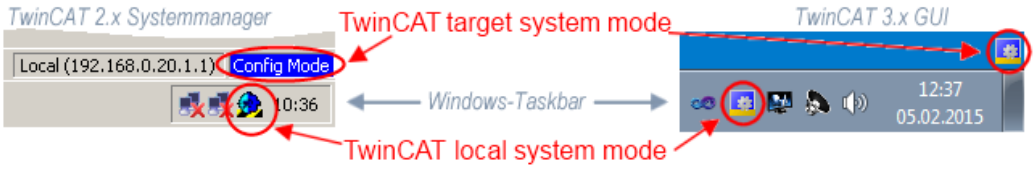


Fig. 106: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

Right-clicking on “I/O Devices” in the configuration tree opens the search dialog.

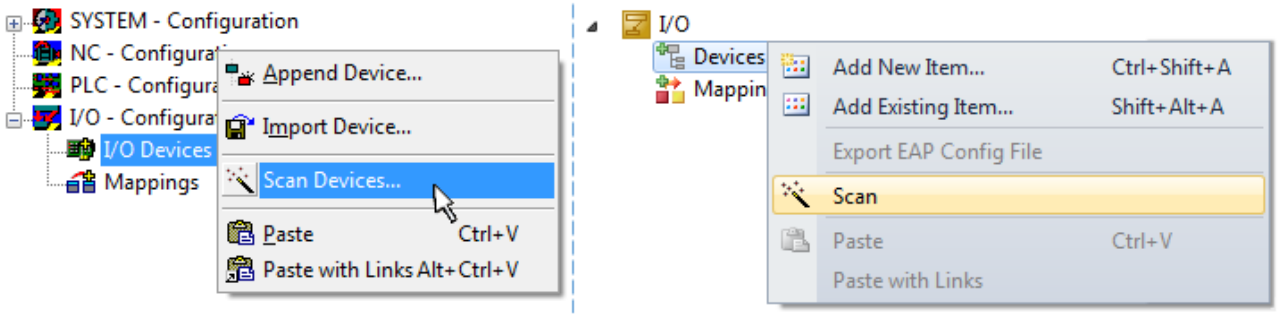


Fig. 107: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

This scan mode attempts to find not only EtherCAT devices (or Ethernet ports that are usable as such), but also NOVRAM, fieldbus cards, SMB etc. However, not all devices can be found automatically.

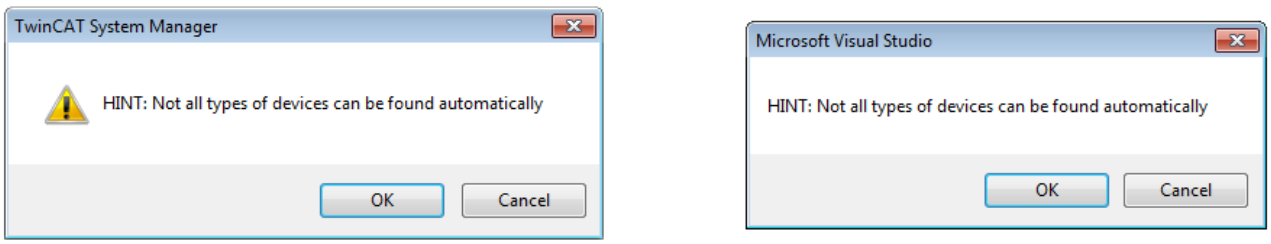


Fig. 108: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

Ethernet ports with installed TwinCAT real-time driver are shown as “RT Ethernet” devices. An EtherCAT frame is sent to these ports for testing purposes. If the scan agent detects from the response that an EtherCAT slave is connected, the port is immediately shown as an “EtherCAT Device” .

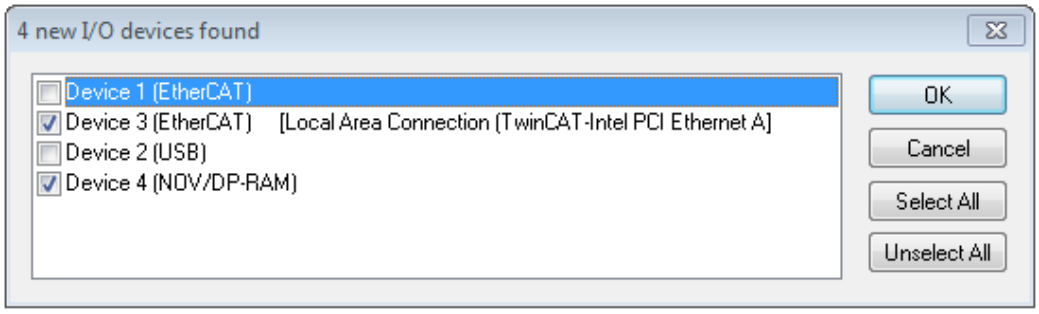


Fig. 109: Detected Ethernet devices

Via respective checkboxes devices can be selected (as illustrated in Fig. “Detected Ethernet devices” e.g. Device 3 and Device 4 were chosen). After confirmation with “OK” a device scan is suggested for all selected devices, see Fig.: “Scan query after automatic creation of an EtherCAT device”.

Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 90].

Detecting/Scanning the EtherCAT devices

Online scan functionality

During a scan the master queries the identity information of the EtherCAT slaves from the slave EEPROM. The name and revision are used for determining the type. The respective devices are located in the stored ESI data and integrated in the configuration tree in the default state defined there.

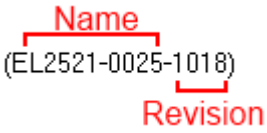


Fig. 110: Example default state

NOTICE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for comparison [▶ 111] with the defined initial configuration.Background: since Beckhoff occasionally increases the revision version of the delivered products for product maintenance reasons, a configuration can be created by such a scan which (with an identical machine construction) is identical according to the device list; however, the respective device revision may differ from the initial configuration.

Example:

Company A builds the prototype of a machine B, which is to be produced in series later on. To do this the prototype is built, a scan of the IO devices is performed in TwinCAT and the initial configuration “B.tsm” is created. The EL2521-0025 EtherCAT terminal with the revision 1018 is located somewhere. It is thus built into the TwinCAT configuration in this way:

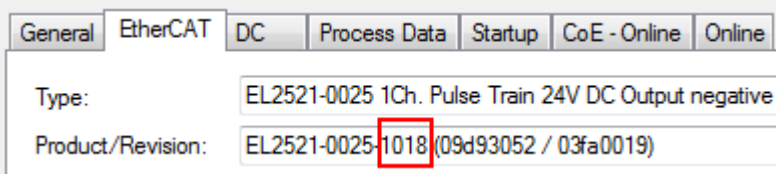


Fig. 111: Installing EthetCAT terminal with revision -1018

Likewise, during the prototype test phase, the functions and properties of this terminal are tested by the programmers/commissioning engineers and used if necessary, i.e. addressed from the PLC “B.pro” or the NC. (the same applies correspondingly to the TwinCAT 3 solution files).

The prototype development is now completed and series production of machine B starts, for which Beckhoff continues to supply the EL2521-0025-0018. If the commissioning engineers of the series machine production department always carry out a scan, a B configuration with the identical contents results again for each machine. Likewise, A might create spare parts stores worldwide for the coming series-produced machines with EL2521-0025-1018 terminals.

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and **a new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of “B.tsm” or even “B.pro” is therefore unnecessary. The series-produced machines can continue to be built with “B.tsm” and “B.pro”; it makes sense to perform a comparative scan [▶ 111] against the initial configuration “B.tsm” in order to check the built machine.

However, if the series machine production department now doesn’t use “B.tsm”, but instead carries out a scan to create the productive configuration, the revision **-1019** is automatically detected and built into the configuration:

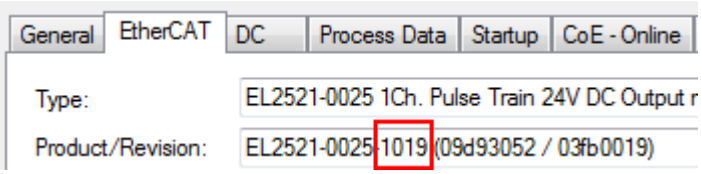


Fig. 112: Detection of EtherCAT terminal with revision -1019

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since a new configuration is essentially created. According to the compatibility rule, however, this means that no EL2521-0025-**1018** should be built into this machine as a spare part (even if this nevertheless works in the vast majority of cases).

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration “B2.tsm” created in this way. Þ if series machine production is established, the scan should only be performed for informative purposes for comparison with a defined initial configuration. Changes are to be made with care!

If an EtherCAT device was created in the configuration (manually or through a scan), the I/O field can be scanned for devices/slaves.



Fig. 113: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

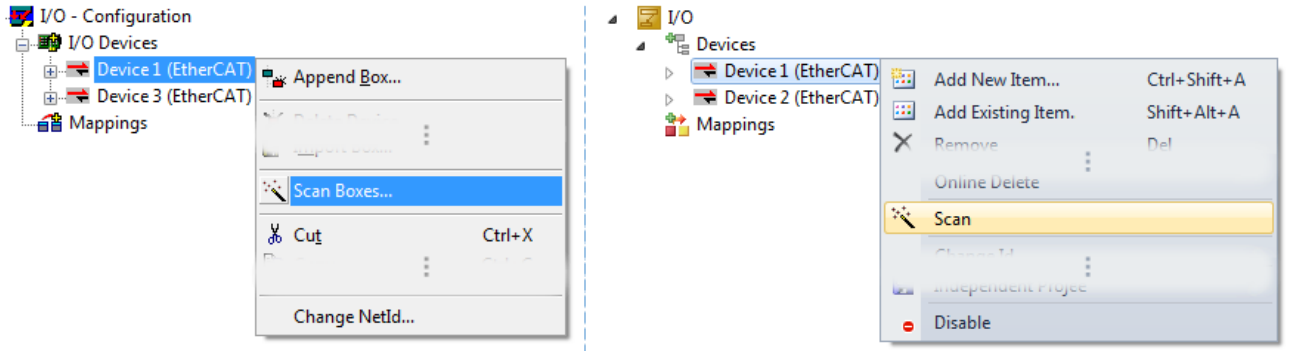


Fig. 114: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

In the System Manager (TwinCAT 2) or the User Interface (TwinCAT 3) the scan process can be monitored via the progress bar at the bottom in the status bar.

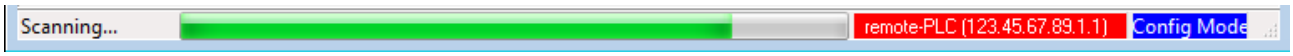


Fig. 115: *Scan progressexemplary by TwinCAT 2*

The configuration is established and can then be switched to online state (OPERATIONAL).



Fig. 116: Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)

In Config/FreeRun mode the System Manager display alternates between blue and red, and the EtherCAT device continues to operate with the idling cycle time of 4 ms (default setting), even without active task (NC, PLC).

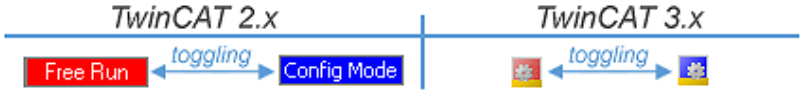


Fig. 117: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 118: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

The EtherCAT system should then be in a functional cyclic state, as shown in Fig. *Online display example*.

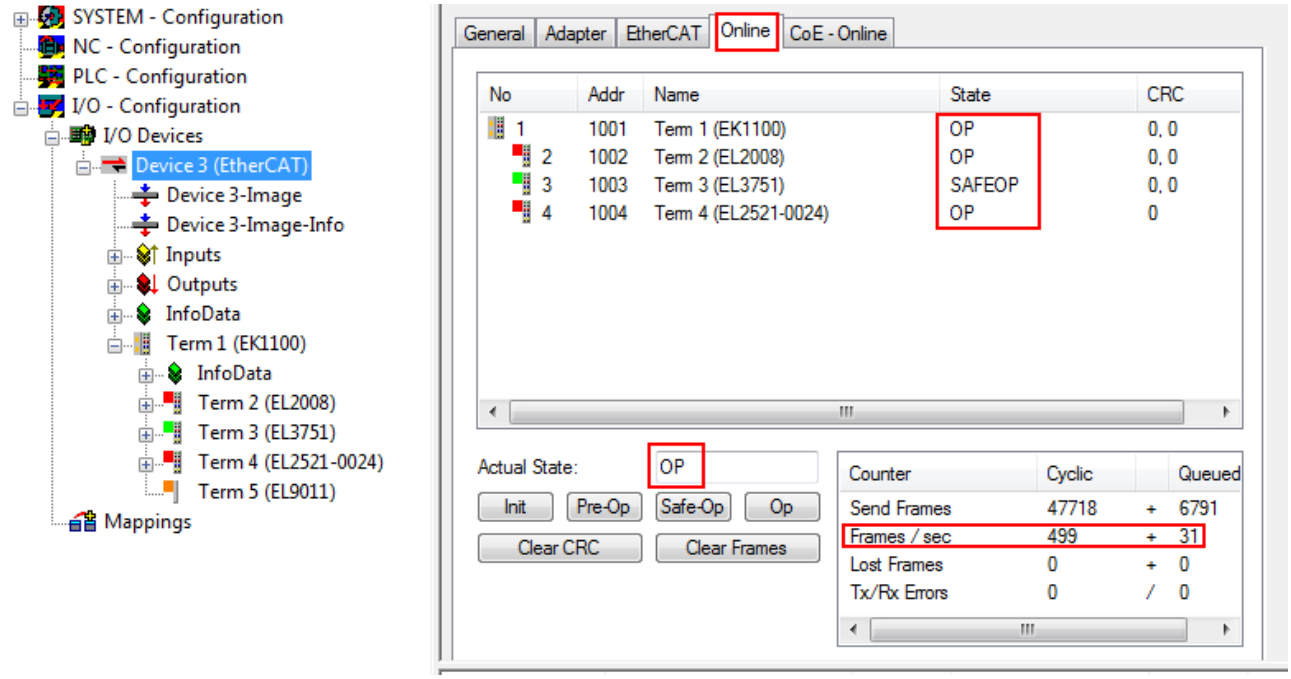


Fig. 119: Online display example

Please note:

- all slaves should be in OP state
- the EtherCAT master should be in “Actual State” OP
- “frames/sec” should match the cycle time taking into account the sent number of frames
- no excessive “LostFrames” or CRC errors should occur

The configuration is now complete. It can be modified as described under manual procedure [▶ 101].

Troubleshooting

Various effects may occur during scanning.

- An **unknown device** is detected, i.e. an EtherCAT slave for which no ESI XML description is available. In this case the System Manager offers to read any ESI that may be stored in the device. This case is described in the chapter “Notes regarding ESI device description”.
- **Device are not detected properly**
  Possible reasons include:
  - faulty data links, resulting in data loss during the scan
  - slave has invalid device description

  The connections and devices should be checked in a targeted manner, e.g. via the emergency scan. Then re-run the scan.

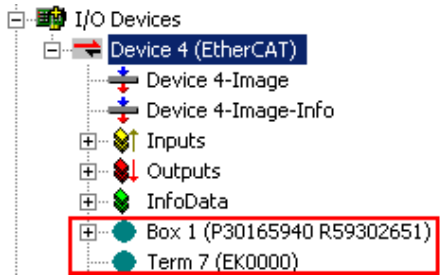


Fig. 120: Faulty identification

In the System Manager such devices may be set up as EK0000 or unknown devices. Operation is not possible or meaningful.

Scan over existing Configuration

NOTICE

Change of the configuration after comparison

With this scan (TwinCAT 2.11 or 3.1) only the device properties vendor (manufacturer), device name and revision are compared at present! A “ChangeTo” or “Copy” should only be carried out with care, taking into consideration the Beckhoff IO compatibility rule (see above). The device configuration is then replaced by the revision found; this can affect the supported process data and functions.

If a scan is initiated for an existing configuration, the actual I/O environment may match the configuration exactly or it may differ. This enables the configuration to be compared.



Fig. 121: Identical configuration (left: TwinCAT 2; right: TwinCAT 3)

If differences are detected, they are shown in the correction dialog, so that the user can modify the configuration as required.

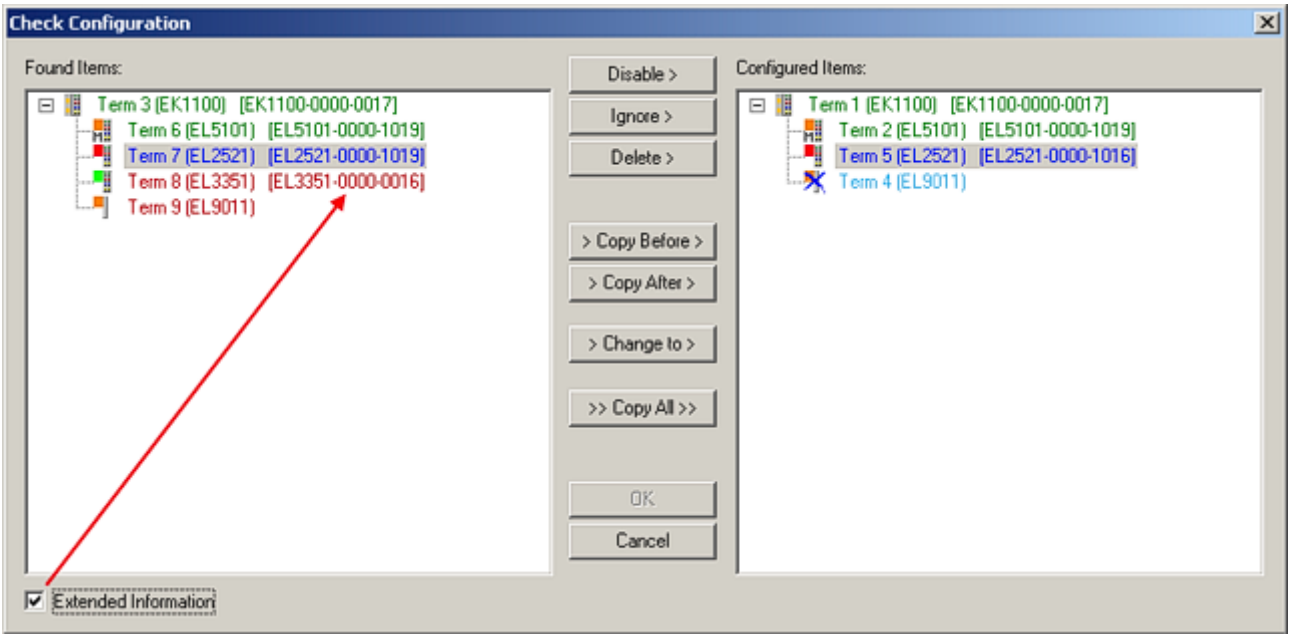


Fig. 122: Correction dialog

It is advisable to tick the “Extended Information” check box to reveal differences in the revision.

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored (“Ignore” button)
red	• This EtherCAT slave is not present on the other side. • It is present, but in a different revision, which also differs in its properties from the one specified. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

Device selection based on revision, compatibility

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The following compatibility rule of thumb is to be assumed for Beckhoff EtherCAT Terminals/ Boxes/ EJ-modules:

device revision in the system >= device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-**1018** is specified in the configuration, an EL2521-0025-**1018** or higher (-**1019**, -**1020**) can be used in practice.

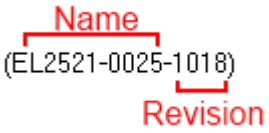


Fig. 123: Name/revision of the terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

In this case the process image of the device is shown in the configuration tree and can be parameterized as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

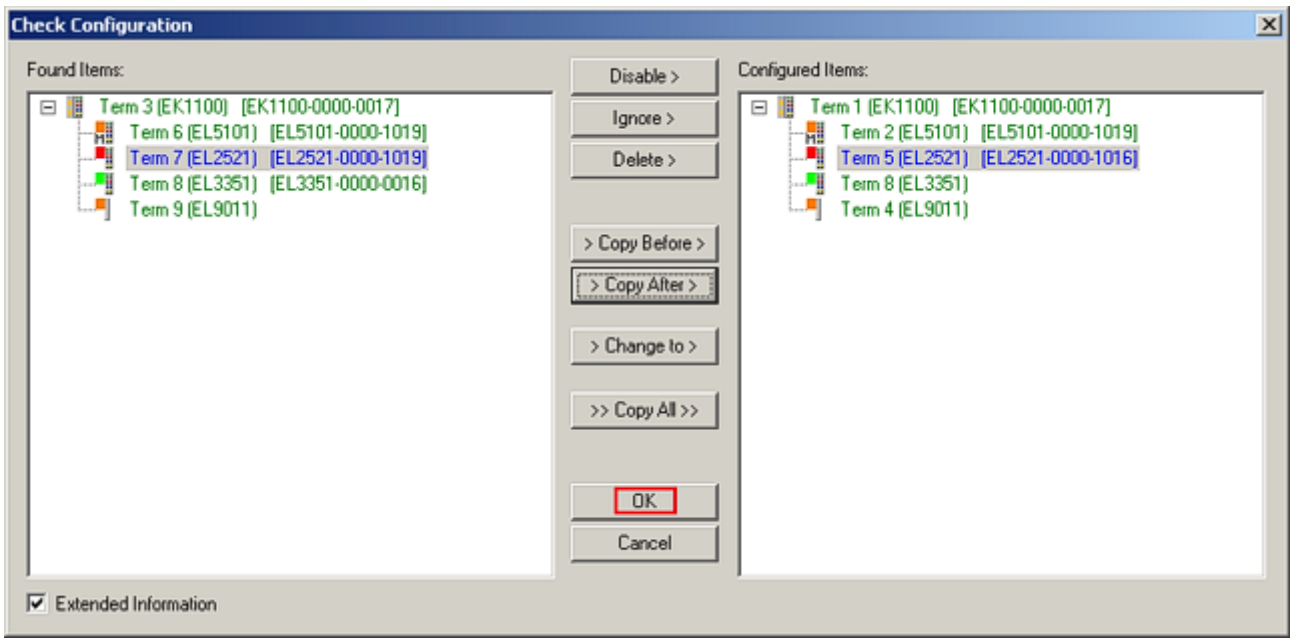


Fig. 124: Correction dialog with modifications

Once all modifications have been saved or accepted, click “OK” to transfer them to the real *.tsm configuration.

Change to Compatible Type

TwinCAT offers a function *Change to Compatible Type…* for the exchange of a device whilst retaining the links in the task.

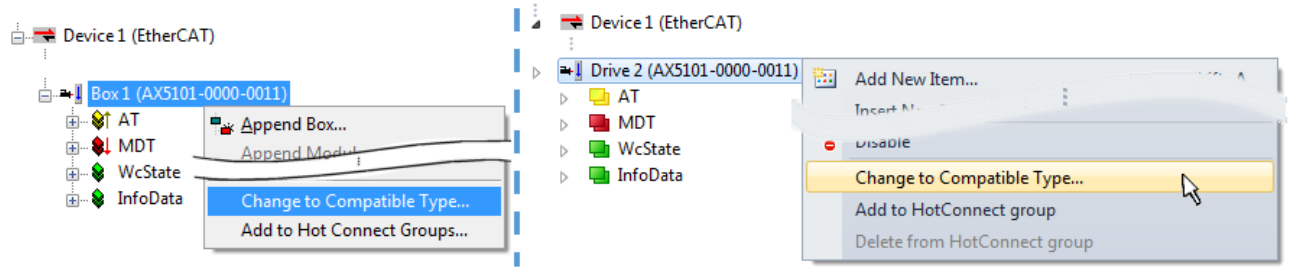


Fig. 125: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

The following elements in the ESI of an EtherCAT device are compared by TwinCAT and assumed to be the same in order to decide whether a device is indicated as "compatible":

- Physics (e.g. RJ45, Ebus...)
- FMMU (additional ones are allowed)
- SyncManager (SM, additional ones are allowed)
- EoE (attributes MAC, IP)
- CoE (attributes SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)
- FoE
- PDO (process data: Sequence, SyncUnit SU, SyncManager SM, EntryCount, Ent-ry.Datatype)

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

The TwinCAT System Manager offers a function for the exchange of a device: Change to Alternative Type

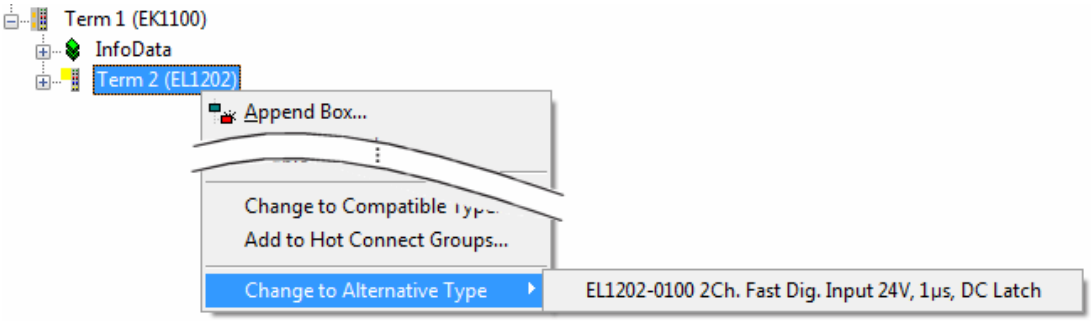


Fig. 126: TwinCAT 2 Dialog Change to Alternative Type

If called, the System Manager searches in the procured device ESI (in this example: EL1202-0000) for details of compatible devices contained there. The configuration is changed and the ESI-EEPROM is overwritten at the same time – therefore this process is possible only in the online state (ConfigMode).

7.2.7 EtherCAT subscriber configuration

In the left-hand window of the TwinCAT 2 System Manager or the Solution Explorer of the TwinCAT 3 Development Environment respectively, click on the element of the terminal within the tree you wish to configure (in the example: EL3751 Terminal 3).

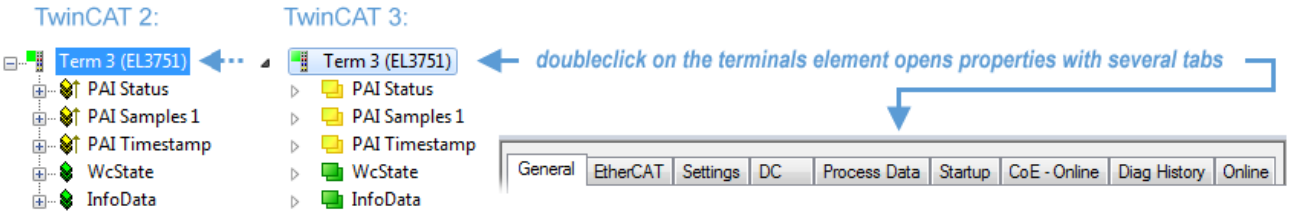


Fig. 127: Branch element as terminal EL3751

In the right-hand window of the TwinCAT System Manager (TwinCAT 2) or the Development Environment (TwinCAT 3), various tabs are now available for configuring the terminal. And yet the dimension of complexity of a subscriber determines which tabs are provided. Thus as illustrated in the example above the terminal EL3751 provides many setup options and also a respective number of tabs are available. On the contrary by the terminal EL1004 for example the tabs “General”, “EtherCAT”, “Process Data” and “Online“ are available only. Several terminals, as for instance the EL6695 provide special functions by a tab with its own terminal name, so “EL6695” in this case. A specific tab “Settings” by terminals with a wide range of setup options will be provided also (e.g. EL3751).

“General” tab

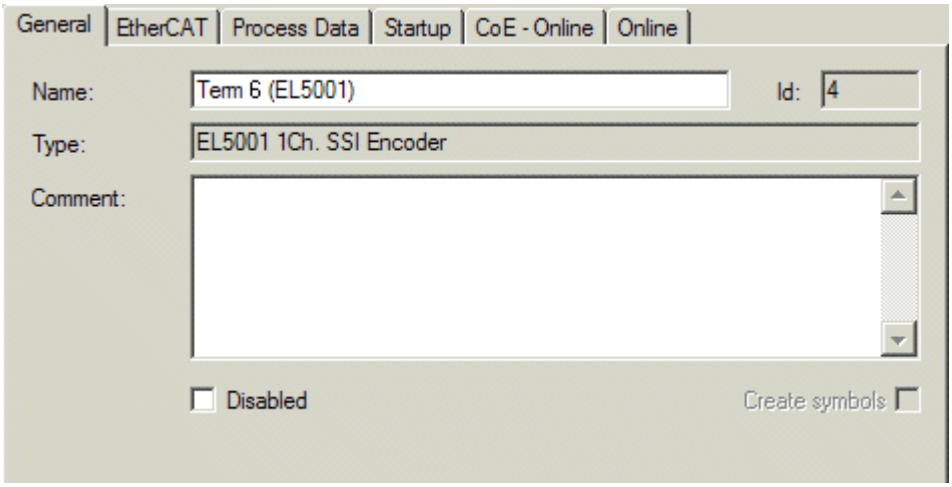


Fig. 128: “General” tab

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

“EtherCAT” tab

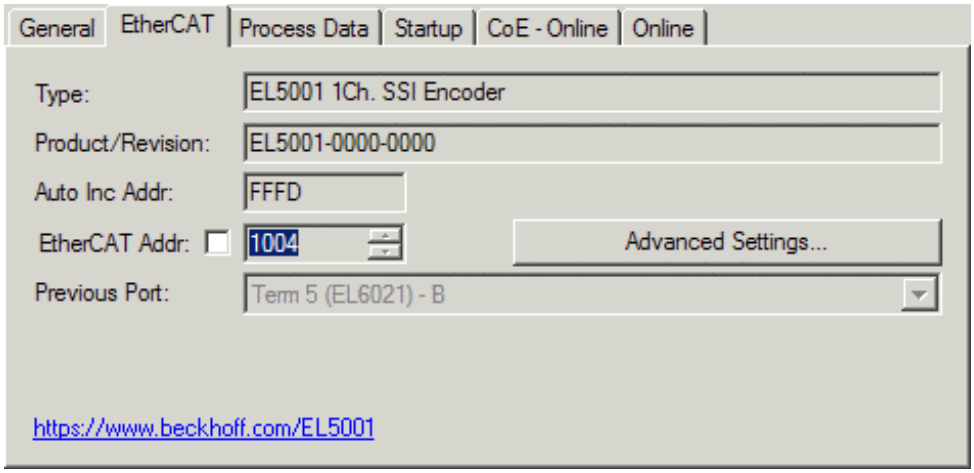


Fig. 129: “EtherCAT” tab

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

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

“Process Data” tab

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

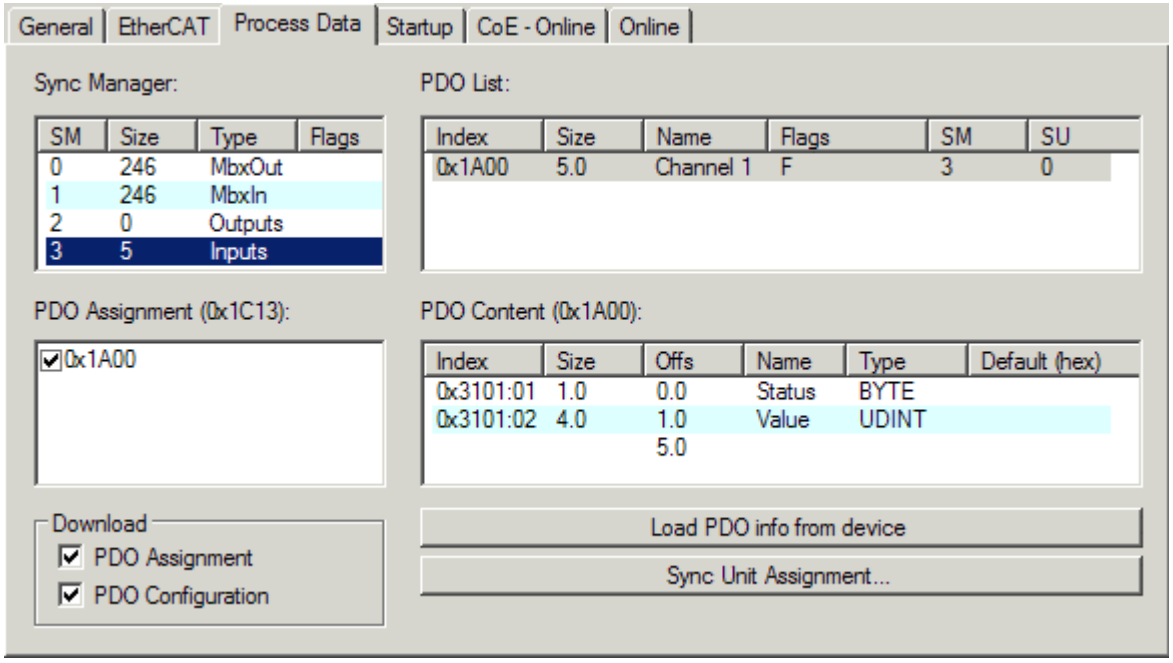


Fig. 130: “Process Data” tab

The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which process data (size in bits/bytes, source location, transmission type) it wants to transfer to or from this slave. Incorrect configuration can prevent successful start-up of the slave.

For Beckhoff EtherCAT EL, ES, EM, EJ and EP slaves the following applies in general:

- The input/output process data supported by the device are defined by the manufacturer in the ESI/XML description. The TwinCAT EtherCAT Master uses the ESI description to configure the slave correctly.
- The process data can be modified in the System Manager. See the device documentation.
  Examples of modifications include: mask out a channel, displaying additional cyclic information, 16-bit display instead of 8-bit data size, etc.
- In so-called “intelligent” EtherCAT devices the process data information is also stored in the CoE directory. Any changes in the CoE directory that lead to different PDO settings prevent successful startup of the slave. It is not advisable to deviate from the designated process data, because the device firmware (if available) is adapted to these PDO combinations.

If the device documentation allows modification of process data, proceed as follows (see Figure *Configuring the process data*).

- A: select the device to configure
- B: in the “Process Data” tab select Input or Output under SyncManager (C)
- D: the PDOs can be selected or deselected
- H: the new process data are visible as linkable variables in the System Manager
  The new process data are active once the configuration has been activated and TwinCAT has been restarted (or the EtherCAT master has been restarted)
- E: if a slave supports this, Input and Output PDO can be modified simultaneously by selecting a so-called PDO record (“predefined PDO settings”).

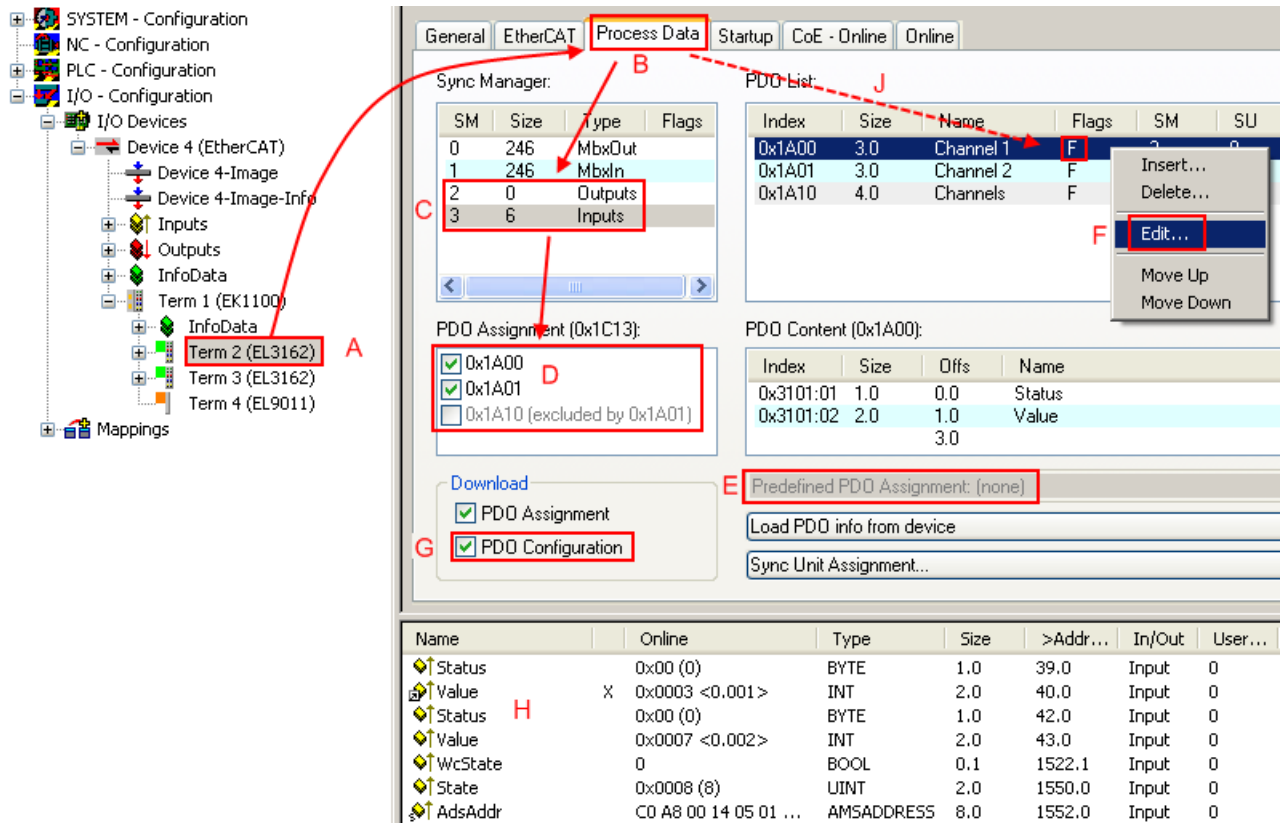


Fig. 131: Configuring the process data

Manual modification of the process data

According to the ESI description, a PDO can be identified as “fixed” with the flag “F” in the PDO overview (Fig. *Configuring the process data*, J). The configuration of such PDOs cannot be changed, even if TwinCAT offers the associated dialog (“Edit”). In particular, CoE content cannot be displayed as cyclic process data. This generally also applies in cases where a device supports download of the PDO configuration, “G”. In case of incorrect configuration the EtherCAT slave usually refuses to start and change to OP state. The System Manager displays an “invalid SM cfg” logger message: This error message (“invalid SM IN cfg” or “invalid SM OUT cfg”) also indicates the reason for the failed start.

A detailed description [▶ 122] can be found at the end of this section.

“Startup” tab

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

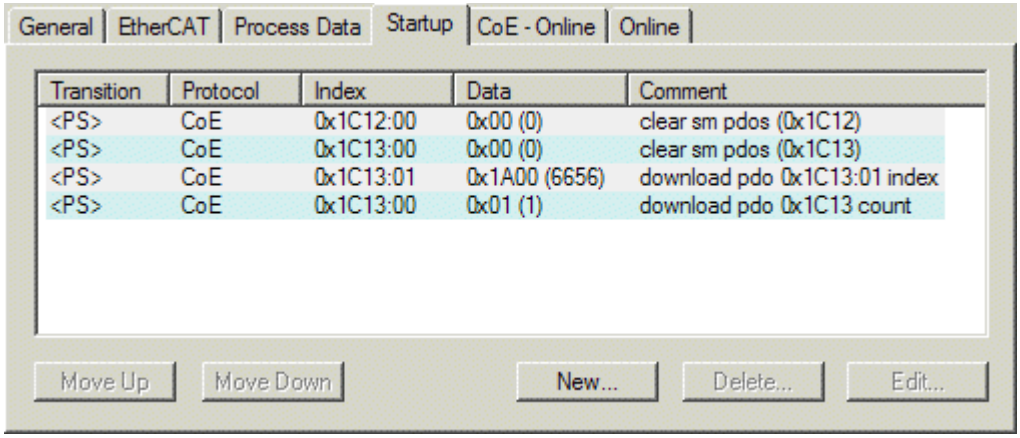


Fig. 132: “Startup” tab

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

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

“CoE - Online” tab

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

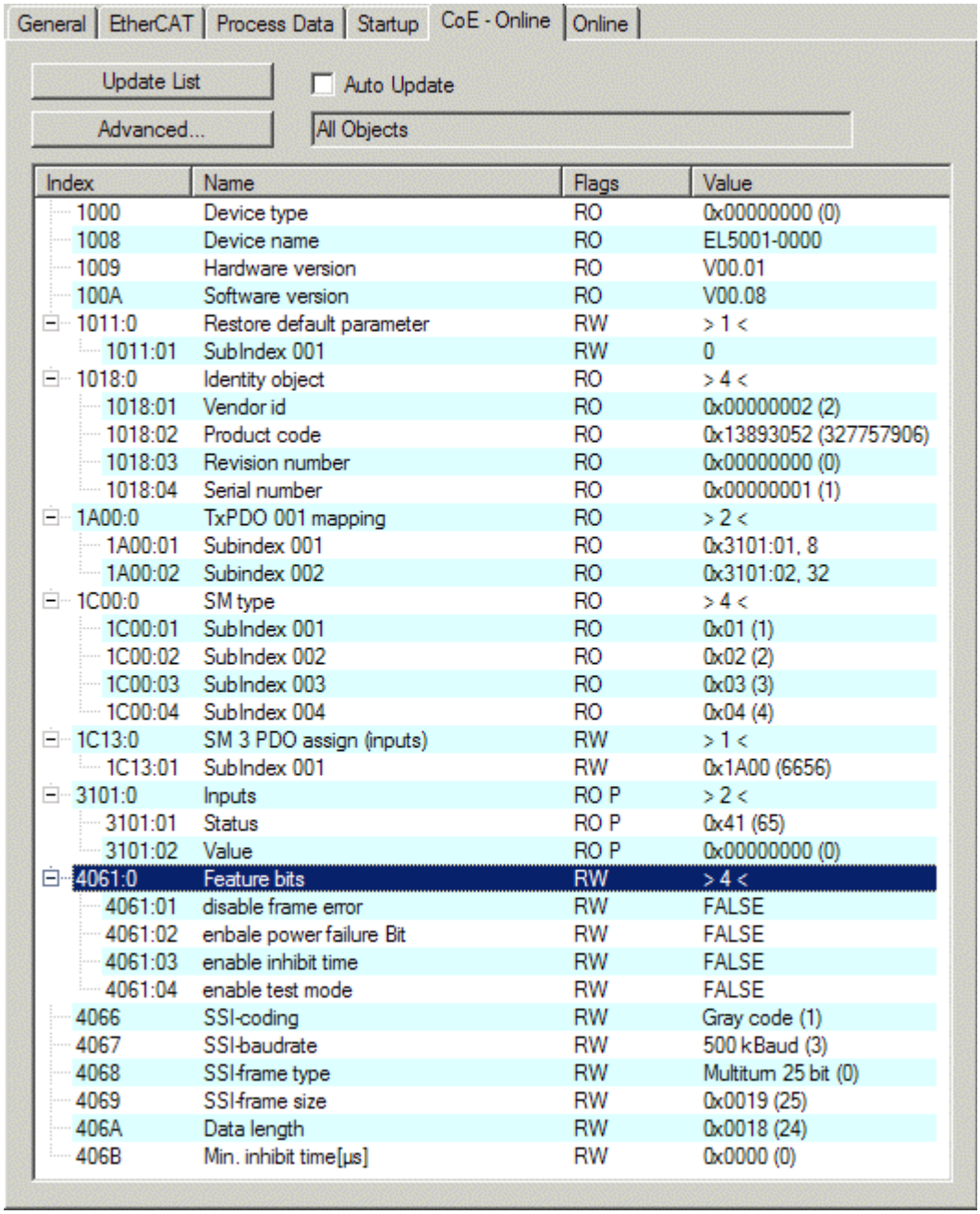


Fig. 133: “CoE - Online” tab

Object list display

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

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

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

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

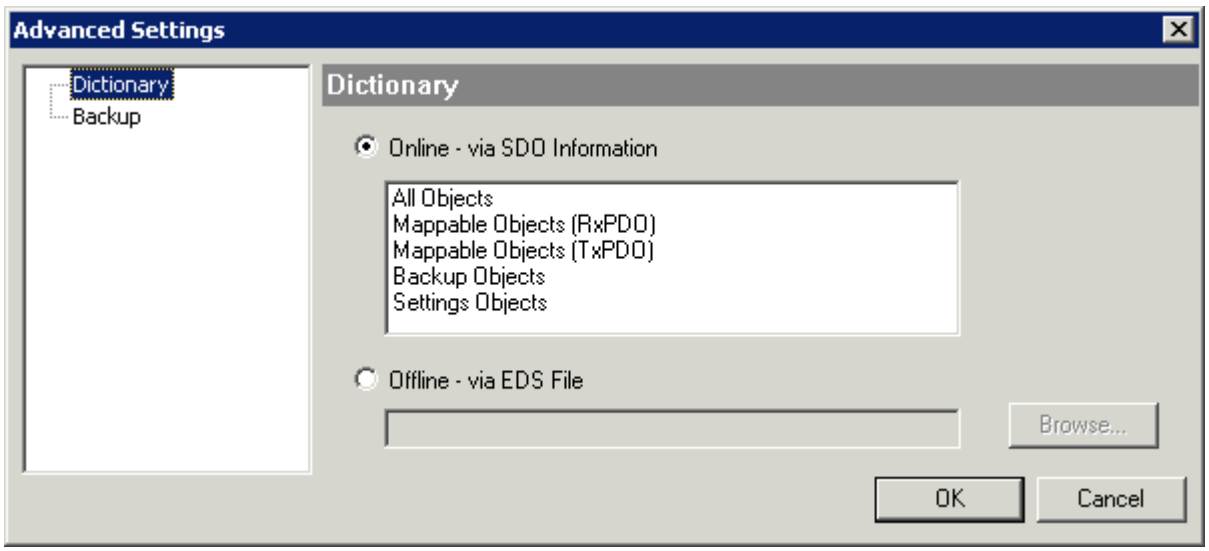


Fig. 134: Dialog “Advanced settings”

Online - via SDO Information	If this option button is selected, the list of the objects included in the object list of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.
Offline - via EDS File	If this option button is selected, the list of the objects included in the object list is read from an EDS file provided by the user.

“Online” tab

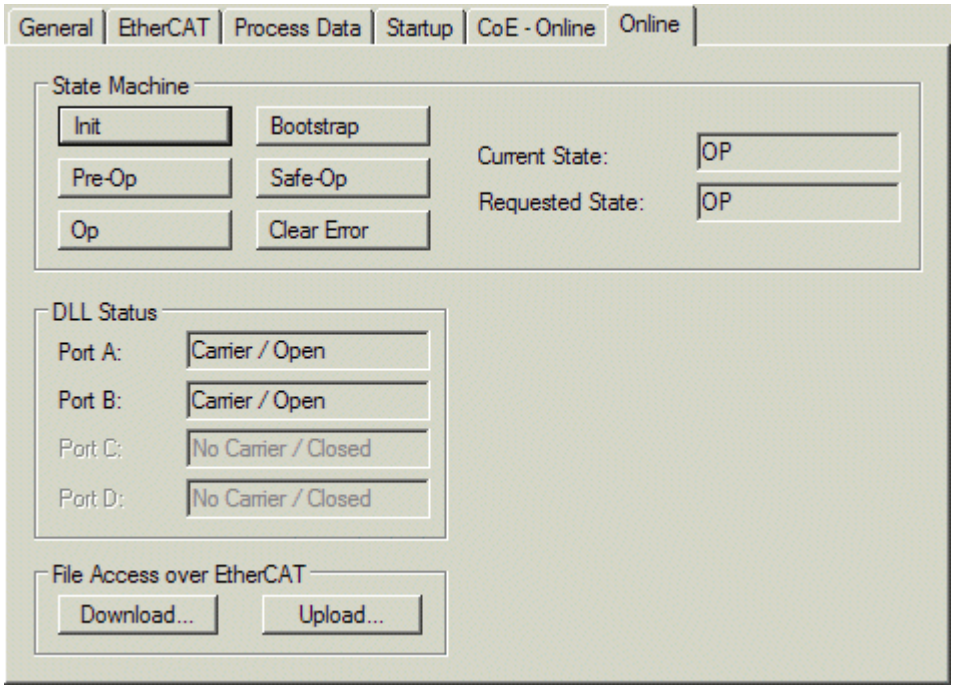


Fig. 135: “Online” tab

State Machine

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

DLL Status

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

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

File Access over EtherCAT

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

“DC” tab (Distributed Clocks)

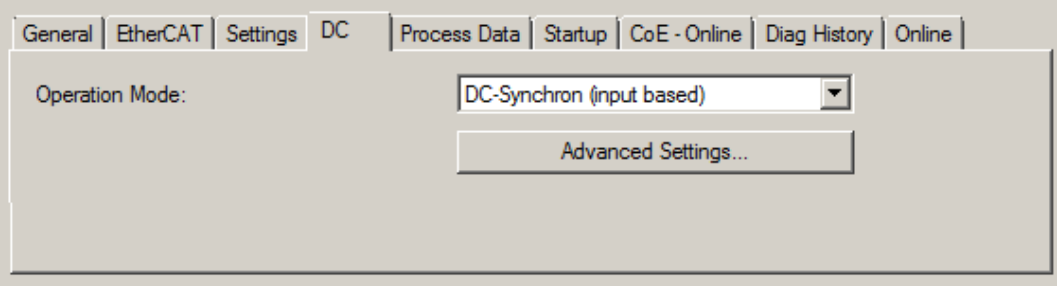


Fig. 136: “DC” tab (Distributed Clocks)

Operation Mode	Options (optional): • FreeRun • SM-Synchron • DC-Synchron (Input based) • DC-Synchron
Advanced Settings…	Advanced settings for readjustment of the real time determinant TwinCAT-clock

Detailed information to Distributed Clocks is specified on http://infosys.beckhoff.com:

Fieldbus Components → EtherCAT Terminals → EtherCAT System documentation → EtherCAT basics → Distributed Clocks

7.2.7.1 Detailed description of Process Data tab

Sync Manager

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

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

PDO Assignment

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

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

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

Activation of PDO assignment

✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,

a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see Online tab [▶ 120]),

b) and the System Manager has to reload the EtherCAT slaves

( button for TwinCAT 2 or  button for TwinCAT 3)

PDO list

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

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

PDO Content

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

Download

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

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the Startup [▶ 117] tab.

PDO Configuration

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

7.2.8 NC configuration (motion)

Installation of the latest XML device description

Please ensure that you have installed the corresponding latest XML device description in TwinCAT. This can be downloaded from the Beckhoff Website and installed according to the installation instructions.

The configuration of the axes and linking in the TwinCAT System Manager (Config mode) are described below, taking the EL5151 as an example. Proceed as follows:

Adding a Motion element

Right-click *Motion*

Select *Add New Item...* in the pull-down menu (Fig. *Motion, add item*).

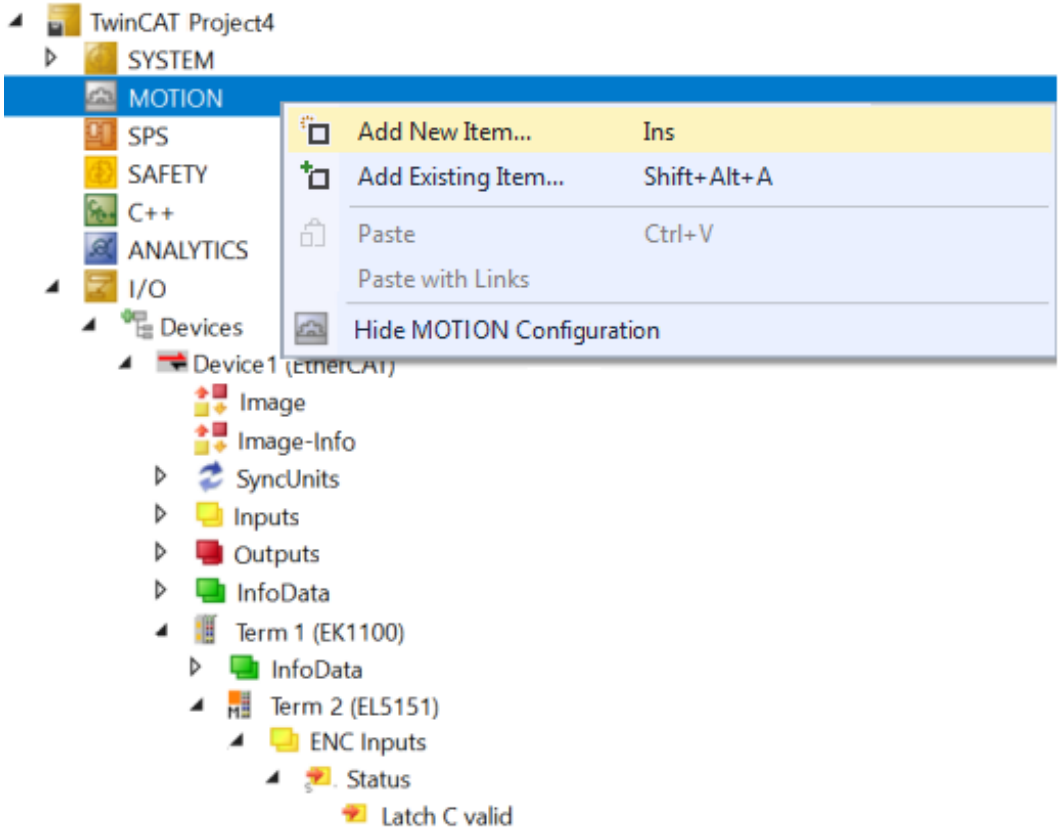


Fig. 137: Motion, add item

Select the configuration type *NC/PTP NCI Configuration*.

Name the task and confirm with *OK* (Fig. *Select type, name and confirm task*)

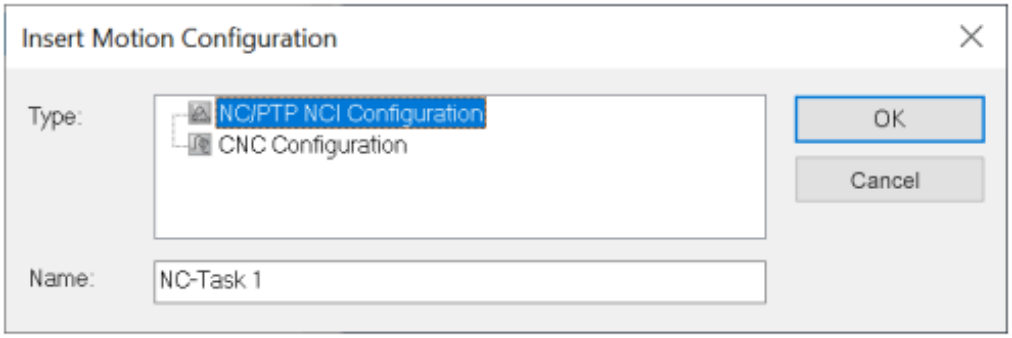


Fig. 138: Select type, name and confirm task

Adding an axis

Right-click *Axes*

Select *Add New Item* in the pull-down menu (Fig. *Insert axis*).

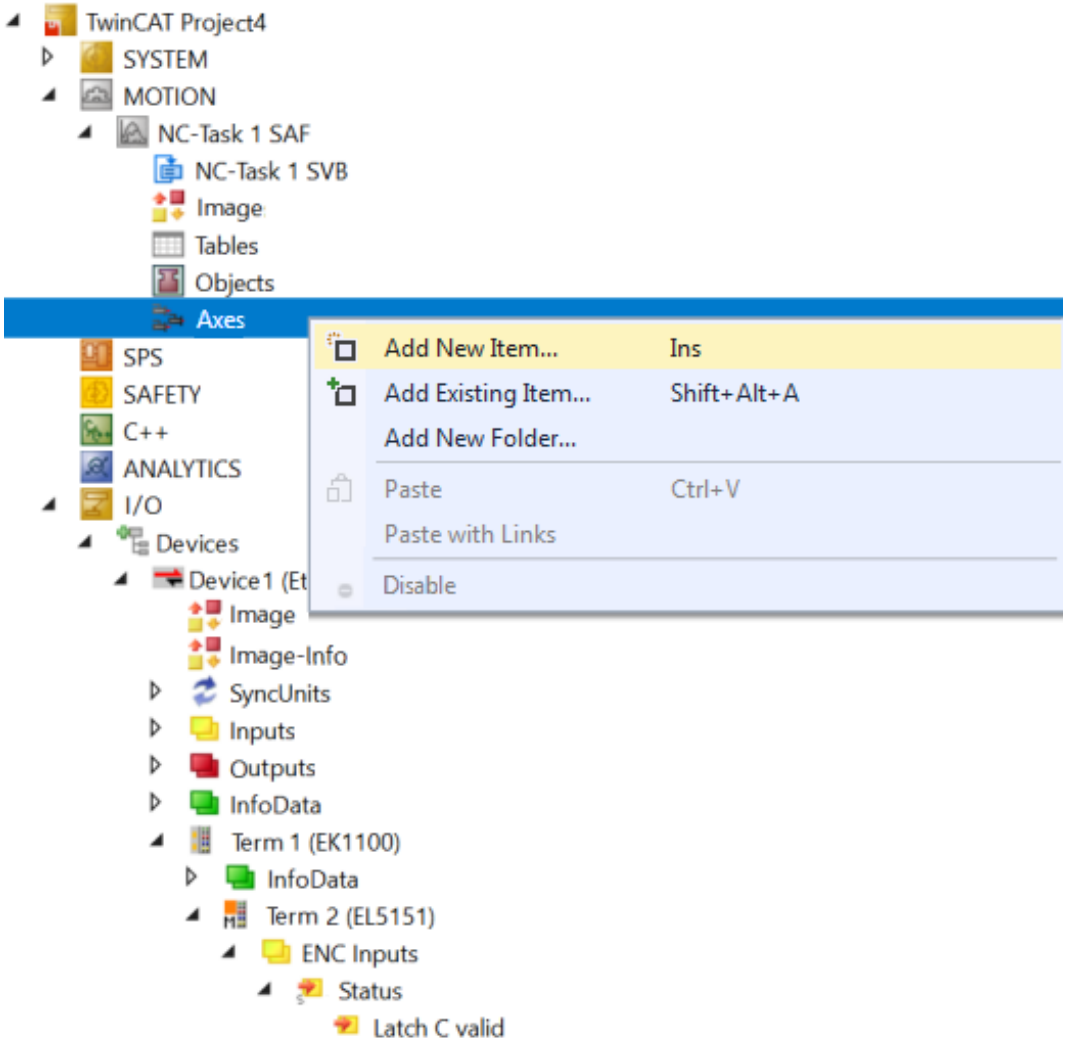


Fig. 139: Insert axis

Choose a name for the axis

Select the *Encoder axis* type and confirm with *OK* (Fig. *Naming the axis and selecting the type*)

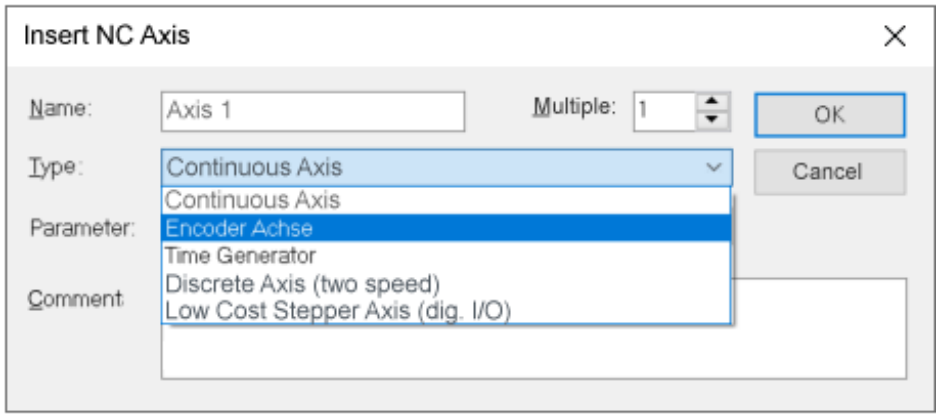


Fig. 140: Entering a name for the axis and selecting a type

Linking the encoder

After selecting the *NC Encoder* tab select *Encoder* (*KL5101/KI5111/IP5109/EL5101/EL5151/Profile MDP 511*) in the *Type* pull-down menu (Fig. *Selecting the encoder*)

Click the button *Linked to...*

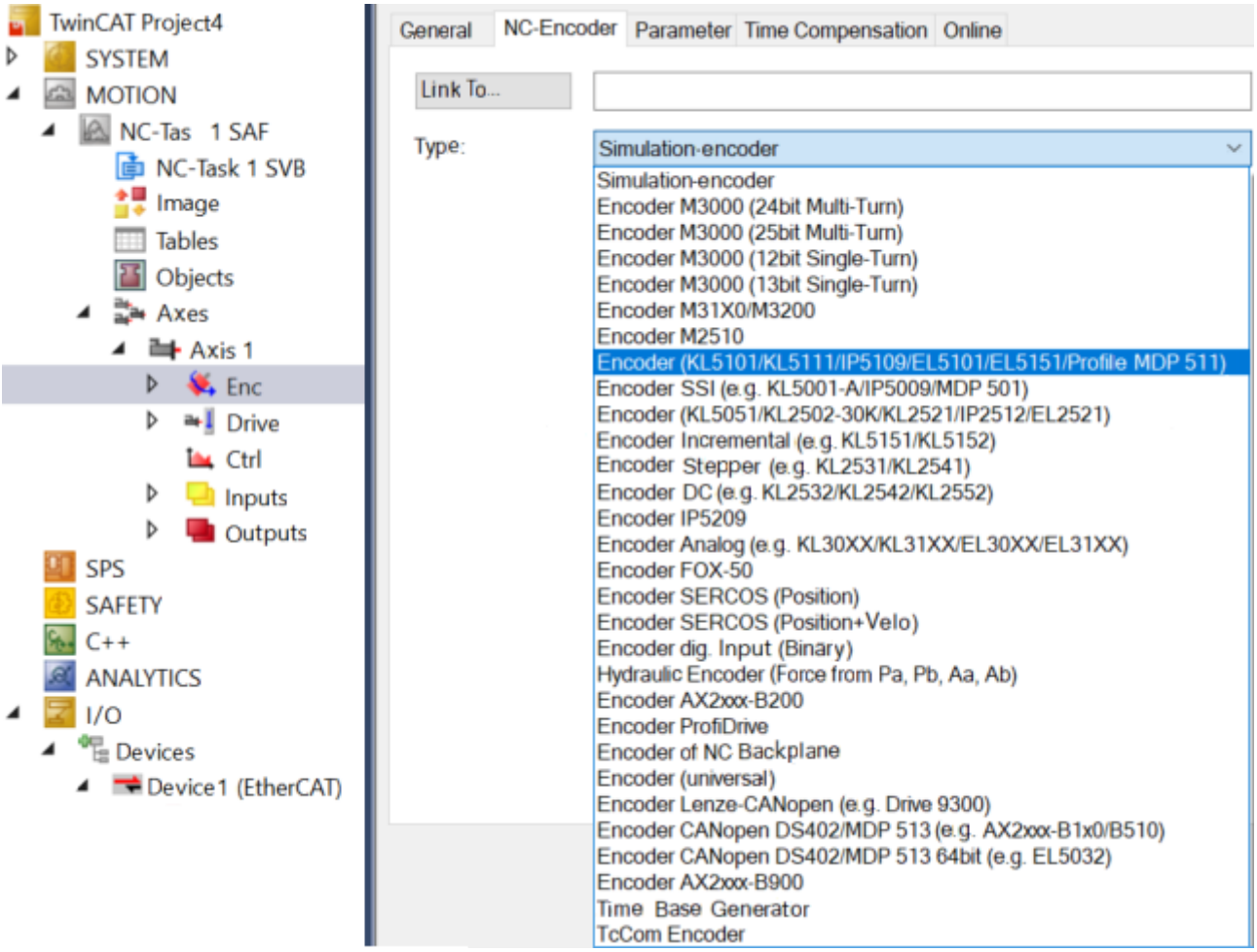


Fig. 141: Selecting the encoder

Select the *EL5151* terminal and confirm with *OK* (Fig. *Select and confirm the encoder terminal*)

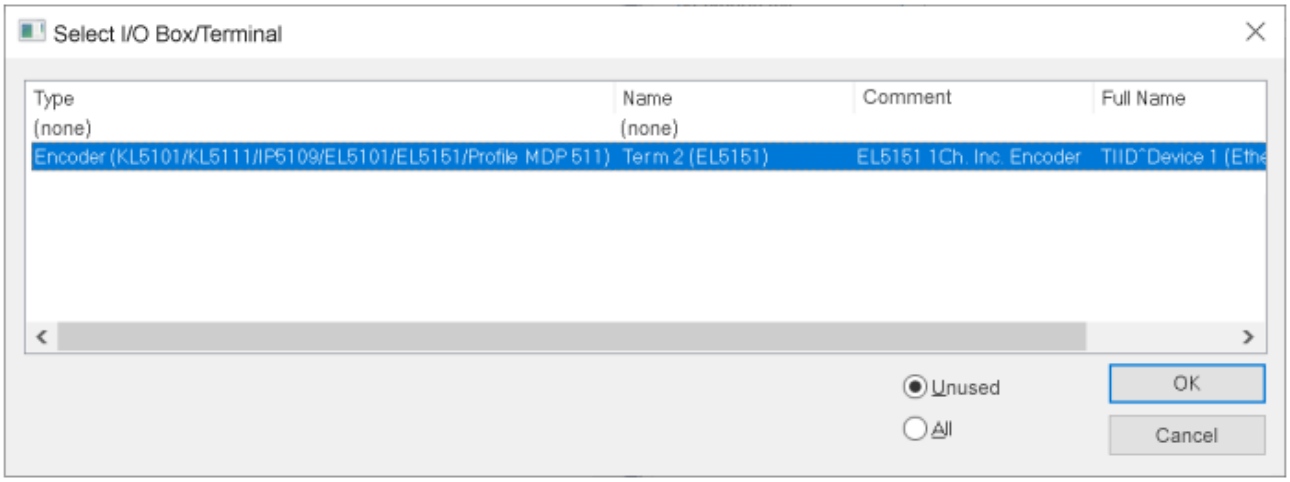


Fig. 142: Selecting and confirming an encoder terminal

Display of the linked inputs and outputs

The corresponding inputs and outputs of the EL5151 are now linked to the NC task (Fig. *Inputs and outputs of the EL5151 linked to the NC task*)

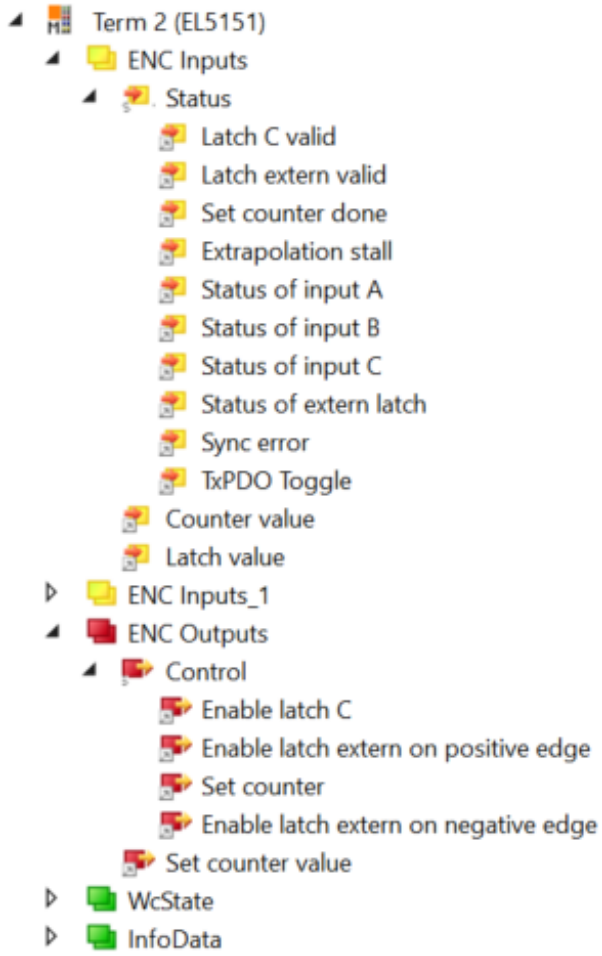


Fig. 143: Inputs and outputs of the EL5151 linked to the NC task

7.3 General Commissioning Instructions for an EtherCAT Slave

This summary briefly deals with a number of aspects of EtherCAT Slave operation under TwinCAT. More detailed information on this may be found in the corresponding sections of, for instance, the EtherCAT System Documentation.

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

Generally speaking an EtherCAT Slave provides a variety of diagnostic information that can be used by the controlling task.

This diagnostic information relates to differing levels of communication. It therefore has a variety of sources, and is also updated at various times.

Any application that relies on I/O data from a fieldbus being correct and up to date must make diagnostic access to the corresponding underlying layers. EtherCAT and the TwinCAT System Manager offer comprehensive diagnostic elements of this kind. Those diagnostic elements that are helpful to the controlling task for diagnosis that is accurate for the current cycle when in operation (not during commissioning) are discussed below.

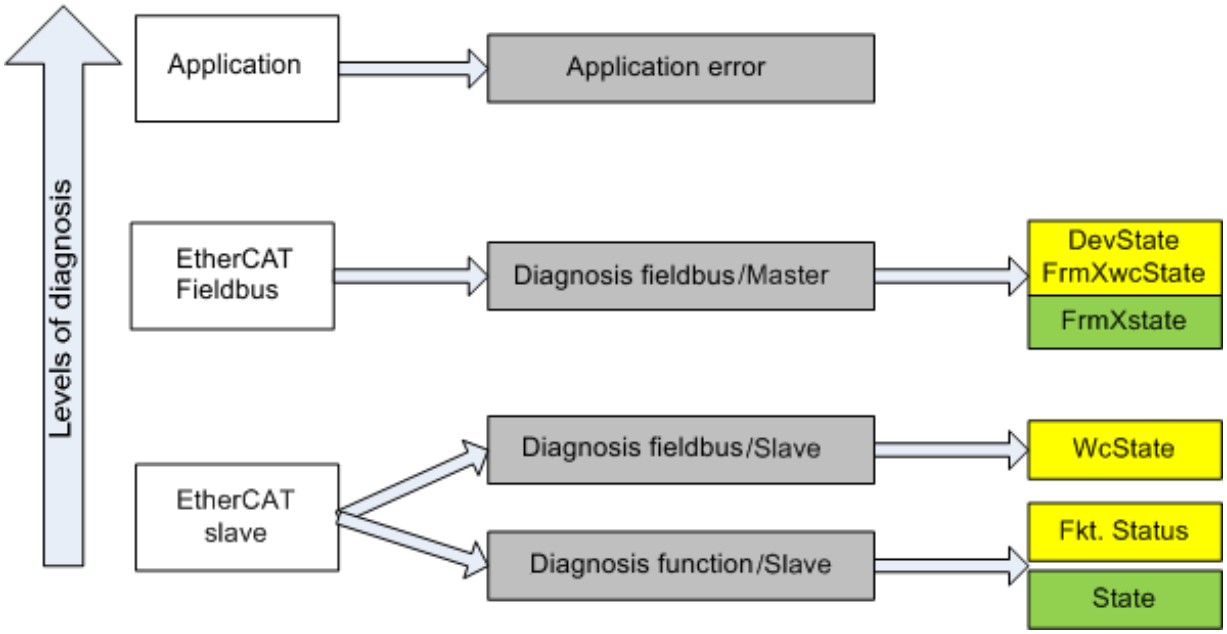


Fig. 144: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

- communication diagnosis typical for a slave (diagnosis of successful participation in the exchange of process data, and correct operating mode)
  This diagnosis is the same for all slaves.

  as well as
- function diagnosis typical for a channel (device-dependent)
  See the corresponding device documentation

The colors in Fig. *Selection of the diagnostic information of an EtherCAT Slave* also correspond to the variable colors in the System Manager, see Fig. *Basic EtherCAT Slave Diagnosis in the PLC*.

Colour	Meaning
yellow	Input variables from the Slave to the EtherCAT Master, updated in every cycle
red	Output variables from the Slave to the EtherCAT Master, updated in every cycle
green	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore useful to read such variables through ADS.

Fig. *Basic EtherCAT Slave Diagnosis in the PLC* shows an example of an implementation of basic EtherCAT Slave Diagnosis. A Beckhoff EL3102 (2-channel analogue input terminal) is used here, as it offers both the communication diagnosis typical of a slave and the functional diagnosis that is specific to a channel. Structures are created as input variables in the PLC, each corresponding to the process image.

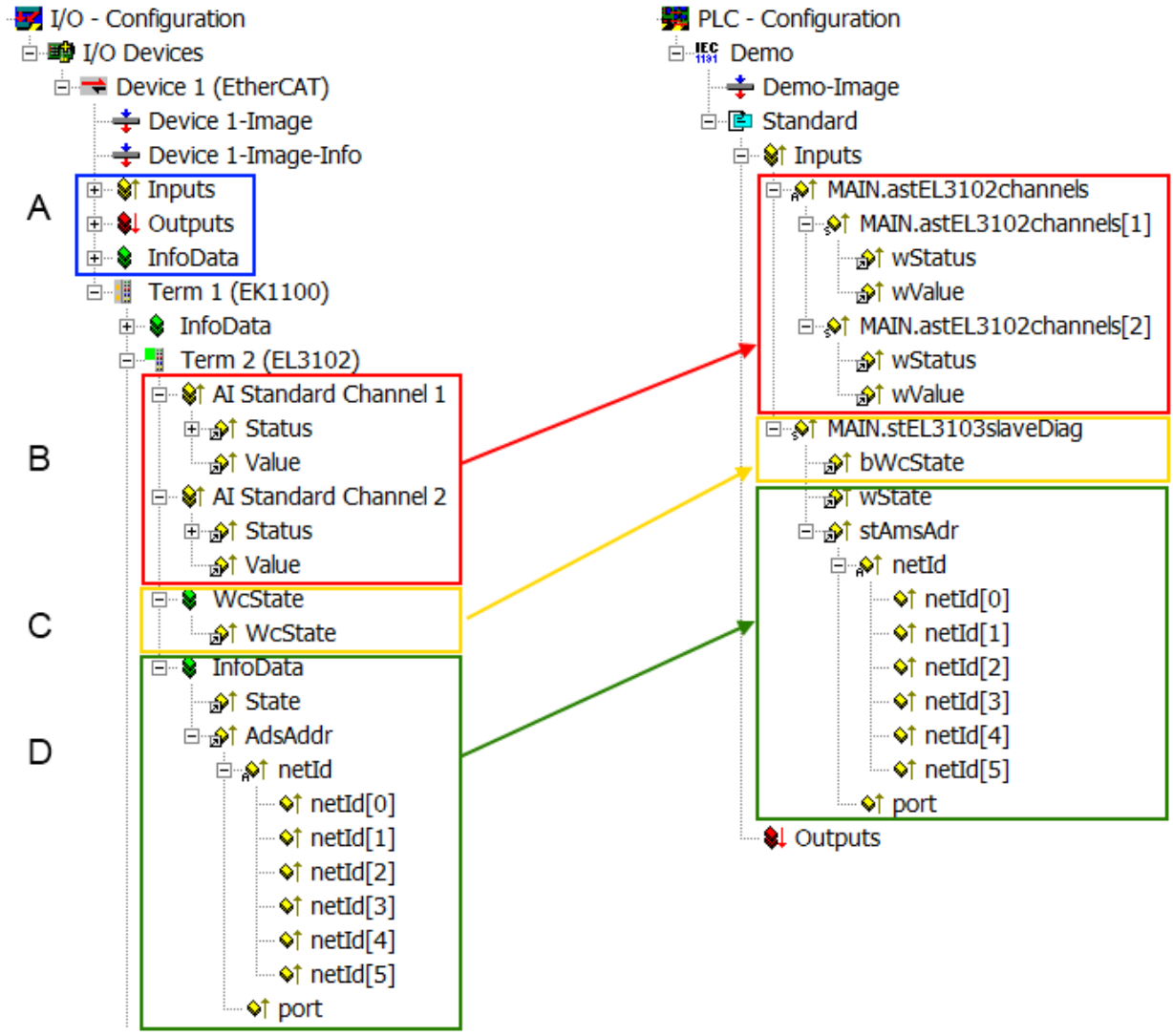


Fig. 145: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated cyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: • CoE in the Master for communication with/through the Slaves • Functions from *TcEtherCAT.lib* • Perform an OnlineScan
B	In the example chosen (EL3102) the EL3102 comprises two analogue input channels that transmit a single function status for the most recent cycle.	Status • the bit significations may be found in the device documentation • other devices may supply more information, or none that is typical of a slave	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the function status must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
C	For every EtherCAT Slave that has cyclic process data, the Master displays, using what is known as a WorkingCounter, whether the slave is participating successfully and without error in the cyclic exchange of process data. This important, elementary information is therefore provided for the most recent cycle in the System Manager 1. at the EtherCAT Slave, and, with identical contents 2. as a collective variable at the EtherCAT Master (see Point A) for linking.	WcState (Working Counter) 0: valid real-time communication in the last cycle 1: invalid real-time communication This may possibly have effects on the process data of other Slaves that are located in the same SyncUnit	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the communication status of the EtherCAT Slave must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
D	Diagnostic information of the EtherCAT Master which, while it is represented at the slave for linking, is actually determined by the Master for the Slave concerned and represented there. This information cannot be characterized as real-time, because it • is only rarely/never changed, except when the system starts up • is itself determined acyclically (e.g. EtherCAT Status)	State current Status (INIT..OP) of the Slave. The Slave must be in OP (=8) when operating normally. *AdsAddr* The ADS address is useful for communicating from the PLC/task via ADS with the EtherCAT Slave, e.g. for reading/writing to the CoE. The AMS-NetID of a slave corresponds to the AMS-NetID of the EtherCAT Master; communication with the individual Slave is possible via the *port* (= EtherCAT address).	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore possible to read such variables through ADS.

NOTICE

Diagnostic information

It is strongly recommended that the diagnostic information made available is evaluated so that the application can react accordingly.

CoE Parameter Directory

The CoE parameter directory (CanOpen-over-EtherCAT) is used to manage the set values for the slave concerned. Changes may, in some circumstances, have to be made here when commissioning a relatively complex EtherCAT Slave. It can be accessed through the TwinCAT System Manager, see Fig. *EL3102, CoE directory*:

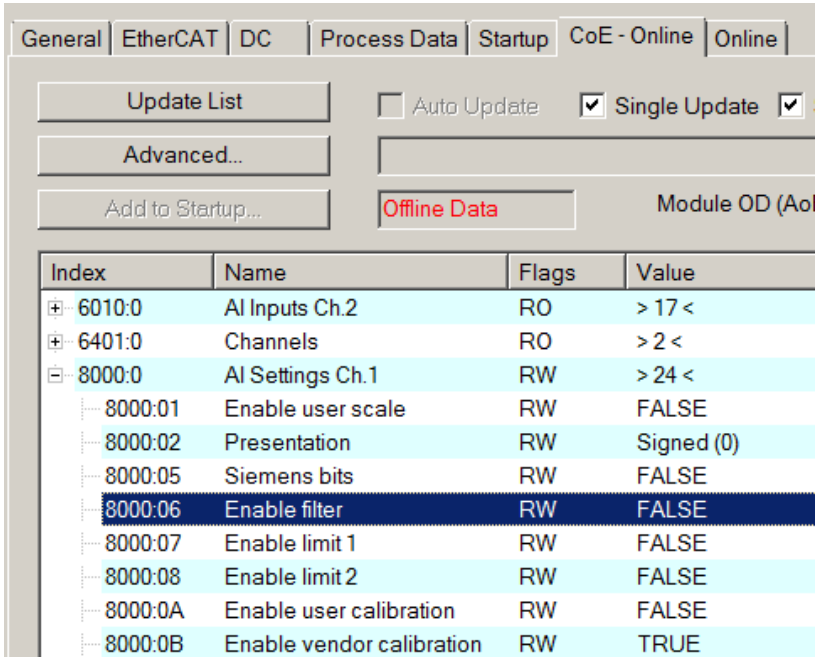


Fig. 146: EL3102, CoE directory

EtherCAT System Documentation

The comprehensive description in the EtherCAT System Documentation (EtherCAT Basics --> CoE Interface) must be observed!

A few brief extracts:

- Whether changes in the online directory are saved locally in the slave depends on the device. EL terminals (except the EL66xx) are able to save in this way.
- The user must manage the changes to the StartUp list.

Commissioning aid in the TwinCAT System Manager

Commissioning interfaces are being introduced as part of an ongoing process for EL/EP EtherCAT devices. These are available in TwinCAT System Managers from TwinCAT 2.11R2 and above. They are integrated into the System Manager through appropriately extended ESI configuration files.

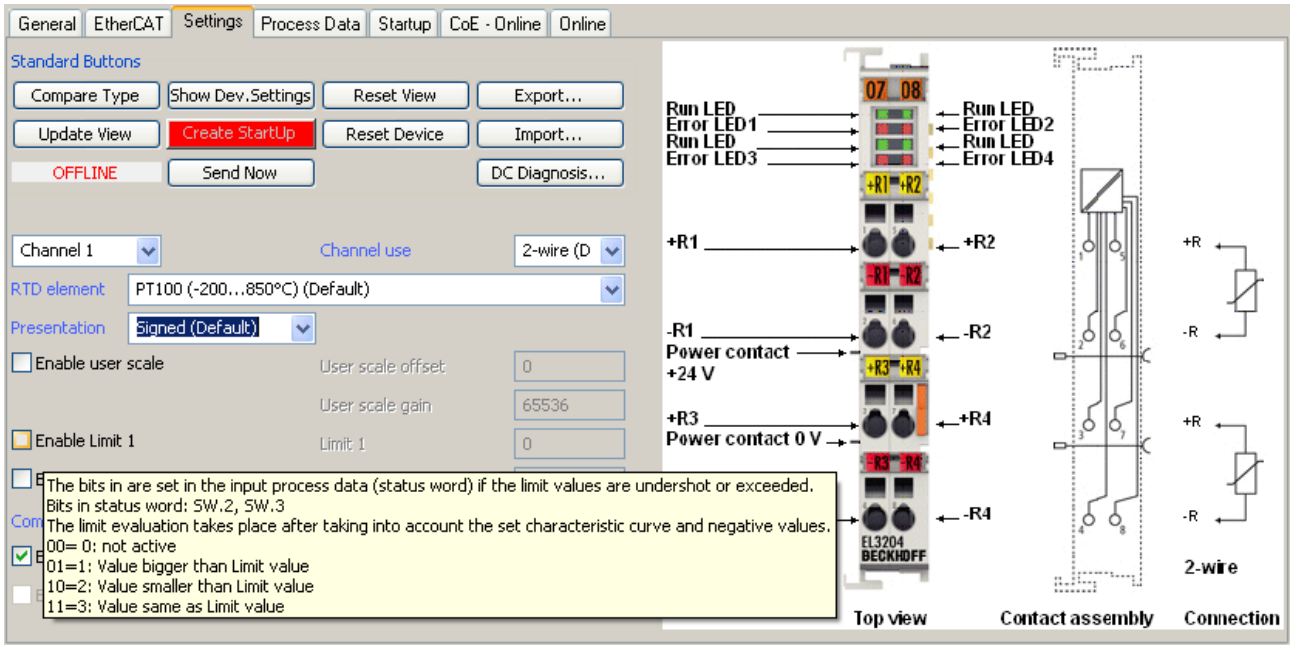


Fig. 147: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

- CoE Parameter Directory
- DC/FreeRun mode
- the available process data records (PDO)

Although the “Process Data”, “DC”, “Startup” and “CoE-Online” that used to be necessary for this are still displayed, it is recommended that, if the commissioning aid is used, the automatically generated settings are not changed by it.

The commissioning tool does not cover every possible application of an EL/EP device. If the available setting options are not adequate, the user can make the DC, PDO and CoE settings manually, as in the past.

EtherCAT State: automatic default behaviour of the TwinCAT System Manager and manual operation

After the operating power is switched on, an EtherCAT Slave must go through the following statuses

- INIT
- PREOP
- SAFEOP
- OP

to ensure sound operation. The EtherCAT Master directs these statuses in accordance with the initialization routines that are defined for commissioning the device by the ES/XML and user settings (Distributed Clocks (DC), PDO, CoE). See also the section on "Principles of Communication, EtherCAT State Machine [▶ 32]" in this connection. Depending how much configuration has to be done, and on the overall communication, booting can take up to a few seconds.

The EtherCAT Master itself must go through these routines when starting, until it has reached at least the OP target state.

The target state wanted by the user, and which is brought about automatically at start-up by TwinCAT, can be set in the System Manager. As soon as TwinCAT reaches the status RUN, the TwinCAT EtherCAT Master will approach the target states.

Standard setting

The advanced settings of the EtherCAT Master are set as standard:

- EtherCAT Master: OP
- Slaves: OP
  This setting applies equally to all Slaves.

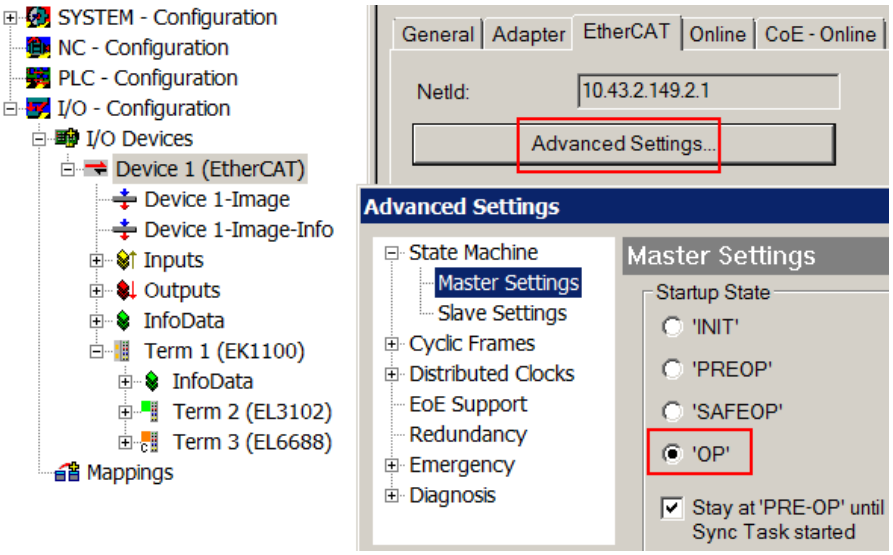


Fig. 148: Default behaviour of the System Manager

In addition, the target state of any particular Slave can be set in the “Advanced Settings” dialogue; the standard setting is again OP.

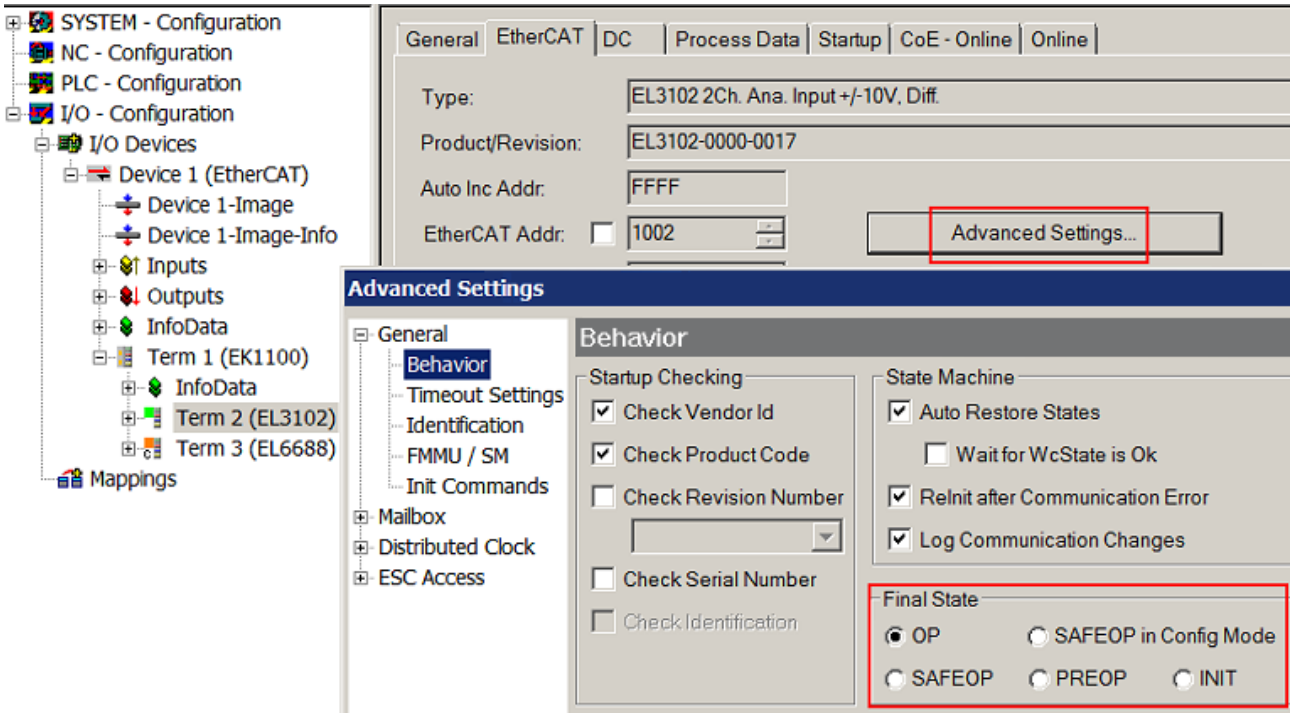


Fig. 149: Default target state in the Slave

Manual Control

There are particular reasons why it may be appropriate to control the states from the application/task/PLC. For instance:

- for diagnostic reasons
- to induce a controlled restart of axes
- because a change in the times involved in starting is desirable

In that case it is appropriate in the PLC application to use the PLC function blocks from the *TcEtherCAT.lib*, which is available as standard, and to work through the states in a controlled manner using, for instance, *FB_EcSetMasterState*.

It is then useful to put the settings in the EtherCAT Master to INIT for master and slave.

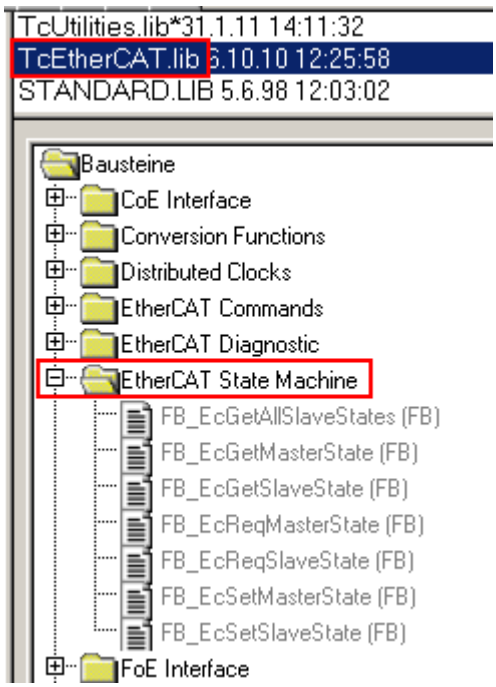
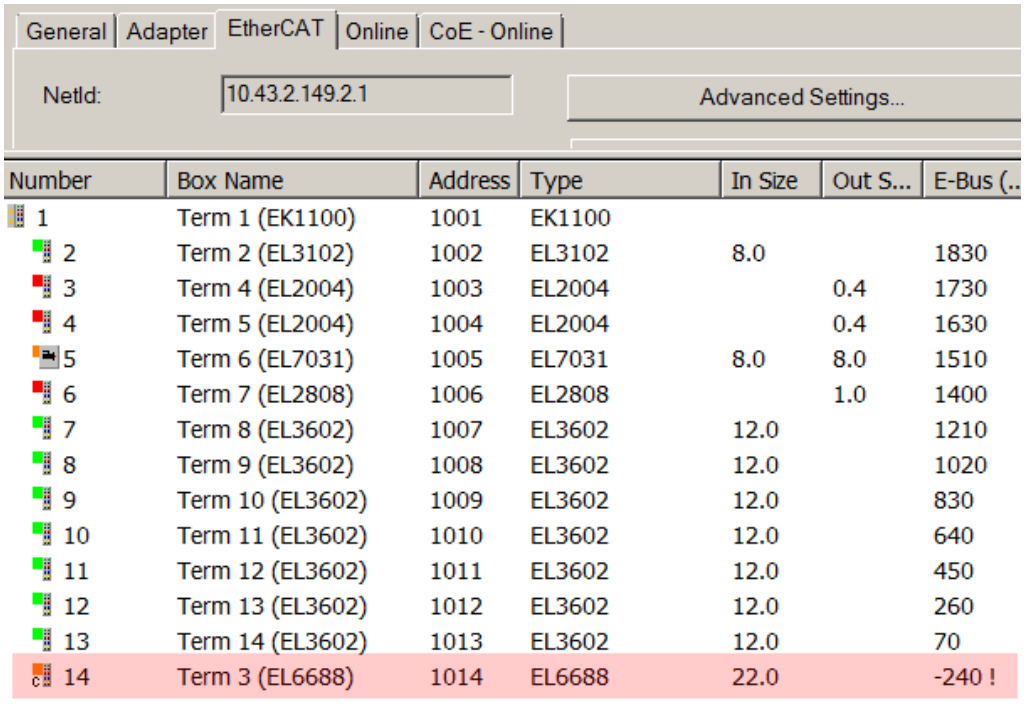


Fig. 150: PLC function blocks

Note regarding E-Bus current

EL/ES terminals are placed on the DIN rail at a coupler on the terminal strand. 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. 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 as a column value. 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	Address	Type	In Size	Out S...	E-Bus (..
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL3102)	1002	EL3102	8.0		1830
3	Term 4 (EL2004)	1003	EL2004		0.4	1730
4	Term 5 (EL2004)	1004	EL2004		0.4	1630
5	Term 6 (EL7031)	1005	EL7031	8.0	8.0	1510
6	Term 7 (EL2808)	1006	EL2808		1.0	1400
7	Term 8 (EL3602)	1007	EL3602	12.0		1210
8	Term 9 (EL3602)	1008	EL3602	12.0		1020
9	Term 10 (EL3602)	1009	EL3602	12.0		830
10	Term 11 (EL3602)	1010	EL3602	12.0		640
11	Term 12 (EL3602)	1011	EL3602	12.0		450
12	Term 13 (EL3602)	1012	EL3602	12.0		260
13	Term 14 (EL3602)	1013	EL3602	12.0		70
14	Term 3 (EL6688)	1014	EL6688	22.0		-240 !

Fig. 151: Illegally exceeding the E-Bus current

From TwinCAT 2.11 and above, a warning message “E-Bus Power of Terminal...” is output in the logger window when such a configuration is activated:

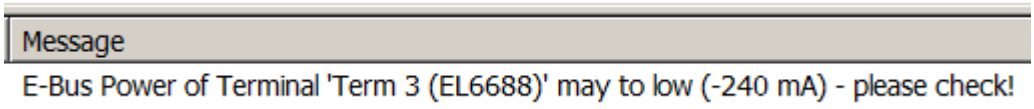


Fig. 152: Warning message for exceeding E-Bus current

NOTICE

Caution! Malfunction possible!

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

8 EL5131 - Commissioning

8.1 Process data

Process data	Description
Operation modes [▶ 141]	The scope of the process data can be selected via "Predefined PDO Assignment".
Synchronicity mode [▶ 144]	In addition to the frame-triggered operation mode (SM mode), a mode based on synchronization via the distributed clocks is available.

8.1.1 Sync Manager (SM)

The scope of process data offered can be modified via the "Process Data" tab (see Fig. *Process Data tab SM3, EL5131 (default)* below).

A detailed description for setting the process data can be found in chapter EtherCAT subscriber configuration [▶ 115].

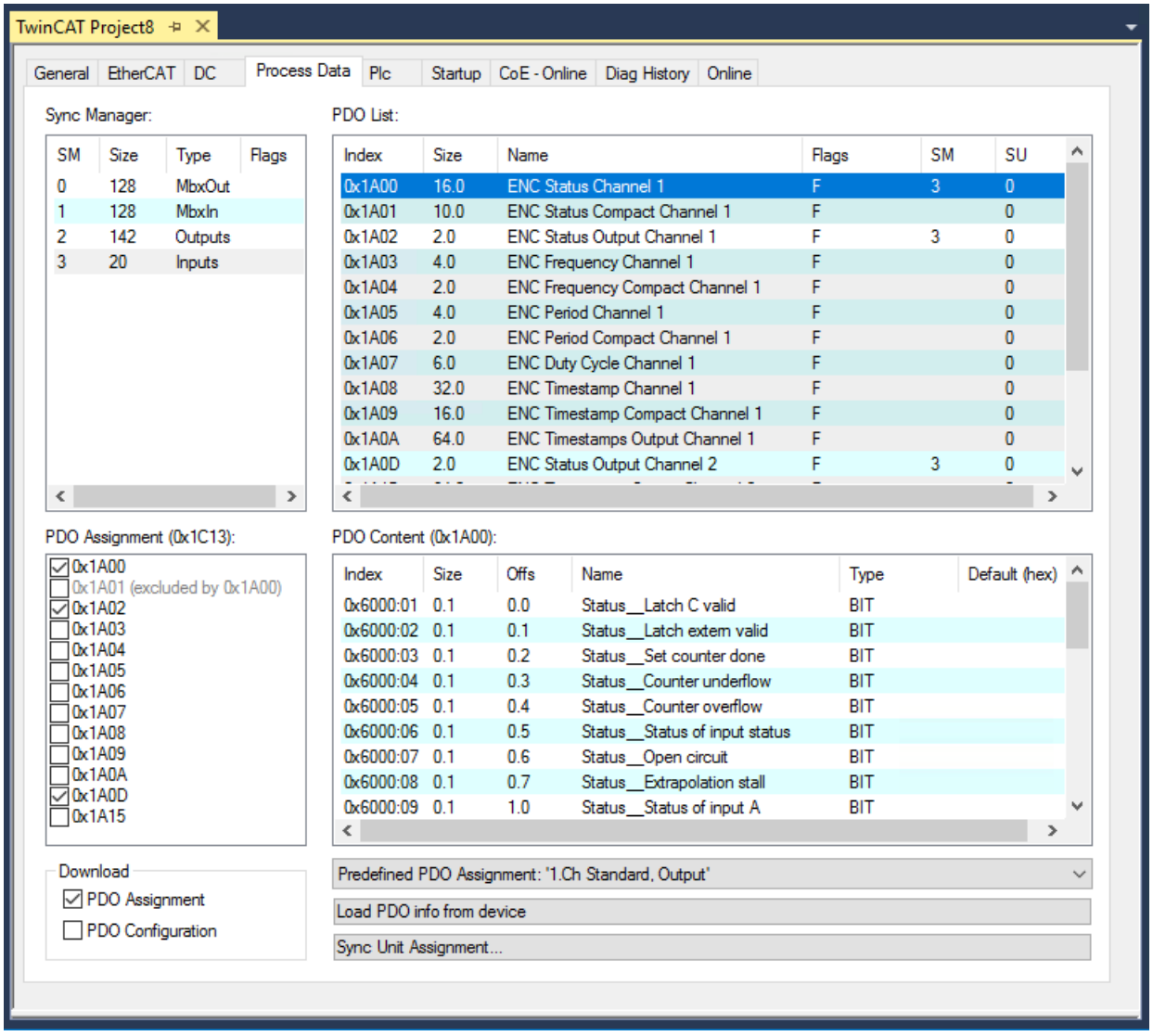


Fig. 153: EL5131 - Process data tab SM3 (default)

8.1.2 PDO assignment

The EL5131 offers a wide range of functions and thus also a large volume of process data.

8.1.2.1 SM3 - Inputs (0x1A00 ... 0x1A15)

0x1A00 - ENC Status Channel 1 (16.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x6000:01 [▶ 223] - Status_Latch C valid (0.1) 0x6000:02 [▶ 223] - Status_Latch extern valid (0.1) 0x6000:03 [▶ 223] - Status_Set counter done (0.1) 0x6000:04 [▶ 223] - Status_Counter underflow (0.1) 0x6000:05 [▶ 223] - Status_Counter overflow (0.1) 0x6000:06 [▶ 223] - Status_Status of input status (0.1) 0x6000:07 [▶ 223] - Status_Open circuit (0.1) 0x6000:08 [▶ 223] - Status_Extrapolation stall (0.1) 0x6000:09 [▶ 223] - Status_Status of input A (0.1) 0x6000:0A [▶ 223] - Status_Status of input B (0.1) 0x6000:0B [▶ 223] - Status_Status of input C (0.1) 0x6000:0C [▶ 223] - Status_Status of input gate (0.1) 0x6002:0D [▶ 224] - Status_Diag (0.1) 0x6002:0E [▶ 224] - Status_TxPDO State (0.1) 0x6002:0F [▶ 224] - Status_Input cycle counter (0.2) 0x6002:11 [▶ 224] - Status_Software gate valid (0.1) 0x6002:12 [▶ 224] - Status_Latch extern 2 valid (0.1) 0x6002:13 [▶ 224] - Status_Direction inversion detected (0.1) 0x6002:14 [▶ 224] - Status_Status of extern Latch (0.1) 0x6002:15 [▶ 224] - Status_Counter value out of range (0.1) 0x6000:11 [▶ 223] - Counter value (4.0) 0x6000:12 [▶ 223] - Latch value (4.0) 0x6000:22 [▶ 223] - Latch value 2 (4.0)	0x1A01 [▶ 237] - ENC Status Compact Channel 1 (10.0)

0x1A01 - ENC Status Compact Channel 1 (10.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x6000:01 [▶ 223] - Status_Latch C valid (0.1) 0x6000:02 [▶ 223] - Status_Latch extern valid (0.1) 0x6000:03 [▶ 223] - Status_Set counter done (0.1) 0x6000:04 [▶ 223] - Status_Counter underflow (0.1) 0x6000:05 [▶ 223] - Status_Counter overflow (0.1) 0x6000:06 [▶ 223] - Status_Status of input status (0.1) 0x6000:07 [▶ 223] - Status_Open circuit (0.1) 0x6000:08 [▶ 223] - Status_Extrapolation stall (0.1) 0x6000:09 [▶ 223] - Status_Status of input A (0.1) 0x6000:0A [▶ 223] - Status_Status of input B (0.1) 0x6000:0B [▶ 223] - Status_Status of input C (0.1) 0x6000:0C [▶ 223] - Status_Status of input gate (0.1) 0x6002:0D [▶ 224] - Status_Diag (0.1) 0x6002:0E [▶ 224] - Status_TxPDO State (0.1) 0x6002:0F [▶ 224] - Status_Input cycle counter (0.2) 0x6002:11 [▶ 224] - Status_Software gate valid (0.1) 0x6002:12 [▶ 224] - Status_Latch extern 2 valid (0.1) 0x6002:13 [▶ 224] - Status_Direction inversion detected (0.1) 0x6002:14 [▶ 224] - Status_Status of extern Latch (0.1) 0x6002:15 [▶ 224] - Status_Counter value out of range (0.1) 0x6000:11 [▶ 223] - Counter value (2.0) 0x6000:12 [▶ 223] - Latch value (2.0) 0x6000:22 [▶ 223] - Latch value 2 (2.0)	0x1A00 [▶ 236] - ENC Status Channel 1 (16.0)

0x1A02 - ENC Status Output Channel 1 (2.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x6003:01 [▶ 224] - Status Output_Status of digital output (0.1) 0x6003:02 [▶ 224] - Status Output_Status of tristate (0.1)	-

0x1A03 - ENC Frequency Channel 1 (4.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x6000:13 [▶ 223] - Frequency value (4.0)	0x1A04 [▶ 238] - ENC Frequency Compact Channel 1 (2.0)

0x1A04 - ENC Frequency Compact Channel 1 (2.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x6000:13 [▶ 223] - Frequency value (2.0)	0x1A03 [▶ 238] - ENC Frequency Channel 1 (4.0)

0x1A05 - ENC Period Channel 1 (4.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x6000:14 [▶ 223] - Period value (4.0)	0x1A06 [▶ 238] - ENC Period Compact Channel 1 (2.0)

0x1A06 - ENC Period Compact Channel 1 (2.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index Name Size (Byte.Bit)**
0x6000:14 [▶ 223] - Period value (2.0)	0x1A05 [▶ 238] - ENC Period Channel 1 (4.0)

0x1A07 - ENC Duty Cycle Channel 1 (6.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index Name Size (Byte.Bit)**
0x6000:23 [▶ 223] - Duty Cycle (2.0) 0x6000:24 [▶ 223] - Duty Cycle min (2.0) 0x6000:25 [▶ 223] - Duty Cycle max (2.0)	-

0x1A08 - ENC Timestamp Channel 1 (32.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index Name Size (Byte.Bit)**
0x6000:16 [▶ 223] - Timestamp (8.0) 0x6000:1F [▶ 223] - Timestamp C (8.0) 0x6000:20 [▶ 223] - Timestamp latch (8.0) 0x6000:21 [▶ 223] - Timestamp latch 2 (8.0)	0x1A09 [▶ 239] - ENC Timestamp Compact Channel 1 (16.0)

0x1A09 - ENC Timestamp Compact Channel 1 (16.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index Name Size (Byte.Bit)**
0x6000:16 [▶ 223] - Timestamp (4.0) 0x6000:1F [▶ 223] - Timestamp C (4.0) 0x6000:20 [▶ 223] - Timestamp latch (4.0) 0x6000:21 [▶ 223] - Timestamp latch 2 (4.0)	0x1A08 [▶ 238] - ENC Timestamp Channel 1 (32.0)

0x1A0A - ENC Timestamp Output Channel 1 (64.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index Name Size (Byte.Bit)**
0x6004:11 [▶ 225] - Timestamp_Timestamp 1 (8.0) 0x6004:12 [▶ 225] - Timestamp_Timestamp 2 (8.0) 0x6004:13 [▶ 225] - Timestamp_Timestamp 3 (8.0) 0x6004:14 [▶ 225] - Timestamp_Timestamp 4 (8.0) 0x6004:15 [▶ 225] - Timestamp_Timestamp 5 (8.0) 0x6004:16 [▶ 225] - Timestamp_Timestamp 6 (8.0) 0x6004:17 [▶ 225] - Timestamp_Timestamp 7 (8.0) 0x6004:18 [▶ 225] - Timestamp_Timestamp 8 (8.0)	-

0x1A0D - ENC Status Output Channel 2 (2.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x6013:01 [▶ 224] - Status Output_Status of digital output (0.1) 0x6013:02 [▶ 224] - Status Output_Status of tristate (0.1)	-

0x1A15 - ENC Timestamp Output Channel 2 (64.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x6014:11 [▶ 225] - Timestamp_Timestamp 1 (8.0) 0x6014:12 [▶ 225] - Timestamp_Timestamp 2 (8.0) 0x6014:13 [▶ 225] - Timestamp_Timestamp 3 (8.0) 0x6014:14 [▶ 225] - Timestamp_Timestamp 4 (8.0) 0x6014:15 [▶ 225] - Timestamp_Timestamp 5 (8.0) 0x6014:16 [▶ 225] - Timestamp_Timestamp 6 (8.0) 0x6014:17 [▶ 225] - Timestamp_Timestamp 7 (8.0) 0x6014:18 [▶ 225] - Timestamp_Timestamp 8 (8.0)	-

8.1.2.2 SM2 - Outputs (0x1600 ... 0x1605)

0x1600 - ENC Control Channel 1 (6.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x7000:01 [▶ 225] - Control_Enable latch C (0.1) 0x7000:02 [▶ 225] - Control_Enable latch extern on positive edge (0.1) 0x7000:03 [▶ 225] - Control_Set counter (0.1) 0x7000:04 [▶ 225] - Control_Enable latch extern on negative edge (0.1) 0x7000:08 [▶ 225] - Control_Set counter on latch C (0.1) 0x7000:09 [▶ 225] - Control_Set software gate (0.1) 0x7000:0A [▶ 225] - Control_Set counter on latch extern on positive edge (0.1) 0x7000:0B [▶ 225] - Control_Set counter on latch extern on negative edge (0.1) 0x7000:0C [▶ 225] - Control_Enable latch extern 2 on positive edge (0.1) 0x7000:0D [▶ 225] - Control_Enable latch extern 2 on negative edge (0.1) 0x7000:11 [▶ 225] - Set counter value (4.0)	0x1601 [▶ 232] - ENC Control Compact Channel 1 (4.0)

0x1601 - ENC Control Compact Channel 1 (4.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x7000:01 [▶ 225] - Control_Enable latch C (0.1) 0x7000:02 [▶ 225] - Control_Enable latch extern on positive edge (0.1) 0x7000:03 [▶ 225] - Control_Set counter (0.1) 0x7000:04 [▶ 225] - Control_Enable latch extern on negative edge (0.1) 0x7000:08 [▶ 225] - Control_Set counter on latch C (0.1) 0x7000:09 [▶ 225] - Control_Set software gate (0.1) 0x7000:0A [▶ 225] - Control_Set counter on latch extern on positive edge (0.1) 0x7000:0B [▶ 225] - Control_Set counter on latch extern on negative edge (0.1) 0x7000:0C [▶ 225] - Control_Enable latch extern 2 on positive edge (0.1) 0x7000:0D [▶ 225] - Control_Enable latch extern 2 on negative edge (0.1) 0x7000:11 [▶ 225] - Set counter value (4.0)	0x1600 [▶ 231] - ENC Control Channel 1 (8.0)

0x1602 - ENC Control Output Channel 1 (68.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x7001:01 [▶ 226] - Control Output_Control_Set digital output (0.1) 0x7001:02 [▶ 226] - Control Output_Control_Enable thresholds output (0.1) 0x7001:03 [▶ 226] - Control Output_Control_Set tristate (0.1) 0x7001:04 [▶ 226] - Control Output_Control_Disable thresholds cw (0.1) 0x7001:05 [▶ 226] - Control Output_Control_Disable thresholds ccw (0.1) 0x7002:11 [▶ 226] - Control Output_Number of output events (1.0) 0x7003:01 [▶ 226] - Control Output_Output state_Output state 1 (0.1) 0x7003:02 [▶ 226] - Control Output_Output state_Output state 2 (0.1) 0x7003:03 [▶ 226] - Control Output_Output state_Output state 3 (0.1) 0x7003:04 [▶ 226] - Control Output_Output state_Output state 4 (0.1) 0x7003:05 [▶ 226] - Control Output_Output state_Output state 5 (0.1) 0x7003:06 [▶ 226] - Control Output_Output state_Output state 6 (0.1) 0x7003:07 [▶ 226] - Control Output_Output state_Output state 7 (0.1) 0x7003:08 [▶ 226] - Control Output_Output state_Output state 8 (0.1) 0x7004:11 [▶ 227] - Control Output_Threshold_Threshold 1 (4.0) 0x7004:12 [▶ 227] - Control Output_Threshold_Threshold 2 (4.0) 0x7004:13 [▶ 227] - Control Output_Threshold_Threshold 3 (4.0) 0x7004:14 [▶ 227] - Control Output_Threshold_Threshold 4 (4.0) 0x7004:15 [▶ 227] - Control Output_Threshold_Threshold 5 (4.0) 0x7004:16 [▶ 227] - Control Output_Threshold_Threshold 6 (4.0) 0x7004:17 [▶ 227] - Control Output_Threshold_Threshold 7 (4.0) 0x7004:18 [▶ 227] - Control Output_Threshold_Threshold 8 (4.0) 0x7005:11 [▶ 228] - Control Output_Duration_Duration 1 (4.0) 0x7005:12 [▶ 228] - Control Output_Duration_Duration 2 (4.0) 0x7005:13 [▶ 228] - Control Output_Duration_Duration 3 (4.0) 0x7005:14 [▶ 228] - Control Output_Duration_Duration 4 (4.0) 0x7005:15 [▶ 228] - Control Output_Duration_Duration 5 (4.0) 0x7005:16 [▶ 228] - Control Output_Duration_Duration 6 (4.0) 0x7005:17 [▶ 228] - Control Output_Duration_Duration 7 (4.0) 0x7005:18 [▶ 228] - Control Output_Duration_Duration 8 (4.0)	-

0x1605 - ENC Control Output Channel 2 (68.0)	
Contents **Index - Name Size (Byte.Bit)**	**Excluded PDOs** **Index - Name Size (Byte.Bit)**
0x7011:01 [▶ 226] - Control Output_Control_Set digital output (0.1) 0x7011:02 [▶ 226] - Control Output_Control_Enable thresholds output (0.1) 0x7011:03 [▶ 226] - Control Output_Control_Set tristate (0.1) 0x7011:04 [▶ 226] - Control Output_Control_Disable thresholds cw (0.1) 0x7011:05 [▶ 226] - Control Output_Control_Disable thresholds ccw (0.1) 0x7012:11 [▶ 226] - Control Output_Number of output events (1.0) 0x7013:01 [▶ 226] - Control Output_Output state_Output state 1 (0.1) 0x7013:02 [▶ 226] - Control Output_Output state_Output state 2 (0.1) 0x7013:03 [▶ 226] - Control Output_Output state_Output state 3 (0.1) 0x7013:04 [▶ 226] - Control Output_Output state_Output state 4 (0.1) 0x7013:05 [▶ 226] - Control Output_Output state_Output state 5 (0.1) 0x7013:06 [▶ 226] - Control Output_Output state_Output state 6 (0.1) 0x7013:07 [▶ 226] - Control Output_Output state_Output state 7 (0.1) 0x7013:08 [▶ 226] - Control Output_Output state_Output state 8 (0.1) 0x7014:11 [▶ 227] - Control Output_Threshold_Threshold 1 (4.0) 0x7014:12 [▶ 227] - Control Output_Threshold_Threshold 2 (4.0) 0x7014:13 [▶ 227] - Control Output_Threshold_Threshold 3 (4.0) 0x7014:14 [▶ 227] - Control Output_Threshold_Threshold 4 (4.0) 0x7014:15 [▶ 227] - Control Output_Threshold_Threshold 5 (4.0) 0x7014:16 [▶ 227] - Control Output_Threshold_Threshold 6 (4.0) 0x7014:17 [▶ 227] - Control Output_Threshold_Threshold 7 (4.0) 0x7014:18 [▶ 227] - Control Output_Threshold_Threshold 8 (4.0) 0x7015:11 [▶ 228] - Control Output_Duration_Duration 1 (4.0) 0x7015:12 [▶ 228] - Control Output_Duration_Duration 2 (4.0) 0x7005:13 [▶ 228] - Control Output_Duration_Duration 3 (4.0) 0x7015:14 [▶ 228] - Control Output_Duration_Duration 4 (4.0) 0x7015:15 [▶ 228] - Control Output_Duration_Duration 5 (4.0) 0x7015:16 [▶ 228] - Control Output_Duration_Duration 6 (4.0) 0x7015:17 [▶ 228] - Control Output_Duration_Duration 7 (4.0) 0x7015:18 [▶ 228] - Control Output_Duration_Duration 8 (4.0)	-

8.1.3 Predefined PDO Assignment

The "Predefined PDO Assignment" enables a simplified selection of the process data. The desired function is selected on the lower part of the Process Data tab. As a result, all necessary PDOs are automatically enabled and the unnecessary PDOs are disabled.

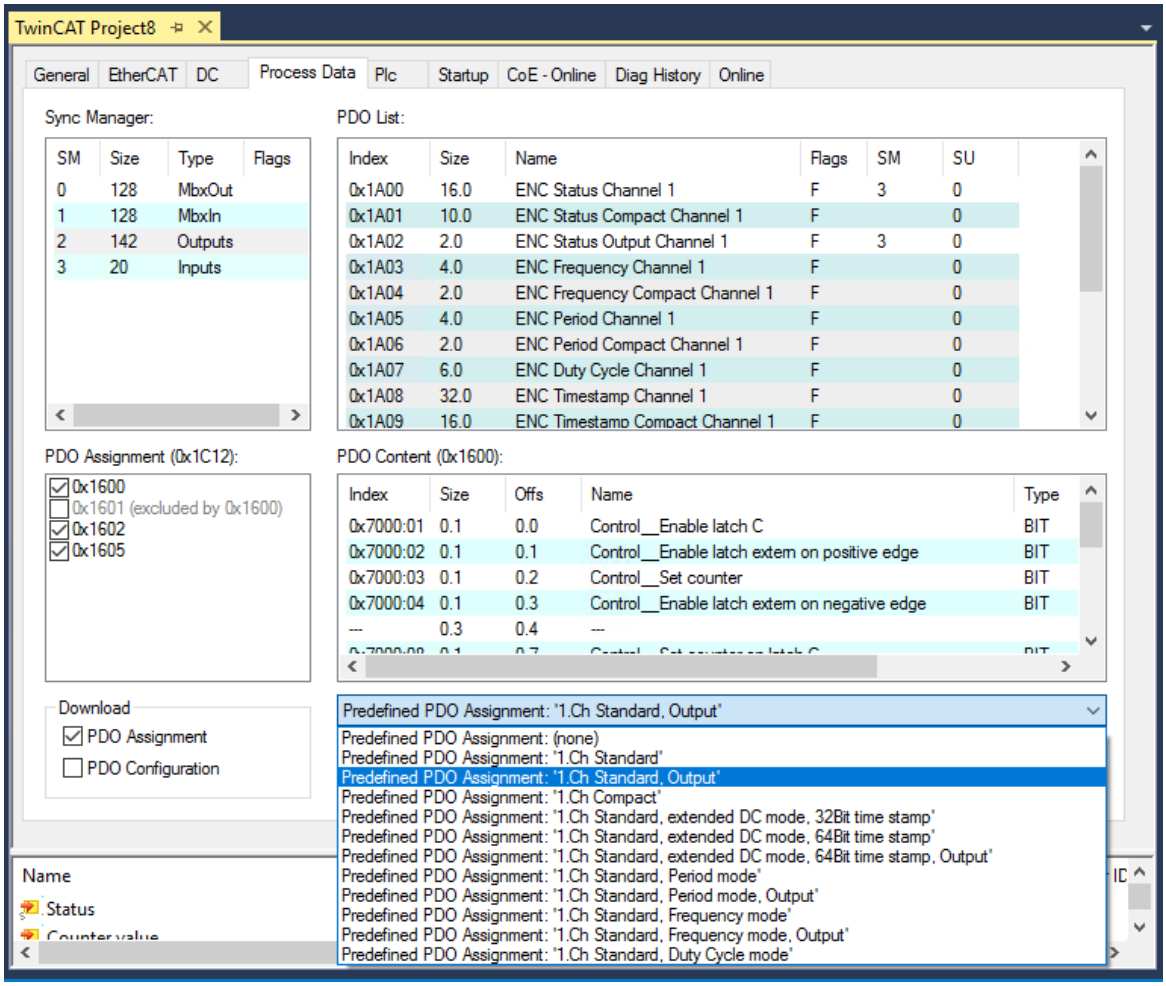


Fig. 154: EL5131 - Process data, Predefined PDO (default: Ch 1 Standard Output)

11 Predefined PDO assignments are available in modes Standard and Compact.

Note on using the timestamp function

In order to be able to use the timestamp function, the terminal must be operated in "DC Synchronous" or "DC Synchronous (input based)" mode.

The following Predefined PDOs are available:

Predefined PDO Assignments for Compact mode

Predefined PDO Assignment	PDO assignment
1.Ch Compact	**SM3:** 0x1A01 [▶ 237] - ENC Status Compact Channel 1 (10.0) **SM2:** 0x1601 [▶ 232] - ENC Control Compact Channel 1 (4.0)

Predefined PDO Assignments for Standard mode

Predefined PDO Assignment	PDO assignment
1.Ch Standard	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0)
1.Ch Standard, Output	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) 0x1A02 [▶ 237] - ENC Status Output Channel 1 (2.0) 0x1A0D [▶ 239] - ENC Status Output Channel 2 (2.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0) 0x1602 [▶ 233] - ENC Control Output Channel 1 (68.0) 0x1605 [▶ 234] - ENC Control Output Channel 2 (68.0)
1.Ch Standard, extended DC mode, 32Bit time stamp	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) 0x1A03 [▶ 238] - ENC Frequency Channel 1 (4.0) 0x1A05 [▶ 238] - ENC Period Channel 1 (4.0) 0x1A07 [▶ 238] - ENC Duty cycle Channel 1 (6.0) 0x1A09 [▶ 239] - ENC Timestamp Compact Channel 1 (16.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0)
1.Ch Standard, extended DC mode, 64Bit time stamp	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) 0x1A03 [▶ 238] - ENC Frequency Channel 1 (4.0) 0x1A05 [▶ 238] - ENC Period Channel 1 (4.0) 0x1A07 [▶ 238] - ENC Duty cycle Channel 1 (6.0) 0x1A08 [▶ 238] - ENC Timestamp Channel 1 (32.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0)
1.Ch Standard, extended DC mode, 64Bit time stamp	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) 0x1A02 [▶ 237] - ENC Status Output Channel 1 (2.0) 0x1A03 [▶ 238] - ENC Frequency Channel 1 (4.0) 0x1A05 [▶ 238] - ENC Period Channel 1 (4.0) 0x1A07 [▶ 238] - ENC Duty cycle Channel 1 (6.0) 0x1A08 [▶ 238] - ENC Timestamp Channel 1 (32.0) 0x1A0A [▶ 239] - ENC Timestamp Output Channel 1 (64.0) 0x1A0D [▶ 239] - ENC Status Output Channel 2 (2.0) 0x1A15 [▶ 240] - ENC Timestamp Output Channel 2 (64.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0) 0x1602 [▶ 233] - ENC Control Output Channel 1 (68.0) 0x1605 [▶ 234] - ENC Control Output Channel 2 (68.0)
1.Ch Standard, Period mode	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) 0x1A05 [▶ 238] - ENC Period Channel 1 (4.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0)
1.Ch Standard, Period mode, Output	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) 0x1A02 [▶ 237] - ENC Status Output Channel 1 (2.0) 0x1A05 [▶ 238] - ENC Period Channel 1 (4.0) 0x1A0D [▶ 239] - ENC Status Output Channel 2 (2.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0) 0x1602 [▶ 233] - ENC Control Output Channel 1 (68.0) 0x1605 [▶ 234] - ENC Control Output Channel 2 (68.0)

Predefined PDO Assignment	PDO assignment
1.Ch Standard, Frequency mode	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) 0x1A03 [▶ 238] - ENC Frequency Channel 1 (4.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0)
1.Ch Standard, Frequency mode, Output	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) 0x1A02 [▶ 237] - ENC Status Output Channel 1 (2.0) 0x1A03 [▶ 238] - ENC Frequency Channel 1 (4.0) 0x1A0D [▶ 239] - ENC Status Output Channel 2 (2.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0) 0x1602 [▶ 233] - ENC Control Output Channel 1 (68.0) 0x1605 [▶ 234] - ENC Control Output Channel 2 (68.0)
1.Ch Standard, Duty cycle mode,	**SM3:** 0x1A00 [▶ 236] - ENC Status Channel 1 (16.0) 0x1A07 [▶ 238] - ENC Duty cycle Channel 1 (6.0) **SM2:** 0x1600 [▶ 231] - ENC Control Channel 1 (6.0)

8.1.4 Synchronicity mode

The terminal can be operated in three different operation modes. Depending on the operation mode, the process data are synchronized by the terminal at different times and made available to the EtherCAT frame. For further information, please refer to the EtherCAT system documentation chapter Distributed Clocks -> Basic priciples.

The following operating modes are available for selection in the "DC" tab:

Operation mode	Description
FreeRun / SM Synchron	Cyclic, frame-triggered exchange of process data. An Ethernet frame triggers the process data provision for the next retrieving frame.
DC Synchronous	Cyclically constant determination of the counter value by the integrated distributed clocks unit. The configuration corresponds to an output module. The local SYNC event is triggered shortly after the passage of the EtherCAT frame. This means that the output data just delivered is output immediately.
DC-Synchron (input based)	Cyclically constant determination of the counter value by the integrated distributed clocks unit. The configuration corresponds to an input module. The local SYNC event is triggered before the EtherCAT frame. This makes current input data available for onward transport.

8.1.5 EtherCAT cycle time depending on Predefined PDO Assignment

The EtherCAT cycle time depends on the selection of the process data to be transferred. The following table provides an overview of the recommended cycle time, depending on the "Predefined PDO Assignment".

- The specifications refer to a multiple of the "Base Time" to be set via the TwinCAT Master.
- If a faster cycle time is used, the process data 0x6002:0F "Input Cycle Counter" must be used to monitor when new process data are delivered.

EL5131 - Predefined PDO Assignment	EtherCAT cycle time	
	Min.	Max.
1. Ch. Standard	100 µs typ.	150 µs typ.
1. Ch. Standard, Output	187.5 µs typ.	300 µs typ.
1. Ch. Compact	100 µs typ.	150 µs typ.
1. Ch. Standard, extended DC mode, 32Bit time stamp	125 µs typ.	200 µs typ.
1. Ch. Standard, extended DC mode, 64Bit time stamp	150 µs typ.	200 µs typ.
1. Ch. Standard, extended DC mode, 64Bit time stamp, Output	350 µs typ.	500 µs typ.
1. Ch Standard, Period mode	100 µs typ.	150 µs typ.
1. Ch. Standard, Period mode, Output	187.5 µs typ.	333 µs typ.
1. Ch. Standard, Frequency mode	100 µs typ.	150 µs typ.
1. Ch. Standard, Frequency mode, Output	187.5 µs typ.	333 µs typ.
1. Ch. Standard, Duty Cycle mode	125 µs typ.	150 µs typ.

8.1.6 Illustration of the process data and structural contents

EL5131 - Illustration of the process data and structural contents

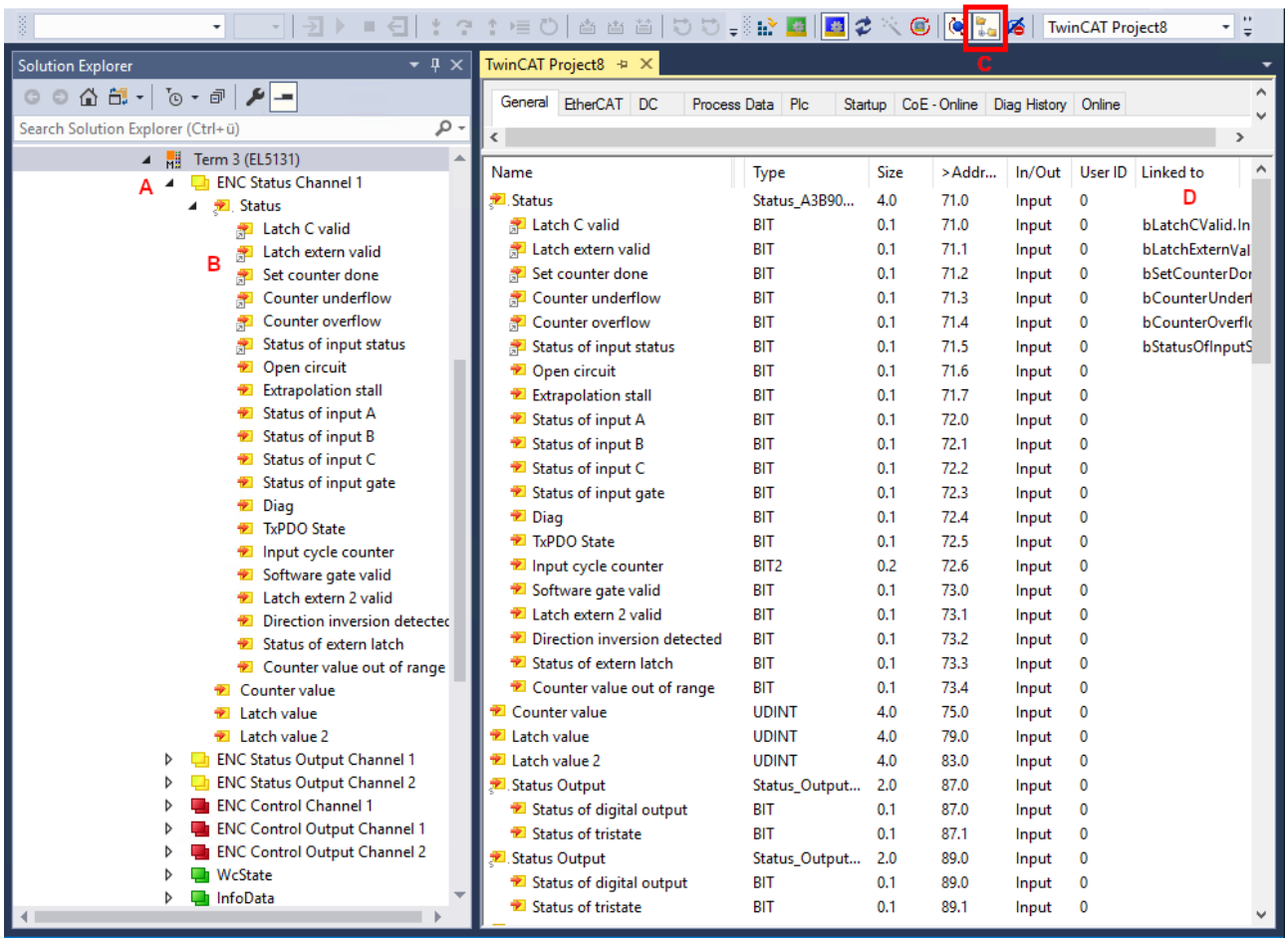


Fig. 155: Online representation of the process data and structural contents in the System Manager

The plain text display of the bit meanings is particularly helpful for commissioning, but also for linking with the PLC program.
By right-clicking on the status variable in the configuration tree (A), the structure can be opened for linking (B).
Activation of the *Show Sub Variables* button (C) displays all subvariables and links to the PLC (D).

8.2 Overview of functions

The EL5131 offers a wide range of functions. The following tables give an overview. The detailed description can be found in the corresponding chapters.

Basic functions	Description
Selection of encoder type [▶ 148]	Encoders or counters/pulse generators with signal levels according to RS422, TTL or OpenCollector can be selected.
Encoder operating voltage [▶ 149]	The encoder supply can be set separately for each channel to either 5 V_{DC}, 12 V_{DC} or 24 V_{DC}.
Evaluation of the counter value [▶ 149]	The input signal can be subject to 4-fold, 2-fold or 1-fold evaluation.
Counter limits [▶ 149]	The value range within which counting takes place can be defined.
Counter overflow / counter underflow [▶ 151]	If the count limit is exceeded or underrun, this is indicated in separate process data.
Counting direction [▶ 157]	The counting direction can be adapted to the application.
Reversion of rotation [▶ 152]	The counting direction and a reversion of rotation can be detected and output via the process data. This function can also be used for standstill monitoring [▶ 158].
Reset counter value [▶ 154]	Allows a recurring reset of the counter value via zero pulse C or an edge (positive/negative) at the Latch extern input.
Set counter value [▶ 155]	The counter value can be set at runtime to a predefined counter value via a PLC variable, zero pulse C or an edge (positive/negative) at the Latch extern input.
Save counter value [▶ 159]	The current counter value can be saved, independent of the cycle time, in separate process data via an edge (positive/negative) at the Latch extern and Gate/Latch input or the zero pulse C. You can parameterize whether the function is executed at each external edge or only once after each activation. By using both latch inputs a workpiece measurement [▶ 162] can also be realized.
Lock counter value [▶ 163]	The counter value can be disabled via an edge (positive/negative) at the Gate input or a PLC variable.

To use the extended functions, the corresponding functions must be selected via the "Predefined PDO Assignment [▶ 141]". The assignment is described in chapter "Process data".

Extended functionalities	Description
Frequency measurement [▶ 164]	The average frequency of the input signal within a given timeframe can be output directly.
Period value measurement [▶ 166]	The period value of the last period within the PLC cycle can be output directly.
Velocity, speed calculation [▶ 167]	The average velocity or speed of the input signal within a specified timeframe can be output directly.
Duty cycle measurement [▶ 169]	The ratio of pulse duration t_{ON} to period value T in the last PLC cycle can be output directly.
Timestamp [▶ 172]	A timestamp based on the distributed clocks system (DC) can be output for the last counting pulse, the zero pulse and the Latch extern and Latch extern 2 inputs.
Microincrements [▶ 170]	Allows additional increments (256 steps) to be interpolated between the counted encoder increments, thus increasing the resolution of the counter value.
Adjustable interference pulse filters [▶ 173]	To suppress interference, a filter can be set for each of the input signals.
Diagnostic data [▶ 202]	Error messages are communicated to the EtherCAT Master / TwinCAT via "DiagMessages".

Output function	Description
Switching characteristics [▶ 178]	It is possible to choose between push-pull and push-tristate (high-resistance switchable) mode.
Switching via PLC variable [▶ 181]	The output can be switched via a PLC variable from the application.
Switching function for counter value [▶ 184]	• The output can be switched independently of the PLC cycle when a preset counter value is reached. • The output can be switched for a defined time, independent of the PLC cycle, when a preset counter value is reached (see Switching for defined time [▶ 184]). • The output can be switched in connection with the TwinCAT NC and the function block “MC_DigitalCamSwitch_EL5131 [▶ 193]”.
Switching function for frequency or velocity value [▶ 189]	The output can be switched when a preset frequency or velocity value is reached, independent of the PLC cycle.
State in case of bus error [▶ 179]	Behavior of the output at bus error can be defined

Process data	Description
Operation modes [▶ 141]	The scope of the process data can be selected via "Predefined PDO Assignment".
Synchronicity mode [▶ 144]	In addition to the frame-triggered operation mode (SM mode), a mode based on synchronization via the distributed clocks is available.

Description of the inputs	Description
Zero pulse C [▶ 195]	With incremental encoders, a full revolution is marked by a special marker of the zero pulse C. This can be used for resetting, setting and storing the counter value.
Latch extern [▶ 197]	A Latch input is provided for 24 V_{DC} signals with a min. pulse duration t_{ON} of > 1 µs. This can be used for resetting, setting and storing the counter value.
Gate/Latch [▶ 198]	An external gate input is provided for 24 V_{DC} signals with a minimum pulse duration of t_{ON} > 1 µs. This can be used for resetting, setting and storing the counter value.
Status Input [▶ 200]	If the encoder has an alarm output, this can be evaluated.

Description of the outputs	Description
Digital output 1, digital output 2 [▶ 176]	The outputs can be switched via a PLC variable or when a counter value, frequency or velocity value is reached, independent of the PLC cycle.

8.3 Basic functions

8.3.1 Counter value

The counter value is displayed in index 0x60n0:11 "Counter value". It indicates the current counter value in the terminal. You can adapt it to the application using the following settings.

- Selection of encoder type [▶ 148] (Counter mode)
- Evaluation of the counter value [▶ 149] (Evaluation mode)
- Determining the counter limits [▶ 149] (Reset counter value / Limit counter value)
- Reversion of rotation [▶ 152]

8.3.1.1 Selection of encoder type (Counter mode)

To ensure a correct acquisition of the counter value, you must first select the appropriate encoder type.

The connected encoder is selected via index 0x80n1:1D "Counter mode". Explanations of the supported encoder and signal types can be found in chapter "Supported Encoder / Signal Types".

Selection of encoder type via index 0x80n1:1D “Counter mode” (n = 0 for Ch.1, n = 2 for Ch.2, dependent on the number of channels)	
Value	**Meaning**
0: Encoder RS422 (diff. input)	RS422 encoder with or without zero pulse
1: Counter RS422 (diff. input)	RS422 counter/pulse generator, with or without zero pulse, Direction preset via track B
2: Encoder TTL (single-ended)	TTL encoder with or without zero pulse
3: Counter TTL (single-ended)	TTL counter/pulse generator with or without zero pulse, Direction preset via track B
4: Encoder open collector	Open Collector encoder with or without zero pulse
5: Counter open collector	Open Collector counter/pulse generator with or without zero pulse, direction setting via track B

In the following sections, the settings under subindices 0, 2 and 4 are generally referred to as "encoder" and the settings under subindices 1, 3 and 5 as "counter/pulse generator".

Activating error detection for the C track

In delivery state, error detection on the C track is switched off; the default setting for index 0x80n0:0D "Error detection C" is FALSE.

If an encoder or counter/pulse generator with zero pulse C is used, index 0x80n0:0D "Error detection C" can be set to TRUE at the module. The corresponding LED will then also indicate the error.

8.3.1.2 Encoder operating voltage (supply voltage)

The encoder supply is generated internally from the 24 V of the power contacts. The encoder supply can be set in index 0x8001:17 [▶ 221] "Supply voltage". An operating voltage of 5 V_{DC} is preset. Voltage values of 5 V_{DC}, 12 V_{DC} and 24 V_{DC} can be selected.

The following tolerances apply

Voltage range	Tolerance
5 V_{DC}	+/- 5% (4.75 V … 5.25 V)
12 V_{DC}	+/- 10% (10.8 V … 13.2 V)
24 V_{DC}	-15 % to +20 % (20.4 V … 28.8 V)

NOTICE

Setting the encoder supply voltage

- Before switching to a higher voltage, make sure that the connected encoder supports the selected voltage range!
- Note the output power limitation (5 V_{DC} to 0.3 A, 12 V_{DC} to 0.15 A, and 24 V_{DC} to 0.3 A).
- To write to 0x8001:17 "Supply voltage" you have to set the value 0x72657375 (ASCII: "user") in index 0xF008 [▶ 243] "Code word".

8.3.1.3 Evaluation of the counter value

Evaluation of the counter value is set through index 0x8000:06 "Evaluation mode".

- Evaluation of the input signals (Index 0x8000:06 "Evaluation mode"): The input signal can be subject to 4-fold, 2-fold or 1-fold evaluation.
  - 1-fold evaluation: the rising edges on track A are counted.
  - 2-fold evaluation: the rising and falling edges on track A are counted.
  - 4-fold evaluation: the rising and falling edges on track A and track B are counted.

In the delivery state 4-fold evaluation is selected, as this enables the highest resolution of the input signal.

8.3.1.4 Determining the counter limits (Reset counter value / Limit counter value)

Counter limits in delivery state

In delivery state the counter value counts in the range from 0 to the maximum counter depth. When the maximum counter depth is exceeded (counter overflow) the counter starts counting up from zero again. Exceeding the counter is indicated by the "Counter overflow" bits (see chapter Determining the counter limits (Reset counter value / Limit counter value) [▶ 151]).

- PDO Assignment "Standard": 0 to 2^{32}-1
- PDO Assignment "Compact": 0 to 2^{16}-1

When a counter underflow occurs, the count continues at the maximum counter depth even if e.g. "Enable extern reset" is enabled in index 0x80n0:02. The underflow is identified with the corresponding "Counter underflow" bit.

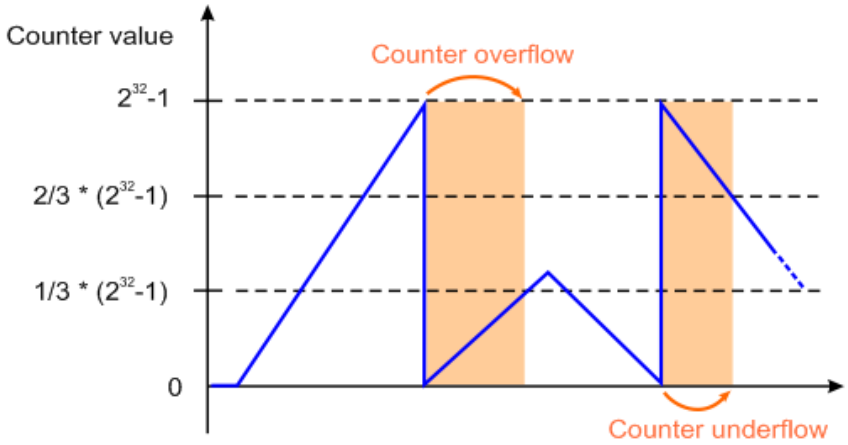


Fig. 156: Counter overflow and underflow in delivery state

Counter limits with defined value range

If counting is only to take place in a defined value range, the counter limits can be adjusted.

Set counter limits

For writing to index 0x8001:1B "Reset counter value" and index 0x8001:1A "Limit counter value" the value 0x72657375 (ASCII: "user") must be set in 0xF008 [▶ 243] "Code word".

- Enter the lower counter limit in index 0x8001:1B "Reset counter value".
- Enter the upper counter limit in index 0x8001:1A "Limit counter value".

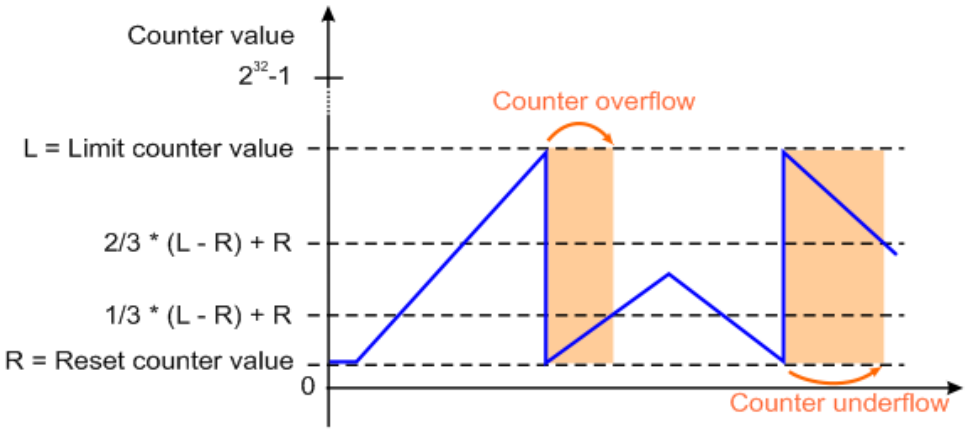


Fig. 157: Counter overflow and underflow with defined value range

Counter limits

- The lower counter limit index 0x8001:1B "Reset counter value" must always be smaller than the upper counter limit 0x8001:1A "Limit counter value".
  If this is not the case, the last entered value is not accepted.
- If the lower counter limit in index 0x8001:1B "Reset counter value" is greater than 0, the Motion Control application (NC/CNC) can only be used in a limited range.

If the counter is parameterized to a counter value outside the counter limits during runtime, this counter value is applied. Exceeding the counter limit is indicated via the process data in index 0x6002:15 "Counter value out of range".

Example:

A value is parameterized in 0x7000:11 "Set counter value" that is out of counter range and the "Set counter" bit in index 0x7000:03 is enabled.
→ The value specified in index 0x7000:11 is accepted.
→ The "Counter value out of range" bit in index 0x6002:15 is set.

If the counter limits are parameterized in such a way that the current counter value is outside these limits, the underflow/overflow of the counter limit is also shown via the process data in index 0x6002:15 "Counter value out of range".

Counter overflow and Counter underflow

An overflow or underflow of the counter limits is indicated by the process data 0x6000:04 "Counter underflow" or 0x6000:05 "Counter overflow".

- The "Counter underflow" bit in index 0x6000:04 is set when an underflow 0x8001:1B "Reset counter value" → 0x8001:1A "Limit counter value" occurs.
  - With the preset parameters, this corresponds to "..00 → ..FF"
    It is reset if 2/3 of the counter range are underrun.
  - For counter limits with fixed value range, this corresponds to:
    2/3 * ("Limit counter value" - "Reset counter value") + "Reset counter value"
- The "Counter overflow" bit 0x6000:05 is set when an overflow 0x8001:1A "Limit counter value" → 0x8001:1B "Reset counter value" occurs.
  - With default parameters "..FF → ..00"
    It is reset if 1/3 of the counter range is exceeded.
  - For counter limits with fixed value range, this corresponds to:
    1/3 * ("Limit counter value" - "Reset counter value") + "Reset counter value"

Example 1:

0x8001:1A "Limit counter value" = 2^{12}-1 = 4095

0x8001:1B "Reset counter value" = 0

"Counter underflow" bit is reset when: 2/3 * 4095 = 2730 is reached.

"Counter overflow" bit is reset when: 1/3 * 4095 = 1365 is reached.

Example 2:

0x8001:1A "Limit counter value" = 2^{12}-1 = 4095

0x8001:1B "Reset counter value" = 400

"Counter underflow" bit is reset when: 2/3 * (4095-400) +400 = 2463 is reached.

"Counter overflow" bit is reset when: 1/3 * (4095-95) +400 = 1232 is reached.

8.3.1.5 Reversion of rotation

- **With an encoder**, the counting direction is determined by the phase position of the signals on tracks A and B.
  - Forward (cw): Signal on track A leads track B by 90 °
  - Backward (ccw): Signal on track A lags track B by 90 °

To adapt the counting direction to the application, this logic can be inverted by setting the bit in index 0x8000:0E "Reversion of rotation".

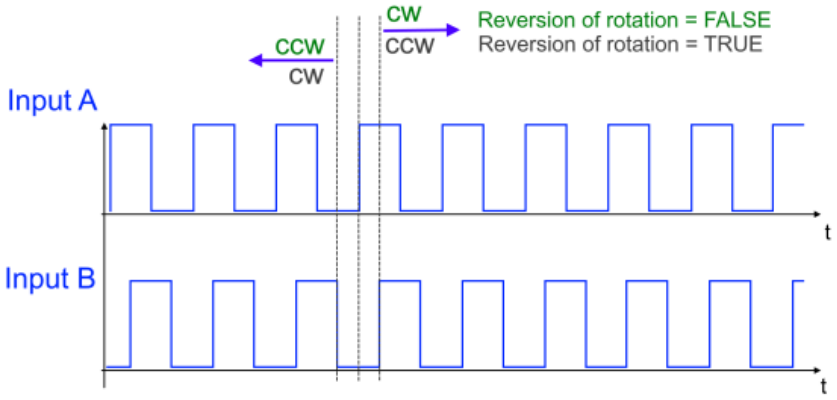


Fig. 158: Reversion of rotation (Index 0x8000:0E "Reversion of rotation") for an encoder

- **With a counter/pulse generator** the counting direction is determined by the level at track B.
  - Forward (cw): LOW level at track B or input open
  - Backward (ccw): HIGH level at track B

Setting the bit in index 0x8000:0E "Reversion of rotation" also inverts the logic of the counting direction. An overview of the resulting counting direction is shown in the following table.

The current level at input B is displayed via process data 0x6000:0B "Status of input B".

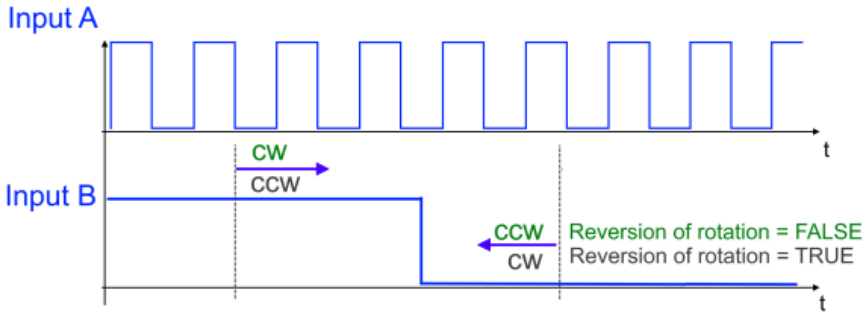


Fig. 159: Reversion of rotation (Index 0x8000:0E "Reversion of rotation") for a counter/pulse generator

0x8001:1D "Counter mode"	Level at input track B	0x80n0:0E "Reversion of rotation"	Resulting counting direction
1: Counter RS422 (diff. input)	Input open / LOW level	FALSE	positive
		TRUE	negative
	RS422 signal level	FALSE	negative
		TRUE	positive
3: Counter TTL (Single-Ended)	Input open / LOW level, voltage level < 0.8 V	FALSE	positive
		TRUE	negative
	TTL voltage level 2.0 V to 6.0 V, with min. 2.1 mA current	FALSE	negative
		TRUE	positive
5: Counter open collector	Input open / LOW level, voltage level < 0.8 V	FALSE	positive
		TRUE	negative
	Open collector voltage level 2.0 V to 6.0 V, with min. 2.1 mA current	FALSE	negative
		TRUE	positive

Status LED input B as counter RS422 (diff. input)

If the input of track B is open in counter mode "RS422 (diff. Input)", a wire break is detected and the status LED of input B lights up red. The error detection and indication via the LED can be disabled via index 0x8000:0C "Error detection B".

8.3.2 Counter value reset

A recurring reset of the counter value (index 0x6000:11 [▶ 223] "Counter value") to "0" can be done in the following ways:

- Input zero pulse C: [▶ 154] a positive edge at the zero pulse C input ("Enable C reset")
- Input Latch: [▶ 154] an edge (positive/negative) at the Latch extern input ("Enable extern reset")

The settings are made in the configuration data, so it is not necessary to reactivate the device after a reset.

The functions "Enable C reset" (index 0x8000:01 [▶ 219]) and "Enable extern reset" (index 0x80n00:02) cannot be reset simultaneously.

Counter value reset via the zero pulse C input (Enable C reset)

The counter value can be set to the value specified in index 0x8001:1B [▶ 221] "Reset counter value" for every full revolution of the encoder via the zero pulse C.

- Presetting of the reset value via index 0x8001:1B [▶ 221] "Reset counter value", (Default: 0)
- To activate this function set the bit in index 0x8000:01 [▶ 219] "Enable C reset".
- There is no status message via the process data.

Counter value reset via the Latch extern input (Enable extern reset)

The counter value can be set to "0" via the Latch extern input.

- Presetting of the reset value via index 0x8001:1B [▶ 221] "Reset counter value", (Default: 0)
- To activate this function set the bit in index 0x8000:02 [▶ 219] "Enable extern reset".
- Index 0x8000:10 [▶ 219] "Extern reset polarity" can be used to specify at which edge the Latch extern input is active.
  - 0: "Fall" - the counter is set to "0" with a falling edge
  - 1: "Rise" - the counter is set to "0" with a rising edge

There is no status message via the process data.

8.3.3 Set counter value

The counter value can be set to a predefined value at runtime and thus be used for synchronization with other processes. The preset can be activated in the following ways:

- PLC variable: [▶ 155] the counter value can be set from the PLC application (Set counter value)
- Zero pulse C input: [▶ 155] via the zero pulse C of the encoder (Set counter on latch C)
- Latch input: [▶ 156] a positive or negative edge at the Latch extern input (Set counter on latch extern on positive/negative edge)

Multiple activation of "Set counter on ..."

If several commands are activated to accept the preset counter value, only the command that is set first is accepted by the terminal. All other commands are ignored but remain activated, depending on the system.

- The counter value is set to the preset counter value at the first activated event and confirmed by the bit "Set counter done" (index 0x60n0:03 [▶ 223]).
- The counter value specification cannot be reactivated until all activated commands for the transfer of the counter value have been deactivated. This is confirmed by setting the "Set counter done" bit (index 0x60n0:03) to FALSE.

Set counter value via a PLC variable (Set counter value)

The counter value can be set to a predefined value during runtime via the process data (0x70n0:03 "Set counter"). In the PLC this bit can be linked to a digital input, for example, or used directly as a variable.

- Presetting the counter value via index 0x70n0:11 "Set counter value"
- Activation of the counter value setting via the PLC variable: Index 0x70n0:03 "Set counter"
- For confirmation the "Set counter done" bit in index 0x60n0:03 is set to TRUE.
- The counter value setting cannot be reactivated until index 0x70n0:03 "Set counter" has been set to FALSE.

Set counter value via input zero pulse C (Set counter on Latch C)

The counter value can be set to a preset value during runtime via the process data by means of the zero pulse C.

- Counter value setting via index 0x70n0:11 [▶ 225] "Set counter value"
- Activation of the counter value setting via zero pulse C:
  - Index 0x70n0:08 [▶ 225] "Set counter on latch C" = TRUE
    With the next zero pulse the counter value is set to the counter value specified in index 0x70n0:11 [▶ 225] (Set counter value).
- For confirmation the "Set counter done" bit in index 0x60n0:03 is set to TRUE.
- The counter value setting via the zero pulse C cannot be reactivated until the "Set counter on latch C" bit in index 0x70n0:08 is set to FALSE.

Set counter via the Latch extern input (Set counter on latch extern on positive/negative edge)

The counter value can be set to a preset value during runtime via the process data by the positive or negative edge at the Latch extern input.

- Counter value setting via index 0x70n0:11 "Set counter value"
- Activation of the counter value setting via
  - the positive edge at the Latch extern input:
    index 0x70n0:0A "Set counter on latch extern on positive edge"
  - the negative edge at the Latch extern input:
    index 0x70n0:0B "Set counter on latch extern on negative edge"
- If the bit is set (TRUE) in index 0x70n0:0A or 0x070n0:0B, the counter value is set to the specified value at the next rising or falling edge at the Latch extern input.
- For confirmation the "Set counter done" bit (index 0x60n0:03) is set to TRUE.
- The counter value specification cannot be reactivated until index 0x70n0:0A/0B "Set counter on latch extern on positive/negative edge" is set to FALSE.

8.3.4 Detect counting direction

The terminal can be parameterized as follows via index 0x8001:1C "Direction inversion hysteresis":

- Detect counting direction [▶ 157]: positive direction of rotation is output as "0", negative direction of rotation as "1" in the process data via index 0x6002:13 "Direction inversion detected".

or

- Detect reversion of rotation: [▶ 158] If direction inversion is detected, this is output in the process data via index 0x6002:13 "Direction inversion detected". This function can also be used for standstill monitoring [▶ 158].

Detect counting direction

- Setting the hysteresis, via index 0x8001:1C "Direction inversion hysteresis" = 0
- To detect the counting direction, the position value of the last PLC cycle (pos 1) is subtracted from the position value of the current PLC cycle (pos 2).
- The result is output during the next PLC cycle via index 0x6002:13 "Direction inversion detected". A positive result is interpreted as a positive direction of rotation (0x6002:13 "Direction inversion detected" = 0), a negative result as a negative direction of rotation (0x6002:13 "Direction inversion detected" = 1).

0x8001:1C "Direction inversion hysteresis" = 0		
Change in position	**Direction of rotation**	**0x6002:13 "Direction inversion detected"**
pos 2 - pos 1 > 0	Positive direction of rotation (CW)	0
pos 2 - pos 1 < 0	Negative direction of rotation (CCW)	1
pos 2 - pos 1 = 0	No change of direction of rotation	No state change
Counter overflow	No change of direction of rotation	No state change
Counter underflow	No change of direction of rotation	No state change

- When the terminal is started up, a positive direction of rotation is assumed in the first PLC cycle (0x6002:13 "Direction inversion detected" = 0), since a second position value is initially missing for the calculation.
- If an overflow (0x6000:05 "Counter overflow") or underflow (0x60n00:04 "Counter underflow") of the counter value is detected, the result is corrected internally for the next PLC cycle. See figure (A) below.
- If the counter value is changed externally via the C track or the Latch extern input [▶ 155] and this leads to a negative position jump, this is indicated via process data 0x6002:13 "Direction inversion detected". See figure (B) below
- If reversion of rotation [▶ 152] (index 0x8000:0E "Reversion of rotation" = TRUE) is enabled, the logic of the direction of rotation detection also changes.

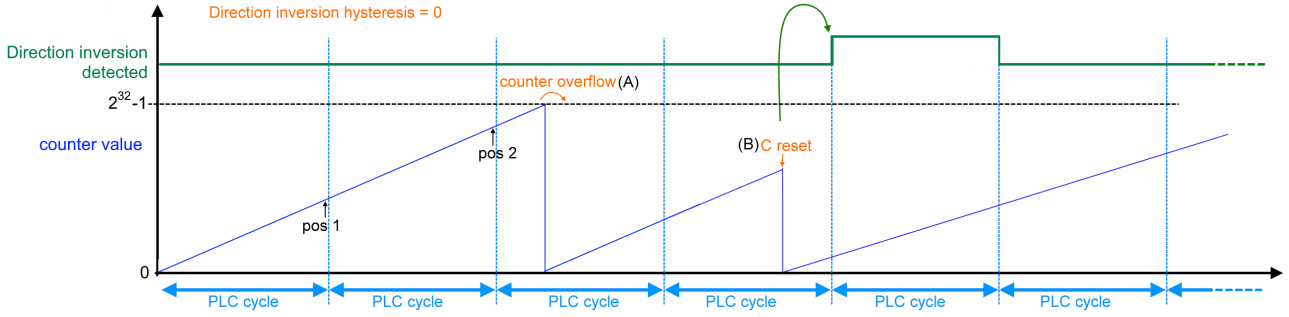


Fig. 160: Detection of the counting direction

Detect reversion of rotation

- Input of the hysteresis in number of increments, via index 0x8001:1C "Direction inversion hysteresis". A value greater than 0 must be selected.
- If a direction inversion is detected, an internal counter evaluates the edges detected directly one after the other for each direction of rotation and compares them with the number entered in index 0x8001:1C.
- If the counter value exceeds the value entered in index 0x8001:1C, the bit in index 0x6002:13 "Direction inversion detected" is set during the next PLC cycle. The bit remains active for one cycle only.
- The bit remains active if a further direction inversion is detected within the next cycle.
- The hysteresis is considered separately for each PLC cycle, so the counter is restarted for each cycle. There is no monitoring across PLC cycles.
- If reversion of rotation [▶ 152] (index 0x8000:0E "Reversion of rotation" = TRUE) is enabled, the logic of the direction of rotation detection also changes.
- If several changes of direction occur within one PLC cycle, the display only indicates that at least one change of direction has been detected.

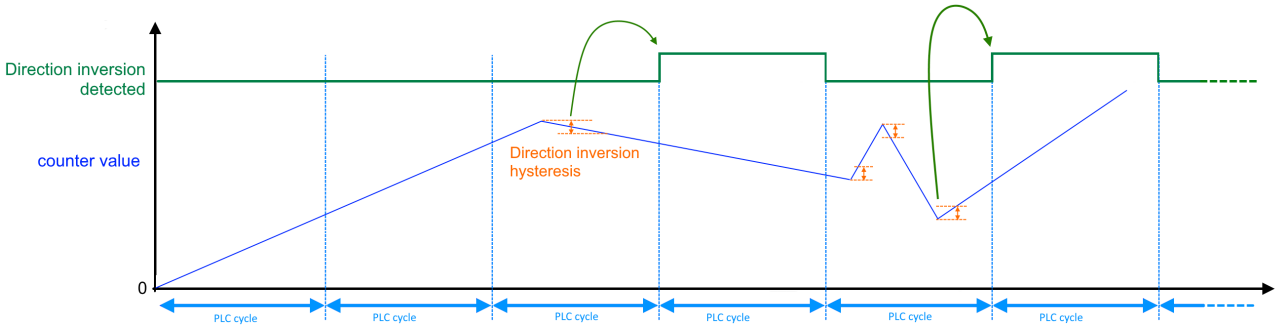


Fig. 161: Detection of reversion of rotation

Standstill monitoring

Detection of direction inversion can also be used for standstill monitoring. If an almost constant counter value is expected, the hysteresis (index 0x80n1:1C) can be used to monitor whether the value is within the window. A slight jitter of the counter value is therefore still accepted as a permissible value. If this value is exceeded, bit 0x6002:13 "Direction inversion detected" is set during the next PLC cycle.

- If the direction changes again within the hysteresis [number of increments] specified in index 0x8001:1C, no change of direction is indicated (standstill). See figure (A) below.
- If the hysteresis [number of increments] specified in index 0x8001:1C is exceeded after a change of direction, the change of direction is indicated in the next cycle by setting the "Direction inversion detected" bit in index 0x6002:13. See figure (B) below.

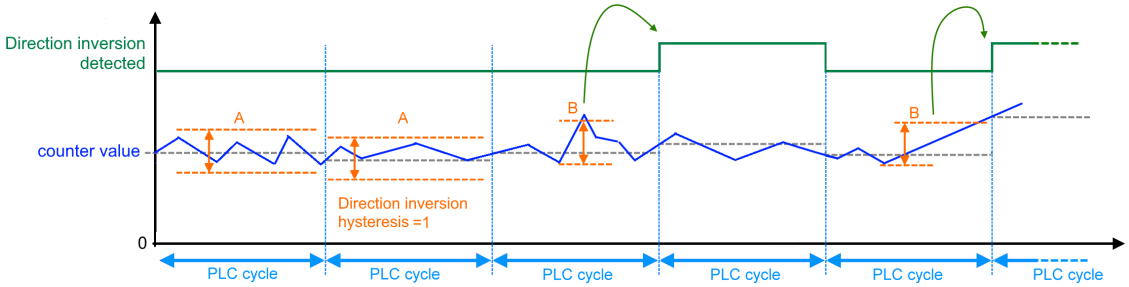


Fig. 162: Standstill monitoring with hysteresis and detection of reversion of rotation

Drifting of the counter value

Reliable detection of a drifting counter value can only be guaranteed if the value in index 0x8001:1C "Direction inversion hysteresis" is "1", or by additional monitoring of the counter value in the PLC.

8.3.5 Save counter value

The latch function enables the current counter value to be stored in separate process data, independent of the cycle time. The Gate/Latch input can be parameterized as a second external latch input with separate process data. The latch function can be triggered as follows:

- Latch input: [▶ 160] positive/negative edge at the Latch input
  (Enable latch extern on positive/negative edge)
- Gate/Latch input [▶ 161]: positive/negative edge at the Gate/Latch input
  (Enable latch extern 2 on positive/negative edge)
- Zero pulse C input: [▶ 162] via zero pulse C (Enable latch C)

Index 0x8000:22 [▶ 219] "Enable continuous latch extern" and index 0x8000:23 [▶ 219] "Enable continuous latch extern 2" can be used to parameterize whether the function is executed at every parameterized external edge at the Latch input or only once after every activation.

A workpiece measurement [▶ 162] can be realized by using two independent Latch inputs.

Multiple activation of the latch function

If several commands are activated simultaneously to save the counter value in the "Latch value" process data (index 0x60n0:12 [▶ 223]), only the command that is set first is accepted by the terminal.

- The counter value is stored in the "Latch value" (index 0x60n0:12) at the next occurring event and confirmed with the corresponding bit.
- All other activated events are ignored.
- The counter value storage cannot be reactivated until all activated commands for latching the value have been deactivated. This also applies if they were activated after confirmation by the occurring event.

Save counter value via a positive/negative edge at the Latch input (Enable latch extern on positive/ negative edge)

- The counter value at the Latch extern input can be saved via:
  - Index 0x7000:02 [▶ 225] "Enable latch extern on positive edge" = TRUE
    At the first external latch pulse with positive edge the current counter value is stored in index 0x6000:12 [▶ 223] "Latch value".
  - Index 0x7000:04 [▶ 225] "Enable latch extern on negative edge" = TRUE
    At the first external latch pulse with negative edge the current counter value is stored in index 0x6000:12 [▶ 223] "Latch value".
  - Simultaneous activation of 0x7000:02 and 0x7000:04
    On the first external Latch pulse, regardless of the polarity of the edge, the current counter value is stored in index 0x6000:12 "Latch value".
- Specification whether it is necessary to reactivate the command to save the counter value via:
  - "Enable continuous latch extern" index 0x8000:22 [▶ 219] = FALSE
    The following pulses at the latch extern input have no influence on the latch value in index 0x6000:12 "Latch value" if the bit in index 0x7000:02 or 0x7000:04 is set.
    Only when index 0x6000:02 [▶ 223] "Latch extern valid" is FALSE, a new counter value can be written to the latch input in index 0x60n0:12 "Latch value"
  - "Enable continuous latch extern" index 0x8000:22 [▶ 219] = TRUE
    The counter value is written to index 0x60n00:12 "Latch value" with each parameterized edge at the latch extern input.
    A new activation of index 0x7000:02 or 0x7000:04 is omitted.
- Saving of the counter value in index 0x6000:12 "Latch value" is confirmed via the "Latch extern valid" bit (index 0x6000:02).
- The status of the Latch extern input can be monitored via index 0x6002:14 [▶ 224] "Status of extern latch".

Simultaneous activation of "Enable continuous latch extern" and "Enable latch C"

When using 0x7000:02/0x7000:04 "Enable latch extern on positive/negative edge", by activating 0x8000:22 "Enable continuous latch extern" simultaneously with 0x7000:01 "Enable latch C"

- Only the first event, either pos./neg. edge at the Latch extern input or the zero pulse C, is written to index 0x6000:12 "Latch value".
- Saving of the counter value is confirmed by index 0x6000:02 "Latch extern valid" or index 0x6000:01 [▶ 223] "Latch C valid".
- All further edges at the Latch extern input are ignored.
- The counter value storage cannot be reactivated until all activated commands for latching the value have been deactivated.

Save counter value via a positive/negative edge at the Gate/Latch input (Enable latch extern 2 on positive/negative edge)

The Gate/Latch input may be used as a second latch input (Latch extern 2). To do this, the gate should be disabled.

Deactivation of the gate

Set the "Gate polarity" bit (index 0x8000:04 [▶ 219]) to "0 – Disable gate" in order to be able to use the Latch extern 2 input without disabling the counter value after the latch event.

- The counter value at the Latch extern 2 input can be saved via:
  - Index 0x7000:0C [▶ 225] "Enable latch extern 2 on positive edge" = TRUE
    At the first external latch pulse with positive edge at the Gate/Latch input the current counter value is stored in index 0x6000:22 [▶ 223] "Latch value 2".
  - Index 0x7000:0D [▶ 225] "Enable latch extern 2 on negative edge" = TRUE
    At the first external pulse with negative edge at the Gate/Latch input the current counter value is stored in index 0x6000:22 "Latch value 2".
  - Simultaneous activation of 0x7000:0C and 0x7000:0D
    On the first pulse at the Gate/Latch input, regardless of the polarity of the edge, the current counter value is stored in index 0x6000:22 "Latch value 2".
- Specification whether it is necessary to reactivate the command to save the counter value via:
  - "Enable continuous latch extern 2" Index 0x8000:23 [▶ 219] = FALSE
    The following pulses at the Gate/Latch input have no influence on the Latch value in index 0x6000:22 "Latch value 2" if the bit in index 0x7000:0C or 0x7000:0D is set.
    Only when index 0x6002:12 [▶ 224] "Latch extern 2 valid" is FALSE, a new counter value can be written to the Gate/Latch input in index 0x6000:22 "Latch value 2".
  - "Enable continuous latch extern 2" index 0x8000:23 [▶ 219] = TRUE
    The counter value is written to index 0x6002:12 "Latch value 2" for each parameterized edge at the Gate/Latch input.
    There is no need to reactivate index 0x7000:0C or 0x7000:0D.
- The status of the Gate/Latch input can be recorded via index 0x6000:0C [▶ 223] "Status of input gate".
- Saving of the counter value in index 0x6000:22 "Latch value 2" is confirmed via bit 0x6002:12 "Latch extern 2 valid".

Simultaneous use of Gate und Latch extern 2

When using index 0x8000:04 [▶ 219] "Gate polarity"(1 = "Enable pos. gate") and simultaneously activating index 0x7000:0C [▶ 225] "Enable latch extern 2 on positive edge", the current counter value is initially stored in index 0x6000:22 [▶ 223] "Latch value 2" when a positive edge is detected at the Gate/Latch input. The counter value is then blocked.

The same applies to the use of index 0x80n00:04 "Gate polarity" (2 = "Enable neg. gate") and simultaneous activation of index 0x7000:0D [▶ 225] "Enable latch extern 2 on negative edge" at a negative edge at the Gate/Latch input.

Save counter value via the zero pulse C input (Enable latch C)

The counter value can be stored at runtime via the process data and the zero pulse in "Latch value" (index 0x60n0:12 [▶ 223]).

- The function is activated by setting the bit in "Enable latch C" (index 0x70n0:01) [▶ 225] to TRUE.
- The current counter value is saved in "Latch value" (index 0x60n0:12) with the next zero pulse at input C. The subsequent pulses have no influence on the latch value.
- The "Latch C valid" bit (index 0x60n0:01 [▶ 223]) is set to TRUE.
- After reactivating "Enable Latch C" (index 0x70n0:01) a new counter value cannot be written to the Latch input until the values of the "Enable Latch C" bit (index 0x70n0:01) and the "Latch C valid" bit (index 0x60n0:01) are FALSE.

Workpiece measurement

The two latch functions (Latch extern and Latch extern 2) can be used to measure workpieces or distances between two workpieces. To use the Latch extern 2 function, the gate should be disabled via index 0x8000:04 [▶ 219] "Gate polarity" ("Disable gate" = 0).

Depending on the activation of the indices, the counter value may be stored via a rising or falling edge.

"Continuous latch" for workpiece measurement

To prevent the stored value from being overwritten, it is advisable to set the setting in index 0x8000:22 [▶ 219] "Enable continuous latch extern" to FALSE for the workpiece measurement.

Workpiece measurement sequence (example)

- Index 0x7000:02 [▶ 225] "Enable latch extern on positive edge" = TRUE
  At the first external latch pulse with positive edge at Latch extern input the current counter value is stored in index 0x6000:12 [▶ 223] "Latch value".
- Saving of the counter value in index 0x6000:12 "Latch value" is confirmed via bit 0x6000:02 [▶ 223] "Latch extern valid".
- Index 0x7000:0D [▶ 225] "Enable latch extern 2 on negative edge" = TRUE:
  At the first external pulse with negative edge at the Gate/Latch input the current counter value is stored in index 0x6000:22 [▶ 223] "Latch value 2".
- Saving of the counter value in index 0x6000:22 "Latch value 2" is confirmed via the "Latch extern 2 valid" bit in index 0x6002:12.
- The end of the measurement is confirmed via the two activated bits in index 0x6000:02 "Latch extern valid" and index 0x6002:12 "Latch extern 2 valid".
- The workpiece length can be calculated from the difference between the two values "Latch value" and "Latch value 2".
- A new workpiece measurement can be started after disabling the bits in index 0x7000:02 "Enable latch extern on positive edge" and index 0x7000:0D "Enable latch extern 2 on negative edge".

8.3.6 Lock counter value

The gate function enables locking of the counter (0x6000:11 [▶ 223]).
The counter is locked at the first pulse at the Gate/Latch input. Subsequent pulses have no influence on the counter value. This allows a timeframe to be defined in which counting signals are acquired. The gate function can be triggered by:

- Gate input: [▶ 163] a positive / negative edge at the Gate/Latch input (Enable pos./neg. gate),
- PLC variable: [▶ 163] the counter can be locked from the PLC application (Set software gate).

The Gate/Latch input may be used as a second latch input (Latch extern 2).

Lock counter value via a positive/negative edge at the gate input (Enable pos./neg. gate)

- The level at the Gate input at which the counter value is locked during runtime can be specified via index 0x8000:04 [▶ 219] "Gate polarity".
  - 0: Disable gate
    The Gate/Latch input is disabled. It can still be used as Latch extern 2 input.
  - 1: Enable pos. gate
    The counter value is locked with HIGH level at the Gate/Latch input.
  - 2: Enable neg. gate
    The counter value is locked with LOW level at the Gate/Latch input.
- The current level at the Gate input is displayed via process data 0x6000:0C [▶ 223] "Status of input gate".

Simultaneous use of Gate und Latch extern 2

When using index 0x8000:04 [▶ 219] "Gate polarity"(1 = "Enable pos. gate") and simultaneously activating index 0x7000:0C [▶ 225] "Enable latch extern 2 on positive edge", the current counter value is initially stored in index 0x6000:22 [▶ 223] "Latch value 2" when a positive edge is detected at the Gate/Latch input. The counter value is then blocked.

The same applies to the use of index 0x80n0:04 "Gate polarity" (2 = "Enable neg. gate") and simultaneous activation of index 0x7000:0D [▶ 225] "Enable latch extern 2 on negative edge" at a negative edge at the Gate/Latch input.

Locking the counter value via a PLC variable (Set software gate)

The counter value can be locked from the PLC application.

- Index 0x70n0:09 [▶ 225] "Set software gate" = TRUE
  The counter is locked.
- For confirmation the "Software gate valid" bit (0x60n2:11 [▶ 224]) is set to TRUE.
- Index 0x70n0:09 "Set software gate" = FALSE
  The counter is unlocked.

8.4 Extended functionalities

8.4.1 Frequency measurement

The terminal offers the option of calculating and outputting the frequency value in addition to the counter value. The following limit values apply to the determination of measured values:

Measured value output	Measured value limit		Comment
	lower	top	
Frequency value	0.095 Hz	5 MHz	With 4-fold evaluation Signal type: RS422 (diff. input)
	0.095 Hz	1 MHz	With 4-fold evaluation Signal type: TTL (single ended), open collector
	0.095 Hz	100 kHz	With 4-fold evaluation Signal type: open collector

Sequence of frequency measurement (frequency value)

The frequency is calculated from the number of increments or position value changes within a time interval. The result is output via the process data in index 0x6000:13 [▶ 223] "Frequency value".

Frequency = number of periods / time interval	
Number of periods	The number of complete periods on track A within the timeframe (0x8000:11 [▶ 219]) is measured. Counting starts with the first rising edge within the timeframe (index 0x80n0:11 "Frequency window").
Time interval	The time measurement starts with the first rising edge on track A within the "Frequency window" and ends with the last falling edge within the "Frequency window". The time is measured with a resolution of 10 ns. Special case: If no full period is measured within the timeframe (Frequency window), another time window is started. The maximum time measurement for recording an entire period is limited by the frequency wait time.

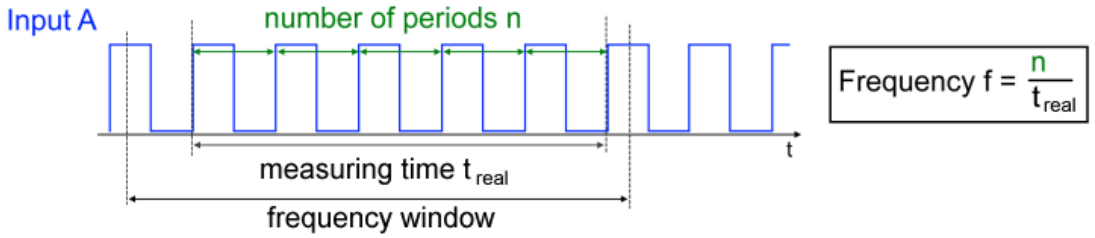


Fig. 163: Frequency measurement - "Frequency value"

- Selection of the frequency window in index 0x80n0:11 "Frequency window", default value 10 ms. Follow the instructions below to select the correct timeframe.
- "Frequency scaling" (index 0x8000:13 [▶ 219]) indicates the resolution of the output frequency. The default value is 0.01 Hz. The frequency can also be output in the unit 1 Hz (set 0x80n0:13 "Frequency scaling" to 1_{dec}). If other resolutions are required, the conversion can be done via 0x8000:1D [▶ 219] "Frequency numerator".
- The calculated frequency value is output in the process data via "Frequency value" (index 0x6000:13 [▶ 223]).
- The calculation is carried out acyclically and without reference to the distributed clock system and is therefore independent of the operation mode.
- Locking the counter value via the gate, a C-reset or external reset has no effect on the frequency calculation.

Correct selection of the timeframe:

In delivery state the timeframe for the frequency measurement is set to 10 ms.

The accuracy and size of the determined frequency depends on the size of the timeframe (0x8000:11 [▶ 219] "Frequency window"), which must be selected to match the application. The minimum value to be entered here is twice the period value of the smallest measured frequency.

$$\text{Frequency window} \geq 2 * \frac{1}{f_{min}}$$

At constant speeds, select a larger timeframe so that the best possible averaging can take place.

In situations where frequent positive or negative accelerations are encountered, select a smaller timeframe in order to be able to respond more quickly to changing position values. Alternatively, period value measurement can be used for situations with variable speeds.

Special case: frequency measurement

If no full period is detected within the "Frequency window" (see diagram (1) below), another time Frequency window is started in order to record at least one full period (see diagram (2) below). This happens until the maximum waiting time (0x8000:17 [▶ 219] "Frequency wait time") has elapsed. Until then the last valid value is output in index 0x6000:13 [▶ 223] "Frequency value".

If no full period is detected within the waiting time, the measurement is discarded and the output in index 0x6000:13 "Frequency value" is set to "0".

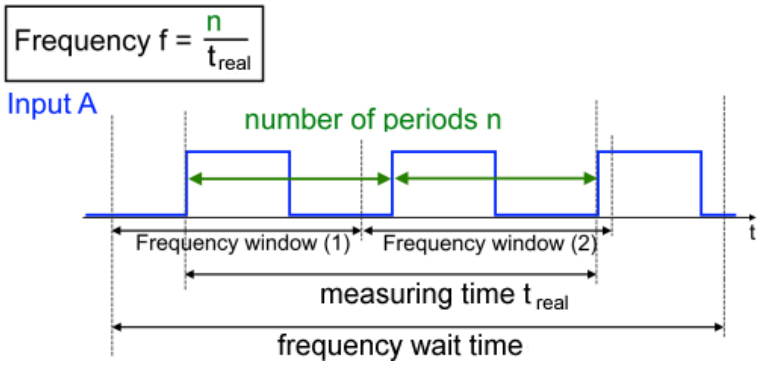


Fig. 164: Frequency measurement outside the frequency window with waiting time

Difference frequency and period value measurement

The period value is recalculated for each cycle and output in the process data. The frequency measurement always averages the measured periods over a timeframe. The "Period value" is therefore a current value, the "Frequency value" is an averaged value.

8.4.2 Period value measurement

The terminal offers the option of calculating and outputting the period value in addition to the counter value. The following limit values apply to the determination of measured values:

Measured value output	Measured value limit		Comment
	lower	top	
Period value	0.20 µs	21 s	With 4-fold evaluation Signal type: RS422 (diff. input)
	1 µs	21 s	With 4-fold evaluation Signal type: TTL (single ended), open collector

Period value measurement sequence (Period value)

- For the period value measurement, the time between two positive edges of input A is measured with a resolution of 10 ns.
- In each case the last period within the PLC cycle is considered and output as separate process data in index 0x6000:14 [▶ 223] "Period value".
- "Period scaling" (index 0x8000:14) indicates the resolution of the output period value. The default value is 10 ns. Alternatively the period value can be output in the unit 100 ns (0x8000:14 "Period scaling" set to 100_{dec}) or 500 ns (0x8000:14 "Period scaling" set to 500_{dec}).

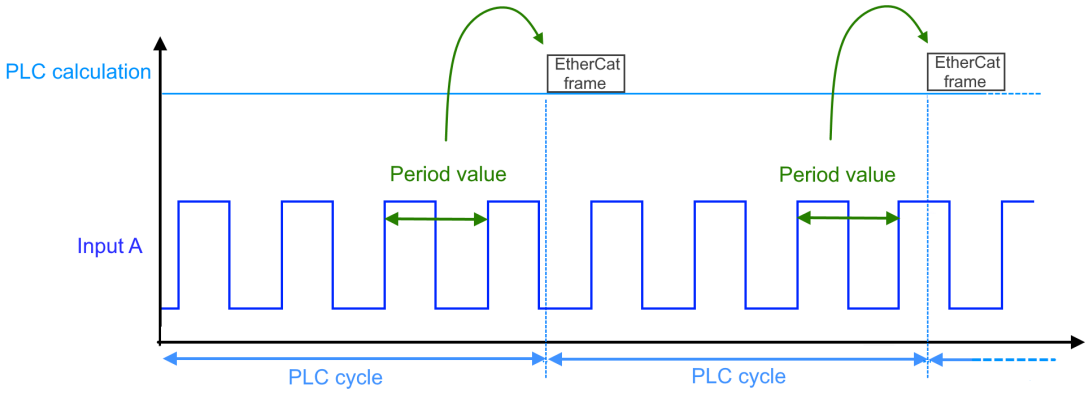


Fig. 165: Period value measurement

- If no full period is detected within a PLC cycle (two positive edges of input A), the last valid value is output.
- The period measurement continues across cycles. As soon as a new value is available, the process data is updated in index 0x6000:14 [▶ 223] "Period value".
- If no full period is detected within 21 seconds, the measurement is discarded and the output in index 0x6000:14 [▶ 223] "Period value" is set to the maximum value, 2147483648.

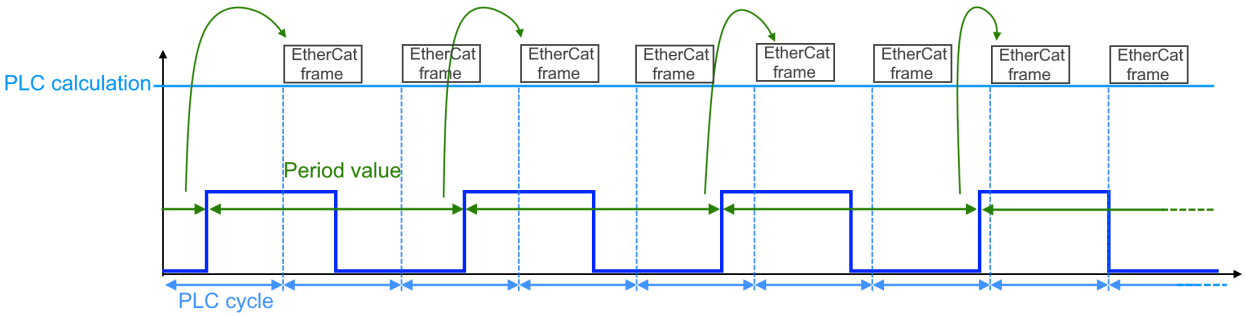


Fig. 166: Period value measurement for long periods

Difference frequency and period value measurement

The period value is recalculated for each cycle and output in the process data. The frequency measurement always averages the measured periods over a timeframe. The "Period value" is therefore a current value, the "Frequency value" is an averaged value.

8.4.3 Velocity, speed calculation

In addition to the counter value, the velocity and speed an also be measured. The measured values are determined through parameterization of the frequency values. The following limit values apply:

Measured value output	Measured value limit		Comment
	lower	top	
Frequency value	0.095 Hz	5 MHz	With 4-fold evaluation Signal type: RS422 (diff. input)
	0.095 Hz	1 MHz	With 4-fold evaluation Signal type: TTL (single ended)
	0.095 Hz	100 kHz	With 4-fold evaluation Signal type: open collector

The frequency measurement is carried out as described in chapter "Frequency measurement (Frequency value)". In addition, the frequency value can be converted using the two indices 0x8000:1D [▶ 219] "Frequency numerator" and 0x8000:1E [▶ 219] "Frequency denominator".

Index (hex)	Name	Meaning	Default
8000:1D	Frequency numerator	Frequency numerator, Factor for normalizing the frequency, e.g., the time unit	0x0000001 (1_{dec})
8000:1E	Frequency denominator	Frequency denominator, Factor for normalizing the frequency, e.g., velocity calculation	0x0000001 (1_{dec})

Thus, the frequency can be normalized for application-specific speed measurement. The measured value continues to be output in the process data via "Frequency value" (index 0x6000:13 [▶ 223]).

The calculated measured value has the following relationship to the frequency:

$$\text{Determination of measured values} = \text{Frequency} * \frac{\text{Frequency numerator}}{\text{Frequency denominator}}$$

Example for velocity calculation in m/s

- 200 mm travel corresponds to one full mechanical revolution of an encoder.
- The encoder has a 12-bit resolution, i.e. 4096 lines, which corresponds to 4096 periods on track A.

The velocity value is determined as follows:

$$v\left[\frac{mm}{s}\right] = \text{Frequency value}\frac{periods}{s} * \frac{200}{4096}\frac{mm}{periods}$$

$$v\left[\frac{m}{s}\right] = \text{Frequency value}\frac{periods}{s} * \frac{200}{4096}\frac{mm}{periods} * \frac{1}{1000}\frac{m}{mm}$$

Index (hex)	Name	Example: velocity in m/s	Comment
8000:1D [▶ 219]	Frequency numerator	0x00000C8 (200_{dec})	Travel path, conversion to m via "Frequency denominator"
8000:1E [▶ 219]	Frequency denominator	0x003E8000 (4096000_{dec})	Specification of the number of periods over the travel path and conversion of the travel path to m

Example Calculation of speed f_{rot} in revolutions/s

- After one full rotation, 4-fold evaluation (0x8000:06 [▶ 219] "Evaluation mode") results in a numerator value of 16384 increments. This corresponds to 4096 periods on track A.
- "Frequency numerator" can be used to convert the time unit into minutes. Note the scaling of the frequency value (0x8000:13 [▶ 219] "Frequency scaling"):

Conversion into minutes depending on "Frequency scaling			
Index (hex)	**Entry**	**Index (hex)**	**Entry**
0x8000:13 [▶ 219] Frequency scaling	100_{dec}: 0.01 Hz	0x8000:1D [▶ 219] Frequency numerator	0x00001770 (6000_{dec})
	1_{dec}: 1 Hz		0x0000003C (60_{dec})

$$f_{rot}\left[\frac{\text{rotations}}{\text{s}}\right] = \text{Frequency value}\frac{\text{periods}}{\text{s}} * \frac{6000}{4096}\frac{\text{rotations}}{\text{periods}}$$

$$f_{rot}\left[\frac{\text{rotations}}{\text{min}}\right] = \text{Frequency value}\frac{\text{periods}}{\text{s}} * 60\frac{\text{s}}{\text{min}} * \frac{6000}{4096}\frac{\text{rotations}}{\text{periods}}$$

Index (hex)	Ex. Speed f_{rot} in rpm	Comment
8000:1D [▶ 219] Frequency numerator	0x0001770 (6000_{dec})	Output of the value in 0.01 rpm
8000:1E [▶ 219] Frequency denominator	0x0001000 (4096_{dec})	Conversion factor in periods / m

8.4.4 Duty cycle evaluation

The duty cycle describes the ratio of pulse duration t_{ON} to period value T.

Duty cycle [0.01%] = Pulse duration t_{ON}/Period value T * 100

The following process data are available:

Index (hex)	Name	Meaning	Default
60n0:23	Duty cycle	Ratio of pulse duration t_{ON} to period value T. Output in 1/100	0x0000 (0_{dec})
60n0:24	Duty cycle min	Smallest measured duty cycle value during the current measurement	0x2710 (10000_{dec})
60n0:25	Duty cycle max	Largest measured duty cycle value during the current measurement	0x0000 (0_{dec})

The duty cycle is measured as follows:

- The pulse duration t_{ON} is measured. This is the time between the rising and falling edge of input A.
- The period value T is measured. This is the time between two positive edges of input A.
- The times are measured with a resolution of 10 ns.
- In each case the last period within the PLC cycle is considered and output as separate process data in index 0x60n0:23 "Duty cycle".
- The output is 1/100 of the measured value.
- The duty cycle measurement delivers valid results in RS422 mode up to 1 MHz with an accuracy of typically 1% - 2%. Larger measuring errors may occur at input frequencies above 1 MHz.

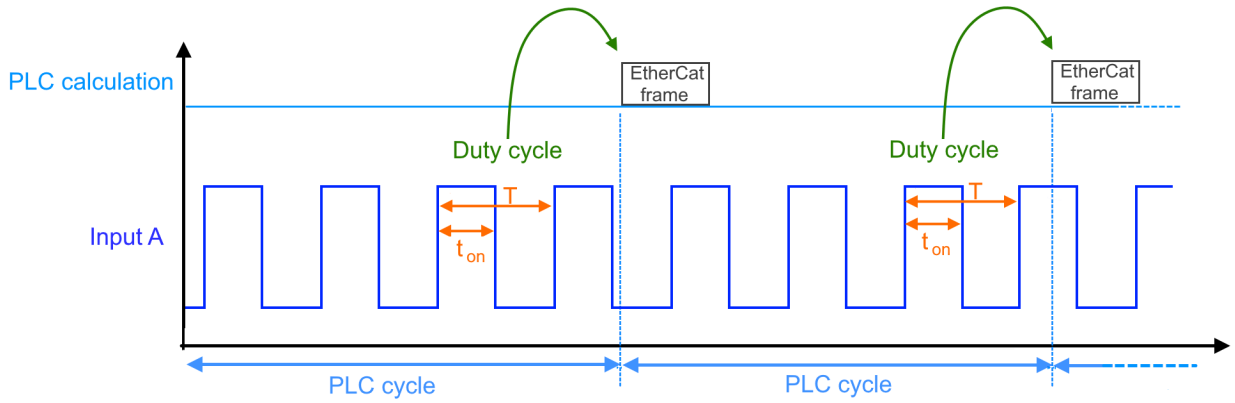


Fig. 167: Duty cycle measurement

The smallest and largest measured duty cycle value can be reset via the command object in index 0xFB00:01 "Request".

0xFB00:01 "Request"	Description
0x9130	Index 0x6000:24 "Duty cycle min" for channel 1 is set to zero
0x9140	Index 0x6000:25 "Duty cycle max" for channel 1 is set to zero
0x9131	Index 0x6010:24 "Duty cycle min" for channel 2 is set to zero
0x9141	Index 0x6010:25 "Duty cycle max" for channel 2 is set to zero

8.4.5 Micro-increments

The "micro-increments" function offers the option to interpolate additional increments between the counted encoder increments and thus increase the resolution of the counter value.

Functional principle of micro-increments

The following figure illustrates the functional principle of micro-increments. It is based on the interpolation of increments (orange) within real measured encoder increments (blue). To simplify the display, only four micro-increments are shown between the encoder increments. The current velocity is measured based on the internal period value measurement, and the micro-increments are interpolated accordingly. The interpolation resolution is 8 bits, which corresponds to 256 values. The counter value with micro-increments comes close to the real axis position. This is particularly useful at slow velocities, since an encoder with low resolution becomes a high-resolution axis encoder when micro-increments are enabled.

Example:

- Encoder with 1,024 lines
- 4-fold evaluation
- Micro-increments enabled, 8 bits

1024 lines * 4-fold evaluation * 256 micro-increments = 1,048,567 lines

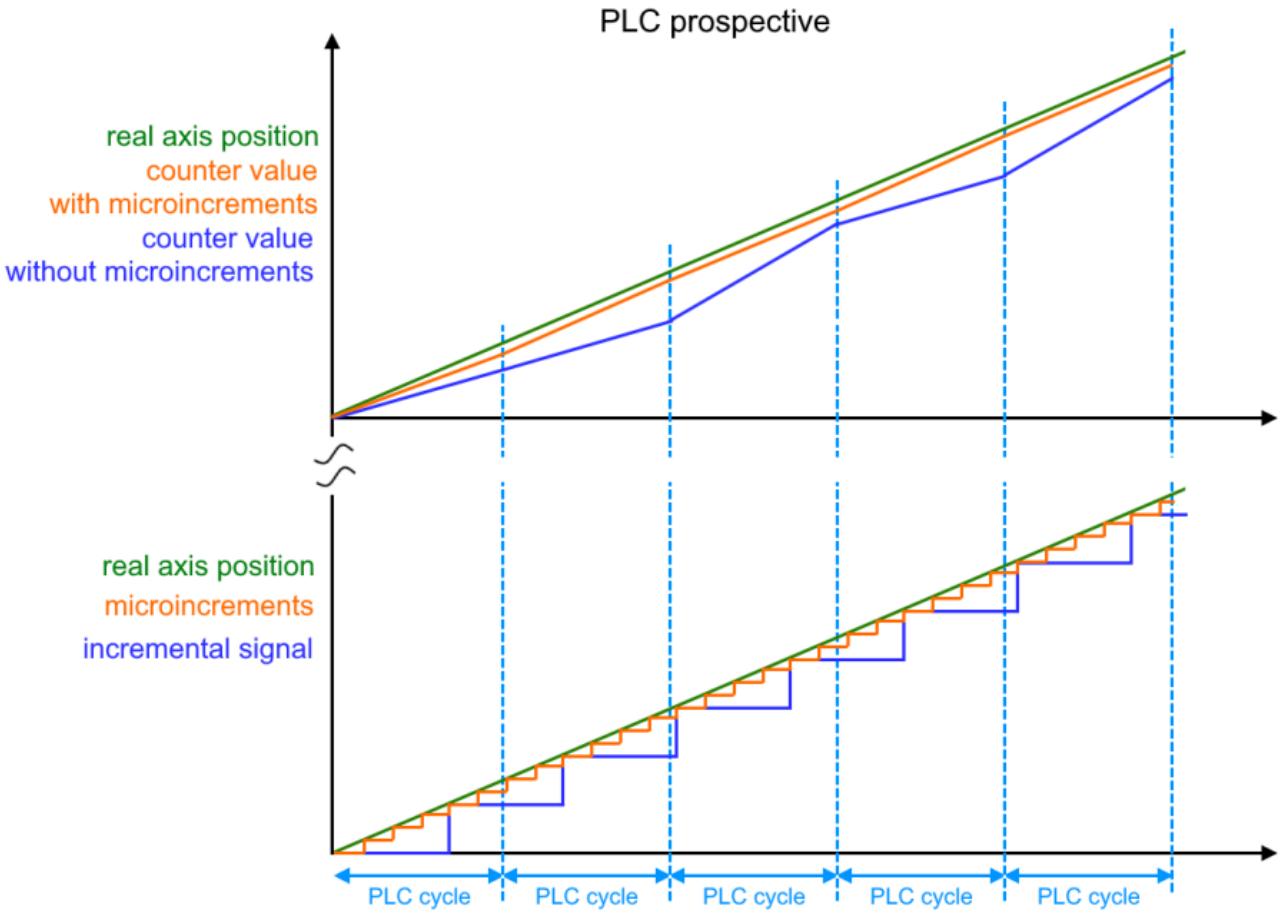


Fig. 168: Functional principle of micro-increments

DC mode for micro-increments

In order to be able to use the micro-increments function in a meaningful way, the terminal should be operated in "DC synchron" or "DC synchron (input based)" mode. Due to the distributed clocks technology the counter value is determined in a cyclically constant manner.

Microincrements sequence

- Microincrements are enabled via index 0x8000:0A "Enable micro increments"
- The microincrements are represented in the last 8 bits of index 0x6000:11 "Counter value".
- The period value of the input signal should be greater than 1.34 µs, so that the microincrements can still be calculated. If this is not the case, the overrun is indicated by index 0x6000:08 "Extrapolation stall" in the process data.
- If the "Extrapolation stall" bit is TRUE, the 8 bits of the microincrements in the "Counter value" are set to zero.

8.4.6 Timestamp function

The timestamp is made possible by the distributed clocks technology in the EtherCAT system. This local clock can be used to synchronize the data acquisition.

Timestamps on the following input signals are available:

- last counter change in the PLC cycle [▶ 172]
- Zero pulse C input: [▶ 172] when storing the counter value on the positive edge of the zero pulse C
- Latch input: [▶ 173] when storing the counter value on the selected edge of the Latch input
- Gate/Latch input [▶ 173], when storing the counter value on the selected edge of the Gate/Latch input
- The digital 24 V outputs [▶ 173]: when switching the outputs on reaching the respective threshold value (Threshold n)

Note on using the timestamp function

In order to be able to use the timestamp function, the terminal must be operated in "DC Synchronous" or "DC Synchronous (input based)" mode.

The timestamp is provided in a range of 64 bits, optionally also with only 32 bits. The timestamps are contained in the following "Predefined PDO Assignment":

Predefined PDO Assignment
1.Ch Standard, extended DC mode, 32 Bit time stamp
1.Ch Standard, extended DC mode, 64 Bit time stamp
1.Ch Standard, extended DC mode, 64 Bit time stamp, Output

They can be selected or deselected as optional PDOs.

PDO assignment	Name	Description
0x1A08 [▶ 137]	ENC Timestamp Channel 1	Contains timestamps in the 64 bit range for channel 1
0x1A09 [▶ 137]	ENC Timestamp Compact Channel 1	Contains timestamps in the 32 bit range for channel 1
0x1A0A [▶ 137]	ENC Timestamp Output Channel 1	Contains timestamps in the 64-bit range for output 1
0x1A15 [▶ 138]	ENC Timestamp Output Channel 2	Contains timestamps in the 64-bit range for output 2

Timestamp for the last counter change in the PLC cycle

The process data index 0x6000:16 [▶ 223] "Timestamp" indicates the timestamp of the last registered counting pulse in the PLC cycle. This means, depending on the counter evaluation mode [▶ 149] (index 0x8000:06 [▶ 219] "Evaluation mode") the timestamp is output based on:

0x8000:06 "Evaluation mode"		0x6000:16 "Timestamp“
0	4-fold (default)	last rising or falling edge on track A or B
1	1-fold	last rising edge on track A
2	2-fold	last rising or falling edge on track A

Timestamp on input zero pulse C

The process data index 0x60p0:1F [▶ 223] "Timestamp C" outputs the timestamp of the last registered positive edge of the zero pulse C when the counter value is saved.

Parameterization of the zero pulse C input		0x60p0:1F: "Timestamp C" issues the timestamp based on:
0x70p0:01	Enable Latch C	the positive edge of the zero pulse C at which the counter value was stored in index 0x60p0:12 [▶ 223] "Latch value"
Depending on the number of channels applies: p = 0 for Ch1, p = 1 for Ch2		

Timestamp for the Latch extern input

The process data index 0x60p0:20 [▶ 223] "Timestamp latch" indicates the timestamp on the selected edge of the Latch extern input. This means that, depending on the parameterization of the Latch extern input, the timestamp is valid for:

Parameterization of the Latch input		0x60p0:20 "Timestamp latch" issues the timestamp based on:
0x70p0:02 [▶ 225]	Enable latch extern on positive edge	the last rising edge at the Latch extern input at which the counter value was stored in index 0x60p0:12 [▶ 223] "Latch value"
0x70p0:04 [▶ 225]	Enable Latch extern on negative edge	the last falling edge at the Latch extern input at which the counter value was stored in index 0x60p0:12 "Latch value"
Depending on the number of channels applies: p = 0 for Ch1, p = 1 for Ch2		

Timestamp for Gate/Latch input

The process data 0x60p0:21 [▶ 223] "Timestamp latch 2" indicates the timestamp on the selected edge of the Gate/Latch input. The Gate/Latch input can also be used as Latch extern 2 input. Depending on the parameterization of the Latch extern 2 input the timestamp is valid for:

Parameterization of the Latch 2 input		0x60p0:21 "Timestamp latch 2" issues the timestamp based on:
0x70p0:0C [▶ 225]	Enable Latch extern 2 on positive edge	the last rising edge at the Gate/Latch input at which the counter value was stored in index 0x60p0:22 [▶ 223] "Latch value 2"
0x70p0:0D [▶ 225]	Enable Latch extern 2 on negative edge	the last falling edge at the Gate/Latch input at which the counter value was stored in index 0x60p0:22 "Latch value 2"
Depending on the number of channels applies: p = 0 for Ch1, p = 1 for Ch2		

Timestamp function for outputs

The process date 0x6004 [▶ 225] "ENC Timestamps Output Ch.1" / 0x6014 "ENC Timestamps Output Ch.2" provides the timestamp when the threshold (Threshold n) was reached and the output switched. The size of the timestamp is 64 bits.

Parameterization of the timestamp for the outputs		0x60p4:1n "Timestamp n" provides the timestamp based on:
0x70p1:02 [▶ 226]	Control_Enable thresholds output = TRUE	the reaching of the threshold values (Threshold n), which are enabled via 0x70p2:11 "Number of output events".
0x70p2:11 [▶ 226]	Number of output events = n	
Dependent on the number of channels applies: p = 0 for Ch1, p = 1 for Ch2) Dependent on the number of activated thresholds "Number of output events": 1 ≤ n ≤ 8		

8.4.7 Adjustable interference pulse filters

Input filters are used for interference suppression at the encoder and digital inputs. Different filter frequencies can be parameterized according to the application.

The individual inputs are subject to the following filter frequencies:

Inputs	Max. recommended filter frequency
Encoder inputs: track A, track B, track C	RS422 mode: 5 MHz TTL mode: 1 MHz Open Collector: 100 kHz
Latch input	1 MHz
Gate/Latch input	1 MHz
Status Input	100 kHz (not adjustable)

Filter function sequence

- The filter is disabled on delivery.
- The filter can be enabled via index 0x8000:08 "Disable filter" = FALSE.
- The filter is parameterized via index 0x8001:19 "Filter settings". The following filter frequencies are available:

Index 0x8001:19 "Filter settings"		Default parameters for 0x8001:1D "Counter mode"	
Setting	**Meaning**	**Setting**	**Meaning**
10_{dec}: 10 kHz	10 kHz filter		
25_{dec}: 25 kHz	25 kHz filter		
50_{dec}: 50 kHz	50 kHz filter		
100_{dec}: 100 kHz	100 kHz filter	4_{dec}	Encoder open collector
		5_{dec}	Counter open collector
250_{dec}: 250 kHz	250 kHz filter		
500_{dec}: 500 kHz	500 kHz filter		
1000_{dec}: 1 MHz	1 MHz filter	2_{dec}	Encoder TTL (single-ended)
		3_{dec}	Counter TTL (single-ended)
2500_{dec}: 2.5 MHz	2.5 MHz filter		
5000_{dec}: 5 MHz	5 MHz filter	0_{dec}	Encoder RS422 (diff. input)
		1_{dec}	Counter RS422 (diff. input)

- If the filter is parameterized < 1 MHz, it is also active for the "Latch" and "Gate" inputs.
- If a frequency overrun is detected when the filter is enabled (0x8000:08 and 0x8001:19), a counter in index 0xA000:16 is incremented.

Index (hex)	Name	Description
0xA000:14	Filter violation counter extern latch	Filter frequency overshoot counter for the Latch extern input
0xA000:15	Filter violation counter input gate	Filter frequency overshoot counter for the Gate input
0xA000:16	Filter violation counter	Filter frequency overshoot counter for the encoder input signals with enabled filter

The internal error counters can be reset as follows:

- Terminal transition from "PREOP" to "OP" status

or

- Enter the following in the command object "0xFB00:01 Request"

0xFB00:01 "Request"	Description
0x9152	Sets the internal error counter for 0xA000:14 "Filter violation counter extern latch" for channel 1 to zero
0x9153	Sets the internal error counter for 0xA000:15 "Filter violation counter input gate" for channel 1 to zero
0x9154	Sets the internal error counter for 0xA000:16 "Filter violation counter" for channel 1 to zero

NOTICE

Fast digital inputs – interference from interfering devices

Please note that the input wiring has very little filtering. It has been optimized for fast signal transmission from the input to the evaluation unit. In other words, rapid level changes/pulses in the µs range and/or high-frequency interference signals from devices (e.g. proportional valves, stepper motor or DC motor output stages) arrive at the evaluation unit almost unfiltered/unattenuated. These interferences can be incorrectly detected as a signal.

- To suppress interference, an additional input filter can be parameterized.
- Furthermore, EMC-compliant cabling and the use of separate power supply units for the terminal and the devices causing interference are recommended.

8.4.8 Plausibility check

The plausibility check of the input signals serves as an extended diagnosis to detect interference signals and to identify and suppress the step changes in the counter value caused by them.

Functional principle of the plausibility check

The square wave signals of an incremental encoder with track A and B are phase-shifted by 90°. This means that only certain transitions in the signal curve are permitted or "plausible", e.g. with a rising A-edge the signal on track B cannot also rise.

The plausibility of the signal sequence is checked in the terminal. If invalid signal transitions occur, these are detected and displayed accordingly if plausibility detection is activated.

Plausibility check sequence

- The plausibility check can be enabled via index 0x8000:21 "Enable encoder plausibility check"
- If a plausibility error is detected, it is displayed as follows:

Error diagnosis	Description
DiagMassage, type "Error", text ID 0x8312	Encoder plausibility error (channel n)
0xA0n0:13 "Encoder plausibility error counter"	The error counter is incremented when a plausibility error is detected
0x60n2:0E TxPDO State = 1	The associated TxPDO data are not valid

The plausibility error counter can be reset as follows:

- Transition from PREOP to OP

or

- enter the following in the command object in index 0xFB00:01 "Request"

0xFB00:01 "Request"	Description
0x9151	Sets the internal error counter 0xA000:13 "Encoder plausibility error counter" for channel 1 to zero

8.5 Output function

Two 24 V_{DC} digital outputs can be parameterized and switched independently of each other (n = 0 for output 1, n = 1 for output 2).

The following options are available for parameterization:

- Set switching characteristics [▶ 178]:
  Push-Pull [▶ 26] or Push-Tristate [▶ 26] mode
- Set state in case of bus error [▶ 179]:
  Definition of the safe state of the output in case of a bus error

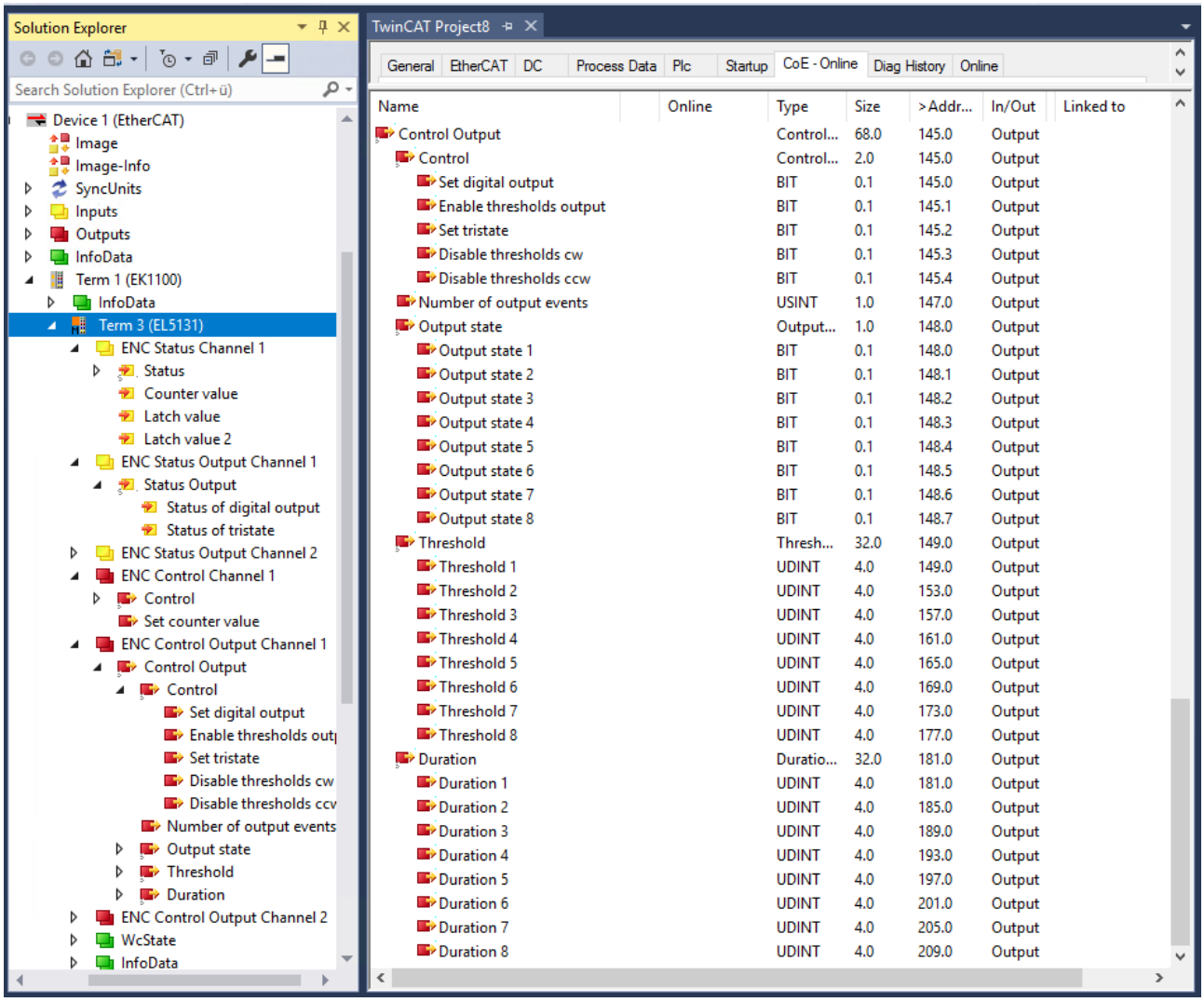
Furthermore, the following options are available for switching the outputs:

1. With disabled automatic switching function
   ("Control_enable thresholds output" 0x70n1:02 = FALSE)
   - via PLC variable [▶ 181]:
     The output can be set from the PLC application.
2. With active automatic switching function [▶ 182]
   ("Control_enable thresholds output" 0x70n1:02 = TRUE)
   - when specified counter values are reached [▶ 184]:
     - Up to eight threshold values ("Threshold 1" - 8) can be specified for switching per output.
     - The status of the output TRUE or FALSE can be selected as desired ("Output state 1" - 8).
     - Furthermore, it is possible to switch the outputs for defined times ("Duration 1" - 8) when predefined values are reached.
     - A timestamp function is available for the outputs when the threshold value (Threshold n) is reached.
       Observe the notes in the chapter Timestamp function for outputs [▶ 192]!
     - The implementation of a digital cam controller [▶ 183] is possible.
   - when specified frequency or velocity values are reached [▶ 189]:
     - Up to two threshold values are available per output.
     - The status of the outputs is fixed:
       "Threshold 1": "Output state 1" = TRUE
       "Threshold 2": "Output state 2" = FALSE
     - The function to switch outputs for a defined time (duration) is **not** available for specified frequency and velocity values.
     - A timestamp function is available for the outputs when the threshold value (Threshold n) is reached.
       Observe the notes in the chapter Timestamp function for outputs [▶ 192]!

Value specification and output data via the process data

The value specification takes place in each case via the process data [▶ 135]. This allows very short reaction times, independent of the PLC cycle.

The output data of the outputs are displayed in the area of the "ENC Control Output Channel n", the corresponding feedbacks or status information under "ENC Status Output Channel n" (see following figure)

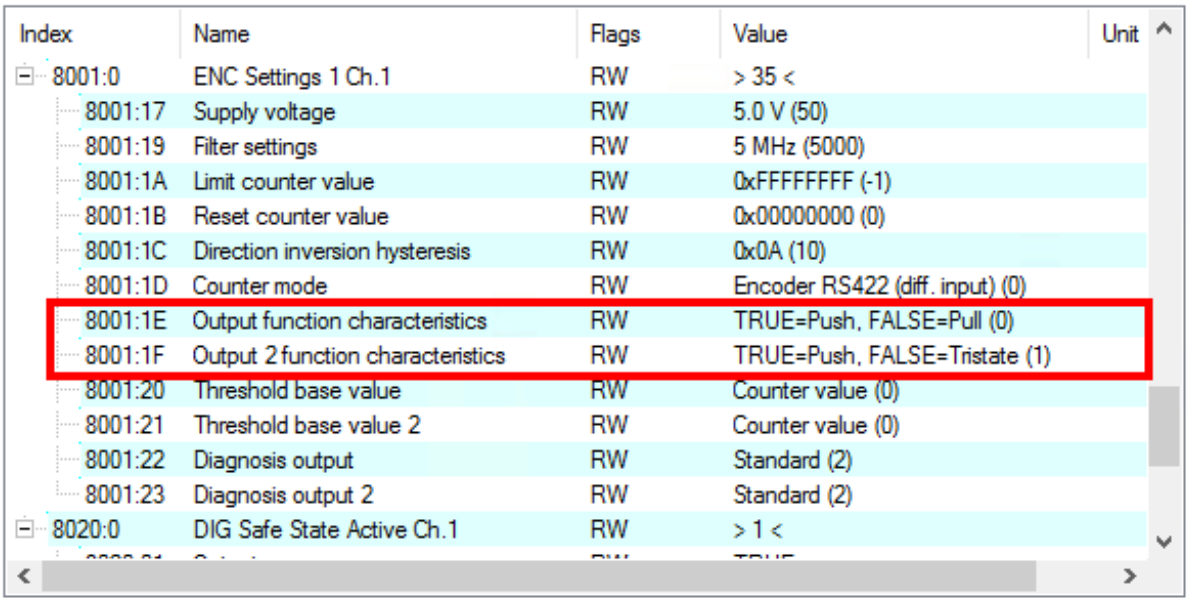


Display of the output data

8.5.1 Set switching characteristics

The switching characteristics of the outputs are defined via the operation mode. The following operation modes are available:

Index 0x8001:1E "Output 1 function characteristic" / Index 0x8001:1F "Output 2 function characteristic"	
Value	**Meaning**
0: TRUE = Push, FALSE = Pull	Output 1 / Output 2 is actively switched to 24 V and 0 V via a push-pull output stage.
1: TRUE = Push, FALSE = Tristate	Output 1 / Output 2 is actively switched to 24 V and high-resistance.



Set switching characteristics via 0x8001:1E / 0x8001:1F

NOTICE



Observe notes on switching behavior

Also observe the descriptions and notes on the switching behavior in chapter "Outputs [▶ 201]"

8.5.2 Set state in case of bus error

DIG Safe State Active (index 0x80n0:01 [▶ 221]) / DIG Safe State Value (index 0x80n1:01 [▶ 222])

The setting in “DIG Safe State Active” (index 0x80n0:01) defines whether the outputs should assume a safe state in the case of a bus error. The safe state of the output in the case of a bus error is defined with “DIG Safe State Value” (index 0x80n1:01).

1. **“DIG Safe State Active“ = TRUE and**
   - **“DIG Safe State Value“ = TRUE**: the output is switched on.
2. **“DIG Safe State Active“ = TRUE and**
   - **“DIG Safe State Value“ = FALSE**: the output is switched off
3. **“DIG Safe State Active“ = FALSE**
   - The state of the output is retained. Entries in “DIG Safe State Value” (index 0x80n1:01) have no effect.

Flow-chart illustration of the sequence in case of a bus error

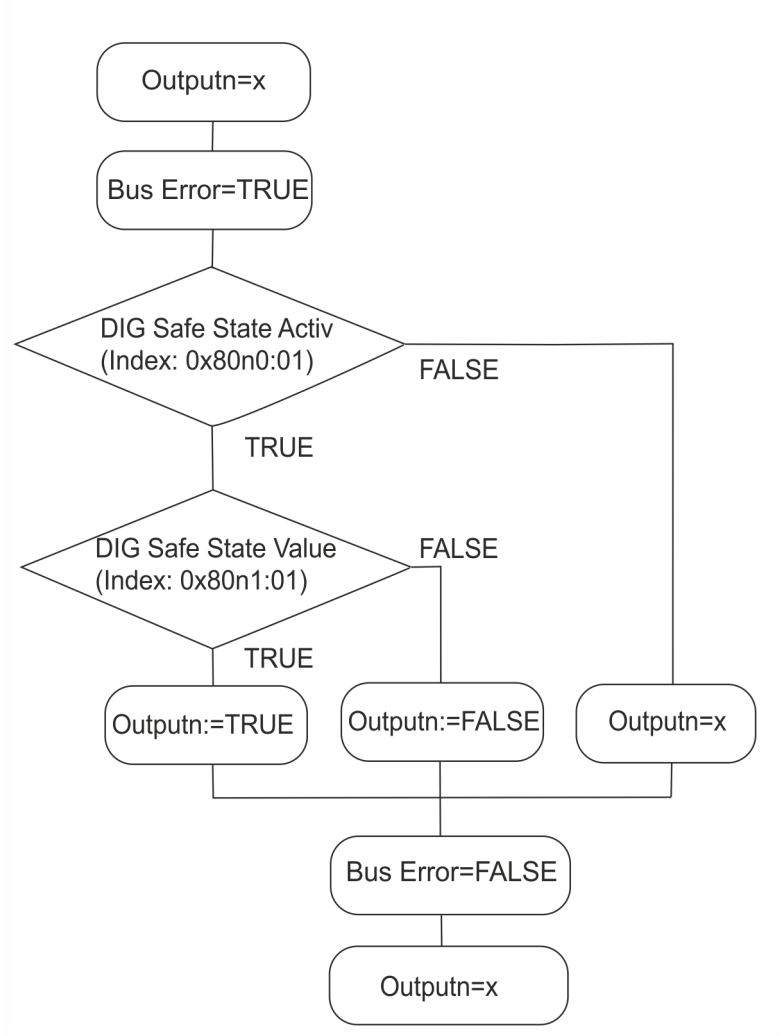


Fig. 169: Change of state of the outputs in the case of a bus error

Tabular example:

DIG Safe State Active Index 0x80n0:01	DIG Safe State Value Index 0x80n1:01	Output before bus error	Output during bus error	Output after bus error
TRUE	TRUE	FALSE	TRUE	FALSE
		TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	FALSE	FALSE
		TRUE	FALSE	TRUE
FALSE	FALSE / TRUE	FALSE	FALSE	FALSE
		TRUE	TRUE	TRUE

Graphical example:

a) Safe State Active = TRUE, Safe State Value = TRUE

b) Safe State Active = TRUE, Safe State Value = FALSE

c) Safe State Active = FALSE, Safe State Value = TRUE

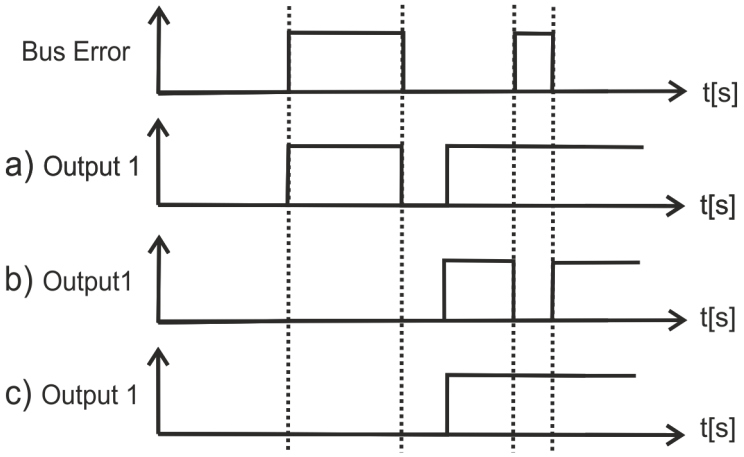


Fig. 170: Graphical illustration of the channel state during a bus error

NOTICE

Watchdog settings

Observe the notes in the chapter "Notes on watchdog settings [▶ 31]"!

8.5.3 Switching outputs via PLC variable

The output can be switched at runtime via the process data "Control_Set digital output" and "Control_Set tristate".

Depending on the operation mode, writing of the output is triggered via the EtherCAT frame or cyclically by the local DC (Distributed Clocks). This makes it possible to set the output, e.g. to check the general function, without activating the automatic switching function.

Switching via PLC variable only with disabled automatic switching function

If the automatic switching function is enabled
(0x70n1:02 "Control_enable thresholds output" = TRUE),
the setting of the output via the PLC variables is ignored.

- Make sure that the automatic switching function is disabled:
  0x70n1:02 "Control_enable thresholds output" = FALSE

The output can take the following status (n = 0 for output 1, n = 1 for output 2):

Control_Set tristate (0x70n1:03)	Control_Set digital output (0x70n1:01)	Meaning
FALSE	TRUE	The output is actively pulled to 24 V
	FALSE	The output is actively pulled to 0 V
TRUE	TRUE	The output is switched to high-resistance
	FALSE	

8.5.4 Switching outputs with preset values

The automatic switching function allows the outputs to be switched based on up to eight predefined threshold values per output. Two switching functions are available for selection:

1. Switching function for counter value
   This function enables the implementation of a digital cam controller [▶ 183]
2. Switching function for frequency value

The following overview shows the differences between the two switching functions. A detailed description of the setting options is given in the following chapters.
The following applies: n = 0 for output 1, n = 1 for output 2

Set reference value of the switching value	**"Counter value (0)" [▶ 184] "Threshold base value" 0x8001:2n = 0**	**"Frequency value (2) [▶ 189]" "Threshold base value" 0x8001:2n = 2**
Set threshold values	"Threshold 1" ... "Threshold 8": Position value, 32-bit	"Threshold 1" and "Threshold 2": Position value, 32-bit
Set output state	Freely selectable TRUE or FALSE for "Output state 1" ... "Output state 8"	Fixed "Output state 1" = TRUE "Output state 2" = FALSE "Output state 3 ... "Output state 8" are not available
Set switching duration	"Duration 1" ... "Duration 8" max. 24,000 µs (0 = waits for next switching value)	Not available
Set number of valid switch values	"Number of output events" = 1 ... 8	"Number of output events" = 1 (hysteresis) "Number of output events" = 2
Set active counting direction	**Forward and backward**: "Disable threshold cw = FALSE and "Disable threshold ccw = FALSE **Forward only**: "Disable threshold cw = FALSE and "Disable threshold ccw = TRUE **Only backwards:** "Disable threshold cw = TRUE and "Disable threshold ccw = FALSE	Not available
Activate automatic switching function	"Enable threshold output" = TRUE	

8.5.4.1 Implementation of a digital cam controller

To trigger sequences highly dynamically and synchronously to a rotating or linear motion, a digital cam controller can be used. This means that a camera or a laser can be controlled on the basis of a position, for example.

The function "Switching with preset counter values" allows this, by appropriate parameterization, with position or time cams on a digital output track.

- Two digital outputs are available, representing two independent cam tracks, as both operate independently.
- Individual cams can be activated or deactivated independently on each track.
- Furthermore, cam positions can be changed per PLC cycle and direction-independent cams can be defined.

Implementation with up to four position cams

- Up to four position cams can be implemented, each with separate TRUE and FALSE threshold values.
- Two threshold values are required for each position cam to define the switch-on and switch-off position.
- If TRUE threshold value smaller FALSE threshold value:
  For all positions between switch-on and switch-off position the output is switched TRUE. For all other position values the output is FALSE.
  The switch-on position corresponds to
  - in positive direction of rotation (CW) the TRUE threshold value,
  - in negative direction of rotation (CCW) the FALSE threshold value.
- The switching behavior is inverted if TRUE threshold value greater than FALSE threshold value.

See examples in chapter "Switching with preset counter values":

- Example of a position cam with different counting directions [▶ 186]
- Example for position cams synchronous to a linear movement [▶ 187]

Implementation with up to eight time cams:

- Up to eight time cams can be implemented.
- Only one “Threshold” is required per time cam.
- When the time cam is reached, the output is switched TRUE (switch-on position) or FALSE (switch-off position) for the duration defined in “Duration” according to the specification for the “Output state”.
- If “Duration n = 0” for a time cam, the current “Output state n” remains active until the next time cam is reached.

See the example in chapter “Switching with preset counter values”:

- Example for three time cams [▶ 188]

Using the function block MC_DigitalCamSwitch_EL5131

If the axis is controlled from the TwinCAT Motion function environment, the NC, the function block “MC_DigitalCamSwitch_EL5131” [▶ 193] can be used.

8.5.4.2 Switching with preset counter values

To switch the outputs based on preset counter values, the following settings must be made for each output. The following applies: n = 0 for output 1, n = 1 for output 2.

Select process data or suitable Predefined PDO Assignment	"1 Ch. Standard, Output" "1 Ch. Standard, extended DC mode, 64 Bit time stamp, Output" "1 Ch. Standard, Period mode, Output" "1 Ch. Standard, Frequency mode, Output"

Select reference value of the switching value	Index 0x8001:2n "Threshold base value" = 0: "Counter value (0)",

Activate automatic switching function	PDO 0x70n1:02 "Control Enable thresholds output" = TRUE As soon as the automatic switching function is enabled, the switching state specified via the PLC variable is ignored. *Notice* **If the switching function is enabled at standstill, the output is set according to the last known state. If there is no state message, the output state = FALSE**

Specify number of currently valid threshold values	Up to eight valid threshold values can be specified which are to be processed. If n threshold values are specified, the threshold values Threshold 1 ... Threshold n are always active. A specific selection, e.g. of the threshold values Threshold 3 and Threshold 4, is therefore not possible. The value should be at least "1" so that the first threshold value is enabled. PDO 0x70n2:11 "Number of output events" ≥ 1,

specify active counting direction [▶ 186]	**0x70n1:04 "Disable thresholds cw"**	**0x70n1:05 "Disable thresholds ccw"**
Both (default):	**FALSE:** Output is switched in positive direction of rotation according to the specifications for "Threshold 1" ... 8.*)	**FALSE:** Output is switched in negative direction of rotation according to the specifications for "Threshold 1" ... 8.*)
Only backwards	**TRUE:** When enabled in positive direction of travel, the output is set to FALSE regardless of the current state. When enabled at standstill, the output is set according to the last known direction of rotation.	**FALSE:** Output is switched in negative direction of rotation according to the specifications for "Threshold 1" ... 8.*)
Only forwards	**FALSE:** Output is switched in positive direction of rotation according to the specifications for "Threshold 1" ... 8.*)	**TRUE:** When enabled in negative direction of travel, the output is set to FALSE regardless of the current state. When enabled at standstill, the output is set according to the last known direction of rotation

*)*Notice* **The output is switched when the counter value is reached by a counter pulse. If the counter value is set to a preset counter value at runtime, the output does not switch.**

The following settings are possible per threshold value:

Specify threshold values	Up to eight 32-bit threshold values can be specified. *Notice* **0 is a valid value, i.e. the output is switched at zero crossing if the threshold value is valid according to the setting in "Number of output events".** 0x70n4:11 to 0x70n4:18 "Threshold 1" to "Threshold 8"

Set output state	Per threshold value the state of the output can be freely defined (TRUE or FALSE). PDO 0x70n3:01 "Output state 1" to 0x70n3:08 "Output state 8"

Specify switching duration [▶ 188]	To switch the output for a defined time (time cam [▶ 183]) a switching duration in µs can be specified for each of the eight threshold values if required. If the switching duration has not yet expired before the next threshold value is reached, the time restarts with the next threshold value. The maximum value is 24,000 µs. If no switching duration is defined (Duration n = 0), the current status of the output remains active until the next threshold value is reached (position cam [▶ 183]). PDO 0x70n5:11 "Duration 1" to 0x70n5:18 "Duration 8

Status of the output

The current output status is reported back via the PDOs 0x60n3:01 "Status of digital output" and 0x60n3:02 "Status of tristate".
Note that the process data is updated with the EtherCAT cycle time, but the switching of the output by means of the switching function is asynchronous.

Example of a position cam with different counting directions

The following graphic shows the switching behavior of the output with a position cam, realized by two thresholds, depending on the active counting directions with the following settings for output 1.

Index	Value	Meaning
0x8001:20 "Threshold base value"	0	Switching function with preset counter values active
0x7001:02 "Control_Enable thresholds output"	TRUE	Automatic switching function active
0x7002:11 "Number of output events"	2	The first two threshold values (Threshold 1 a. 2) are active
0x7003:01 "Output state 1"	TRUE	Status of the output for the first threshold (Threshold 1) is TRUE
0x7003:02 "Output state 2"	FALSE	Status of the output for the second threshold (Threshold 2) is FALSE
0x7001:04 "Control_Disable thresholds cw"	FALSE	Switching function in forward direction (cw) active (see following Fig. **A and C**)
	TRUE	Switching function in forward direction (cw) **not** active (see following Fig. **B**)
0x7001:05 "Control_Disable thresholds ccw"	FALSE	Switching function in backward direction (ccw) active (see following Fig. **A and B**)
	TRUE	Switching function in backward direction (ccw) **not** active (see following Fig. **C**)

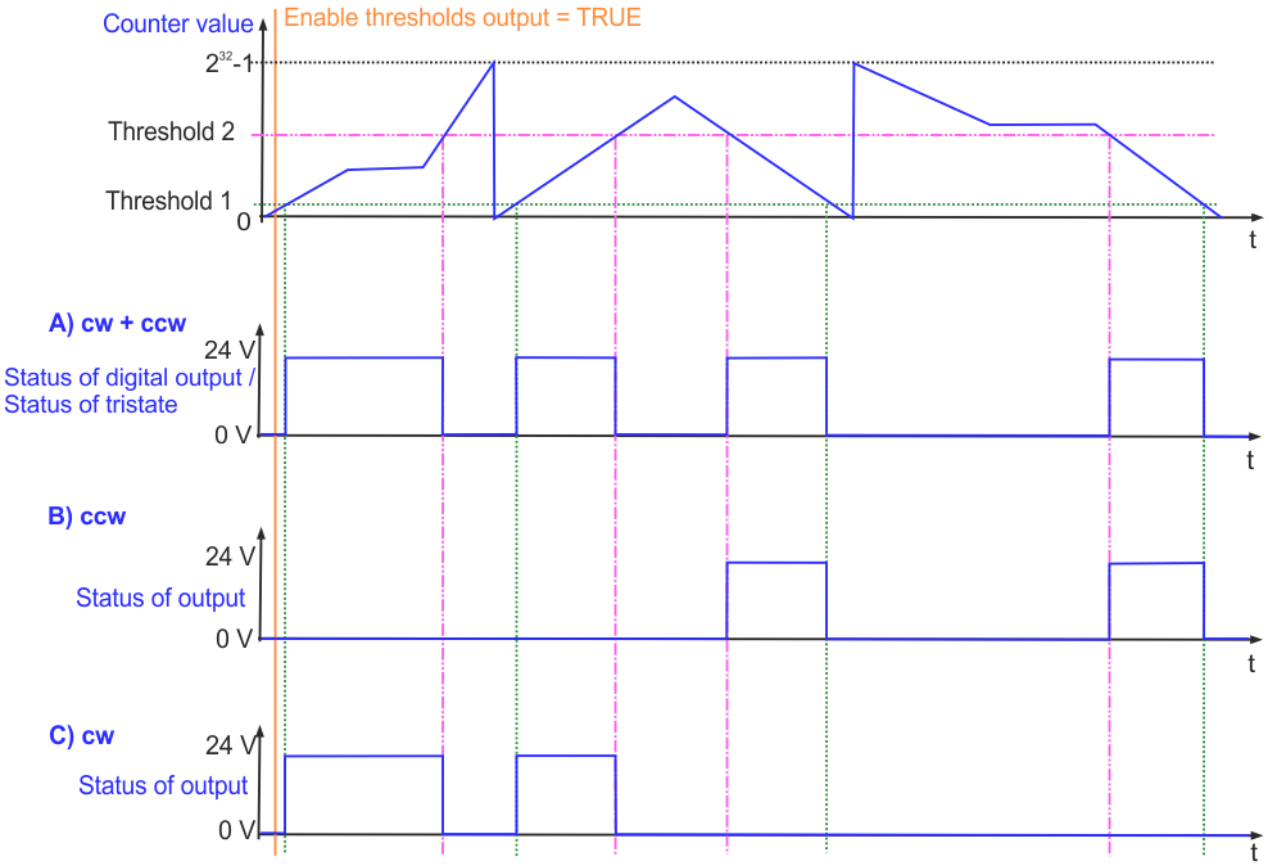


Fig. 171: A) active in both counting directions, B) active in backward direction only, C) active in forward direction only

Example for position cams synchronous to a linear movement

To switch the output synchronously to a linear movement, the switching position (Threshold) must be defined as a switching cam depending on the counter value.

Three examples are shown in the following figure. For all positions within the gray highlighted areas the status of the output is TRUE.

- (A) A position cam is only valid in forward motion
- (B) A position cam is only valid in backward motion
- (C) Three position cams are each valid for both directions of movement

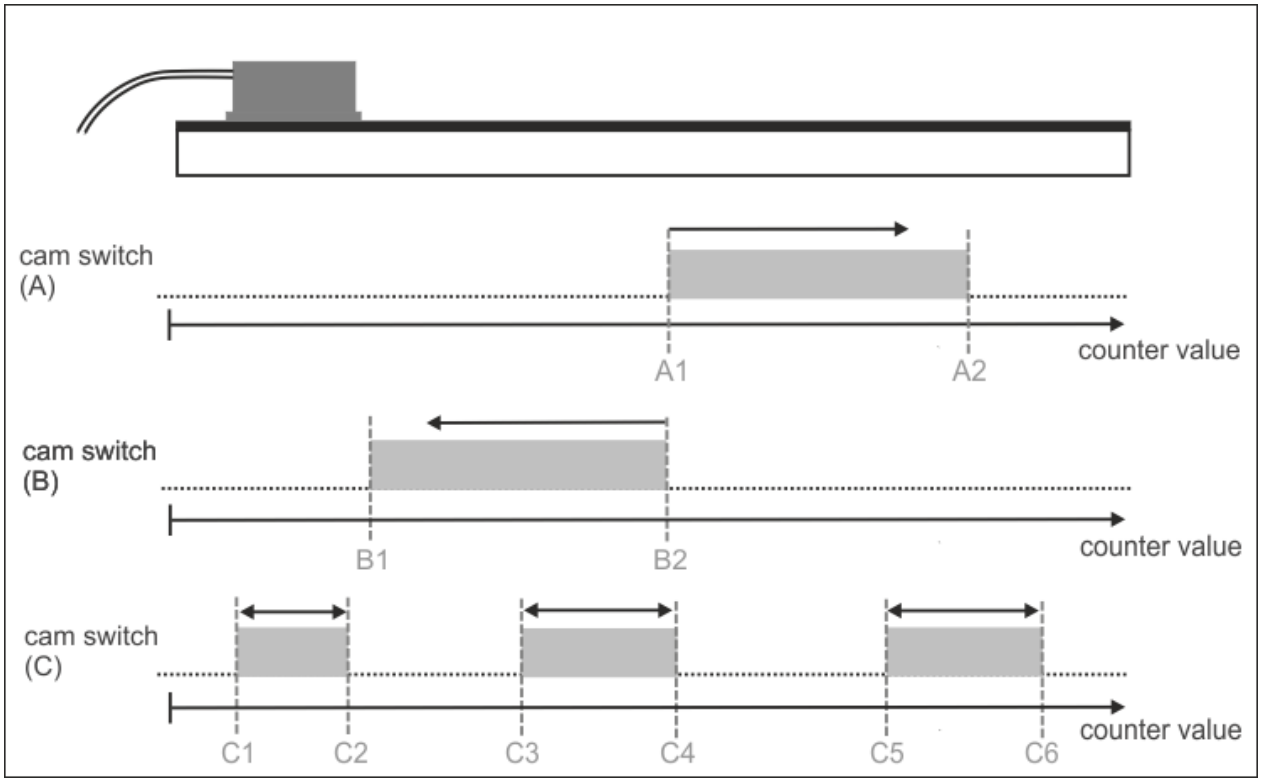


Fig. 172: (A): Position cam forward only, (B): Position cam only backward, (C): 3 position cams for both directions

"Cam switch"	"Number of output events" (0x7001:11)	"Threshold" (0x70n4:01) … (0x70n4:16)	"Output state" (0x70n3:01) … (0x70n3:06)	"Duration" (0x70n5:11) ... (0x70n5:16)	"Direction" (0x70n1:04) (0x70n1:05)
(A)	2	"Threshold 1" = A1 "Threshold 2" = A2	"Output state 1" = TRUE "Output state 2" = FALSE	"Duration 1" = 0 "Duration 2" = 0	"Disable thresholds cw" = FALSE "Disable thresholds ccw" = TRUE
(B)	2	"Threshold 1" = B1 "Threshold 2" = B2	"Output state 1" = TRUE "Output state 2" = FALSE	"Duration 1" = 0 "Duration 2" = 0	"Disable thresholds cw" = TRUE "Disable thresholds ccw" = FALSE
(C)	6	"Threshold 1" = C1 "Threshold 2" = C2 "Threshold 3" = C3 "Threshold 4" = C4 "Threshold 5" = C5 "Threshold 6" = C6	"Output state 1" = TRUE "Output state 2" = FALSE "Output state 3" = TRUE "Output state 4" = FALSE "Output state 5" = TRUE "Output state 6" = FALSE	"Duration 1" = 0 "Duration 2" = 0 "Duration 3" = 0 "Duration 4" = 0 "Duration 5" = 0 "Duration 6" = 0	"Disable thresholds cw" = FALSE "Disable thresholds ccw" = FALSE

Example for three time cams

The following graphic shows the switching behavior of the output with three active time cams. Three threshold values "Threshold 1" ... "Threshold 3" are active for defined switching times "Duration 1" ... "Duration 3" with the following settings for output 1.

Index	Value	Meaning
0x8001:20 "Threshold base value"	0	Switching function with preset counter values active
0x7001:02 "Control_Enable thresholds output"	TRUE	Automatic switching function active
0x7002:11 "Number of output events"	3	The first two threshold values (Threshold 1, 2 and 3) are active
0x7003:01 "Output state 1"	TRUE	Status of the output for the first threshold (Threshold 1) is TRUE
0x7003:02 "Output state 2"	TRUE	Status of the output for the second threshold (Threshold 2) is TRUE
0x7003:03 "Output state 3"	TRUE	Status of the output for the third threshold (Threshold 3) is TRUE
0x7001:04 "Control_Disable thresholds cw"	FALSE	Switching function in forward direction (**cw**) active
0x7001:05 "Control_Disable thresholds ccw"	FALSE	Switching function in reverse direction (**ccw**) active

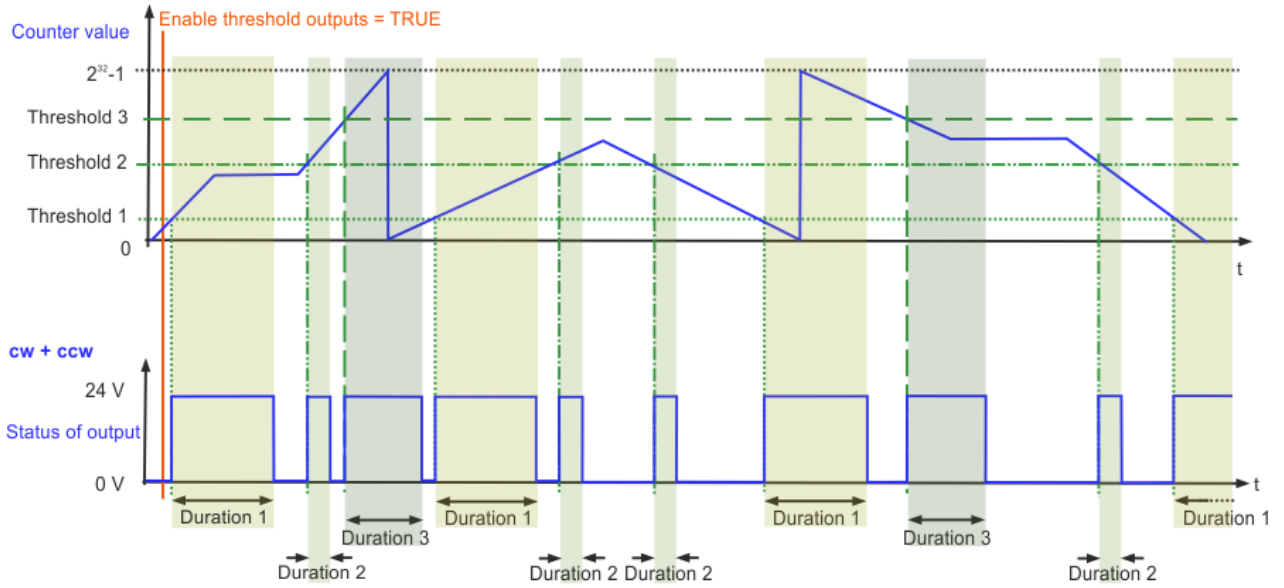


Fig. 173: Switch output for defined times (duration)

8.5.4.3 Switching with preset frequency values

The following settings are required to switch outputs on the basis of specified frequency values. The following applies: n = 0 for output 1, n = 1 for output 2.

Select process data or suitable Predefined PDO Assignment	"1 Ch. Standard, Output" "1 Ch. Standard, extended DC mode, 64 Bit time stamp, Output" "1 Ch. Standard, Period mode, Output" "1 Ch. Standard, Frequency mode, Output"

Select reference value of the switching value	Index 0x8001:2n "Threshold base value" = 2: "Frequency value (2)",

Parameterization frequency or velocity measurement	Parameterization of the frequency or velocity measurement see chapter Frequency measurement [▶ 164] and Velocity, speed calculation [▶ 167]

Activate automatic switching function	PDO 0x70n1:02 "Control Enable thresholds output" = TRUE As soon as the automatic switching function is enabled, the switching state specified via the PLC variable is ignored. **Notice If the switching function is enabled at standstill, the output is set according to the last known state. If there is no state message, the output state = FALSE**

Specify number of currently valid threshold values	Up to two valid threshold values can be specified that are to be processed (Threshold 1 and Threshold 2) PDO 0x70n2:11 "Number of output events": • "Number of output events" = 1: Switch output with a threshold value with hysteresis [▶ 190] function • "Number of output events" = 2 Switch output over two thresholds [▶ 191]

Specify threshold values	Max. two 32-bit threshold values can be specified. **Notice 0 is a valid value, i.e. the output is switched at zero crossing if the threshold value is valid according to the specification in "Number of output events".** • For "Number of output events" = 1 (hysteresis) select for ◦ positive switching logic: "Threshold 1" > "Threshold 2". ◦ negative switching logic: "Threshold 2" > "Threshold 1". • For "Number of output events" = 2 select for ◦ Positive switching logic: "Threshold 1" < "Threshold 2" ◦ negative switching logic: "Threshold 1" > "Threshold 2

Status of the output	The status of the output is fixed for each threshold. • "Threshold 1": PDO 0x70n3:01 "Output state 1" = TRUE • "Threshold 2": PDO 0x70n3:02 "Output state 2" = FALSE

Duration	**Notice When switching with specified frequency or velocity values, no duration can be specified.**

Status of the output

The current output status is reported back via the PDOs 0x60n3:01 "Status of digital output" and 0x60n3:02 "Status of tristate".
Note that the process data is updated with the EtherCAT cycle time, but the switching of the output by means of the switching function is asynchronous.

Switch output with a threshold with hysteresis function

Proceed as follows if only a threshold is to be specified for switching the output so that it is switched on when a defined frequency or velocity is reached and switched off again via a hysteresis function when the frequency or velocity falls below a defined value.

Example of the hysteresis function for positive switching logic (n = 0 for output 1, n = 1 for output 2)

1. "Number of output events" = 1
2. Specification of the switching threshold at which the output is to be switched on: "Threshold 1",
   the status of the output is fixed at "Output state 1" = TRUE,
   an input via the corresponding process data is ignored.
3. Specification of the switching threshold at which the output is to be switched off again "Threshold 2".
   For positive switching logic, "Threshold 1" > "Threshold 2" must be selected,
   the transition range is to be interpreted as hysteresis.

⇨ The current status of the output is reported via the PDOs 0x60n3:01 "Status of digital output" and 0x60n3:02 "Status of tristate".
Please note that the process data is updated with the EtherCAT cycle time, but the switching of the output is asynchronous.

Examples of switching behavior for frequency value specification with hysteresis function:

1. For positive switching logic, select "Threshold 1" > "Threshold 2".

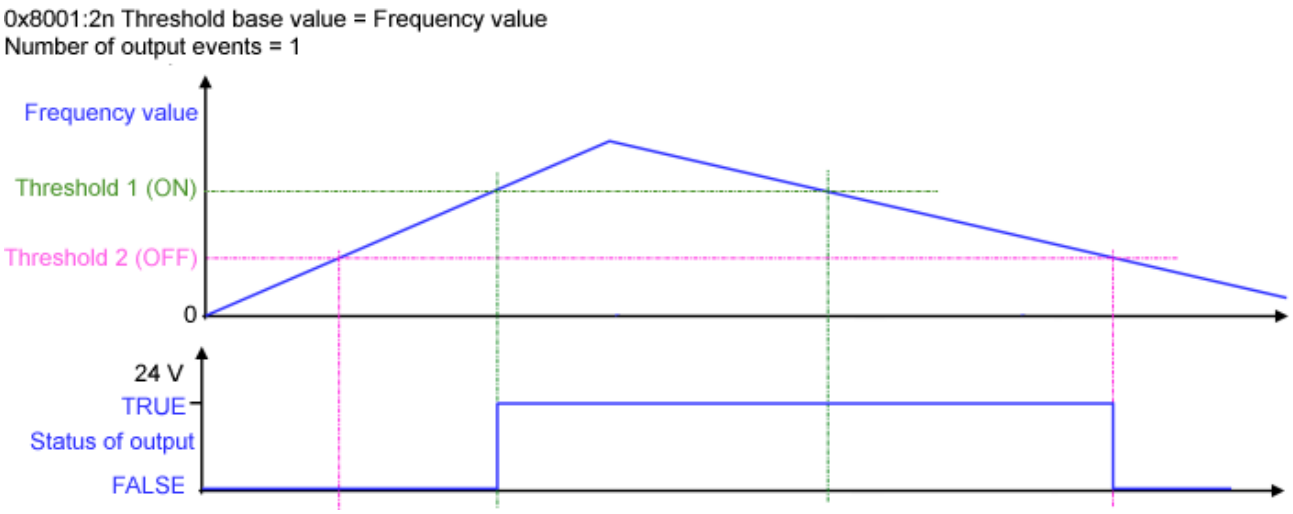


Fig. 174: Switching behavior for frequency value specification with hysteresis, positive switching logic

2. For negative switching logic, select "Threshold 2" > "Threshold 1".

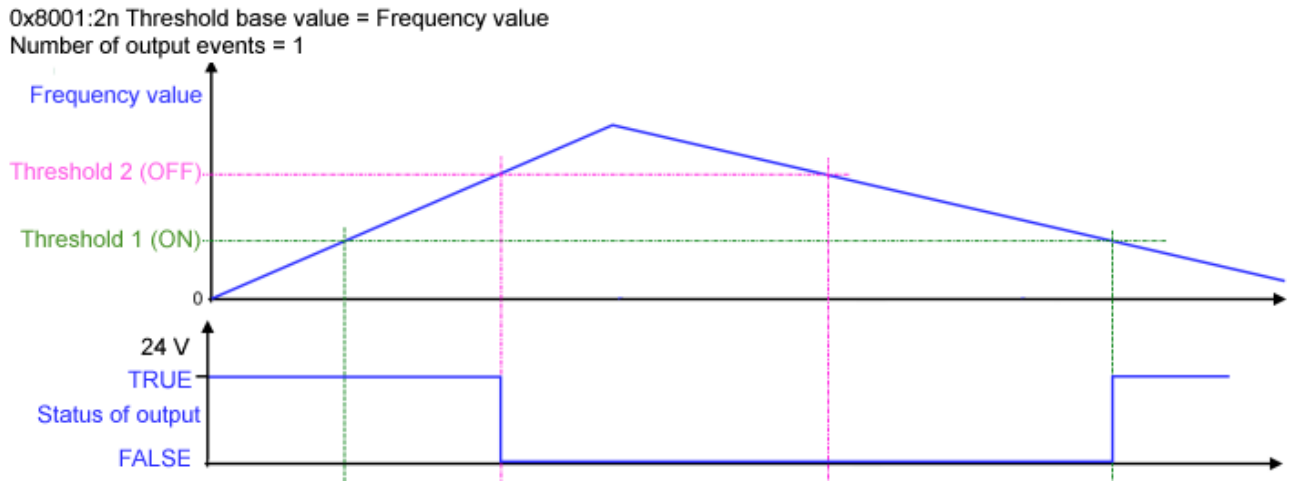


Fig. 175: Switching behavior for frequency value specification with hysteresis, negative switching logic

Output switching via two threshold values (n = 0 for output 1, n = 1 for output 2)

If the output is to be switched based on two threshold values, both positive and negative switching logic can be implemented. The threshold values are to be selected as follows:

- Positive switching logic: Threshold 1 < Threshold 2
- Negative switching logic: Threshold 1 > Threshold 2

Example for switching the output with two threshold values with positive switching logic

1. "Number of output events" = 2
2. Specification of the switching threshold at which the output is to be switched on:
   "Threshold 1", the status of the output is fixed to "output state" = TRUE,
   an input via the corresponding process data is ignored.
3. Specification of the switching threshold at which the output is to be switched off again
   "Threshold 2"
4. Select "Threshold 1 < Threshold 2" for positive switching logic

⇨ The current status of the output is reported via the PDOs 0x60n3:01 "Status of digital output" and 0x60n3:02 "Status of tristate".
Please note that the process data is updated with the EtherCAT cycle time, but the switching of the output is asynchronous.

Examples Switching behavior with two threshold values:

1. Select "Threshold 1 < Threshold 2" for positive switching logic.

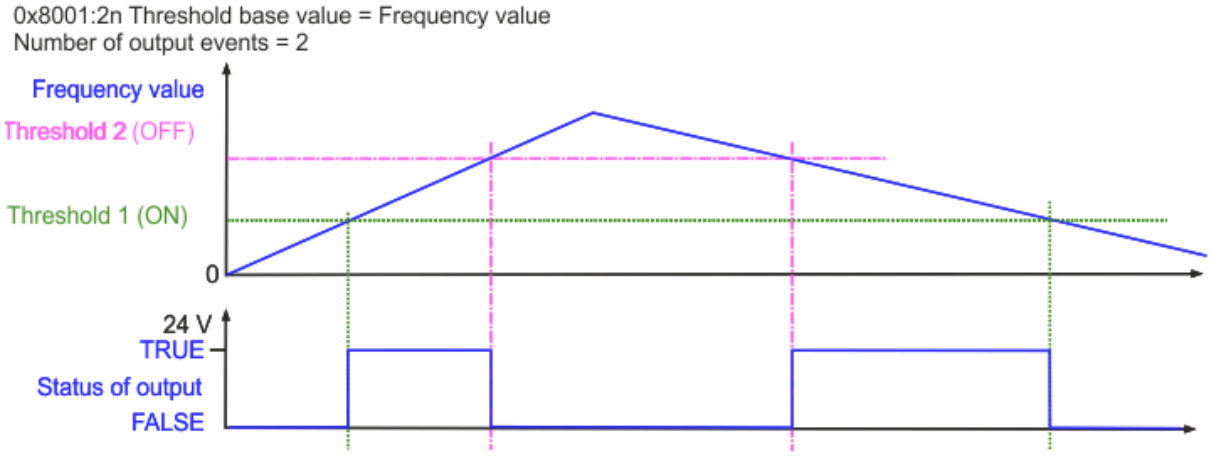


Fig. 176: Switch output with two threshold values, positive switching logic

2. Select "Threshold 1 > Threshold 2" for negative switching logic.

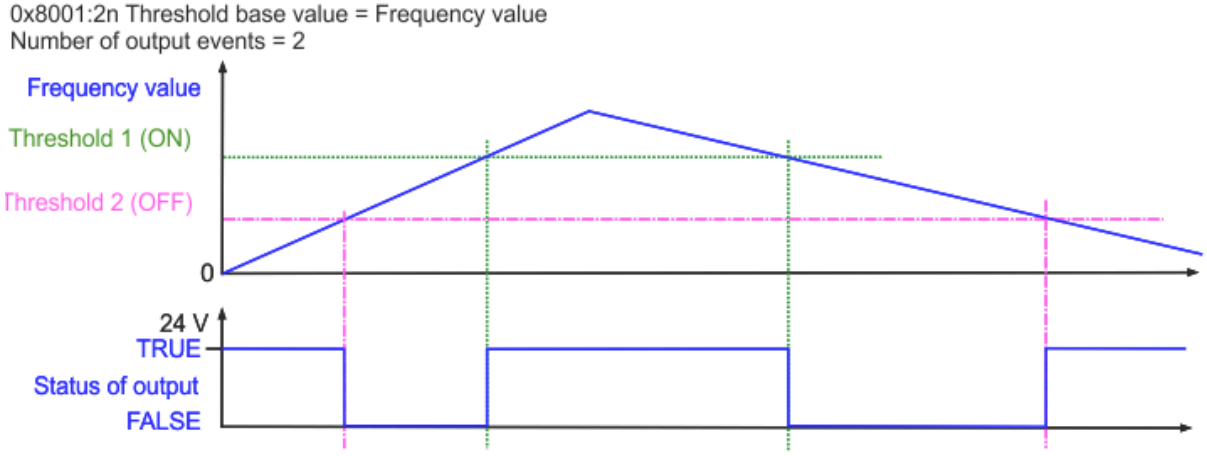


Fig. 177: Switch output with two threshold values, negative switching logic

8.5.4.4 Timestamp function for outputs

Note on using the timestamp function

In order to be able to use the timestamp function, the terminal must be operated in "DC Synchronous" or "DC Synchronous (input based)" mode.

The timestamp is provided in a range of 64 bits, optionally also with only 32 bits. The timestamps for the outputs are contained in the following "Predefined PDO Assignment":

Predefined PDO Assignment
1.Ch Standard, extended DC mode, 64 Bit time stamp, Output

They can be selected or deselected as optional PDOs.

PDO assignment	Name	Description
0x1A0A [▶ 137]	ENC Timestamp Output Channel 1	Contains timestamps in the 64-bit range for output 1
0x1A15 [▶ 138]	ENC Timestamp Output Channel 2	Contains timestamps in the 64-bit range for output 2

The process date 0x6004 [▶ 225] "ENC Timestamps Output Ch.1" / 0x6014 "ENC Timestamps Output Ch.2" provides the timestamp when the threshold (Threshold n) was reached and the output switched. The size of the timestamp is 64 bits.

Parameterization of the timestamp for the outputs		0x60p4:1n "Timestamp n" provides the timestamp based on:
0x70p1:02 [▶ 226]	Control_Enable thresholds output = TRUE	the reaching of the threshold values (Threshold n), which are enabled via 0x70p2:11 "Number of output events".
0x70p2:11 [▶ 226]	Number of output events = n	
Dependent on the number of channels applies: p = 0 for Ch1, p = 1 for Ch2) Dependent on the number of activated thresholds "Number of output events": 1 ≤ n ≤ 8		

Deleting timestamps by reducing the active thresholds

If the number of enabled thresholds is reduced via "Number of output events" or set to "0", the timestamps of the non-active thresholds are deleted.

8.5.5 Function block “MC_DigitalCamSwitch_EL5131”

If the axis is controlled from the TwinCAT Motion function environment, the NC, the axis parameters are stored in the NC, including the adjustment of the scaling factor for the encoder. In addition to the actual encoder position, the specification of the individual cams is decisive for the implementation of a digital cam controller. In order to have a consistent calculation basis, it is helpful to use the position data scaled by the NC directly to determine the cam positions.

The library TF5065 “Motion Control XFC/ XFC NC I” provides a function block “MC_DigitalCamSwitch_EL5131” for the commissioning of a cam controller based on the EL5131.

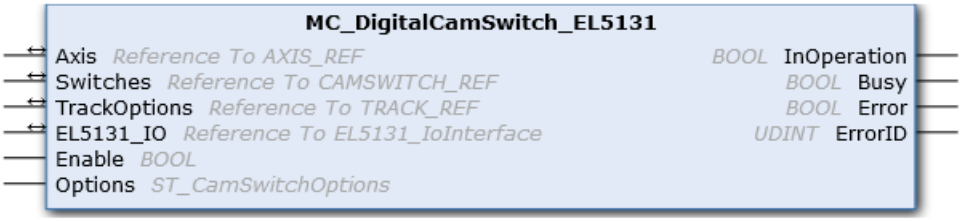


Fig. 178: Function block “MC_DigitalCamSwitch_EL5131”

The use of the function block “MC_DigitalCamSwitch_EL5131” offers the following advantages:

- Processing of the position data scaled by the NC,
  the additional conversion of the “Counter Value” in the PLC is no longer necessary.
- Organization of all parameters of the cam tracks and the individual cams in prepared data structures (data type: CAMSWITCH_REF, data type: MC_CamSwitch) allow structured parameterization as position or time cams
- Individual configuration of the two digital 24 V outputs as independent cam tracks
- Simplified declaration of variables through defined data structure for EL5131; these no longer need to be declared individually
- Fast error analysis via defined status bits

Requirements

Development environment	PLC libraries to include
TwinCAT 3.1 Build 4024.55	Tc2_MC2 3.3.59 Tc2_MC2_XFC 3.3.25

Notes on using the function block “MC_DigitalCamSwitch_EL5131”

The EL5131 provides two digital outputs that represent two independent cam tracks.

For each cam track

- a separate function block must be used.
- the number of cams used can be specified. Up to four position cams or up to eight time cams are available.
- the switch-on time of the individual cam can be defined for time cams.
- the direction in which the cams are active (positive, negative or both) can be defined.

No mixed operation of different directions within one cam track

Mixed operation in different directions is not permitted within the cam track.

- Make sure that the same direction is selected for each cam within the cam track.

Further information can be found in the TwinCAT 3 documentation TF5065 “Motion Control XFC/XFC NC I” in the chapter “Cam controller” – “MC_DigitalCamSwitch_EL5131”.

8.6 Inputs

The number of available inputs depends on the operation mode.
The table gives a short overview of the functions of the inputs of the EL5131.
Read the detailed description of the functions in the respective chapters.

Input functions

Counter value	PLC variable	Zero pulse C [▶ 195]	Latch input [▶ 197]	Gate/Latch input [▶ 198]
Set [▶ 155]	Enable setting: 0x7000:03 "Set counter" = TRUE to the value: 0x7000:11 "Set counter value"	Enable setting: 0x7000:08 "Set counter on latch C" = TRUE to the value: 0x7000:11 "Set counter value"	Enable setting: 0x7000:0A/0B "Set counter on latch extern on positive/ negative edge" = TRUE to the value: 0x7000:11 "Set counter value"	-
Reset [▶ 154]	-	Enable resetting: 0x8000:01 "Enable C reset" to the value: 0x8001:1B "Reset counter value"	Enable resetting: 0x8000:02 "Enable external reset" to the value: 0x8001:1B "Reset counter value"	-
Save [▶ 159]	-	Enable saving: 0x7000:01 "Enable latch C" = TRUE Save current counter value in: 0x6000:12 "Latch value"	Enable saving: 0x7000:02/04 "Enable latch extern on positive/ negative edge" = TRUE Save current counter value in: 0x6000:12 "Latch value"	Enable saving via latch 2 input: 0x7000:0C/0D "Enable latch extern 2 on positive/ negative edge" = TRUE Save current counter value in: 0x6000:22 "Latch value 2"
Lock [▶ 163]	Lock: 0x7000:09 "Set software gate" = TRUE	-	-	Enable locking with HIGH level: 0x8000:04 "Gate polarity" = 1 with LOW level: 0x8000:04 "Gate polarity" = 2

8.6.1 Zero pulse C input

With incremental encoders, a full revolution is marked by a special marker of the zero pulse C. This pulse can be used in the terminal as follows:

- Counter value reset [▶ 195]: the counter value is reset to the value set in index 0x8001:1B [▶ 221] "Reset counter value" (default setting: "0") (Enable C reset)
- Set counter value: [▶ 195] the counter value is set to a predefined value (Set counter on Latch C)
- Save counter value: [▶ 196] the counter value is saved in the latch register (Enable latch C)

The current level at the zero pulse C input is displayed via process data 0x60n0:0B [▶ 223] "Status of Input C".

Status of Input C

Zero pulse C has only a short pulse duration, depending on the rotational speed. Due to the cyclic request, it is possible that the HIGH level of the zero pulse is not transferred congruently into the process data 0x60n0:0B [▶ 223] "Status of Input C".

Counter value reset via the zero pulse C input (Enable C reset)

The counter value can be set to the value specified in index 0x8001:1B [▶ 221] "Reset counter value" for every full revolution of the encoder via the zero pulse C.

- Presetting of the reset value via index 0x8001:1B [▶ 221] "Reset counter value", (Default: 0)
- To activate this function set the bit in index 0x8000:01 [▶ 219] "Enable C reset".
- There is no status message via the process data.

Set counter value via input zero pulse C (Set counter on Latch C)

The counter value can be set to a preset value during runtime via the process data by means of the zero pulse C.

- Counter value setting via index 0x70n0:11 [▶ 225] "Set counter value"
- Activation of the counter value setting via zero pulse C:
  - Index 0x70n0:08 [▶ 225] "Set counter on latch C" = TRUE
    With the next zero pulse the counter value is set to the counter value specified in index 0x70n0:11 [▶ 225] (Set counter value).
- For confirmation the "Set counter done" bit in index 0x60n0:03 is set to TRUE.
- The counter value setting via the zero pulse C cannot be reactivated until the "Set counter on latch C" bit in index 0x70n0:08 is set to FALSE.

Multiple activation of "Set counter on ..."

If several commands are activated to accept the preset counter value, only the command that is set first is accepted by the terminal. All other commands are ignored but remain activated, depending on the system.

- The counter value is set to the preset counter value at the first activated event and confirmed by the bit "Set counter done" (index 0x60n0:03 [▶ 223]).
- The counter value specification cannot be reactivated until all activated commands for the transfer of the counter value have been deactivated. This is confirmed by setting the "Set counter done" bit (index 0x60n0:03) to FALSE.

Save counter value via the zero pulse C input (Enable latch C)

The counter value can be stored at runtime via the process data and the zero pulse in "Latch value" (index 0x60n0:12 [▶ 223]).

- The function is activated by setting the bit in "Enable latch C" (index 0x70n0:01) [▶ 225] to TRUE.
- The current counter value is saved in "Latch value" (index 0x60n0:12) with the next zero pulse at input C. The subsequent pulses have no influence on the latch value.
- The "Latch C valid" bit (index 0x60n0:01 [▶ 223]) is set to TRUE.
- After reactivating "Enable Latch C" (index 0x70n0:01) a new counter value cannot be written to the Latch input until the values of the "Enable Latch C" bit (index 0x70n0:01) and the "Latch C valid" bit (index 0x60n0:01) are FALSE.



Multiple activation of the latch function

If several commands are activated simultaneously to save the counter value in the "Latch value" process data (index 0x60n0:12 [▶ 223]), only the command that is set first is accepted by the terminal.

- The counter value is stored in the "Latch value" (index 0x60n0:12) at the next occurring event and confirmed with the corresponding bit.
- All other activated events are ignored.
- The counter value storage cannot be reactivated until all activated commands for latching the value have been deactivated. This also applies if they were activated after confirmation by the occurring event.

8.6.2 Latch input (Latch extern)

The terminal provides a latch input (Latch extern) for 24 V_{DC} signals with a minimum pulse duration of t_{ON} > 1 µs. This can be used as follows:

- Counter value reset [▶ 197]:
  the counter value is reset to the value set in index 0x8001:1B "Reset counter value" (default setting: "0") (Enable extern reset)
- Set counter value [▶ 197]:
  set the counter value to a predefined counter value via a positive or negative edge (Set counter on latch extern on positive/negative edge)
- Save counter value [▶ 198]:
  save the counter value in separate process data via a positive or negative edge (Enable latch extern on positive/negative edge)

Counter value reset via the Latch extern input (Enable extern reset)

The counter value can be set to "0" via the Latch extern input.

- Presetting of the reset value via index 0x8001:1B [▶ 221] "Reset counter value", (Default: 0)
- To activate this function set the bit in index 0x8000:02 [▶ 219] "Enable extern reset".
- Index 0x8000:10 [▶ 219] "Extern reset polarity" can be used to specify at which edge the Latch extern input is active.
  - 0: "Fall" - the counter is set to "0" with a falling edge
  - 1: "Rise" - the counter is set to "0" with a rising edge

There is no status message via the process data.

Set counter via the Latch extern input (Set counter on latch extern on positive/negative edge)

The counter value can be set to a preset value during runtime via the process data by the positive or negative edge at the Latch extern input.

- Counter value setting via index 0x70n0:11 "Set counter value"
- Activation of the counter value setting via
  - the positive edge at the Latch extern input:
    index 0x70n0:0A "Set counter on latch extern on positive edge"
  - the negative edge at the Latch extern input:
    index 0x70n0:0B "Set counter on latch extern on negative edge"
- If the bit is set (TRUE) in index 0x70n0:0A or 0x070n0:0B, the counter value is set to the specified value at the next rising or falling edge at the Latch extern input.
- For confirmation the "Set counter done" bit (index 0x60n0:03) is set to TRUE.
- The counter value specification cannot be reactivated until index 0x70n0:0A/0B "Set counter on latch extern on positive/negative edge" is set to FALSE.

Save counter value via a positive/negative edge at the Latch input (Enable latch extern on positive/ negative edge)

- The counter value at the Latch extern input can be saved via:
  - Index 0x7000:02 [▶ 225] "Enable latch extern on positive edge" = TRUE
    At the first external latch pulse with positive edge the current counter value is stored in index 0x6000:12 [▶ 223] "Latch value".
  - Index 0x7000:04 [▶ 225] "Enable latch extern on negative edge" = TRUE
    At the first external latch pulse with negative edge the current counter value is stored in index 0x6000:12 [▶ 223] "Latch value".
  - Simultaneous activation of 0x7000:02 and 0x7000:04
    On the first external Latch pulse, regardless of the polarity of the edge, the current counter value is stored in index 0x6000:12 "Latch value".
- Specification whether it is necessary to reactivate the command to save the counter value via:
  - "Enable continuous latch extern" index 0x8000:22 [▶ 219] = FALSE
    The following pulses at the latch extern input have no influence on the latch value in index 0x6000:12 "Latch value" if the bit in index 0x7000:02 or 0x7000:04 is set.
    Only when index 0x6000:02 [▶ 223] "Latch extern valid" is FALSE, a new counter value can be written to the latch input in index 0x60n0:12 "Latch value"
  - "Enable continuous latch extern" index 0x8000:22 [▶ 219] = TRUE
    The counter value is written to index 0x60n00:12 "Latch value" with each parameterized edge at the latch extern input.
    A new activation of index 0x7000:02 or 0x7000:04 is omitted.
- Saving of the counter value in index 0x6000:12 "Latch value" is confirmed via the "Latch extern valid" bit (index 0x6000:02).
- The status of the Latch extern input can be monitored via index 0x6002:14 [▶ 224] "Status of extern latch".

8.6.3 Gate/Latch input

The terminal provides an external Gate/Latch input for 24 V_{DC} signals with a minimum pulse duration of t_{ON} > 1 µs. This can be used as follows:

- Lock counter value [▶ 198]:
  lock the counter value via a positive or negative edge (Enable pos/neg. gate)
- Save counter value [▶ 199]:
  save the counter value in separate process data via a positive or negative edge (Enable latch extern 2 on positive/negative edge)

Lock counter value via a positive/negative edge at the gate input (Enable pos./neg. gate)

- The level at the Gate input at which the counter value is locked during runtime can be specified via index 0x8000:04 [▶ 219] "Gate polarity".
  - 0: Disable gate
    The Gate/Latch input is disabled. It can still be used as Latch extern 2 input.
  - 1: Enable pos. gate
    The counter value is locked with HIGH level at the Gate/Latch input.
  - 2: Enable neg. gate
    The counter value is locked with LOW level at the Gate/Latch input.
- The current level at the Gate input is displayed via process data 0x6000:0C [▶ 223] "Status of input gate".

Save counter value via a positive/negative edge at the Gate/Latch input (Enable latch extern 2 on positive/negative edge)

The Gate/Latch input may be used as a second latch input (Latch extern 2). To do this, the gate should be disabled.

Deactivation of the gate

Set the "Gate polarity" bit (index 0x8000:04 [▶ 219]) to "0 – Disable gate" in order to be able to use the Latch extern 2 input without disabling the counter value after the latch event.

- The counter value at the Latch extern 2 input can be saved via:
  - Index 0x7000:0C [▶ 225] "Enable latch extern 2 on positive edge" = TRUE
    At the first external latch pulse with positive edge at the Gate/Latch input the current counter value is stored in index 0x6000:22 [▶ 223] "Latch value 2".
  - Index 0x7000:0D [▶ 225] "Enable latch extern 2 on negative edge" = TRUE
    At the first external pulse with negative edge at the Gate/Latch input the current counter value is stored in index 0x6000:22 "Latch value 2".
  - Simultaneous activation of 0x7000:0C and 0x7000:0D
    On the first pulse at the Gate/Latch input, regardless of the polarity of the edge, the current counter value is stored in index 0x6000:22 "Latch value 2".
- Specification whether it is necessary to reactivate the command to save the counter value via:
  - "Enable continuous latch extern 2" Index 0x8000:23 [▶ 219] = FALSE
    The following pulses at the Gate/Latch input have no influence on the Latch value in index 0x6000:22 "Latch value 2" if the bit in index 0x7000:0C or 0x7000:0D is set.
    Only when index 0x6002:12 [▶ 224] "Latch extern 2 valid" is FALSE, a new counter value can be written to the Gate/Latch input in index 0x6000:22 "Latch value 2".
  - "Enable continuous latch extern 2" index 0x8000:23 [▶ 219] = TRUE
    The counter value is written to index 0x6002:12 "Latch value 2" for each parameterized edge at the Gate/Latch input.
    There is no need to reactivate index 0x7000:0C or 0x7000:0D.
- The status of the Gate/Latch input can be recorded via index 0x6000:0C [▶ 223] "Status of input gate".
- Saving of the counter value in index 0x6000:22 "Latch value 2" is confirmed via bit 0x6002:12 "Latch extern 2 valid".

Simultaneous use of Gate und Latch extern 2

When using index 0x8000:04 [▶ 219] "Gate polarity"(1 = "Enable pos. gate") and simultaneously activating index 0x7000:0C [▶ 225] "Enable latch extern 2 on positive edge", the current counter value is initially stored in index 0x6000:22 [▶ 223] "Latch value 2" when a positive edge is detected at the Gate/Latch input. The counter value is then blocked.

The same applies to the use of index 0x80n00:04 "Gate polarity" (2 = "Enable neg. gate") and simultaneous activation of index 0x7000:0D [▶ 225] "Enable latch extern 2 on negative edge" at a negative edge at the Gate/Latch input.

Multiple activation of the latch function

If several commands are activated simultaneously to save the counter value in the "Latch value" process data (index 0x60n0:12 [▶ 223]), only the command that is set first is accepted by the terminal.

- The counter value is stored in the "Latch value" (index 0x60n0:12) at the next occurring event and confirmed with the corresponding bit.
- All other activated events are ignored.
- The counter value storage cannot be reactivated until all activated commands for latching the value have been deactivated. This also applies if they were activated after confirmation by the occurring event.

8.6.4 Input Status Input

If the encoder has an error signal or status output, this can be connected to the "Status Input" input of the terminal and evaluated. The input is 5 V compatible.

The error signal or status output at the encoder is usually implemented with negative logic. This is indicated by the terminal as follows:

Encoder	EL5131		
Alarm output	**Status Input (0x60n0:06 "Status of input status")**	**Status Input LED**	**Meaning**
HIGH level / output open	TRUE	Off	e.g. encoder OK
LOW level / PullDown - level is actively pulled down to LOW	FALSE	Red	e.g. encoder faulty

If an overvoltage is present at the Status Input, it is displayed as follows:

Fault diagnosis in case of overvoltage at the Status Input at channel n	Display
DiagMessage, type "Error", text ID 0x8315	Error status input (channel n)
0xA0n0:05 "Error input status"	TRUE
LED: Status Input Ch. n	Red

NOTICE

Wiring of the Status Input

In the terminal the Status Input is internally connected to 5 V via a pull-up resistor (1 kΩ). The encoder output must actively pull the signal against GND. The resistance must be dimensioned so that it is less than 120 Ω.

External power supply is not recommended. If an external supply is used, the maximum permitted voltage is 5 V against GND.

8.7 Outputs

Two 24 V_{DC} - digital outputs are available, which can be parameterized and switched independently of each other.

The following options are available for parameterization:

- Set switching characteristics [▶ 178]:
  Push-Pull [▶ 26] or Push-Tristate [▶ 26] mode
- Set state in case of bus error [▶ 179]:

Furthermore, the following options are available for switching the outputs:

manual	Via PLC variable [▶ 181]	
Automatic switching function [▶ 182]	when specified counter values [▶ 184] are reached • 1 … 8 threshold values • "Output state 1" ... "Output state 8" freely selectable (TRUE, FALSE) • Timestamp function [▶ 172] available • Switching possible for a defined time [▶ 184] • Realization of a digital cam controller [▶ 183] possible	when reaching predefined frequency and velocity values [▶ 189] • Max. 2 threshold values • "Output state" fixed ("Output state 1": TRUE, "Output state 2": FALSE) • Timestamp function [▶ 172] available

The parameterization of the outputs is described in the chapter Output function [▶ 176].

Each output has a separate short circuit/overload diagnosis (see chapter Short circuit/overload diagnosis of the outputs [▶ 217]).

9 Diagnostics

9.1 Diagnostics - basic principles of diag messages

DiagMessages designates a system for the transmission of messages from the EtherCAT Slave to the EtherCAT Master/TwinCAT. The messages are stored by the device in its own CoE under 0x10F3 and can be read by the application or the System Manager. An error message referenced via a code is output for each event stored in the device (warning, error, status change).

Definition

The *DiagMessages* system is defined in the ETG (EtherCAT Technology Group) in the guideline ETG.1020, chapter 13 “Diagnosis handling”. It is used so that pre-defined or flexible diagnostic messages can be conveyed from the EtherCAT Slave to the Master. In accordance with the ETG, the process can therefore be implemented supplier-independently. Support is optional. The firmware can store up to 250 DiagMessages in its own CoE.

Each DiagMessage consists of

- Diag Code (4-byte)
- Flags (2-byte; info, warning or error)
- Text ID (2-byte; reference to explanatory text from the ESI/XML)
- Timestamp (8-byte, local slave time or 64-bit Distributed Clock time, if available)
- Dynamic parameters added by the firmware

The DiagMessages are explained in text form in the ESI/XML file belonging to the EtherCAT device: on the basis of the Text ID contained in the DiagMessage, the corresponding plain text message can be found in the languages contained in the ESI/XML. In the case of Beckhoff products these are usually German and English.

Via the entry *NewMessagesAvailable* the user receives information that new messages are available.

DiagMessages can be confirmed in the device: the last/latest unconfirmed message can be confirmed by the user.

In the CoE both the control entries and the history itself can be found in the CoE object 0x10F3:

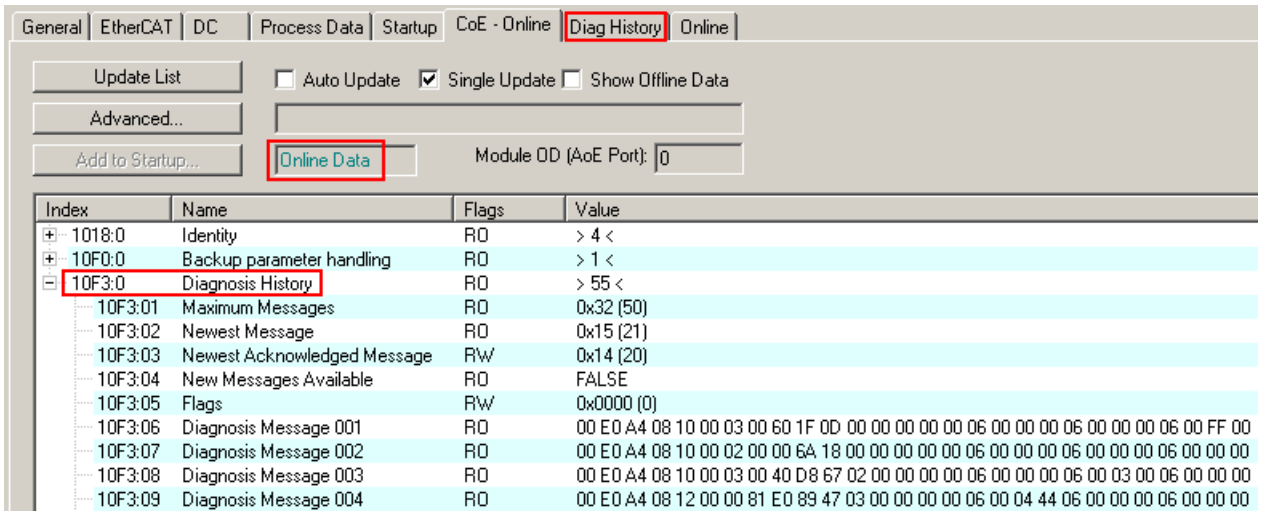


Fig. 179: DiagMessages in the CoE

The subindex of the latest *DiagMessage* can be read under 0x10F3:02.

Support for commissioning

The DiagMessages system is to be used above all during the commissioning of the plant. The diagnostic values e.g. in the StatusWord of the device (if available) are helpful for online diagnosis during the subsequent continuous operation.

TwinCAT System Manager implementation

From TwinCAT 2.11 DiagMessages, if available, are displayed in the device's own interface. Operation (collection, confirmation) also takes place via this interface.

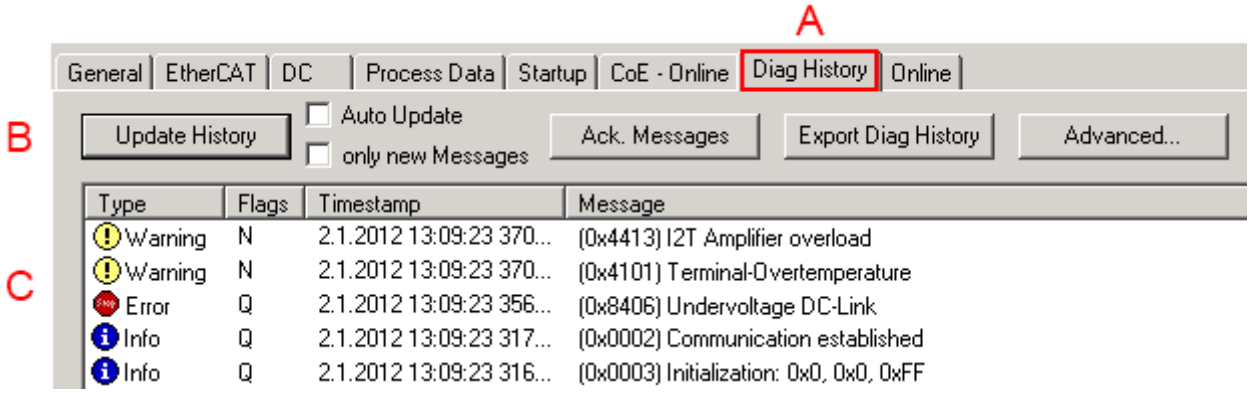


Fig. 180: Implementation of the DiagMessage system in the TwinCAT System Manager

The operating buttons (B) and the history read out (C) can be seen on the Diag History tab (A). The components of the message:

- Info/Warning/Error
- Acknowledge flag (N = unconfirmed, Q = confirmed)
- Time stamp
- Text ID
- Plain text message according to ESI/XML data

The meanings of the buttons are self-explanatory.

DiagMessages within the ADS Logger/Eventlogger

From TwinCAT 3.1 build 4022 onwards, DiagMessages sent by the terminal are shown by the TwinCAT ADS Logger. Given that DiagMessages are represented IO- comprehensive at one place, commissioning will be simplified. In addition, the logger output could be stored into a data file – hence DiagMessages are available long-term for analysis.

DiagMessages are actually only available locally in CoE 0x10F3 in the terminal and can be read out manually if required, e.g. via the DiagHistory mentioned above.

In the latest developments, the EtherCAT Terminals are set by default to report the presence of a DiagMessage as emergency via EtherCAT; the event logger can then retrieve the DiagMessage. The function is activated in the terminal via 0x10F3:05, so such terminals have the following entry in the StartUp list by default:

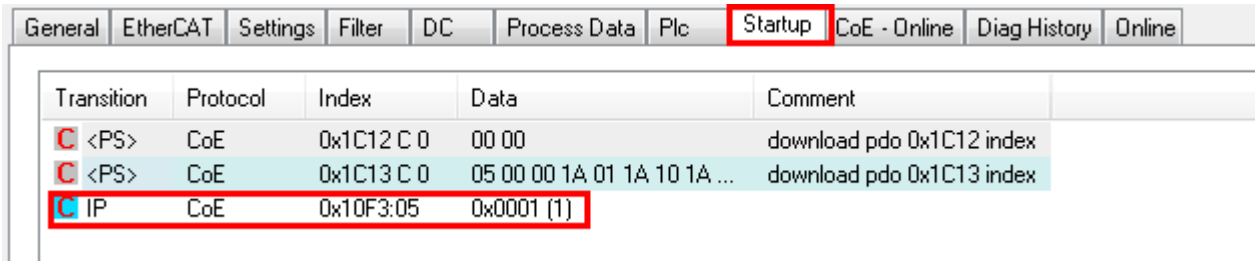


Fig. 181: Startup List

If the function is to be deactivated because, for example, many messages come in or the EventLogger is not used, the StartUp entry can be deleted or set to 0. The value can then be set back to 1 later from the PLC via CoE access if required.

Reading messages into the PLC

- In preparation -

Interpretation

Time stamp

The time stamp is obtained from the local clock of the terminal at the time of the event. The time is usually the distributed clock time (DC) from register x910.

Please note: When EtherCAT is started, the DC time in the reference clock is set to the same time as the local IPC/TwinCAT time. From this moment the DC time may differ from the IPC time, since the IPC time is not adjusted. Significant time differences may develop after several weeks of operation without a EtherCAT restart. As a remedy, external synchronization of the DC time can be used, or a manual correction calculation can be applied, as required: The current DC time can be determined via the EtherCAT master or from register x901 of the DC slave.

Structure of the Text ID

The structure of the MessageID is not subject to any standardization and can be supplier-specifically defined. In the case of Beckhoff EtherCAT devices (EL, EP) it usually reads according to **xyzz**:

x	y	zz
0: Systeminfo 2: reserved 1: Info 4: Warning 8: Error	0: System 1: General 2: Communication 3: Encoder 4: Drive 5: Inputs 6: I/O general 7: reserved	Error number

Example: Message 0x4413 --> Drive Warning Number 0x13

Overview of text IDs

Specific text IDs are listed in the device documentation.

Text ID	Type	Place	Text Message	Additional comment
0x0001	Information	System	No error	No error
0x0002	Information	System	Communication established	Connection established
0x0003	Information	System	Initialization: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1000	Information	System	Information: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1012	Information	System	EtherCAT state change Init - PreOp	
0x1021	Information	System	EtherCAT state change PreOp - Init	
0x1024	Information	System	EtherCAT state change PreOp - Safe-Op	
0x1042	Information	System	EtherCAT state change SafeOp - PreOp	
0x1048	Information	System	EtherCAT state change SafeOp - Op	
0x1084	Information	System	EtherCAT state change Op - SafeOp	
0x1100	Information	General	Detection of operation mode completed: 0x%X, %d	Detection of the mode of operation ended
0x1135	Information	General	Cycle time o.k.: %d	Cycle time OK
0x1157	Information	General	Data manually saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved manually
0x1158	Information	General	Data automatically saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved automatically
0x1159	Information	General	Data deleted (Idx: 0x%X, SubIdx: 0x%X)	Data deleted
0x117F	Information	General	Information: 0x%X, 0x%X, 0x%X	Information
0x1201	Information	Communication	Communication re-established	Communication to the field side restored This message appears, for example, if the voltage was removed from the power contacts and re-applied during operation.
0x1300	Information	Encoder	Position set: %d, %d	Position set - StartInputhandler
0x1303	Information	Encoder	Encoder Supply ok	Encoder power supply unit OK
0x1304	Information	Encoder	Encoder initialization successfully, channel: %X	Encoder initialization successfully completed
0x1305	Information	Encoder	Sent command encoder reset, channel: %X	Send encoder reset command
0x1400	Information	Drive	Drive is calibrated: %d, %d	Drive is calibrated
0x1401	Information	Drive	Actual drive state: 0x%X, %d	Current drive status
0x1705	Information		CPU usage returns in normal range (< 85%%)	Processor load is back in the normal range
0x1706	Information		Channel is not in saturation anymore	Channel is no longer in saturation
0x1707	Information		Channel is not in overload anymore	Channel is no longer overloaded
0x170A	Information		No channel range error anymore	A measuring range error is no longer active
0x170C	Information		Calibration data saved	Calibration data were saved
0x170D	Information		Calibration data will be applied and saved after sending the command “0x5AFE”	Calibration data are not applied and saved until the command "0x5AFE" is sent.

Text ID	Type	Place	Text Message	Additional comment
0x2000	Information	System	%s: %s	
0x2001	Information	System	%s: Network link lost	Network connection lost
0x2002	Information	System	%s: Network link detected	Network connection found
0x2003	Information	System	%s: no valid IP Configuration - Dhcp client started	Invalid IP configuration
0x2004	Information	System	%s: valid IP Configuration (IP: %d.%d.%d.%d) assigned by Dhcp server %d.%d.%d.%d	Valid IP configuration, assigned by the DHCP server
0x2005	Information	System	%s: Dhcp client timed out	DHCP client timeout
0x2006	Information	System	%s: Duplicate IP Address detected (%d.%d.%d.%d)	Duplicate IP address found
0x2007	Information	System	%s: UDP handler initialized	UDP handler initialized
0x2008	Information	System	%s: TCP handler initialized	TCP handler initialized
0x2009	Information	System	%s: No more free TCP sockets available	No free TCP sockets available.

Text ID	Type	Place	Text Message	Additional comment
0x4000	Warning		Warning: 0x%X, 0x%X, 0x%X	General warning; parameters depend on event. See device documentation for interpretation.
0x4001	Warning	System	Warning: 0x%X, 0x%X, 0x%X	
0x4002	Warning	System	%s: %s Connection Open (IN:%d OUT:%d API:%dms) from %d.%d.%d.%d successful	
0x4003	Warning	System	%s: %s Connection Close (IN:%d OUT:%d) from %d.%d.%d.%d successful	
0x4004	Warning	System	%s: %s Connection (IN:%d OUT:%d) with %d.%d.%d.%d timed out	
0x4005	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Error: %u)	
0x4006	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Input Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4007	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Output Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4008	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (RPI:%dms not supported -> API:%dms)	
0x4101	Warning	General	Terminal-Overtemperature	Overtemperature. The internal temperature of the terminal exceeds the parameterized warning threshold.
0x4102	Warning	General	Discrepancy in the PDO-Configuration	The selected PDOs do not match the set operating mode. Sample: Drive operates in velocity mode, but the velocity PDO is but not mapped in the PDOs.
0x417F	Warning	General	Warning: 0x%X, 0x%X, 0x%X	
0x428D	Warning	General	Challenge is not Random	
0x4300	Warning	Encoder	Subincrements deactivated: %d, %d	Sub-increments deactivated (despite activated configuration)
0x4301	Warning	Encoder	Encoder-Warning	General encoder error
0x4302	Warning	Encoder	Maximum frequency of the input signal is nearly reached (channel %d)	
0x4303	Warning	Encoder	Limit counter value was reduced because of the PDO configuration (channel %d)	
0x4304	Warning	Encoder	Reset counter value was reduced because of the PDO configuration (channel %d)	
0x4400	Warning	Drive	Drive is not calibrated: %d, %d	Drive is not calibrated
0x4401	Warning	Drive	Starttype not supported: 0x%X, %d	Start type is not supported
0x4402	Warning	Drive	Command rejected: %d, %d	Command rejected
0x4405	Warning	Drive	Invalid modulo subtype: %d, %d	Modulo sub-type invalid
0x4410	Warning	Drive	Target overrun: %d, %d	Target position exceeded
0x4411	Warning	Drive	DC-Link undervoltage (Warning)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x4412	Warning	Drive	DC-Link overvoltage (Warning)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x4413	Warning	Drive	I2T-Model Amplifier overload (Warning)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x4414	Warning	Drive	I2T-Model Motor overload (Warning)	• The motor is being operated outside the parameterized rated values.

Text ID	Type	Place	Text Message	Additional comment
				• The I2T-model of the motor is incorrectly parameterized.
0x4415	Warning	Drive	Speed limitation active	The maximum speed is limited by the parameterized objects (e.g. velocity limitation, motor speed limitation). This warning is output if the set velocity is higher than one of the parameterized limits.
0x4416	Warning	Drive	Step lost detected at position: 0x%X%X	Step loss detected
0x4417	Warning	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized warning threshold
0x4418	Warning	Drive	Limit: Current	Limit: current is limited
0x4419	Warning	Drive	Limit: Amplifier I2T-model exceeds 100%%	The threshold values for the maximum current were exceeded.
0x441A	Warning	Drive	Limit: Motor I2T-model exceeds 100%%	Limit: Motor I2T-model exceeds 100%
0x441B	Warning	Drive	Limit: Velocity limitation	The threshold values for the maximum speed were exceeded.
0x441C	Warning	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x4600	Warning	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x4610	Warning	General IO	Wrong output voltage range	Output voltage not in the correct range
0x4705	Warning		Processor usage at %d %%	Processor load at %d %%
0x470A	Warning		EtherCAT Frame missed (change Settings or DC Operation Mode or Sync0 Shift Time)	EtherCAT frame missed (change DC Operation Mode or Sync0 Shift Time under Settings)

Text ID	Type	Place	Text Message	Additional comment
0x8000	Error	System	%s: %s	
0x8001	Error	System	Error: 0x%X, 0x%X, 0x%X	General error; parameters depend on event. See device documentation for interpretation.
0x8002	Error	System	Communication aborted	Communication aborted
0x8003	Error	System	Configuration error: 0x%X, 0x%X, 0x%X	General; parameters depend on event. See device documentation for interpretation.
0x8004	Error	System	%s: Unsuccessful FwdOpen-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8005	Error	System	%s: FwdClose-Request sent to %d.%d.%d.%d (%s)	
0x8006	Error	System	%s: Unsuccessful FwdClose-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8007	Error	System	%s: Connection with %d.%d.%d.%d (%s) closed	
0x8100	Error	General	Status word set: 0x%X, %d	Error bit set in the status word
0x8101	Error	General	Operation mode incompatible to PDO interface: 0x%X, %d	Mode of operation incompatible with the PDO interface
0x8102	Error	General	Invalid combination of Inputs and Outputs PDOs	Invalid combination of input and output PDOs
0x8103	Error	General	No variable linkage	No variables linked
0x8104	Error	General	Terminal-Overtemperature	The internal temperature of the terminal exceeds the parameterized error threshold. Activation of the terminal is prevented
0x8105	Error	General	PD-Watchdog	Communication between the fieldbus and the output stage is secured by a Watchdog. The axis is stopped automatically if the fieldbus communication is interrupted. • The EtherCAT connection was interrupted during operation. • The Master was switched to Config mode during operation.
0x8135	Error	General	Cycle time has to be a multiple of 125 µs	The IO or NC cycle time divided by 125 µs does not produce a whole number.
0x8136	Error	General	Configuration error: invalid sampling rate	Configuration error: Invalid sampling rate
0x8137	Error	General	Electronic type plate: CRC error	Content of the external name plate memory invalid.
0x8140	Error	General	Sync Error	Real-time violation
0x8141	Error	General	Sync%X Interrupt lost	Sync%X Interrupt lost
0x8142	Error	General	Sync Interrupt asynchronous	Sync Interrupt asynchronous
0x8143	Error	General	Jitter too big	Jitter limit violation
0x817F	Error	General	Error: 0x%X, 0x%X, 0x%X	
0x8200	Error	Communication	Write access error: %d, %d	Error while writing
0x8201	Error	Communication	No communication to field-side (Auxiliary voltage missing)	• There is no voltage applied to the power contacts. • A firmware update has failed.
0x8281	Error	Communication	Ownership failed: %X	
0x8282	Error	Communication	To many Keys founded	
0x8283	Error	Communication	Key Creation failed: %X	
0x8284	Error	Communication	Key loading failed	
0x8285	Error	Communication	Reading Public Key failed: %X	
0x8286	Error	Communication	Reading Public EK failed: %X	
0x8287	Error	Communication	Reading PCR Value failed: %X	
0x8288	Error	Communication	Reading Certificate EK failed: %X	
0x8289	Error	Communication	Challenge could not be hashed: %X	
0x828A	Error	Communication	Tickstamp Process failed	
0x828B	Error	Communication	PCR Process failed: %X	
0x828C	Error	Communication	Quote Process failed: %X	
0x82FF	Error	Communication	Bootmode not activated	Boot mode not activated
0x8300	Error	Encoder	Set position error: 0x%X, %d	Error while setting the position

Text ID	Type	Place	Text Message	Additional comment
0x8301	Error	Encoder	Encoder increments not configured: 0x%X, %d	Encoder increments not configured
0x8302	Error	Encoder	Encoder error	The amplitude of the resolver is too small
0x8303	Error	Encoder	Encoder power missing (channel %d)	
0x8304	Error	Encoder	Encoder communication error, channel: %X	Encoder communication error
0x8305	Error	Encoder	EnDat2.2 is not supported, channel: %X	EnDat2.2 is not supported
0x8306	Error	Encoder	Delay time, tolerance limit exceeded, 0x%X, channel: %X	Runtime measurement, tolerance exceeded
0x8307	Error	Encoder	Delay time, maximum value exceeded, 0x%X, channel: %X	Runtime measurement, maximum value exceeded
0x8308	Error	Encoder	Unsupported ordering designation, 0x%X, channel: %X (only 02 and 22 is supported)	Wrong EnDat order ID
0x8309	Error	Encoder	Encoder CRC error, channel: %X	Encoder CRC error
0x830A	Error	Encoder	Temperature %X could not be read, channel: %X	Temperature cannot be read
0x830C	Error	Encoder	Encoder Single-Cycle-Data Error, channel. %X	CRC error detected. Check the transmission path and the CRC polynomial
0x830D	Error	Encoder	Encoder Watchdog Error, channel. %X	The sensor has not responded within a predefined time period
0x8310	Error	Encoder	Initialisation error	
0x8311	Error	Encoder	Maximum frequency of the input signal is exceeded (channel %d)	
0x8312	Error	Encoder	Encoder plausibility error (channel %d)	
0x8313	Error	Encoder	Configuration error (channel %d)	
0x8314	Error	Encoder	Synchronisation error	
0x8315	Error	Encoder	Error status input (channel %d)	
0x8400	Error	Drive	Incorrect drive configuration: 0x%X, %d	Drive incorrectly configured
0x8401	Error	Drive	Limiting of calibration velocity: %d, %d	Limitation of the calibration velocity
0x8402	Error	Drive	Emergency stop activated: 0x%X, %d	Emergency stop activated
0x8403	Error	Drive	ADC Error	Error during current measurement in the ADC
0x8404	Error	Drive	Overcurrent	Overcurrent in phase U, V or W
0x8405	Error	Drive	Invalid modulo position: %d	Modulo position invalid
0x8406	Error	Drive	DC-Link undervoltage (Error)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x8407	Error	Drive	DC-Link overvoltage (Error)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x8408	Error	Drive	I2T-Model Amplifier overload (Error)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x8409	Error	Drive	I2T-Model motor overload (Error)	• The motor is being operated outside the parameterized rated values. • The I2T-model of the motor is incorrectly parameterized.
0x840A	Error	Drive	Overall current threshold exceeded	Total current exceeded
0x8415	Error	Drive	Invalid modulo factor: %d	Modulo factor invalid
0x8416	Error	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized error threshold. The motor stops immediately. Activation of the output stage is prevented.
0x8417	Error	Drive	Maximum rotating field velocity exceeded	Rotary field speed exceeds the value specified for dual use (EU 1382/2014).
0x841C	Error	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.

Text ID	Type	Place	Text Message	Additional comment
0x8550	Error	Inputs	Zero crossing phase %X missing	Zero crossing phase %X missing
0x8551	Error	Inputs	Phase sequence Error	Wrong direction of rotation
0x8552	Error	Inputs	Overcurrent phase %X	Overcurrent phase %X
0x8553	Error	Inputs	Overcurrent neutral wire	Overcurrent neutral wire
0x8581	Error	Inputs	Wire broken Ch %D	Wire broken Ch %d
0x8600	Error	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x8601	Error	General IO	Supply voltage to low	Supply voltage too low
0x8602	Error	General IO	Supply voltage to high	Supply voltage too high
0x8603	Error	General IO	Over current of supply voltage	Overcurrent of supply voltage
0x8610	Error	General IO	Wrong output voltage range	Output voltage not in the correct range
0x8611	Error	General IO	Output voltage to low	Output voltage too low
0x8612	Error	General IO	Output voltage to high	Output voltage too high
0x8613	Error	General IO	Over current of output voltage	Overcurrent of output voltage
0x8700	Error		Channel/Interface not calibrated	Channel/interface not synchronized
0x8701	Error		Operating time was manipulated	Operating time was manipulated
0x8702	Error		Oversampling setting is not possible	Oversampling setting not possible
0x8703	Error		No slave controller found	No slave controller found
0x8704	Error		Slave controller is not in Bootstrap	Slave controller is not in bootstrap
0x8705	Error		Processor usage to high (>= 100%%)	Processor load too high (>= 100%%)
0x8706	Error		Channel in saturation	Channel in saturation
0x8707	Error		Channel overload	Channel overload
0x8708	Error		Overloadtime was manipulated	Overload time was manipulated
0x8709	Error		Saturationtime was manipulated	Saturation time was manipulated
0x870A	Error		Channel range error	Measuring range error for the channel
0x870B	Error		no ADC clock	No ADC clock available
0xFFFF	Information		Debug: 0x%X, 0x%X, 0x%X	Debug: 0x%X, 0x%X, 0x%X

9.2 EL5131 - Diagnostics

The EL5131 offers the following diagnostic options:

- Overflow and underflow of the counter limits [▶ 212]
- RS422 - wire break and short circuit detection (open circuit) [▶ 213]
- Monitoring of the encoder operating voltage [▶ 214]
- Status Input diagnostics [▶ 214]
- Plausibility check [▶ 215]
- Exceeding the filter frequency [▶ 216]
- Short circuit/overload diagnosis of outputs [▶ 217]

9.2.1 Overflow and underflow of the counter limits

Counter overflow and Counter underflow

An overflow or underflow of the counter limits is indicated by the process data 0x6000:04 "Counter underflow" or 0x6000:05 "Counter overflow".

- The "Counter underflow" bit in index 0x6000:04 is set when an underflow 0x8001:1B "Reset counter value" → 0x8001:1A "Limit counter value" occurs.
  - With the preset parameters, this corresponds to "..00 → ..FF"
    It is reset if 2/3 of the counter range are underrun.
  - For counter limits with fixed value range, this corresponds to:
    2/3 * ("Limit counter value" - "Reset counter value") + "Reset counter value"
- The "Counter overflow" bit 0x6000:05 is set when an overflow 0x8001:1A "Limit counter value" → 0x8001:1B "Reset counter value" occurs.
  - With default parameters "..FF → ..00"
    It is reset if 1/3 of the counter range is exceeded.
  - For counter limits with fixed value range, this corresponds to:
    1/3 * ("Limit counter value" - "Reset counter value") + "Reset counter value"

Example 1:

0x8001:1A "Limit counter value" = 2^{12}-1 = 4095

0x8001:1B "Reset counter value" = 0

"Counter underflow" bit is reset when: 2/3 * 4095 = 2730 is reached.

"Counter overflow" bit is reset when: 1/3 * 4095 = 1365 is reached.

Example 2:

0x8001:1A "Limit counter value" = 2^{12}-1 = 4095

0x8001:1B "Reset counter value" = 400

"Counter underflow" bit is reset when: 2/3 * (4095-400) +400 = 2463 is reached.

"Counter overflow" bit is reset when: 1/3 * (4095-95) +400 = 1232 is reached.

9.2.2 RS422 - wire break and short-circuit detection (open circuit)

In the RS422 (differential input) modes it is possible to detect a wire break or short circuit at the individual encoder inputs.

- In case of wire break, e.g. between input A and input $\overline{A}$,
  - the differential voltage V_{ID} is almost 0 V,
  - which leads to an error with low differential voltage.
- In case of a short circuit, e.g. between input A and input $\overline{A}$, the error behavior is similar to a wire break and also leads to error detection.

Activation of error detection for each channel			
Index (hex)	**Name**	**Description**	
80n0:0B *	Error detection A	TRUE	Broken wire and short circuit detection for encoder input A enabled
		FALSE	Broken wire and short circuit detection for encoder input A disabled
80n0:0C *	Error detection B	TRUE	Broken wire and short circuit detection for encoder input B enabled
		FALSE	Broken wire and short circuit detection for encoder input B disabled
80n0:0D *	Error detection C	TRUE	Broken wire and short circuit detection for encoder input C enabled
		FALSE	Broken wire and short circuit detection for encoder input C disabled
*) depending on the number of channels (n = 0 for channel 1 and n = 1 for channel 2)			

Error detection using the example of a wire break or short circuit between inputs A and $\overline{A}$.		
Error diagnosis	**Display**	**Description**
LED A1	Green	A TRUE level is present
	Red	An error (open circuit) was detected
0x60n0:07 "Open circuit"	TRUE	Group error message for "Open circuit" A wire break or short circuit has occurred at one of the encoder inputs
	FALSE	There is no "open circuit" error.
0x60n0:0F "TxPDO State"	TRUE	The position data are invalid.
	FALSE	The position data are valid.
0xA0n0:01 "Error A"	TRUE	An "open circuit" error (wire break or short circuit) has occurred at encoder input A.
	FALSE	There is no "open circuit" error.
Diag Message	Text ID: 0x831B	Wire break or short circuit track A (channel n)

Error bits not permanently set in case of wire break at an encoder input

If a wire break is only present at one encoder input (e.g. only track A), it may happen in individual cases that the differential voltage (V_{ID}) is above the limit range (V_{IDLow}) due to the applied common mode voltage (V_{CM}).
This means that the error is not clearly identified.

The corresponding error bits ("Open circuit" and "Error A") are not permanently present!

9.2.3 Monitoring of the encoder operating voltage

A lack of encoder operating voltage is indicated by:

Error diagnosis	Display	Description
LED Power ENC. Ch.1	green	Encoder supply voltage is present
	off	Encoder supply voltage is not present
0xA0n0:04 "Field power failure"	TRUE	Encoder supply voltage is not present
	FALSE	Encoder supply voltage is present
DiagMessage, text ID	0x8303	Encoder supply voltage is not present

NOTICE

Encoder operating voltage generated from the 24 V_{DC} power contacts

The encoder operating voltage is generated from the 24 V_{DC} supply of the power contacts.
If the power contacts have no supply, no encoder operating voltage can be provided.

9.2.4 Status Input diagnostics

If the encoder has an error signal or status output, this can be connected to the "Status Input" input of the terminal and evaluated. The input is 5 V compatible.

The error signal or status output at the encoder is usually implemented with negative logic. This is indicated by the terminal as follows:

Encoder	EL5131		
Alarm output	**Status Input (0x60n0:06 "Status of input status")**	**Status Input LED**	**Meaning**
HIGH level / output open	TRUE	Off	e.g. encoder OK
LOW level / PullDown - level is actively pulled down to LOW	FALSE	Red	e.g. encoder faulty

If an overvoltage is present at the Status Input, it is displayed as follows:

Fault diagnosis in case of overvoltage at the Status Input at channel n	Display
DiagMessage, type "Error", text ID 0x8315	Error status input (channel n)
0xA0n0:05 "Error input status"	TRUE
LED: Status Input Ch. n	Red

9.2.5 Plausibility check

The plausibility check of the input signals serves as an extended diagnosis to detect interference signals and to identify and suppress the step changes in the counter value caused by them.

Functional principle of the plausibility check

The square wave signals of an incremental encoder with track A and B are phase-shifted by 90°. This means that only certain transitions in the signal curve are permitted or "plausible", e.g. with a rising A-edge the signal on track B cannot also rise.

The plausibility of the signal sequence is checked in the terminal. If invalid signal transitions occur, these are detected and displayed accordingly if plausibility detection is activated.

Plausibility check sequence

- The plausibility check can be enabled via index 0x8000:21 "Enable encoder plausibility check"
- If a plausibility error is detected, it is displayed as follows:

Error diagnosis	Description
DiagMassage, type "Error", text ID 0x8312	Encoder plausibility error (channel n)
0xA0n0:13 "Encoder plausibility error counter"	The error counter is incremented when a plausibility error is detected
0x60n2:0E TxPDO State = 1	The associated TxPDO data are not valid

The plausibility error counter can be reset as follows:

- Transition from PREOP to OP

or

- enter the following in the command object in index 0xFB00:01 "Request"

0xFB00:01 "Request"	Description
0x9151	Sets the internal error counter 0xA000:13 "Encoder plausibility error counter" for channel 1 to zero

9.2.6 Exceeding the filter frequency

Input filters are used for interference suppression at the encoder and digital inputs. Different filter frequencies can be parameterized according to the application.

The individual inputs are subject to the following filter frequencies:

Inputs	Max. recommended filter frequency
Encoder inputs: track A, track B, track C	RS422 mode: 5 MHz TTL mode: 1 MHz Open Collector: 100 kHz
Latch input	1 MHz
Gate/Latch input	1 MHz
Status Input	100 kHz (not adjustable)

- If a frequency overrun is detected when the filter is enabled (0x8000:08 and 0x8001:19), a counter in index 0xA000:16 is incremented.

Index (hex)	Name	Description
0xA000:14	Filter violation counter extern latch	Filter frequency overshoot counter for the Latch extern input
0xA000:15	Filter violation counter input gate	Filter frequency overshoot counter for the Gate input
0xA000:16	Filter violation counter	Filter frequency overshoot counter for the encoder input signals with enabled filter

NOTICE



Note Parameterization of adjustable interference pulse filters

Please note the description and notes for parameterization in chapter “Adjustable interference pulse filters [▶ 173]”.

9.2.7 Short circuit/overload diagnosis of outputs

Each output has its own short circuit/overload diagnosis.

Setting the response time per output				
Index (hex)	**Name**	**Value**	**Name**	**Reaction time***
8001:22 8001:23	Diagnosis output Diagnosis output 2	0_{dec}	Disable	Error detection disabled
		1_{dec}	Fast	10 ms
		2_{dec}	Standard (default)	500 ms
		3_{dec}	Slow	1 s
*The reaction time applies to the LED display, the "Diag Messages" are sent max. every second regardless of the setting.				

If there is a short circuit/overload of the output, this is indicated as follows:

Error diagnosis in case of short circuit/overload of the output	Display
LED: Digital output 1 Digital output 2	• If an error is present for at least the time set in 0x8001:22/23 "Diagnosis output"/"Diagnosis output 2", the error detection is indicated via the LED. • If no more errors are detected, the LED goes out after a switch-off delay of 1 s.
0xA000:08 "Error output" 0xA000:09 "Error output 2"	TRUE
DiagMessage, type "Error", text ID 0x8318	Error output (channel %d, output %d)

10 EL5131 - object description and parameterization

EtherCAT XML Device Description

The presentation 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 (double-click on the respective object) or via the Process Data tab (assignment of PDOs).

10.1 Restore object

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters [▶ 256]	Restore default parameters	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If this object is set to **"0x64616F6C"** in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0_{dec})

10.2 Configuration data

Index 8000 ENC Settings 0 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	ENC Settings 0 Ch.1	Maximum subindex	UINT8	RO	0x23 (35_{dec})
8000:01	Enable C reset [▶ 154]	The counter is reset via the C input.	BOOLEAN	RW	0x00 (0_{dec})
8000:02	Enable extern reset	A counter reset is triggered via the external latch input (24 V)	BOOLEAN	RW	0x00 (0_{dec})
8000:04	Gate polarity [▶ 163]	0: Disable gate 1: Enable pos. gate (gate disables with HIGH level) 2: Enable neg. gate (gate disables with LOW level)	BIT2	RW	0x01 (1_{dec})
8000:06	Evaluation mode [▶ 149]	0: 4-fold (quadruple evaluation) 1: 1-fold (single evaluation) 2: 2-fold (double evaluation)	BIT2	RW	0x00 (0_{dec})
8000:08	Disable filter [▶ 173]	0: Enables input filter (inputs A, /A, B, /B, C, /C only) 1: Disables input filter If a filter is enabled, a signal edge must be present for at least 2.4 µs in order to be counted as an increment.	BOOLEAN	RW	0x01 (1_{dec})
8000:0A	Enable micro increments [▶ 170]	When enabled, the terminal interpolates microincrements between the integer encoder increments in DC mode. The lower 8 bits of the counter value are used for display. A 32-bit counter thus becomes a 24+8 bit counter, a 16-bit counter becomes an 8+8 bit counter.	BOOLEAN	RW	0x00 (0_{dec})
8000:0B	Error detection A	A wire break or short circuit on track A is indicated in index 0x6000:07 [▶ 223] and as process data. Diagnostics is only possible, if the associated input is wired differentially.	BOOLEAN	RW	0x01 (1_{dec})
8000:0C	Error detection B	A wire break or short circuit on track B is indicated in index 0x6000:07 [▶ 223] and as process data. Diagnostics is only possible, if the associated input is wired differentially.	BOOLEAN	RW	0x01 (1_{dec})
8000:0D	Error detection C	A wire break or short circuit on track C is indicated in index 0x6000:07 [▶ 223] and as process data. Diagnostics is only possible, if the associated input is wired differentially.	BOOLEAN	RW	0x00 (0_{dec})
8000:0E	Reversion of rotation [▶ 152]	Activates reversion of rotation	BOOLEAN	RW	0x00 (0_{dec})
8000:10	Extern reset polarity	0: Fall (with the falling edge the counter is set to zero) 1: Rise (with the rising edge the counter is set to zero)	BIT1	RW	0x01 (1_{dec})
8000:11	Frequency window [▶ 164]	This is the minimum time over which the frequency is determined; default value 10 ms [resolution: 1 µs]. The frequency is calculated from the number of increments (changes of position value) in the “frequency window” time interval. The determined frequency is output in index 0x6000:13 [▶ 223] and as process data. The frequency calculation is carried out locally without distributed clocks function.	UINT16	RW	0x2710 (10000_{dec})
8000:13	Frequency scaling [▶ 164]	Scaling of the frequency measurement (must be divided by this value to obtain the unit in Hz): 100: "0.01 Hz" (default) 1: "1 Hz"	UINT32	RW	0x00000064 (100_{dec})
8000:14	Period scaling	Resolution of the period value in the process data: 10: "10 ns" Period duration value is multiple of 10 ns 100: "100 ns" Period duration value is multiple of 100 ns 500: "500 ns" Period duration value is multiple of 500 ns	UINT32	RW	0x0000000A (10_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:17	Frequency Wait Time [▶ 164]	Waiting time [ms] for frequency measurement If no full period is detected within the Frequency window [▶ 165], another frequency window is started to record a full period. This happens until the maximum frequency wait time has elapsed. At least double the period value of the minimum frequency to be measured should be entered here. t >= 2* (1 / f_{min})	UINT16	RW	0x53E2 (21474_{dec})
8000:1D	Frequency numerator [▶ 167]	Frequency numerator, frequency scaling	UINT32	RW	0x00000001 (1_{dec})
8000:1E	Frequency denominator [▶ 167]	Frequency denominator, used for scaling the frequency and the velocity calculation (increments/unit).	UINT32	RW	0x00000001 (1_{dec})
8000:21	Enable encoder plausibility check [▶ 215]	Activation of plausibility check	BOOLEAN	RW	0x00 (0_{dec})
8000:22	Enable continuous latch extern [▶ 160]	FALSE: The following pulses at the Latch input have no influence on the latch value in index 0x60n0:12 "Latch value" when the bit in index 0x70n0:02 or 0x70n0:04 is set. TRUE: The counter value is written to index 0x60n00:12 "Latch value" at each parameterized edge at the latch input. A renewed activation of index 0x70n0:02 or 0x70n0:04 is omitted.	BOOLEAN	RW	0x00 (0_{dec})
8000:23	Enable continuous latch extern 2 [▶ 161]	FALSE: The following pulses at the Latch extern 2 input have no influence on the latch value in index 0x60n0:22 "Latch value 2" when the bit in index 0x70n0:0C or 0x70n0:0D is set. TRUE: The counter value is written to index 0x60n0:22 "Latch value 2" with each parameterized edge at the Latch extern 2 input. A renewed activation of index 0x70n0:02 or 0x70n0:04 is not necessary.	BOOLEAN	RW	0x00 (0_{dec})

Index 8001 ENC Settings 1 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
8001:0	ENC Settings 1 Ch.1	Maximum subindex	UINT8	RO	0x23 (35_{dec})
8001:17	Supply voltage [▶ 149]	Setting the encoder supply 50_{dec}: 5.0 V (default) 120_{dec}: 12.0 V 240_{dec}: 24.0 V Refer to the Note on setting the encoder supply [▶ 149]	UINT32	RW	0x00000032 (50_{dec})
8001:19	Filter settings [▶ 173]	Filter settings: 10_{dec}: 10 kHz 25_{dec}: 25 kHz 50_{dec}: 50 kHz 100_{dec}: 100 kHz 250_{dec}: 250 kHz 500_{dec}: 500 kHz 1000_{dec}: 1 MHz 2500_{dec}: 2.5 MHz 5000_{dec}: 5 MHz (default)	UINT32	RW	0x00001388 (5000_{dec})
8001:1A	Limit counter value [▶ 149]	Specifies the value for the upper counter limit.	UINT32	RW	0xFFFFFFFF (-1_{dec})
8001:1B	Reset counter value [▶ 149]	Specifies the value for the lower counter limit.	UINT32	RW	0x00000000 (0_{dec})
8001:1C	Direction inversion hysteresis [▶ 157]	Enter the hysteresis in number of increments. A value greater than 0 must be selected. If the counter value exceeds the value, the bit in index 0x60n2:13 "Direction inversion detected" is set in the next PLC cycle.	UINT8	RW	0x0A (10_{dec})
8001:1D	Counter mode [▶ 148]	0: Encoder RS422 (diff. Input) 1: Counter RS422 (diff. Input) 2: Encoder TTL (single ended) 3: Counter TTL (single ended) 4: Encoder open collector 5: Counter open collector	UINT32	RW	0x00000000 (0_{dec})
8001:1E	Output function characteristic [▶ 178]	Input of the switching characteristics for output 1 0: TRUE=Push, FALSE=Pull 1: TRUE=Push, FALSE=Tristate	UINT8	RW	0x00 (0_{dec})
8001:1F	Output 2 function characteristic [▶ 178]	Input of the switching characteristics for output 2 0: TRUE=Push, FALSE=Pull 1: TRUE=Push, FALSE=Tristate	UINT8	RW	0x00 (0_{dec})
8001:20	Threshold base value [▶ 182]	Input of the reference value for the switching value of output 1: 0: Counter value 2: Frequency value	UINT8	RW	0x00 (0_{dec})
8001:21	Threshold base value 2 [▶ 182]	Input of the reference value for the switching value of output 2: 0: Counter value 2: Frequency value	UINT8	RW	0x00 (0_{dec})
8001:22	Diagnosis output [▶ 217]	Setting the response time for short-circuit/overload diagnostics at output 1 0: Disable 1: Fast 2: Standard (default) 3: Slow	UINT8	RW	0x02 (2_{dec})
8001:23	Diagnosis output 2 [▶ 217]	Setting the response time for short-circuit/overload diagnostics at output 2 0: Disable 1: Fast 2: Standard (default) 3: Slow	UINT8	RW	0x02 (2_{dec})

Index 80n0 DIG Safe State Active Ch.n

(n = 2 for Ch.1; n = 3 for Ch.2)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	DIG Safe State Active Ch.n	Maximum subindex	UINT8	RO	0x01 (1_{dec})
80n0:01	Active	Enabling of the output state defined in index 0x80n1:01 in case of a bus error 0: Output keeps the current state. 1: Output is switched to the state defined in index 0x80n1.	BOOLEAN	RW	0x01 (1_{dec})

Index 80n1 DIG Safe State Value Ch.n

(n = 2 for Ch.1; n = 3 for Ch.2)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n1:0	DIG Safe State Value Ch.n	Maximum subindex	UINT8	RO	0x01 (1_{dec})
80n1:01	Value	Defines the state of the output in case of a bus error: 0: Output off 1: Output on	BOOLEAN	RW	0x00 (0_{dec})

10.3 Input data

Index 6000 ENC Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	ENC Inputs Ch.1	Maximum subindex	UINT8	RO	0x25 (37_{dec})
6000:01	Latch C valid [▶ 162]	The counter value was stored with the zero pulse C input.	BOOLEAN	RO	0x00 (0_{dec})
6000:02	Latch extern valid [▶ 160]	The counter value was stored via the Latch extern input.	BOOLEAN	RO	0x00 (0_{dec})
6000:03	Set counter done [▶ 155]	The counter was set.	BOOLEAN	RO	0x00 (0_{dec})
6000:04	Counter underflow [▶ 212]	The value has fallen below the lower counter limit. The bit is reset when the counter value has fallen below 2/3 of the counting range.	BOOLEAN	RO	0x00 (0_{dec})
6000:05	Counter overflow [▶ 212]	The upper counter limit was exceeded. The bit is reset when the counter value has exceeded 1/3 of the counting range.	BOOLEAN	RO	0x00 (0_{dec})
6000:06	Status of input status [▶ 200]	State of the status input (alarm "input 1")	BOOLEAN	RO	0x00 (0_{dec})
6000:07	Open circuit	Indicates a wire break. Configuration via index 0x8000:0A, 0x8000:0B, 0x8000:0C [▶ 219]	BOOLEAN	RO	0x00 (0_{dec})
6000:08	Extrapolation stall	The extrapolated part of the counter is invalid. The velocity has fallen below the minimum velocity required to use the microincrements [▶ 170]	BOOLEAN	RO	0x00 (0_{dec})
6000:09	Status of input A	Status of input A	BOOLEAN	RO	0x00 (0_{dec})
6000:0A	Status of input B	Status of input B	BOOLEAN	RO	0x00 (0_{dec})
6000:0B	Status of input C	Status of input C	BOOLEAN	RO	0x00 (0_{dec})
6000:0C	Status of input gate [▶ 163]	The state of the gate input	BOOLEAN	RO	0x00 (0_{dec})
6000:0D	Status of extern latch [▶ 160]	Status of the Latch extern input	BOOLEAN	RO	0x00 (0_{dec})
6000:0E	Sync Error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle. This means a SYNC signal was triggered in the terminal, although no new process data were available (0 = OK, 1 = NOK).	BOOLEAN	RO	0x00 (0_{dec})
6000:0F	TxPDO State	Only in Legacy mode: Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).	BOOLEAN	RO	0x00 (0_{dec})
6000:10	TxPDO Toggle	Only in Legacy mode: The TxPDO Toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
6000:11	Counter value [▶ 148]	Counter value	UINT32	RO	0x00000000 (0_{dec})
6000:12	Latch value [▶ 160]	Latch value	UINT32	RO	0x00000000 (0_{dec})
6000:13	Frequency value [▶ 164]	The frequency (setting of the scaling in index 0x8000:13 [▶ 219])	UINT32	RO	0x00000000 (0_{dec})
6000:14	Period value [▶ 166]	The period value (setting of the scaling in index 0x8000:14 [▶ 219])	UINT32	RO	0x00000000 (0_{dec})
6000:16	Timestamp [▶ 172]	Timestamp of the last counter change	UINT64	RO	
6000:1F	Timestamp C [▶ 172]	Timestamp of the last registered positive edge of zero pulse C	UINT64	RO	

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:20	Timestamp latch [▶ 173]	Timestamp of the last edge (depending on the parameterization of the Latch input) at Latch extern.	UINT64	RO	
6000:21	Timestamp latch 2 [▶ 173]	When the Gate/Latch input is used as Latch extern 2 input: Timestamp of the last edge (depending on the parameterization of the Gate/Latch input) at the Latch extern 2 input.	UINT64	RO	
6000:22	Latch value 2 [▶ 173]	Latch value of the Latch extern 2 input (Gate input is used as second Latch input)	UINT32	RO	0x00000000 (0_{dec})
6000:23	Duty cycle [▶ 169]	indicates the ratio of pulse duration / period value.	UINT16	RO	0x0000 (0_{dec})
6000:24	Duty cycle min	Returns the smallest measured Duty cycle value	UINT16	RO	0x0000 (0_{dec})
6000:25	Duty cycle max	Returns the largest measured Duty cycle value	UINT16	RO	0x0000 (0_{dec})

Index 6002 ENC Inputs status Ch. 1

Index (hex)	Name	Meaning	Data type	Flags	Default
6002:0	ENC Inputs status Ch. 1	Maximum subindex	UINT8	RO	0x15 (21_{dec})
6002:0D	Diag	Indicates that a new message is available in the "Diag History"	BOOLEAN	RO	0x00 (0_{dec})
6002:0E	TxPDO State	Validity of the data of the associated TxPDO (0 = valid, 1 = invalid)	BOOLEAN	RO	0x00 (0_{dec})
6002:0F	Input cycle counter	2-bit counter for synchronization (incremented only if a new value is present)	BIT2	RO	0x00 (0_{dec})
6002:11	Software gate valid	0: Counter unlocked (index 0x70n0:09 [▶ 225] "Set software gate" = FALSE) 1: Counter locked (index 0x70n0:09 "Set software gate" = TRUE)	BOOLEAN	RO	0x00 (0_{dec})
6002:12	Latch extern 2 valid [▶ 161]	0: A new counter value can be stored in index 0x6000:22 [▶ 223] "Latch value 2". 1: No further counter values are stored in 0x6000:22 "Latch value 2".	BOOLEAN	RO	0x00 (0_{dec})
6002:13	Direction inversion detected [▶ 157]	Indicates reversal of the counting direction	BOOLEAN	RO	0x00 (0_{dec})
6002:14	Status of extern latch [▶ 160]	Status of the ext. Latch extern input	BOOLEAN	RO	0x00 (0_{dec})
6002:15	Counter value out of range [▶ 149]	Indicates that the counter value lies outside the parameterized counter limits.	BOOLEAN	RO	0x00 (0_{dec})

Index 60n3 ENC Inputs Ch.n (for n = 0 [output 1], n = 1 [output 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
60n3:0	ENC Inputs Ch.n	Maximum subindex	UINT8	RO	0x02 (2_{dec})
60n3:01	Status of digital output	Status of the digital output n	BOOLEAN	RO	0x00 (0_{dec})
60n3:02	Status of tristate	Tristate status of the digital output n	BOOLEAN	RO	0x00 (0_{dec})

Index 60n4 ENC Timestamps Output Ch.n (for n = 0 [output 1], n = 1 [output 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
60n4:0	ENC Timestamps Output Ch.n	Maximum subindex	UINT8	RO	0x18 (24_{dec})
60n4:11	Timestamp 1	Timestamp when switching the output if Threshold 1 is reached	ULINT64	RO	0x0 (0_{dec})
60n4:12	Timestamp 2	Timestamp when switching the output if Threshold 2 is reached	ULINT64	RO	0x0 (0_{dec})
60n4:13	Timestamp 3	Timestamp when switching the output if Threshold 3 is reached	ULINT64	RO	0x0 (0_{dec})
60n4:14	Timestamp 4	Timestamp when switching the output if Threshold 4 is reached	ULINT64	RO	0x0 (0_{dec})
60n4:15	Timestamp 5	Timestamp when switching the output if Threshold 5 is reached	ULINT64	RO	0x0 (0_{dec})
60n4:16	Timestamp 6	Timestamp when switching the output if Threshold 6 is reached	ULINT64	RO	0x0 (0_{dec})
60n4:17	Timestamp 7	Timestamp when switching the output if Threshold 7 is reached	ULINT64	RO	0x0 (0_{dec})
60n4:18	Timestamp 8	Timestamp when switching the output if Threshold 8 is reached	ULINT64	RO	0x0 (0_{dec})

10.4 Output data

Index 7000 ENC Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	ENC Outputs Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7000:01	Enable latch C [▶ 162]	Enable saving via the zero pulse C input.	BOOLEAN	RO	0x00 (0_{dec})
7000:02	Enable latch extern on positive edge [▶ 160]	Enable saving via the Latch extern input with positive edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:03	Set counter [▶ 155]	Set counter value	BOOLEAN	RO	0x00 (0_{dec})
7000:04	Enable latch extern on negative edge [▶ 160]	Enable saving via the Latch extern input with negative edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:08	Set counter on latch C [▶ 155]	Enable counter value specification via the zero pulse C input	BOOLEAN	RO	0x00 (0_{dec})
7000:09	Set software gate [▶ 163]	Locks the counter via a PLC variable 0: Counter is unlocked 1: Counter is locked	BOOLEAN	RO	0x00 (0_{dec})
7000:0A	Set counter on latch extern on positive edge [▶ 156]	Activates the counter value specification via a positive edge at the Latch extern input. The counter value is specified in index 0x7000:11 "Set counter value".	BOOLEAN	RO	0x00 (0_{dec})
7000:0B	Set counter on latch extern on negative edge [▶ 156]	Activates the counter value specification via a negative edge at the Latch extern input. The counter value is specified in index 0x7000:11 "Set counter value".	BOOLEAN	RO	0x00 (0_{dec})
7000:0C	Enable latch extern 2 on positive edge [▶ 161]	Enable saving via the Gate/Latch input with positive edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:0D	Enable latch extern 2 on negative edge [▶ 161]	Enable saving via the Gate/Latch input with negative edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:11	Set counter value [▶ 156]	The counter value to be set via "Set counter" (index 0x7000:03).	UINT32	RO	0x00000000 (0_{dec})

Index 70n1 ENC Control Output Ch.n (for n = 0 [output 1], n = 1 [output 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
70n1:0	ENC Control Output Ch.n	Maximum subindex	UINT8	RO	0x05 (5_{dec})
70n1:01	Control_Set digital output [▶ 181]	Set output via PLC	BOOLEAN	RO	0x00 (0_{dec})
70n1:02	Control_Enable thresholds output [▶ 184]	Enable automatic switching function for the output. As soon as this is enabled, the preset switching state is ignored via the PLC variable.	BOOLEAN	RO	0x00 (0_{dec})
70n1:03	Control_Set tristate [▶ 181]	Set output via PLC	BOOLEAN	RO	0x00 (0_{dec})
70n1:04	Control_Disable thresholds cw [▶ 184]	Setting the bit disables the counting direction (forward): 0: Counting direction active (default) 1: Counting direction disabled	BOOLEAN	RO	0x00 (0_{dec})
70n1:05	Control_Disable thresholds ccw [▶ 184]	By setting the bit, the counting direction (backward) is disabled: 0: Counting direction active (default) 1: Counting direction disabled	BOOLEAN	RO	0x00 (0_{dec})

Index 70n2 ENC Control 2 Output Ch.n (for n = 0 [output 1], n = 1 [output 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
70n2:0	ENC Control 2 Output Ch.n	Maximum subindex	UINT8	RO	0x11 (17_{dec})
70n2:11	Number of output events	Enter the number of currently valid threshold values. The value should be ≥ 1 to activate the first threshold.	UINT8	RO	0x00 (0_{dec})

Index 70n3 ENC Output States Ch.n (for n = 0 [output 1], n = 1 [output 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
70n3:0	ENC Output states Ch.n	Maximum subindex	UINT8	RO	0x08 (8_{dec})
70n3:01	Output state_Ouptut state 1 [▶ 184]	Input of the status which the output should assume when "Threshold 1” (0x70n4:11) is reached.	BOOLEAN	RO	0x00 (0_{dec})
70n3:02	Output state_Ouptut state 2 [▶ 184]	Input of the status which the output should assume when "Threshold 2” (0x70n4:12) is reached.	BOOLEAN	RO	0x00 (0_{dec})
70n3:03	Output state_Ouptut state 3 [▶ 184]	Input of the status which the output should assume when "Threshold 3" (0x70n4:13) is reached.	BOOLEAN	RO	0x00 (0_{dec})
70n3:04	Output state_Ouptut state 4 [▶ 184]	Input of the status which the output should assume when "Threshold 4" (0x70n4:14) is reached.	BOOLEAN	RO	0x00 (0_{dec})
70n3:05	Output state_Ouptut state 5 [▶ 184]	Input of the status which the output is to assume when "Threshold 5" is reached.	BOOLEAN	RO	0x00 (0_{dec})
70n3:06	Output state_Ouptut state 6 [▶ 184]	Input of the status which the output is to assume when "Threshold 6" is reached.	BOOLEAN	RO	0x00 (0_{dec})
70n3:07	Output state_Ouptut state 7 [▶ 184]	Input of the status which the output is to assume when "Threshold 7" is reached.	BOOLEAN	RO	0x00 (0_{dec})
70n3:08	Output state_Ouptut state 8 [▶ 184]	Input of the status which the output is to assume when "Threshold 8" is reached.	BOOLEAN	RO	0x00 (0_{dec})

Index 70n4 ENC Threshold Output Ch.n (for n = 0 [output 1], n = 1 [output 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
70n4:0	ENC Threshold Output Ch.n	Maximum subindex	UINT8	RO	0x11 (17_{dec})
70n4:11	Threshold_Threshold 1 [▶ 184]	Threshold value 1 (32-bit)	UINT32	RO	0x00000000 (0_{dec})
70n4:12	Threshold_Threshold 2 [▶ 184]	Threshold value 2 (32-bit)	UINT32	RO	0x00000000 (0_{dec})
70n4:13	Threshold_Threshold 3 [▶ 184]	Threshold value 3 (32-bit)	UINT32	RO	0x00000000 (0_{dec})
70n4:14	Threshold_Threshold 4 [▶ 184]	Threshold value 4 (32-bit)	UINT32	RO	0x00000000 (0_{dec})
70n4:15	Threshold_Threshold 5 [▶ 184]	Threshold value 5 (32-bit)	UINT32	RO	0x00000000 (0_{dec})
70n4:16	Threshold_Threshold 6 [▶ 184]	Threshold value 6 (32-bit)	UINT32	RO	0x00000000 (0_{dec})
70n4:17	Threshold_Threshold 7 [▶ 184]	Threshold value 7 (32-bit)	UINT32	RO	0x00000000 (0_{dec})
70n4:18	Threshold_Threshold 8 [▶ 184]	Threshold value 8 (32-bit)	UINT32	RO	0x00000000 (0_{dec})

Index 70n5 ENC Duration Output Ch.n (for n = 0 [output 1], n = 1 [output 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
70n5:0	ENC Duration Output Ch.n	Maximum subindex	UINT8	RO	0x11 (17_{dec})
70n5:11	Duration_Duration 1 [▶ 184]	Switching duration in µs for which the output is to be switched when the value specified in "Threshold 1" (0x70n4:11) has been reached. 0: Status of the output remains until the next "Threshold" value is reached. Maximum switching duration: 24,000 µs	UINT32	RO	0x00000000 (0_{dec})
70n5:12	Duration_Duration 2 [▶ 184]	Switching duration in µs for which the output is to be switched when the value specified in "Threshold 2" (0x70n4:12) has been reached. 0: Status of the output remains until the next "Threshold" value is reached. Maximum switching duration: 24,000 µs	UINT32	RO	0x00000000 (0_{dec})
70n5:13	Duration_Duration 3 [▶ 184]	Switching duration in µs for which the output is to be switched when the value specified in "Threshold 3" (0x70n4:13) has been reached. 0: Status of the output remains until the next "Threshold" value is reached. Maximum switching duration: 24,000 µs	UINT32	RO	0x00000000 (0_{dec})
70n5:14	Duration_Duration 4 [▶ 184]	Switching duration in µs for which the output is to be switched when the value specified in "Threshold 4" (0x70n4:14) has been reached. 0: Status of the output remains until the next "Threshold" value is reached. Maximum switching duration: 24,000 µs	UINT32	RO	0x00000000 (0_{dec})
70n5:15	Duration_Duration 5 [▶ 184]	Switching duration in µs for which the output is to be switched when the value specified in "Threshold 5" (0x70n4:15) has been reached. 0: Status of the output remains until the next "Threshold" value is reached. Maximum switching duration: 24,000 µs	UINT32	RO	0x00000000 (0_{dec})
70n5:16	Duration_Duration 6 [▶ 184]	Switching duration in µs for which the output is to be switched when the value specified in "Threshold 6" (0x70n4:16) has been reached. 0: Status of the output remains until the next "Threshold" value is reached. Maximum switching duration: 24,000 µs	UINT32	RO	0x00000000 (0_{dec})
70n5:17	Duration_Duration 7 [▶ 184]	Switching duration in µs for which the output is to be switched when the value specified in "Threshold 7" (0x70n4:17) has been reached. 0: Status of the output remains until the next "Threshold" value is reached. Maximum switching duration: 24,000 µs	UINT32	RO	0x00000000 (0_{dec})
70n5:18	Duration_Duration 8 [▶ 184]	Switching duration in µs for which the output is to be switched when the value specified in "Threshold 8" (0x70n4:18) has been reached. 0: Status of the output remains until the next "Threshold" value is reached. Maximum switching duration: 24,000 µs	UINT32	RO	0x00000000 (0_{dec})

10.5 Information / diagnostic data (channel specific)

Index A000 ENC Diag data Ch. 1

Index (hex)	Name	Meaning	Data type	Flags	Default
A000:0	ENC Diag data Ch.n	Maximum subindex	UINT8	RO	0x16 (22_{dec})
A000:01	Error A	An "open circuit" error is present at input A	BOOLEAN	RO	0x00 (0_{dec})
A000:02	Error B	An "open circuit" error is present at input B	BOOLEAN	RO	0x00 (0_{dec})
A000:03	Error C	An "open circuit" error is present at input zero pulse C	BOOLEAN	RO	0x00 (0_{dec})
A000:04	Field power failure	Encoder supply voltage is not present	BOOLEAN	RO	0x00 (0_{dec})
A000:05	Error Input status	Over/undervoltage at the Status Input (alarm input)	BOOLEAN	RO	0x00 (0_{dec})
A000:08	Error output	Error at output 1	BOOLEAN	RO	0x00 (0_{dec})
A000:09	Error output 2	Error at output 2	BOOLEAN	RO	0x00 (0_{dec})
A000:13	Encoder plausibility error counter	Number of detected plausibility errors	UINT16	RO	0x0000 (0_{dec})
A000:14	Filter violation counter extern latch	Number of filter overruns detected at the Latch input	UINT16	RO	0x0000 (0_{dec})
A000:15	Filter violation counter input gate	Number of filter overruns detected at the Gate/Latch input	UINT16	RO	0x0000 (0_{dec})
A000:16	Filter violation counter	Number of detected filter violations at the encoder inputs	UINT16	RO	0x0000 (0_{dec})

10.6 Information / diagnostic data (device specific)

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Maximum subindex	UINT8	RO	0x15 (21_{dec})
10F3:01	Maximum Messages	Maximum number of stored messages A maximum of 16 messages can be stored.	UINT8	RO	0x00 (0_{dec})
10F3:02	Newest Messages	Subindex of the latest message	UINT8	RO	0x00 (0_{dec})
10F3:03	Newest Acknowledged Messages	Subindex of the last confirmed message	UINT8	RW	0x00 (0_{dec})
10F3:04	New Messages Available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0_{dec})
10F3:05	Flags	not used	UINT16	RW	0x0000 (0_{dec})
10F3:06	Diagnosis Message 001	Message 1	OCTET-STRING[20]	RO	{0}
…	…	…	…	…	…
10F3:15	Diagnosis Message 016	Message 16	OCTET-STRING[20]	RO	{0}

Index 10F8 Actual Time Stamp

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8:0	Actual Time Stamp	Current time stamp	UINT64	RO	

10.7 Standard objects

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

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: • The Lo-Word contains the CoE profile used (5001). • The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x01FF1389 (33493897_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL5131

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	09

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	10

Index 100B Bootloader version

Index (hex)	Name	Meaning	Data type	Flags	Default
100B:0	Bootloader version	Bootloader version	STRING	RO	

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x140B3052 (335032402_{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	0x00000000 (0_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave: • Low Word ◦ The Low Byte (bit 0-7) of the Low Word contains the year of production. ◦ The High Byte (bit 8-15) of the Low Word contains the week of production. • The High Word (bit 16-31) is 0.	UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

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

Index ENC 1400 RxPDO-Par Control Ch.1

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

Index 1401 ENC RxPDO-Par Control Compact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1401:0	ENC RxPDO-Par Control Compact Ch.1	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1401:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 2	OCTET-STRING[2]	RO	00 16

Index 1600 ENC RxPDO-Map Control Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	ENC RxPDO-Map Control Ch.1	PDO Mapping RxPDO 1	UINT8	RO	0x0D (13_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x01 (Enable latch C))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7000:02, 1
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x03 (Set Counter))	UINT32	RO	0x7000:03, 1
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7000:04, 1
1600:05	SubIndex 005	5. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1600:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x08 (Set Counter on latch C))	UINT32	RO	0x7000:08, 1
1600:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x09 (Set software gate))	UINT32	RO	0x7000:09, 1
1600:08	SubIndex 008	8. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x0A (Set counter on latch extern on positive edge))	UINT32	RO	0x7000:0A, 1
1600:09	SubIndex 009	9. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x0B (Set counter on latch extern on negative edge))	UINT32	RO	0x7000:0B, 1
1600:0A	SubIndex 010	10. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x0C (Enable latch extern 2 on positive edge))	UINT32	RO	0x7000:0C, 1
1600:0B	SubIndex 011	11. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x0D (Enable latch extern 2 on negative edge))	UINT32	RO	0x7000:0D, 1
1600:0C	SubIndex 012	12. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1600:0D	SubIndex 013	13. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x7000:11, 32

Index 1601 ENC RxPDO-Map Control Compact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	ENC RxPDO-Map Control Compact Ch.1	PDO Mapping RxPDO 2	UINT8	RO	0x0D (13_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x01 (Enable latch C))	UINT32	RO	0x7000:01, 1
1601:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7000:02, 1
1601:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x03 (Set Counter))	UINT32	RO	0x7000:03, 1
1601:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7000:04, 1
1601:05	SubIndex 005	5. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1601:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x08 (Set Counter on latch C))	UINT32	RO	0x7000:08, 1
1601:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x09 (Set software gate))	UINT32	RO	0x7000:09, 1
1601:08	SubIndex 008	8. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x0A (Set counter on latch extern on positive edge))	UINT32	RO	0x7000:0A, 1
1601:09	SubIndex 009	9. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x0B (Set counter on latch extern on negative edge))	UINT32	RO	0x7000:0B, 1
1601:0A	SubIndex 010	10. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x0C (Enable latch extern 2 on positive edge))	UINT32	RO	0x7000:0C, 1
1601:0B	SubIndex 011	11. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x0D (Enable latch extern 2 on negative edge))	UINT32	RO	0x7000:0D, 1
1601:0C	SubIndex 012	12. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1601:0D	SubIndex 013	13. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x7000:11, 16

Index 1602 ENC RxPDO-Map Control Output Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	ENC RxPDO-Map Control Ch.1	PDO Mapping RxPDO 3	UINT8	RO	0x1F (31_{dec})
1602:01	SubIndex 001	1. PDO Mapping entry (object 0x7001 (ENC Control Output Ch.1), entry 0x01 (Set digital output))	UINT32	RO	0x7001:01, 1
1602:02	SubIndex 002	2. PDO Mapping entry (object 0x7001 (ENC Control Output Ch.1), entry 0x02 (Enable thresholds output))	UINT32	RO	0x7001:02, 1
1602:03	SubIndex 003	3. PDO Mapping entry (object 0x7001 (ENC Control Output Ch.1), entry 0x03 (Set tristate))	UINT32	RO	0x7001:03, 1
1602:04	SubIndex 004	4. PDO Mapping entry (object 0x7001 (ENC Control Output Ch.1), entry 0x04 (Disable thresholds cw))	UINT32	RO	0x7001:04, 1
1602:05	SubIndex 005	5. PDO Mapping entry (object 0x7001 (ENC Control Output Ch.1), entry 0x05 (Disable thresholds ccw))	UINT32	RO	0x7001:05, 1
1602:06	SubIndex 006	6. PDO Mapping entry (11 bits align)	UINT32	RO	0x0000:00, 11
1602:07	SubIndex 007	7. PDO Mapping entry (object 0x7002 (ENC Control 2 Output Ch.1), entry 0x11 (Number of output events))	UINT32	RO	0x7002:11, 8
1602:08	SubIndex 008	8. PDO Mapping entry (object 0x7003 (ENC Output States Ch.1), entry 0x01 (Output state_Output state 1))	UINT32	RO	0x7003:01, 1
1602:09	SubIndex 009	9. PDO Mapping entry (object 0x7003 (ENC Output States Ch.1), entry 0x02 (Output state_Output state 2))	UINT32	RO	0x7003:02, 1
1602:0A	SubIndex 010	10. PDO Mapping entry (object 0x7003 (ENC Output States Ch.1), entry 0x03 (Output state_Output state 3))	UINT32	RO	0x7003:03, 1
1602:0B	SubIndex 011	11. PDO Mapping entry (object 0x7003 (ENC Output States Ch.1), entry 0x04 (Output state_Output state 4))	UINT32	RO	0x7003:04, 1
1602:0C	SubIndex 012	12. PDO Mapping entry (object 0x7003 (ENC Output States Ch.1), entry 0x05 (Output state_Output state 5))	UINT32	RO	0x7003:05, 1
1602:0D	SubIndex 013	13. PDO Mapping entry (object 0x7003 (ENC Output States Ch.1), entry 0x06 (Output state_Output state 6))	UINT32	RO	0x7003:06, 1
1602:0E	SubIndex 014	14. PDO Mapping entry (object 0x7003 (ENC Output States Ch.1), entry 0x07 (Output state_Output state 7))	UINT32	RO	0x7003:07, 1
1602:0F	SubIndex 015	15. PDO Mapping entry (object 0x7003 (ENC Output States Ch.1), entry 0x08 (Output state_Output state 8))	UINT32	RO	0x7003:08, 1
1602:10	SubIndex 016	16. PDO Mapping entry (object 0x7004 (ENC Threshold Output Ch.1), entry 0x11 (Threshold_Threshold 1))	UINT32	RO	0x7004:11, 32
1602:11	SubIndex 017	17. PDO Mapping entry (object 0x7004 (ENC Threshold Output Ch.1), entry 0x12 (Threshold_Threshold 2))	UINT32	RO	0x7004:12, 32
1602:12	SubIndex 018	18. PDO Mapping entry (object 0x7004 (ENC Threshold Output Ch.1), entry 0x13 (Threshold_Threshold 3))	UINT32	RO	0x7004:13, 32
1602:13	SubIndex 019	19. PDO Mapping entry (object 0x7004 (ENC Threshold Output Ch.1), entry 0x14 (Threshold_Threshold 4))	UINT32	RO	0x7004:14, 32
1602:14	SubIndex 020	20. PDO Mapping entry (object 0x7004 (ENC Threshold Output Ch.1), entry 0x15 (Threshold_Threshold 5))	UINT32	RO	0x7004:15, 32
1602:15	SubIndex 021	21. PDO Mapping entry (object 0x7004 (ENC Threshold Output Ch.1), entry 0x16 (Threshold_Threshold 6))	UINT32	RO	0x7004:16, 32
1602:16	SubIndex 022	22. PDO Mapping entry (object 0x7004 (ENC Threshold Output Ch.1), entry 0x17 (Threshold_Threshold 7))	UINT32	RO	0x7004:17, 32
1602:17	SubIndex 023	23. PDO Mapping entry (object 0x7004 (ENC Threshold Output Ch.1), entry 0x18 (Threshold_Threshold 8))	UINT32	RO	0x7004:18, 32
1602:18	SubIndex 024	24. PDO Mapping entry (object 0x7005 (ENC Duration Output Ch.1), entry 0x11 (Duration_Duration 1))	UINT32	RO	0x7005:11, 32
1602:19	SubIndex 025	25. PDO Mapping entry (object 0x7005 (ENC Duration Output Ch.1), entry 0x12 (Duration_Duration 2))	UINT32	RO	0x7005:12, 32
1602:1A	SubIndex 026	26. PDO Mapping entry (object 0x7005 (ENC Duration Output Ch.1), entry 0x13 (Duration_Duration 3))	UINT32	RO	0x7005:13, 32
1602:1B	SubIndex 027	27. PDO Mapping entry (object 0x7005 (ENC Duration Output Ch.1), entry 0x14 (Duration_Duration 4))	UINT32	RO	0x7005:14, 32
1602:1C	SubIndex 028	28. PDO Mapping entry (object 0x7005 (ENC Duration Output Ch.1), entry 0x15 (Duration_Duration 5))	UINT32	RO	0x7005:15, 32
1602:1D	SubIndex 029	29. PDO Mapping entry (object 0x7005 (ENC Duration Output Ch.1), entry 0x16 (Duration_Duration 6))	UINT32	RO	0x7005:16, 32
1602:1E	SubIndex 030	30. PDO Mapping entry (object 0x7005 (ENC Duration Output Ch.1), entry 0x17 (Duration_Duration 7))	UINT32	RO	0x7005:17, 32
1602:1F	SubIndex 031	31. PDO Mapping entry (object 0x7005 (ENC Duration Output Ch.1), entry 0x18 (Duration_Duration 8))	UINT32	RO	0x7005:18, 32

Index 1605 ENC RxPDO-Map Control Output Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	ENC RxPDO-Map Control Output Ch.2	PDO Mapping RxPDO 6	UINT8	RO	0x1F (31_{dec})
1605:01	SubIndex 001	1. PDO Mapping entry (object 0x7011 (ENC Control Output Ch.2), entry 0x01 (Set digital output))	UINT32	RO	0x7011:01, 1
1605:02	SubIndex 002	2. PDO Mapping entry (object 0x7011 (ENC Control Output Ch.2), entry 0x02 (Enable thresholds output))	UINT32	RO	0x7011:02, 1
1605:03	SubIndex 003	3. PDO Mapping entry (object 0x7011 (ENC Control Output Ch.2), entry 0x03 (Set tristate))	UINT32	RO	0x7011:03, 1
1605:04	SubIndex 004	4. PDO Mapping entry (object 0x7011 (ENC Control Output Ch.2), entry 0x04 (Disable thresholds cw))	UINT32	RO	0x7011:04, 1
1605:05	SubIndex 005	5. PDO Mapping entry (object 0x7011 (ENC Control Output Ch.2), entry 0x05 (Disable thresholds ccw))	UINT32	RO	0x7011:05, 1
1605:06	SubIndex 006	6. PDO Mapping entry (11 bits align)	UINT32	RO	0x0000:00, 11
1605:07	SubIndex 007	7. PDO Mapping entry (object 0x7012 (ENC Control 2 Output Ch.2), entry 0x11 (Number of output events))	UINT32	RO	0x7012:11, 8
1605:08	SubIndex 008	8. PDO Mapping entry (object 0x7013 (ENC Output States Ch.2), entry 0x02 (Output state_Output state 2))	UINT32	RO	0x7013:01, 1
1605:09	SubIndex 009	9. PDO Mapping entry (object 0x7013 (ENC Output States Ch.2), entry 0x02 (Output state_Output state 2))	UINT32	RO	0x7013:02, 1
1605:0A	SubIndex 010	10. PDO Mapping entry (object 0x7013 (ENC Output States Ch.2), entry 0x03 (Output state_Output state 3))	UINT32	RO	0x7013:03, 1
1605:0B	SubIndex 011	11. PDO Mapping entry (object 0x7013 (ENC Output States Ch.2), entry 0x04 (Output state_Output state 4))	UINT32	RO	0x7013:04, 1
1605:0C	SubIndex 012	12. PDO Mapping entry (object 0x7013 (ENC Output States Ch.2), entry 0x05 (Output state_Output state 5))	UINT32	RO	0x7013:05, 1
1605:0D	SubIndex 013	13. PDO Mapping entry (object 0x7013 (ENC Output States Ch.2), entry 0x06 (Output state_Output state 6))	UINT32	RO	0x7013:06, 1
1605:0E	SubIndex 014	14. PDO Mapping entry (object 0x7013 (ENC Output States Ch.2), entry 0x07 (Output state_Output state 7))	UINT32	RO	0x7013:07, 1
1605:0F	SubIndex 015	15. PDO Mapping entry (object 0x7013 (ENC Output States Ch.2), entry 0x08 (Output state_Output state 8))	UINT32	RO	0x7013:08, 1
1605:10	SubIndex 016	16. PDO Mapping entry (object 0x7014 (ENC Threshold Output Ch.1), entry 0x11 (Threshold_Threshold 1))	UINT32	RO	0x7014:11, 32
1605:11	SubIndex 017	17. PDO Mapping entry (object 0x7014 (ENC Threshold Output Ch.2), entry 0x12 (Threshold_Threshold 2))	UINT32	RO	0x7014:12, 32
1605:12	SubIndex 018	18. PDO Mapping entry (object 0x7014 (ENC Threshold Output Ch.2), entry 0x13 (Threshold_Threshold 3))	UINT32	RO	0x7014:13, 32
1605:13	SubIndex 019	19. PDO Mapping entry (object 0x7014 (ENC Threshold Output Ch.2), entry 0x14 (Threshold_Threshold 4))	UINT32	RO	0x7014:14, 32
1605:14	SubIndex 020	20. PDO Mapping entry (object 0x7014 (ENC Threshold Output Ch.2), entry 0x15 (Threshold_Threshold 5))	UINT32	RO	0x7014:15, 32
1605:15	SubIndex 021	21. PDO Mapping entry (object 0x7014 (ENC Threshold Output Ch.2), entry 0x16 (Threshold_Threshold 6))	UINT32	RO	0x7014:16, 32
1605:16	SubIndex 022	22. PDO Mapping entry (object 0x7014 (ENC Threshold Output Ch.2), entry 0x17 (Threshold_Threshold 7))	UINT32	RO	0x7014:17, 32
1605:17	SubIndex 023	23. PDO Mapping entry (object 0x7014 (ENC Threshold Output Ch.1), entry 0x18 (Threshold_Threshold 8))	UINT32	RO	0x7014:18, 32
1605:18	SubIndex 024	24. PDO Mapping entry (object 0x7015 (ENC Duration Output Ch.2), entry 0x11 (Duration_Duration 1))	UINT32	RO	0x7015:11, 32
1605:19	SubIndex 025	25. PDO Mapping entry (object 0x7015 (ENC Duration Output Ch.2), entry 0x12 (Duration_Duration 2))	UINT32	RO	0x7015:12, 32
1605:1A	SubIndex 026	26. PDO Mapping entry (object 0x7015 (ENC Duration Output Ch.2), entry 0x13 (Duration_Duration 3))	UINT32	RO	0x7015:13, 32
1605:1B	SubIndex 027	27. PDO Mapping entry (object 0x7015 (ENC Duration Output Ch.2), entry 0x14 (Duration_Duration 4))	UINT32	RO	0x7015:14, 32
1605:1C	SubIndex 028	28. PDO Mapping entry (object 0x7015 (ENC Duration Output Ch.2), entry 0x15 (Duration_Duration 5))	UINT32	RO	0x7015:15, 32
1605:1D	SubIndex 029	29. PDO Mapping entry (object 0x7015 (ENC Duration Output Ch.2), entry 0x16 (Duration_Duration 6))	UINT32	RO	0x7015:16, 32
1605:1E	SubIndex 030	30. PDO Mapping entry (object 0x7015 (ENC Duration Output Ch.2), entry 0x17 (Duration_Duration 7))	UINT32	RO	0x7015:17, 32
1605:1F	SubIndex 031	31. PDO Mapping entry (object 0x7015 (ENC Duration Output Ch.2), entry 0x18 (Duration_Duration 8))	UINT32	RO	0x7015:18, 32

Index 1800 ENC TxPDO-Par Status Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1800:0	ENC TxPDO-Par Status Ch.1	PDO parameter TxPDO 1	UINT8	RO	0x06 (6_{dec})
1800:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1	OCTET-STRING[14]	RO	01 1A

Index 1801 ENC TxPDO-Par Status Compact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1801:0	ENC TxPDO-Par Status compact Ch.1	PDO parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1801:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2	OCTET-STRING[14]	RO	00 1A

Index 1803 ENC TxPDO-Frequency Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1803:0	ENC TxPDO-Par Frequency Ch.1	PDO parameter TxPDO 4	UINT8	RO	0x06 (6_{dec})
1803:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 4	OCTET-STRING[14]	RO	04 1A

Index 1804 ENC TxPDO-Par Frequency Compact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1804:0	ENC TxPDO-Par Frequency Compact Ch.1	PDO parameter TxPDO 5	UINT8	RO	0x06 (6_{dec})
1804:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 5	OCTET-STRING[14]	RO	03 1A

Index 1805 ENC TxPDO-Par Period Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1805:0	ENC TxPDO-Par Period Ch.1	PDO parameter TxPDO 6	UINT8	RO	0x06 (6_{dec})
1805:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 6	OCTET-STRING[14]	RO	06 1A

Index 1806 ENC TxPDO-Par Period Compact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1806:0	ENC TxPDO-Par Period Compact Ch.1	PDO parameter TxPDO 7	UINT8	RO	0x06 (6_{dec})
1806:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 7	OCTET-STRING[14]	RO	05 1A

Index 1808 ENC TxPDO-Par Timestamp Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1808:0	ENC TxPDO-Par Timestamp Ch.1	PDO parameter TxPDO 9	UINT8	RO	0x06 (6_{dec})
1808:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 9	OCTET-STRING[14]	RO	09 1A

Index 1809 ENC TxPDO-Par Timestamp Compact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1809:0	ENC TxPDO-Par Timestamp Compact Ch.1	PDO parameter TxPDO 10	UINT8	RO	0x06 (6_{dec})
1809:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 10	OCTET-STRING[14]	RO	08 1A

Index 1A00 ENC TxPDO-Map Status Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	ENC TxPDO-Map Status Ch.1	PDO Mapping TxPDO 1	UINT8	RO	0x18 (24_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x01 (Latch C valid))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x02 (Latch extern valid))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x03 (Set counter done))	UINT32	RO	0x6000:03, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x04 (Counter underflow))	UINT32	RO	0x6000:04, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x05 (Counter overflow))	UINT32	RO	0x6000:05, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x06 (Status of input status))	UINT32	RO	0x6000:06, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x07 (Open circuit))	UINT32	RO	0x6000:07, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A00:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0B (Status of input C))	UINT32	RO	0x6000:0B, 1
1A00:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0C (Status of input gate))	UINT32	RO	0x6000:0C, 1
1A00:0D	SubIndex 013	13. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0D (Diag))	UINT32	RO	0x6002:0D, 1
1A00:0E	SubIndex 014	14. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0E (TxPDO State))	UINT32	RO	0x6002:0E, 1
1A00:0F	SubIndex 015	15. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0F (Input cycle counter))	UINT32	RO	0x6002:0F, 2
1A00:10	SubIndex 016	16. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x11 (Software gate valid))	UINT32	RO	0x6002:11, 1
1A00:11	SubIndex 017	17. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x12 (Latch extern 2 valid))	UINT32	RO	0x6002:12, 1
1A00:12	SubIndex 018	18. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x13 (Direction inversion detected))	UINT32	RO	0x6002:13, 1
1A00:13	SubIndex 019	19. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x14 (Status of extern latch))	UINT32	RO	0x6002:14, 1
1A00:14	SubIndex 020	20. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x15 (Counter value out of range))	UINT32	RO	0x6002:15, 1
1A00:15	SubIndex 021	21. PDO Mapping entry (11 bits align)	UINT32	RO	0x0000:00, 11
1A00:16	SubIndex 022	22. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:11, 32
1A00:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:12, 32
1A00:18	SubIndex 024	24. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x22 (Latch value 2))	UINT32	RO	0x6000:22, 32

Index 1A01 ENC TxPDO-Map Status Compact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	ENC TxPDO-Map Status Ch.1	PDO Mapping TxPDO 2	UINT8	RO	0x18 (24_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x01 (Latch C valid))	UINT32	RO	0x6000:01, 1
1A01:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x02 (Latch extern valid))	UINT32	RO	0x6000:02, 1
1A01:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x03 (Set counter done))	UINT32	RO	0x6000:03, 1
1A01:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x04 (Counter underflow))	UINT32	RO	0x6000:04, 1
1A01:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x05 (Counter overflow))	UINT32	RO	0x6000:05, 1
1A01:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x06 (Status of input status))	UINT32	RO	0x6000:06, 1
1A01:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x07 (Open circuit))	UINT32	RO	0x6000:07, 1
1A01:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A01:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A01:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A01:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0B (Status of input C))	UINT32	RO	0x6000:0B, 1
1A01:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0C (Status of input gate))	UINT32	RO	0x6000:0C, 1
1A01:0D	SubIndex 013	13. PDO Mapping entry (object 0x6002(ENC Inputs status Ch.1) entry 0x0D (Diag))	UINT32	RO	0x6002:0D, 1
1A01:0E	SubIndex 014	14. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0E (TxPDO State))	UINT32	RO	0x6002:0E, 1
1A01:0F	SubIndex 015	15. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0F (Input cycle counter))	UINT32	RO	0x6002:0F, 2
1A01:10	SubIndex 016	16. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x11 (Software gate valid))	UINT32	RO	0x6002:11, 1
1A01:11	SubIndex 017	17. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x12 (Latch extern 2 valid))	UINT32	RO	0x6002:12, 1
1A01:12	SubIndex 018	18. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x13 (Direction inversion detected))	UINT32	RO	0x6002:13, 1
1A01:13	SubIndex 019	19. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x14 (Status of extern latch))	UINT32	RO	0x6002:14, 1
1A01:14	SubIndex 020	20. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x15 (Counter value out of range))	UINT32	RO	0x6002:15, 1
1A01:15	SubIndex 021	21. PDO Mapping entry (11 bits align)	UINT32	RO	0x0000:00, 11
1A01:16	SubIndex 022	22. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:11, 16
1A01:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:12, 16
1A01:18	SubIndex 024	24. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x22 (Latch value 2))	UINT32	RO	0x6000:22, 16

Index 1A02 ENC TxPDO-Map Status Output Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	ENC TxPDO-Map Status Output Ch.1	PDO Mapping TxPDO 3	UINT8	RO	0x03 (3_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x6003 (ENC Inputs Ch.1), entry 0x01 (Status of digital output))	UINT32	RO	0x6003:01, 1
1A02:02	SubIndex 002	2. PDO Mapping entry (object 0x6003 (ENC Inputs Ch.1), entry 0x02 (Status of tristate))	UINT32	RO	0x6003:02, 1
1A02:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14

Index 1A03 ENC TxPDO-Map Frequency Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A03:0	ENC TxPDO-Map Frequency Ch.1	PDO Mapping TxPDO 4	UINT8	RO	0x01 (1_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x13 (Frequency value))	UINT32	RO	0x6000:13, 32

Index 1A04 ENC TxPDO-Map Frequency Comact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	ENC TxPDO-Map Frequency Compact Ch.1	PDO Mapping TxPDO 5	UINT8	RO	0x01 (1_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x13 (Frequency value))	UINT32	RO	0x6000:13, 16

Index 1A05 ENC TxPDO-Map Period Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	ENC TxPDO-Map Period Ch.1	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x14 (Period value))	UINT32	RO	0x6000:14, 32

Index 1A06 ENC TxPDO-Map Period Compact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A06:0	ENC TxPDO-Map Period Compact Ch.1	PDO Mapping TxPDO 7	UINT8	RO	0x01 (1_{dec})
1A06:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x14 (Period value))	UINT32	RO	0x6000:14, 16

Index 1A07 ENC TxPDO-Map Duty Cycle Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A07:0	ENC TxPDO-Map Status Compact Legacy Ch.1	PDO Mapping TxPDO 8	UINT8	RO	0x03 (3_{dec})
1A07:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x23 (Duty cycle))	UINT32	RO	0x6000:23, 16
1A07:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x24 (Duty cycle min))	UINT32	RO	0x6000:24, 16
1A07:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x25 (Duty cycle max))	UINT32	RO	0x6000:25, 16

Index 1A08 ENC TxPDO-Map Timestamp Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	ENC TxPDO-Map Timestamp Ch.1	PDO Mapping TxPDO 9	UINT8	RO	0x04 (4_{dec})
1A08:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x16 (Timestamp))	UINT32	RO	0x6000:16, 64
1A08:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x1F (Timestamp C))	UINT32	RO	0x6000:1F, 64
1A08:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x20 (Timestamp latch))	UINT32	RO	0x6000:20, 64
1A08:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x21 (Timestamp latch2))	UINT32	RO	0x6000:21, 64

Index 1A09 ENC TxPDO-Map Timestamp Compact Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A09:0	ENC TxPDO-Map Timestamp Compact Ch.1	PDO Mapping TxPDO 10	UINT8	RO	0x04 (4_{dec})
1A09:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x16 (Timestamp))	UINT32	RO	0x6000:16, 32
1A09:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x1F (Timestamp C))	UINT32	RO	0x6000:1F, 32
1A09:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x20 (Timestamp latch))	UINT32	RO	0x6000:20, 32
1A09:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x21 (Timestamp latch2))	UINT32	RO	0x6000:21, 32

Index 1A0A ENC TxPDO-Map Timestamps Output Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0A:0	ENC TxPDO-Map Timestamps Output Ch.1	PDO Mapping TxPDO 11	UINT8	RO	0x08 (8_{dec})
1A0A:01	SubIndex 001	1. PDO Mapping entry (object 0x6004 (ENC Timestamps Output Ch.1), entry 0x11 (Timestamp 1))	UINT32	RO	0x6004:11, 64
1A0A:02	SubIndex 002	2. PDO Mapping entry (object 0x6004 (ENC Timestamps Output Ch.1), entry 0x12 (Timestamp 2))	UINT32	RO	0x6004:12, 64
1A0A:03	SubIndex 003	3. PDO Mapping entry (object 0x6004 (ENC Timestamps Output Ch.1), entry 0x13 (Timestamp 3))	UINT32	RO	0x6004:13, 64
1A0A:04	SubIndex 004	4. PDO Mapping entry (object 0x6004 (ENC Timestamps Output Ch.1), entry 0x14 (Timestamp 4))	UINT32	RO	0x6004:14, 64
1A0A:05	SubIndex 005	1. PDO Mapping entry (object 0x6004 (ENC Timestamps Output Ch.1), entry 0x15 (Timestamp 5))	UINT32	RO	0x6004:15, 64
1A0A:06	SubIndex 006	1. PDO Mapping entry (object 0x6004 (ENC Timestamps Output Ch.1), entry 0x16 (Timestamp 6))	UINT32	RO	0x6004:16, 64
1A0A:07	SubIndex 007	1. PDO Mapping entry (object 0x6004 (ENC Timestamps Output Ch.1), entry 0x17 (Timestamp 7))	UINT32	RO	0x6004:17, 64
1A0A:08	SubIndex 008	1. PDO Mapping entry (object 0x6004 (ENC Timestamps Output Ch.1), entry 0x18 (Timestamp 8))	UINT32	RO	0x6004:18, 64

Index 1A0D ENC TxPDO-Map Status Output Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0D:0	ENC TxPDO-Map Status Output Ch.1	PDO Mapping TxPDO 14	UINT8	RO	0x03 (3_{dec})
1A0D:01	SubIndex 001	1. PDO Mapping entry (object 0x6013 (ENC Inputs Ch.2), entry 0x01 (Status of digital output))	UINT32	RO	0x6013:01, 1
1A0D:02	SubIndex 002	2. PDO Mapping entry (object 0x6013 (ENC Inputs Ch.2), entry 0x02 (Status of tristate))	UINT32	RO	0x6013:02, 1
1A0D:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14

Index 1A15 ENC TxPDO-Map Timestamps Output Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A15:0	ENC TxPDO-Map Timestamps Output Ch.2	PDO Mapping TxPDO 15	UINT8	RO	0x08 (8_{dec})
1A15:01	SubIndex 001	1. PDO Mapping entry (object 0x6014 (ENC Timestamps Output Ch.2), entry 0x11 (Timestamp 1))	UINT32	RO	0x6014:11, 64
1A15:02	SubIndex 002	2. PDO Mapping entry (object 0x6014 (ENC Timestamps Output Ch.2), entry 0x12 (Timestamp 2))	UINT32	RO	0x6014:12, 64
1A15:03	SubIndex 003	3. PDO Mapping entry (object 0x6014 (ENC Timestamps Output Ch.2), entry 0x13 (Timestamp 3))	UINT32	RO	0x6014:13, 64
1A15:04	SubIndex 004	4. PDO Mapping entry (object 0x6014 (ENC Timestamps Output Ch.2), entry 0x14 (Timestamp 4))	UINT32	RO	0x6014:14, 64
1A15:05	SubIndex 005	5. PDO Mapping entry (object 0x6014 (ENC Timestamps Output Ch.2), entry 0x15 (Timestamp 5))	UINT32	RO	0x6014:15, 64
1A15:06	SubIndex 006	6. PDO Mapping entry (object 0x6014 (ENC Timestamps Output Ch.2), entry 0x16 (Timestamp 6))	UINT32	RO	0x6014:16, 64
1A15:07	SubIndex 007	7. PDO Mapping entry (object 0x6014 (ENC Timestamps Output Ch.2), entry 0x17 (Timestamp 7))	UINT32	RO	0x6014:17, 64
1A15:08	SubIndex 008	8. PDO Mapping entry (object 0x6014 (ENC Timestamps Output Ch.2), entry 0x18 (Timestamp 8))	UINT32	RO	0x6014:18, 64

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x03 (3_{dec})
1C12:01	SubIndex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:02	SubIndex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1602 (5634_{dec})
1C12:03	SubIndex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1605 (5637_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x03 (3_{dec})
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656_{dec})
1C13:02	SubIndex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A02 (6658_{dec})
1C13:03	SubIndex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A0D (6669_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: cycle time of the local timer • Synchronous with SM 2 Event: cycle time of the master • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: Free Run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC-Mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 event (DC mode only) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)	UINT16	RO	0x0807 (2055_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000101D0 (66000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time	Min. time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (000000_{dec})
1C32:08	Command	With this entry the real required process data provision time can be measured. • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09, 0x1C33:03 [▶ 242], 0x1C33:06 [▶ 241], 0x1C33:09 [▶ 242] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	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 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	Cycle time (in ns): • Free Run: cycle time of the local timer • Synchronous with SM 2 Event: cycle time of the master DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: Free Run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC-Mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▶ 241] or 0x1C33:08)	UINT16	RO	0x0807 (2055_{dec})
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000101D0 (66000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, DC mode only)	UINT32	RO	0x000101D0 (66000_{dec})
1C33:07	Minimum delay time	Min. time between SYNC1 event and the reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (000000_{dec})
1C33:08	Command	With this entry the real required process data provision time can be measured. • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09 [▶ 241], 0x1C33:03, 0x1C33:06, 0x1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C33:0C	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})
1C33:0D	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})
1C33:20	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 F000 Modular device profile

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

Index F008 Code word

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

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Maximum subindex	UINT8	RW	0x04 (4_{dec})
F010:01	SubIndex 001	reserved	UINT32	RW	0x000001FF (511_{dec})
F010:02	SubIndex 002	reserved	UINT32	RW	0x000001FF (511_{dec})
F010:03	SubIndex 003	reserved	UINT32	RW	0x00000118 (280_{dec})
F010:04	SubIndex 042	reserved	UINT32	RW	0x00000118 (280_{dec})

11 Appendix

11.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

11.2 Firmware compatibility

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

Note

- It is recommended to use the newest possible firmware for the respective hardware
- Beckhoff is not under any obligation to provide customers with free firmware updates for delivered products.

NOTICE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 244].
If a device is placed in BOOTSTRAP mode for a firmware update, it does not check when downloading whether the new firmware is suitable.
This can result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

EL5131			
Hardware (HW)	**Firmware**	**Revision no.**	**Release date**
04*	01	EL5131-0000-0016	2023/02
	02		2023/03
	03*		2025/02

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

11.3 Firmware Update EL/ES/EM/ELM/EP/EPP/ERPxxxx

This section describes the device update for Beckhoff EtherCAT slaves from the EL/ES, ELM, EM, EK, EP, EPP and ERP series. A firmware update should only be carried out after consultation with Beckhoff support.

NOTICE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website.

To update the firmware, TwinCAT can be operated in the so-called FreeRun mode, a paid license is not required.

The device to be updated can usually remain in the installation location, but TwinCAT has to be operated in the FreeRun. Please make sure that EtherCAT communication is trouble-free (no LostFrames etc.).

Other EtherCAT master software, such as the EtherCAT Configurator, should not be used, as they may not support the complexities of updating firmware, EEPROM and other device components.

Storage locations

An EtherCAT slave stores operating data in up to three locations:

- Each EtherCAT slave has a device description, consisting of identity (name, product code), timing specifications, communication settings, etc.
  This device description (ESI; EtherCAT Slave Information) can be downloaded from the Beckhoff website in the download area as a zip file and used in EtherCAT masters for offline configuration, e.g. in TwinCAT.
  Above all, each EtherCAT slave carries its device description (ESI) electronically readable in a local memory chip, the so-called **ESI EEPROM**. When the slave is switched on, this description is loaded locally in the slave and informs it of its communication configuration; on the other hand, the EtherCAT master can identify the slave in this way and, among other things, set up the EtherCAT communication accordingly.

NOTICE

Application-specific writing of the ESI-EEPROM

The ESI is developed by the device manufacturer according to ETG standard and released for the corresponding product.
- Meaning for the ESI file: Modification on the application side (i.e. by the user) is not permitted.
- Meaning for the ESI EEPROM: Even if a writeability is technically given, the ESI parts in the EEPROM and possibly still existing free memory areas must not be changed beyond the normal update process. Especially for cyclic memory processes (operating hours counter etc.), dedicated memory products such as EL6080 or IPC's own NOVRAM must be used.

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in *.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.

Customers can access the data via the EtherCAT fieldbus and its communication mechanisms. Acyclic mailbox communication or register access to the ESC is used for updating or reading of these data.

The TwinCAT System Manager offers mechanisms for programming all three parts with new data, if the slave is set up for this purpose. Generally the slave does not check whether the new data are suitable, i.e. it may no longer be able to operate if the data are unsuitable.

Simplified update by bundle firmware

The update using so-called **bundle firmware** is more convenient: in this case the controller firmware and the ESI description are combined in a *.efw file; during the update both the firmware and the ESI are changed in the terminal. For this to happen it is necessary

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxxx-xxxx_REV0016_SW01.efw
- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

- ESI/Revision: e.g. by means of an online scan in TwinCAT ConfigMode/FreeRun – this is a convenient way to determine the revision
- Firmware: e.g. by looking in the online CoE of the device

NOTICE

Risk of damage to the device!

✓ Note the following when downloading new device files

a) Firmware downloads to an EtherCAT device must not be interrupted

b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.

c) The power supply must adequately dimensioned. The signal level must meet the specification.

⇨ In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

11.3.1 Device description ESI file/XML

NOTICE

Attention regarding update of the ESI description/EEPROM

Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.

The ESI device description is stored locally on the slave and loaded on start-up. Each device description has a unique identifier consisting of slave name (9 characters/digits) and a revision number (4 digits). Each slave configured in the System Manager shows its identifier in the EtherCAT tab:

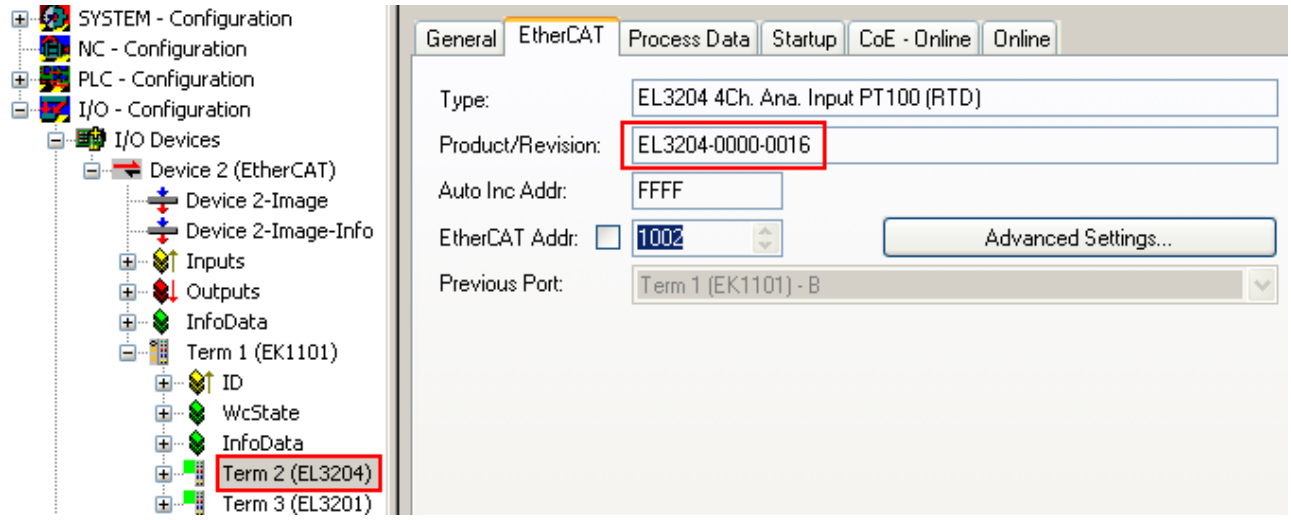


Fig. 182: Device identifier consisting of name EL3204-0000 and revision -0016

The configured identifier must be compatible with the actual device description used as hardware, i.e. the description which the slave has loaded on start-up (in this case EL3204). Normally the configured revision must be the same or lower than that actually present in the terminal network.

For further information on this, please refer to the EtherCAT system documentation.

Update of XML/ESI description

The device revision is closely linked to the firmware and hardware used. Incompatible combinations lead to malfunctions or even final shutdown of the device. Corresponding updates should only be carried out in consultation with Beckhoff support.

Display of ESI slave identifier

The simplest way to ascertain compliance of configured and actual device description is to scan the EtherCAT boxes in TwinCAT mode Config/FreeRun:

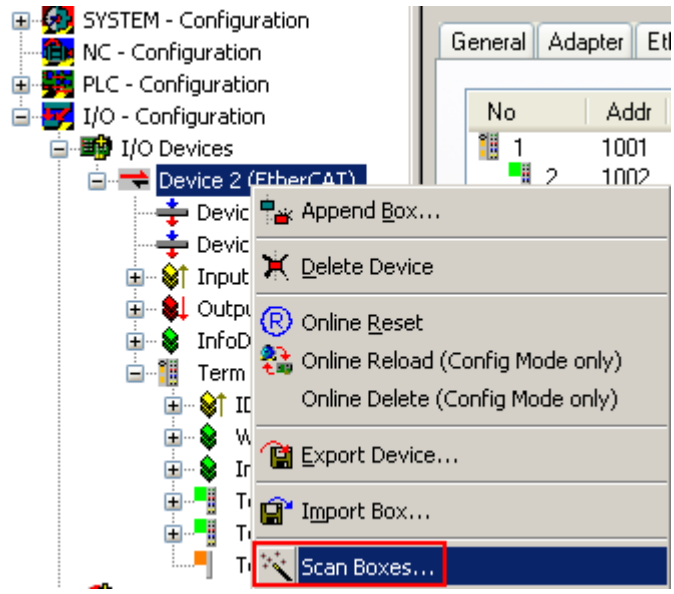


Fig. 183: Scan the subordinate field by right-clicking on the EtherCAT device

If the found field matches the configured field, the display shows



Fig. 184: Configuration is identical

otherwise a change dialog appears for entering the actual data in the configuration.

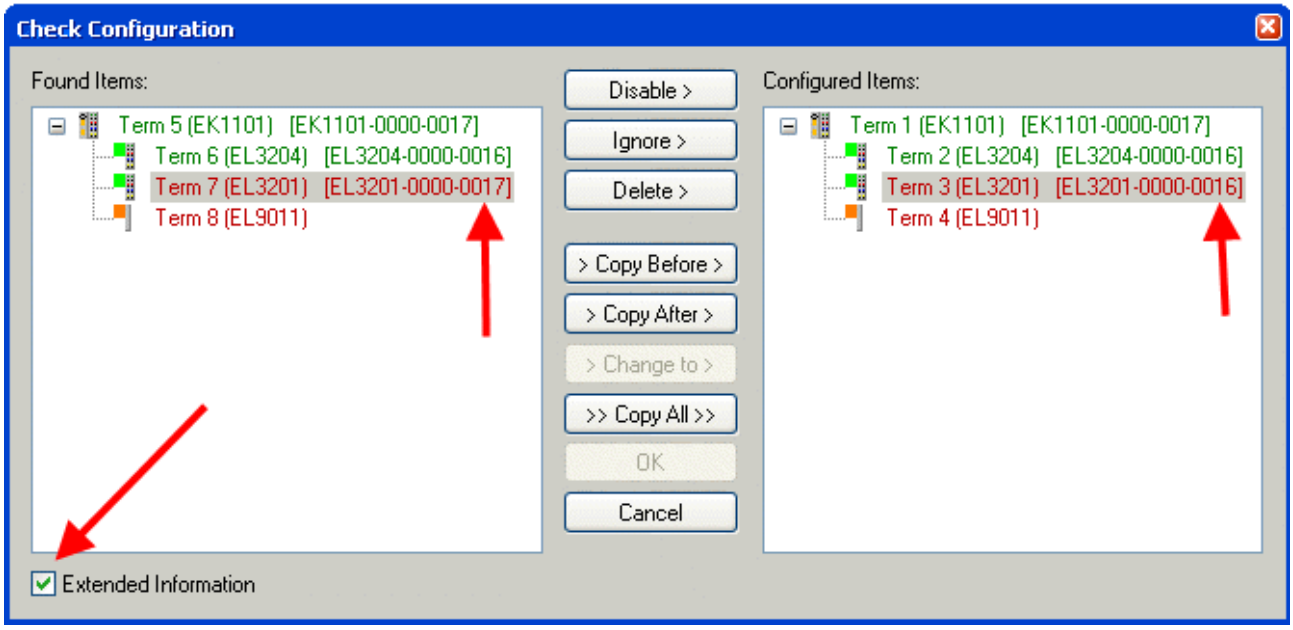


Fig. 185: Change dialog

In this example in Fig. *Change dialog*, an EL3201-0000-**0017** was found, while an EL3201-0000-**0016** was configured. In this case the configuration can be adapted with the *Copy Before* button. The *Extended Information* checkbox must be set in order to display the revision.

Changing the ESI slave identifier

The ESI/EEPROM identifier can be updated as follows under TwinCAT:

- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*

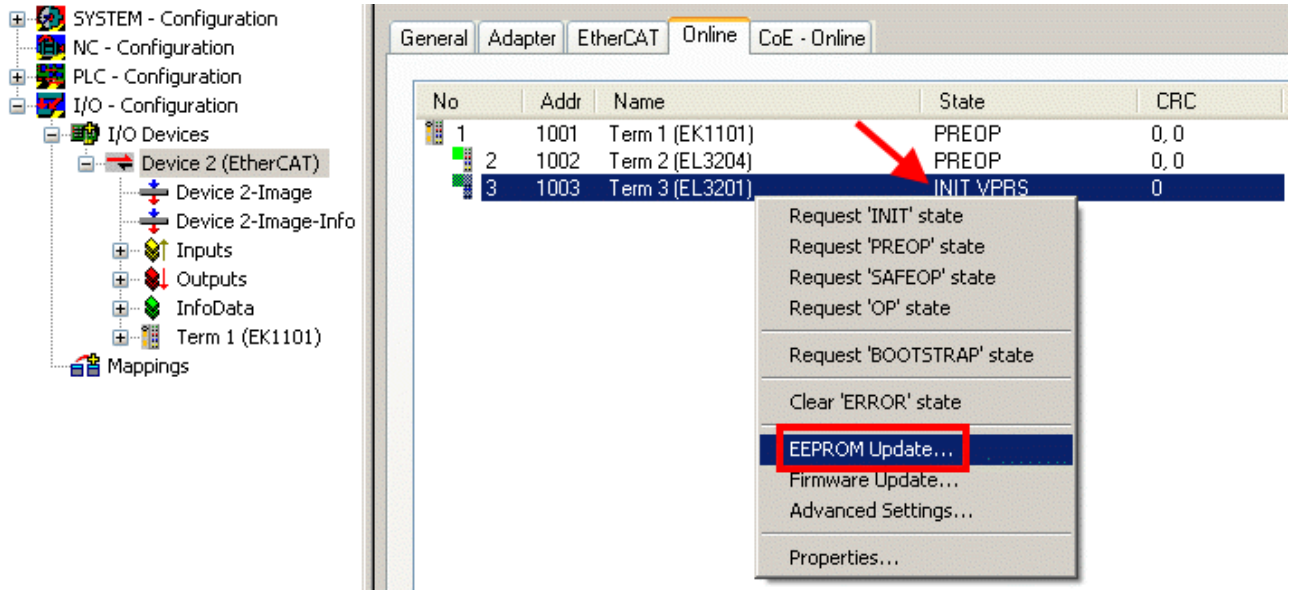


Fig. 186: EEPROM Update

The new ESI description is selected in the following dialog, see Fig. *Selecting the new ESI*. The checkbox *Show Hidden Devices* also displays older, normally hidden versions of a slave.

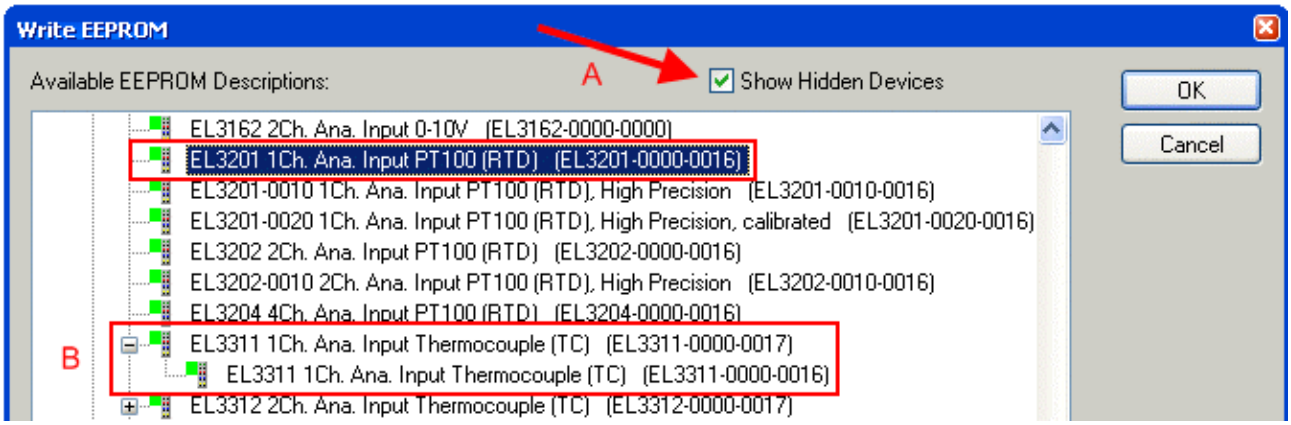


Fig. 187: Selecting the new ESI

A progress bar in the System Manager shows the progress. Data are first written, then verified.

The change only takes effect after a restart.

Most EtherCAT devices read a modified ESI description immediately or after startup from the INIT. Some communication settings such as distributed clocks are only read during power-on. The EtherCAT slave therefore has to be switched off briefly in order for the change to take effect.

11.3.2 Firmware explanation

Determining the firmware version

Determining the version via the TwinCAT System Manager

The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

CoE Online and Offline CoE

Two CoE directories are available:
• **online**: This is offered in the EtherCAT slave by the controller, if the EtherCAT slave supports this. This CoE directory can only be displayed if a slave is connected and operational.
• **offline**: The EtherCAT Slave Information ESI/XML may contain the default content of the CoE. This CoE directory can only be displayed if it is included in the ESI (e.g. “Beckhoff EL5xxx.xml”).

The Advanced button must be used for switching between the two views.

In Fig. *Display of EL3204 firmware version* the firmware version of the selected EL3204 is shown as 03 in CoE entry 0x100A.

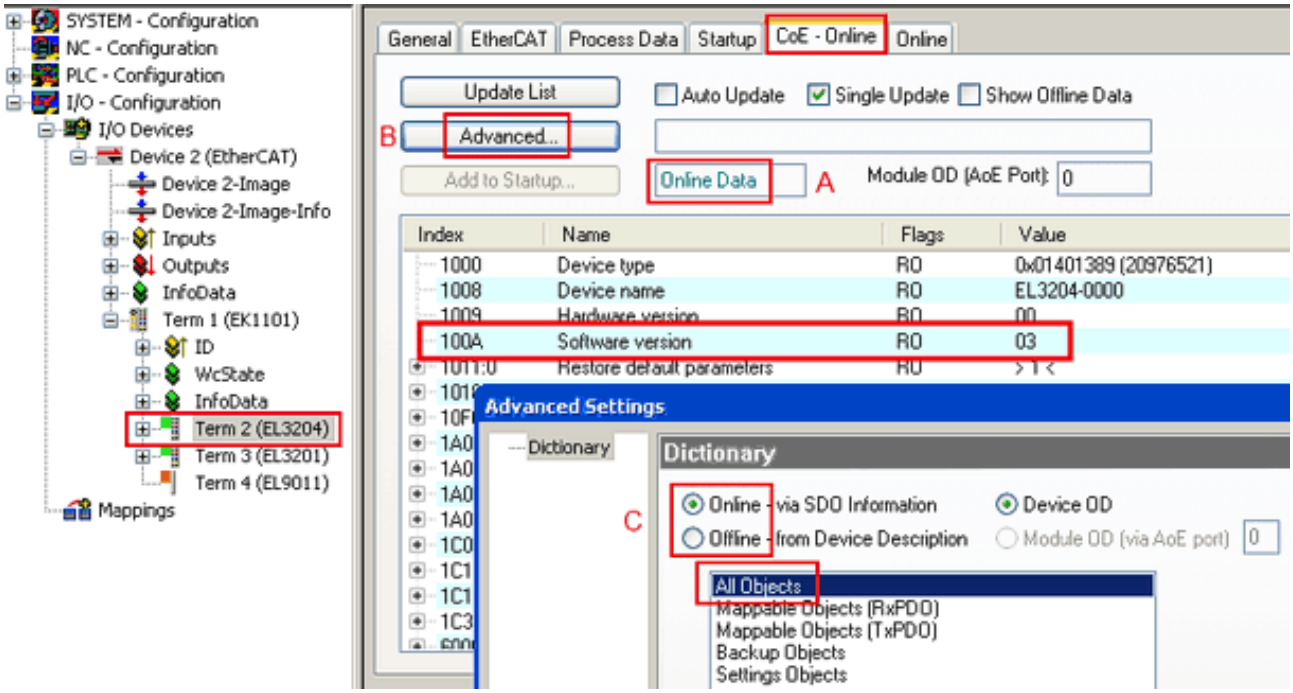


Fig. 188: Display of EL3204 firmware version

In (A) TwinCAT 2.11 shows that the Online CoE directory is currently displayed. If this is not the case, the Online directory can be loaded via the *Online* option in Advanced Settings (B) and double-clicking on *AllObjects*.

11.3.3 Updating controller firmware *.efw

CoE directory

The Online CoE directory is managed by the controller and stored in a dedicated EEPROM, which is generally not changed during a firmware update.

Switch to the *Online* tab to update the controller firmware of a slave, see Fig. *Firmware Update.*

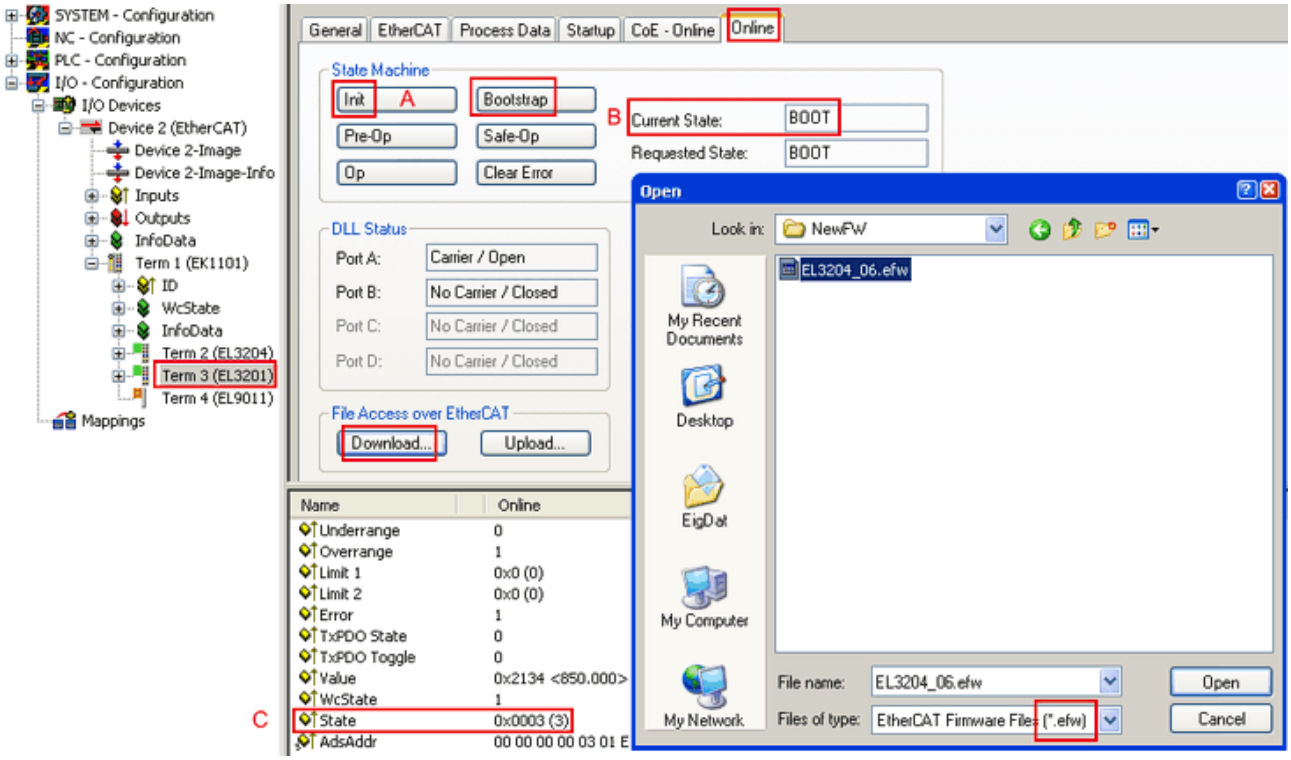
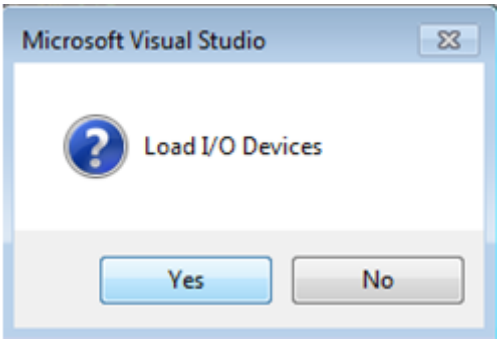
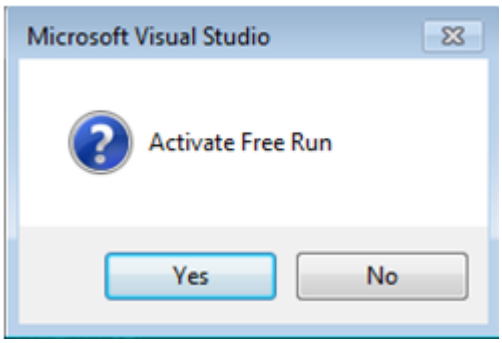


Fig. 189: Firmware Update

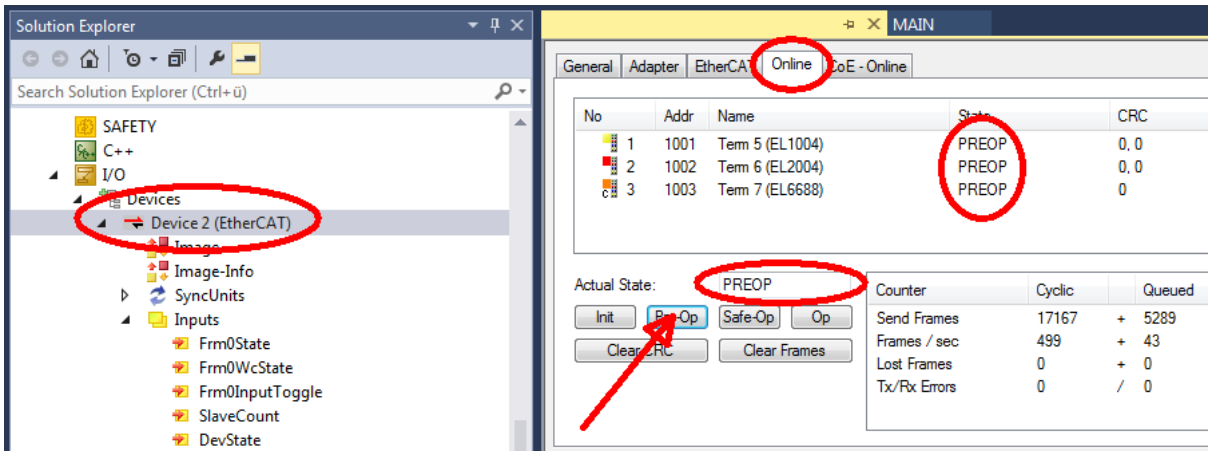
Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

- Switch TwinCAT system to ConfigMode/FreeRun with cycle time >= 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.



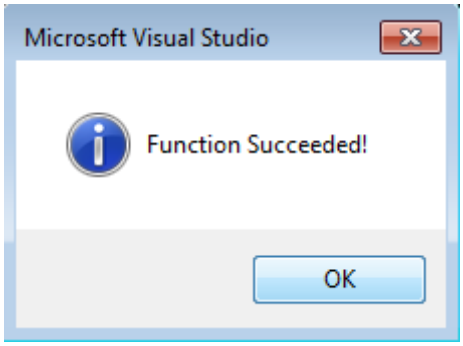


- Switch EtherCAT Master to PreOP



- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP

- Check the current status (B, C)
- Download the new *efw file (wait until it ends). A password will not be necessary usually.



- After the download switch to INIT, then PreOP
- Switch off the slave briefly (don't pull under voltage!)
- Check within CoE 0x100A, if the FW status was correctly overtaken.

11.3.4 FPGA firmware *.rbf

If an FPGA chip deals with the EtherCAT communication an update may be accomplished via an *.rbf file.

- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

The firmware version number included in the terminal serial number contains both firmware components. If one of these firmware components is modified this version number is updated.

Determining the version via the TwinCAT System Manager

The TwinCAT System Manager indicates the FPGA firmware version. Click on the Ethernet card of your EtherCAT strand (Device 2 in the example) and select the *Online* tab.

The *Reg:0002* column indicates the firmware version of the individual EtherCAT devices in hexadecimal and decimal representation.

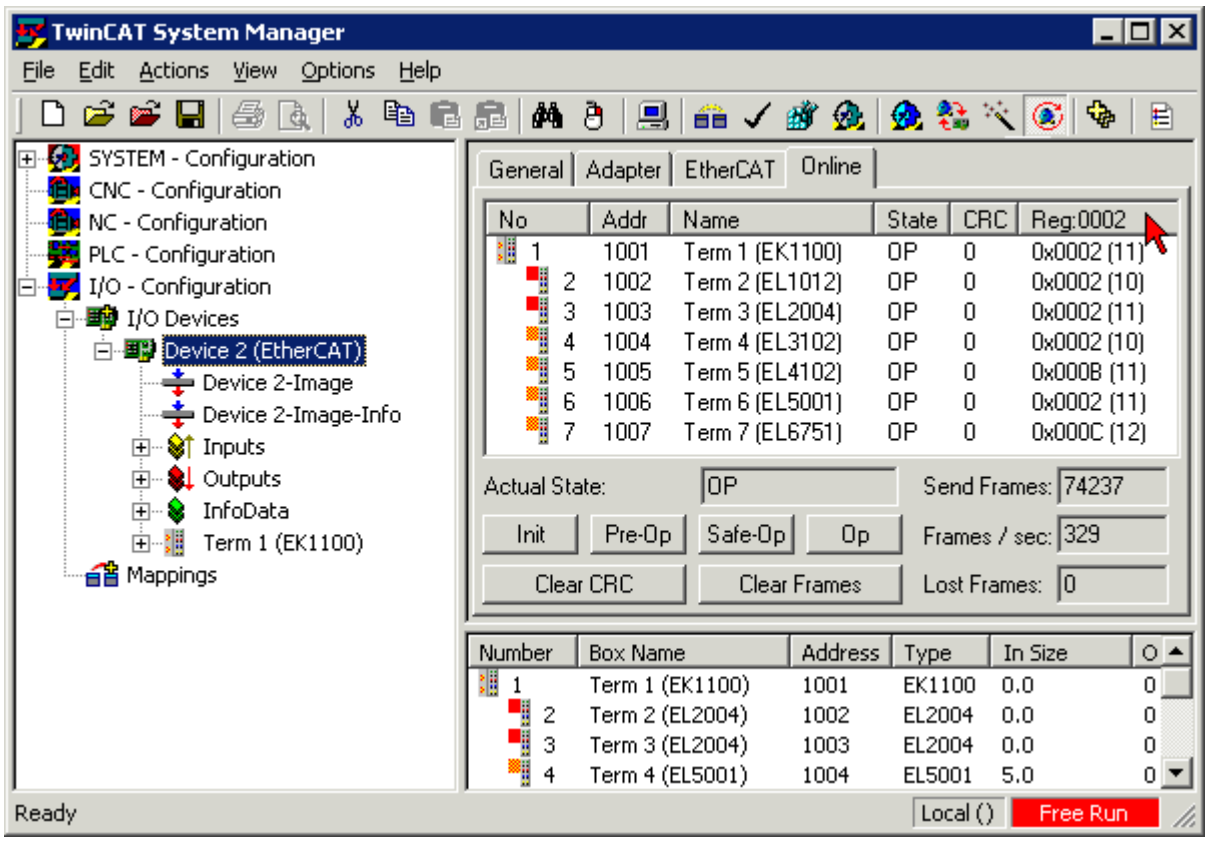


Fig. 190: FPGA firmware version definition

If the column *Reg:0002* is not displayed, right-click the table header and select *Properties* in the context menu.

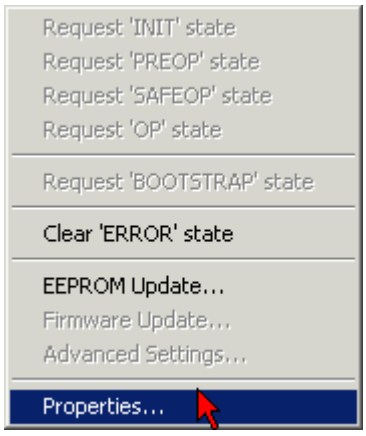


Fig. 191: Context menu *Properties*

The *Advanced Settings* dialog appears where the columns to be displayed can be selected. Under *Diagnosis/***Online View** select the *'0002 ETxxxx Build'* check box in order to activate the FPGA firmware version display.

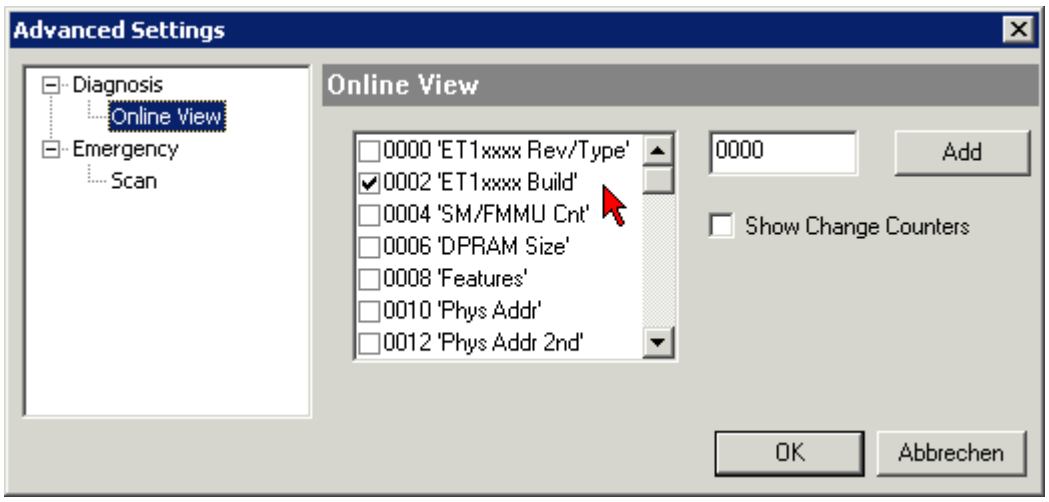


Fig. 192: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

- of an EtherCAT coupler the coupler must have FPGA firmware version 11 or higher;
- of an E-Bus Terminal the terminal must have FPGA firmware version 10 or higher.

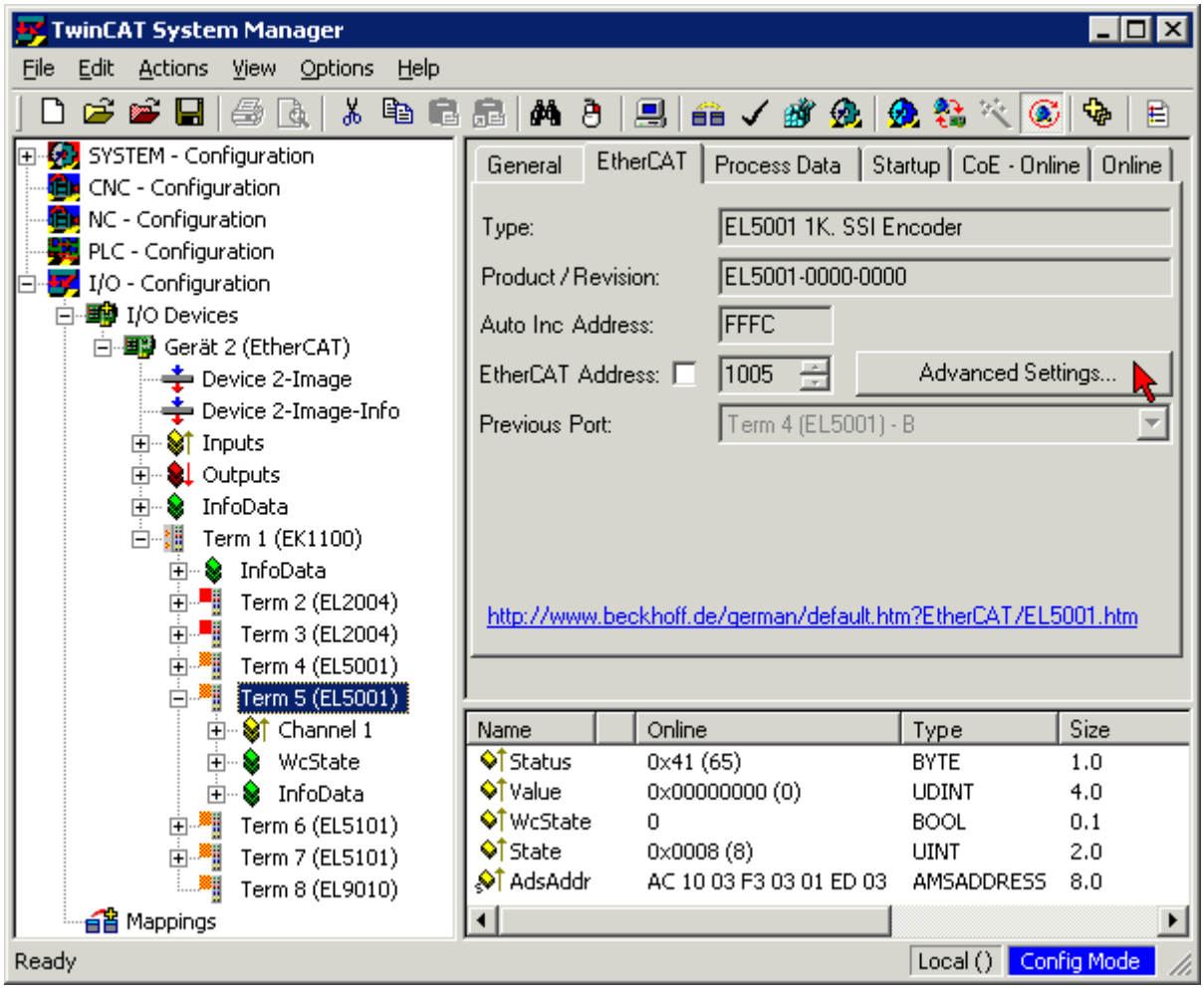
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

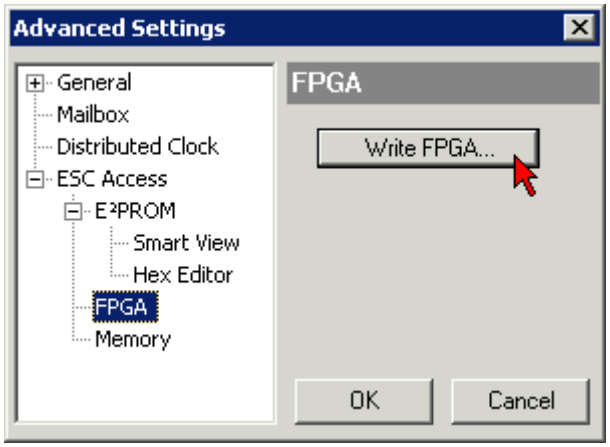
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- Switch TwinCAT system to ConfigMode/FreeRun with cycle time >= 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.

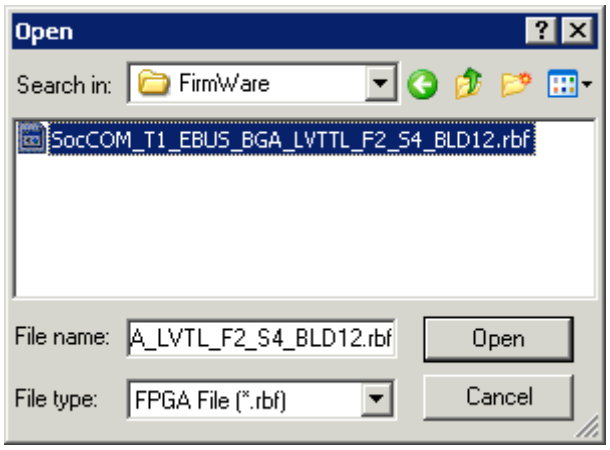
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
  click the *Advanced Settings* button in the *EtherCAT* tab:



- The *Advanced Settings* dialog appears. Under *ESC Access/E²PROM*/FPGA click on *Write FPGA* button:



- Select the file (*.rbf) with the new FPGA firmware, and transfer it to the EtherCAT device:



- Wait until download ends
- Switch slave current less for a short time (don't pull under voltage!). In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.
- Check the new FPGA status

NOTICE

Risk of damage to the device!

A download of firmware to an EtherCAT device must not be interrupted in any case! If you interrupt this process by switching off power supply or disconnecting the Ethernet link, the EtherCAT device can only be recommissioned by the manufacturer!

11.3.5 Simultaneous updating of several EtherCAT devices

The firmware and ESI descriptions of several devices can be updated simultaneously, provided the devices have the same firmware file/ESI.

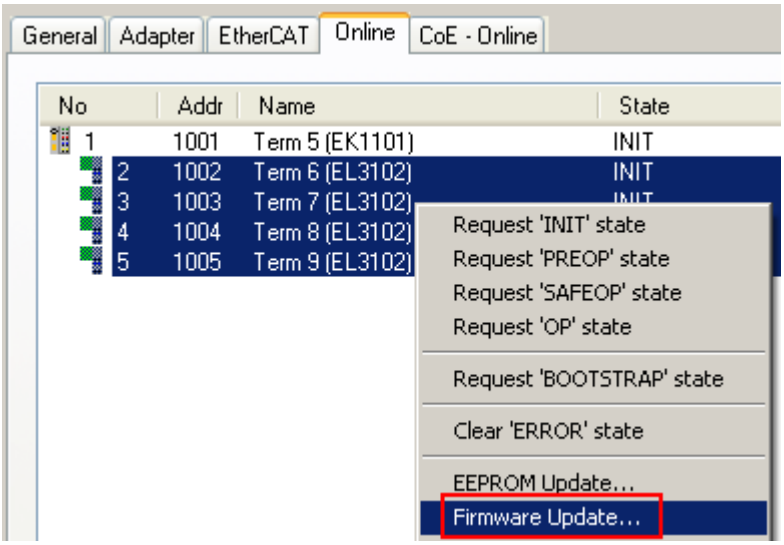


Fig. 193: Multiple selection and firmware update

Select the required slaves and carry out the firmware update in BOOTSTRAP mode as described above.

11.4 Restoring the delivery state

To restore the delivery state (factory settings) of CoE objects for EtherCAT devices (“slaves”), the CoE object *Restore default parameters*, SubIndex 001 can be used via EtherCAT master (e.g. TwinCAT) (see Fig. *Selecting the Restore default parameters PDO*).

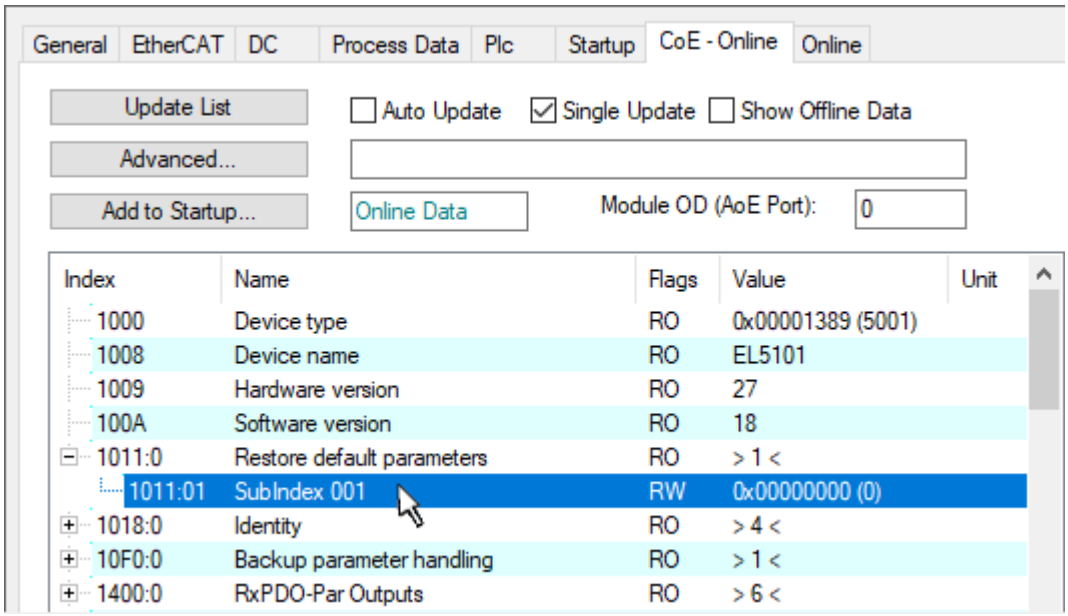


Fig. 194: Selecting the *Restore default parameters* PDO

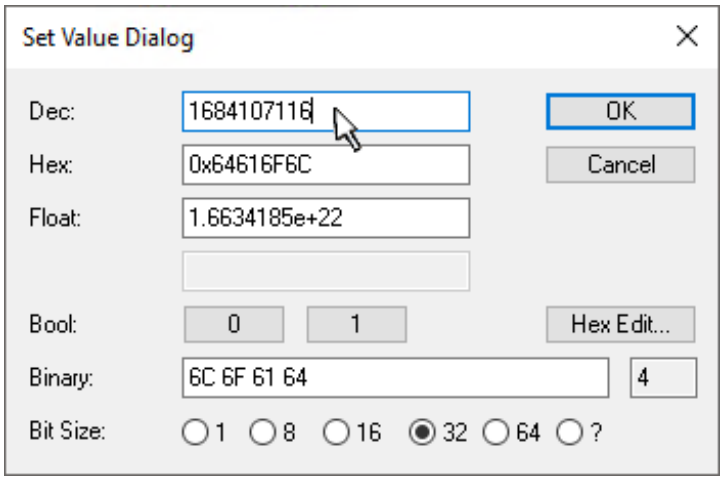


Fig. 195: Entering a restore value in the Set Value dialog

Double-click on *SubIndex 001* to enter the Set Value dialog. Enter the reset value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* (ASCII: “load”) and confirm with *OK* (Fig. *Entering a restore value in the Set Value dialog*).

- All changeable entries in the slave are reset to the default values.
- The values can only be successfully restored if the reset is directly applied to the online CoE, i.e. to the slave. No values can be changed in the offline CoE.
- TwinCAT must be in the RUN or CONFIG/Freerun state for this; that means EtherCAT data exchange takes place. Ensure error-free EtherCAT transmission.
- No separate confirmation takes place due to the reset. A changeable object can be manipulated beforehand for the purposes of checking.
- This reset procedure can also be adopted as the first entry in the startup list of the slave, e.g. in the state transition PREOP->SAFEOP or, as in Fig. *CoE reset as a startup entry*, in SAFEOP->OP.

All backup objects are reset to the delivery state.

Alternative restore value

In some older terminals (FW creation approx. before 2007) the backup objects can be switched with an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164.

An incorrect entry for the restore value has no effect.

11.5 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Support

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

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

Hotline:	+49 5246 963 157
e-mail:	support@beckhoff.com
web:	www.beckhoff.com/support

Service

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

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

Hotline:	+49 5246 963 460
e-mail:	service@beckhoff.com
web:	www.beckhoff.com/service

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
e-mail:	info@beckhoff.com
web:	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

DeviceNet and EtherNet/IP are trademarks of ODVA, Inc.

DSP System Toolbox, Embedded Coder, MATLAB, MATLAB Coder, MATLAB Compiler, MathWorks, Predictive Maintenance Toolbox, Simscape, Simscape™ Multibody™, Simulink, Simulink Coder, Stateflow and ThingSpeak are registered trademarks of The MathWorks, Inc.

EnDat is a trademark of Dr. Johannes Heidenhain GmbH.

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

More Information:
www.beckhoff.com/EL5131

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