

Documentation | EN

# EL5122

2 Channel Incremental Encoder Interface, 5 V single-ended (TTL, Open Collector)





# Table of contents

<b>1 Foreword</b> .....	<b>7</b>
1.1 Notes on the documentation .....	7
1.2 Guide through documentation .....	8
1.3 Safety instructions .....	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).....	13
1.5.4 Electronic access to the BIC (eBIC).....	15
<b>2 Product description</b> .....	<b>17</b>
2.1 EL5122 - Introduction .....	17
2.2 EL5122 - Technical data .....	18
2.3 Start.....	18
2.4 EL51xx series overview .....	19
2.5 Incremental encoder basics .....	20
2.6 Technical properties .....	21
2.6.1 Signal types.....	21
2.6.2 Gate/Latch inputs.....	23
2.6.3 Encoder operating voltage (supply voltage).....	24
<b>3 Basics communication</b> .....	<b>25</b>
3.1 EtherCAT basics .....	25
3.2 EtherCAT cabling – wire-bound .....	25
3.3 General notes for setting the watchdog .....	26
3.4 EtherCAT State Machine .....	28
3.5 CoE Interface .....	30
3.6 Distributed Clock .....	35
<b>4 Mounting and wiring</b> .....	<b>36</b>
4.1 Instructions for ESD protection .....	36
4.2 Installation on mounting rails.....	37
4.3 Connection .....	39
4.3.1 Connection system.....	39
4.3.2 Wiring.....	42
4.3.3 Shielding .....	43
4.4 Note - Power supply .....	43
4.5 Installation positions.....	44
4.6 Positioning of passive Terminals.....	46
4.7 EL5122 - Connection .....	47
4.7.1 TTL mode.....	48
4.7.2 Open Collector mode .....	49
4.8 EL5122 - LEDs.....	50
4.9 Disposal .....	50
<b>5 Commissioning</b> .....	<b>51</b>

5.1	TwinCAT Quick Start.....	51
5.1.1	TwinCAT 2 .....	54
5.1.2	TwinCAT 3 .....	64
5.2	TwinCAT Development Environment .....	77
5.2.1	Installation of the TwinCAT real-time driver .....	78
5.2.2	Notes regarding ESI device description .....	84
5.2.3	TwinCAT ESI Updater .....	88
5.2.4	Distinction between Online and Offline .....	88
5.2.5	OFFLINE configuration creation.....	89
5.2.6	ONLINE configuration creation .....	94
5.2.7	EtherCAT subscriber configuration .....	102
5.2.8	NC configuration (motion) .....	111
5.3	General Commissioning Instructions for an EtherCAT Slave .....	115
<b>6</b>	<b>EL5122 - Commissioning .....</b>	<b>123</b>
6.1	Overview of EL5122 functions .....	123
6.2	Process data .....	124
6.2.1	Sync Manager (SM) .....	124
6.2.2	PDO assignment .....	125
6.2.3	Predefined PDO Assignment .....	130
6.2.4	Synchronicity mode .....	133
6.2.5	EtherCAT cycle time .....	133
6.3	Basic functions .....	134
6.3.1	Counter value .....	134
6.3.2	Reset counter value via gate/latch combination input.....	139
6.3.3	Set counter value via gate/latch combination input.....	140
6.3.4	Save counter value .....	142
6.3.5	Lock counter value .....	144
6.4	Extended functionalities .....	145
6.4.1	Frequency measurement .....	145
6.4.2	Period value measurement .....	147
6.4.3	Velocity, speed calculation.....	148
6.4.4	Adjustable interference pulse filters .....	150
6.4.5	Plausibility check.....	152
6.5	Inputs .....	153
6.5.1	Gate/Latch combination input .....	154
6.6	Diagnostics.....	155
6.6.1	Diagnostics - basic principles of diag messages.....	155
6.6.2	EL5122 diagnostics.....	166
6.7	Object description and parameterization.....	169
6.7.1	Restore object .....	169
6.7.2	Configuration data.....	170
6.7.3	Command object.....	171
6.7.4	Input data .....	172
6.7.5	Output data .....	173
6.7.6	Information / diagnostic data (channel specific).....	173
6.7.7	Information / diagnostic data (device specific) .....	173

6.7.8	Standard objects .....	174
<b>7</b>	<b>Appendix .....</b>	<b>191</b>
7.1	EtherCAT AL Status Codes .....	191
7.2	Firmware compatibility .....	191
7.3	Firmware Update EL/ES/EM/ELM/EP/EPP/ERPxxxx .....	191
7.3.1	Device description ESI file/XML .....	193
7.3.2	Firmware explanation .....	196
7.3.3	Updating controller firmware *.efw .....	196
7.3.4	FPGA firmware *.rbf .....	198
7.3.5	Simultaneous updating of several EtherCAT devices .....	202
7.4	Restoring the delivery state .....	203
7.5	Support and Service .....	204



# 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®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

### Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 with corresponding applications or registrations in various other countries.



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.

## 1.2 Guide through documentation

### NOTICE



#### Further components of documentation

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

Title	Description
<b>EtherCAT System Documentation</b> ( <a href="#">PDF</a> )	<ul style="list-style-type: none"> <li>• System overview</li> <li>• EtherCAT basics</li> <li>• Cable redundancy</li> <li>• Hot Connect</li> <li>• EtherCAT devices configuration</li> </ul>
<b>Infrastructure for EtherCAT/Ethernet</b> ( <a href="#">PDF</a> )	Technical recommendations and notes for design, implementation and testing
<b>Software Declarations I/O</b> ( <a href="#">PDF</a> )	Open source software declarations for Beckhoff I/O components

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

- the “Documentation and Download” area of the respective product page,
- the [Download finder](#),
- the [Beckhoff Information System](#).



## 1.3 Safety instructions

### Safety regulations

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

### Exclusion of liability

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

### Personnel qualification

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

### Signal words

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

#### Personal injury warnings

**⚠ DANGER**

Hazard with high risk of death or serious injury.

**⚠ WARNING**

Hazard with medium risk of death or serious injury.

**⚠ CAUTION**

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

#### Warning of damage to property or environment

**NOTICE**

The environment, equipment, or data may be damaged.

#### Information on handling the product



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

## 1.4 Documentation issue status

Version	Comment
1.1	<ul style="list-style-type: none"><li>• Update chapter "Version identification of EtherCAT devices"</li><li>• Update chapter "Technical data"</li><li>• Update chapter "EL51xx series overview"</li><li>• Update chapter "Mounting and wiring"</li><li>• Update revision status</li><li>• Update structure</li></ul>
1.0	<ul style="list-style-type: none"><li>• First release</li></ul>
0.1	<ul style="list-style-type: none"><li>• First preliminary documentation for EL5122</li></ul>

## 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. “EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)”.
- 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**

KK - week of production (CW, calendar week)

YY - year of production

FF - firmware version

HH - hardware version

Example with serial number 12 06 3A 02:

12 - production week 12

06 - production year 2006

3A - firmware version 3A

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.

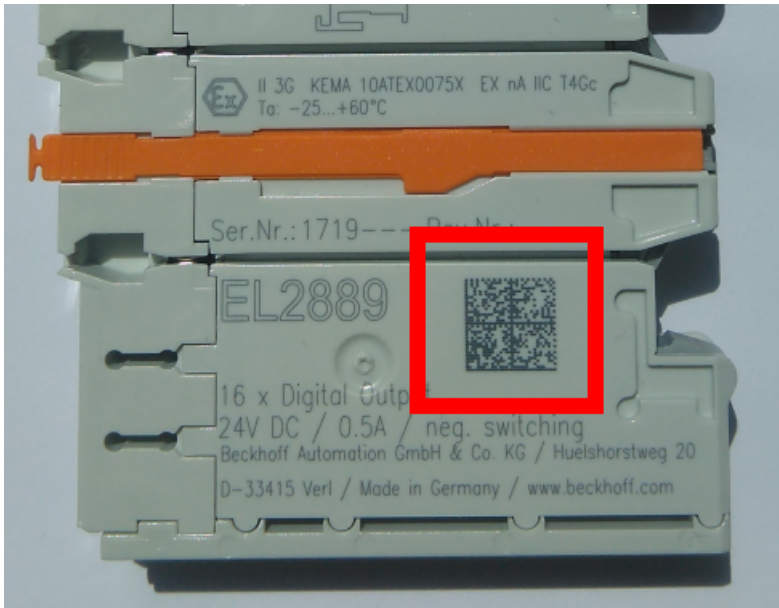


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

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	<b>Beckhoff order number</b>	1P	8	<b>1P</b> 072222
2	Beckhoff Traceability Number (BTN)	<b>Unique serial number, see note below</b>	SBTN	12	<b>S</b> BTNk4p562d7
3	Article description	<b>Beckhoff article description, e.g. EL1008</b>	1K	32	<b>1K</b> EL1809
4	Quantity	<b>Quantity in packaging unit, e.g. 1, 10, etc.</b>	Q	6	<b>Q</b> 1
5	Batch number	Optional: Year and week of production	2P	14	<b>2P</b> 401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	<b>51S</b> 678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	<b>30P</b> 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:

**1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:



Fig. 3: Example DMC **1P**072222**S**BTNk4p562d7**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 information.

## 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.

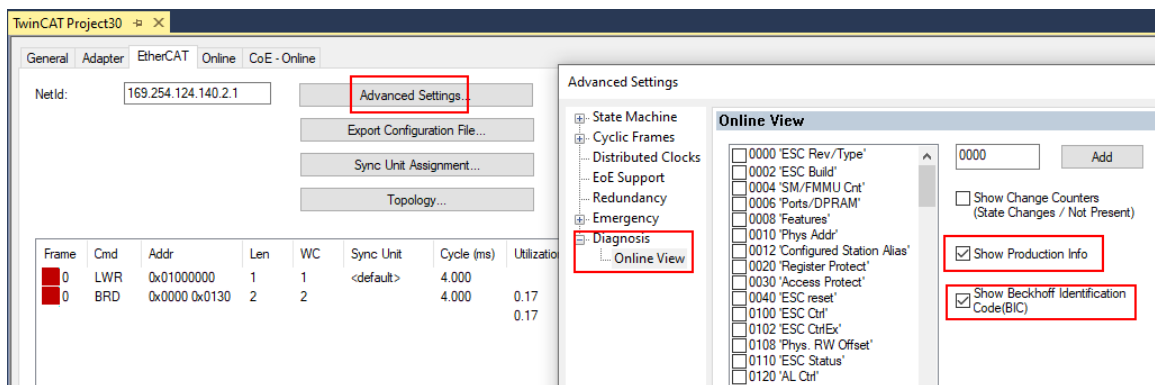
### EtherCAT devices (IP20, IP67)

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:

No	Addr	Name	State	CRC	Fw	Hw	Production Data	ItemNo	BTN	Description	Quantity	BatchNo	SerialNo
1	1001	Term 1 (EK1100)	OP	0.0	0	0	---						
2	1002	Term 2 (EL1018)	OP	0.0	0	0	2020 KW36 Fr	072222	k4p562d7	EL1809	1		678294
3	1003	Term 3 (EL3204)	OP	0.0	7	6	2012 KW24 Sa						
4	1004	Term 4 (EL2004)	OP	0.0	0	0	---	072223	k4p562d7	EL2004	1		678295
5	1005	Term 5 (EL1008)	OP	0.0	0	0	---						
6	1006	Term 6 (EL2008)	OP	0.0	0	12	2014 KW14 Mo						
7	1007	Term 7 (EK1110)	OP	0	1	8	2012 KW25 Mo						

- 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	1P158442SBTN0008jebp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bf277e

- 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\_Uilities* 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 Product description

### 2.1 EL5122 - Introduction

2-channel incremental encoder interface, 5 V single-ended (TTL, Open Collector)

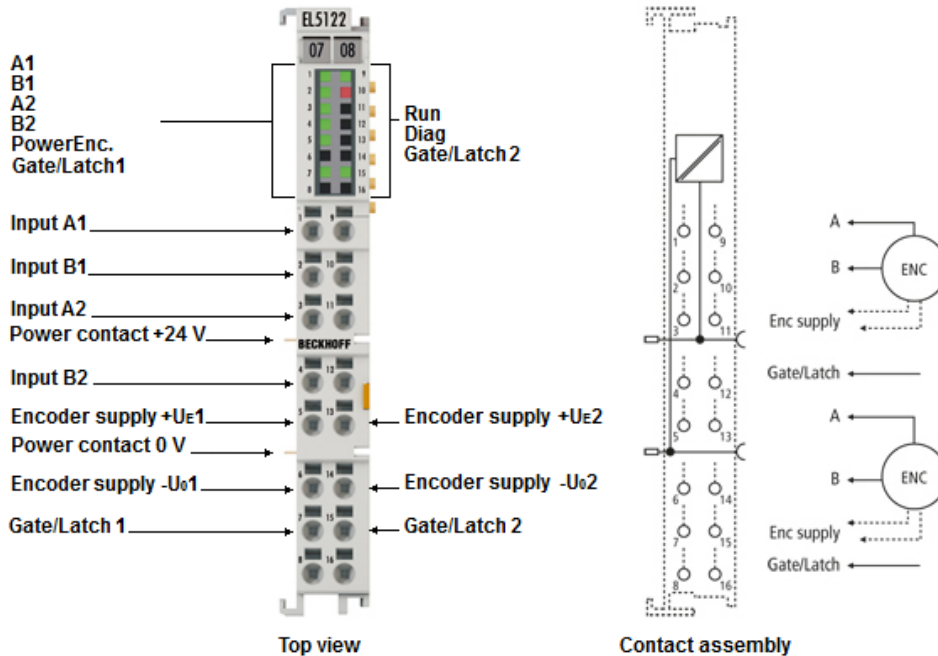


Fig. 4: EL5122

The EL5122 EtherCAT Terminal is an interface for the direct connection of two incremental encoders with A and B track. Encoders with single-ended-signals (TTL and Open Collector) can be connected. Thanks to internal pull-up resistors, no external wiring is required for open collector analysis.

One 24 V digital input is available per channel 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 encoders can be supplied with 5 V, 12 V or 24 V directly via the connection points of the terminal.

Special features:

- Save, lock and set the counter
- Integrated frequency and period measurement
- Optionally usable as a 5 V up/down counter
- Synchronous reading of the position value via Distributed Clocks

In addition, the EL5122 enables the measurement of a period, frequency or velocity with a resolution of 10 ns and supports the synchronous reading of the encoder value together with other input data in the EtherCAT system via the high-precision EtherCAT Distributed Clocks (DC). The use of encoder profiles allows the simple and fast linking of the process data to the Motion Control application.

#### Quick links

[Basics communication \[► 25\]](#)

[Creation of the TwinCAT configuration \[► 94\]](#)

[Process data \[► 124\]](#)

[Configuration data \[► 170\]](#)

[Object description \[► 169\]](#)

[LEDs \[► 50\] and connection \[► 47\]](#)

## 2.2 EL5122 - Technical data

Technical data	EL5122
Encoder type	Incremental, single-ended (TTL, Open Collector), counter, pulse generator
Encoder connection	Single-ended connection (TTL, Open Collector): A, B Counters, pulse generators: A, B
Number of channels	2 x A, B
Additional inputs	Gate/Latch ( $24 V_{DC}$ , $t_{ON} > 1 \mu s$ ) per channel
Encoder operating voltage	$5 V_{DC}$ (preset), $12 V_{DC}$ , $24 V_{DC}$ switchable, 0.3 A sum current (generated from the $24 V_{DC}$ power contacts)
Counter	32 bit (default) or 16 bit switchable
Cut-off frequency	TTL mode: 4 million increments/s with 4-fold evaluation, corresponds to 1 MHz Open Collector: 400.000 increments/s with 4-fold evaluation, corresponds to 100 kHz
Quadrature decoder	4-fold evaluation (preset), 2-fold, 1-fold evaluation switchable
Micro-increments resolution	no
Broken wire detection to sensor	no
Distributed Clocks	yes
Timestamp	no
Special functions	Period duration, frequency and speed measurement
Cycle time	min. 100 $\mu s$
Current consumption via E-bus	typ. 190 mA
Current consumption from power contacts	10 mA + load
Electrical isolation	500 V (E-bus/field voltage)
Configuration	via TwinCAT <a href="#">System Manager</a> [ <a href="#">102</a> ]
Weight	approx. 50 g
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
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
<a href="#">Mounting</a> [ <a href="#">37</a> ]	on 35 mm support rail according to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approvals / markings*	CE, EAC, UKCA

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

## 2.3 Start

For commissioning:

- Mount the EL5122 as described in the chapter [Mounting and wiring](#) [[36](#)]
- Configure the EL5122 in TwinCAT as described in the chapter [Commissioning](#) [[51](#)].
- Parameterize the EL5122 as described in chapter [EL5122 - Commissioning](#) [[123](#)].

## 2.4 EL51xx series overview

Technical data		EL5102 2 x A, B, C	EL5112		EL5122 2 x A, B	EL5131 1 x A, B, C
			1 x A, B, C	2 x A, B		
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 <sub>DC</sub> , 12 V <sub>DC</sub> , 24 V <sub>DC</sub> 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 2 x A, B, C	EL5112		EL5122 2 x A, B	EL5131 1 x A, B, C
			1 x A, B, C	2 x A, B		
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

## 2.5 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 ( $\bar{A}$ ,  $\bar{B}$ ,  $\bar{C}$ ) 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, $\bar{A}$ , B, $\bar{B}$ , C, $\bar{C}$
RS422 encoder	without zero pulse	A, $\bar{A}$ , B, $\bar{B}$
RS422 counter or pulse generator	with zero pulse	A, $\bar{A}$ , C, $\bar{C}$ ; counting direction specification via track B (B, $\bar{B}$ )
RS422 counter or pulse generator	without zero pulse	A, $\bar{A}$ ; counting direction specification via track B (B, $\bar{B}$ )
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.

### Cyclical output

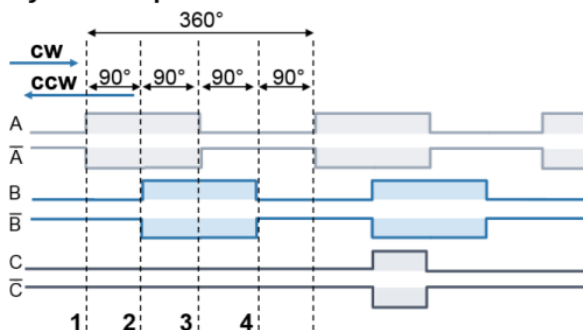


Fig. 5: Incremental signals

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.

## 2.6 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<sub>DC</sub> 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<sub>DC</sub>).

The terminal EL5122 provides the following inputs with the respective technical characteristics:

- Single-ended signals from TTL encoders and Open Collector encoders are supported.
- Gate/Latch input

The terminal also provides a parameterizable encoder supply.

- Encoder operating voltage

### 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  $\mu$ s 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.

### 2.6.1 Signal types

#### Supported encoders / signal types

Possible encoder connections are single-ended signals from TTL encoders and signals from Open Collector encoders through internal pull-up resistors.

The following signal types are supported:

Encoder	Signal type	Setting in index 0x80n1:1D "Counter mode"	Cut-off frequency	Comments
Encoder 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.
Counter/pulse generator without zero pulse track C		3		
Encoder without zero pulse track C	open collector	4	400,000 increments / s with 4-fold evaluation, corresponds to 100 kHz per track	No broken wire detection
Counter/pulse generator without zero pulse track C		5		

The correct wiring for the respective encoder can be found in chapter [Connection](#) [► 47].

### 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.

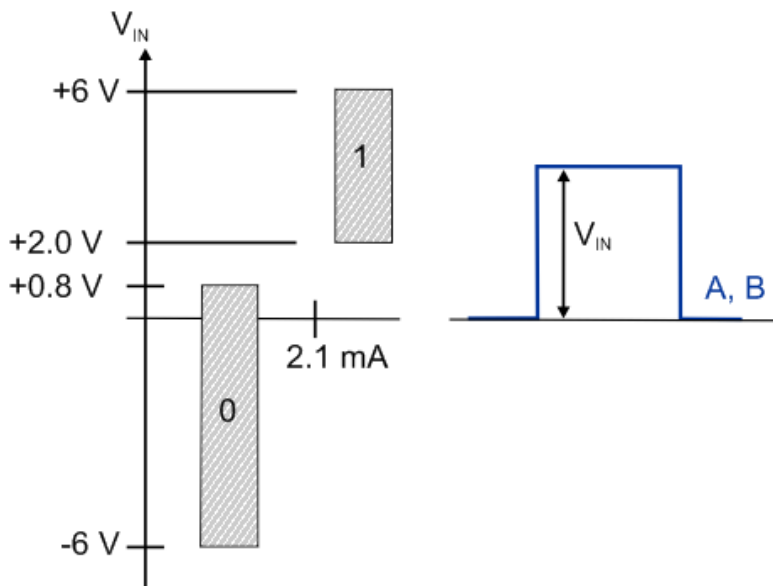


Fig. 6: TTL, Open Collector signal level (single-ended signal)

Key:

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

#### 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 are connected to 5 V via pull-up resistors (1 k $\Omega$ ).

### 2.6.2 Gate/Latch inputs

- The terminal provides one digital 24 V<sub>DC</sub> input per channel. The function of these inputs is explained in the chapter [Gate/Latch \(external latch\)](#) [▶ 154].

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

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

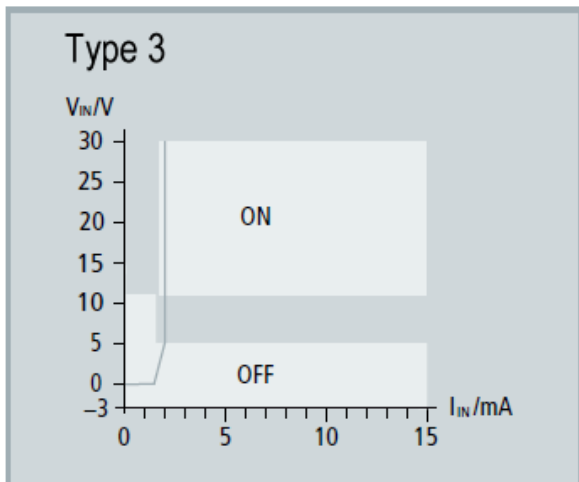


Fig. 7: Characteristic curve - input 24 V<sub>DC</sub> type 3

**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" is active, the stored value will be overwritten several times due to bouncing. As a result, you obtain not the first-saved, but the last-saved value in the parameter 0x60n0:12 "Latch value".
- 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.

### 2.6.3 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 \[▶ 171\]](#) "Supply voltage". An operating voltage of 5 V<sub>DC</sub> is preset. Voltage values of 5 V<sub>DC</sub>, 12 V<sub>DC</sub> and 24 V<sub>DC</sub> can be selected. The setting applies to both channels. Before switching to higher voltages, ensure that both encoders support the voltage range.

The following tolerances apply

Voltage range	Tolerance
5 V <sub>DC</sub>	+/- 5% (4.75 V ... 5.25 V)
12 V <sub>DC</sub>	+/- 10% (10.8 V ... 13.2 V)
24 V <sub>DC</sub>	-15% to +20% (20.4 V ... 28.8 V)

#### **i** Setting the encoder supply via index [0x8001:17 \[▶ 171\]](#)

The encoder supply is set centrally for both channels via the index [0x8001:17 \[▶ 171\]](#) (channel 1). The corresponding index [0x8011:17](#) of the second channel has no parameterization function.

#### NOTICE

##### Setting the encoder supply voltage

- Before switching to a higher voltage, make sure that the connected encoders support the selected voltage range!
- To write to [0x80n1:17](#) "Supply voltage" you have to set the value [0x72657375](#) (ASCII: "user") in index [0xF008 \[▶ 190\]](#) "Code word".



### 3 Basics communication

#### 3.1 EtherCAT basics

Please refer to the [EtherCAT System Documentation](#) for the EtherCAT fieldbus basics.

#### 3.2 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.

##### **i** Recommended cables

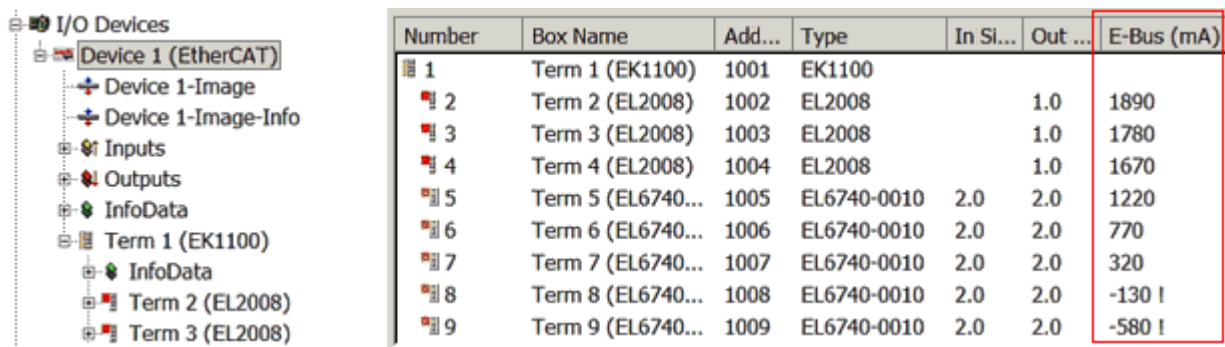
- It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
- RJ45 connector, field assembly ZS1090-0005
- EtherCAT cable, field assembly ZB9010, ZB9020

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.



The screenshot shows the 'I/O Devices' tree on the left, expanded to 'Device 1 (EtherCAT)'. The tree includes sub-items for 'Device 1-Image', 'Device 1-Image-Info', 'Inputs', 'Outputs', 'InfoData', and three terminal blocks: 'Term 1 (EK1100)', 'Term 2 (EL2008)', and 'Term 3 (EL2008)'. To the right, a table displays the current calculation for each terminal. The 'E-Bus (mA)' column is highlighted with a red box.

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

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!

### 3.3 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 (ESC) features two watchdogs:

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

Their times are individually parameterized in TwinCAT as follows:

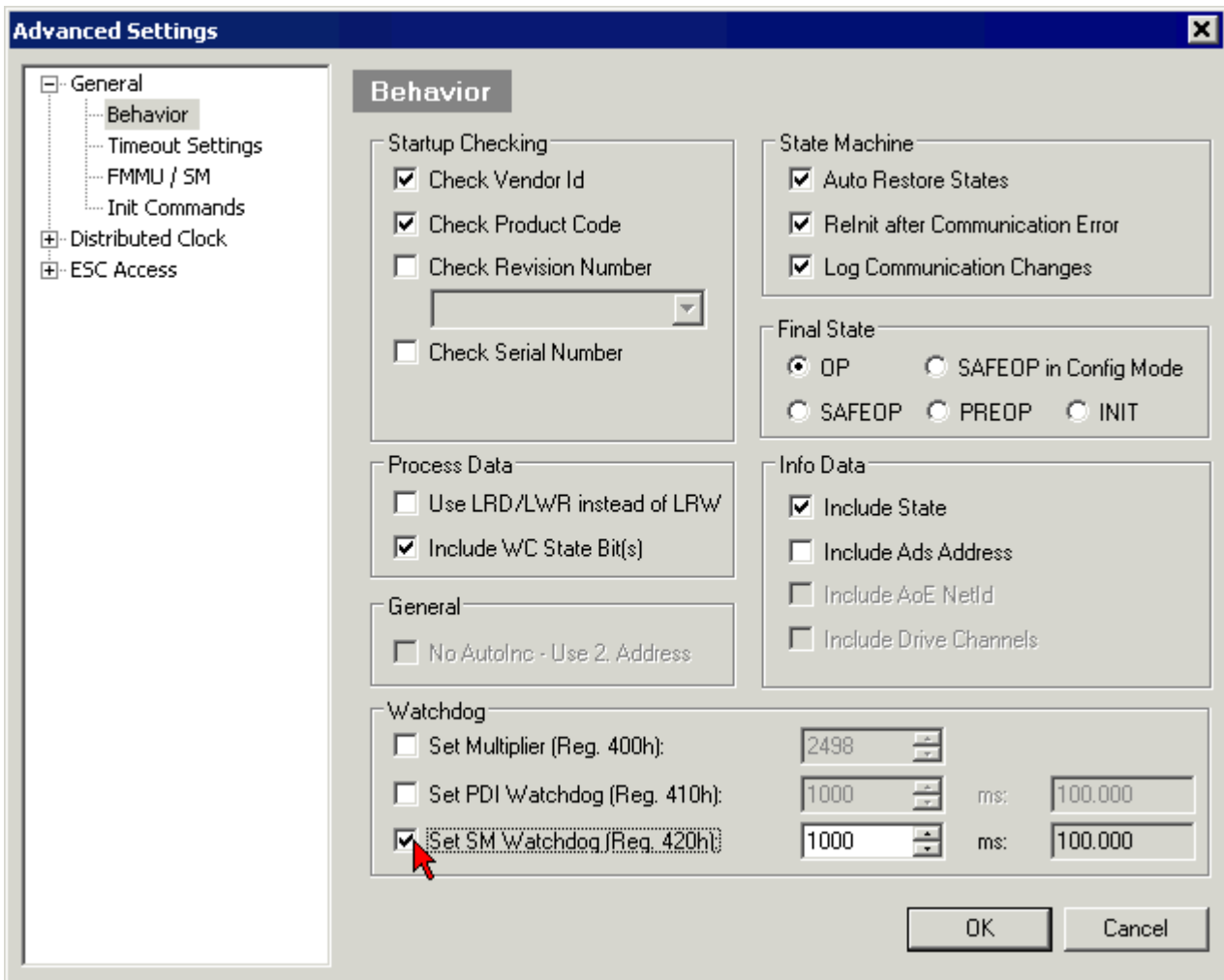


Fig. 9: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. x0400) 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 x0400/0410/0420: 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 Reg. 400/420 but executed by the 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 EtherCAT slave controller (ESC) for longer than the set and activated PDI watchdog time, this watchdog is triggered.

PDI (Process Data Interface) 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 =  $[1/25 \text{ MHz} * (\text{Watchdog multiplier} + 2)] * \text{PDI/SM watchdog}$

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

The value in Multiplier + 2 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 version -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.

## 3.4 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 and
- Operational
- Boot

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

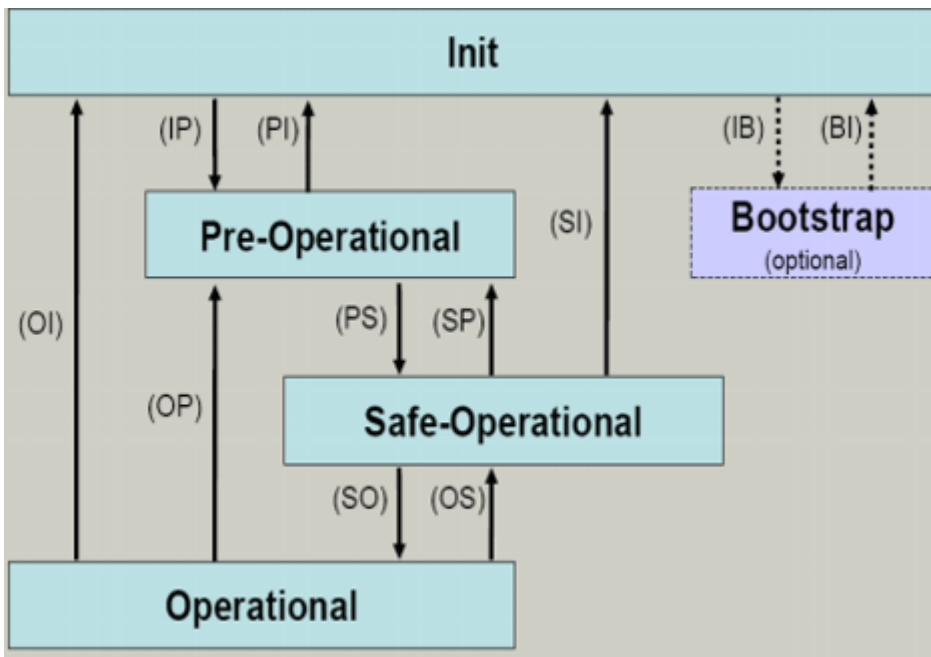


Fig. 10: States of the EtherCAT State Machine

**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 FMMU channels and, if the slave supports configurable mapping, PDO mapping or the sync manager PDO assignment. In this state the settings for the process data transfer and perhaps terminal-specific parameters that may differ from the default settings are also transferred.

**Safe-Operational (Safe-Op)**

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

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**

**i** The default set watchdog monitoring sets the outputs of the 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 watchdog 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.

## 3.5 CoE Interface

### General description

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

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

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

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

- Index: 0x0000 ... 0xFFFF (0...65535<sub>dec</sub>)
- SubIndex: 0x00...0xFF (0...255<sub>dec</sub>)

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 ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

---

### **i** 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:

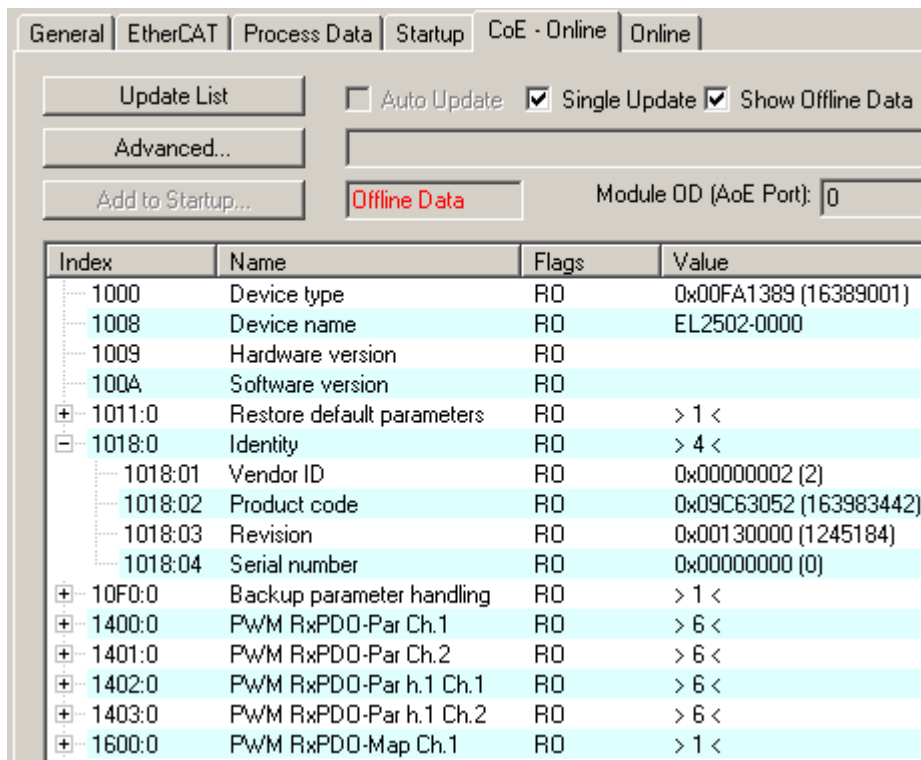


Fig. 11: "CoE Online" tab

The figure above 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), 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 [TwinCAT3 | 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. This can be done in write or read mode

- via the System Manager (Fig. "CoE Online" tab) by clicking  
This is useful for commissioning of the system/slaves. Click on the row of the index to be parameterized and enter a value in the "SetValue" dialog.
- from the control system/PLC via ADS, e.g. through 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.

**i 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 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.
- Function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

**i 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.

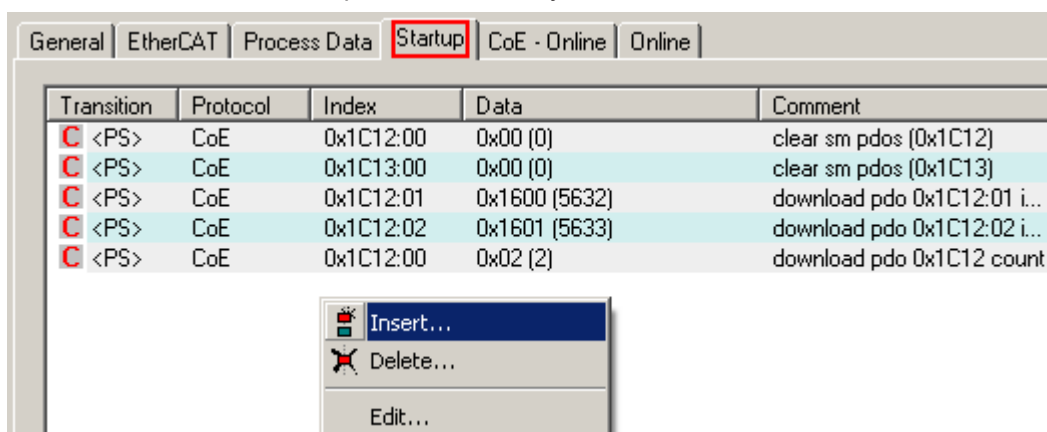


Fig. 12: Startup list in the TwinCAT System Manager

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

**Online/offline list**

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



In both cases a CoE 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** is shown in red.

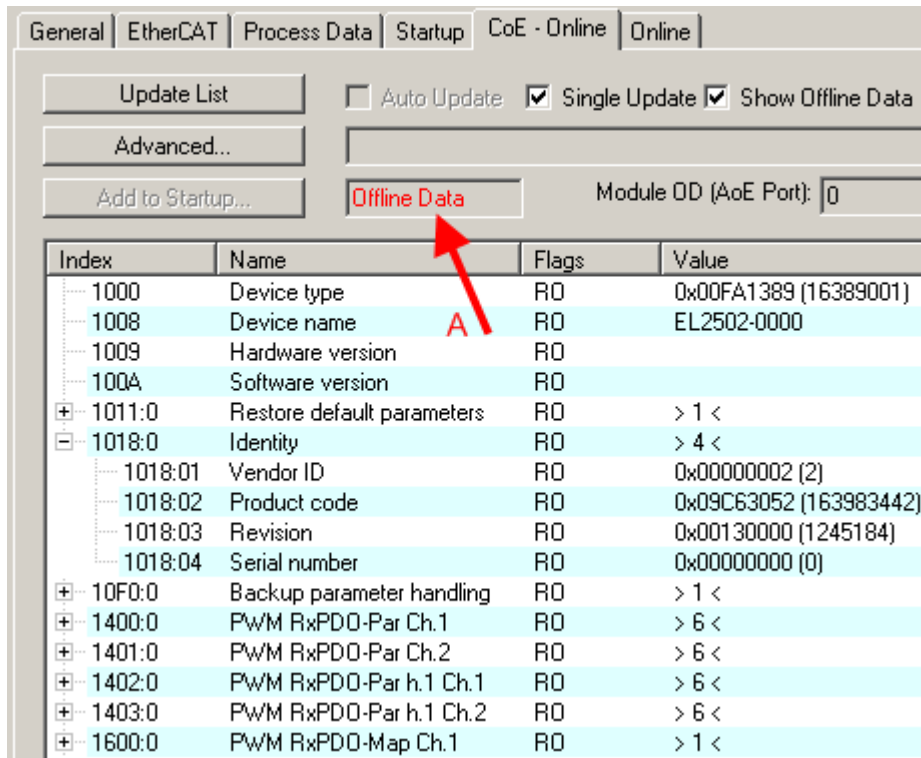


Fig. 13: Offline list

- 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 version of the equipment according to the electronic information is displayed
  - **Online** is shown in green.

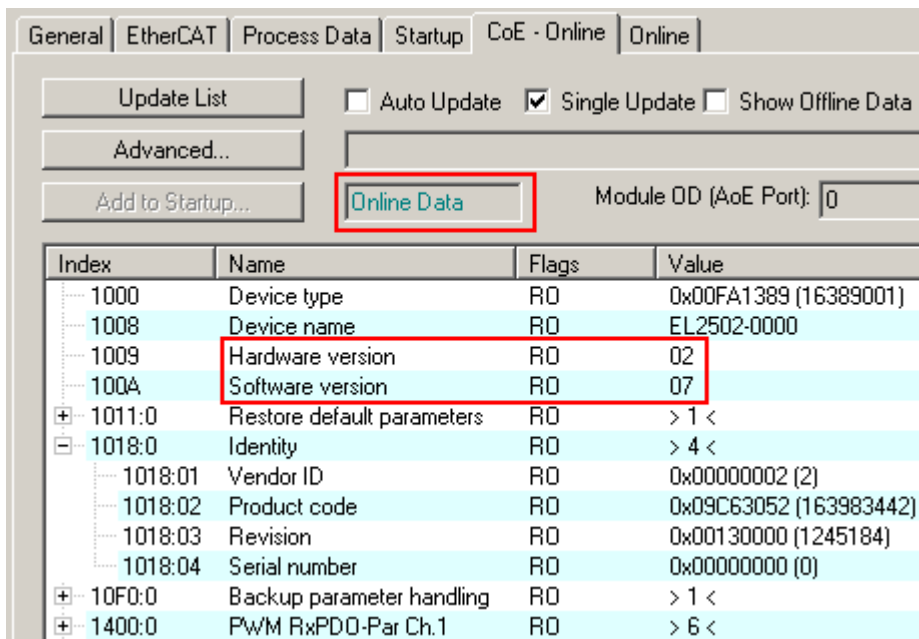


Fig. 14: Online list

### Channel-based order

The CoE list is available in EtherCAT devices that usually feature several functionally equivalent channels. For example, a 4-channel analog 0...10 V 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  $16_{dec}/10_{hex}$  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.

## 3.6 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](#).

## 4 Mounting and wiring

### 4.1 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.

- Please ensure you are electrostatically discharged and avoid touching the contacts of the device directly.
- Avoid contact with highly insulating materials (synthetic fibers, plastic film etc.).
- Surroundings (working place, packaging and personnel) should be grounded probably, when handling with the devices.
- Each assembly must be terminated at the right hand end with an [EL9011](#) or [EL9012](#) bus end cap, to ensure the protection class and ESD protection.

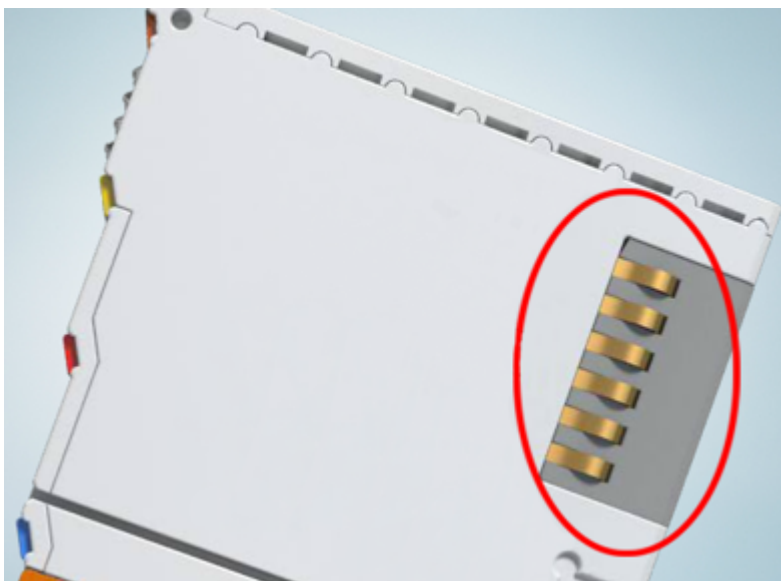


Fig. 15: Spring contacts of the Beckhoff I/O components

## 4.2 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

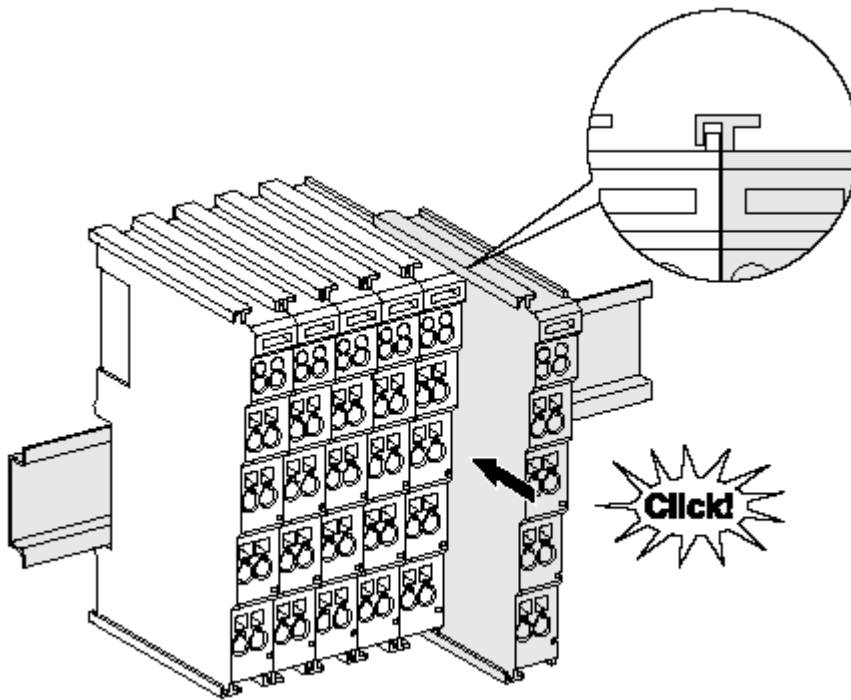


Fig. 16: Attaching on mounting rail

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.

#### **i** 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).

## Disassembly

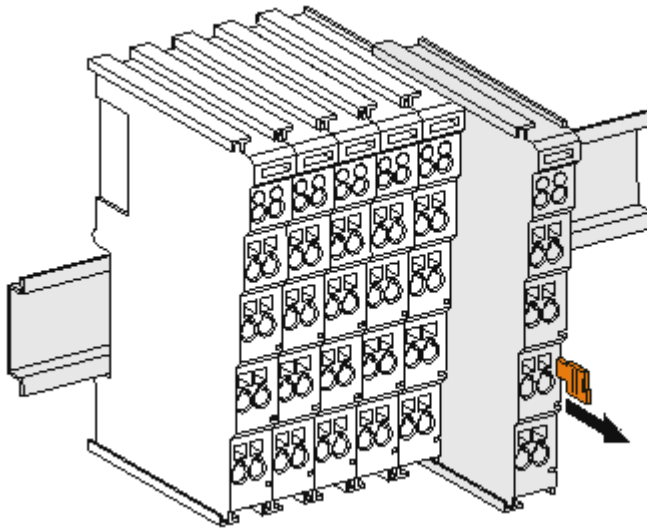


Fig. 17: Disassembling of terminal

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.

## 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 K-Bus/E-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 on the Bus Coupler (up to 24 V) or for higher voltages via power feed terminals.

### ● Power Contacts

**i** 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 (KL91xx, KL92xx or EL91xx, EL92xx) interrupt the power contacts and thus represent the start of a new supply rail.

## PE power contact

The power contact labeled PE can be used as a protective earth. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

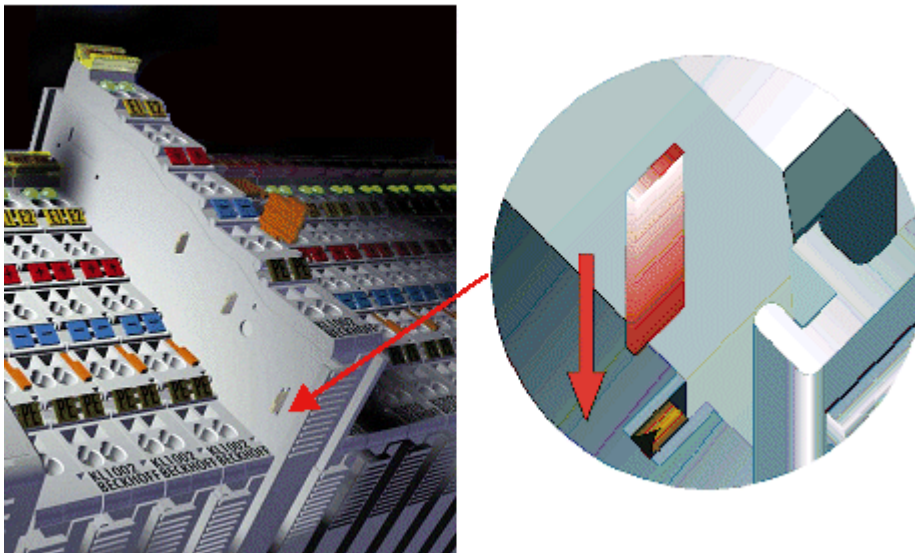


Fig. 18: Power contact on left side

**NOTICE**

**Possible damage of the device**

Note that, for reasons of electromagnetic compatibility, the PE 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 PE line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the PE 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.

**⚠ WARNING**

**Risk of electric shock!**

The PE power contact must not be used for other potentials!

## 4.3 Connection

### 4.3.1 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 ELxxxx and KLxxxx series have been tried and tested for years. They feature integrated screwless spring force technology for fast and simple assembly.

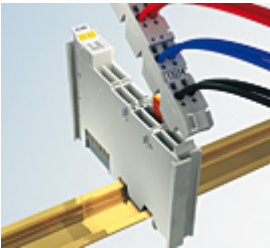
**Pluggable wiring (ESxxxx / KSxxxx)**

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<sup>2</sup> and 2.5 mm<sup>2</sup> 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 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**

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

---

**Ultrasonically “bonded” (ultrasonically welded) conductors**

---

**● Ultrasonically “bonded” conductors**

**i** It is also possible to connect the Standard and High Density Terminals with ultrasonically “bonded” (ultrasonically welded) conductors. In this case, please note the tables concerning the wire-size width [[▶ 42](#)]!

---

### 4.3.2 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

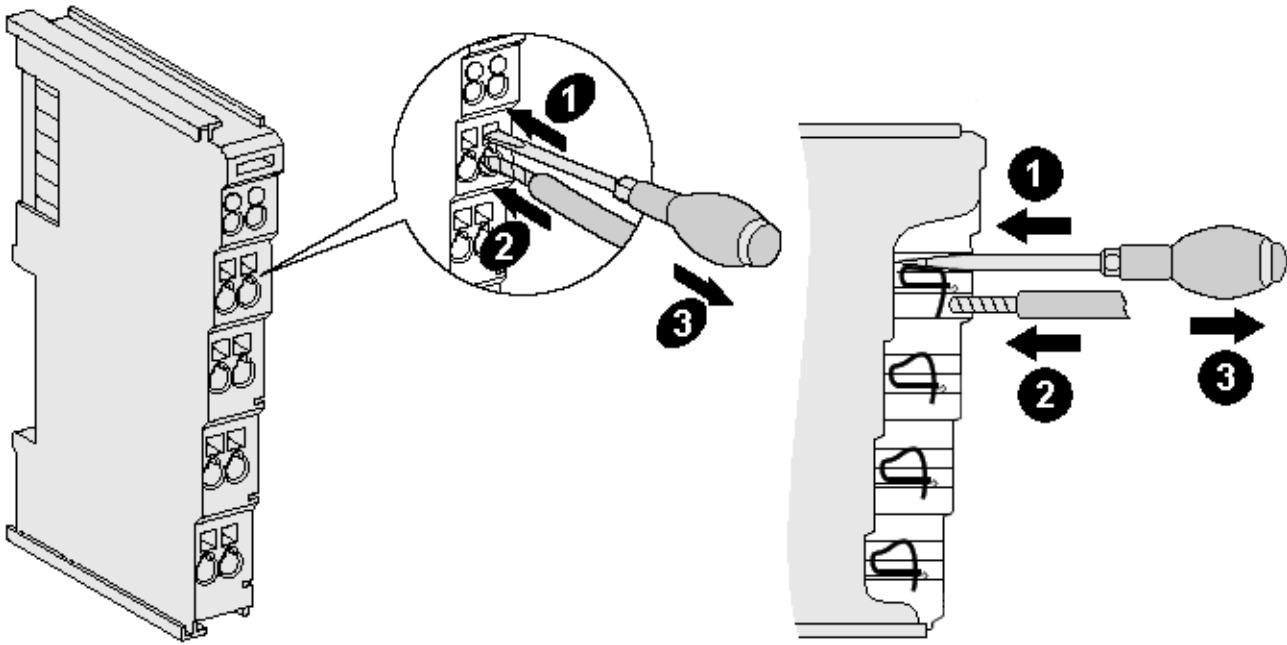


Fig. 22: Connecting a cable on a terminal point

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:

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. The terminal point closes automatically when the pressure is released, holding the wire securely and permanently.

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 mm <sup>2</sup>	0.08 ... 2.5 mm <sup>2</sup>
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm <sup>2</sup>	0.08 ... 2.5 mm <sup>2</sup>
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm <sup>2</sup>	0.14 ... 1.5 mm <sup>2</sup>
Wire stripping length	8 ... 9 mm	9 ... 10 mm

#### High Density Terminals ([HD Terminals](#) [▶ 40]) with 16 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 mm <sup>2</sup>
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm <sup>2</sup>
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm <sup>2</sup>
Wire size width (ultrasonically "bonded" conductors)	only 1.5 mm <sup>2</sup> (see notice [▶ 41])
Wire stripping length	8 ... 9 mm

### 4.3.3 Shielding

**Shielding**

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

### 4.4 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 (Safety Extra Low Voltage) supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV (Protective Extra Low Voltage) supply also requires a safe connection to the protective conductor.

## 4.5 Installation positions

### NOTICE

#### Constraints regarding installation position and operating temperature range

Please refer to the technical data for a terminal to ascertain whether any restrictions regarding the installation position and/or the operating temperature range have been specified. When installing high power dissipation terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

#### Optimum installation position (standard)

The optimum 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 for standard installation position*). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. "From below" is relative to the acceleration of gravity.

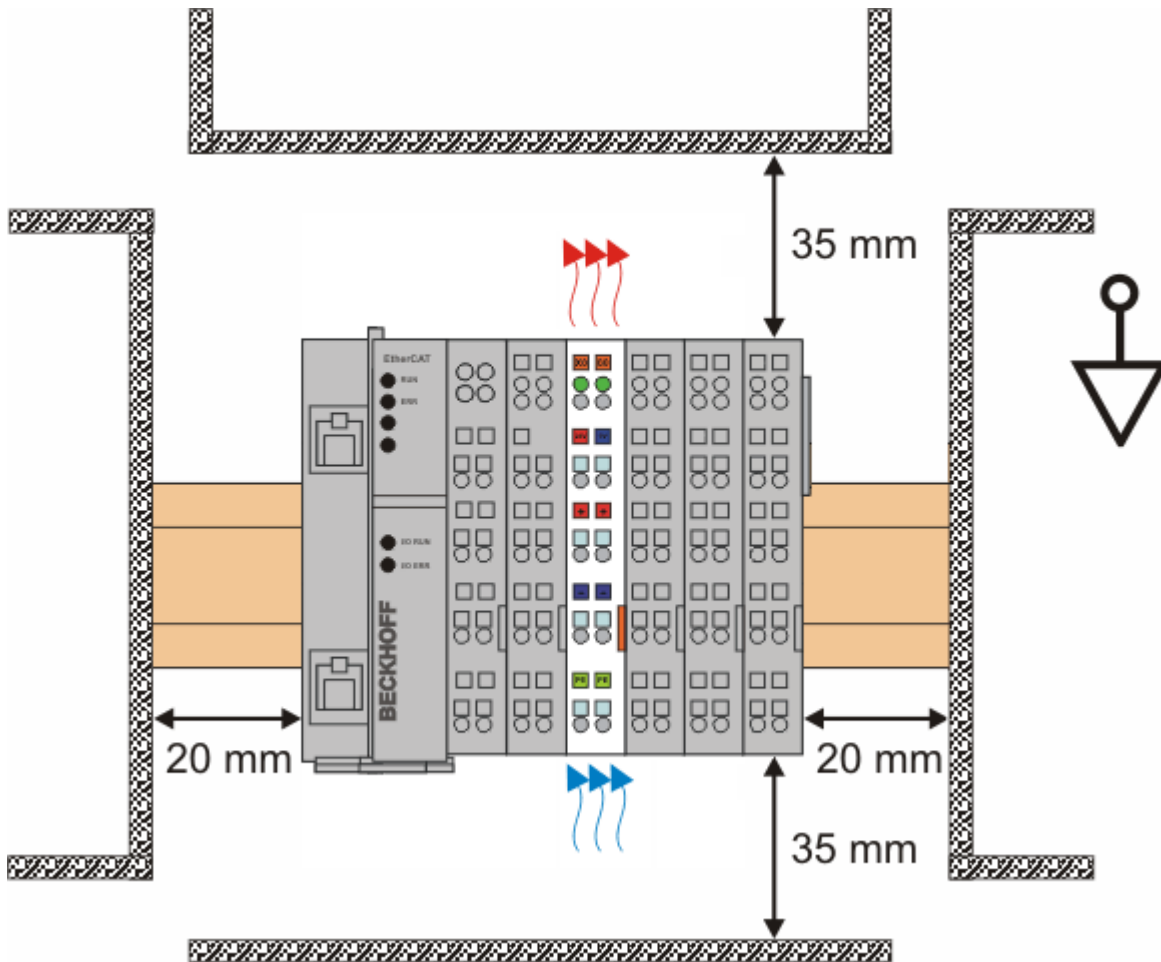


Fig. 23: Recommended distances for standard installation position

Compliance with the distances shown in Fig. *Recommended distances for standard installation position* is recommended.

#### Other installation positions

All other installation positions are characterized by different spatial arrangement of the mounting rail - see Fig *Other installation positions*.

The minimum distances to ambient specified above also apply to these installation positions.

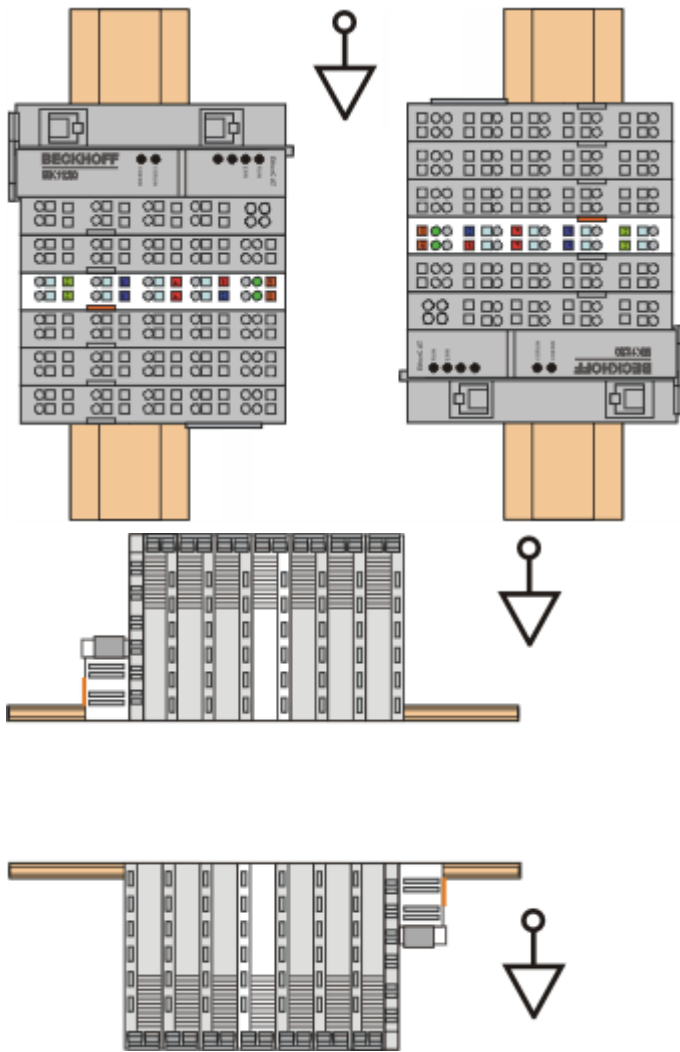


Fig. 24: Other installation positions

## 4.6 Positioning of passive Terminals

### **i** 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)

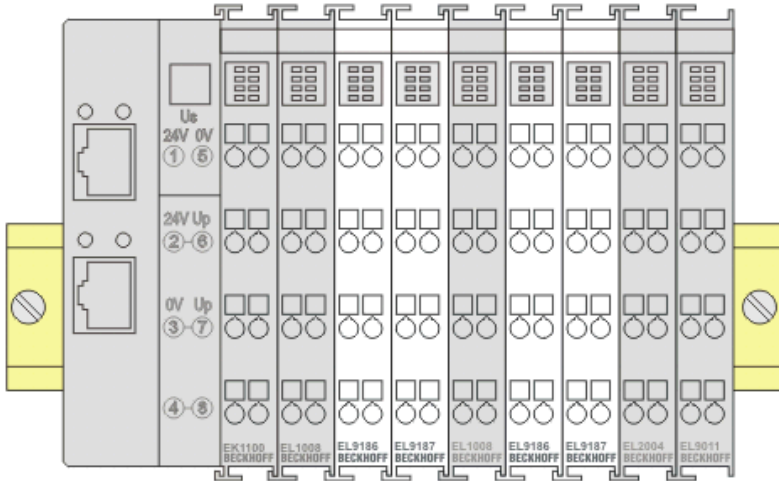


Fig. 25: Correct positioning

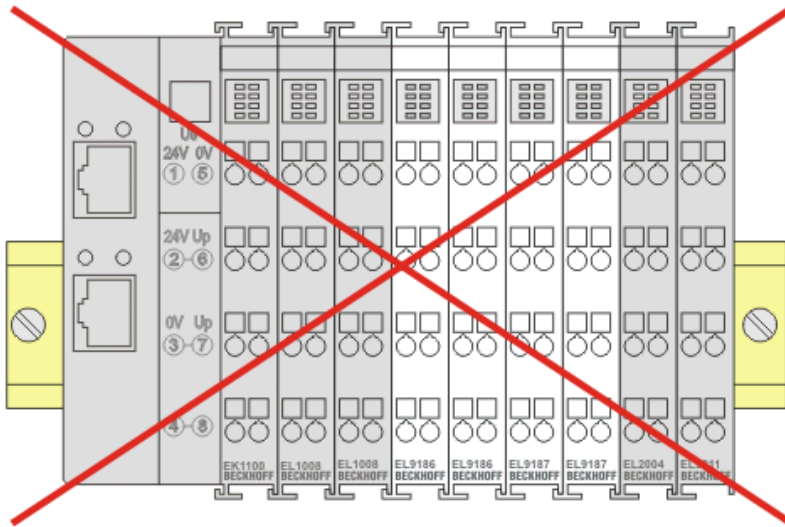


Fig. 26: Incorrect positioning

## 4.7 EL5122 - Connection

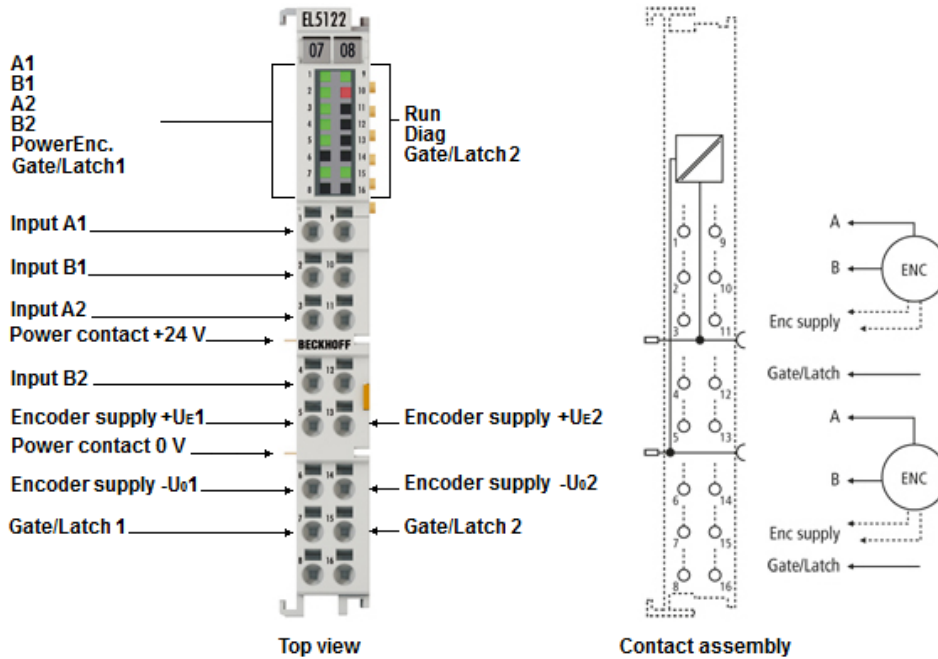


Fig. 27: EL5122 - Connection

Terminal point	No.	Comment
A1	1	Encoder input A1
B1	2	Encoder input B1
A2	3	Encoder input A2
B2	4	Encoder input B2
+Ue1	5	Encoder supply (5 V default, parameterizable 12 V, 24 V)
-Uo1	6	0 V encoder supply
Gate/Latch1	7	Gate/Latch combination input for encoder 1
n.c.	8	Do not connect the terminal point
n.c.	9	Do not connect the terminal point
n.c.	10	Do not connect the terminal point
n.c.	11	Do not connect the terminal point
n.c.	12	Do not connect the terminal point
+Ue2	13	Encoder supply (5 V default, parameterizable 12 V, 24 V)
-Uo2	14	0 V encoder supply
Gate/Latch2	15	Gate/Latch combination input for encoder 2
n.c.	16	Do not connect the terminal point



### Setting the encoder supply via index [0x8001:17](#) [[▶ 171](#)]

The encoder supply is set centrally for both channels via the index [0x8001:17](#) [[▶ 171](#)] (channel 1). The corresponding index [0x8011:17](#) of the second channel has no parameterization function.

### NOTICE

#### Setting the encoder supply voltage

- Before switching to a higher voltage, make sure that the connected encoders support the selected voltage range!
- To write to [0x80n1:17](#) "Supply voltage" you have to set the value [0x72657375](#) (ASCII: "user") in index [0xF008](#) [[▶ 190](#)] "Code word".

### 4.7.1 TTL mode

#### Connection of TTL encoders without zero pulse

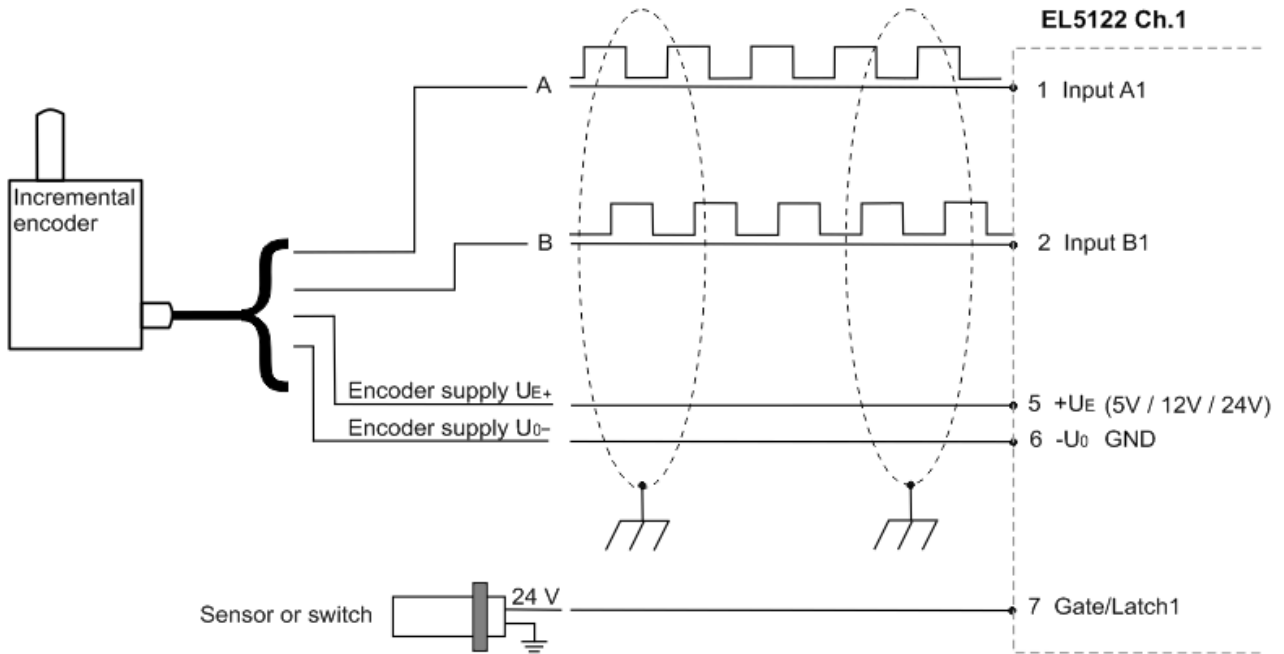


Fig. 28: Connection of TTL encoder without zero pulse

#### Connection of TTL counters / pulse generators without zero pulse

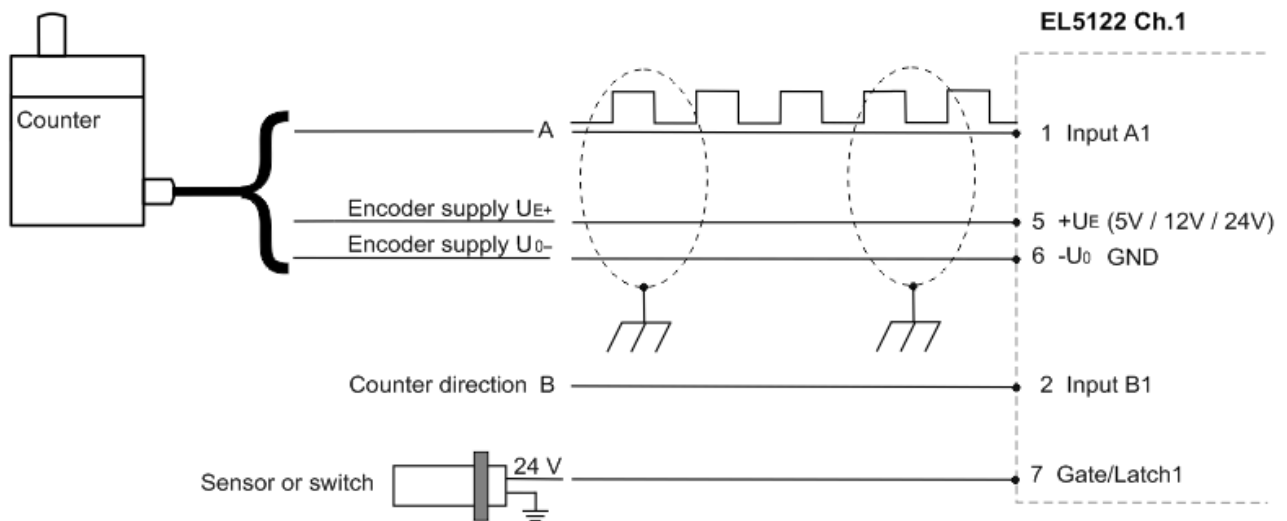


Fig. 29: Connection of counter / TTL pulse generator – mode without zero pulse



#### Connection instructions

- If a counter / pulse generator is connected, the B track determines the counting direction.



### 4.7.2 Open Collector mode

#### Connection of Open Collector encoders without zero pulse

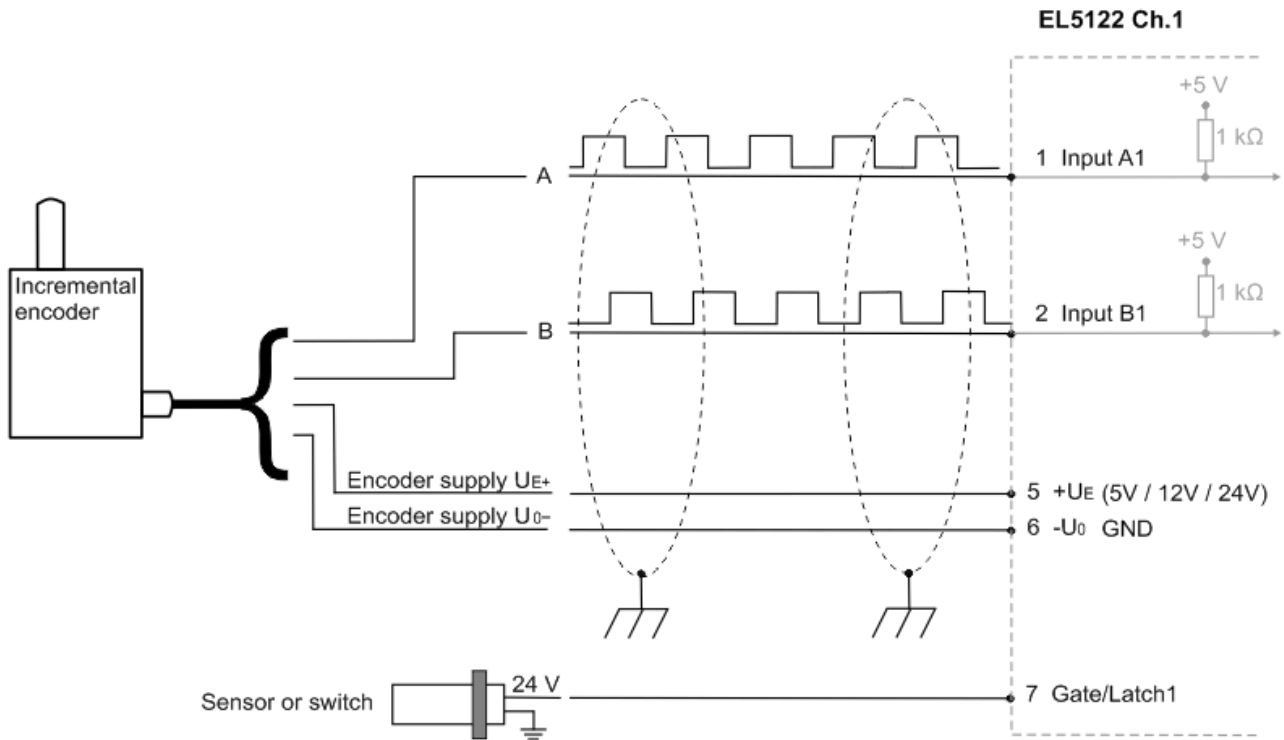


Fig. 30: Connection of Open Collector encoder - mode without zero pulse

#### Connection of Open Collector counters / pulse generators without zero pulse

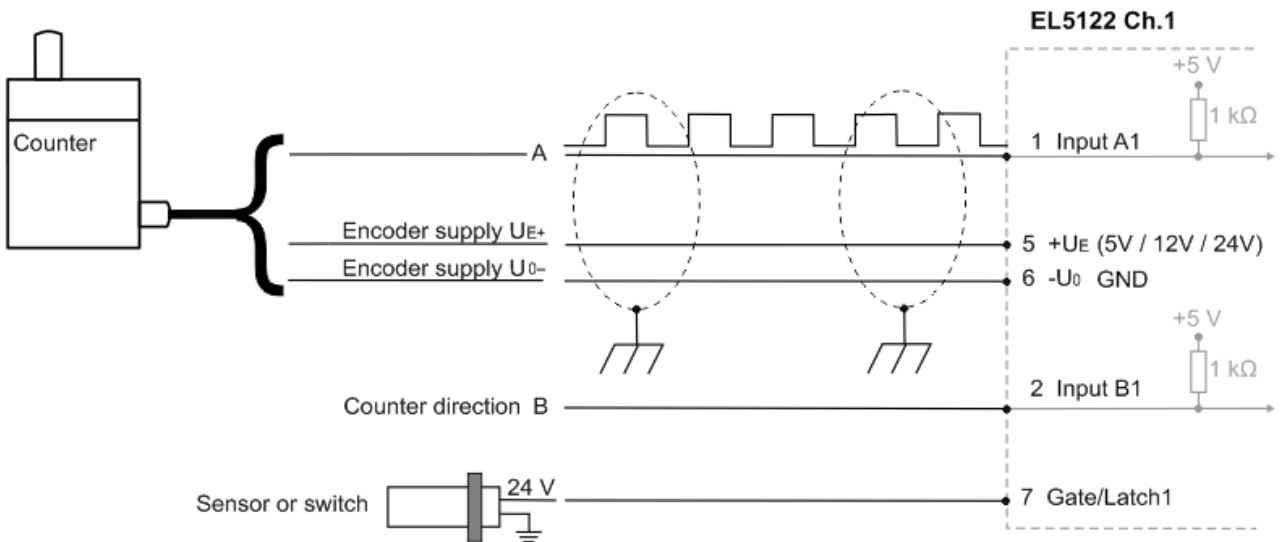


Fig. 31: Connection of counter / Open Collector pulse generator – mode without zero pulse



#### Connection instructions

- If a counter / pulse generator is connected, the B track determines the counting direction.

## 4.8 EL5122 - LEDs

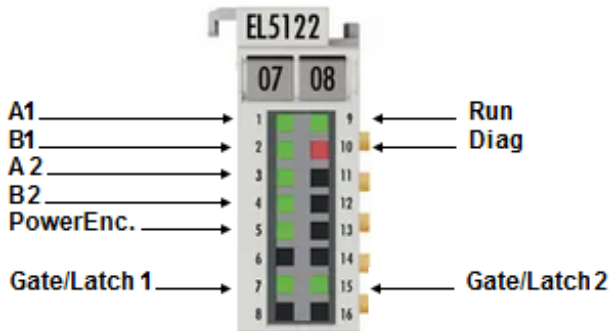


Fig. 32: EL5122 - LEDs

No.	Name	Color	Description	
1	A1	green	indicates TRUE level	
2	B1			
3	A2			
4	B2			
5	PowerEnc.	green	Operating voltage display for incremental encoder power supply	
7	Gate/Latch 1	green	is lit if a signal (+24 V) is present at the gate/latch input	
9	Run	green	This LED indicates the terminal's operating state:	
			off	State of the <a href="#">EtherCAT State Machine [► 28]</a> : <b>INIT</b> = initialization of the terminal or <b>BOOTSTRAP</b> = function for <a href="#">firmware updates [► 191]</a> of the terminal
			flashing	State of the EtherCAT State Machine: <b>PREOP</b> = function for mailbox communication and different default settings set
			Single flash	State of the EtherCAT State Machine: <b>SAFEOP</b> = verification of the <a href="#">Sync Manager [► 102]</a> channels and the distributed clocks. Outputs remain in safe state
	on	State of the EtherCAT State Machine: <b>OP</b> = normal operating state; mailbox and process data communication is possible		
10	Diag	red	Initialization process active or state of the EtherCAT State Machine: <b>BOOT</b>	
15	Gate/Latch 2	green	is lit if a signal (+24 V) is present at the gate/latch input	

## 4.9 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.

## 5 Commissioning

### 5.1 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:

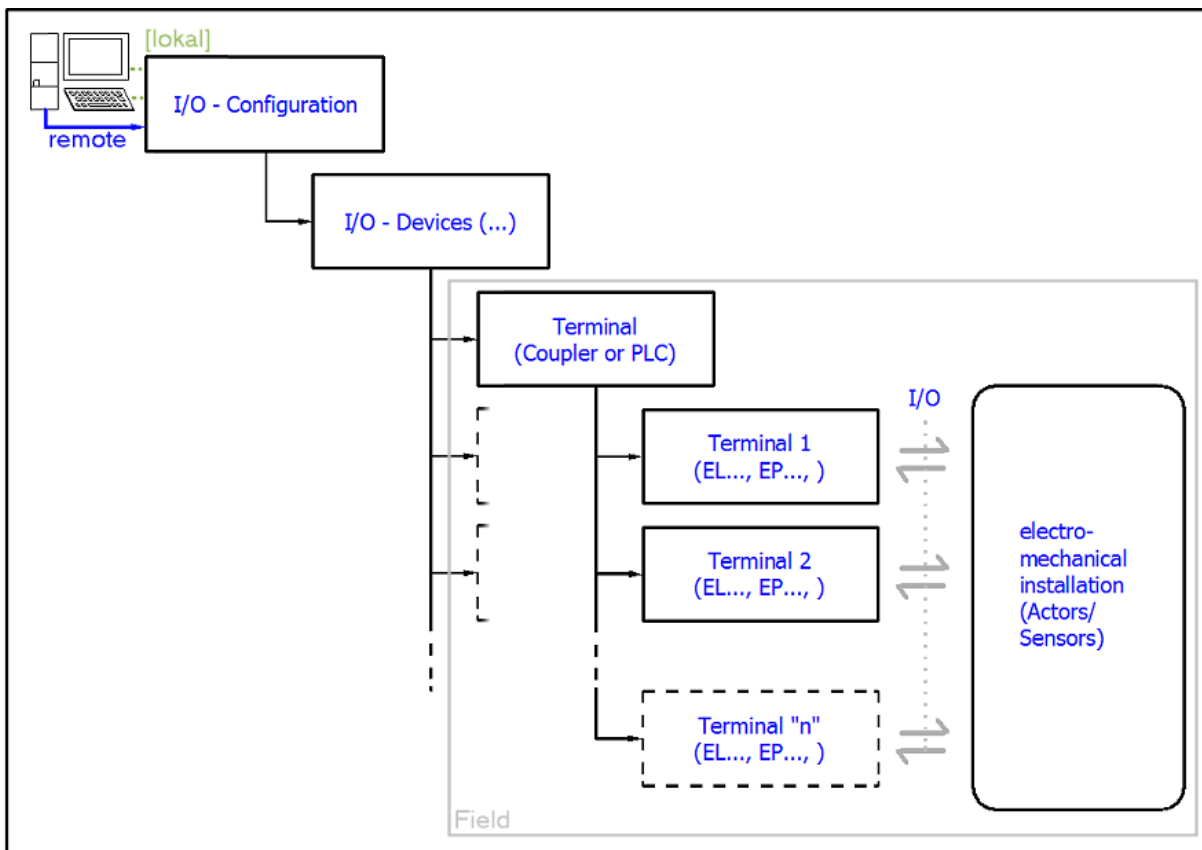


Fig. 33: Relationship between user side (commissioning) and installation

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<sub>DC</sub>)
- 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<sub>DC</sub>; 0.5 A)
- (Optional via X000: a link to an external PC for the user interface)



## 5.1.1 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:

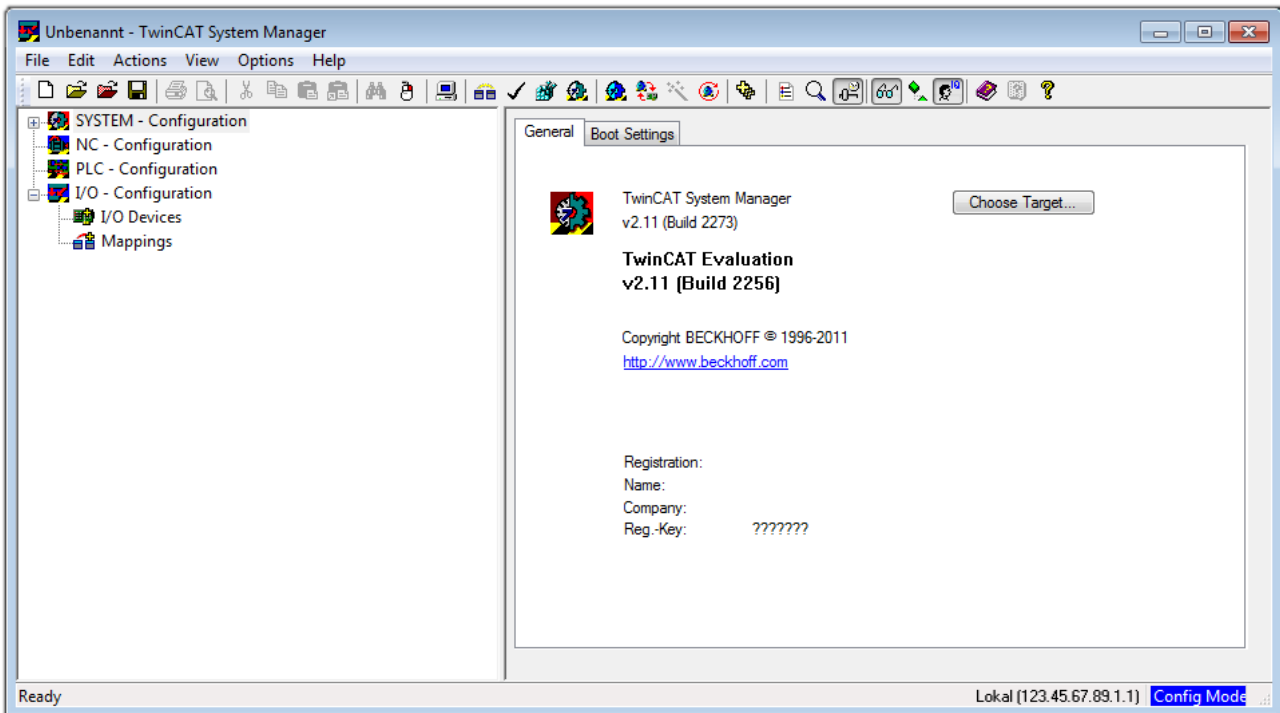



Fig. 35: Initial TwinCAT 2 user interface

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](#) [▶ 56]”.

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:

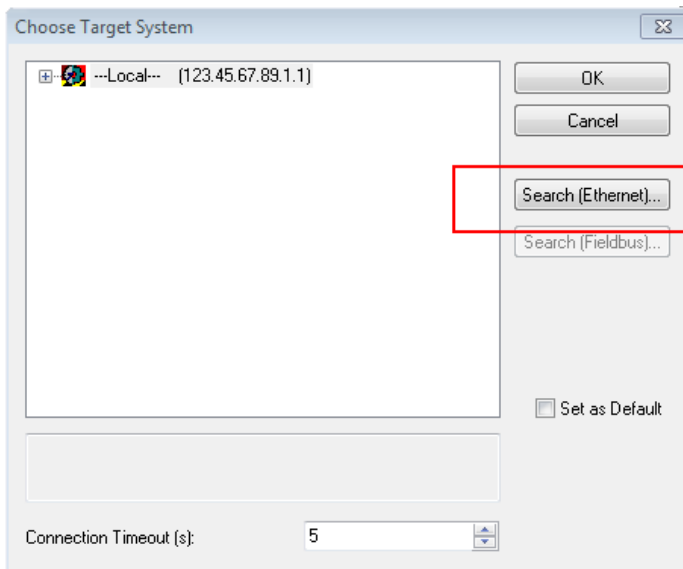


Fig. 36: Selection of the target system

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

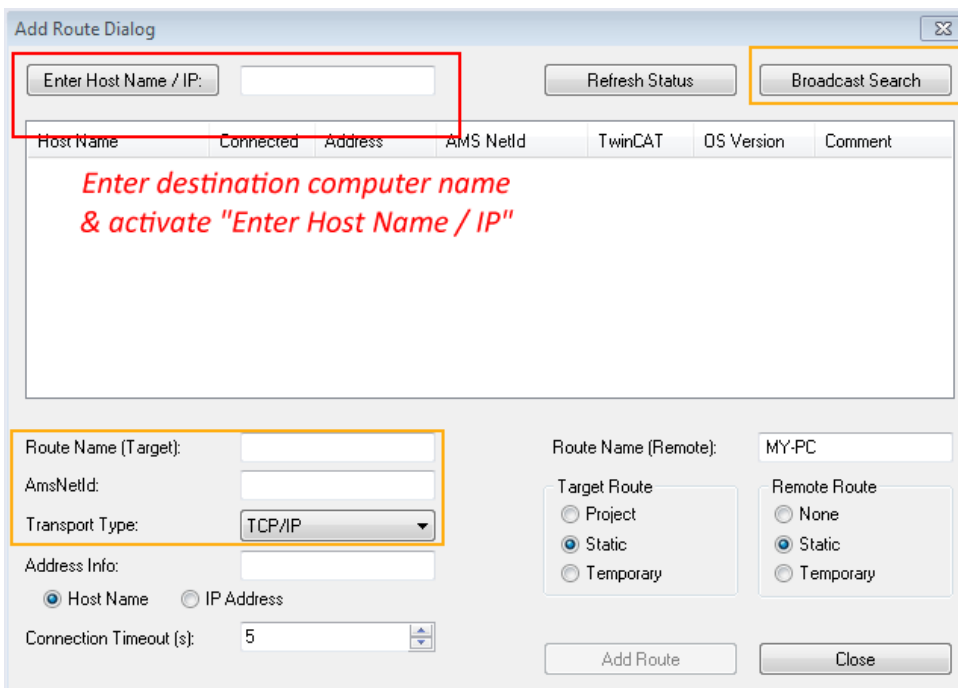
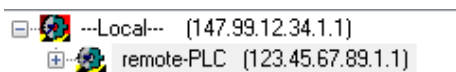


Fig. 37: specify the PLC for access by the TwinCAT System Manager: selection of the target system



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).

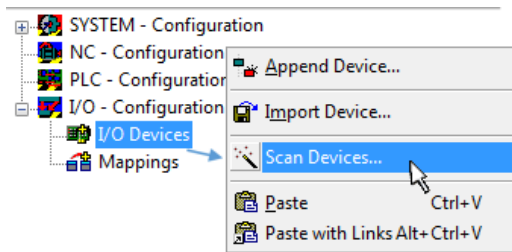


Fig. 38: Select “Scan Devices...”

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

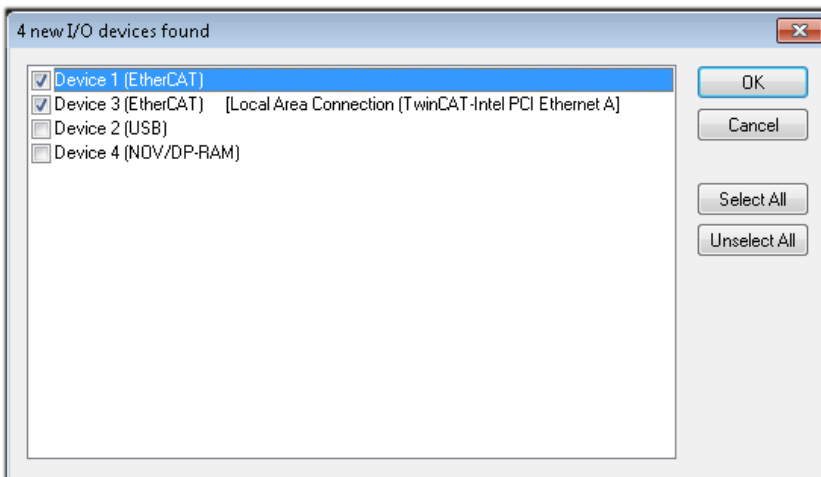


Fig. 39: Automatic detection of I/O devices: selection of the devices to be integrated

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 \[▶ 52\]](#) described at the beginning of this section, the result is as follows:



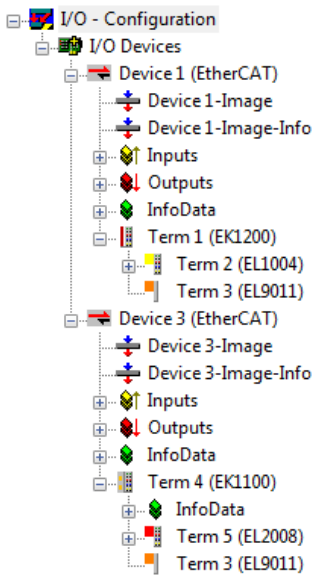


Fig. 40: Mapping of the configuration in the TwinCAT 2 System Manager

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:

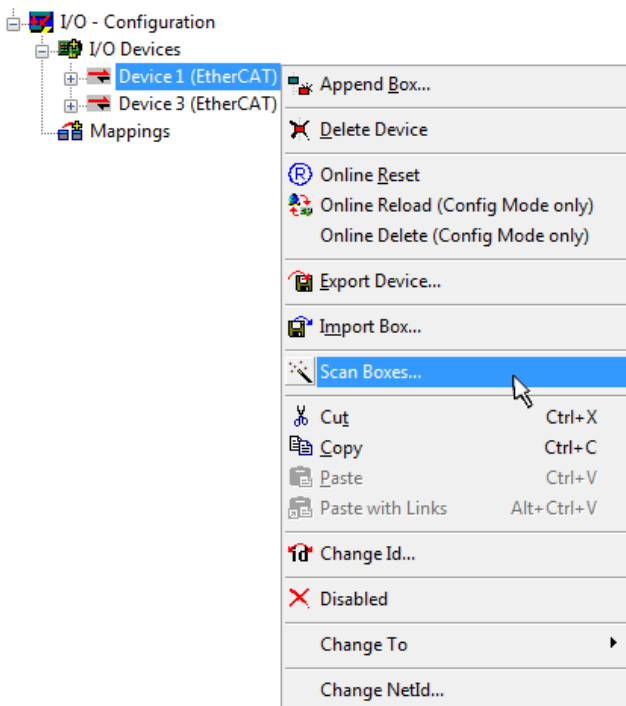


Fig. 41: Reading of individual terminals connected to a device

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:

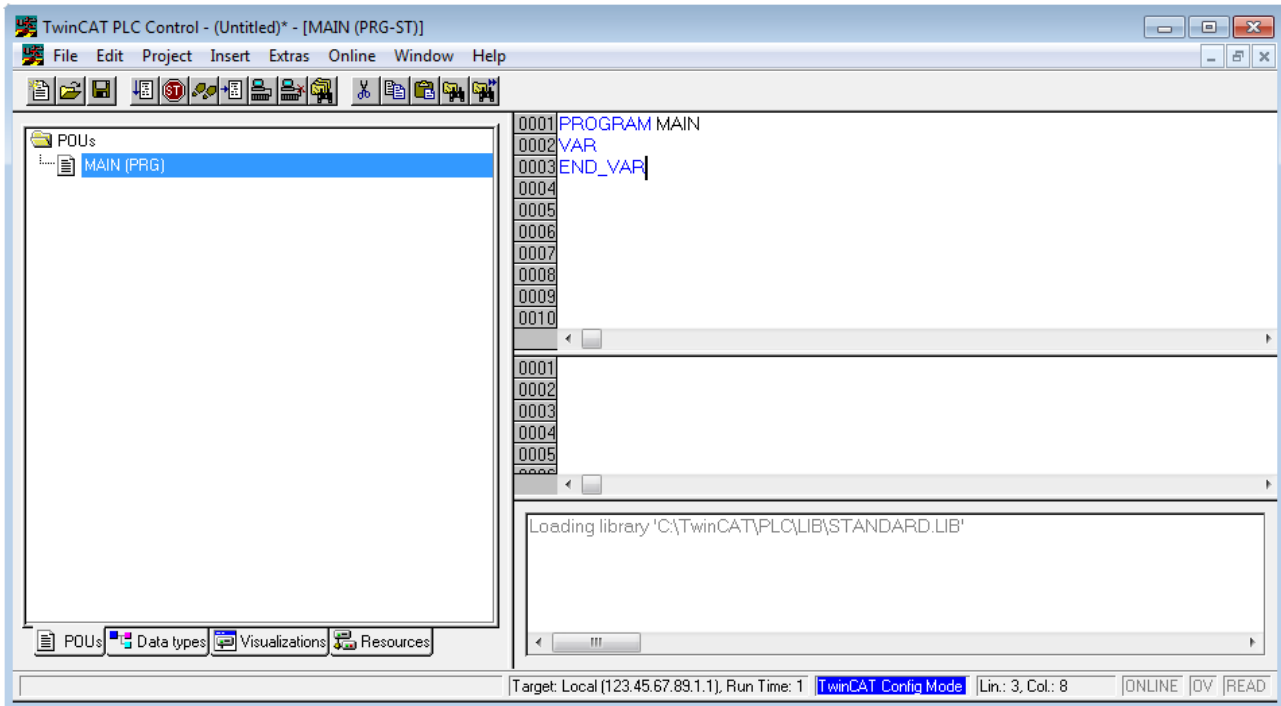


Fig. 42: TwinCAT PLC Control after startup

Example variables and an example program have been created and stored under the name "PLC\_example.pro":

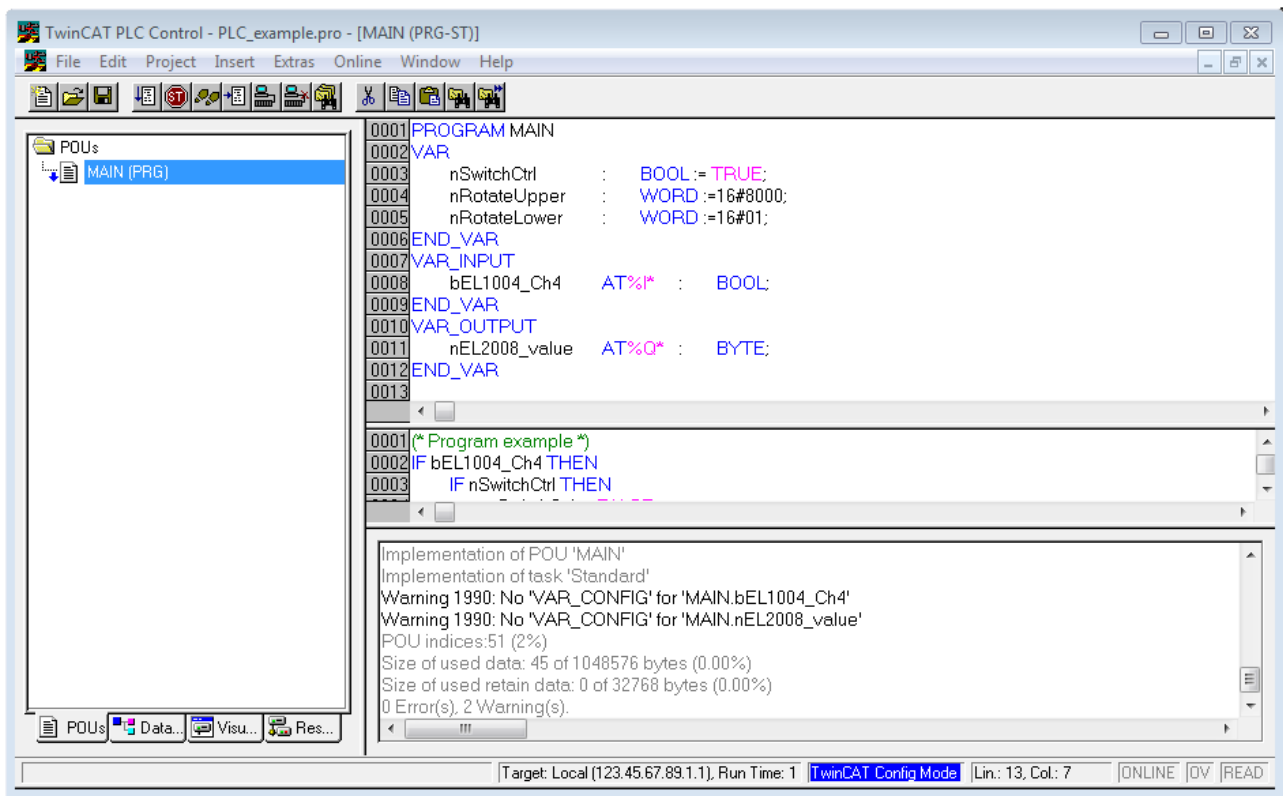


Fig. 43: Example program with variables after a compile process (without variable integration)

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...”:

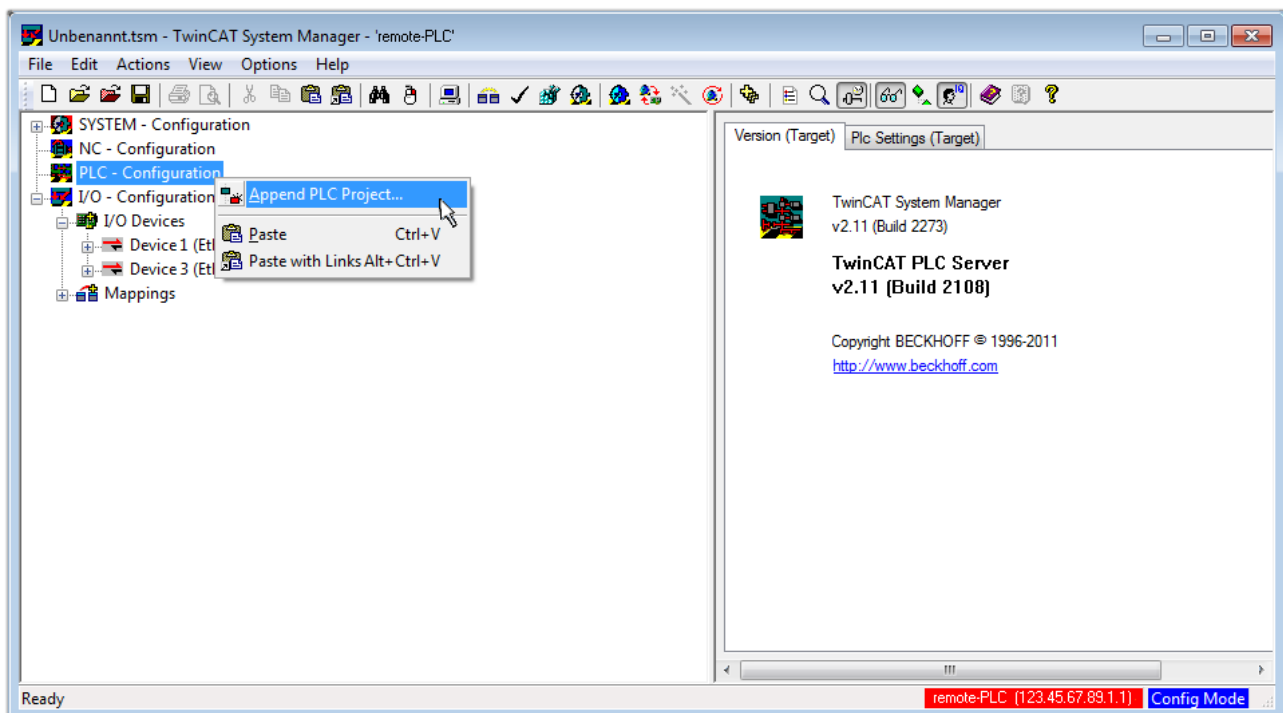


Fig. 44: Appending the TwinCAT PLC Control project

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:

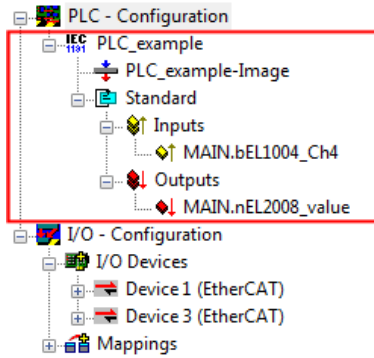


Fig. 45: PLC project integrated in the PLC configuration of the System Manager

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”:

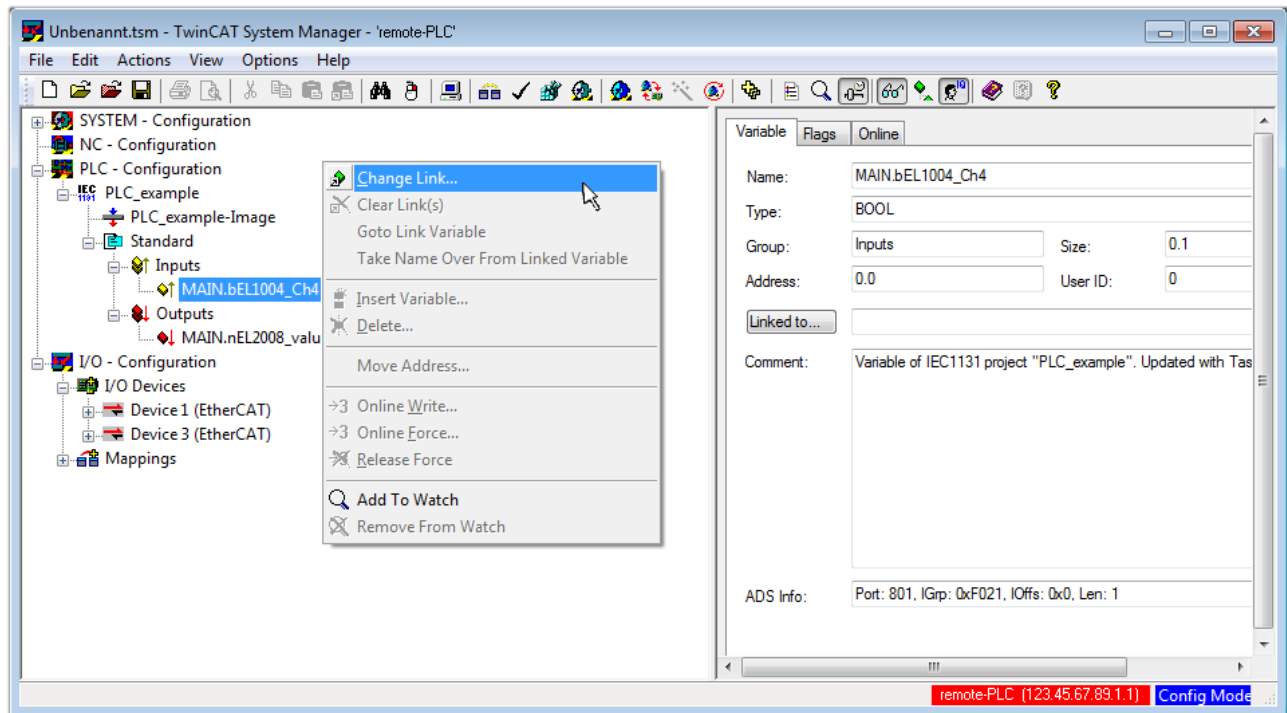


Fig. 46: Creating the links between PLC variables and process objects

In the window that opens, the process object for the “bEL1004\_Ch4” BOOL-type variable can be selected from the PLC configuration tree:

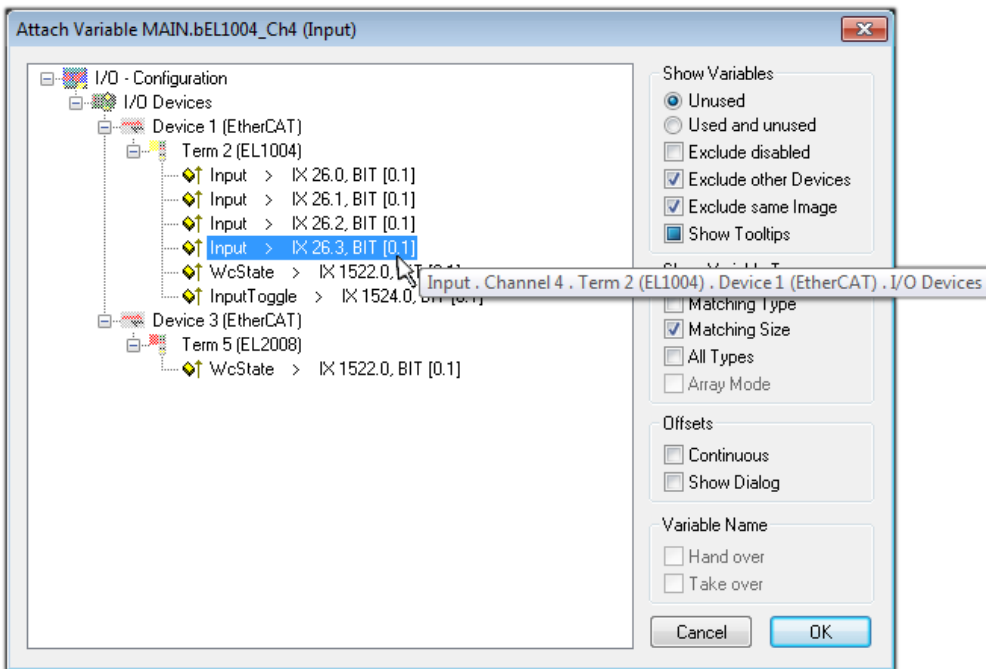


Fig. 47: Selecting BOOL-type PDO

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:

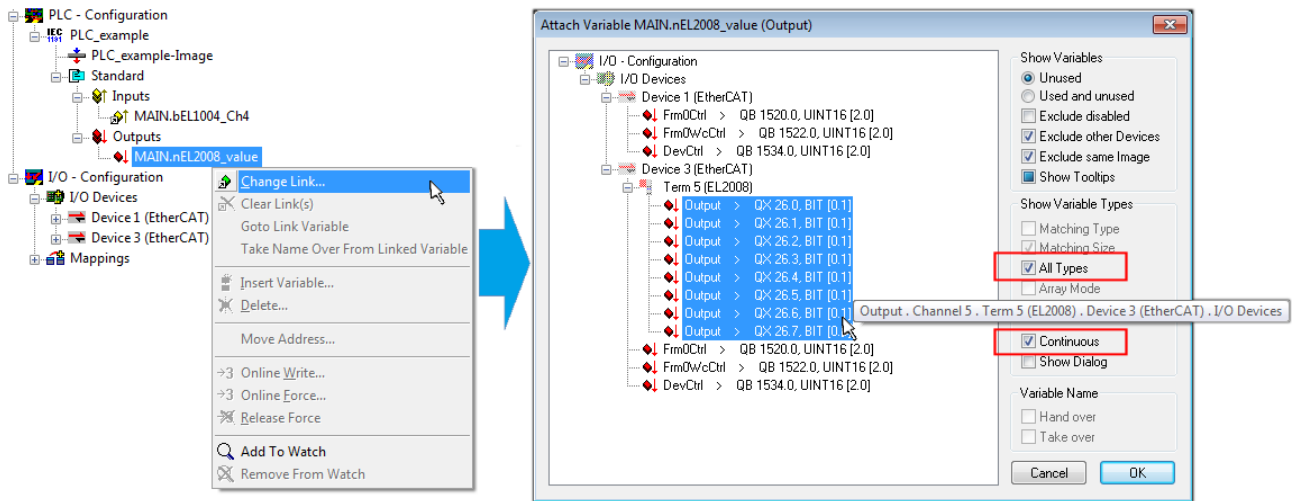



Fig. 48: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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:

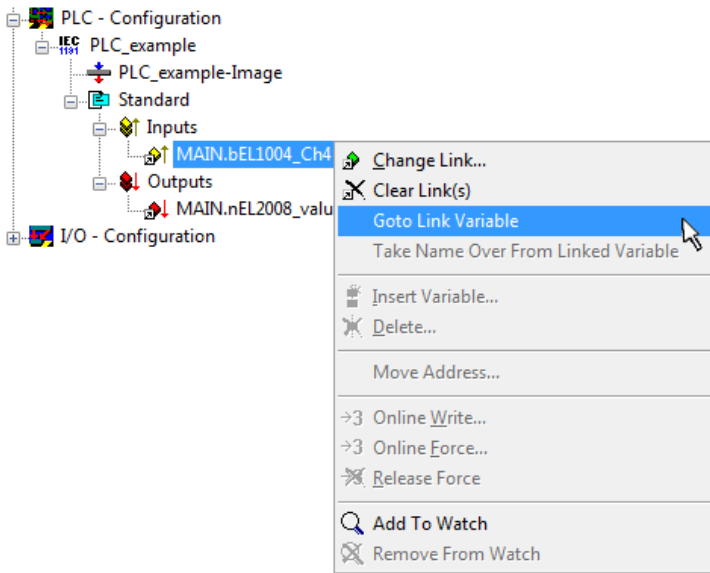

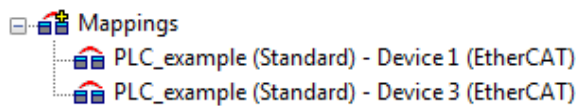


Fig. 49: Application of a “Goto Link Variable”, using “MAIN.bEL1004\_Ch4” as an example

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 RTime 0% 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...”:

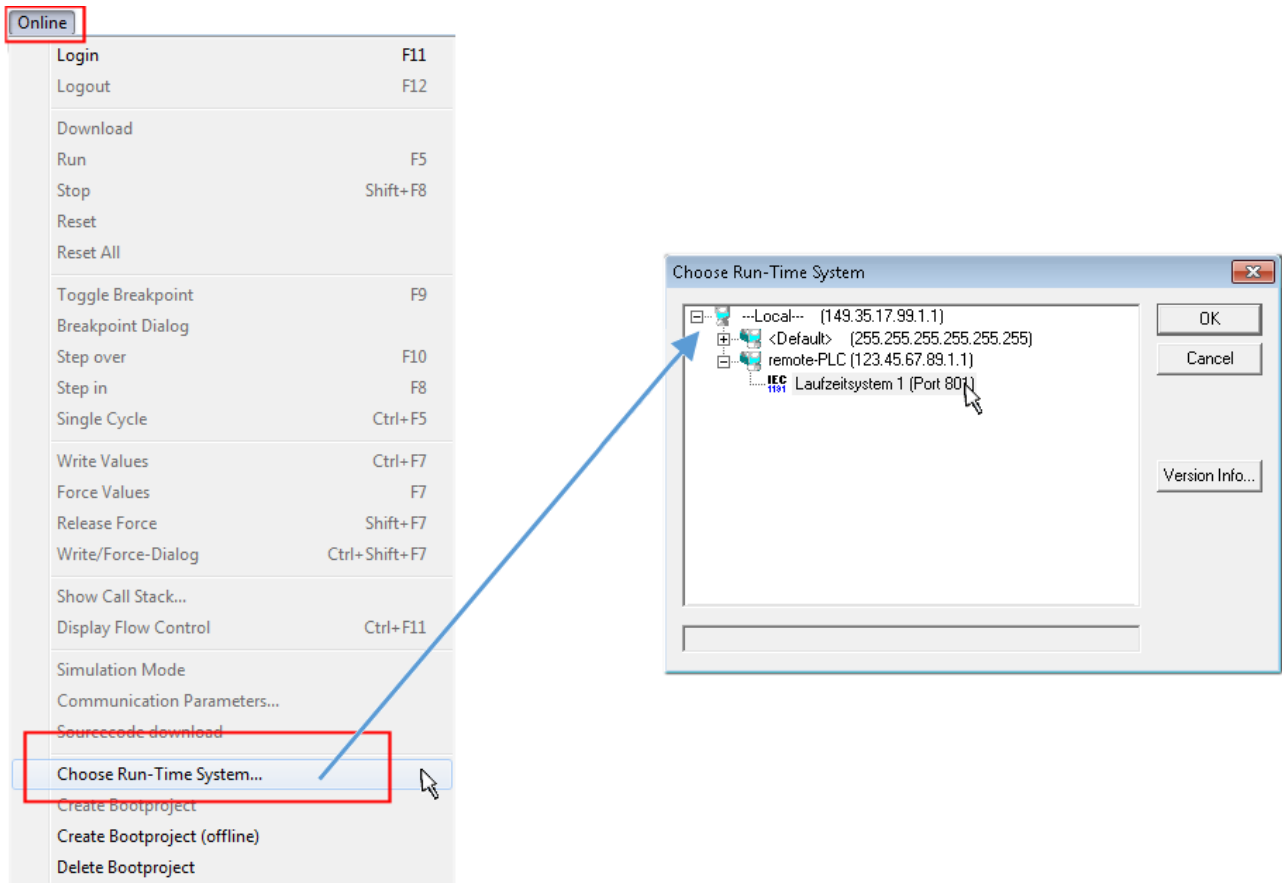



Fig. 50: Choose target system (remote)

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:

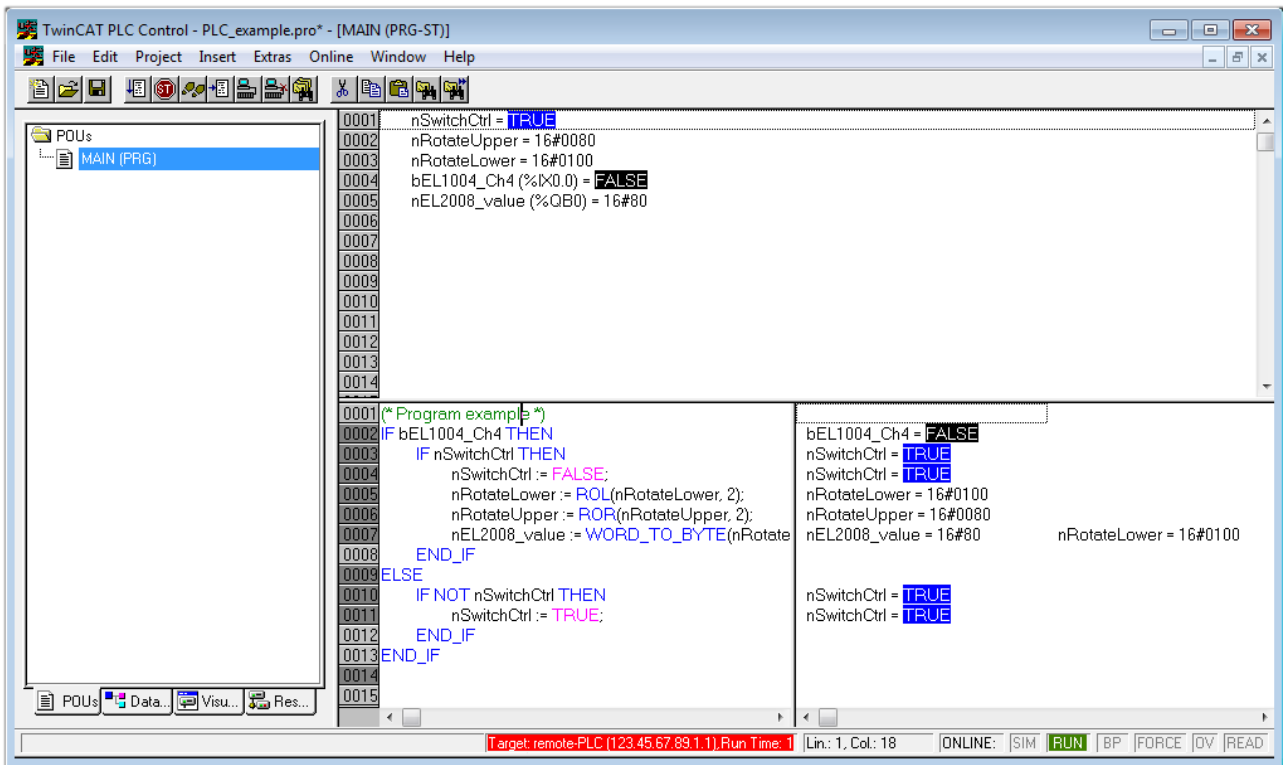


Fig. 51: PLC Control logged in, ready for program startup

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

## 5.1.2 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. 52: 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):

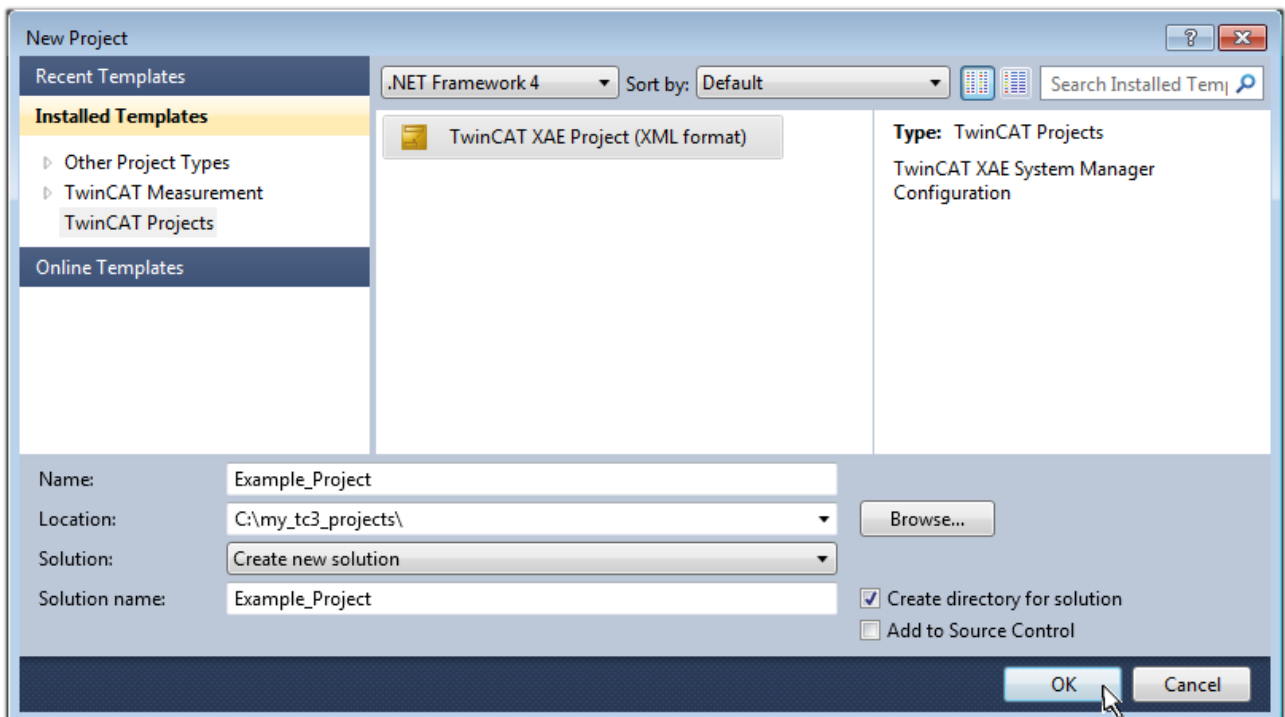


Fig. 53: Create new TwinCAT 3 project

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

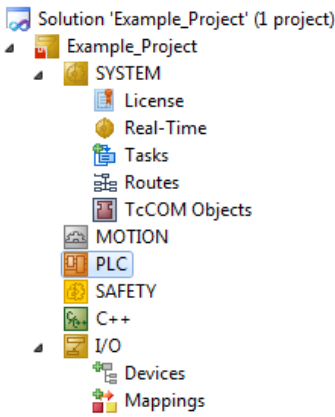
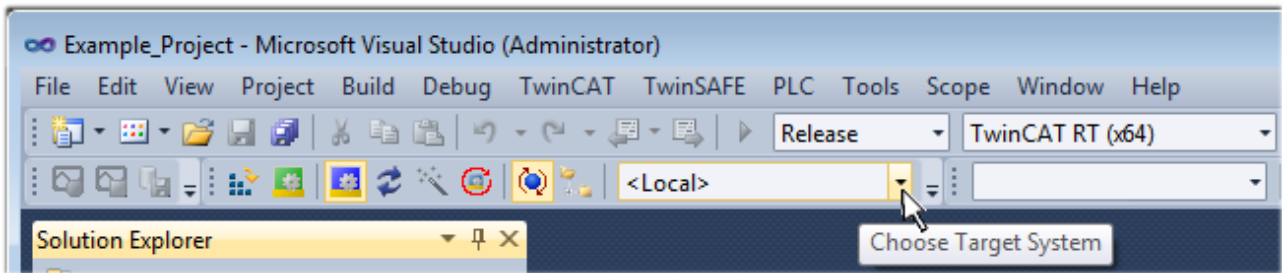


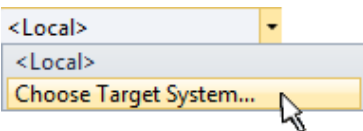
Fig. 54: New TwinCAT 3 project in the project folder explorer

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 [▶ 67]”.

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:

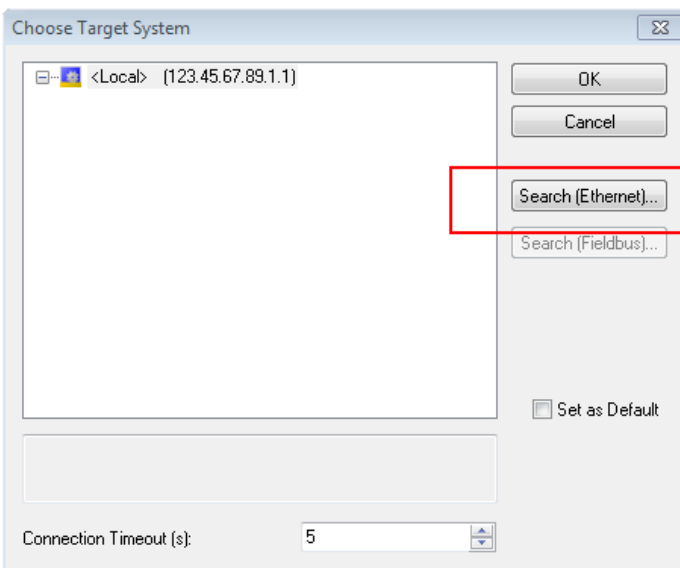


Fig. 55: Selection dialog: Choose the target system

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

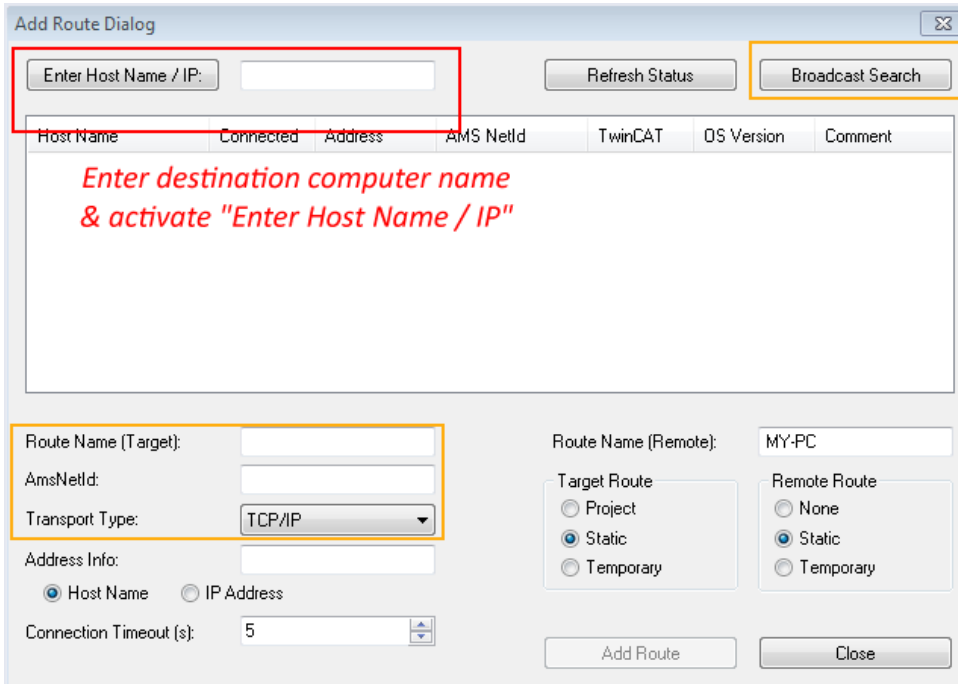
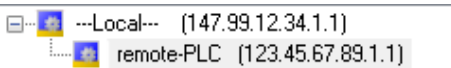


Fig. 56: specify the PLC for access by the TwinCAT System Manager: selection of the target system


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)”.

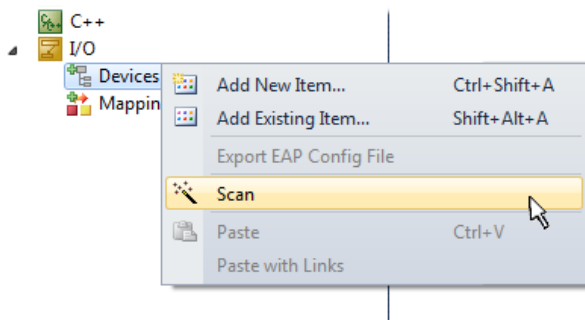


Fig. 57: Select “Scan”

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

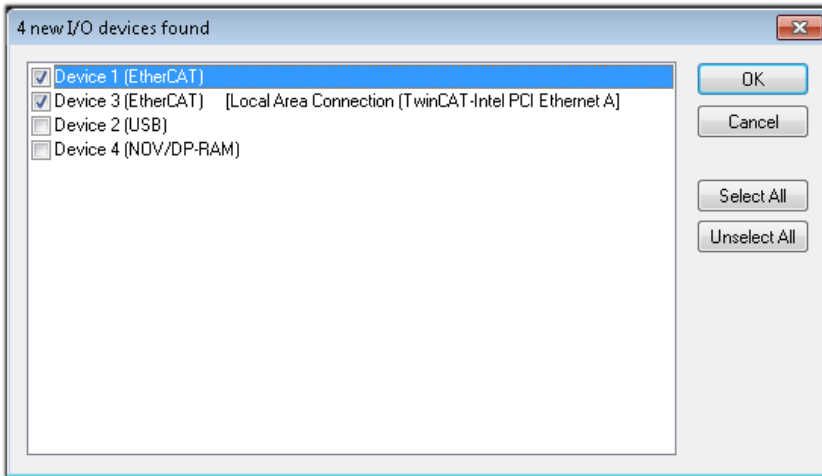


Fig. 58: Automatic detection of I/O devices: selection of the devices to be integrated

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 \[▶ 52\]](#) described at the beginning of this section, the result is as follows:

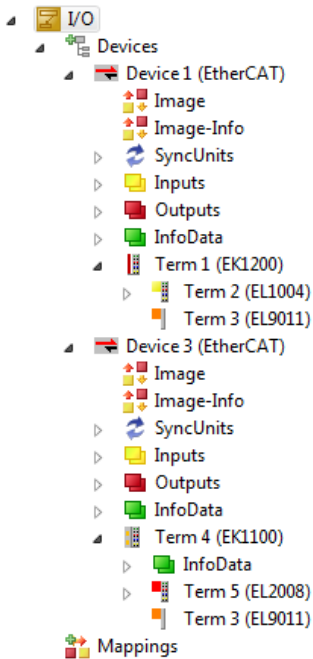


Fig. 59: Mapping of the configuration in VS shell of the TwinCAT 3 environment

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:

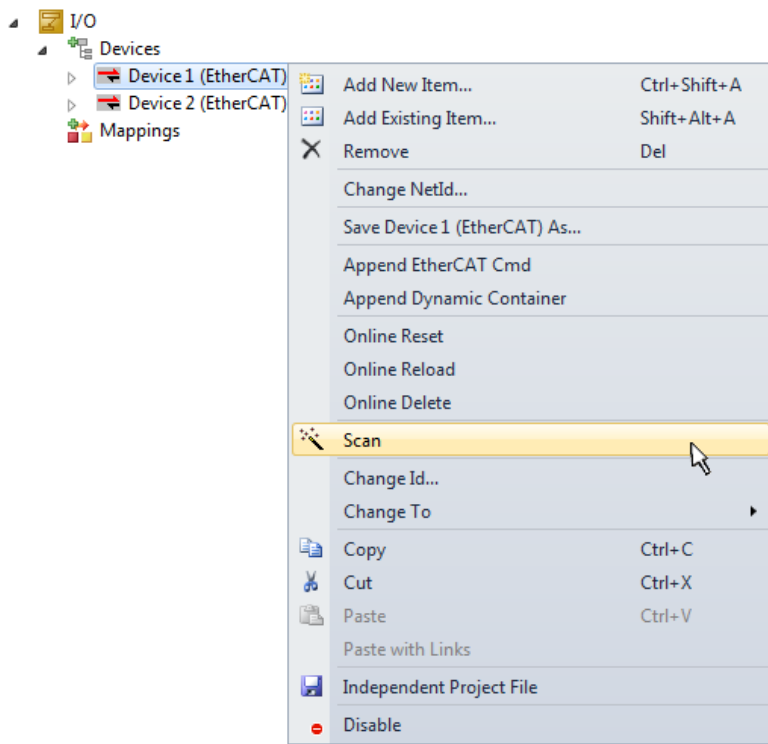


Fig. 60: Reading of individual terminals connected to a device

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....":

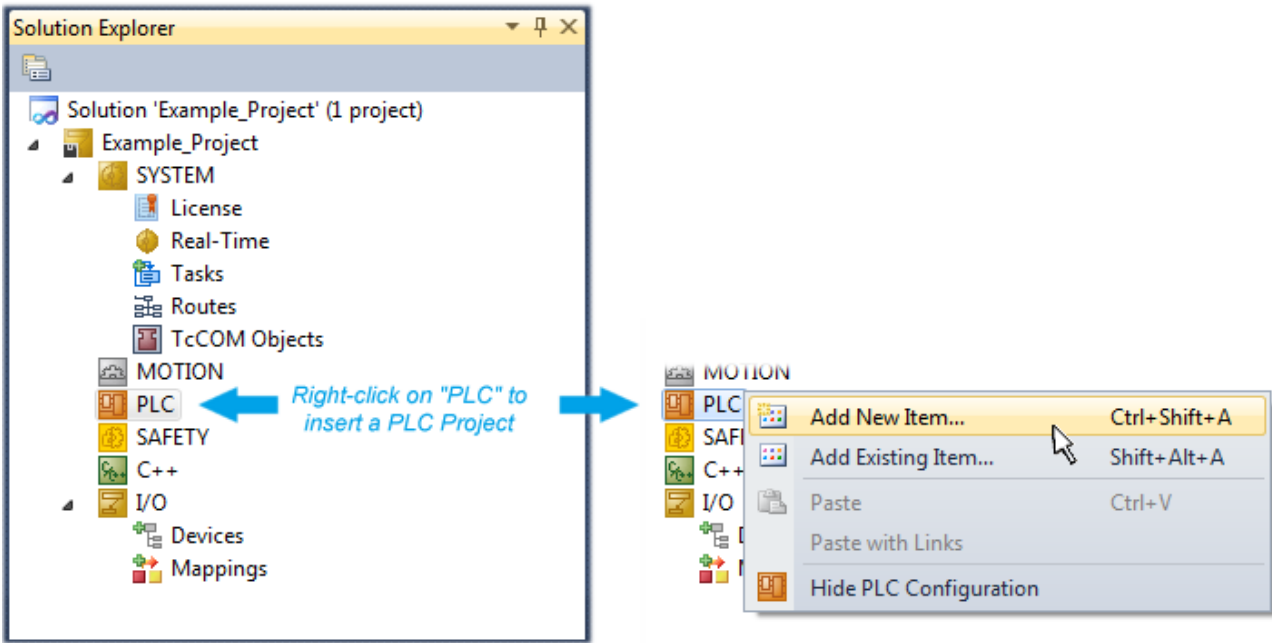


Fig. 61: Adding the programming environment in “PLC”

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

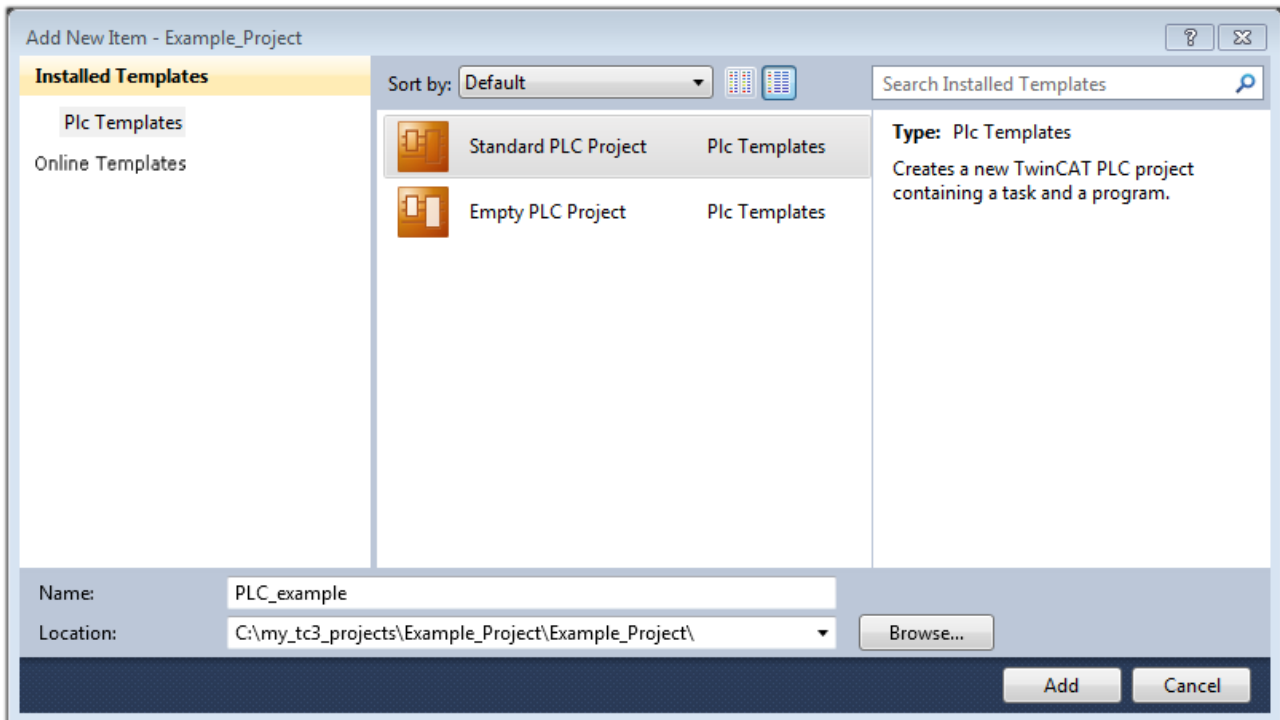


Fig. 62: Specifying the name and directory for the PLC programming environment

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:

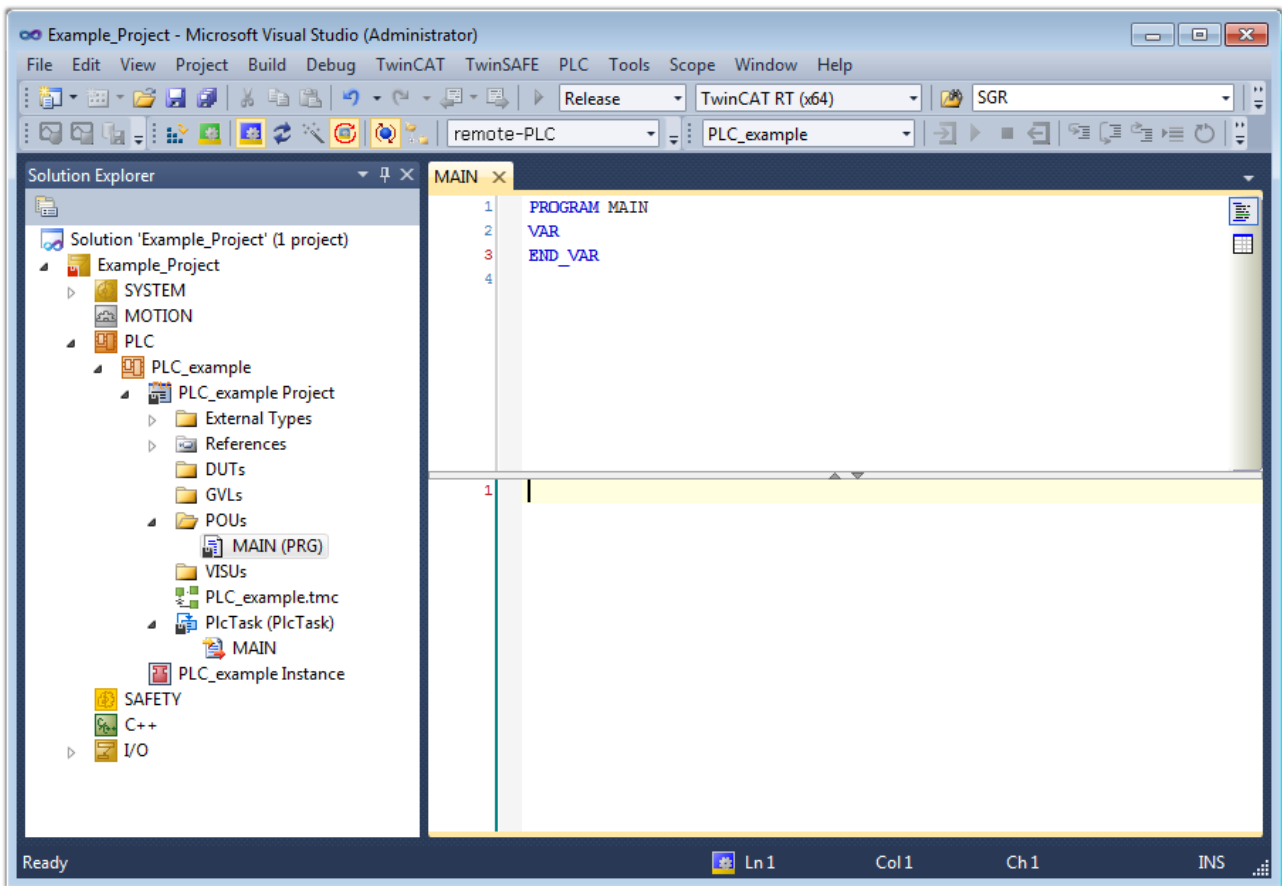


Fig. 63: Initial “Main” program for the standard PLC project

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

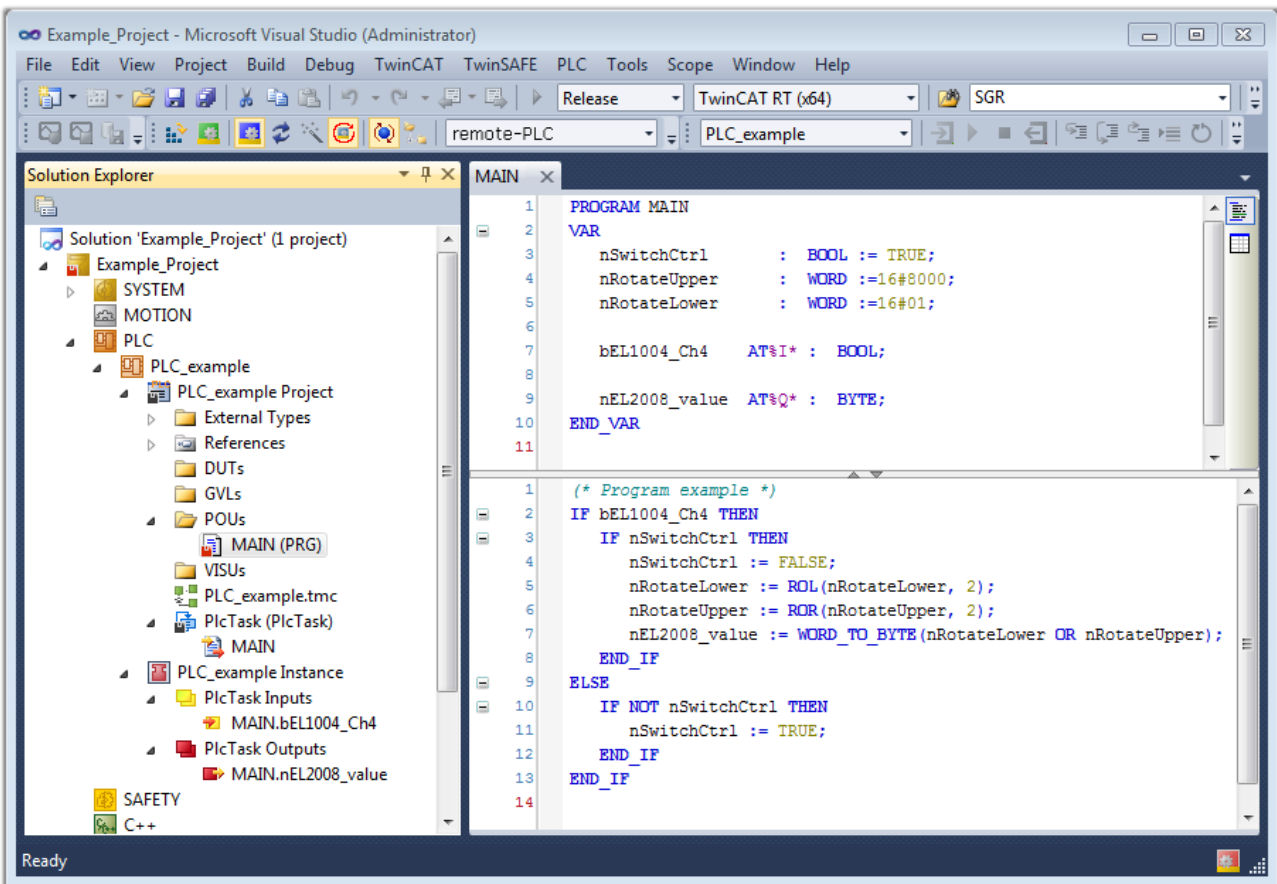


Fig. 64: Example program with variables after a compile process (without variable integration)

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

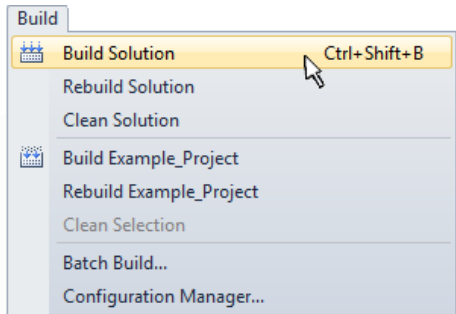
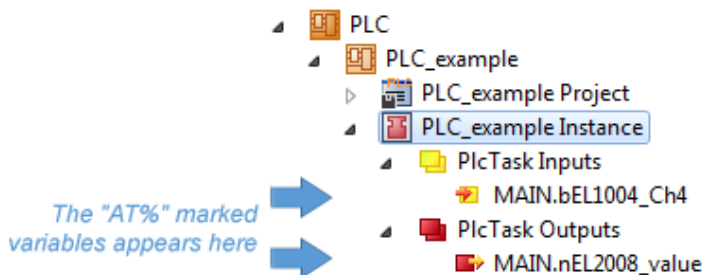


Fig. 65: Start program compilation

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:



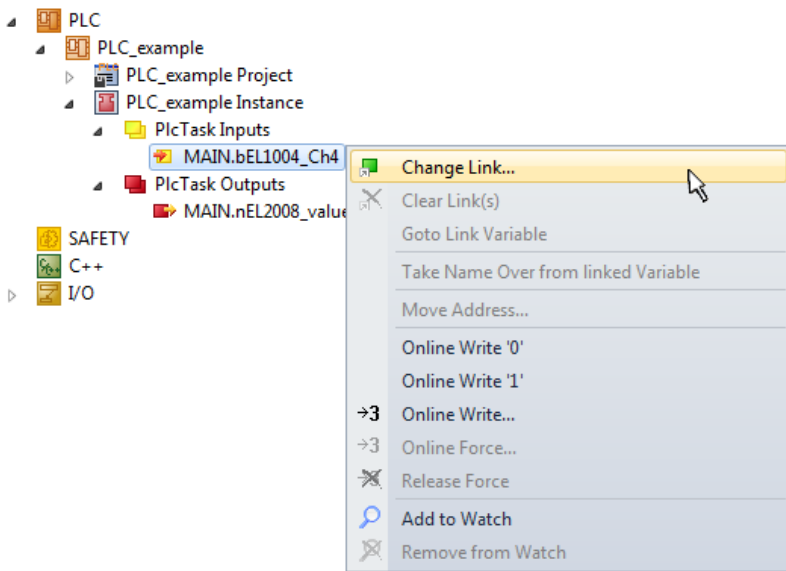


Fig. 66: Creating the links between PLC variables and process objects

In the window that opens, the process object for the “bEL1004\_Ch4” BOOL-type variable can be selected from the PLC configuration tree:

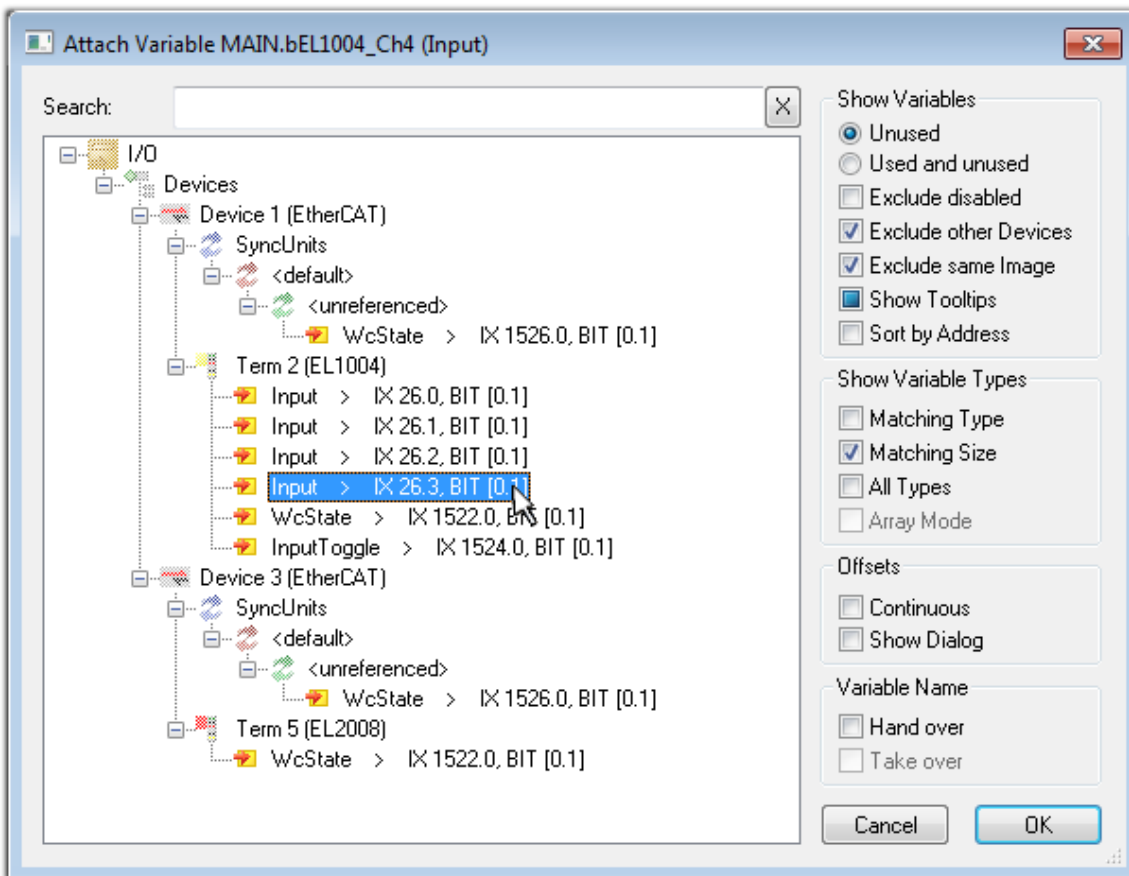


Fig. 67: Selecting BOOL-type PDO

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:

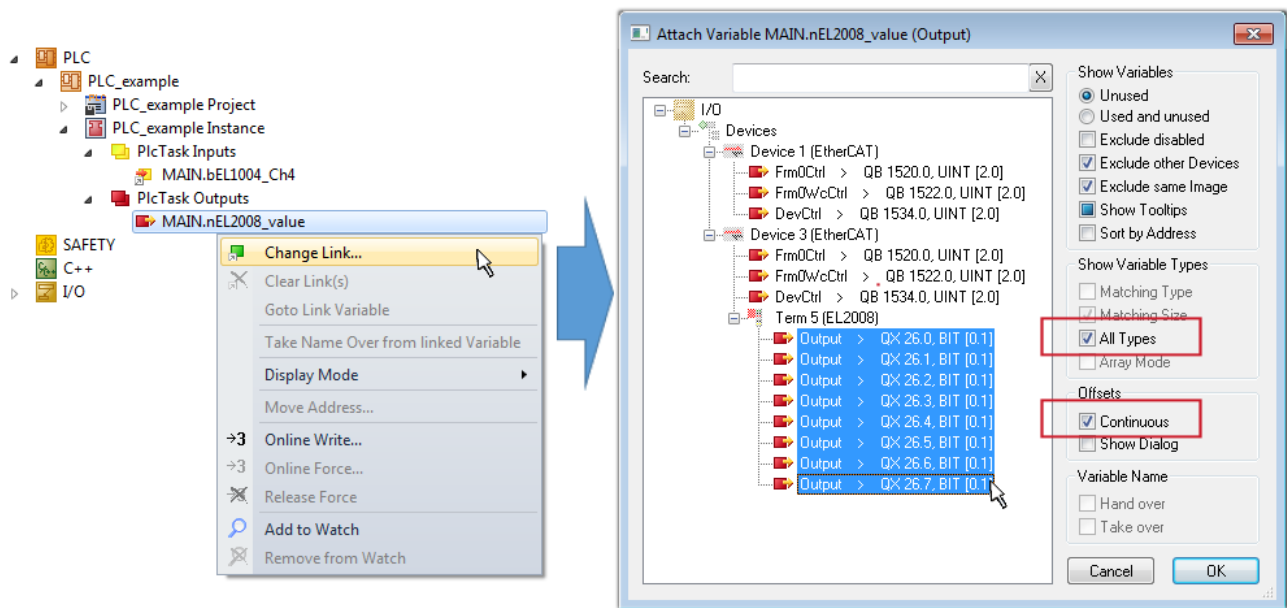



Fig. 68: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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:

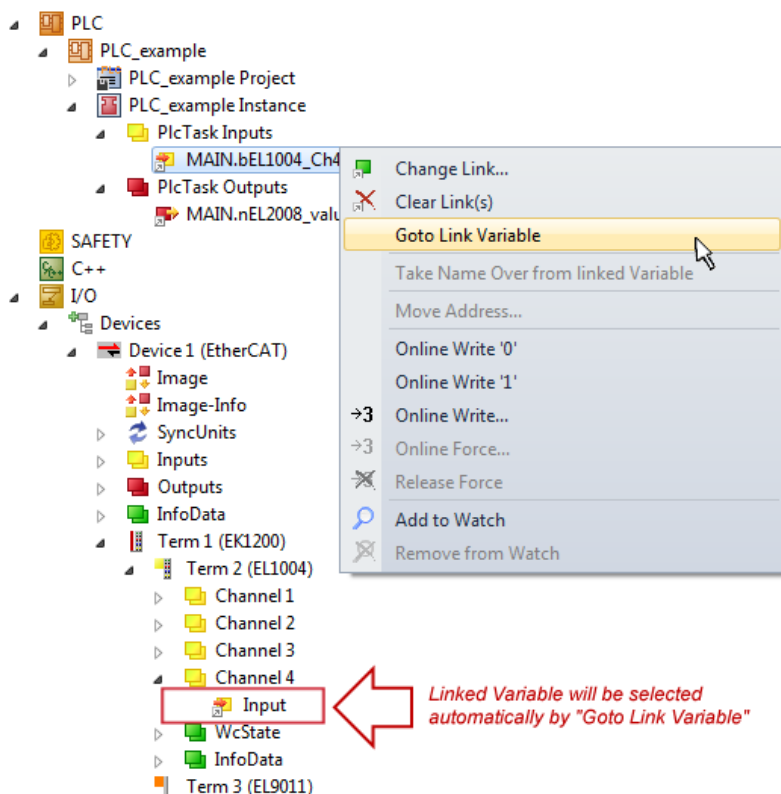


Fig. 69: Application of a “Goto Link Variable”, using “MAIN.bEL1004\_Ch4” as an example

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**

**i** 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.

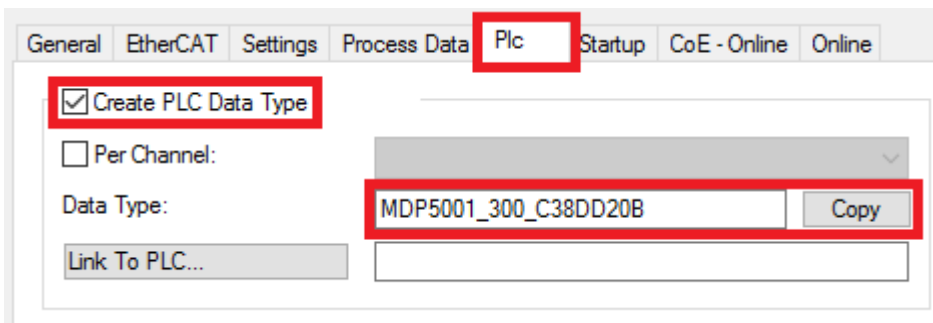


Fig. 70: Creating a PLC data type

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

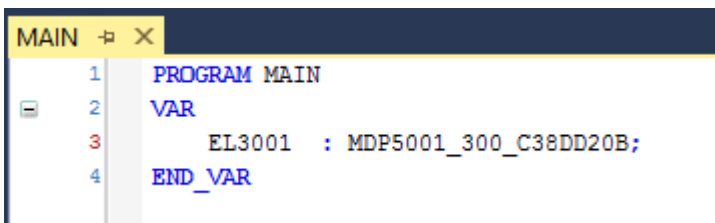


Fig. 71: Instance\_of\_struct

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.

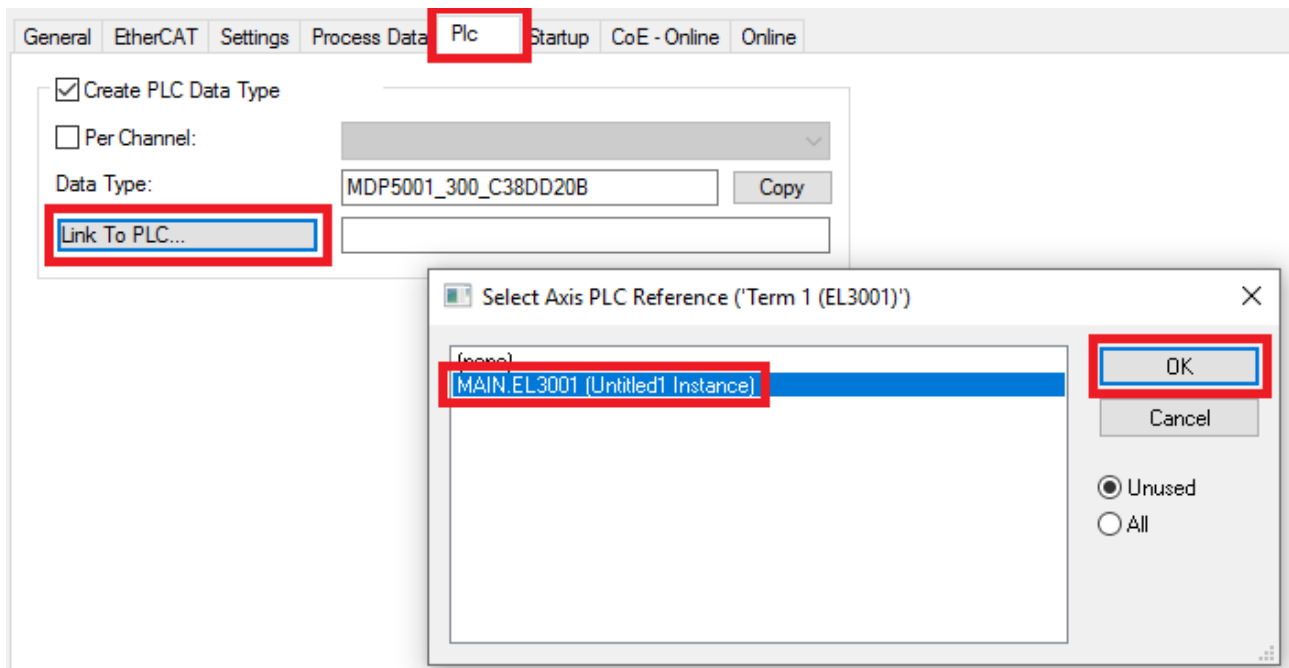


Fig. 72: Linking the structure

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

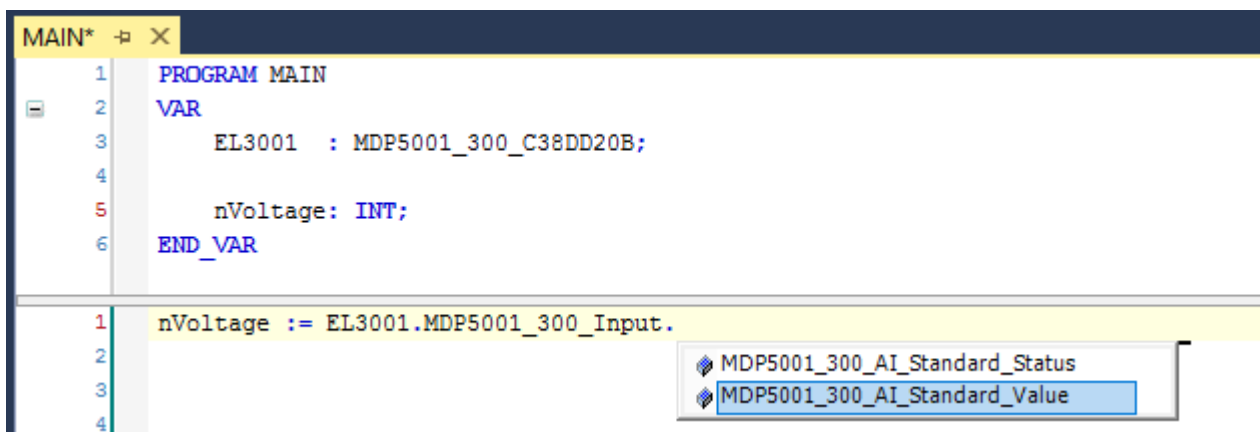

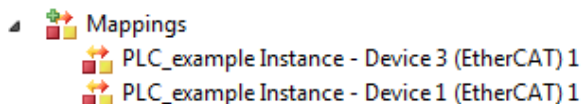


Fig. 73: Reading a variable from the structure of the process data


**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:

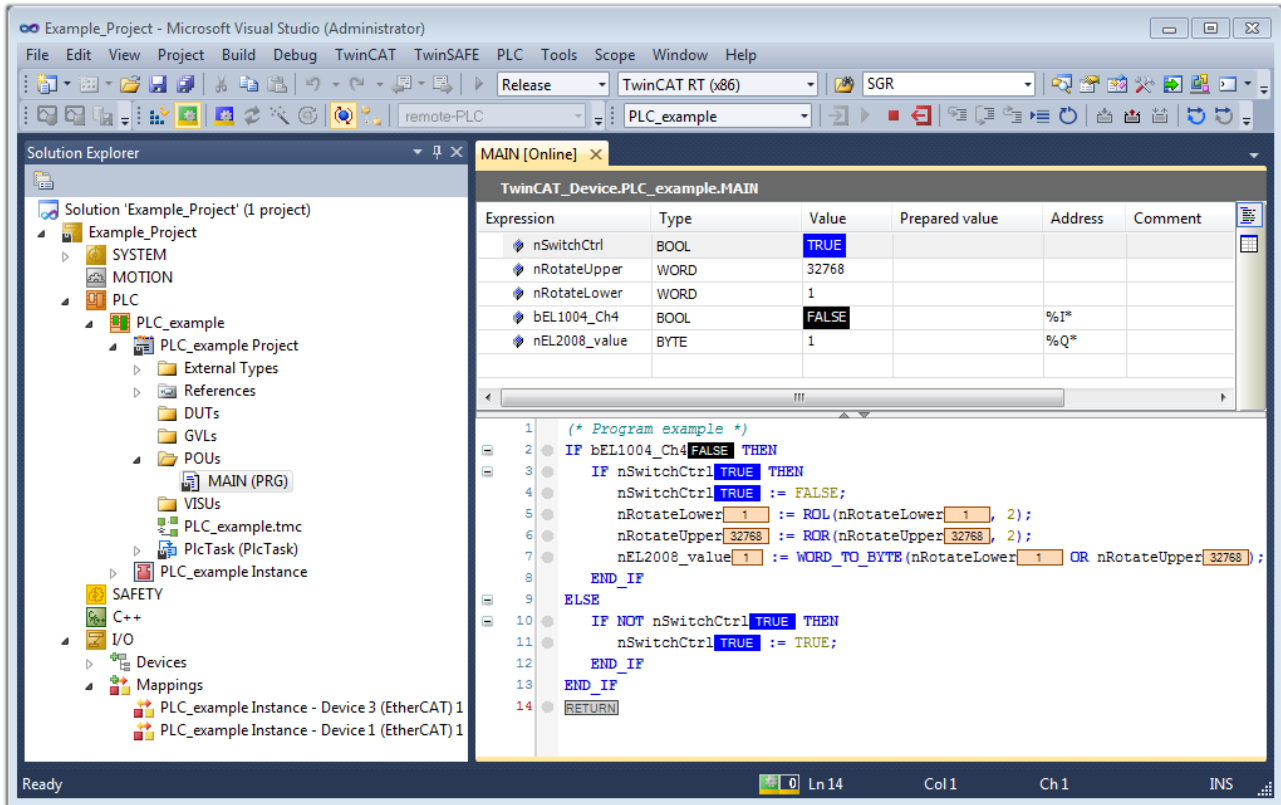




Fig. 74: TwinCAT 3 development environment (VS shell): logged-in, after program startup

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).

## 5.2 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>.

## 5.2.1 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.

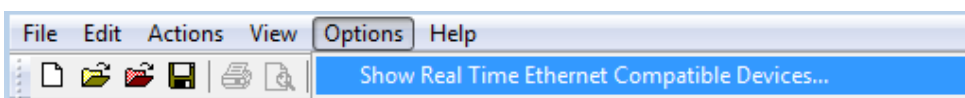


Fig. 75: System Manager "Options" (TwinCAT 2)

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

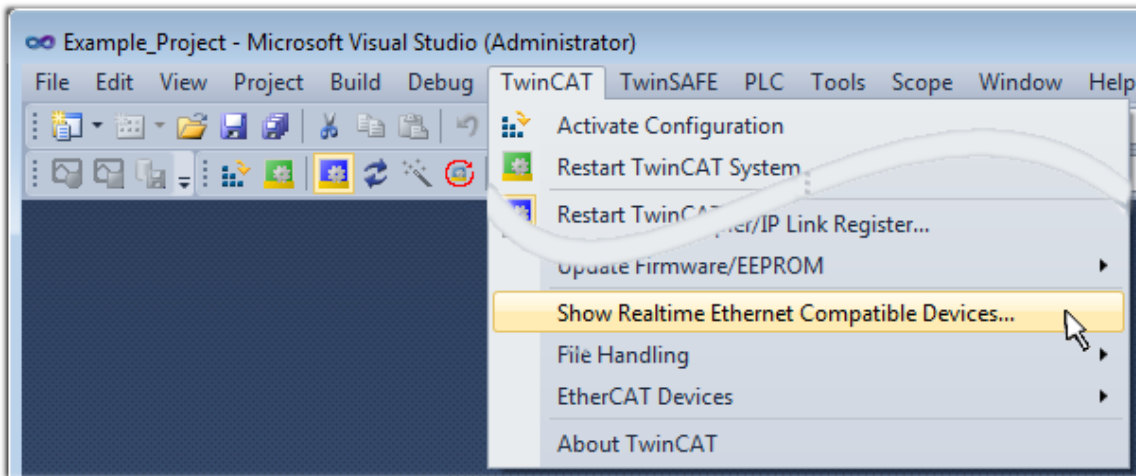


Fig. 76: Call up under VS Shell (TwinCAT 3)

**B: Via TcRtelInstall.exe in the TwinCAT directory**

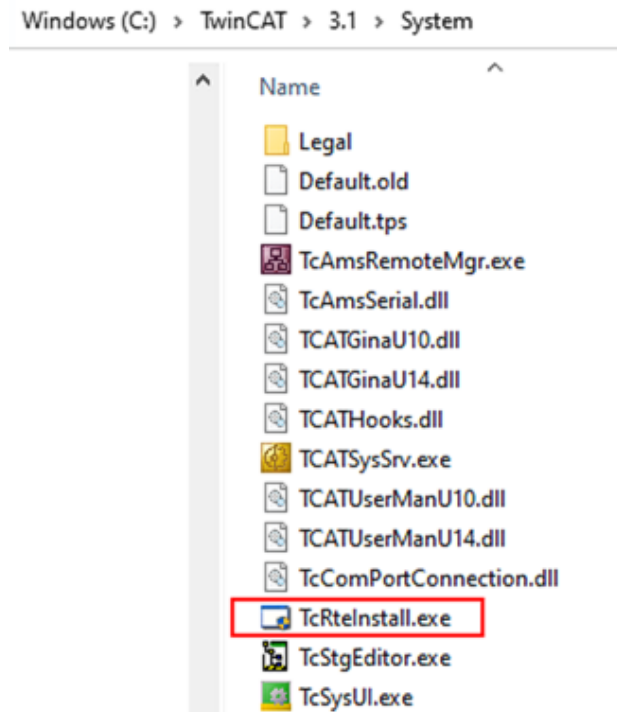


Fig. 77: TcRtelInstall in the TwinCAT directory

In both cases, the following dialog appears:



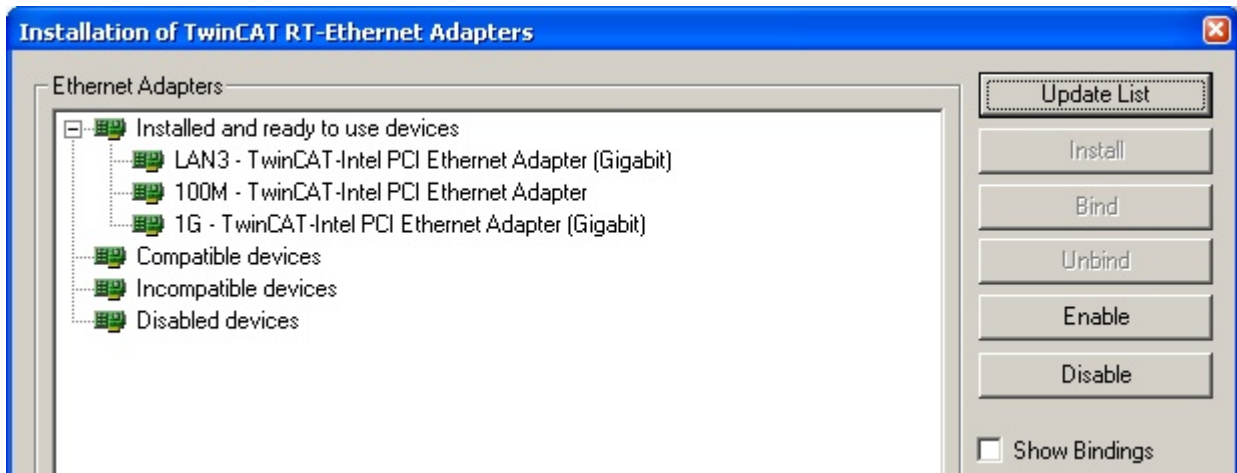


Fig. 78: Overview of network interfaces

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

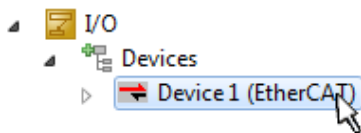
A Windows warning regarding the unsigned driver can be ignored.

**Alternatively** an EtherCAT-device can be inserted first of all as described in chapter [Offline configuration creation](#), section “Creating the EtherCAT device” [▶ 89] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices...”):



Fig. 79: 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)



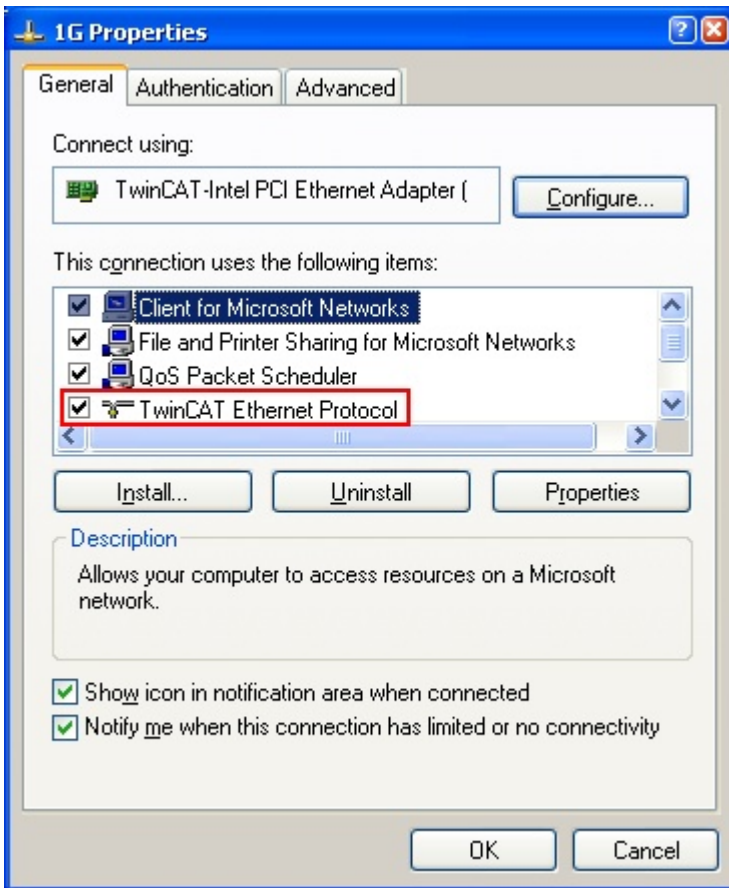


Fig. 80: Windows properties of the network interface

A correct setting of the driver could be:

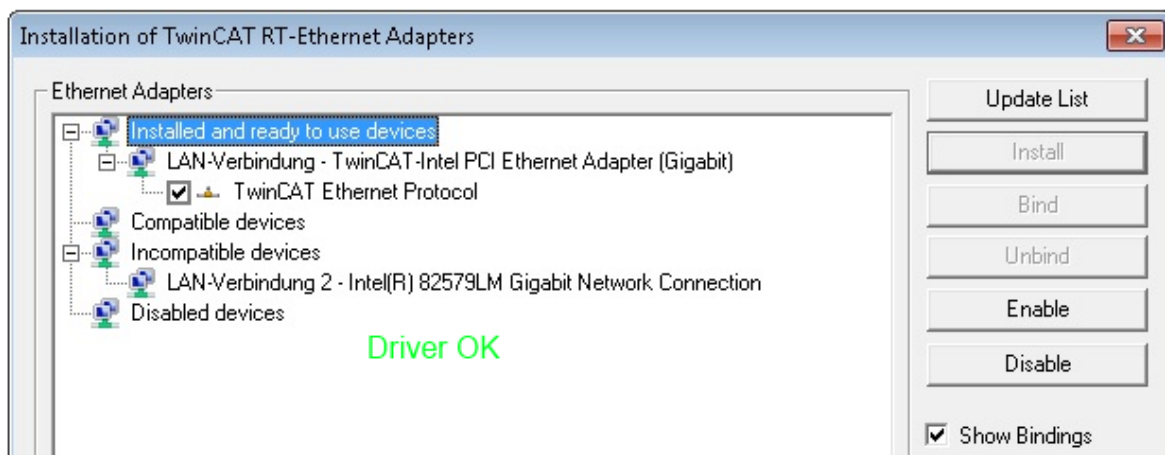


Fig. 81: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

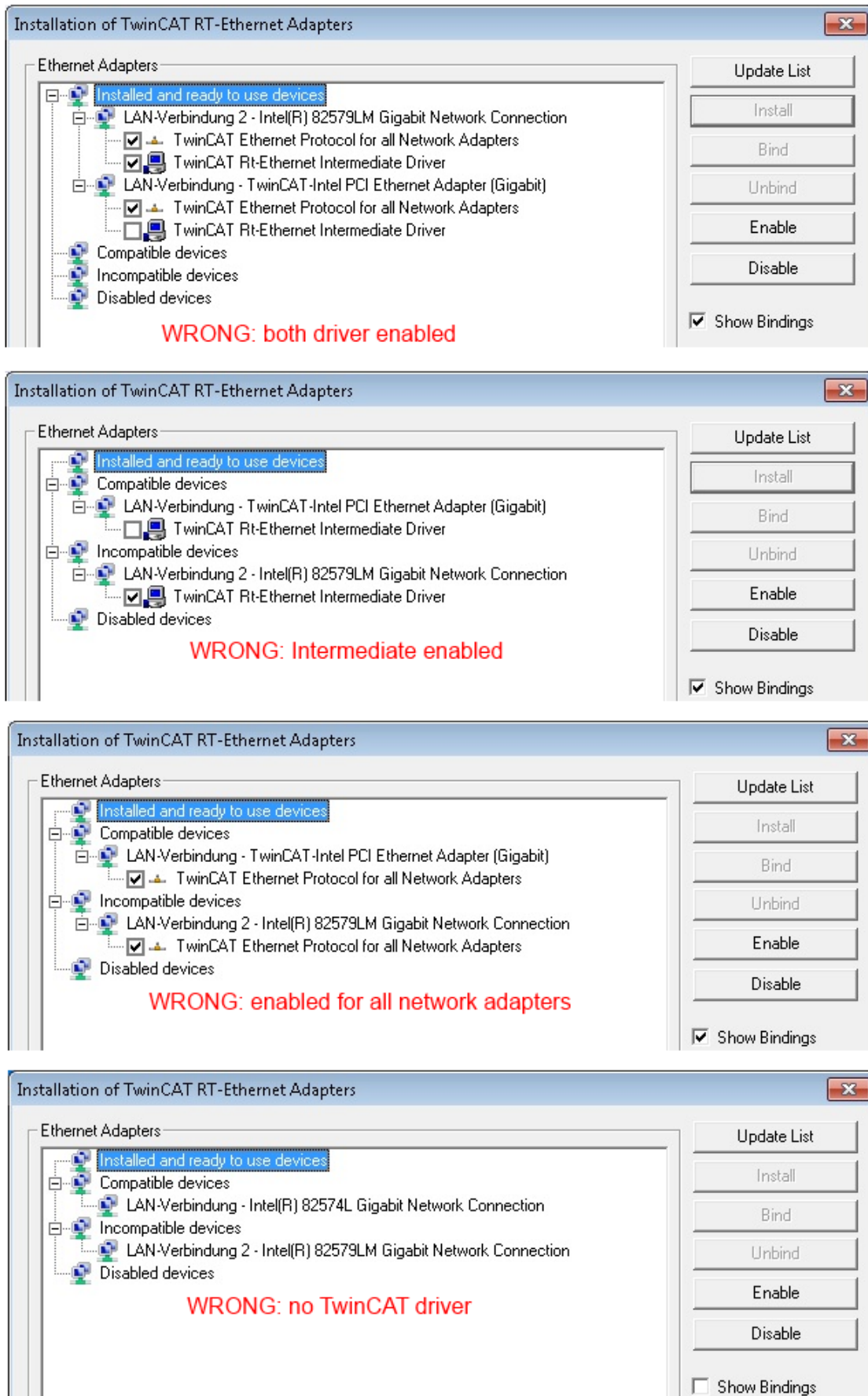


Fig. 82: Incorrect driver settings for the Ethernet port

IP address of the port used

**i** 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.

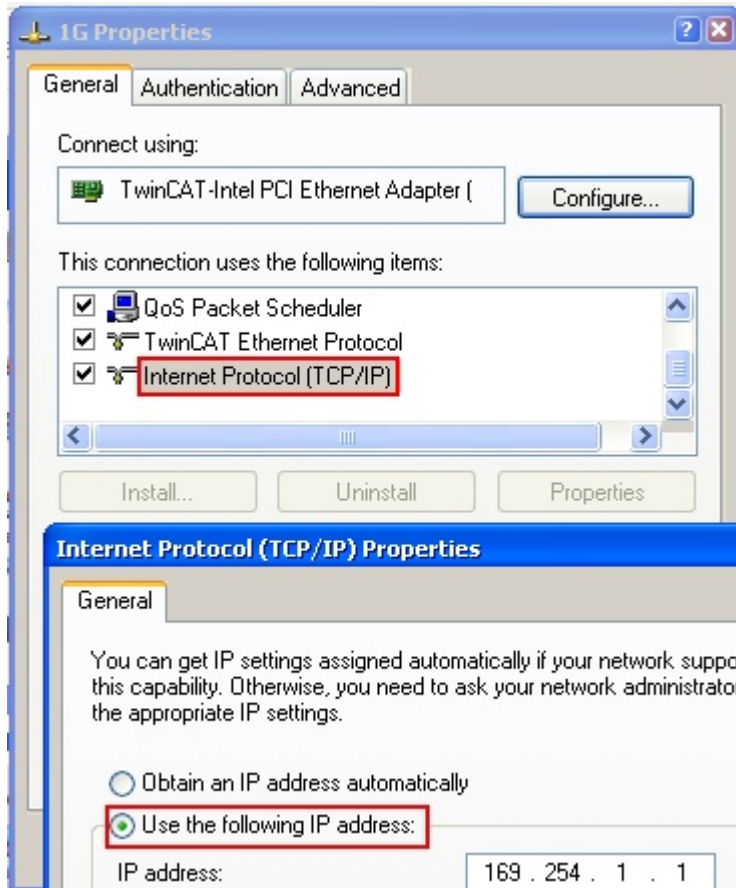


Fig. 83: TCP/IP setting for the Ethernet port

## 5.2.2 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](#) [▶ 88] 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”

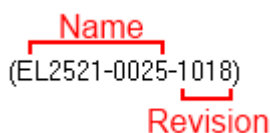


Fig. 84: Identifier structure

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.

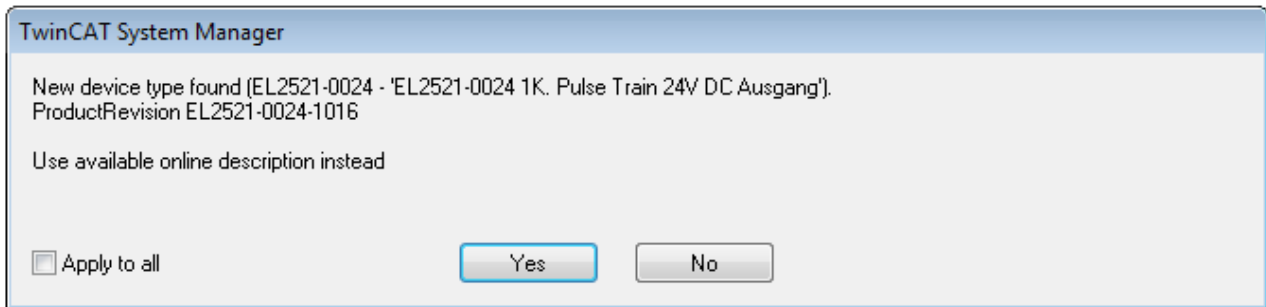


Fig. 85: OnlineDescription information window (TwinCAT 2)

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

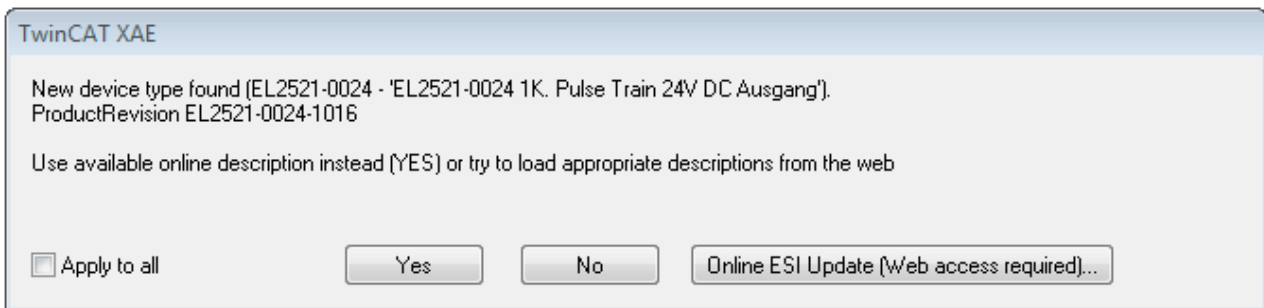


Fig. 86: Information window OnlineDescription (TwinCAT 3)

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 [▶ 89]”.

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. 87: File OnlineDescription.xml created by the System Manager

If 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*).

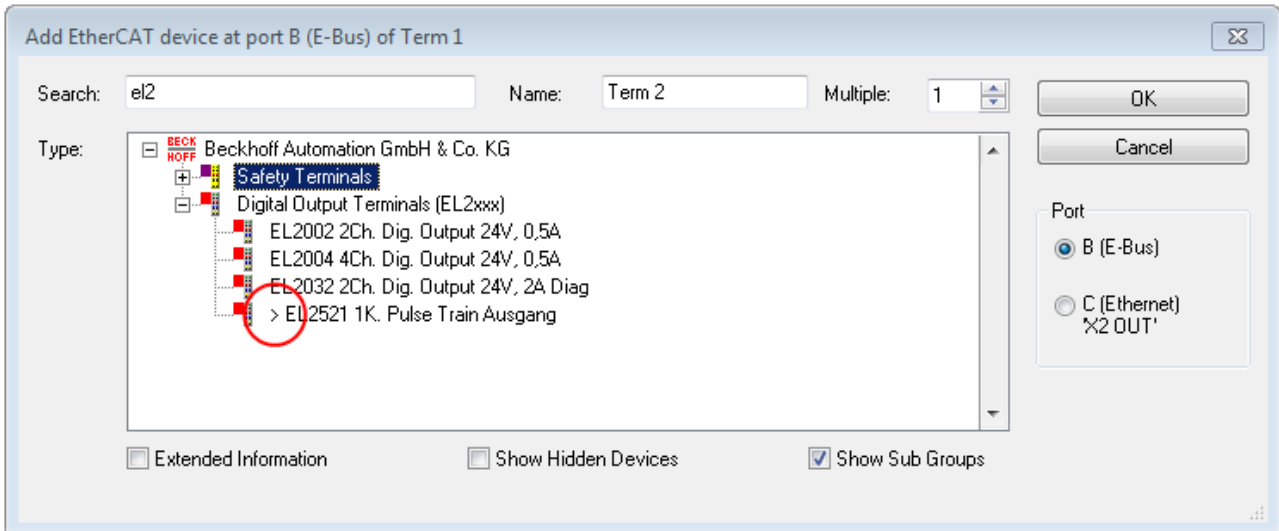


Fig. 88: Indication of an online recorded ESI of EL2521 as an example

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

### **i** 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.

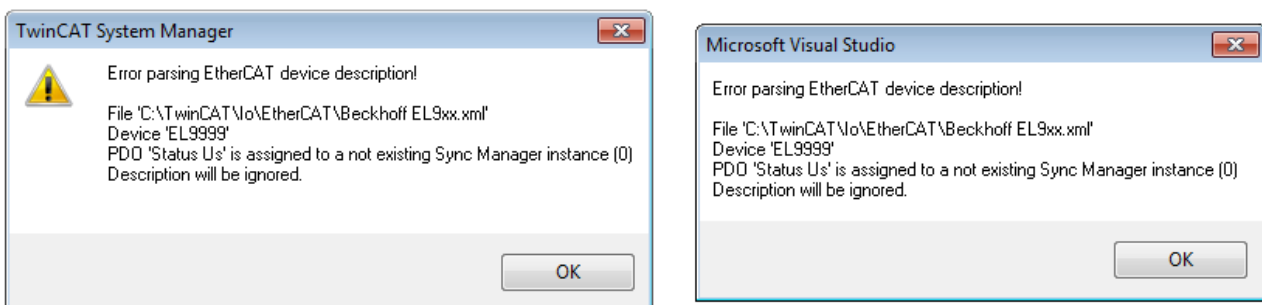


Fig. 89: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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



### 5.2.3 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:

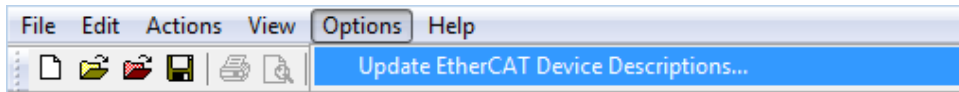


Fig. 90: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

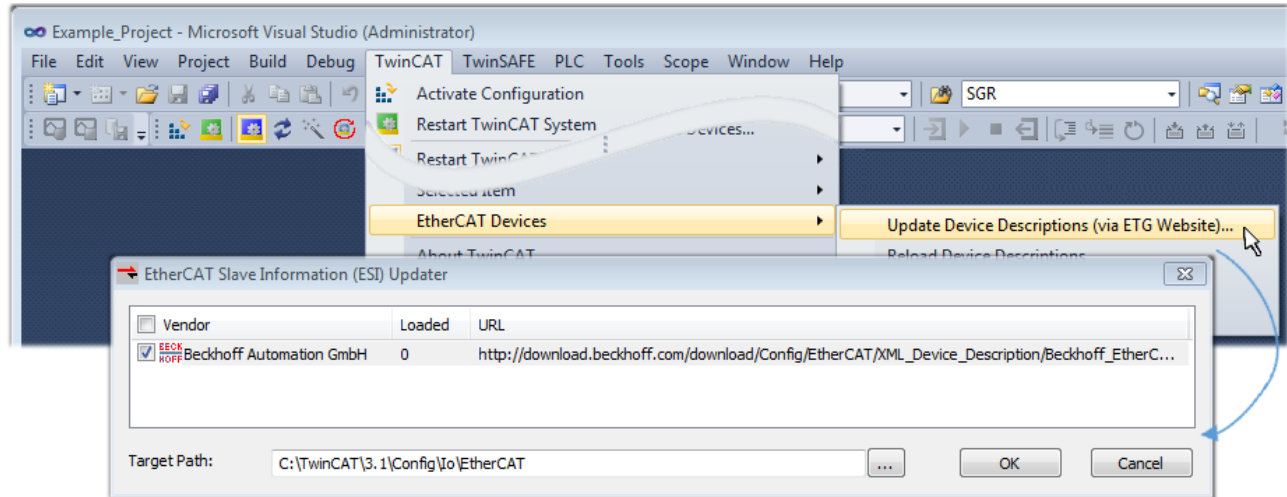


Fig. 91: Using the ESI Updater (TwinCAT 3)

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)...”.

### 5.2.4 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” \[▶ 84\]](#).

#### 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 [▶ 94] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 95]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 98]

The scan with existing configuration [▶ 99] can also be carried out for comparison.

## 5.2.5 OFFLINE configuration creation

### Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

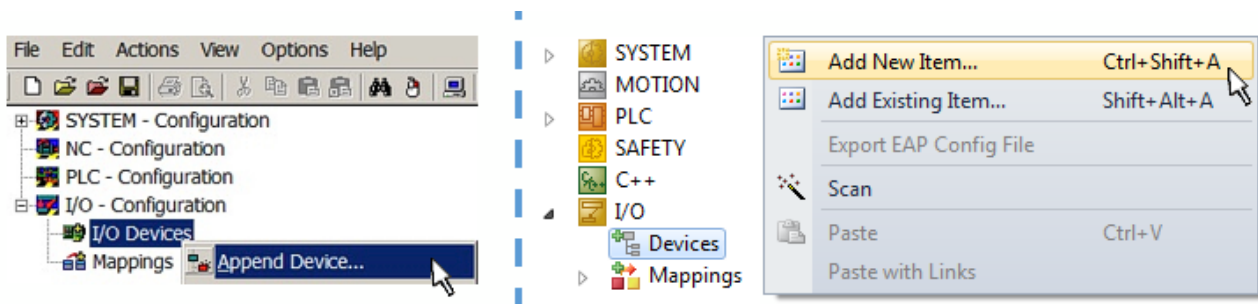


Fig. 92: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

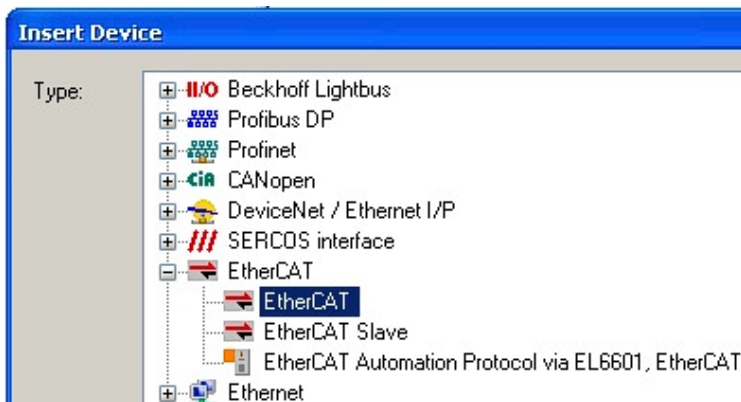


Fig. 93: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

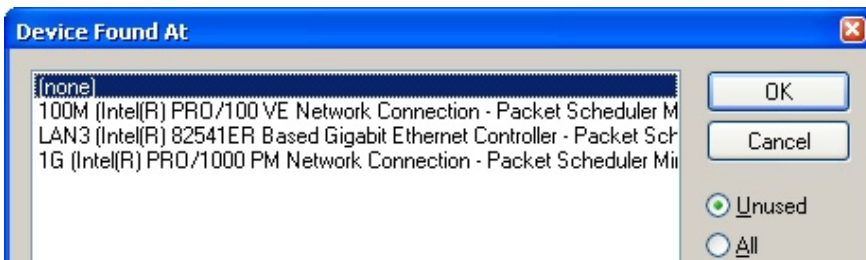


Fig. 94: Selecting the Ethernet port

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)”.

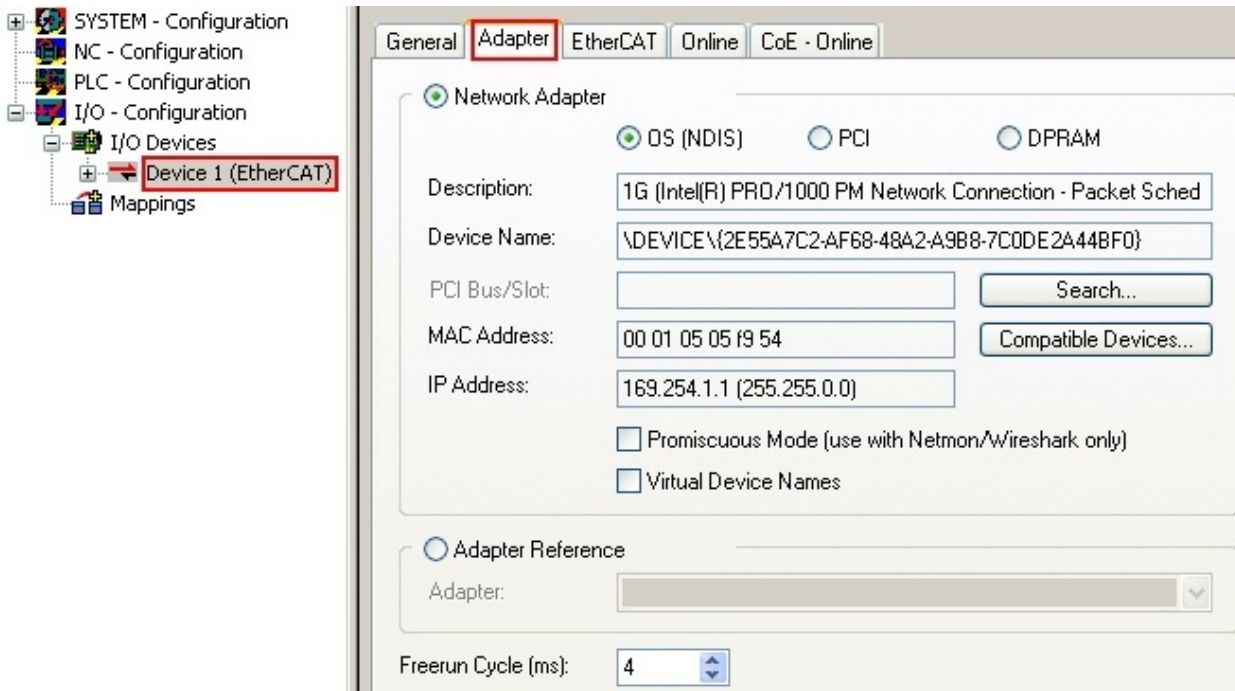
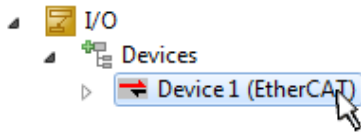


Fig. 95: EtherCAT device properties (TwinCAT 2)

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



**i** **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 \[▶ 78\]](#).

**Defining EtherCAT slaves**

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



Fig. 96: 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).

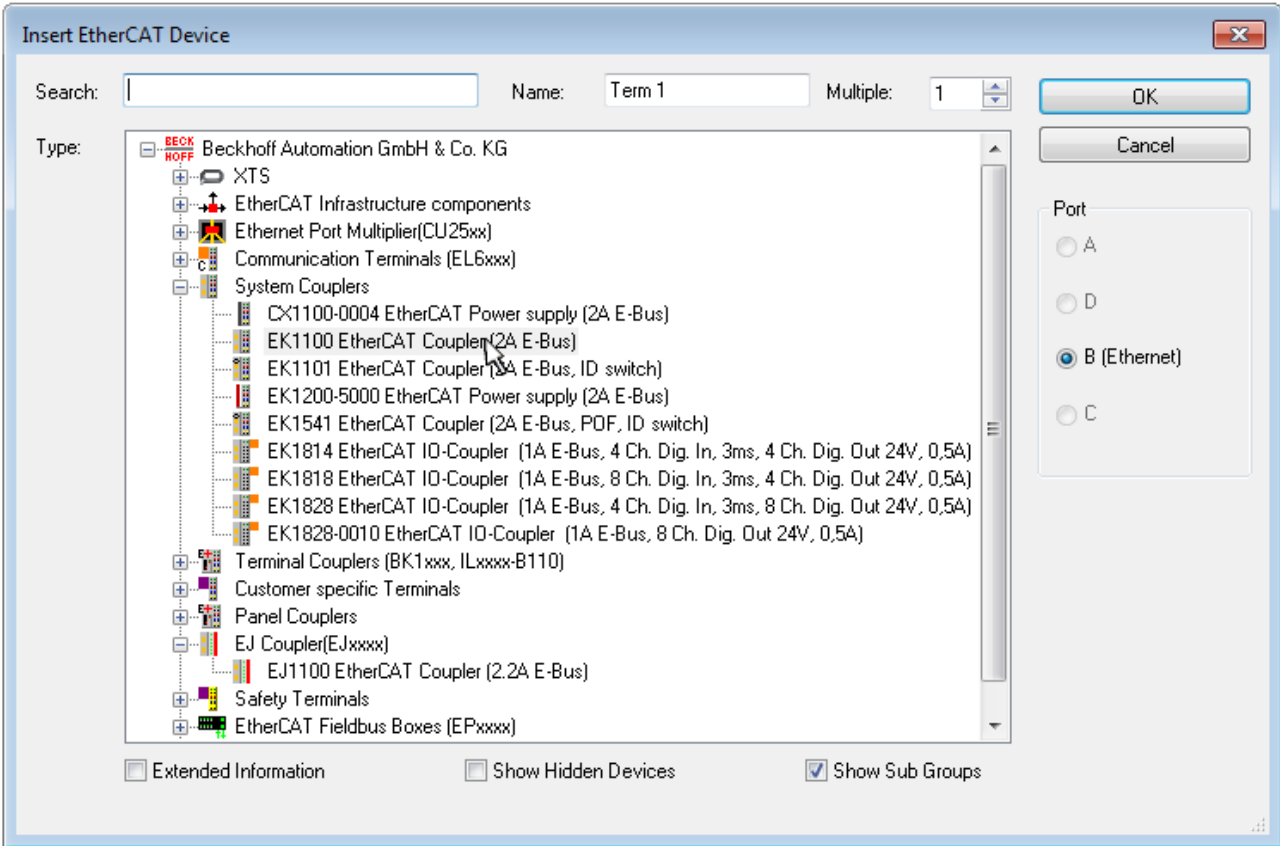


Fig. 97: Selection dialog for new EtherCAT device

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”.

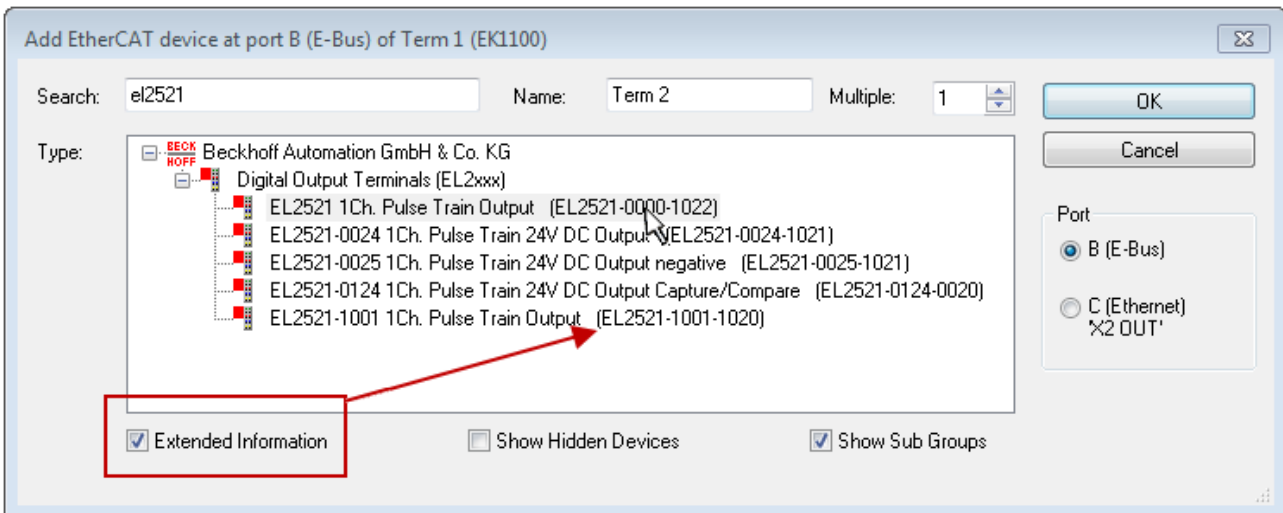


Fig. 98: Display of device revision

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”.

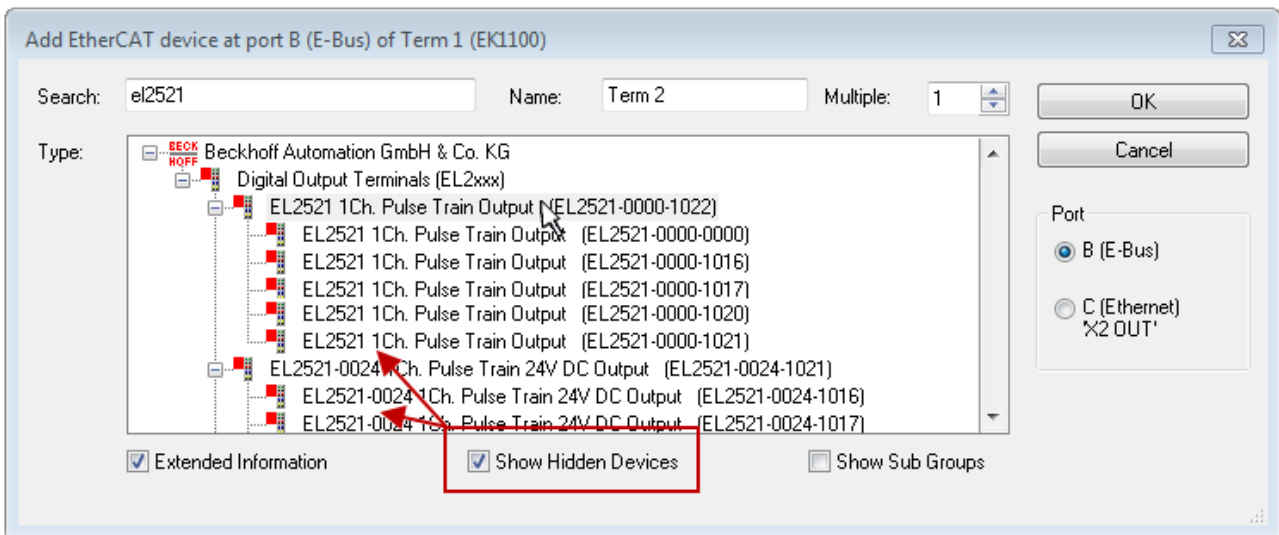


Fig. 99: Display of previous revisions

**i 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.

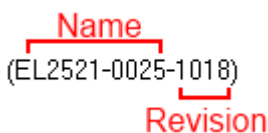


Fig. 100: Name/revision of the terminal

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, ...

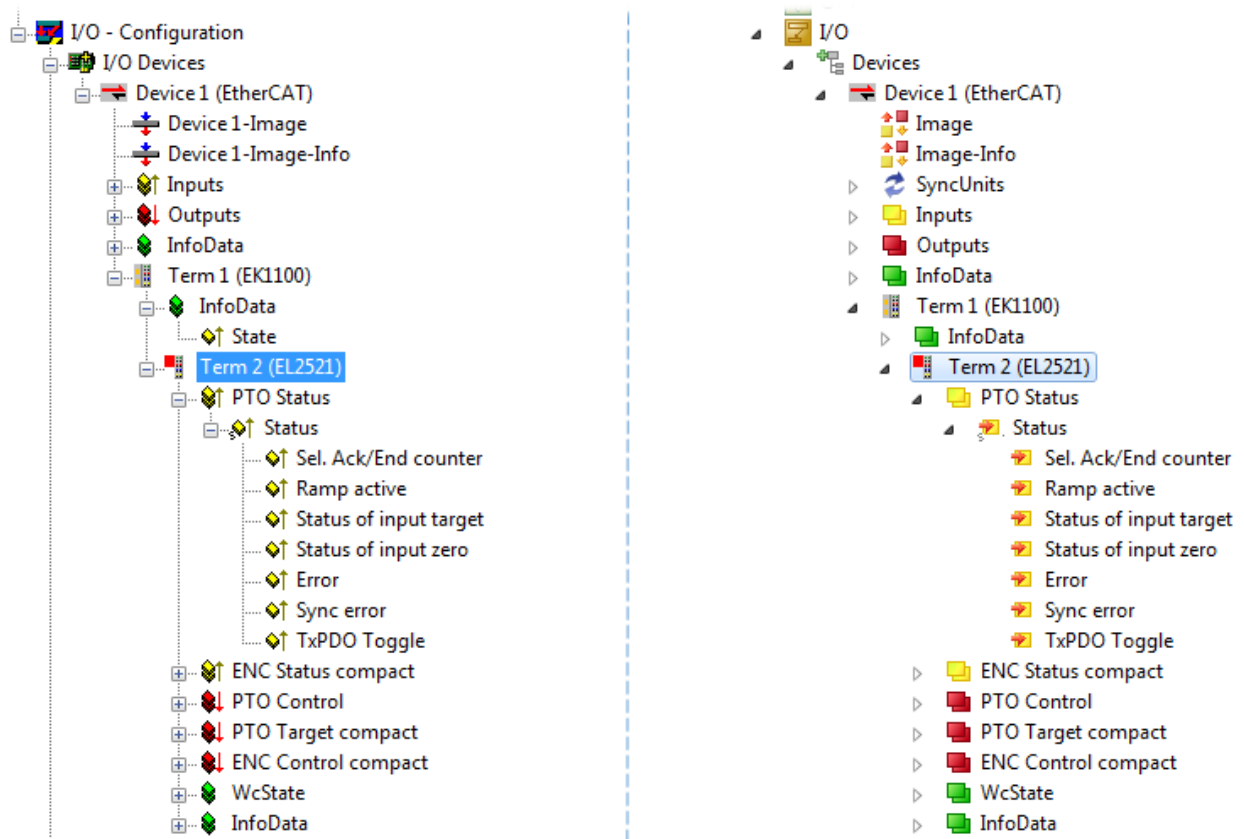




Fig. 101: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



## 5.2.6 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:  .
- 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

**i** 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.

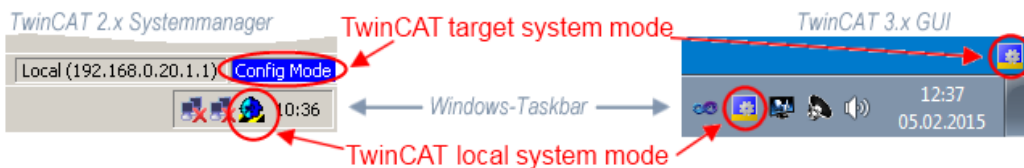


Fig. 102: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

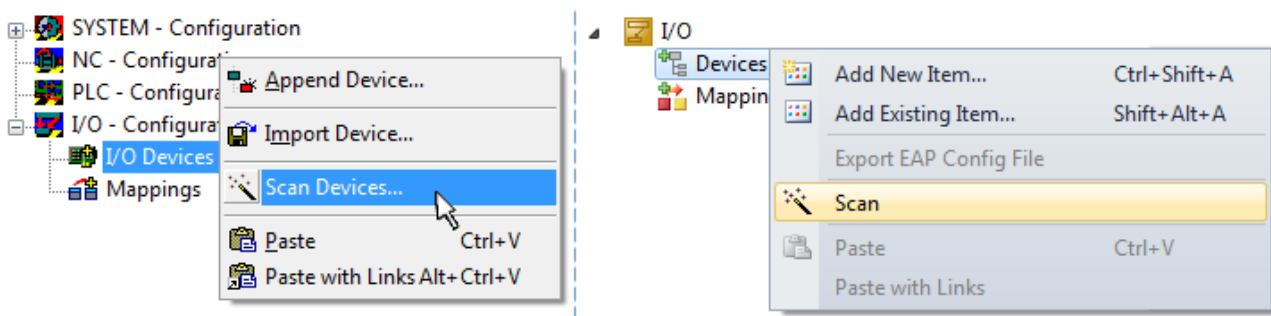


Fig. 103: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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

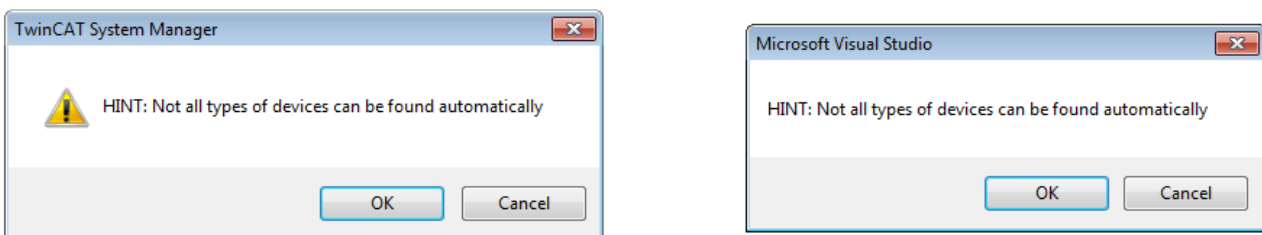


Fig. 104: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

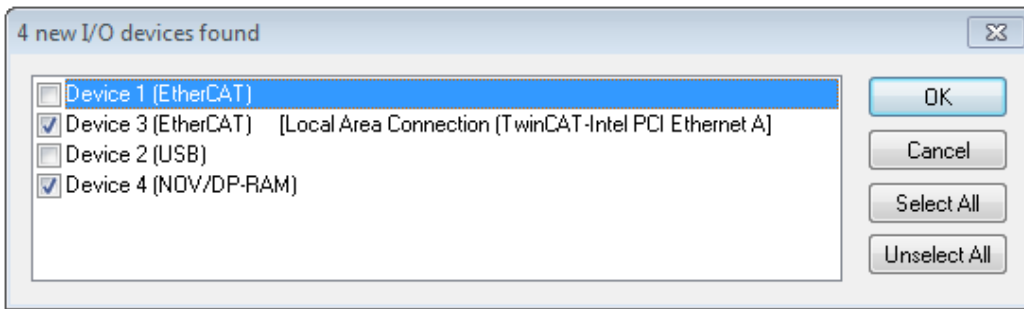


Fig. 105: Detected Ethernet devices

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](#) [▶ 78].

**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.

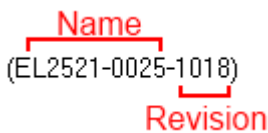


Fig. 106: Example default state

**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](#) [▶ 99] 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:



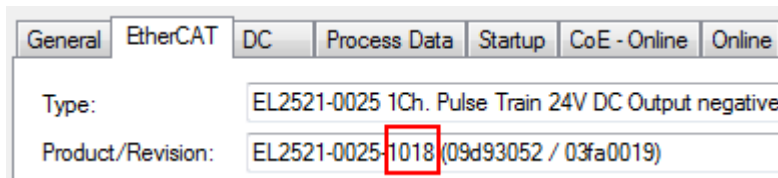


Fig. 107: Installing EtherCAT terminal with revision -1018

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 [► 99] 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:

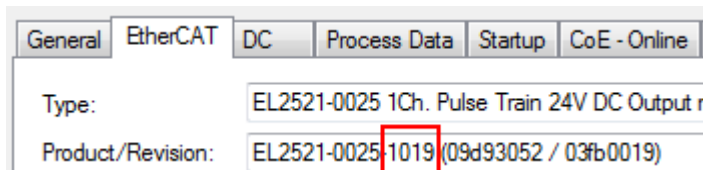


Fig. 108: Detection of EtherCAT terminal with revision -1019

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



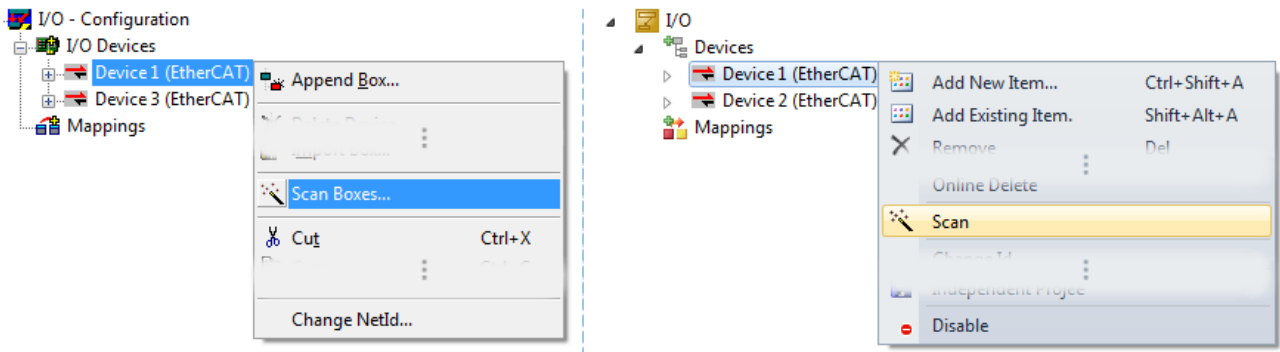


Fig. 110: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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.

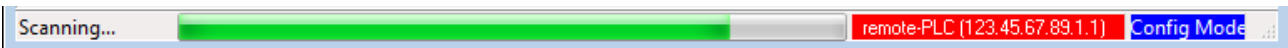


Fig. 111: Scan progress exemplary by TwinCAT 2

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



Fig. 112: 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. 113: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 114: 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*.

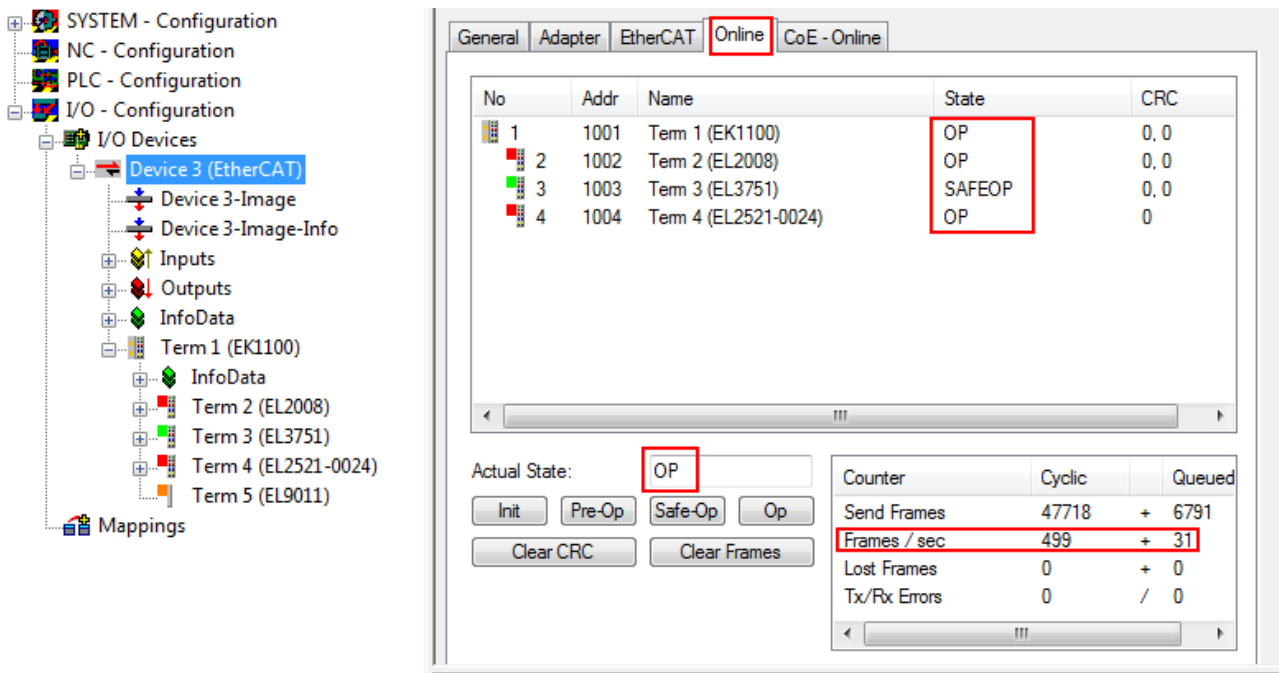


Fig. 115: Online display example

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 \[► 89\]](#).

### 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.

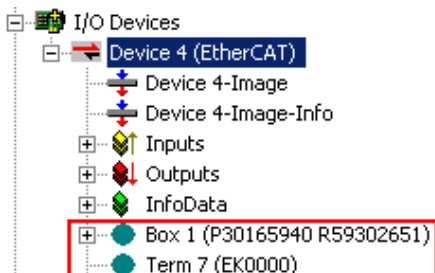


Fig. 116: Faulty identification

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. 117: 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.

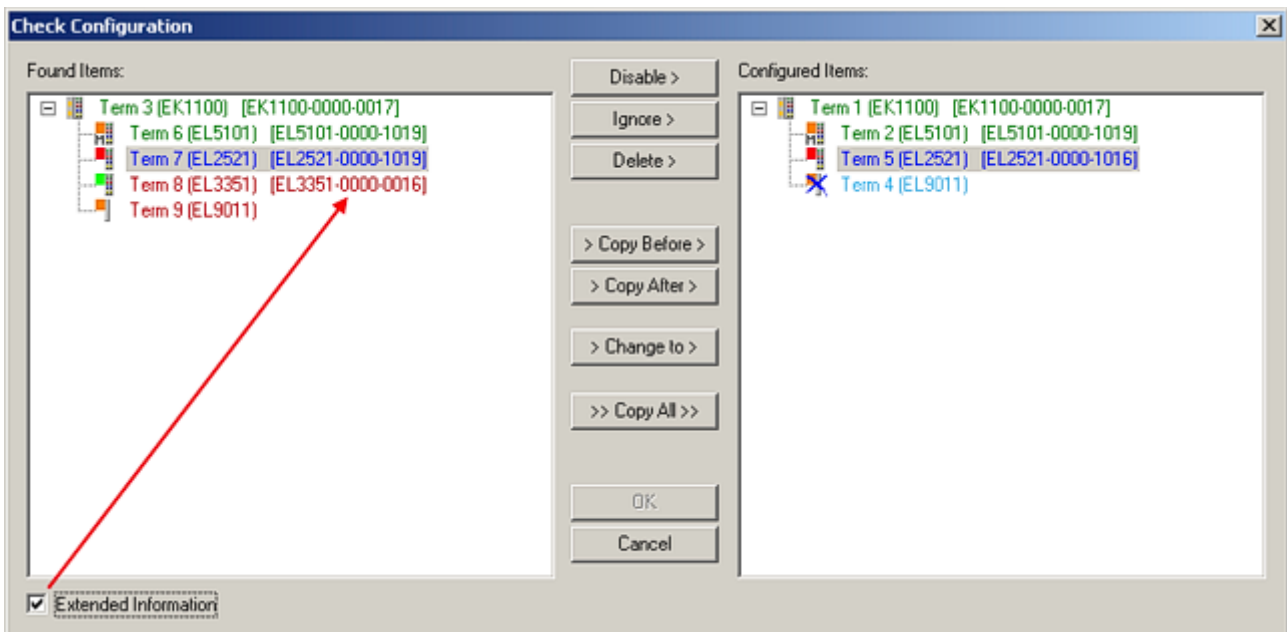


Fig. 118: Correction dialog

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	<ul style="list-style-type: none"> <li>This EtherCAT slave is not present on the other side.</li> <li>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.</li> </ul>

### **i** 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 $\geq$ 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.

Name  
(EL2521-0025-1018)  
Revision

Fig. 119: 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, ...

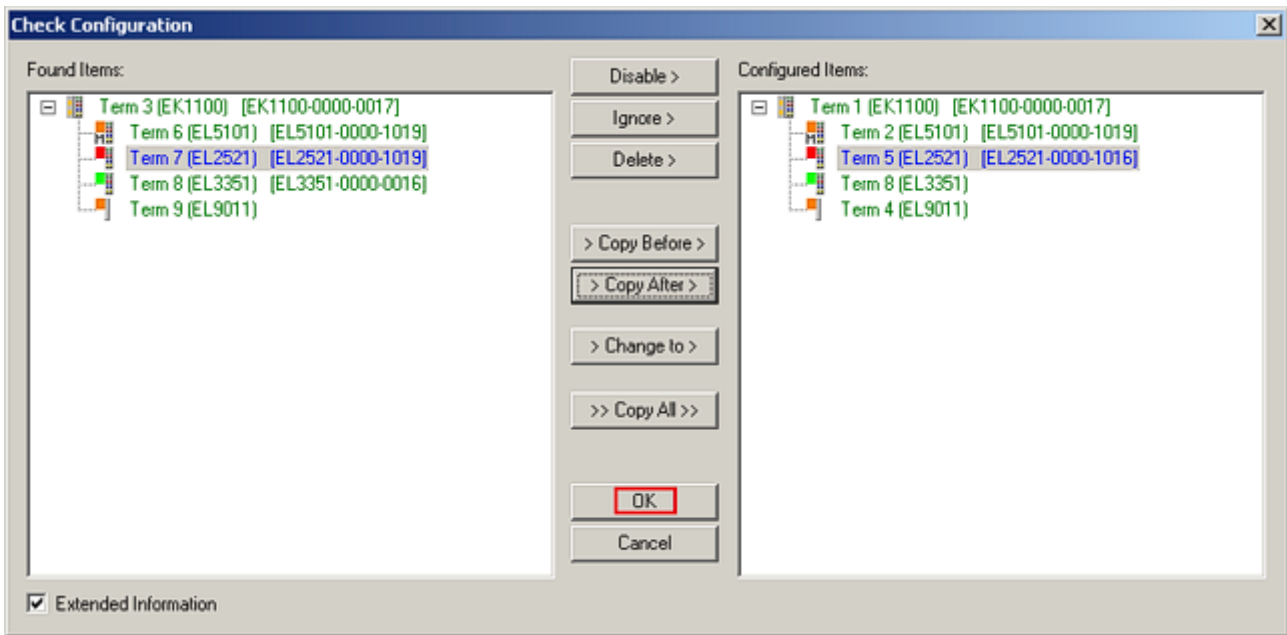


Fig. 120: Correction dialog with modifications

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.

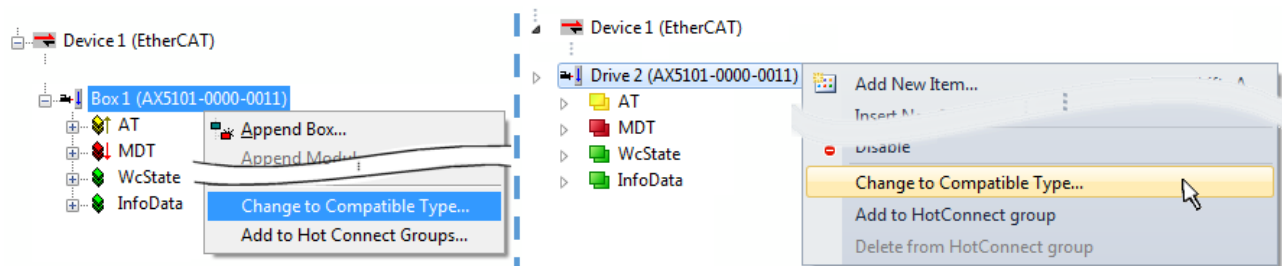


Fig. 121: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

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

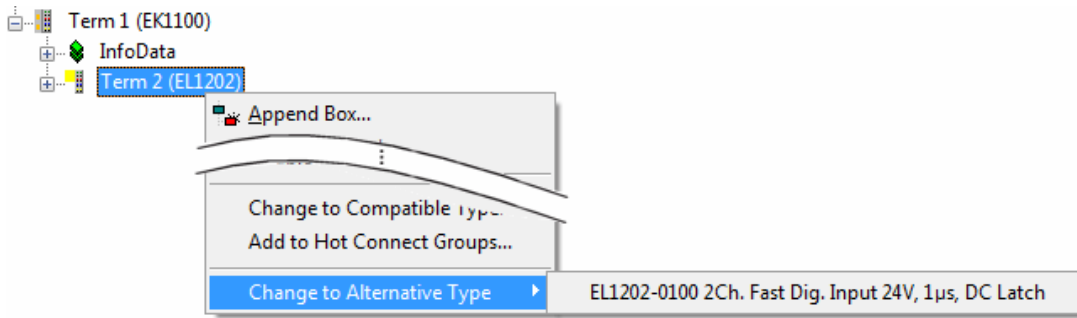


Fig. 122: TwinCAT 2 Dialog Change to Alternative Type

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).

### 5.2.7 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).

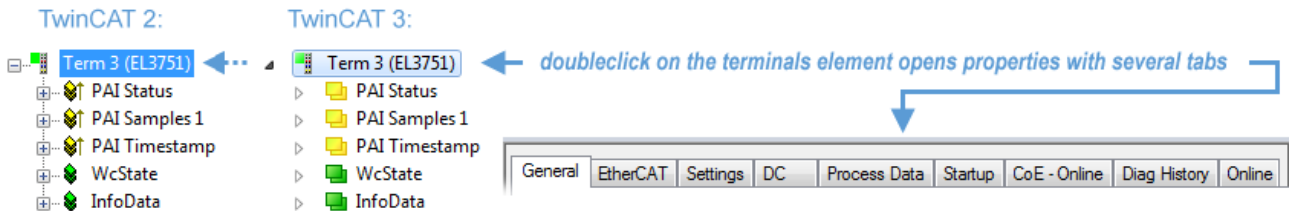


Fig. 123: Branch element as terminal EL3751

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

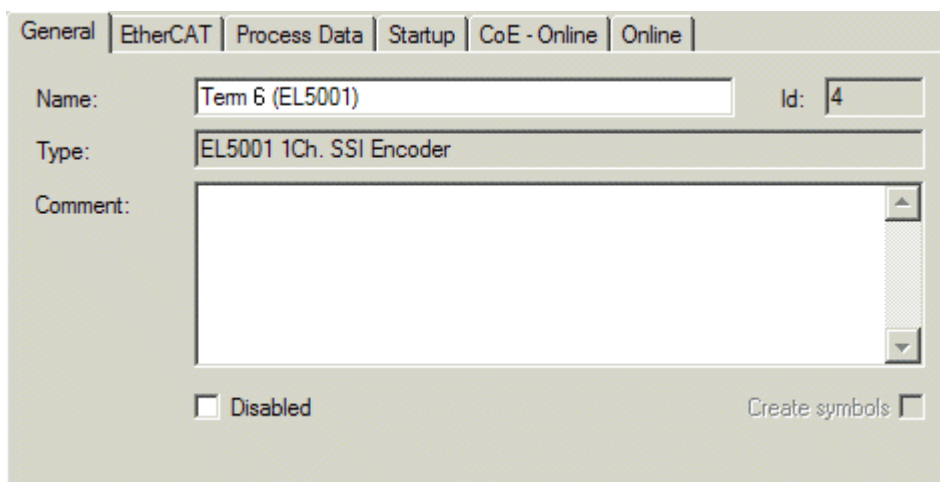


Fig. 124: “General” tab

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

**“EtherCAT” tab**

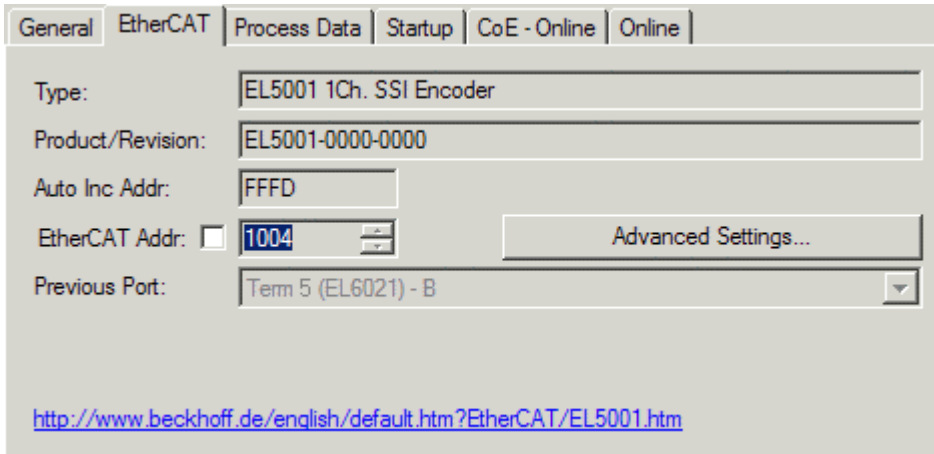


Fig. 125: “EtherCAT” tab

<b>Type</b>	EtherCAT device type
<b>Product/Revision</b>	Product and revision number of the EtherCAT device
<b>Auto Inc Addr.</b>	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 <sub>hex</sub> . For each further slave the address is decremented by 1 (FFFF <sub>hex</sub> , FFFE <sub>hex</sub> etc.).
<b>EtherCAT Addr.</b>	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.
<b>Previous Port</b>	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.
<b>Advanced Settings</b>	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 (**Process Data Objects, 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.



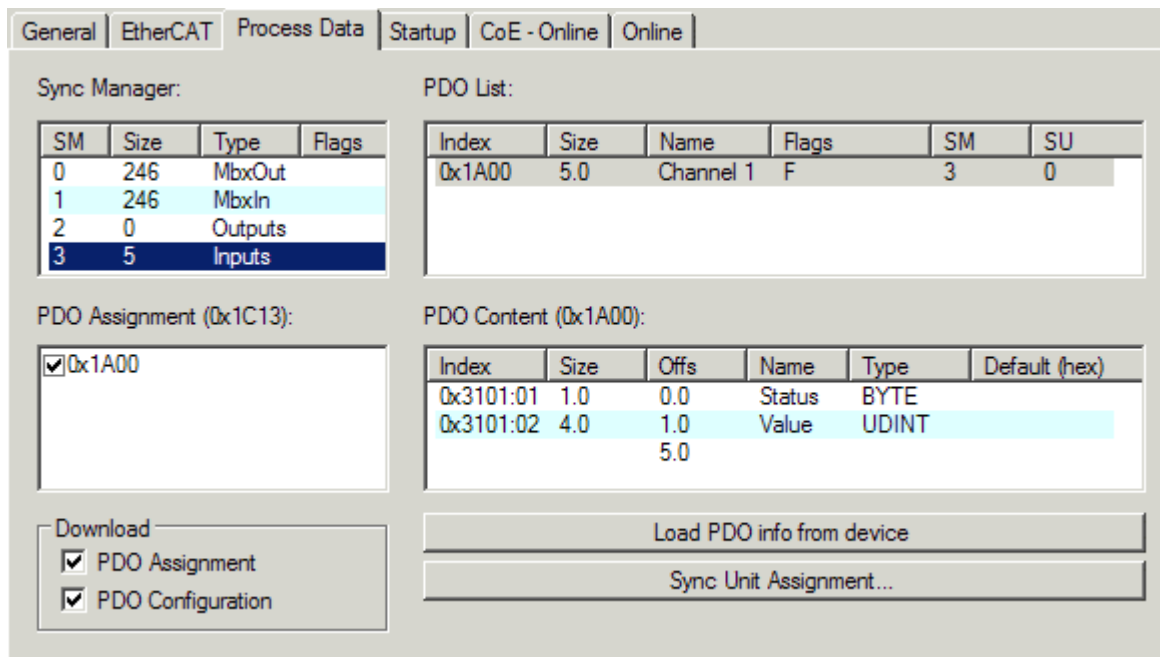


Fig. 126: "Process Data" tab

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").



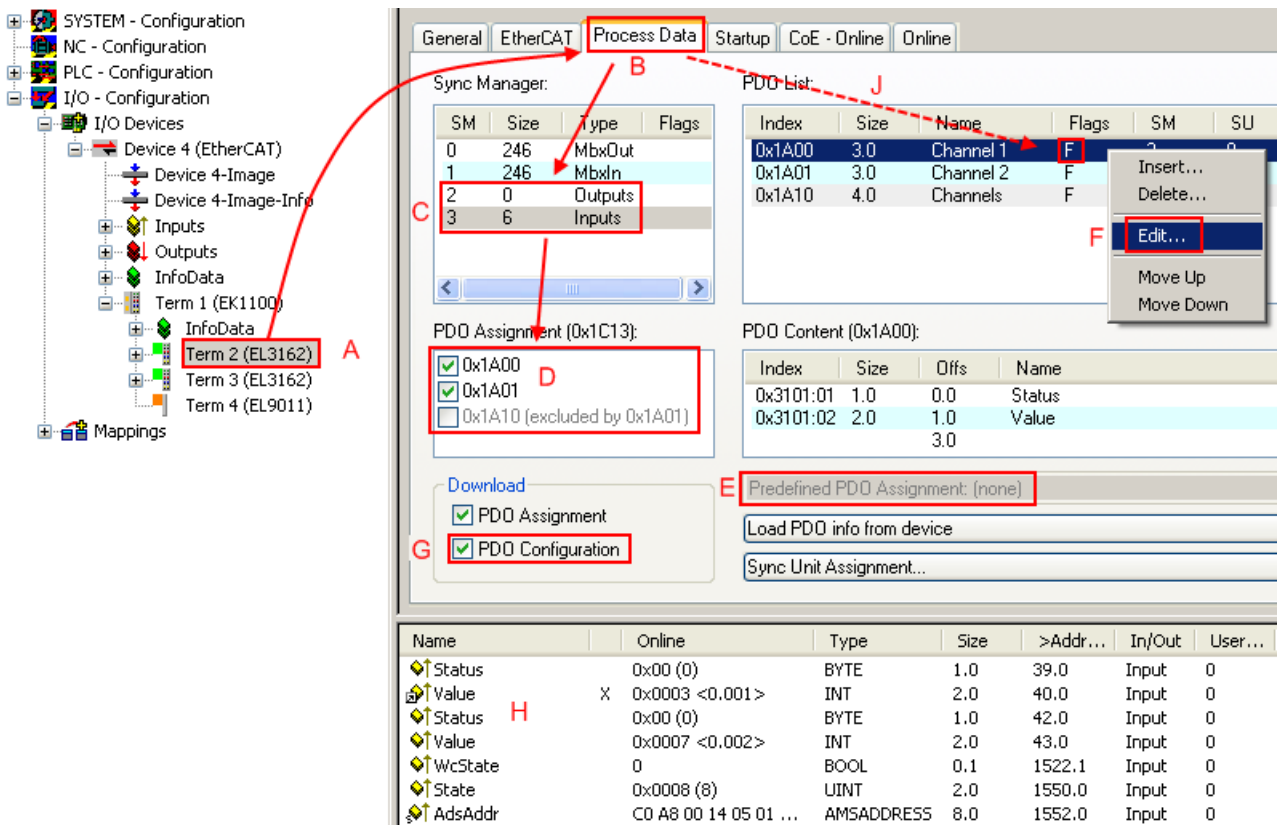


Fig. 127: Configuring the process data

**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 [► 110] 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.

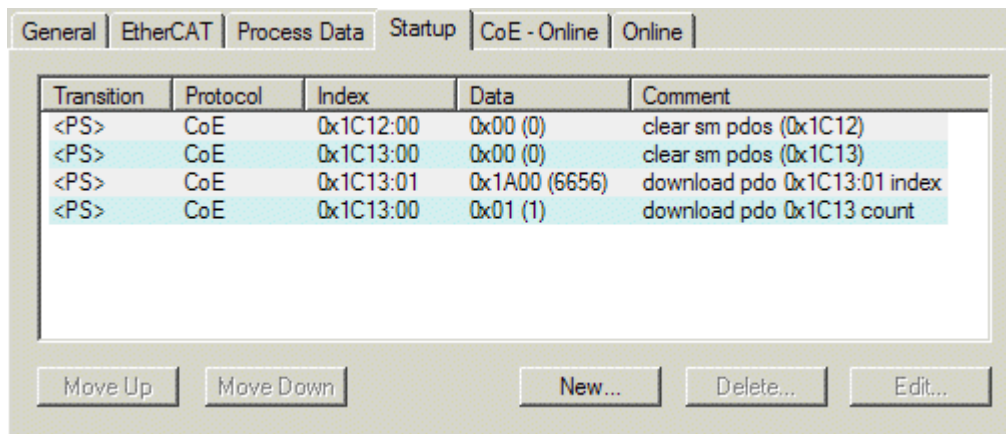


Fig. 128: "Startup" tab

Column	Description
Transition	Transition to which the request is sent. This can either be <ul style="list-style-type: none"> <li>the transition from pre-operational to safe-operational (PS), or</li> <li>the transition from safe-operational to operational (SO).</li> </ul> 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.

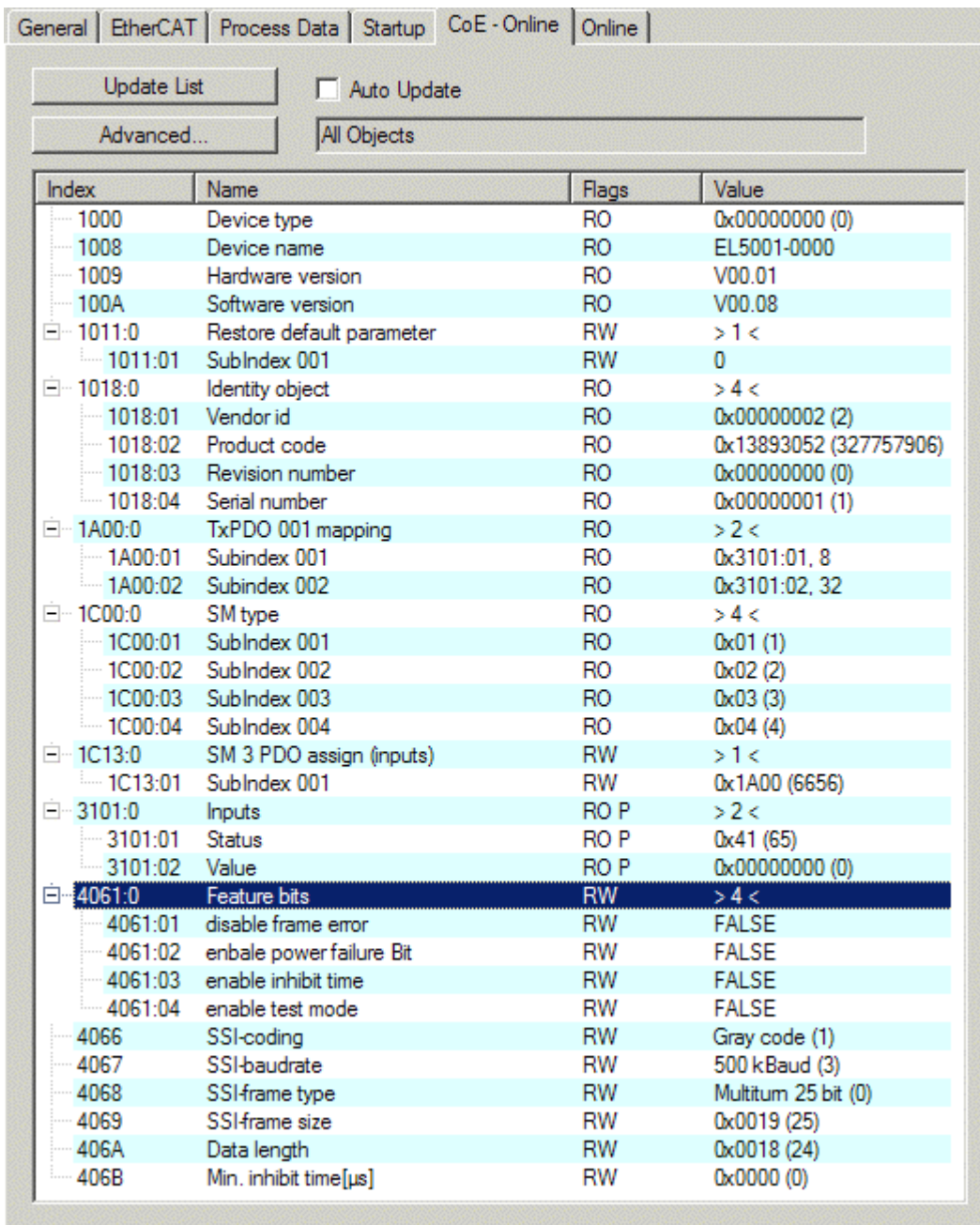


Fig. 129: "CoE - Online" tab

**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.

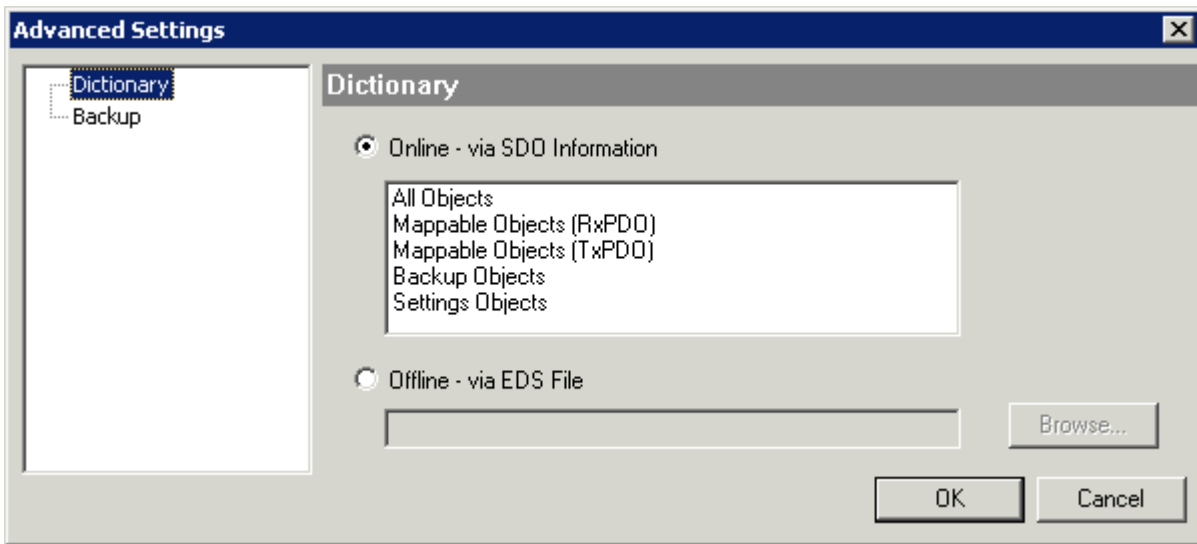


Fig. 130: Dialog “Advanced settings”

**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

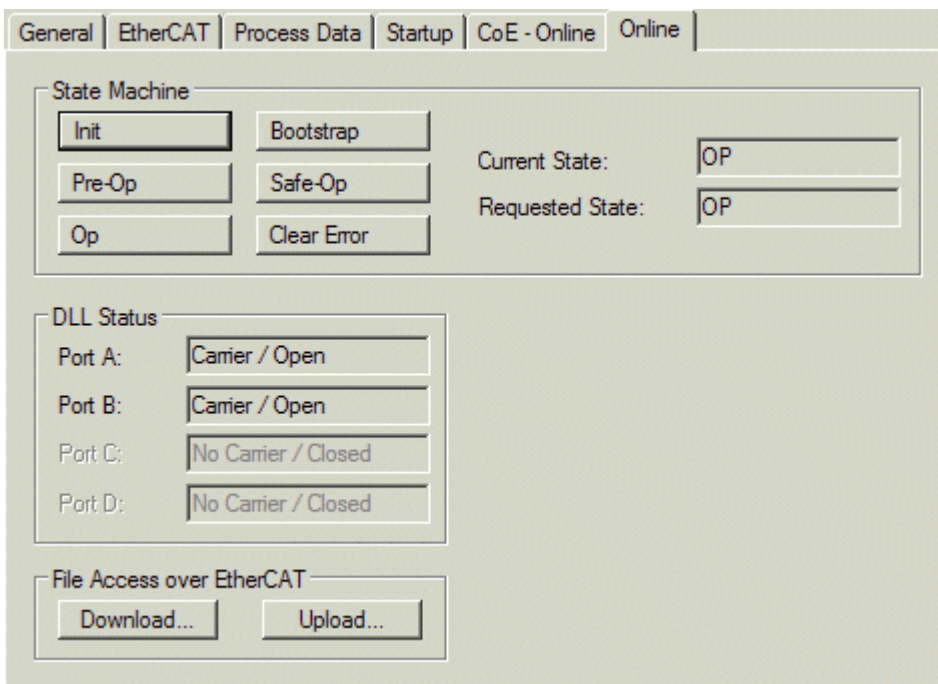


Fig. 131: “Online” tab

**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)**

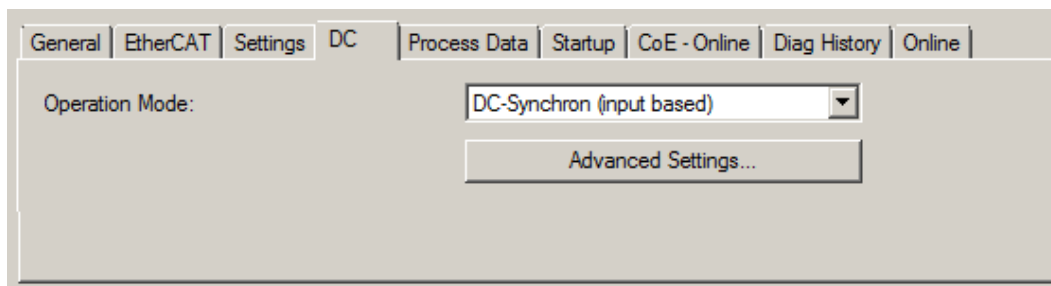


Fig. 132: “DC” tab (Distributed Clocks)

- 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

### 5.2.7.1 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 \[▶ 108\]](#)),
  - 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 <i>PDO Assignment</i> 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 [▶ 105] 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.

**5.2.8 NC configuration (motion)**

**● Installation of the latest XML device description**

**i** 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*).

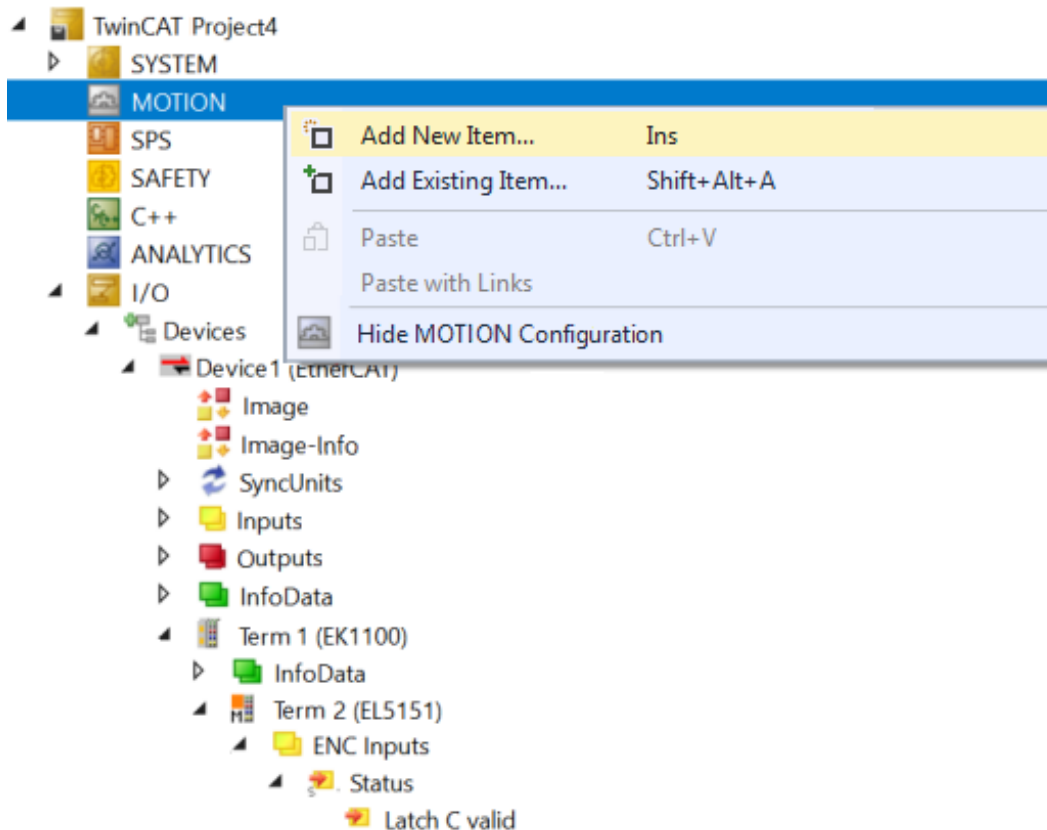


Fig. 133: Motion, add item

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

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

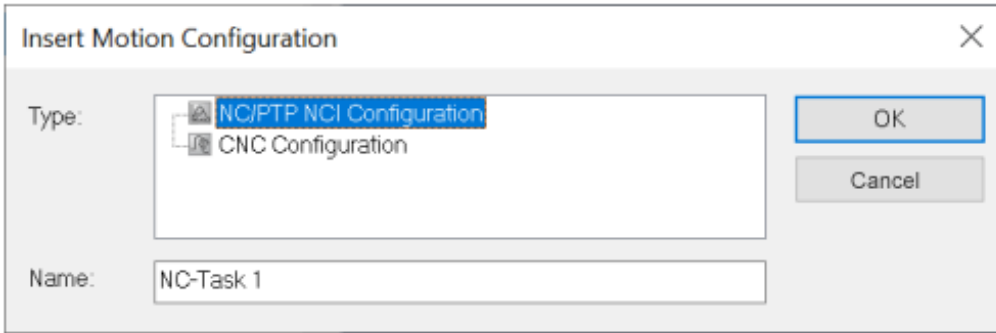


Fig. 134: Select type, name and confirm task

**Adding an axis**

Right-click *Axes*

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

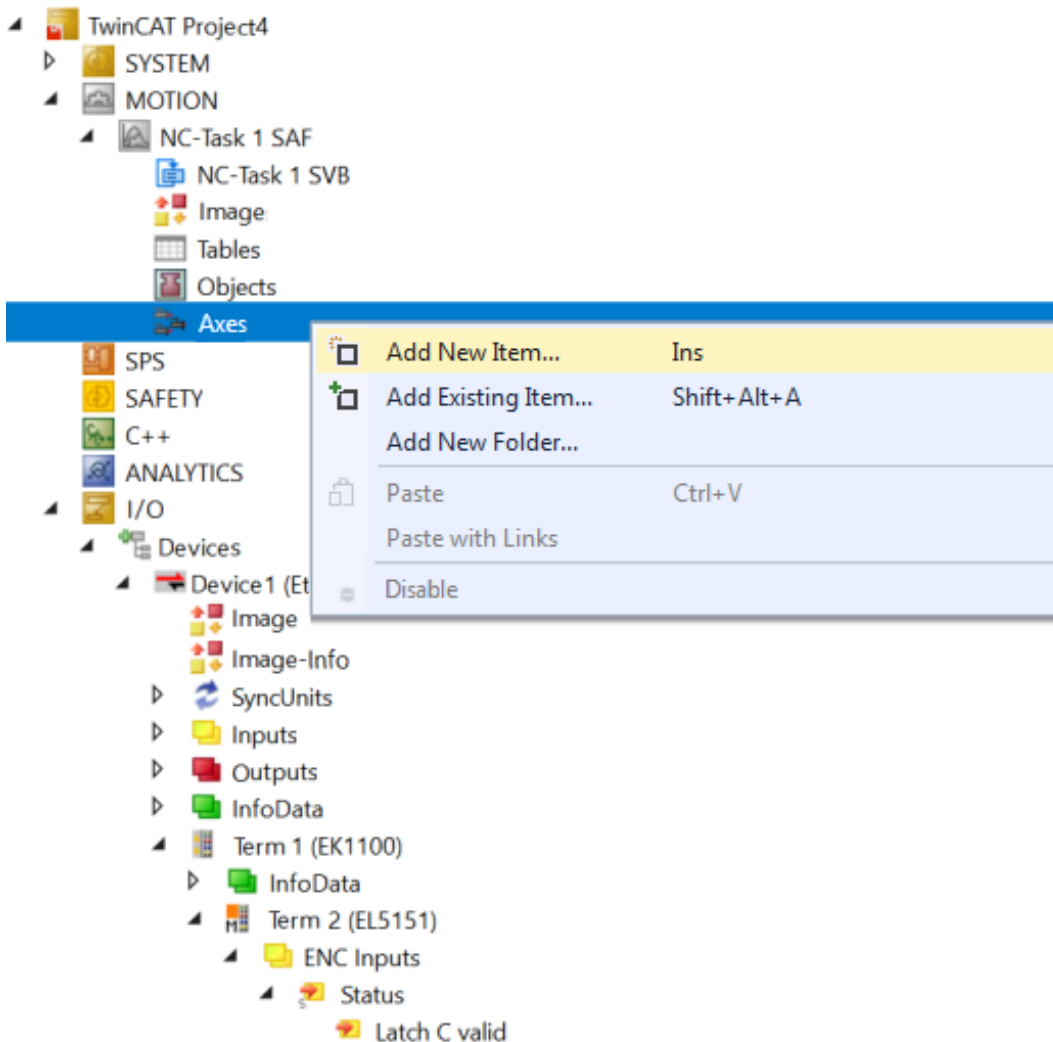


Fig. 135: Insert axis

Choose a name for the axis

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



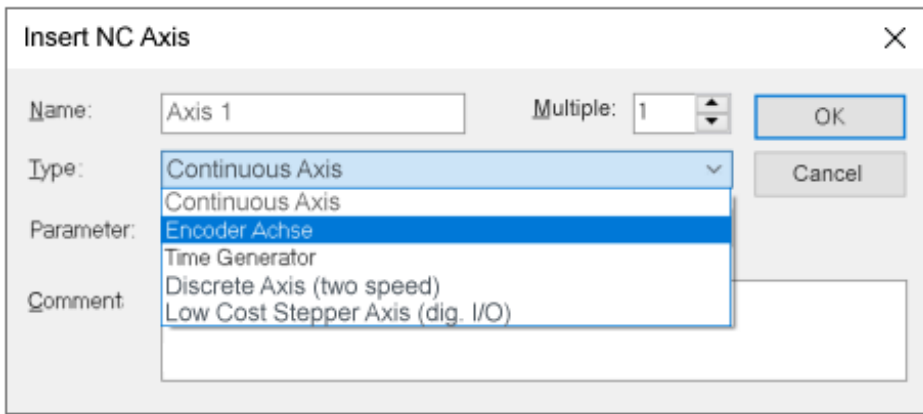


Fig. 136: Entering a name for the axis and selecting a type

**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...*

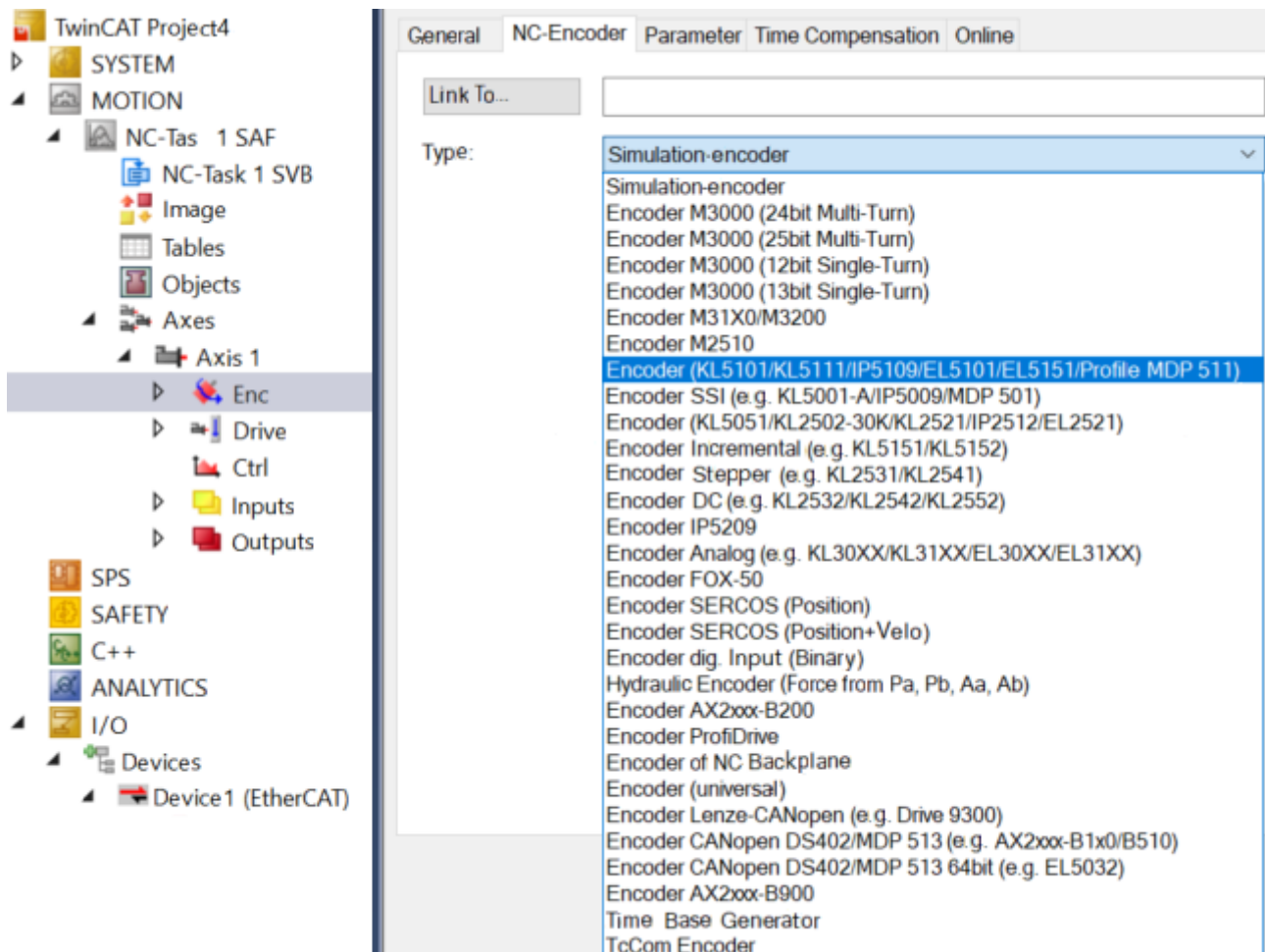


Fig. 137: Selecting the encoder

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

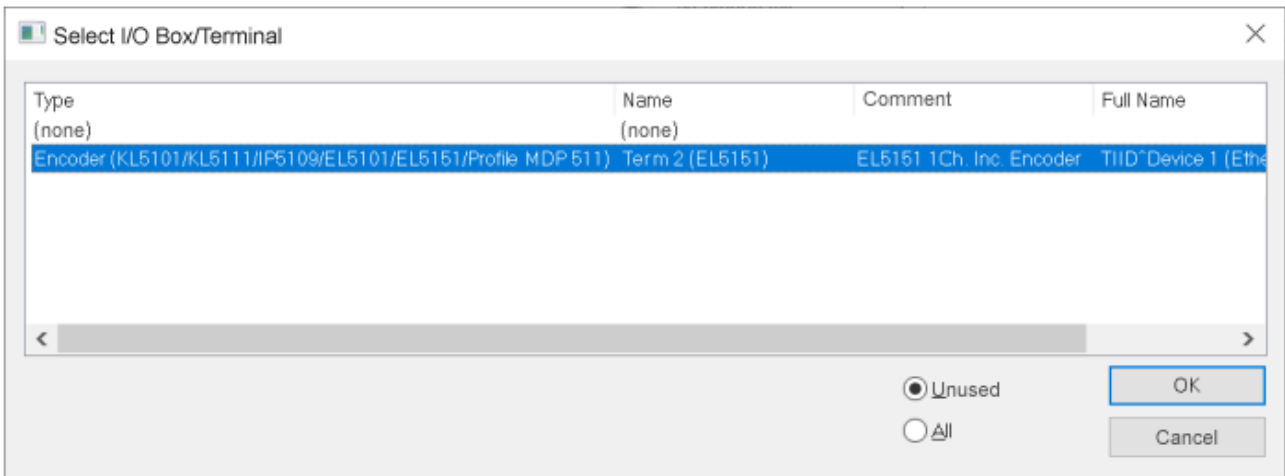


Fig. 138: Selecting and confirming an encoder terminal

**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*)

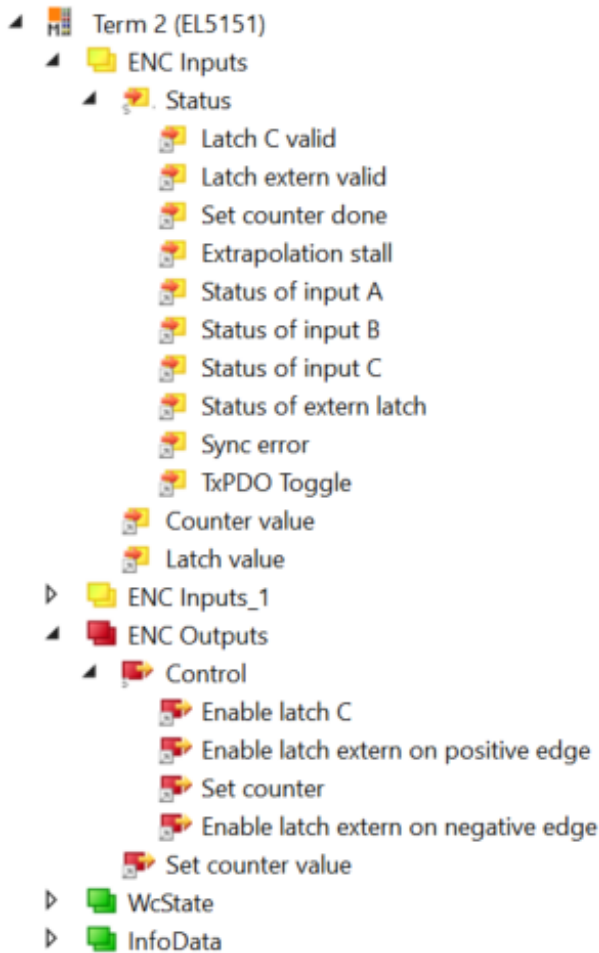


Fig. 139: Inputs and outputs of the EL5151 linked to the NC task

## 5.3 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.

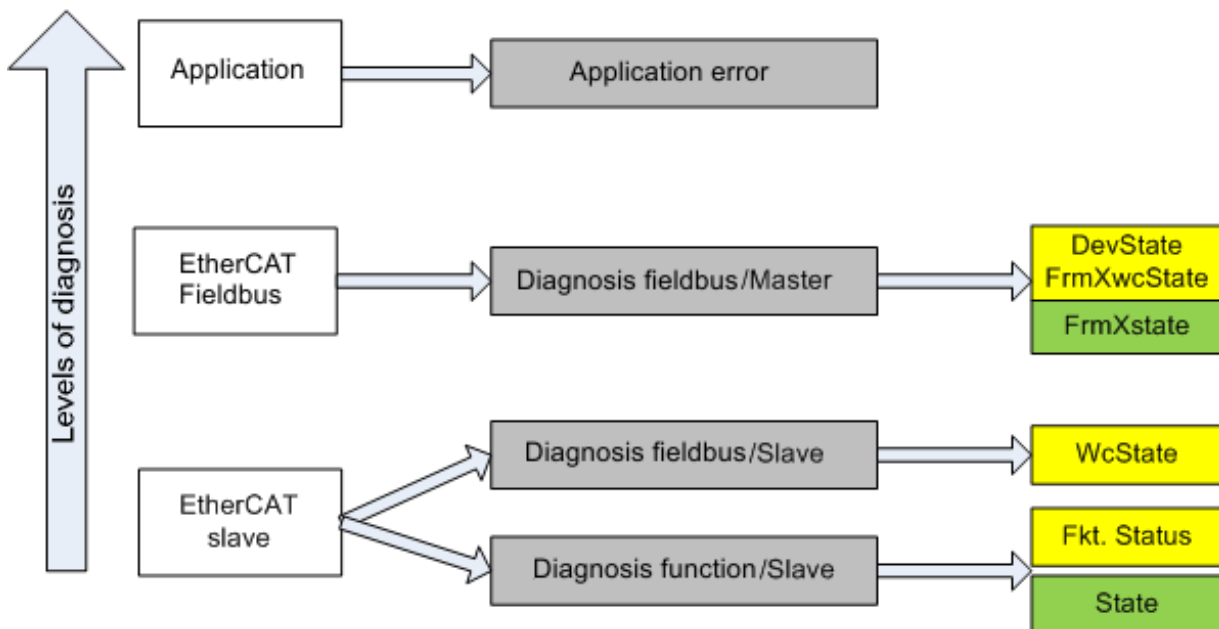


Fig. 140: Selection of the diagnostic information of an EtherCAT Slave

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.

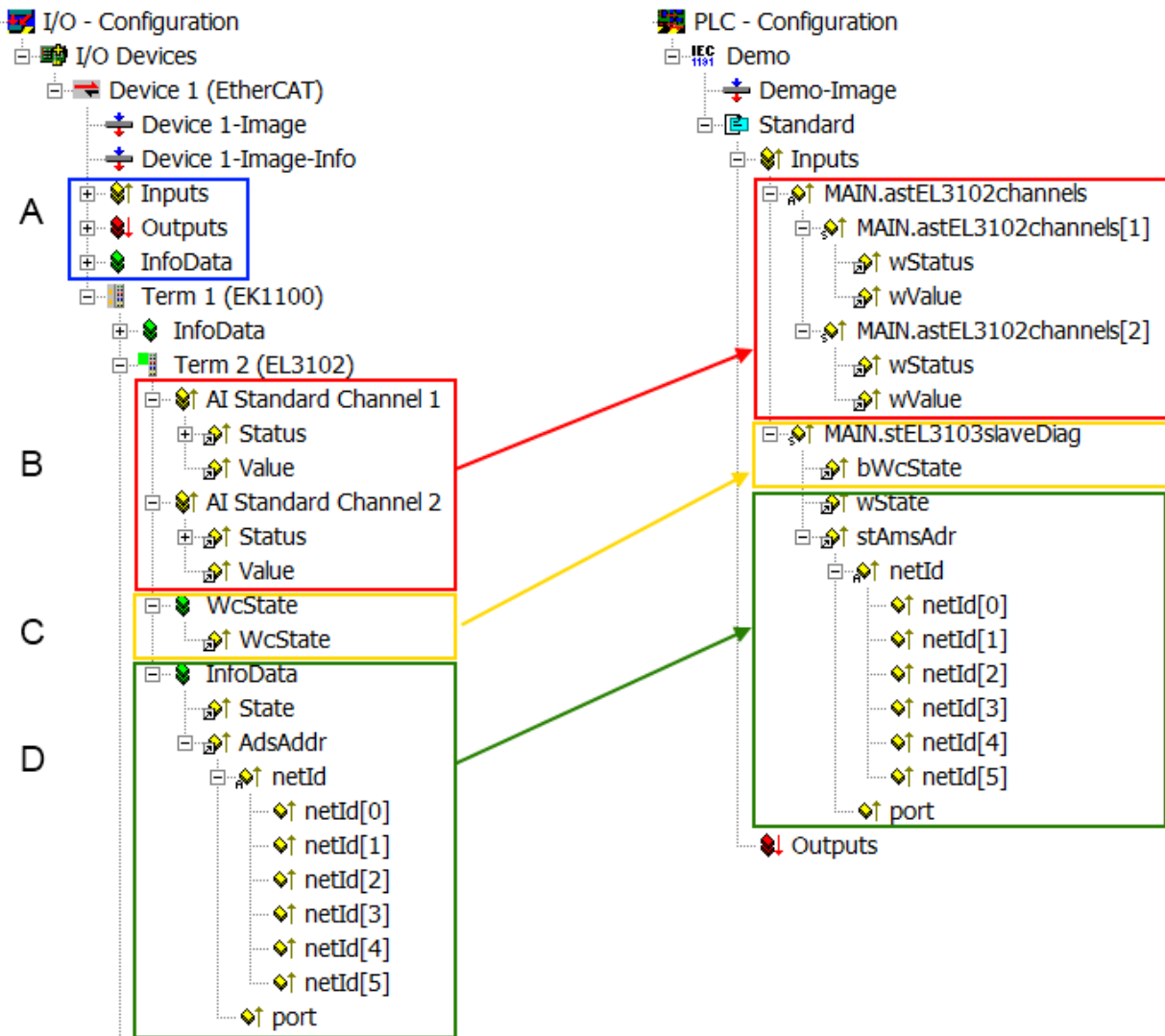


Fig. 141: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated acyclically (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: <ul style="list-style-type: none"> <li>• CoE in the Master for communication with/through the Slaves</li> <li>• Functions from <i>TcEtherCAT.lib</i></li> <li>• Perform an OnlineScan</li> </ul>
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 <ul style="list-style-type: none"> <li>• the bit significations may be found in the device documentation</li> <li>• other devices may supply more information, or none that is typical of a slave</li> </ul>	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 <ol style="list-style-type: none"> <li>1. at the EtherCAT Slave, and, with identical contents</li> <li>2. as a collective variable at the EtherCAT Master (see Point A) for linking.</li> </ol>	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 <ul style="list-style-type: none"> <li>• is only rarely/never changed, except when the system starts up</li> <li>• is itself determined acyclically (e.g. EtherCAT Status)</li> </ul>	State current Status (INIT..OP) of the Slave. The Slave must be in OP (=8) when operating normally. <i>AdsAddr</i> 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 <i>port</i> (= 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*:

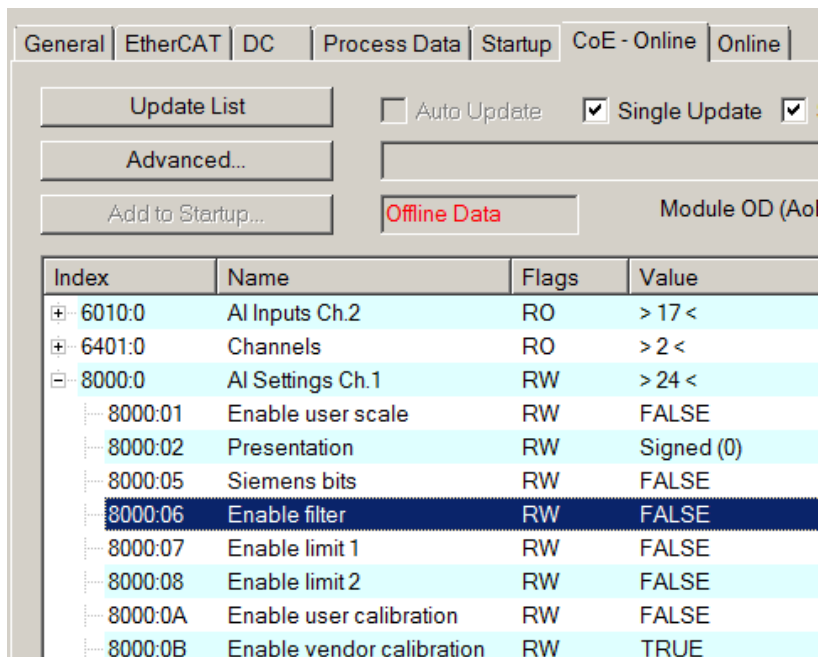


Fig. 142: EL3102, CoE directory

### **i** 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.

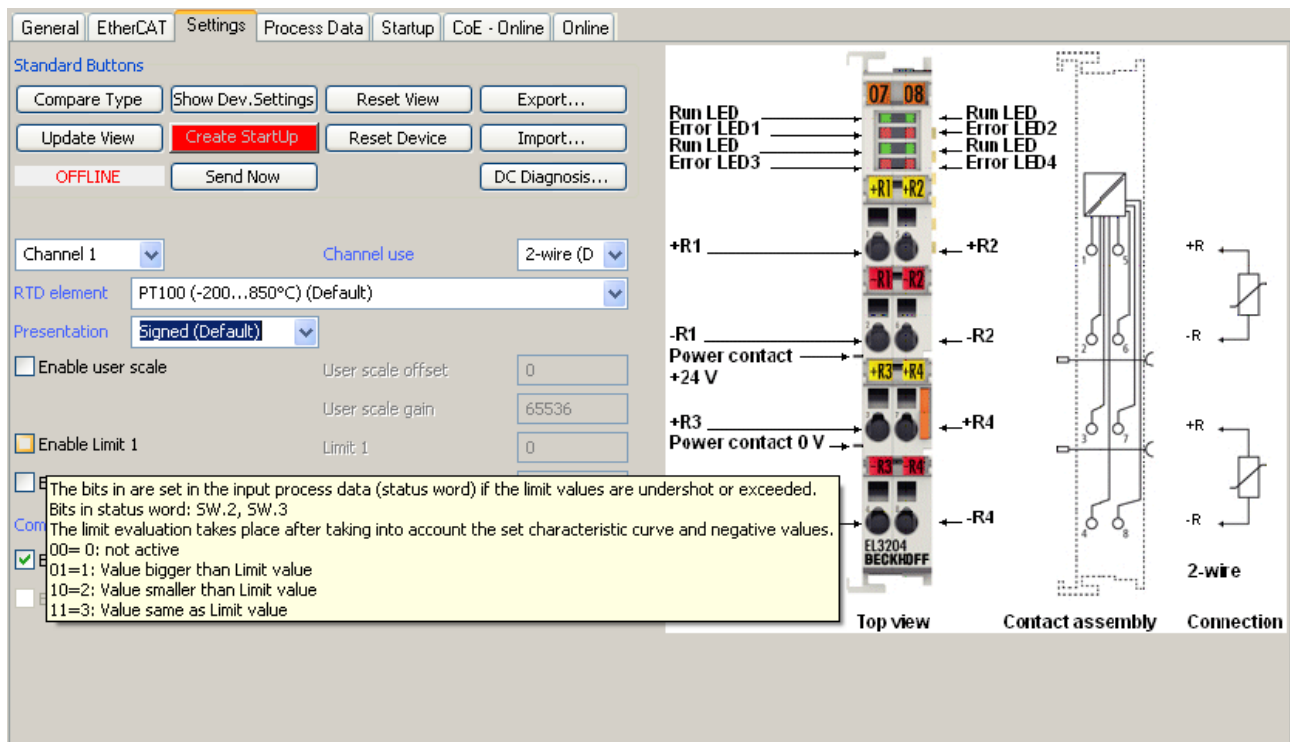


Fig. 143: Example of commissioning aid for a EL3204

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 \[► 28\]](#)" 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.

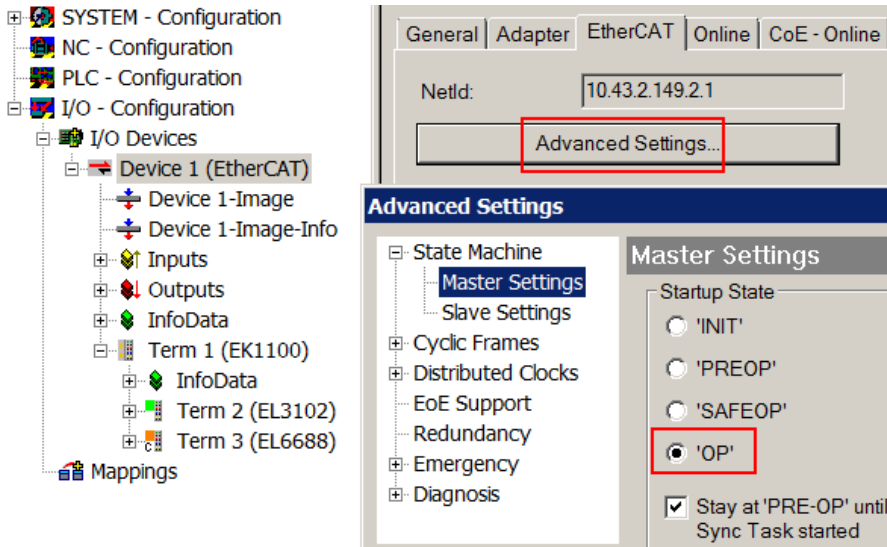


Fig. 144: Default behaviour of the System Manager

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

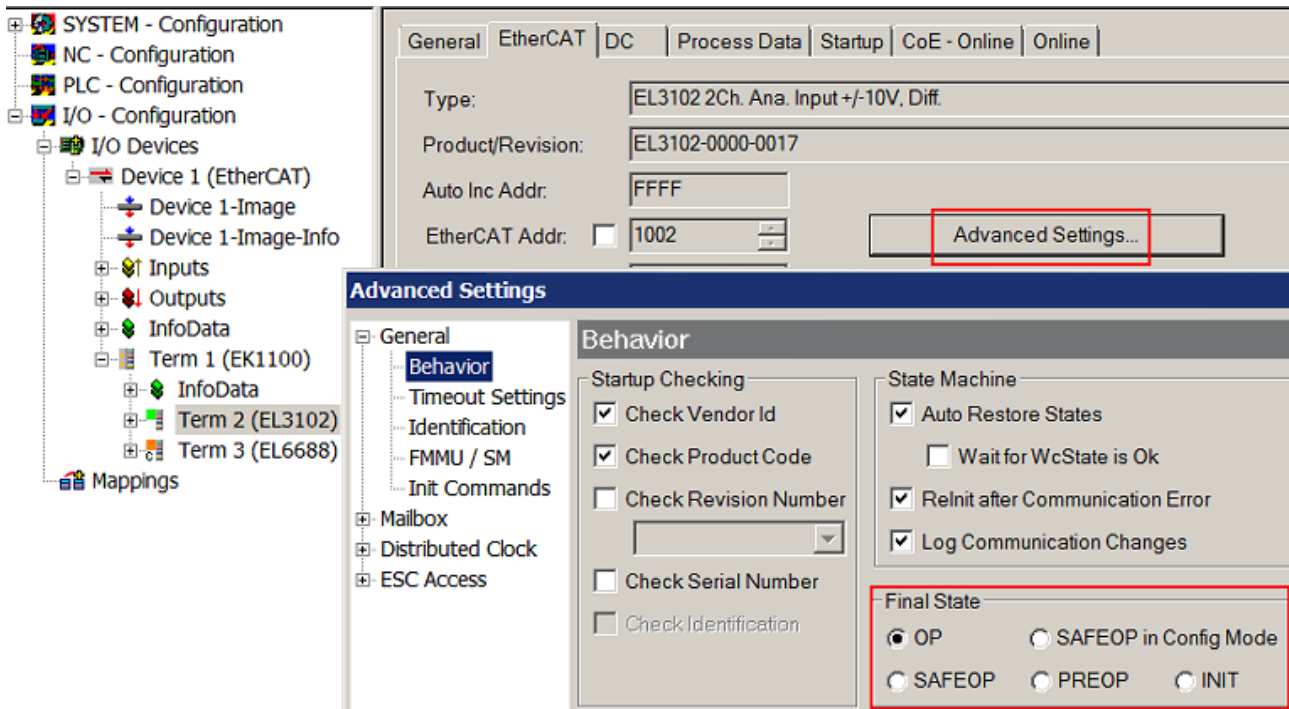


Fig. 145: Default target state in the Slave

**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.

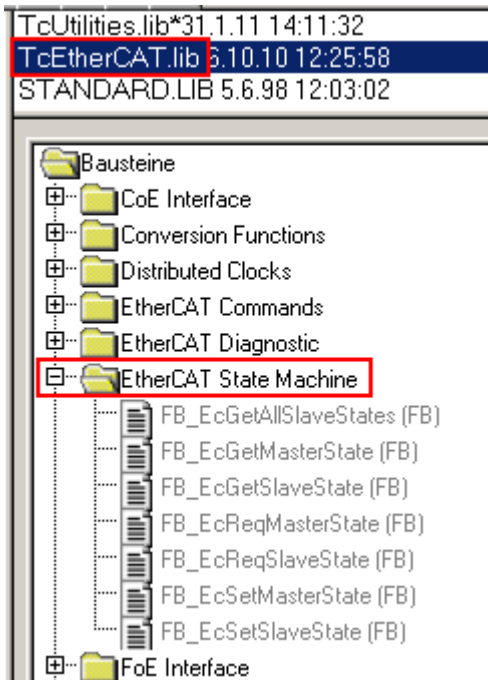


Fig. 146: PLC function blocks

**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.

General   Adapter   EtherCAT   Online   CoE - Online						
NetId:		10.43.2.149.2.1		Advanced Settings...		
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. 147: 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:

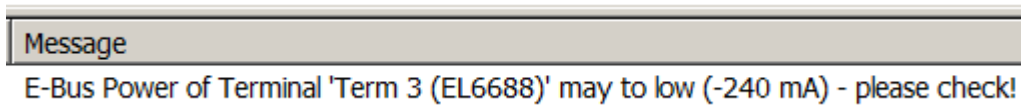


Fig. 148: Warning message for exceeding E-Bus current

<b>NOTICE</b>
<p><b>Caution! Malfunction possible!</b></p> <p>The same ground potential must be used for the E-Bus supply of all EtherCAT terminals in a terminal block!</p>

## 6 EL5122 - Commissioning

### 6.1 Overview of EL5122 functions

The EL5122 offers a wide range of functions. The following table provides an overview of the available functions. A detailed description can be found in the individual chapters.

To use the extended functions, the corresponding functions must be selected via the "Predefined PDO Assignment". The assignment is described in chapter "[Process data](#) [[124](#)]".

Basic functions	Description
<a href="#">Selecting the encoder type</a> [ <a href="#">134</a> ]	Encoders or counters/pulse generators with the signal levels according to TTL or Open Collector can be selected.
<a href="#">Encoder operating voltage</a> [ <a href="#">135</a> ]	The encoder supply is set centrally for both channels either to 5 V <sub>DC</sub> , 12 V <sub>DC</sub> or 24 V <sub>DC</sub> .
<a href="#">Evaluation of the counter value</a> [ <a href="#">135</a> ]	The input signal can be subject to 4-fold, 2-fold or 1-fold evaluation.
<a href="#">Counter limits</a> [ <a href="#">135</a> ]	The value range within which counting takes place can be defined.
<a href="#">Counter overflow / counter underflow</a> [ <a href="#">166</a> ]	If the count limit is exceeded or underrun, this is indicated in separate process data.
<a href="#">Counting direction</a> [ <a href="#">138</a> ]	The counting direction can be adapted to the application.
<a href="#">Counter value reset</a> [ <a href="#">139</a> ]	Allows repetitive resetting of the counter value via an edge (positive/negative) at the Gate/Latch input.
<a href="#">Set counter value</a> [ <a href="#">140</a> ]	The counter value can be set at runtime to a predefined counter value via a PLC variable or an edge (positive/negative) at the Gate/Latch input.
<a href="#">Save counter value</a> [ <a href="#">142</a> ]	The current counter value can be stored, independent of the cycle time, in a separate process datum via an edge (positive/negative) at the Gate/Latch input. You can parameterize whether the function is executed at each external edge or only once after each activation.
<a href="#">Lock counter value</a> [ <a href="#">144</a> ]	The counter value can be locked by an edge (positive/negative) on the Gate/Latch input or by a PLC variable.

Extended functionalities	Description
<a href="#">Frequency measurement</a> [ <a href="#">145</a> ]	The average frequency of the input signal within a given timeframe can be output directly.
<a href="#">Period value measurement</a> [ <a href="#">147</a> ]	The period value of the last period within the PLC cycle can be output directly.
<a href="#">Velocity, speed calculation</a> [ <a href="#">148</a> ]	The average velocity or speed of the input signal within a specified timeframe can be output directly.
<a href="#">Adjustable interference pulse filters</a> [ <a href="#">150</a> ]	To suppress interference, a filter can be set for each of the input signals.
<a href="#">Diagnostic data</a> [ <a href="#">155</a> ]	Error messages are communicated to the EtherCAT Master / TwinCAT via "DiagMessages".

Process data	Description
<a href="#">Operation modes</a> [ <a href="#">130</a> ]	The scope of the process data can be selected via "Predefined PDO Assignment".
<a href="#">Synchronicity mode</a> [ <a href="#">133</a> ]	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
<a href="#">Gate/Latch</a> [ <a href="#">154</a> ]	A Gate/Latch input is provided for 24 V <sub>DC</sub> signals with a min. pulse duration t <sub>ON</sub> of > 1 μs. This can be used for resetting, setting and storing the counter value.

## 6.2 Process data

### 6.2.1 Sync Manager (SM)

The scope of the offered process data can be changed via the "Process Data" tab (see illustration below).

A detailed description for setting the process data can be found in chapter [EtherCAT subscriber configuration](#) [▶ 103].

General | EtherCAT | DC | **Process Data** | Startup | CoE - Online | Diag History | Online

Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	12	Outputs	
3	24	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	12.0	ENC Status Channel 1	F	3	0
0x1A01	8.0	ENC Status Compact Channel 1	F		0
0x1A02	8.0	ENC Status Counter Channel 1	F		0
0x1A03	6.0	ENC Status Compact Counter Channel 1	F		0
0x1A04	4.0	ENC Frequency Channel 1	F		0
0x1A05	2.0	ENC Frequency Compact Channel 1	F		0
0x1A06	4.0	ENC Period Channel 1	F		0
0x1A07	2.0	ENC Period Compact Channel 1	F		0
0x1A08	12.0	ENC Status Channel 2	F	3	0
0x1A09	8.0	ENC Status Compact Channel 2	F		0
0x1A0A	8.0	ENC Status Counter Channel 2	F		0
0x1A0B	6.0	ENC Status Compact Counter Channel 2	F		0
0x1A0C	4.0	ENC Frequency Channel 2	F		0
0x1A0D	2.0	ENC Frequency Compact Channel 2	F		0

PDO Assignment (0x1C13):

- 0x1A00
- 0x1A01 (excluded by 0x1A00)
- 0x1A02 (excluded by 0x1A00)
- 0x1A03 (excluded by 0x1A00)
- 0x1A04
- 0x1A05
- 0x1A06
- 0x1A07
- 0x1A08
- 0x1A09 (excluded by 0x1A08)
- 0x1A0A (excluded by 0x1A08)
- 0x1A0B (excluded by 0x1A08)
- 0x1A0C
- 0x1A0D

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	D
--	0.1	0.0	--		
0x6000:02	0.1	0.1	Status__Latch extem valid	BOOL	
0x6000:03	0.1	0.2	Status__Set counter done	BOOL	
0x6000:04	0.1	0.3	Status__Counter underflow	BOOL	
0x6000:05	0.1	0.4	Status__Counter overflow	BOOL	
--	0.1	0.5	--		
--	0.1	0.6	--		
--	0.1	0.7	--		
0x6000:09	0.1	1.0	Status__Status of input A	BOOL	
0x6000:0A	0.1	1.1	Status__Status of input B	BOOL	
--	0.1	1.2	--		

Download

- PDO Assignment
- PDO Configuration

Predefined PDO Assignment: '2.Ch Standard'

Load PDO info from device

Sync Unit Assignment...

Fig. 149: EL5122 - Process data tab SM3 (default)

## 6.2.2 PDO assignment

The EL5122 offers a wide range of functions and thus also a large volume of process data.

### 6.2.2.1 SM3 - Inputs (0x1A00 .. 0x1A0F)

#### Channel 1 (0x1A00 .. 0x1A07):

0x1A00 - ENC Status Channel 1 (12.0)	
Contents Index - name size (byte.bit)	Excluded PDOs Index - name size (byte.bit)
0x6000:02 [▶ 172] - Status_Latch extern valid (0.1)	0x1A01 [▶ 182] - ENC Status Compact Channel 1 (8.0) 0x1A02 [▶ 183] - ENC Status Counter Channel 1 (8.0) 0x1A03 [▶ 183] - ENC Status Compact Counter Channel 1 (6.0)
0x6000:03 [▶ 172] - Status_Set counter done (0.1)	
0x6000:04 [▶ 172] - Status_Counter underflow (0.1)	
0x6000:05 [▶ 172] - Status_Counter overflow (0.1)	
0x6000:09 [▶ 172] - Status_Status of input A (0.1)	
0x6000:0A [▶ 172] - Status_Status of input B (0.1)	
0x6000:0C [▶ 172] - Status_Status of input gate (0.1)	
0x6002:0D [▶ 172] - Status_Status Diag (0.1)	
0x6002:0E [▶ 172] - Status_Status TxPDO State (0.1)	
0x6002:0F [▶ 172] - Status_Input cycle counter (0.2)	
0x6002:11 [▶ 172] - Status_Software gate valid (0.1)	
0x6002:15 [▶ 172] - Status_Counter value out of range (0.1)	
0x6000:11 [▶ 172] - Counter value (4.0)	
0x6000:12 [▶ 172] - Latch value (4.0)	

0x1A01 - ENC Status Compact Channel 1 (8.0)	
Contents Index - name size (byte.bit)	Excluded PDOs Index - name size (byte.bit)
0x6000:02 [▶ 172] - Status_Latch extern valid (0.1)	0x1A00 [▶ 181] - ENC Status Channel 1 (12.0) 0x1A02 [▶ 183] - ENC Status Counter Channel 1 (8.0) 0x1A03 [▶ 183] - ENC Status Compact Counter Channel 1 (6.0)
0x6000:03 [▶ 172] - Status_Set counter done (0.1)	
0x6000:04 [▶ 172] - Status_Counter underflow (0.1)	
0x6000:05 [▶ 172] - Status_Counter overflow (0.1)	
0x6000:09 [▶ 172] - Status_Status of input A (0.1)	
0x6000:0A [▶ 172] - Status_Status of input B (0.1)	
0x6000:0C [▶ 172] - Status_Status of input gate (0.1)	
0x6002:0D [▶ 172] - Status_Status Diag (0.1)	
0x6002:0E [▶ 172] - Status_Status TxPDO State (0.1)	
0x6002:0F [▶ 172] - Status_Input cycle counter (0.2)	
0x6002:11 [▶ 172] - Status_Software gate valid (0.1)	
0x6002:15 [▶ 172] - Status_Counter value out of range (0.1)	
0x6000:11 [▶ 172] - Counter value (2.0)	
0x6000:12 [▶ 172] - Latch value (2.0)	

0x1A02 - ENC Status Counter Channel 1 (8.0)	
Contents Index - name size (byte.bit)	Excluded PDOs Index - name size (byte.bit)
0x6000:03 [▶ 172] - Status_Set counter done (0.1)	0x1A00 [▶ 181] - ENC Status Channel 1 (12.0) 0x1A01 [▶ 182] - ENC Status Compact Channel 1 (8.0) 0x1A03 [▶ 183] - ENC Status Compact Counter Channel 1 (6.0)
0x6002:0D [▶ 172] - Status_Status Diag (0.1)	
0x6002:0E [▶ 172] - Status_Status TxPDO State (0.1)	
0x6002:0F [▶ 172] - Status_Input cycle counter (0.2)	
0x6002:11 [▶ 172] - Status_Software gate valid (0.1)	
0x6000:11 [▶ 172] - Counter value (4.0)	

<b>0x1A03 - ENC Status Compact Counter Channel 1 (6.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index name size (byte.bit)</b>
0x6000:03 [▶ 172] - Status_Set counter done (0.1)	0x1A00 [▶ 181] - ENC Status Channel 1 (12.0)
0x6002:0D [▶ 172] - Status_Status Diag (0.1)	0x1A01 [▶ 182] - ENC Status Compact Channel 1 (8.0)
0x6002:0E [▶ 172] - Status_Status TxPDO State (0.1)	0x1A02 [▶ 183] - ENC Status Counter Channel 1 (8.0)
0x6002:0F [▶ 172] - Status_Input cycle counter (0.2)	
0x6002:11 [▶ 172] - Status_Software gate valid (0.1)	
0x6000:11 [▶ 172] - Counter value (2.0)	

<b>0x1A04 - ENC Frequency Channel 1 (4.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index - name size (byte.bit)</b>
0x6000:13 [▶ 172] - Frequency value (4.0)	0x1A05 [▶ 183] - ENC Frequency Compact Channel 1 (4.0)

<b>0x1A05 - ENC Frequency Compact Channel 1 (2.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index - name size (byte.bit)</b>
0x6000:11 [▶ 172] - Counter value (2.0)	0x1A04 [▶ 183] - ENC Frequency Channel 1 (4.0)

<b>0x1A06 - ENC Period Channel 1 (4.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index name size (byte.bit)</b>
0x6000:14 [▶ 172] - Period value (4.0)	0x1A07 [▶ 184] - ENC Period Compact Channel 1 (2.0)

<b>0x1A07 - ENC Period Compact Channel 1 (2.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index name size (byte.bit)</b>
0x6000:14 [▶ 172] - Period value (2.0)	0x1A06 [▶ 184] - ENC Period Channel 1 (4.0)

### Channel 2 (0x1A08 .. 0x1A0F):

<b>0x1A08 - ENC Status Channel 2 (12.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index name size (byte.bit)</b>
0x6010:02 [▶ 172] - Status_Latch extern valid (0.1)	0x1A09 [▶ 185] - ENC Status Compact Channel 2 (8.0)
0x6010:03 [▶ 172] - Status_Set counter done (0.1)	0x1A0A [▶ 186] - ENC Status Counter Channel 2 (8.0)
0x6010:04 [▶ 172] - Status_Counter underflow (0.1)	0x1A0B [▶ 186] - ENC Status Compact Counter Channel 2 (6.0)
0x6010:05 [▶ 172] - Status_Counter overflow (0.1)	
0x6010:09 [▶ 172] - Status_Status of input A (0.1)	
0x6010:0A [▶ 172] - Status_Status of input B (0.1)	
0x6010:0C [▶ 172] - Status_Status of input gate (0.1)	
0x6012:0D [▶ 172] - Status_Status Diag (0.1)	
0x6012:0E [▶ 172] - Status_Status TxPDO State (0.1)	
0x6012:0F [▶ 172] - Status_Input cycle counter (0.2)	
0x6012:11 [▶ 172] - Status_Software gate valid (0.1)	
0x6012:15 [▶ 172] - Status_Counter value out of range (0.1)	
0x6010:11 [▶ 172] - Counter value (4.0)	
0x6010:12 [▶ 172] - Latch value (4.0)	

<b>0x1A09 - ENC Status Compact Channel 2 (8.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index name size (byte.bit)</b>
0x6010:02 [ <a href="#">172</a> ] - Status_Latch extern valid (0.1)	0x1A08 [ <a href="#">184</a> ] - ENC Status Channel 2 (12.0) 0x1A0A [ <a href="#">186</a> ] - ENC Status Counter Channel 2 (8.0) 0x1A0B [ <a href="#">186</a> ] - ENC Status Compact Counter Channel 2 (6.0)
0x6010:03 [ <a href="#">172</a> ] - Status_Set counter done (0.1)	
0x6010:04 [ <a href="#">172</a> ] - Status_Counter underflow (0.1)	
0x6010:05 [ <a href="#">172</a> ] - Status_Counter overflow (0.1)	
0x6010:09 [ <a href="#">172</a> ] - Status_Status of input A (0.1)	
0x6010:0A [ <a href="#">172</a> ] - Status_Status of input B (0.1)	
0x6010:0C [ <a href="#">172</a> ] - Status_Status of input gate (0.1)	
0x6012:0D [ <a href="#">172</a> ] - Status_Status Diag (0.1)	
0x6012:0E [ <a href="#">172</a> ] - Status_Status TxPDO State (0.1)	
0x6012:0F [ <a href="#">172</a> ] - Status_Input cycle counter (0.2)	
0x6012:11 [ <a href="#">172</a> ] - Status_Software gate valid (0.1)	
0x6012:15 [ <a href="#">172</a> ] - Status_Counter value out of range (0.1)	
0x6010:11 [ <a href="#">172</a> ] - Counter value (2.0)	
0x6010:12 [ <a href="#">172</a> ] - Latch value (2.0)	

<b>0x1A0A - ENC Status Counter Channel 2 (8.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index name size (byte.bit)</b>
0x6010:03 [ <a href="#">172</a> ] - Status_Set counter done (0.1)	0x1A08 [ <a href="#">184</a> ] - ENC Status Channel 2 (12.0) 0x1A09 [ <a href="#">185</a> ] - ENC Status Compact Channel 2 (8.0) 0x1A0B [ <a href="#">186</a> ] - ENC Status Compact Counter Channel 2 (6.0)
0x6012:0D [ <a href="#">172</a> ] - Status_Status Diag (0.1)	
0x6012:0E [ <a href="#">172</a> ] - Status_Status TxPDO State (0.1)	
0x6012:0F [ <a href="#">172</a> ] - Status_Input cycle counter (0.2)	
0x6012:11 [ <a href="#">172</a> ] - Status_Software gate valid (0.1)	
0x6010:11 [ <a href="#">172</a> ] - Counter value (4.0)	

<b>0x1A0B - ENC Status Compact Counter Channel 2 (6.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index name size (byte.bit)</b>
0x6010:03 [ <a href="#">172</a> ] - Status_Set counter done (0.1)	0x1A08 [ <a href="#">184</a> ] - ENC Status Channel 2 (12.0) 0x1A09 [ <a href="#">185</a> ] - ENC Status Compact Channel 2 (8.0) 0x1A0A [ <a href="#">186</a> ] - ENC Status Counter Channel 2 (8.0)
0x6012:0D [ <a href="#">172</a> ] - Status_Status Diag (0.1)	
0x6012:0E [ <a href="#">172</a> ] - Status_Status TxPDO State (0.1)	
0x6012:0F [ <a href="#">172</a> ] - Status_Input cycle counter (0.2)	
0x6012:11 [ <a href="#">172</a> ] - Status_Software gate valid (0.1)	
0x6010:11 [ <a href="#">172</a> ] - Counter value (4.0)	

<b>0x1A0C - ENC Frequency Channel 2 (4.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index name size (byte.bit)</b>
0x6010:13 [ <a href="#">172</a> ] - Frequency value (4.0)	0x1A0D [ <a href="#">186</a> ] - ENC Frequency Compact Channel 2 (2.0)

<b>0x1A0D - ENC Frequency Compact Channel 2 (2.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index - name size (byte.bit)</b>
0x6010:13 [ <a href="#">172</a> ] - Frequency value (2.0)	0x1A0C [ <a href="#">186</a> ] - ENC Frequency Channel 2 (4.0)

<b>0x1A0E - ENC Period Channel 2 (4.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index - name size (byte.bit)</b>
0x6010:14 [ <a href="#">172</a> ] - Period value (4.0)	0x1A0F [ <a href="#">187</a> ] - Period Compact Channel 2 (2.0)

<b>0x1A0F - ENC Period Compact Channel 2 (2.0)</b>	
<b>Contents</b>	<b>Excluded PDOs</b>
<b>Index - name size (byte.bit)</b>	<b>Index - name size (byte.bit)</b>
0x6010:14 [ <a href="#">172</a> ] - Period value (2.0)	0x1A0E [ <a href="#">187</a> ] - ENC Period Channel 2 (4.0)



## 6.2.2.2 SM2 - Outputs (0x1600 .. 0x1607)

### Channel 1 (0x1600 .. 0x1603):

0x1600 - ENC Control Channel 1 (6.0)	
Contents Index - name size (byte.bit)	Excluded PDOs Index - name size (byte.bit)
0x7000:02 [▶ 173] - Control_Enable latch extern on positive edge (0.1)	0x1601 [▶ 176] - ENC Control Compact Channel 1 (4.0) 0x1602 [▶ 177] - ENC Control Counter Channel 1 (6.0) 0x1603 [▶ 177] - ENC Control Compact Counter Channel 1 (4.0)
0x7000:03 [▶ 173] - Control_Set counter (0.1)	
0x7000:04 [▶ 173] - Control_Enable latch extern on negative edge (0.1)	
0x7000:09 [▶ 173] - Control_Set software gate (0.1)	
0x7000:0A [▶ 173] - Control_Set counter on latch extern on positive edge (0.1)	
0x7000:0B [▶ 173] - Control_Set counter on latch extern on negative edge (0.1)	
0x7000:11 [▶ 173] - Set counter value (4.0)	

0x1601 - ENC Control Compact Channel 1 (4.0)	
Contents Index - name size (byte.bit)	Excluded PDOs Index - name size (byte.bit)
0x7000:02 [▶ 173] - Control_Enable latch extern on positive edge (0.1)	0x1600 [▶ 176] - ENC Control Channel 1 (6.0) 0x1602 [▶ 177] - ENC Control Counter Channel 1 (6.0) 0x1603 [▶ 177] - ENC Control Compact Counter Channel 1 (4.0)
0x7000:03 [▶ 173] - Control_Set counter (0.1)	
0x7000:04 [▶ 173] - Control_Enable latch extern on negative edge (0.1)	
0x7000:09 [▶ 173] - Control_Set software gate (0.1)	
0x7000:0A [▶ 173] - Control_Set counter on latch extern on positive edge (0.1)	
0x7000:0B [▶ 173] - Control_Set counter on latch extern on negative edge (0.1)	
0x7000:11 [▶ 173] - Set counter value (2.0)	

0x1602 - ENC Control Counter Channel 1 (6.0)	
Contents Index - name size (byte.bit)	Excluded PDOs Index - name size (byte.bit)
0x7000:03 [▶ 173] - Control_Set counter (0.1)	0x1600 [▶ 176] - ENC Control Channel 1 (6.0) 0x1601 [▶ 176] - ENC Control Compact Channel 1 (4.0) 0x1603 [▶ 177] - ENC Control Compact Counter Channel 1 (4.0)
0x7000:09 [▶ 173] - Control_Set software gate (0.1)	
0x7000:11 [▶ 173] - Set counter value (4.0)	

0x1603 - ENC Control Compact Counter Channel 1 (4.0)	
Contents Index - name size (byte.bit)	Excluded PDOs Index - name size (byte.bit)
0x7000:03 [▶ 173] - Control_Set counter (0.1)	0x1600 [▶ 176] - ENC Control Channel 1 (6.0) 0x1601 [▶ 176] - ENC Control Compact Channel 1 (4.0) 0x1602 [▶ 177] - ENC Control Counter Channel 1 (6.0)
0x7000:09 [▶ 173] - Control_Set software gate (0.1)	
0x7000:11 [▶ 173] - Set counter value (2.0)	

### Channel 2 (0x1604 .. 0x1607):

0x1604 - ENC Control Channel 2 (6.0)	
Contents Index - name size (byte.bit)	Excluded PDOs Index - name size (byte.bit)
0x7010:02 [▶ 173] - Control_Enable latch extern on positive edge (0.1)	0x1605 [▶ 178] - ENC Control Compact Channel 2 (4.0) 0x1606 [▶ 178] - ENC Control Counter Channel 2 (6.0) 0x1607 [▶ 178] - ENC Control Compact Counter Channel 2 (4.0)
0x7010:03 [▶ 173] - Control_Set counter (0.1)	
0x7010:04 [▶ 173] - Control_Enable latch extern on negative edge (0.1)	
0x7010:09 [▶ 173] - Control_Set software gate (0.1)	
0x7010:0A [▶ 173] - Control_Set counter on latch extern on positive edge (0.1)	
0x7010:0B [▶ 173] - Control_Set counter on latch extern on negative edge (0.1)	
0x7010:11 [▶ 173] - Set counter value (4.0)	



<b>0x1605 - ENC Control Compact Channel 2 (4.0)</b>	
<b>Contents</b> Index - name size (byte.bit)	<b>Excluded PDOs</b> Index - name size (byte.bit)
0x7010:02 [ <a href="#">▶ 173</a> ] - Control_Enable latch extern on positive edge (0.1)	0x1604 [ <a href="#">▶ 177</a> ] - ENC Control Channel 2 (6.0)
0x7010:03 [ <a href="#">▶ 173</a> ] - Control_Set counter (0.1)	0x1606 [ <a href="#">▶ 178</a> ] - ENC Control Counter Channel 2 (6.0)
0x7010:04 [ <a href="#">▶ 173</a> ] - Control_Enable latch extern on negative edge (0.1)	0x1607 [ <a href="#">▶ 178</a> ] - ENC Control Compact Counter Channel 2 (4.0)
0x7010:09 [ <a href="#">▶ 173</a> ] - Control_Set software gate (0.1)	
0x7010:0A [ <a href="#">▶ 173</a> ] - Control_Set counter on latch extern on positive edge (0.1)	
0x7010:0B [ <a href="#">▶ 173</a> ] - Control_Set counter on latch extern on negative edge (0.1)	
0x7010:11 [ <a href="#">▶ 173</a> ] - Set counter value (2.0)	

<b>0x1606 - ENC Control Counter Channel 2 (6.0)</b>	
<b>Contents</b> Index - name size (byte.bit)	<b>Excluded PDOs</b> Index - name size (byte.bit)
0x7010:03 [ <a href="#">▶ 173</a> ] - Control_Set counter (0.1)	0x1604 [ <a href="#">▶ 177</a> ] - ENC Control Channel 2 (6.0)
0x7010:09 [ <a href="#">▶ 173</a> ] - Control_Set software gate (0.1)	0x1605 [ <a href="#">▶ 178</a> ] - ENC Control Compact Channel 2 (4.0)
0x7010:11 [ <a href="#">▶ 173</a> ] - Set counter value (4.0)	0x1607 [ <a href="#">▶ 178</a> ] - ENC Control Compact Counter Channel 2 (4.0)

<b>0x1607 - ENC Control Compact Counter Channel 2 (4.0)</b>	
<b>Contents</b> Index - name size (byte.bit)	<b>Excluded PDOs</b> Index - name size (byte.bit)
0x7010:03 [ <a href="#">▶ 173</a> ] - Control_Set counter (0.1)	0x1604 [ <a href="#">▶ 177</a> ] - ENC Control Channel 2 (6.0)
0x7010:09 [ <a href="#">▶ 173</a> ] - Control_Set software gate (0.1)	0x1605 [ <a href="#">▶ 178</a> ] - ENC Control Compact Channel 2 (4.0)
0x7010:11 [ <a href="#">▶ 173</a> ] - Set counter value (2.0)	0x1606 [ <a href="#">▶ 178</a> ] - ENC Control Counter Channel 2 (6.0)

### 6.2.3 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 activated and the unnecessary PDOs are deactivated.

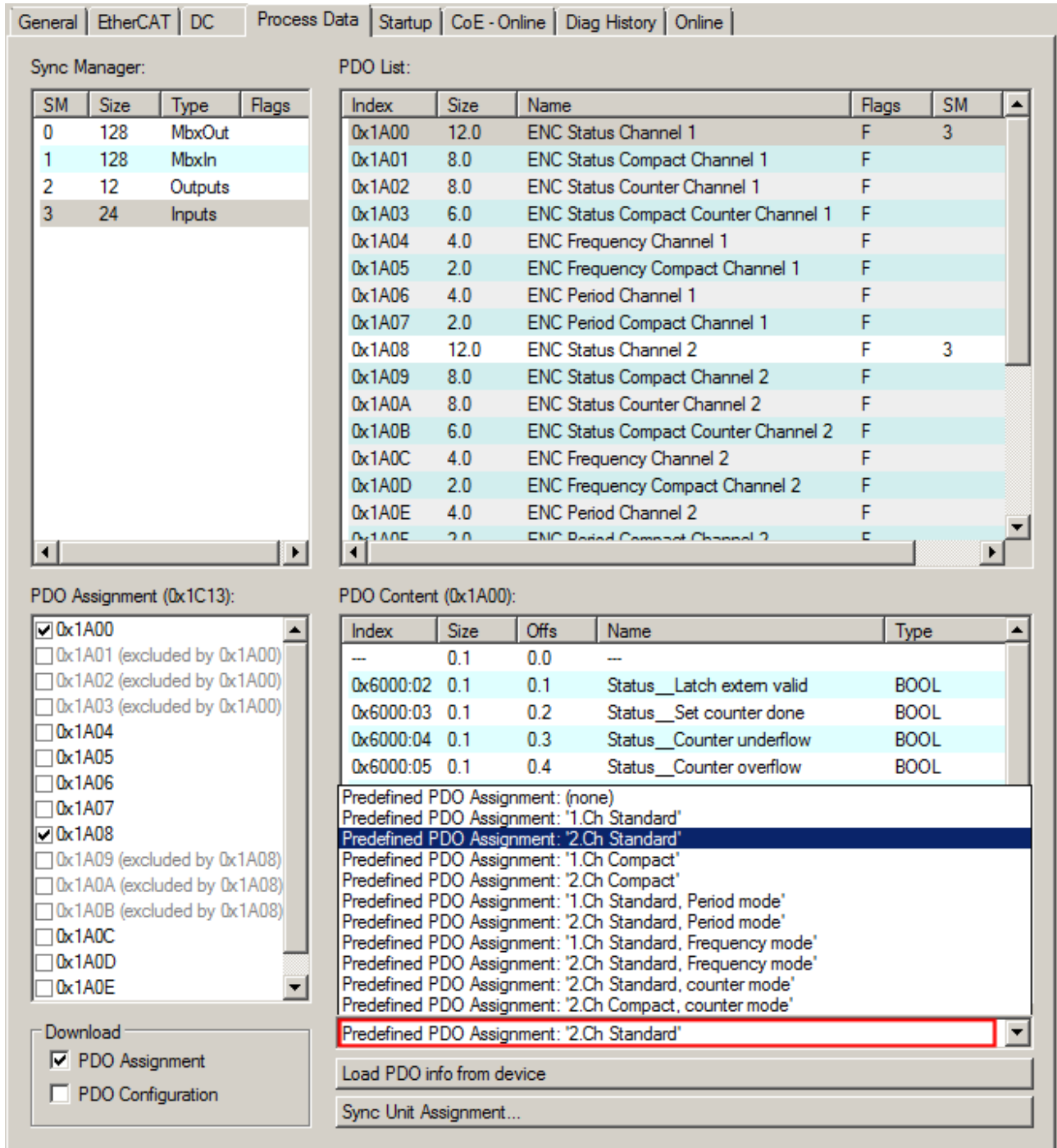


Fig. 150: EL5122 - Process data, Predefined PDO (default: Ch 2 Standard)

There is a choice of 10 PDO assignments for operation in the Standard, Compact and Counter modes:

**Predefined PDO Assignments for Compact mode**

Predefined PDO Assignment	PDO Assignment
1.Ch Compact	<b>SM3:</b> 0x1A01 [▶ 125] - ENC Status Compact Channel 1 (8.0) <b>SM2:</b> 0x1601 [▶ 128] - ENC Control Compact Channel 1 (4.0)
2.Ch Compact	<b>SM3:</b> 0x1A01 [▶ 125] - ENC Status Compact Channel 1 (8.0) 0x1A09 [▶ 127] - ENC Status Compact Channel 2 (8.0) <b>SM2:</b> 0x1601 [▶ 128] - ENC Control Compact Channel 1 (4.0) 0x1605 [▶ 178] - ENC Control Compact Channel 2 (4.0)

**Predefined PDO Assignments for Standard mode**

Predefined PDO Assignment	PDO Assignment
1.Ch Standard	<b>SM3:</b> 0x1A00 [▶ 125] - ENC Status Channel 1 (12.0) <b>SM2:</b> 0x1600 [▶ 128] - ENC Control Channel 1 (6.0)
2.Ch Standard	<b>SM3:</b> 0x1A00 [▶ 125] - ENC Status Channel 1 (12.0) 0x1A08 [▶ 126] - ENC Status Channel 2 (12.0) <b>SM2:</b> 0x1600 [▶ 128] - ENC Control Channel 1 (6.0) 0x1604 [▶ 128] - ENC Control Channel 2 (6.0)
1.Ch Standard, Period mode	<b>SM3:</b> 0x1A00 [▶ 125] - ENC Status Channel 1 (12.0) 0x1A06 [▶ 126] - ENC Period Channel 1 (4.0) <b>SM2:</b> 0x1600 [▶ 128] - ENC Control Channel 1 (6.0)
2.Ch Standard, Period mode	<b>SM3:</b> 0x1A00 [▶ 125] - ENC Status Channel 1 (12.0) 0x1A06 [▶ 126] - ENC Period Channel 1 (4.0) 0x1A08 [▶ 126] - ENC Status Channel 2 (12.0) 0x1A0E [▶ 127] - ENC Period Channel 2 (4.0) <b>SM2:</b> 0x1600 [▶ 128] - ENC Control Channel 1 (6.0) 0x1604 [▶ 128] - ENC Control Channel 2 (6.0)
1.Ch Standard, Frequency mode	<b>SM3:</b> 0x1A00 [▶ 125] - ENC Status Channel 1 (12.0) 0x1A04 [▶ 126] - ENC Frequency Channel 1 (4.0) <b>SM2:</b> 0x1600 [▶ 128] - ENC Control Channel 1 (6.0)
2.Ch Standard, Frequency mode	<b>SM3:</b> 0x1A00 [▶ 125] - ENC Status Channel 1 (12.0) 0x1A04 [▶ 126] - ENC Frequency Channel 1 (4.0) 0x1A08 [▶ 126] - ENC Status Channel 2 (12.0) 0x1A0C [▶ 127] - ENC Frequency Channel 2 (4.0) <b>SM2:</b> 0x1600 [▶ 128] - ENC Control Channel 1 (6.0) 0x1604 [▶ 128] - ENC Control Channel 2 (6.0)

**Predefined PDO Assignments for Counter mode**

Predefined PDO Assignment	PDO Assignment
2.Ch Standard, counter mode,	<b>SM3:</b> <u>0x1A02</u> [▶ 125] - ENC Status Counter Channel 1 (8.0) <u>0x1A0A</u> [▶ 127] - ENC Status Counter Channel 2 (8.0) <b>SM2:</b> <u>0x1602</u> [▶ 128] - ENC Control Counter Channel 1 (6.0) <u>0x1606</u> [▶ 129] - ENC Control Counter Channel 2 (6.0)
2.Ch Compact, counter mode,	<b>SM3:</b> <u>0x1A03</u> [▶ 126] - ENC Status Compact Counter Channel 1 (6.0) <u>0x1A0B</u> [▶ 127] - ENC Status Compact Counter Channel 2 (6.0) <b>SM2:</b> <u>0x1603</u> [▶ 128] - ENC Control Compact Counter Channel 1 (4.0) <u>0x1607</u> [▶ 129] - ENC Control Compact Counter Channel 2 (4.0)

## 6.2.4 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 principles](#).

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.

## 6.2.5 EtherCAT cycle time

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.

EL5122 Predefined PDO Assignment	EtherCAT cycle time	
	Min.	Max.
1. Ch. Standard	66.6 µs typ.	100 µs typ.
2. Ch. Standard	125 µs typ.	150 µs typ.
1. Ch. Compact	66.6 µs typ.	100 µs typ.
2. Ch. Compact	100 µs typ.	125 µs typ.
1. Ch. Standard, Periode mode	71.4 µs typ.	125 µs typ.
2. Ch. Standard, Periode mode	125 µs typ.	150 µs typ.
1. Ch. Standard, Frequency mode	71.4 µs typ.	125 µs typ.
2. Ch. Standard, Frequency mode	125 µs typ.	150 µs typ.
2. Ch. Standard, Counter mode	100 µs typ.	150 µs typ.
2. Ch. Compact, Counter mode	100 µs typ.	150 µs typ.

## 6.3 Basic functions

### 6.3.1 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 \[► 134\]](#) (Counter mode)
- [Evaluation of the counter value \[► 135\]](#) (Evaluation mode)
- [Determining the counter limits \[► 135\]](#) (Reset counter value / Limit counter value)
- [Reversion of rotation \[► 138\]](#)

#### 6.3.1.1 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"

Index (hex)	Name	Meaning
80n1:1D	Counter mode	Selecting the encoder
	2: Encoder TTL (single-ended)	TTL encoder without zero pulse
	3: Counter TTL (single-ended)	TTL counter/pulse generator with/without zero pulse. Direction preset via track B
	4: Encoder open collector	Open Collector encoder with/without zero pulse
	5: Counter open collector	Open collector counter/pulse generator without zero pulse, Direction preset via track B

In the following, the settings of the values 2 and 4 are generally referred to as "Encoder" and the settings of the values 3 and 5 are generally referred to as "Counter/pulse generator".

### 6.3.1.2 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 \[► 171\]](#) "Supply voltage". An operating voltage of 5 V<sub>DC</sub> is preset. Voltage values of 5 V<sub>DC</sub>, 12 V<sub>DC</sub> and 24 V<sub>DC</sub> can be selected. The setting applies to both channels. Before switching to higher voltages, ensure that both encoders support the voltage range.

The following tolerances apply

Voltage range	Tolerance
5 V <sub>DC</sub>	+/- 5% (4.75 V ... 5.25 V)
12 V <sub>DC</sub>	+/- 10% (10.8 V ... 13.2 V)
24 V <sub>DC</sub>	-15% to +20% (20.4 V ... 28.8 V)



#### Setting the encoder supply via index [0x8001:17 \[► 171\]](#)

The encoder supply is set centrally for both channels via the index [0x8001:17 \[► 171\]](#) (channel 1). The corresponding index [0x8011:17](#) of the second channel has no parameterization function.

#### NOTICE

##### Setting the encoder supply voltage

- Before switching to a higher voltage, make sure that the connected encoders support the selected voltage range!
- To write to [0x80n1:17](#) "Supply voltage" you have to set the value [0x72657375](#) (ASCII: "user") in index [0xF008 \[► 190\]](#) "Code word".

### 6.3.1.3 Evaluation of the counter value

Evaluation of the counter value is set through index [0x80n0:06](#) "Evaluation mode".

- Evaluation of the input signals (index [0x80n0:06](#) "Evaluation mode"): The input signal can be subject to 4-fold, 2-fold or 1-fold evaluation.
  - 1-fold evaluation: the rising edges at track A are counted.
  - 2-fold evaluation: the rising and falling edges at track A are counted.
  - 4-fold evaluation: the rising and falling edges at tracks A and B are counted.

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

### 6.3.1.4 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. The overflow of the counter is indicated by the bits "Counter overflow" (see chapter [Determining the counter limits \(Reset counter value / Limit counter value \[► 137\]\)](#)).

- PDO assignment "Standard": 0 to 2<sup>32</sup>-1
- PDO assignment "Compact": 0 to 2<sup>16</sup>-1

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

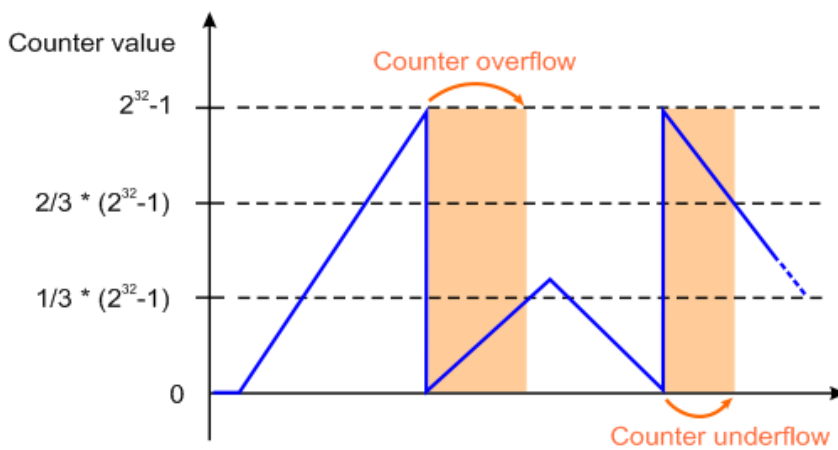


Fig. 151: Counter overflow and underflow in delivery state

### 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

**i** For writing to index 0x80n1:1B "Reset counter value" and index 0x80n1:1A "Limit counter value" the value 0x72657375 (ASCII: "user") must be set in 0xF008 [▶ 190] "Code word".

- Enter the lower counter limit in index 0x80n1:1B "Reset counter value".
- Enter the upper counter limit in index 0x80n1:1A "Limit counter value".

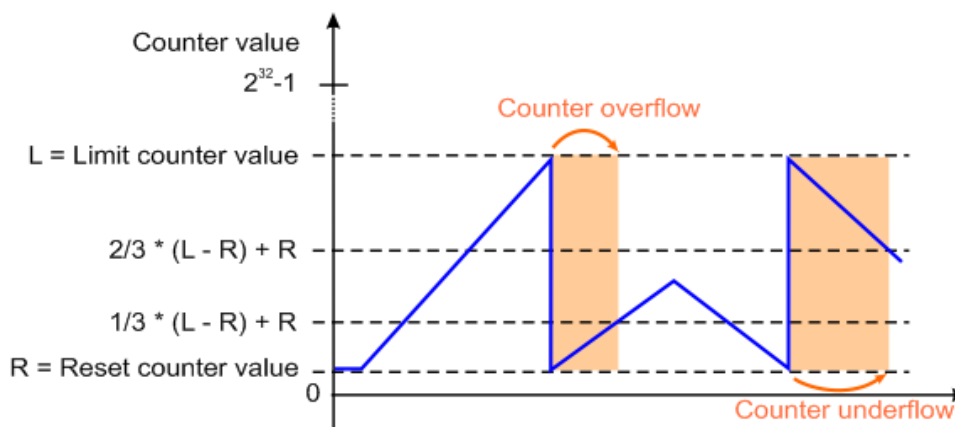


Fig. 152: Counter overflow and underflow with defined value range

#### ● Counter limits

- i**
- The lower counter limit index 0x80n1:1B "Reset counter value" must always be smaller than the upper counter limit 0x80n1:1A "Limit counter value".  
If this is not the case, the last value entered is not accepted.
  - If the lower counter limit in index 0x80n1:1B "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 0x60n2:15 "Counter value out of range".

#### Example:

In 0x70n0:11 "Set counter value" a value is parameterized that is outside the counter limits and the "Set counter" bit in index 0x70n0:03 is activated.

→ The value specified in index 0x70n0:11 is applied.

→ The "Counter value out of range" bit in index 0x60n2:15 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 0x60n2:15 "Counter value out of range".

### Counter overflow and underflow

An overflow or underflow of the counter limits is indicated by the process data 0x60n0:04 "Counter underflow" or 0x60n0:05 "Counter overflow".

- The "Counter underflow" bit in index 0x60n0:04 is set when an underflow 0x80n1:1B "Reset counter value" → 0x80n1:1A "Limit counter value" occurs.
  - With the preset parameters this corresponds to "..00 → ..FF"  
It is reset if 2/3 of the counting range are underrun.
  - For counting limits with fixed values, this corresponds to:  
 $2/3 * ("Limit\ counter\ value" - "Reset\ counter\ value") + "Reset\ counter\ value"$
- The "Counter overflow" bit 0x60n0:05 is set when an overflow 0x80n1:1A "Limit counter value" → 0x80n1:1B "Reset counter value" occurs.
  - With default parameters "..FF → ..00"  
It is reset if 1/3 of the counter range is exceeded.
  - For counting limits with fixed values, this corresponds to:  
 $1/3 * ("Limit\ counter\ value" - "Reset\ counter\ value") + "Reset\ counter\ value"$

#### Example 1:

0x80n1:1A "Limit counter value" =  $2^{12}-1 = 4095$

0x80n1:1B "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:

0x80n1:1A "Limit counter value" =  $2^{12}-1 = 4095$

0x80n1:1B "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.

### 6.3.1.5 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°
  - Reverse (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 0x80n0:0E "Reversion of rotation".

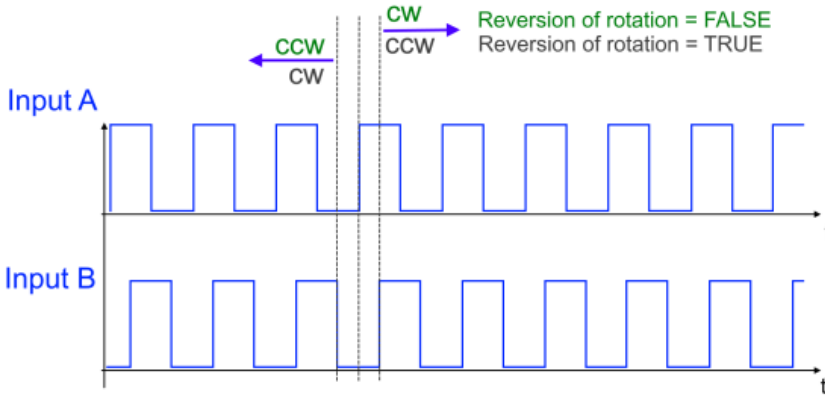


Fig. 153: Reversion of counting direction (Index 0x80n0:0E "Reversion of rotation") for an encoder

- **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
  - Reverse (ccw): HIGH level at track B

Setting the bit in index 0x80n0:0E "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 0x60n0:0A "Status of input B".

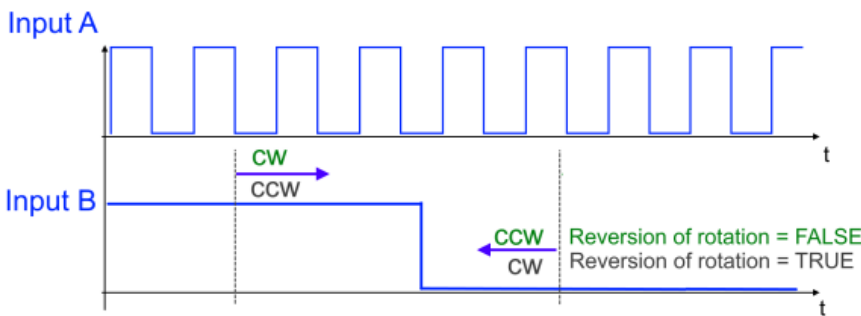


Fig. 154: Reversion of counting direction (Index 0x80n0:0E "Reversion of rotation") for a counter/pulse generator

0x80n1:1D "Counter mode"	Level at input track B	0x80n0:0E "Reversion of rotation"	Resulting counting direction
3: TTL counter (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

## 6.3.2 Reset counter value via gate/latch combination input

A recurring reset of the counter value (index [0x60n0:11](#) [[▶ 172](#)] "Counter value") to "0" can be done in the following ways:

- Gate/Latch combination input: 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.

### Reset counter value via gate/latch combination input (Enable extern reset)

The counter value can be set to zero via the external gate/latch combination input.

- Presetting of the reset value via index [0x80n1:1B](#) [[▶ 171](#)] "Reset counter value", (Default: 0)
- To activate this function set the bit in index [0x80n0:02](#) [[▶ 170](#)] "Enable extern reset".
- Index [0x80n0:10](#) [[▶ 170](#)] "Extern reset polarity" can be used to specify at which edge the gate/latch combination input is active.
  - 0: "Fall" - the counter is set to zero with a falling edge
  - 1: "Rise" - the counter is set to zero with a rising edge

There is no status message via the process data.

---

#### **i** Activation of gate and latch functions via the same edge at the gate/latch input

When the gate function and a latch function is activated via the same edge at the gate/latch input, the latch function is always executed first, after which the counter value is disabled via the gate function.

##### **Example:**

Index [0x80n0:04](#) [[▶ 170](#)] "Gate polarity" (1 = "Enable pos. gate") and

index [0x80n0:02](#) "Enable extern reset" = 1 with [0x80n0:10](#) "Extern reset polarity" = 1 (reset on positive edge at the gate/latch input)

With a positive edge at the gate/latch input the current counter value is first set to zero (reset). 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 [0x80n0:02](#) "Enable extern reset" with [0x80n0:10](#) "Extern reset polarity" = 0 (reset on negative edge at the gate/latch input).

---

### 6.3.3 Set counter value via gate/latch combination input

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: [▶ 140] the counter value can be set from the PLC application (Set counter value)
- Gate/Latch input [▶ 154]: a positive or negative edge at the Latch extern input (Set counter on latch extern on positive/negative edge)

#### 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 gate/latch combination input (Set counter on latch extern on positive/negative edge)

The terminal offers the option of using the Gate/Latch input as a latch input (external latch). To do this, the gate should be deactivated.

---

### ● Deactivation of the gate



Set the "Gate polarity" bit (index [0x80n0:04](#) [[▶ 170](#)]) to "0 - Disable gate" in order to be able to use the Latch extern input without locking the counter value after the latch event.

---

The counter value can be set to a preset value during runtime via the process data by the positive or negative edge at the gate/latch combination input.

- Counter value specification via object [0x70n0:11](#) "Set counter value"
  - Activation of the counter value setting via
    - the positive edge at the external gate/latch combination input: index [0x70n0:0A](#) "Set counter on latch extern on positive edge"
    - the negative edge at the external gate/latch combination input: index [0x70n0:0B](#) "Set counter on latch extern on negative edge"
  - If the bit is set (TRUE) in index [0x70n0:0A](#) or [0x70n0:0B](#), the counter value is set to the specified value at the next rising or falling edge at the gate/latch combination 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.
- 

### ● 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](#) [[▶ 172](#)]).
  - 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.
-

### 6.3.4 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 an external latch input with a separate process datum. The latch function can be triggered as follows:

- [Gate/Latch input \[▶ 143\]](#): positive/negative edge at the Gate/Latch input (Enable latch extern on positive/negative edge)

Index [0x80n0:22 \[▶ 170\]](#) "Enable continuous latch extern" can be used to parameterize whether the function is executed at every parameterized edge at the external latch input or only once after every activation.

---

#### ● **Multiple activation of the latch function**

**i** If several commands are activated simultaneously to save the counter value in the "Latch value" process data (index [0x60n0:12 \[▶ 172\]](#)), 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 Gate/Latch input (Enable latch extern on positive/negative edge)

The terminal offers the option of using the Gate/Latch input as a latch input (external latch). To do this, the gate should be deactivated.

### ● Deactivation of the gate

**i** Set the "Gate polarity" bit (index [0x80n0:04 \[▶ 170\]](#)) to "0 - Disable gate" in order to be able to use the Latch extern input without locking the counter value after the latch event.

- The counter value at the Latch extern input can be saved via:
  - [Index 0x70n0:02 \[▶ 173\]](#) "Enable latch extern on positive edge" = TRUE  
The current counter value is stored in index [0x60n0:12 \[▶ 172\]](#) "Latch value" at the first external pulse with positive edge at the Gate/Latch input.
  - [Index 0x70n0:04 \[▶ 173\]](#) "Enable latch extern on negative edge" = TRUE  
The current counter value is stored in index [0x60n0:12](#) "Latch value" at the first external pulse with negative edge at the Gate/Latch input.
  - Simultaneous activation of [0x70n0:02](#) and [0x70n0:04](#)  
The current counter value is stored in index [0x60n0:12](#) "Latch value" at the first pulse at the Gate/Latch input, independent of the edge polarity.
- Specification whether it is necessary to reactivate the command to save the counter value via:
  - "Enable continuous latch extern" [Index 0x80n0:22 \[▶ 170\]](#) = FALSE  
The following pulses at the Gate/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.  
A new counter value can only be written to the Gate/Latch input in index [0x60n0:12](#) "Latch value" if index [0x60n0:02 \[▶ 172\]](#) "Latch extern valid" is FALSE.
  - "Enable continuous latch extern" [Index 0x80n0:22 \[▶ 170\]](#) = TRUE  
The counter value is written to index [0x60n0:12](#) "Latch value" at every parameterized edge at the Gate/Latch input.  
There is no need to reactivate index [0x70n0:02](#) or [0x70n0:04](#).
- The status of the Gate/Latch input can be recorded via index [0x60n0:0C \[▶ 172\]](#) "Status of input gate".
- Saving of the counter value in index [0x60n0:12](#) "Latch value" is confirmed via bit [0x60n0:02](#) "Latch extern valid".

### ● Activation of gate and latch functions via the same edge at the gate/latch input

**i** When the gate function and a latch function is activated via the same edge at the gate/latch input, the latch function is always executed first, after which the counter value is disabled via the gate function.

#### Example:

[Index 0x80n0:04 \[▶ 170\]](#) "Gate polarity" (1 = "Enable pos. gate") and

[index 0x80n0:02](#) "Enable extern reset" = 1 with [0x80n0:10](#) "Extern reset polarity" = 1 (reset on positive edge at the gate/latch input)

With a positive edge at the gate/latch input the current counter value is first set to zero (reset). 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 0x80n0:02](#) "Enable extern reset" with [0x80n0:10](#) "Extern reset polarity" = 0 (reset on negative edge at the gate/latch input).

## 6.3.5 Lock counter value

The gate function enables locking of the counter ([0x60n0:11](#) [[▶ 172](#)]).

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/Latch combination input](#) [[▶ 154](#)]: a positive / negative edge at the Gate/Latch input (Enable pos./neg. gate),
- [PLC variable](#): [[▶ 144](#)] the counter can be locked from the PLC application (Set software gate).

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

- The level at the gate/latch combination input at which the counter value is locked during runtime can be specified via index [0x80n0:04](#) [[▶ 170](#)] "Gate polarity".
  - 0: Disable gate  
The gate/latch combination input is deactivated to disable the counter. It can still be used as external latch input for storage.
  - 1: Enable pos. gate  
The counter value is locked with HIGH level at the gate/latch combination input.
  - 2: Enable neg. gate  
The counter value is locked with LOW level at the gate/latch combination input
- The current level at the gate/latch combination input is displayed via process record [0x60n0:0C](#) [[▶ 172](#)] "Status of input gate".

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

The counter value can be locked from the PLC application.

- Index [0x70n0:09](#) [[▶ 173](#)] "Set software gate" = TRUE  
The counter is locked.
- For confirmation the "Software gate valid" bit ([0x60n2:11](#) [[▶ 172](#)]) is set to TRUE.
- Index [0x70n0:09](#) "Set software gate" = FALSE  
The counter is unlocked.

### ● **Activation of gate and latch functions via the same edge at the gate/latch input**

**I** When the gate function and a latch function is activated via the same edge at the gate/latch input, the latch function is always executed first, after which the counter value is disabled via the gate function.

#### **Example:**

Index [0x80n0:04](#) [[▶ 170](#)] "Gate polarity" (1 = "Enable pos. gate") and

index [0x80n0:02](#) "Enable extern reset" = 1 with [0x80n0:10](#) "Extern reset polarity" = 1 (reset on positive edge at the gate/latch input)

With a positive edge at the gate/latch input the current counter value is first set to zero (reset). 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 [0x80n0:02](#) "Enable extern reset" with [0x80n0:10](#) "Extern reset polarity" = 0 (reset on negative edge at the gate/latch input).



## 6.4 Extended functionalities

### 6.4.1 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	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 [0x60n0:13](#) [[▶ 172](#)] "Frequency value".

Frequency = number of periods / time interval	
Number of periods	The number of complete periods on track A within the timeframe ( <a href="#">0x80n0:11</a> [ <a href="#">▶ 170</a> ]) is measured. Counting starts with the first rising edge within the timeframe (index <a href="#">0x80n0:11</a> "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.

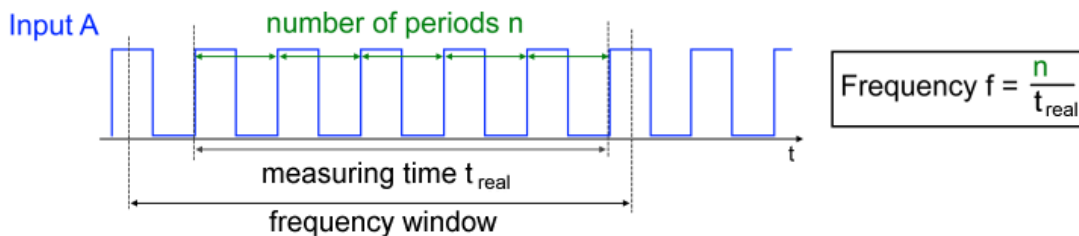


Fig. 155: Frequency measurement - "Frequency value"

- The frequency window is selected in index [0x80n0:11](#) "Frequency window". The default value is 10 ms. Follow the instructions below to select the correct timeframe.
- "Frequency scaling" (index [0x80n0:13](#) [[▶ 170](#)]) 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<sub>dec</sub>). If other resolutions are required, the conversion can be done via [0x80n0:1D](#) [[▶ 170](#)] "Frequency numerator".
- The calculated frequency value is output in the process data via "Frequency value" (index [0x60n0:13](#)) [[▶ 172](#)].
- 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 or external reset has no effect on the frequency calculation.

### Correct selection of the frequency window:

In delivery state the frequency window of the frequency measurement is set to 10 ms.

The accuracy and size of the determined frequency depends on the size of the timeframe (0x80n0:11 [▶ 170] "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 (0x80n0:17 [▶ 170] "Frequency wait time") has elapsed. Until then the last valid value is output in index 0x60n0:13 [▶ 172] "Frequency value".

If no full period is detected within the waiting time, the measurement is discarded and the output in index 0x60n0:13 "Frequency value" is set to "0".

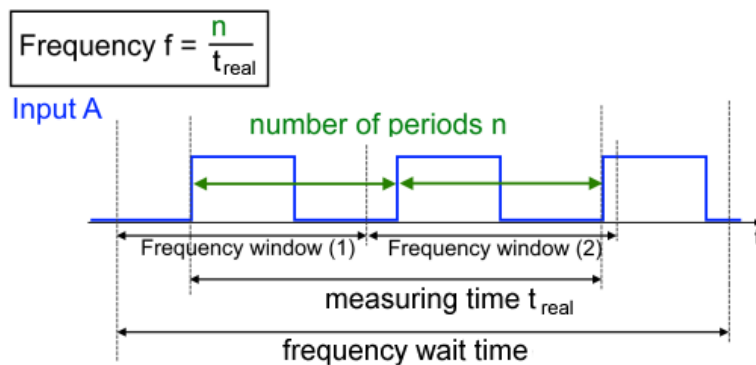


Fig. 156: Frequency measurement outside the frequency window with waiting time

### ● Difference frequency and period value measurement

**i** 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.

### 6.4.2 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	1 $\mu$ s	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 `0x60n0:14 [▶ 172]` "Period value".
- "Period scaling" (index `0x80n0:14`) 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 (`0x80n0:14` "Period scaling" set to  $100_{dec}$ ) or 500 ns (`0x80n0:14` "Period scaling" set to  $500_{dec}$ ).

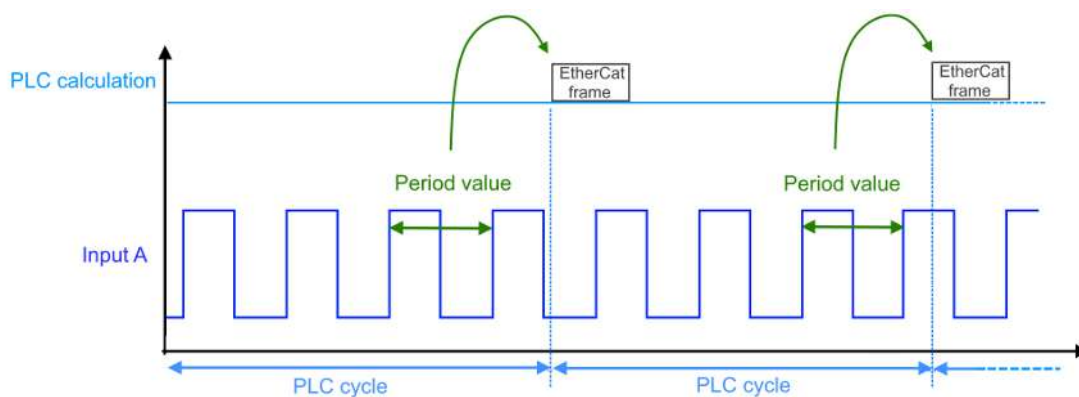


Fig. 157: Period value measurement

- 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 `0x60n0:14 [▶ 172]` "Period value".
- If no full period is detected within 21 seconds, the measurement is discarded and the output in index `0x60n0:14 [▶ 172]` "Period value" is set to the maximum value, 2147483648.

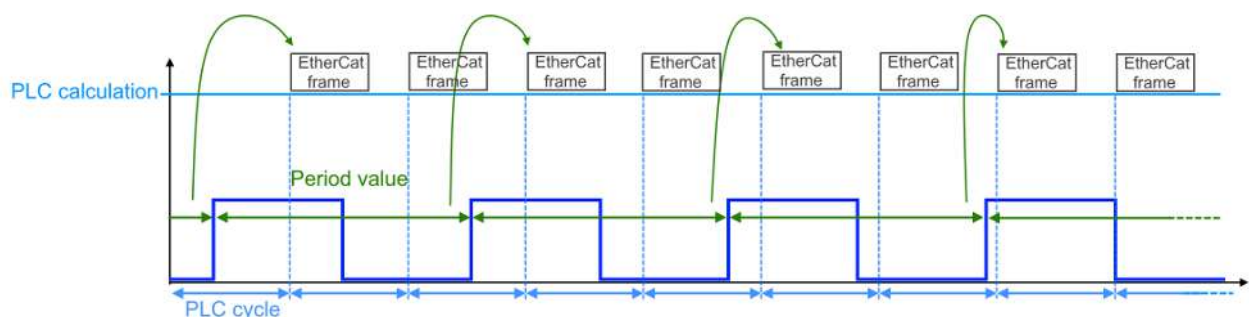


Fig. 158: Period value measurement for long periods

#### **i** 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.

### 6.4.3 Velocity, speed calculation

In addition to the counter value, the terminal offers the option of measuring the velocity and speed. 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	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 [0x80n0:1D \[► 170\]](#) "Frequency numerator" and [0x80n0:1E \[► 170\]](#) "Frequency denominator".

Index (hex)	Name	Meaning	Default
80n0:1D	Frequency numerator	Frequency numerator Factor for normalizing the frequency, e.g., the time unit	0x0000001 (1 <sub>dec</sub> )
80n0:1E	Frequency denominator	Frequency denominator, Factor for normalizing the frequency, e.g., speed calculation	0x0000001 (1 <sub>dec</sub> )

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 [0x60n0:13 \[► 172\]](#)).

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:

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

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

Index (hex)	Name	Example: velocity in m/s	Comment
<a href="#">80n0:1D [► 170]</a>	Frequency numerator	0x00000C8 (200 <sub>dec</sub> )	Travel path, conversion to m via "Frequency denominator"
<a href="#">80n0:1E [► 170]</a>	Frequency denominator	0x003E8000 (4096000 <sub>dec</sub> )	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 (0x80n0:06 [▶ 170] "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 (0x80n0:13 [▶ 170] "Frequency scaling"):

Conversion to minutes depending on "Frequency scaling"			
Index (hex)	Entry	Index (hex)	Entry
0x80n0:13 [▶ 170] Frequency scaling	100 <sub>dec</sub> : 0.01 Hz	0x80n0:1D [▶ 170] Frequency numerator	0x00001770 (6000 <sub>dec</sub> )
	1 <sub>dec</sub> : 1 Hz		0x0000003C (60 <sub>dec</sub> )

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

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

Index (hex)	Example: Speed $f_{rot}$ in rpm	Comment
80n0:1D [▶ 170] Frequency numerator	0x0001770 (6000 <sub>dec</sub> )	Output of the value in 0.01 rpm
80n0:1E [▶ 170] Frequency denominator	0x0001000 (4096 <sub>dec</sub> )	Conversion factor in periods / m

## 6.4.4 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	TTL mode: 1 MHz Open Collector: 100 kHz
Gate/Latch input	1 MHz

### Filter function sequence

- The filter is disabled on delivery.
- The filter can be enabled via index 0x80n0:08 "Disable filter" = FALSE.
- The filter is parameterized via index 0x80n1:19 "Filter settings". The following filter frequencies are available:

Index 0x80n1:19 "Filter settings"		Default parameters for 0x80n1:1D "Counter mode"	
Setting	Meaning	Setting	Meaning
10 <sub>dec</sub> : 10 kHz	10 kHz filter		
25 <sub>dec</sub> : 25 kHz	25 kHz filter		
50 <sub>dec</sub> : 50 kHz	50 kHz filter		
100 <sub>dec</sub> : 100 kHz	100 kHz filter	0x80n1:1D "Counter mode"	
		4 <sub>dec</sub>	Encoder open collector
		5 <sub>dec</sub>	Counter open collector
250 <sub>dec</sub> : 250 kHz	250 kHz filter		
500 <sub>dec</sub> : 500 kHz	500 kHz filter		
1000 <sub>dec</sub> : 1 MHz	1 MHz filter	0x80n1:1D "Counter mode"	
		2 <sub>dec</sub>	Encoder TTL (single-ended)
		3 <sub>dec</sub>	Counter TTL (single-ended)
2500 <sub>dec</sub> : 2.5 MHz	2.5 MHz filter		
5000 <sub>dec</sub> : 5 MHz	5 MHz filter		

- If the filter is parameterized < 1MHz, it is also active for the digital 24 V inputs.
- If a frequency overrun is detected with the activated filter (0x80n0:08 and 0x80n1:19), a counter in index 0xA0n0:16 is incremented.

Index (hex)	Name	Description
0xA0n0:15	Filter violation counter input gate	Filter frequency overshoot counter for the Gate input
0xA0n0:16	Filter violation counter	Filter frequency overshoot counter for the encoder input signals with activated 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"

0xFB001:01 „Request“	Description
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
0x9163	Sets the internal error counter for 0xA010:15 "Filter violation counter input gate" for channel 2 to zero
0x9164	Sets the internal error counter for 0xA010:16 "Filter violation counter" for channel 2 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  $\mu\text{s}$  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.

## 6.4.5 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 0x80n0:21 "Enable encoder plausibility check"
- If a plausibility error is detected, it is displayed as follows:

Error diagnosis	Description
DiagMessage, 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"

0xFB001:01 "Request"	Description
0x9151	Sets the internal error counter 0xA000:13 "Encoder plausibility error counter" for channel 1 to zero
0x9161	Sets the internal error counter 0xA010:13 "Encoder plausibility error counter" for channel 2 to zero



## 6.5 Inputs

### Functions of the inputs

The following table provides a brief overview of the input functions. Please refer to the detailed function descriptions in the respective chapters.

Counter value	PLC variable	Gate/latch input (Latch extern)
<b>Set</b> [▶ 140]	Enable setting: 0x70n0:03 "Set counter" = TRUE  to the value: 0x70n0:11 "Set counter value"	Enable setting: 0x70n0:0A/0B "Set counter on latch extern on positive/negative edge" = TRUE  to the value: 0x70n0:11 "Set counter value"
<b>Reset</b> [▶ 139]	-	Enable resetting: 0x80n0:02 "Enable extern reset"  to the value: 0x80n1:1B "Reset counter value"
<b>Save</b> [▶ 142]	-	Enable saving: 0x70n0:02/04 "Enable latch extern on positive/negative edge" = TRUE  Save current counter value in: 0x60n0:12 "Latch value"
<b>Lock</b> [▶ 144]	Lock: 0x70n0:09 "Set software gate" = TRUE	Enable locking  with HIGH level: 0x80n0:04 "Gate polarity" = 1  with LOW level: 0x80n0:04 "Gate polarity" = 2

## 6.5.1 Gate/Latch combination input

The terminal provides an external Gate/Latch combination input for 24 V<sub>DC</sub> signals with a min. pulse duration of  $t_{ON} > 1 \mu s$ . This can be used as follows:

- Set counter value: set the counter value to a predefined counter value via a positive or negative edge (Set counter on latch extern on positive/negative edge)
- Counter value reset: reset the counter value to the value set in index 0x80n0:1B "Reset counter value" (default setting: zero) (Enable extern reset)
- Save counter value: save the counter value in separate process data via a positive or negative edge (Enable latch extern on positive/negative edge)
- Lock counter value: lock the counter value via a positive or negative edge (Enable pos/neg. gate)

### ● Simultaneous use of Gate and Latch

**I** When using index 0x80n0:04 [▶ 170] "Gate polarity" (1 = "Enable pos. gate") and simultaneously activating index 0x70n0:02 "Enable latch extern on positive edge", the current counter value is initially stored in index 0x60n0:12 "Latch value" 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 0x70n0:04 "Enable latch extern on negative edge" at a negative edge at the Gate/Latch input.

### Set counter value via gate/latch combination input (Set counter on latch extern on positive/negative edge)

The terminal offers the option of using the Gate/Latch input as a latch input (external latch). To do this, the gate should be deactivated.

### ● Deactivation of the gate

**I** Set the "Gate polarity" bit (index 0x80n0:04 [▶ 170]) to "0 - Disable gate" in order to be able to use the Latch extern input without locking the counter value after the latch event.

The counter value can be set to a preset value during runtime via the process data by the positive or negative edge at the gate/latch combination input.

- Counter value specification via object 0x70n0:11 "Set counter value"
- Activation of the counter value setting via
  - the positive edge at the external gate/latch combination input: index 0x70n0:0A "Set counter on latch extern on positive edge"
  - the negative edge at the external gate/latch combination input: index 0x70n0:0B "Set counter on latch extern on negative edge"
- If the bit is set (TRUE) in index 0x70n0:0A or 0x70n0:0B, the counter value is set to the specified value at the next rising or falling edge at the gate/latch combination 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.

### Reset counter value via gate/latch combination input (Enable extern reset)

The counter value can be set to zero via the external gate/latch combination input.

- Presetting of the reset value via index 0x80n1:1B [▶ 171] "Reset counter value", (Default: 0)
- To activate this function set the bit in index 0x80n0:02 [▶ 170] "Enable extern reset".
- Index 0x80n0:10 [▶ 170] "Extern reset polarity" can be used to specify at which edge the gate/latch combination input is active.
  - 0: "Fall" - the counter is set to zero with a falling edge
  - 1: "Rise" - the counter is set to zero with a rising edge

There is no status message via the process data.

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

- Save the counter value at the external gate/latch combination input via:
  - Index [0x70n0:02 \[▶ 173\]](#) "Enable latch extern on positive edge" = TRUE  
At the first external latch pulse with positive edge the current counter value is stored in index [0x60n0:12 \[▶ 172\]](#) "Latch value".
  - Index [0x70n0:04 \[▶ 173\]](#) "Enable latch extern on negative edge" = TRUE  
At the first external latch pulse with negative edge the current counter value is stored in index [0x60n0:12 \[▶ 172\]](#) "Latch value".
  - Simultaneous activation of [0x70n0:02](#) and [0x70n0:04](#)  
The current counter value is stored in index [0x60n0:12](#) "Latch value", at the first external latch pulse, independent of the edge polarity.
- Specification whether it is necessary to reactivate the command to save the counter value via:
  - "Enable continuous latch extern" Index [0x80n0:22 \[▶ 170\]](#) = FALSE  
The following pulses at the gate/latch combination 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".  
A new counter value can only be written to the gate/latch combination input in index [0x60n0:12](#) "Latch value" if index [0x60n0:02 \[▶ 172\]](#) "Latch extern valid" is FALSE.
  - "Enable continuous latch extern" Index [0x80n0:22 \[▶ 170\]](#) = TRUE  
The counter value is written to index [0x60n0:12](#) "Latch value" at every parameterized edge at the gate/latch combination input.  
There is no need to reactivate index [0x70n0:02](#) or [0x70n0:04](#).
- Saving of the counter value in index [0x60n0:12](#) "Latch value" is confirmed via the "Latch extern valid" bit (index [0x60n0:02](#)).
- The status of the gate/latch combination input can be monitored via index [0x60n0:0C](#) "Status of input gate".

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

- The level at the gate/latch combination input at which the counter value is locked during runtime can be specified via index [0x80n0:04 \[▶ 170\]](#) "Gate polarity".
  - 0: Disable gate  
The gate/latch combination input is deactivated to disable the counter. It can still be used as external latch input for storage.
  - 1: Enable pos. gate  
The counter value is locked with HIGH level at the gate/latch combination input.
  - 2: Enable neg. gate  
The counter value is locked with LOW level at the gate/latch combination input
- The current level at the gate/latch combination input is displayed via process record [0x60n0:0C \[▶ 172\]](#) "Status of input gate".

## 6.6 Diagnostics

### 6.6.1 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

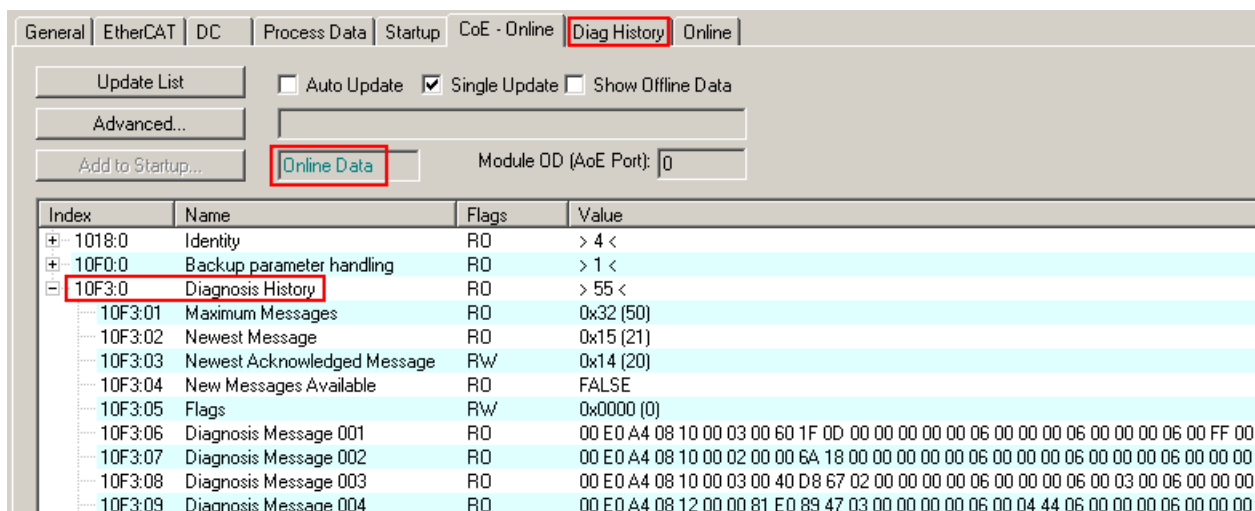
- Diag Code (4-byte)
- Flags (2-byte; info, warning or error)
- Text ID (2-byte; reference to explanatory text from the ESI/XML)
- Timestamp (8-byte, local slave time or 64-bit Distributed Clock time, if available)
- Dynamic parameters added by the firmware

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:



Index	Name	Flags	Value
1018:0	Identity	RO	> 4 <
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 55 <
10F3:01	Maximum Messages	RO	0x32 (50)
10F3:02	Newest Message	RO	0x15 (21)
10F3:03	Newest Acknowledged Message	R/W	0x14 (20)
10F3:04	New Messages Available	RO	FALSE
10F3:05	Flags	R/W	0x0000 (0)
10F3:06	Diagnosis Message 001	RO	00 E0 A4 08 10 00 03 00 60 1F 0D 00 00 00 00 00 06 00 00 00 06 00 00 00 06 00 FF 00
10F3:07	Diagnosis Message 002	RO	00 E0 A4 08 10 00 02 00 00 6A 18 00 00 00 00 00 06 00 00 00 06 00 00 00 06 00 00 00
10F3:08	Diagnosis Message 003	RO	00 E0 A4 08 10 00 03 00 40 D8 67 02 00 00 00 00 06 00 00 00 06 00 03 00 06 00 00 00
10F3:09	Diagnosis Message 004	RO	00 E0 A4 08 12 00 00 81 E0 89 47 03 00 00 00 00 06 00 04 44 06 00 00 00 06 00 00 00

Fig. 159: *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.

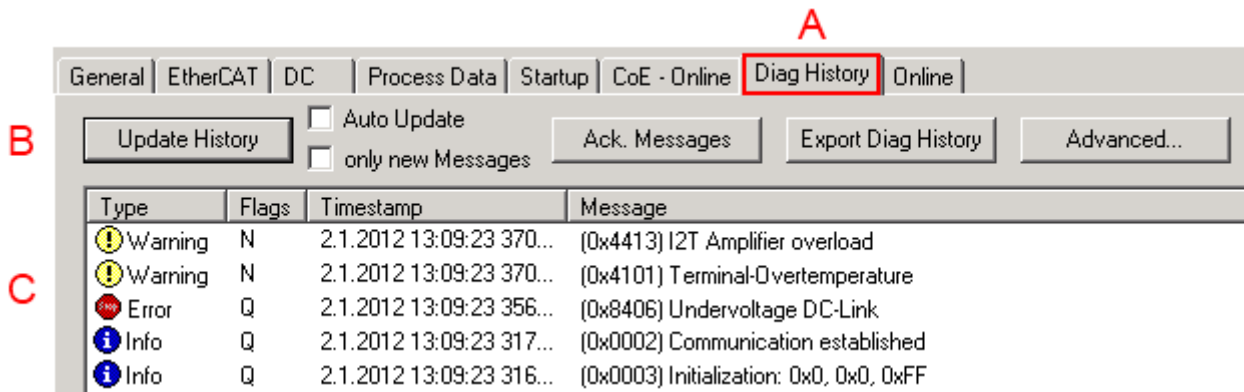


Fig. 160: Implementation of the DiagMessage system in the TwinCAT System Manager

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:

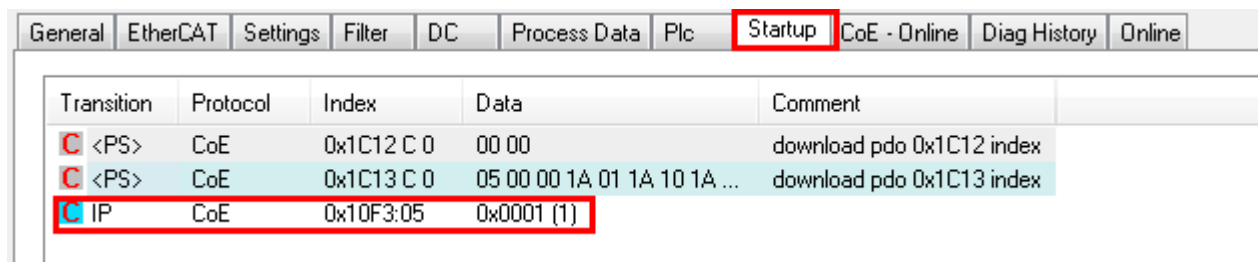


Fig. 161: StartUp List

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**:

<b>x</b>	<b>y</b>	<b>zz</b>
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)	<ul style="list-style-type: none"> <li>The amplifier is being operated outside the specification.</li> <li>The I2T-model of the amplifier is incorrectly parameterized.</li> </ul>
0x4414	Warning	Drive	I2T-Model Motor overload (Warning)	<ul style="list-style-type: none"> <li>The motor is being operated outside the parameterized rated values.</li> </ul>

Text ID	Type	Place	Text Message	Additional comment
				<ul style="list-style-type: none"> <li>The I2T-model of the motor is incorrectly parameterized.</li> </ul>
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. <ul style="list-style-type: none"> <li>The EtherCAT connection was interrupted during operation.</li> <li>The Master was switched to Config mode during operation.</li> </ul>
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)	<ul style="list-style-type: none"> <li>There is no voltage applied to the power contacts.</li> <li>A firmware update has failed.</li> </ul>
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)	<ul style="list-style-type: none"> <li>The amplifier is being operated outside the specification.</li> <li>The I2T-model of the amplifier is incorrectly parameterized.</li> </ul>
0x8409	Error	Drive	I2T-Model motor overload (Error)	<ul style="list-style-type: none"> <li>The motor is being operated outside the parameterized rated values.</li> <li>The I2T-model of the motor is incorrectly parameterized.</li> </ul>
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

## 6.6.2 EL5122 diagnostics

The EL5122 offers the following diagnostic options:

- [Overflow and underflow of the counter limits \[► 166\]](#)
- [Monitoring of the encoder operating voltage \[► 166\]](#)
- [Plausibility check \[► 167\]](#)
- [Filter frequency overshoot \[► 168\]](#)

### 6.6.2.1 Counter overflow and underflow

An overflow or underflow of the counter limits is indicated by the process data 0x60n0:04 "Counter underflow" or 0x60n0:05 "Counter overflow".

- The "Counter underflow" bit in index 0x60n0:04 is set when an underflow 0x80n1:1B "Reset counter value" → 0x80n1:1A "Limit counter value" occurs. With default parameters this corresponds to ".00 → ..FF"  
It is reset if 2/3 of the counting range are underrun.
- The "Counter overflow" bit 0x60n0:05 is set when an overflow 0x80n1:1A "Limit counter value" → 0x80n1:1B "Reset counter value" occurs. With default parameters ".00 → ..FF"  
It is reset if 1/3 of the counter range is exceeded.

### 6.6.2.2 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<sub>DC</sub> power contacts

The encoder operating voltage is generated from the 24 V<sub>DC</sub> supply of the power contacts. If the power contacts have no supply, no encoder operating voltage can be provided.

### 6.6.2.3 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 0x80n0:21 "Enable encoder plausibility check"
- If a plausibility error is detected, it is displayed as follows:

Error diagnosis	Description
DiagMessage, 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"

0xFB001:01 "Request"	Description
0x9151	Sets the internal error counter 0xA000:13 "Encoder plausibility error counter" for channel 1 to zero
0x9161	Sets the internal error counter 0xA010:13 "Encoder plausibility error counter" for channel 2 to zero

### 6.6.2.4 Filter frequency overshoot

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	TTL mode: 1 MHz Open Collector: 100 kHz
Gate/Latch input	1 MHz

- If a frequency overrun is detected with the activated filter (0x80n0:08 and 0x80n1:19), a counter in index 0xA0n0:16 is incremented.

Index (hex)	Name	Description
0xA0n0:15	Filter violation counter input gate	Filter frequency overshoot counter for the Gate input
0xA0n0:16	Filter violation counter	Filter frequency overshoot counter for the encoder input signals with activated filter

#### NOTICE



#### Note Parameterization of adjustable interference pulse filters

Please note the description and notes for parameterization in chapter "[Adjustable interference pulse filters \[►\\_150\]](#)".



## 6.7 Object description and parameterization

### EtherCAT XML Device Description

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

#### NOTICE



#### Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE - Online tab (with a double click on the respective object) or via the Process Data tab (assignment of PDOs). A detailed description can be found in the EtherCAT System-Documentation in chapter “EtherCAT subscriber configuration”

Please note the general CoE notes in the EtherCAT System Documentation in chapter “CoE-interface” when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary,
- existence of current XML description
- use "CoE reload" for resetting changes

### 6.7.1 Restore object

#### Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	<a href="#">Restore default parameters [► 203]</a>	Restore default parameters	UINT8	RO	0x01 (1 <sub>dec</sub> )
1011:01	SubIndex 001	If this object is set to “ <b>0x64616F6C</b> ” in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0 <sub>dec</sub> )

## 6.7.2 Configuration data

### Index 80n0 ENC Settings 0 Ch.n (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	ENC Settings 0 Ch.n	Maximum subindex	UINT8	RO	0x22 (34 <sub>dec</sub> )
80n0:02	<a href="#">Enable extern reset</a> <a href="#">▶ 139</a>	A counter reset is triggered via the external latch input (24 V).	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
80n0:04	Gate polarity	0: Disable gate 1: Enable pos. gate (gate locks with HIGH signal level) 2: Enable neg. gate (gate locks with LOW signal level)	BIT2	RW	0x01 (1 <sub>dec</sub> )
80n0:06	<a href="#">Evaluation mode</a> <a href="#">▶ 135</a>	0: 4-fold (four-fold evaluation) 1: 1-fold (single evaluation) 2: 2-fold (two-fold evaluation)	BIT2	RW	0x00 (0 <sub>dec</sub> )
80n0:08	<a href="#">Disable filter</a> <a href="#">▶ 150</a>	0: Activates input filter (inputs A, B only) 1: Deactivates the input filter If a filter is activated a signal edge must be present for at least 2.4 µs in order to be counted as an increment.	BOOLEAN	RW	0x01 (1 <sub>dec</sub> )
80n0:0E	<a href="#">Reversion of rotation</a> <a href="#">▶ 138</a>	Activates reversion of rotation	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
80n0:10	<a href="#">Extern reset polarity</a> <a href="#">▶ 154</a>	0: Fall (the counter is set to zero with a falling edge) 1: Rise (the counter is set to zero with a rising edge)	BIT1	RW	0x01 (1 <sub>dec</sub> )
80n0:11	<a href="#">Frequency window</a> <a href="#">▶ 145</a>	This is the minimum time over which the frequency is determined. Default 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 <a href="#">0x60n0:13</a> <a href="#">▶ 172</a> and as process data. The frequency calculation is carried out locally without distributed clocks function.	UINT16	RW	0x2710 (10000 <sub>dec</sub> )
80n0:13	<a href="#">Frequency scaling</a> <a href="#">▶ 145</a>	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 <sub>dec</sub> )
80n0:14	Period scaling	Resolution of the period value in the process data:  10: "10 ns" Period value is a multiple of 10 ns 100: "100 ns" Period value is a multiple of 100 ns 500: "500 ns" Period value is a multiple of 500 ns	UINT32	RW	0x0000000A (10 <sub>dec</sub> )
80n0:17	<a href="#">Frequency Wait Time</a> <a href="#">▶ 145</a>	Waiting time [ms] for frequency measurement  If no full period is detected within the <a href="#">frequency window</a> <a href="#">▶ 146</a> , 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. $t \geq 2 * (1 / f_{min})$	UINT16	RW	0x53E2 (21474 <sub>dec</sub> )
80n0:1D	<a href="#">Frequency numerator</a> <a href="#">▶ 148</a>	Frequency counter value, frequency scaling	UINT32	RW	0x00000001 (1 <sub>dec</sub> )
80n0:1E	<a href="#">Frequency denominator</a> <a href="#">▶ 148</a>	frequency counter value, used for scaling the frequency and the velocity calculation (increments / unit).	UINT32	RW	0x00000001 (1 <sub>dec</sub> )
80n0:21	<a href="#">Enable encoder plausibility check</a> <a href="#">▶ 167</a>	Activation of plausibility check	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
80n0:22	<a href="#">Enable continuous latch extern</a> <a href="#">▶ 155</a>	FALSE: The following pulses at the Latch input have no influence on the latch value in index <a href="#">0x60n0:12</a> "Latch value" when the bit in index <a href="#">0x70n0:02</a> or <a href="#">0x70n0:04</a> is set.  TRUE: The counter value is written to index <a href="#">0x60n0:12</a> "Latch value" at every parameterized edge at the Latch input. There is no need to reactivate index <a href="#">0x70n0:02</a> or <a href="#">0x70n0:04</a> .	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )

**Index 80n1 ENC Settings 1 Ch.n (for n=0 [channel 1], n=1 [channel 2])**

Index (hex)	Name	Meaning	Data type	Flags	Default
80n1:0	ENC Settings 1 Ch.n	Maximum subindex	UINT8	RO	0x1D (29 <sub>dec</sub> )
80n1:17	Supply voltage [▶ 135]	Setting the sensor supply 50 <sub>dec</sub> : 5.0 V (default) 120 <sub>dec</sub> : 12.0 V 240 <sub>dec</sub> : 24.0 V	UINT32	RW	0x00000032 (50 <sub>dec</sub> )
80n1:19	Filter settings [▶ 150]	Filter settings: 10 <sub>dec</sub> : 10 kHz 25 <sub>dec</sub> : 25 kHz 50 <sub>dec</sub> : 50 kHz 100 <sub>dec</sub> : 100 kHz 250 <sub>dec</sub> : 250 kHz 500 <sub>dec</sub> : 500 kHz 1000 <sub>dec</sub> : 1 MHz (default) 2500 <sub>dec</sub> : 2.5 MHz 5000 <sub>dec</sub> : 5 MHz	UINT32	RW	0x000003E8 (1000 <sub>dec</sub> )
80n1:1A	Limit counter value [▶ 135]	Specifies the value for the upper counter limit.	UINT32	RW	0xFFFFFFFF (-1 <sub>dec</sub> )
80n1:1B	Reset counter value [▶ 135]	Specifies the value for the lower counter limit.	UINT32	RW	0x00000000 (0 <sub>dec</sub> )
80n1:1D	Counter mode [▶ 134]	2: Encoder TTL (single-ended) 3: Counter TTL (single-ended) 4: Encoder open collector 5: Counter open collector	UINT32	RW	0x00000002 (2 <sub>dec</sub> )

**6.7.3 Command object**

**Index FB00 RMB Command**

Index (hex)	Name	Meaning	Data type	Flags	Default
FB00:0	RMB command	Max. Subindex	UINT8	RO	0x03 (3 <sub>dec</sub> )
FB00:01	Request	Commands can be sent to the terminal via the request object. Command:  Resetting the error counter for plausibility errors [▶ 167]: 0x9151 Set index 0xA000:13 "Encoder plausibility error counter" for channel 1 to zero 0x9161 Set index 0xA010:13 "Encoder plausibility error counter" for channel 2 to zero  Resetting the internal error counter [▶ 150]: 0x9153 Set index 0xA000:15 "Filter violation counter input gate" for channel 1 to zero 0x9154 Set index 0xA000:16 "Filter violation counter" for channel 1 to zero 0x9163 Set index 0xA010:15 "Filter violation counter input gate" for channel 2 to zero 0x9164 Set index 0xA010:16 "Filter violation counter" for channel 2 to zero	OCTET-STRING[2]	RW	{0}
FB00:02	Status	Status of the command currently being executed 0: Command executed without error 255: Command is being executed	UINT8	RO	0x00 (0 <sub>dec</sub> )
FB00:03	Response	Optional return value of the command	OCTET-STRING[4]	RO	{0}

## 6.7.4 Input data

### Index 60n0 ENC Inputs Ch.n (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	ENC Inputs Ch.n	Maximum subindex	UINT8	RO	0x14 (20 <sub>dec</sub> )
60n0:02	<a href="#">Latch extern valid</a> [▶ 143]	The counter value was stored via the Latch extern input.	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:03	<a href="#">Set counter done</a> [▶ 140]	The counter was set.	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:04	<a href="#">Counter underflow</a> [▶ 137]	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 <sub>dec</sub> )
60n0:05	<a href="#">Counter overflow</a> [▶ 137]	The upper counter limit was exceeded. The bit is reset when the counter value has fallen below 1/3 of the counting range.	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:09	Status of input A	Status of input A	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:0A	Status of input B	Status of input B	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:0C	Status of input gate	The state of the gate input	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:11	<a href="#">Counter value</a> [▶ 134]	Counter value	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
60n0:12	<a href="#">Latch value</a> [▶ 142]	Latch value	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
60n0:13	<a href="#">Frequency value</a> [▶ 145]	The frequency (setting of the scaling in index <a href="#">0x80n0:13</a> [▶ 170])	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
60n0:14	<a href="#">Period value</a> [▶ 147]	The period value (setting of the scaling in index <a href="#">0x80n0:14</a> [▶ 170])	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

### Index 60n2 ENC Input status Ch. n (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
60n2:0	ENC Inputs status Ch. n	Maximum subindex	UINT8	RO	0x15 (21 <sub>dec</sub> )
60n2:0D	Diag	Indicates that a new message is available in the "Diag History"	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n2:0E	TxPDO State	Validity of the data of the associated TxPDO (0 = valid, 1 = invalid)	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n2:0F	Input cycle counter	2-bit counter for synchronization (incremented only if a new value is present)	BIT2	RO	0x00 (0 <sub>dec</sub> )
60n2:11	Software gate valid	0:Counter unlocked (index <a href="#">0x70n0:09</a> [▶ 173] "Set software gate" = FALSE) 1:Counter locked (index <a href="#">0x70n0:09</a> "Set software gate" = TRUE)	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n2:15	<a href="#">Counter value out of range</a> [▶ 135]	Indicates that the counter value has exceeded the counter limit.	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

## 6.7.5 Output data

### Index 70n0 ENC Outputs Ch.n (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
70n0:0	ENC Outputs Ch.n	Maximum subindex	UINT8	RO	0x11 (17 <sub>dec</sub> )
70n0:02	Enable latch extern on positive edge [▶ 143]	Enable saving via the Latch extern input with positive edge.	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
70n0:03	Set counter [▶ 140]	Set counter value	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
70n0:04	Enable latch extern on negative edge [▶ 143]	Enable saving via the Latch extern input with negative edge.	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
70n0:09	Set software gate [▶ 144]	Locks the counter via a PLC variable 0: Counter is unlocked 1: Counter is locked	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
70n0:0A	Set counter on latch extern on positive edge [▶ 140]	Enables counter value specification via a positive edge at the Latch extern input. The counter value is specified in index 0x70n0:11 "Set counter value".	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
70n0:0B	Set counter on latch extern on negative edge [▶ 140]	Enables counter value specification via a negative edge at the Latch extern input. The counter value is specified in index 0x70n0:11 "Set counter value".	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
70n0:11	Set counter value [▶ 140]	The counter value to be set via "Set counter" (index 0x70n0:03).	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

## 6.7.6 Information / diagnostic data (channel specific)

### Index A0n0 ENC Diag data Ch. n (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
A0n0:0	ENC Diag data Ch.n	Maximum subindex	UINT8	RO	0x16 (22 <sub>dec</sub> )
A0n0:01	Error A	An "open circuit" error is present at input A	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A0n0:02	Error B	An "open circuit" error is present at input B	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A0n0:04	Field power failure	Encoder supply voltage is not present	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A0n0:13	Encoder plausibility error counter	Number of detected plausibility errors	UINT16	RO	0x0000 (0 <sub>dec</sub> )
A0n0:15	Filter violation counter input gate	Number of filter overruns detected at the Gate/Latch input	UINT16	RO	0x0000 (0 <sub>dec</sub> )
A0n0:16	Filter violation counter	Number of detected filter violations at the encoder inputs	UINT16	RO	0x0000 (0 <sub>dec</sub> )

## 6.7.7 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 <sub>dec</sub> )
10F3:01	Maximum Messages	Maximum number of stored messages A maximum of 16 messages can be stored.	UINT8	RO	0x00 (0 <sub>dec</sub> )
10F3:02	Newest Messages	Subindex of the latest message	UINT8	RO	0x00 (0 <sub>dec</sub> )
10F3:03	Newest Acknowledged Messages	Subindex of the last confirmed message	UINT8	RW	0x00 (0 <sub>dec</sub> )
10F3:04	New Messages Available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
10F3:05	Flags	not used	UINT16	RW	0x0000 (0 <sub>dec</sub> )
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	

**6.7.8 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 <sub>dec</sub> )

**Index 1008 Device name**

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL5122

**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 <sub>dec</sub> )
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2 <sub>dec</sub> )
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x14023052 (335687762 <sub>dec</sub> )
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 <sub>dec</sub> )
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

**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 <sub>dec</sub> )
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

**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 <sub>dec</sub> )
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 02 16 03 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 <sub>dec</sub> )
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 02 16 03 16

**Index 1402 ENC RxPDO-Par Control Counter Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1402:0	ENC RxPDO-Par Control Counter Ch.1	PDO Parameter RxPDO 3	UINT8	RO	0x06 (6 <sub>dec</sub> )
1402:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 3	OCTET-STRING[6]	RO	00 16 01 16 03 16

**Index 1403 ENC RxPDO-Par Control Compact Counter Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1403:0	ENC RxPDO-Par Control Compact Counter Ch.1	PDO Parameter RxPDO 4	UINT8	RO	0x06 (6 <sub>dec</sub> )
1403:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 4	OCTET-STRING[2]	RO	00 16 01 16 02 16

**Index ENC 1404 RxPDO-Par Control Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1404:0	ENC RxPDO-Par Control Ch.2	PDO Parameter RxPDO 5	UINT8	RO	0x06 (6 <sub>dec</sub> )
1404:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 5	OCTET-STRING[6]	RO	05 16 06 16 07 16

**Index 1405 ENC RxPDO-Par Control Compact Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1405:0	ENC RxPDO-Par Control Compact Ch.2	PDO Parameter RxPDO 6	UINT8	RO	0x06 (6 <sub>dec</sub> )
1405:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 6	OCTET-STRING[6]	RO	04 16 06 16 07 16

**Index 1406 ENC RxPDO-Par Control Counter Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1406:0	ENC RxPDO-Par Control Counter Ch.2	PDO Parameter RxPDO 7	UINT8	RO	0x06 (6 <sub>dec</sub> )
1406:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 7	OCTET-STRING[6]	RO	04 16 05 16 07 16

**Index 1407 ENC RxPDO-Par Control Compact Counter Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1407:0	ENC RxPDO-Par Control Compact Counter Ch.2	PDO Parameter RxPDO 8	UINT8	RO	0x06 (6 <sub>dec</sub> )
1407:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 8	OCTET-STRING[2]	RO	04 16 05 16 06 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 <sub>dec</sub> )
1600:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 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 (1 bits align)	UINT32	RO	0x0000:00, 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 (1 bits align)	UINT32	RO	0x0000:00, 1
1600:0B	SubIndex 011	11. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 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 <sub>dec</sub> )
1601:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 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 (1 bits align)	UINT32	RO	0x0000:00, 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 (1 bits align)	UINT32	RO	0x0000:00, 1
1601:0B	SubIndex 011	11. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 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 Counter Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	ENC RxPDO-Map Control Counter Ch.1	PDO Mapping RxPDO 3	UINT8	RO	0x06 (6 <sub>dec</sub> )
1602:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1602:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x03 (Set Counter))	UINT32	RO	0x7000:03, 1
1602:03	SubIndex 003	3. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1602:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x09 (Set software gate))	UINT32	RO	0x7000:09, 1
1602:05	SubIndex 005	5. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1602:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x7000:11, 32

**Index 1603 ENC RxPDO-Map Control Compact Counter Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1603:0	ENC RxPDO-Map Control Compact Counter Ch.1	PDO Mapping RxPDO 4	UINT8	RO	0x06 (6 <sub>dec</sub> )
1603:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1603:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x03 (Set Counter))	UINT32	RO	0x7000:03, 1
1603:03	SubIndex 003	3. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1603:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x09 (Set software gate))	UINT32	RO	0x7000:09, 1
1603:05	SubIndex 005	5. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1603:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x7000:11, 16

**Index 1604 ENC RxPDO-Map Control Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	ENC RxPDO-Map Control Ch.2	PDO Mapping RxPDO 5	UINT8	RO	0x0D (13 <sub>dec</sub> )
1604:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1604:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7010:02, 1
1604:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x03 (Set Counter))	UINT32	RO	0x7010:03, 1
1604:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7010:04, 1
1604:05	SubIndex 005	5. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1604:06	SubIndex 006	6. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1604:07	SubIndex 007	7. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x09 (Set software gate))	UINT32	RO	0x7010:09, 1
1604:08	SubIndex 008	8. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x0A (Set counter on latch extern on positive edge))	UINT32	RO	0x7010:0A, 1
1604:09	SubIndex 009	9. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x0B (Set counter on latch extern on negative edge))	UINT32	RO	0x7010:0B, 1
1604:0A	SubIndex 010	10. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1604:0B	SubIndex 011	11. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1604:0C	SubIndex 012	12. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1604:0D	SubIndex 013	13. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x11 (Set counter value))	UINT32	RO	0x7010:11, 32

**Index 1605 ENC RxPDO-Map Control Compact Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	ENC RxPDO-Map Control Ch.2	PDO Mapping RxPDO 6	UINT8	RO	0x0D (13 <sub>dec</sub> )
1605:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1605:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7010:02, 1
1605:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x03 (Set Counter))	UINT32	RO	0x7010:03, 1
1605:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7010:04, 1
1605:05	SubIndex 005	5. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1605:06	SubIndex 006	6. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1605:07	SubIndex 007	7. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x09 (Set software gate))	UINT32	RO	0x7010:09, 1
1605:08	SubIndex 008	8. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x0A (Set counter on latch extern on positive edge))	UINT32	RO	0x7010:0A, 1
1605:09	SubIndex 009	9. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x0B (Set counter on latch extern on negative edge))	UINT32	RO	0x7010:0B, 1
1605:0A	SubIndex 010	10. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1605:0B	SubIndex 011	11. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1605:0C	SubIndex 012	12. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1605:0D	SubIndex 013	13. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x11 (Set counter value))	UINT32	RO	0x7010:11, 16

**Index 1606 ENC RxPDO-Map Control Counter Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1606:0	ENC RxPDO-Map Control Counter Ch.2	PDO Mapping RxPDO 7	UINT8	RO	0x06 (6 <sub>dec</sub> )
1606:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1606:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x03 (Set counter))	UINT32	RO	0x7010:03, 1
1606:03	SubIndex 003	3. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1606:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x09 (Set software gate))	UINT32	RO	0x7010:09, 1
1606:05	SubIndex 005	5. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1606:06	SubIndex 006	6. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x11 (Set counter value))	UINT32	RO	0x7010:11, 32

**Index 1607 ENC RxPDO-Map Control Compact Counter Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1607:0	ENC RxPDO-Map Control Compact Ch.2	PDO Mapping RxPDO 8	UINT8	RO	0x06 (06 <sub>dec</sub> )
1607:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1607:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x03 (Set counter))	UINT32	RO	0x7010:03, 1
1607:03	SubIndex 003	3. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1607:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x09 (Set software gate))	UINT32	RO	0x7010:09, 1
1607:05	SubIndex 005	5. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1607:06	SubIndex 006	6. PDO Mapping entry (object 0x7010 (ENC Outputs Ch.2), entry 0x11 (Set counter value))	UINT32	RO	0x7010:11, 16

**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 <sub>dec</sub> )
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 02 1A 03 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 <sub>dec</sub> )
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 02 1A 03 1A

**Index 1802 ENC TxPDO-Par Status Counter Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1802:0	ENC TxPDO-Par Status Counter Ch.1	PDO parameter TxPDO 3	UINT8	RO	0x06 (6 <sub>dec</sub> )
1802:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 3	OCTET-STRING[14]	RO	00 1A 01 1A 03 1A

**Index 1803 ENC TxPDO-Par Status Compact Counter Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1803:0	ENC TxPDO-Par Status Compact Counter Ch.1	PDO parameter TxPDO 4	UINT8	RO	0x06 (6 <sub>dec</sub> )
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	00 1A 01 1A 02 1A

**Index 1804 ENC TxPDO-Par Frequency Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1804:0	ENC TxPDO-Par Frequency Ch.1	PDO parameter TxPDO 5	UINT8	RO	0x06 (6 <sub>dec</sub> )
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	05 1A 00 00 00 00

**Index 1805 ENC TxPDO-Par Frequency Compact Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1805:0	ENC TxPDO-Par Frequency Compact Ch.1	PDO parameter TxPDO 6	UINT8	RO	0x06 (6 <sub>dec</sub> )
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	04 1A 00 00 00 00

**Index 1806 ENC TxPDO-Par Period Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1806:0	ENC TxPDO-Par Period Ch.1	PDO parameter TxPDO 7	UINT8	RO	0x06 (6 <sub>dec</sub> )
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	07 1A 00 00 00 00

**Index 1807 ENC TxPDO-Par Period Compact Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1807:0	ENC TxPDO-Par Period Compact Ch.1	PDO parameter TxPDO 8	UINT8	RO	0x06 (6 <sub>dec</sub> )
1807:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 8	OCTET-STRING[14]	RO	06 1A 00 00 00 00

**Index 1808 ENC TxPDO-Par Status Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1808:0	ENC TxPDO-Par Status Ch.2	PDO parameter TxPDO 9	UINT8	RO	0x06 (6 <sub>dec</sub> )
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 0A 1A 0B 1A

**Index 1809 ENC TxPDO-Par Status Compact Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1809:0	ENC TxPDO-Par Status compact Ch.2	PDO parameter TxPDO 10	UINT8	RO	0x06 (6 <sub>dec</sub> )
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 0A 1A 0B 1A

**Index 180A ENC TxPDO-Par Status Counter Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
180A:0	ENC TxPDO-Par Status Counter Ch.2	PDO parameter TxPDO 11	UINT8	RO	0x06 (6 <sub>dec</sub> )
180A:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 11	OCTET-STRING[14]	RO	08 1A 09 1A 0B 1A

**Index 180B ENC TxPDO-Par Status Compact Counter Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
180B:0	ENC TxPDO-Par Status Compact Counter Ch.2	PDO parameter TxPDO 12	UINT8	RO	0x06 (6 <sub>dec</sub> )
180B:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 12	OCTET-STRING[14]	RO	08 1A 09 1A 0A 1A

**Index 180C ENC TxPDO-Par Frequency Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
180C:0	ENC TxPDO-Par Frequency Ch.2	PDO parameter TxPDO 13	UINT8	RO	0x06 (6 <sub>dec</sub> )
180C:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 13	OCTET-STRING[14]	RO	0D 1A 00 00 00 00

**Index 180D ENC TxPDO-Par Frequency Compact Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
180D:0	ENC TxPDO-Par Frequency Compact Ch.2	PDO parameter TxPDO 14	UINT8	RO	0x06 (6 <sub>dec</sub> )
180D:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 14	OCTET-STRING[14]	RO	0C 1A 00 00 00 00

**Index 180E ENC TxPDO-Par Period Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
180E:0	ENC TxPDO-Par Period Ch.2	PDO parameter TxPDO 15	UINT8	RO	0x06 (6 <sub>dec</sub> )
180E:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 15	OCTET-STRING[14]	RO	0F 1A 00 00 00 00

**Index 180F ENC TxPDO-Par Period Compact Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
180F:0	ENC TxPDO-Par Period Compact Ch.2	PDO parameter TxPDO 16	UINT8	RO	0x06 (6 <sub>dec</sub> )
180F:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 16	OCTET-STRING[14]	RO	0E 1A 00 00 00 00

**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 <sub>dec</sub> )
1A00:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 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 (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (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 (1 bits align)	UINT32	RO	0x0000:00, 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 (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:12	SubIndex 018	18. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:13	SubIndex 019	19. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 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 (3 bits align)	UINT32	RO	0x0000:00, 3
1A00:16	SubIndex 022	22. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A00:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:11, 32
1A00:18	SubIndex 024	24. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:12, 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 <sub>dec</sub> )
1A01:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 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 (1 bits align)	UINT32	RO	0x0000:00, 1
1A01:07	SubIndex 007	7. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A01:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A01:09	SubIndex 009	9. PDO Mapping entry (object 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 (1 bits align)	UINT32	RO	0x0000:00, 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 (1 bits align)	UINT32	RO	0x0000:00, 1
1A01:12	SubIndex 018	18. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A01:13	SubIndex 019	19. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 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 (3 bits align)	UINT32	RO	0x0000:00, 3
1A01:16	SubIndex 022	22. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A01:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:11, 16
1A01:18	SubIndex 024	24. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:12, 16



**Index 1A02 ENC TxPDO-Map Status Counter Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	ENC TxPDO-Map Status Counter Ch.1	PDO Mapping TxPDO 3	UINT8	RO	0x0B (11 <sub>dec</sub> )
1A02:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A02:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x03 (Set counter done))	UINT32	RO	0x6000:03, 1
1A02:03	SubIndex 003	3. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A02:04	SubIndex 004	4. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A02:05	SubIndex 005	5. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0D (Diag))	UINT32	RO	0x6002:0D, 1
1A02:06	SubIndex 006	6. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0E (TxPDO State))	UINT32	RO	0x6002:0E, 1
1A02:07	SubIndex 007	7. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0F (Input cycle counter))	UINT32	RO	0x6002:0F, 2
1A02:08	SubIndex 008	8. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x11 (Software gate valid))	UINT32	RO	0x6002:11, 1
1A02:09	SubIndex 009	9. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A02:0A	SubIndex 010	10. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1A02:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:11, 32

**Index 1A03 ENC TxPDO-Map Status Compact Counter Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A03:0	ENC TxPDO-Map Status Compact Counter Ch.1	PDO Mapping TxPDO 4	UINT8	RO	0x0B (11 <sub>dec</sub> )
1A03:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A03:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x03 (Set counter done))	UINT32	RO	0x6000:03, 1
1A03:03	SubIndex 003	3. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A03:04	SubIndex 004	4. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A03:05	SubIndex 005	5. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0D (Diag))	UINT32	RO	0x6002:0D, 1
1A03:06	SubIndex 006	6. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0E (TxPDO State))	UINT32	RO	0x6002:0E, 1
1A03:07	SubIndex 007	7. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x0F (Input cycle counter))	UINT32	RO	0x6002:0F, 2
1A03:08	SubIndex 008	8. PDO Mapping entry (object 0x6002 (ENC Inputs status Ch.1), entry 0x11 (Software gate valid))	UINT32	RO	0x6002:11, 1
1A03:09	SubIndex 009	9. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A03:0A	SubIndex 010	10. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1A03:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:11, 16

**Index 1A04 ENC TxPDO-Map Frequency Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	ENC TxPDO-Map Frequency Ch.1	PDO Mapping TxPDO 5	UINT8	RO	0x01 (1 <sub>dec</sub> )
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x13 (Frequency value))	UINT32	RO	0x6000:13, 32

**Index 1A05 ENC TxPDO-Map Frequency Compact Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	ENC TxPDO-Map Frequency Compact Ch.1	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1 <sub>dec</sub> )
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x01 (Frequency value))	UINT32	RO	0x6000:13, 16

**Index 1A06 ENC TxPDO-Map Period Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A06:0	ENC TxPDO-Map Period Ch.1	PDO Mapping TxPDO 7	UINT8	RO	0x01 (1 <sub>dec</sub> )
1A06:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x14 (Period value))	UINT32	RO	0x6000:14, 32

**Index 1A07 ENC TxPDO-Map Period Compact Ch.1**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A07:0	ENC TxPDO-Map Period Compact Ch.1	PDO Mapping TxPDO 8	UINT8	RO	0x01 (1 <sub>dec</sub> )
1A07:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x14 (Period value))	UINT32	RO	0x6000:14, 16

**Index 1A08 ENC TxPDO-Map Status Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	ENC TxPDO-Map Status Ch.1	PDO Mapping TxPDO 9	UINT8	RO	0x18 (24 <sub>dec</sub> )
1A08:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x02 (Latch extern valid))	UINT32	RO	0x6010:02, 1
1A08:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x03 (Set counter done))	UINT32	RO	0x6010:03, 1
1A08:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x04 (Counter underflow))	UINT32	RO	0x6010:04, 1
1A08:05	SubIndex 005	5. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x05 (Counter overflow))	UINT32	RO	0x6010:05, 1
1A08:06	SubIndex 006	6. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:07	SubIndex 007	7. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:09	SubIndex 009	9. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x09 (Status of input A))	UINT32	RO	0x6010:09, 1
1A08:0A	SubIndex 010	10. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x0A (Status of input B))	UINT32	RO	0x6010:0A, 1
1A08:0B	SubIndex 011	11. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:0C	SubIndex 012	12. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x0C (Status of input gate))	UINT32	RO	0x6010:0C, 1
1A08:0D	SubIndex 013	13. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0D (Diag))	UINT32	RO	0x6012:0D, 1
1A08:0E	SubIndex 014	14. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0E (TxPDO State))	UINT32	RO	0x6012:0E, 1
1A08:0F	SubIndex 015	15. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0F (Input cycle counter))	UINT32	RO	0x6012:0F, 2
1A08:10	SubIndex 016	16. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x11 (Software gate valid))	UINT32	RO	0x6012:11, 1
1A08:11	SubIndex 017	17. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:12	SubIndex 018	18. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:13	SubIndex 019	19. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:14	SubIndex 020	20. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x15 (Counter value out of range))	UINT32	RO	0x6012:15, 1
1A08:15	SubIndex 021	21. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A08:16	SubIndex 022	22. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A08:17	SubIndex 023	23. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x11 (Counter value))	UINT32	RO	0x6010:11, 32
1A08:18	SubIndex 024	24. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x12 (Latch value))	UINT32	RO	0x6010:12, 32



**Index 1A09 ENC TxPDO-Map Status Compact Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A09:0	ENC TxPDO-Map Status Ch.1	PDO Mapping TxPDO 10	UINT8	RO	0x18 (24 <sub>dec</sub> )
1A09:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x02 (Latch extern valid))	UINT32	RO	0x6010:02, 1
1A09:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x03 (Set counter done))	UINT32	RO	0x6010:03, 1
1A09:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x04 (Counter underflow))	UINT32	RO	0x6010:04, 1
1A09:05	SubIndex 005	5. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x05 (Counter overflow))	UINT32	RO	0x6010:05, 1
1A09:06	SubIndex 006	6. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:07	SubIndex 007	7. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:09	SubIndex 009	9. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x09 (Status of input A))	UINT32	RO	0x6010:09, 1
1A09:0A	SubIndex 010	10. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x0A (Status of input B))	UINT32	RO	0x6010:0A, 1
1A09:0B	SubIndex 011	11. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:0C	SubIndex 012	12. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x0C (Status of input gate))	UINT32	RO	0x6010:0C, 1
1A09:0D	SubIndex 013	13. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0D (Diag))	UINT32	RO	0x6012:0D, 1
1A09:0E	SubIndex 014	14. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0E (TxPDO State))	UINT32	RO	0x6012:0E, 1
1A09:0F	SubIndex 015	15. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0F (Input cycle counter))	UINT32	RO	0x6012:0F, 2
1A09:10	SubIndex 016	16. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x11 (Software gate valid))	UINT32	RO	0x6012:11, 1
1A09:11	SubIndex 017	17. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:12	SubIndex 018	18. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:13	SubIndex 019	19. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:14	SubIndex 020	20. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x15 (Counter value out of range))	UINT32	RO	0x6012:15, 1
1A09:15	SubIndex 021	21. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A09:16	SubIndex 022	22. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A09:17	SubIndex 023	23. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x11 (Counter value))	UINT32	RO	0x6010:11, 16
1A09:18	SubIndex 024	24. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x12 (Latch value))	UINT32	RO	0x6010:12, 16

**Index 1A0A ENC TxPDO-Map Status Counter Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0A:0	ENC TxPDO-Map Status Counter Ch.2	PDO Mapping TxPDO 11	UINT8	RO	0x0B (11 <sub>dec</sub> )
1A0A:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A0A:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x03 (Set counter done))	UINT32	RO	0x6010:03, 1
1A0A:03	SubIndex 003	3. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A0A:04	SubIndex 004	4. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A0A:05	SubIndex 005	5. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0D (Diag))	UINT32	RO	0x6012:0D, 1
1A0A:06	SubIndex 006	6. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0E (TxPDO State))	UINT32	RO	0x6012:0E, 1
1A0A:07	SubIndex 007	7. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0F (Input cycle counter))	UINT32	RO	0x6012:0F, 2
1A0A:08	SubIndex 008	8. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x11 (Software gate valid))	UINT32	RO	0x6012:11, 1
1A0A:09	SubIndex 009	9. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A0A:0A	SubIndex 010	10. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1A0A:0B	SubIndex 011	11. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x11 (Counter value))	UINT32	RO	0x6010:11, 32

**Index 1A0B ENC TxPDO-Map Status Compact Counter Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0B:0	ENC TxPDO-Map Status Compact Counter Ch.2	PDO Mapping TxPDO 12	UINT8	RO	0x0B (11 <sub>dec</sub> )
1A0B:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A0B:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x03 (Set counter done))	UINT32	RO	0x6010:03, 1
1A0B:03	SubIndex 003	3. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A0B:04	SubIndex 004	4. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A0B:05	SubIndex 005	5. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0D (Diag))	UINT32	RO	0x6012:0D, 1
1A0B:06	SubIndex 006	6. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0E (TxPDO State))	UINT32	RO	0x6012:0E, 1
1A0B:07	SubIndex 007	7. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x0F (Input cycle counter))	UINT32	RO	0x6012:0F, 2
1A0B:08	SubIndex 008	8. PDO Mapping entry (object 0x6012 (ENC Inputs status Ch.2), entry 0x11 (Software gate valid))	UINT32	RO	0x6012:11, 1
1A0B:09	SubIndex 009	9. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A0B:0A	SubIndex 010	10. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1A0B:0B	SubIndex 011	11. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x11 (Counter value))	UINT32	RO	0x6010:11, 16

**Index 1A0C ENC TxPDO-Map Frequency Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0C:0	ENC TxPDO-Map Frequency Ch.2	PDO Mapping TxPDO 13	UINT8	RO	0x01 (1 <sub>dec</sub> )
1A0C:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x13 (Frequency value))	UINT32	RO	0x6010:13, 32

**Index 1A0D ENC TxPDO-Map Frequency Compact Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0D:0	ENC TxPDO-Map Frequency Compact Ch.2	PDO Mapping TxPDO 14	UINT8	RO	0x01 (1 <sub>dec</sub> )
1A0D:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x13 (Frequency value))	UINT32	RO	0x6010:13, 16

**Index 1A0E ENC TxPDO-Map Period Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0E:0	ENC TxPDO-Map Period Ch.2	PDO Mapping TxPDO 15	UINT8	RO	0x01 (1 <sub>dec</sub> )
1A0E:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x14 (Period value))	UINT32	RO	0x6010:14, 32

**Index 1A0F ENC TxPDO-Map Period Compact Ch.2**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0F:0	ENC TxPDO-Map Period Compact Ch.2	PDO Mapping TxPDO 16	UINT8	RO	0x01 (1 <sub>dec</sub> )
1A0F:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (ENC Inputs Ch.2), entry 0x14 (Period value))	UINT32	RO	0x6010:14, 16

**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 <sub>dec</sub> )
1C00:01	SubIndex 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1 <sub>dec</sub> )
1C00:02	SubIndex 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2 <sub>dec</sub> )
1C00:03	SubIndex 003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3 <sub>dec</sub> )
1C00:04	SubIndex 004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4 <sub>dec</sub> )

**Index 1C12 RxPDO assign**

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2 <sub>dec</sub> )
1C12:01	SubIndex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632 <sub>dec</sub> )
1C12:02	SubIndex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1604 (5636 <sub>dec</sub> )

**Index 1C13 TxPDO assign**

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x02 (2 <sub>dec</sub> )
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656 <sub>dec</sub> )
1C13:02	SubIndex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A08 (6664 <sub>dec</sub> )

## 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 <sub>dec</sub> )
1C32:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> <li>0: Free Run</li> <li>1: Synchron with SM 2 Event</li> <li>2: DC-Mode - Synchron with SYNC0 Event</li> <li>3: DC-Mode - Synchron with SYNC1 Event</li> </ul>	UINT16	RW	0x0001 (1 <sub>dec</sub> )
1C32:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> <li>Free Run: Cycle time of the local timer</li> <li>Synchron with SM 2 Event: Master cycle time</li> <li>DC mode: SYNC0/SYNC1 Cycle Time</li> </ul>	UINT32	RW	0x000F4240 (1000000 <sub>dec</sub> )
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> <li>Bit 0 = 1: free run is supported</li> <li>Bit 1 = 1: Synchron with SM 2 Event is supported</li> <li>Bit 2-3 = 01: DC mode is supported</li> <li>Bit 4-5 = 10: Output Shift with SYNC1 event (only DC mode)</li> <li>Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)</li> </ul>	UINT16	RO	0x0807 (2055 <sub>dec</sub> )
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000101D0 (66000 <sub>dec</sub> )
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:07	Minimum delay time	Min. time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:08	Command	With this entry the real required process data provision time can be measured. <ul style="list-style-type: none"> <li>0: Measurement of the local cycle time is stopped</li> <li>1: Measurement of the local cycle time is started</li> </ul> <p>The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09, 0x1C33:03 [▶ 189], 0x1C33:06 [▶ 188], 0x1C33:09 [▶ 189] are updated with the maximum measured values. For a subsequent measurement the measured values are reset</p>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
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 <sub>dec</sub> )
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 <sub>dec</sub> )
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 <sub>dec</sub> )

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 <sub>dec</sub> )
1C33:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> <li>0: Free Run</li> <li>1: Synchron with SM 3 Event (no outputs available)</li> <li>2: DC - Synchron with SYNC0 Event</li> <li>3: DC - Synchron with SYNC1 Event</li> <li>34: Synchron with SM 2 Event (outputs available)</li> </ul>	UINT16	RW	0x0022 (34 <sub>dec</sub> )
1C33:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> <li>Free Run: Cycle time of the local timer</li> <li>Synchron with SM 2 Event: Master cycle time</li> </ul> DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000 <sub>dec</sub> )
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> <li>Bit 0: free run is supported</li> <li>Bit 1: Synchron with SM 2 Event is supported (outputs available)</li> <li>Bit 1: Synchron with SM 3 Event is supported (no outputs available)</li> <li>Bit 2-3 = 01: DC mode is supported</li> <li>Bit 4-5 = 01: Input Shift through local event (outputs available)</li> <li>Bit 4-5 = 10: Input Shift with SYNC1 Event (no outputs available)</li> <li>Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [► 188] or 0x1C33:08)</li> </ul>	UINT16	RO	0x0807 (2055 <sub>dec</sub> )
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000101D0 (66000 <sub>dec</sub> )
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x000101D0 (66000 <sub>dec</sub> )
1C33:07	Minimum delay time	Min. time between SYNC1 event and the reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:08	Command	With this entry the real required process data provision time can be measured. <ul style="list-style-type: none"> <li>0: Measurement of the local cycle time is stopped</li> <li>1: Measurement of the local cycle time is started</li> </ul> The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09 [► 188], 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 <sub>dec</sub> )
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
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 <sub>dec</sub> )
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 <sub>dec</sub> )
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 <sub>dec</sub> )

**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 <sub>dec</sub> )
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16 <sub>dec</sub> )
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0002 (2 <sub>dec</sub> )

**Index F008 Code word**

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0	Code word	reserved	UINT32	RW	0x00000000 (0 <sub>dec</sub> )

**Index F010 Module list**

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Maximum subindex	UINT8	RW	0x02 (2 <sub>dec</sub> )
F010:01	SubIndex 001	reserved	UINT32	RW	0x000001FF (511 <sub>dec</sub> )
F010:02	SubIndex 002	reserved	UINT32	RW	0x000001FF (511 <sub>dec</sub> )

## 7 Appendix

### 7.1 EtherCAT AL Status Codes

For detailed information please refer to the [EtherCAT system description](#).

### 7.2 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.

<b>NOTICE</b>			
<b>Risk of damage to the device!</b>			
Pay attention to the instructions for firmware updates on the <a href="#">separate page</a> [▶ 191].			
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!			

EL5122			
Hardware (HW)	Firmware	Revision no.	Release date
00 - 01*	01	EL5122-0000-0016	2020/06
	02*	EL5122-0000-0017	2023/11

\*) 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.

### 7.3 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.

<b>NOTICE</b>
<b>Only use TwinCAT 3 software!</b>
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 <a href="#">Beckhoff website</a> .
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 NOVDRAM 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. ELxxx-xxx\_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.



### 7.3.1 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:

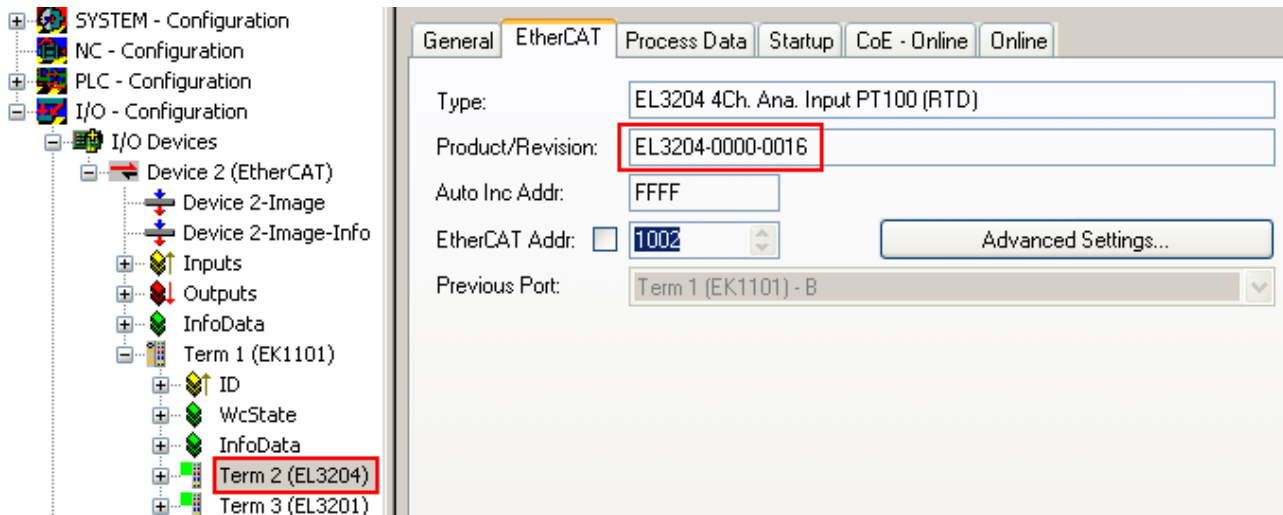


Fig. 162: Device identifier consisting of name EL3204-0000 and revision -0016

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**

**i** 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:

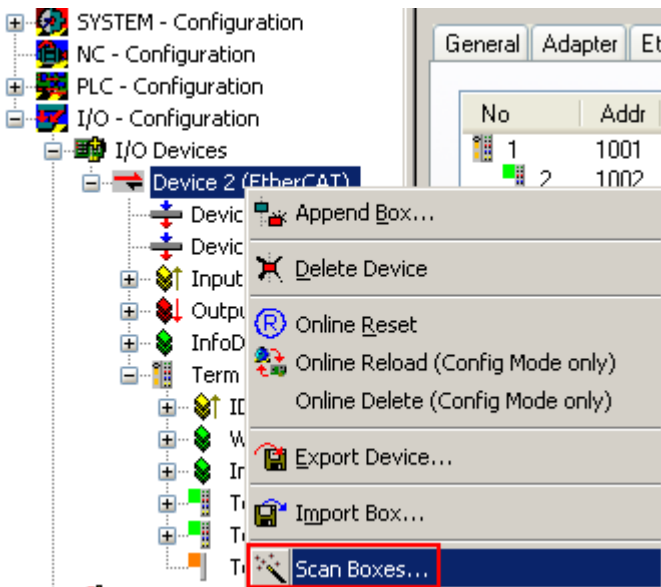


Fig. 163: Scan the subordinate field by right-clicking on the EtherCAT device

If the found field matches the configured field, the display shows



Fig. 164: Configuration is identical

otherwise a change dialog appears for entering the actual data in the configuration.

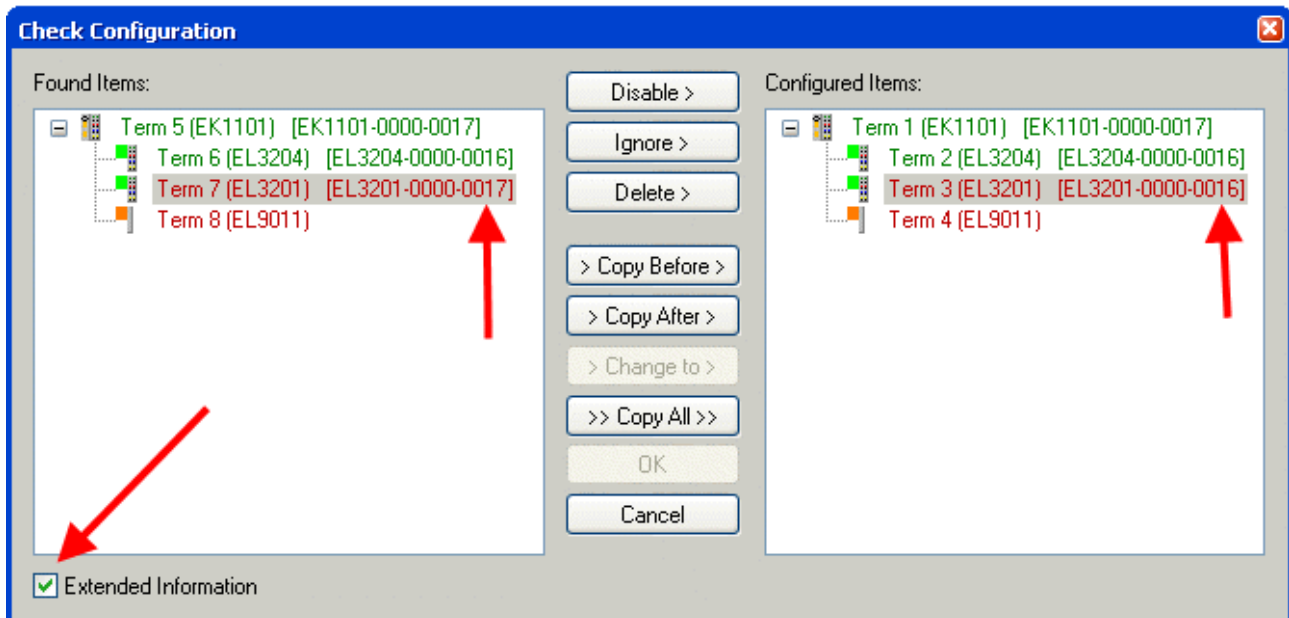


Fig. 165: Change dialog

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*

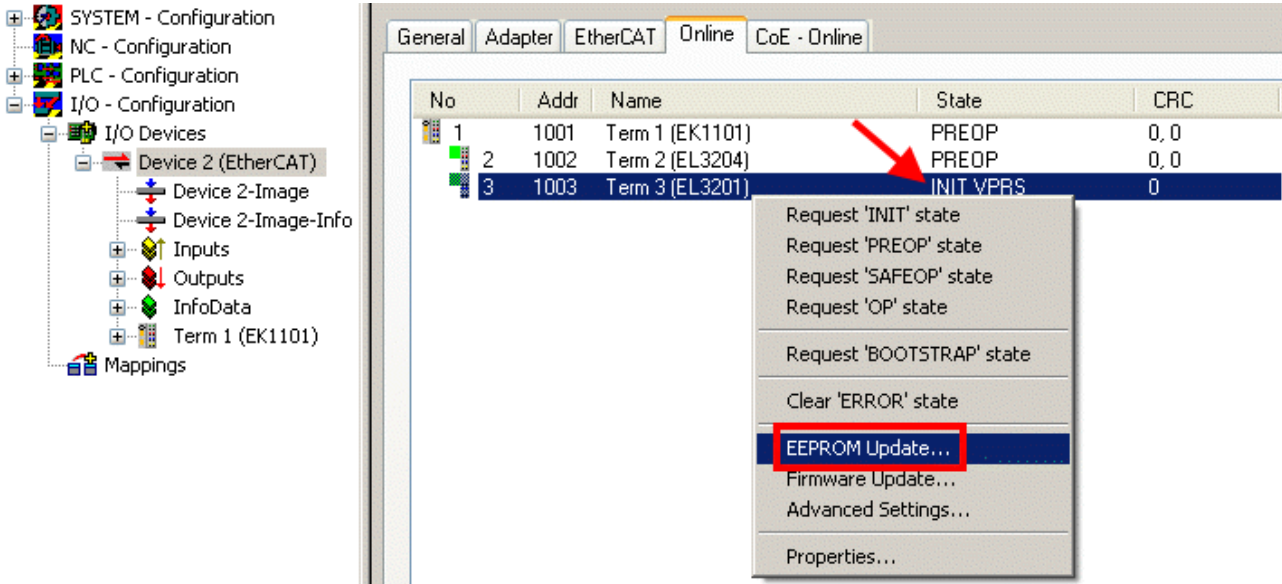


Fig. 166: EEPROM Update

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.

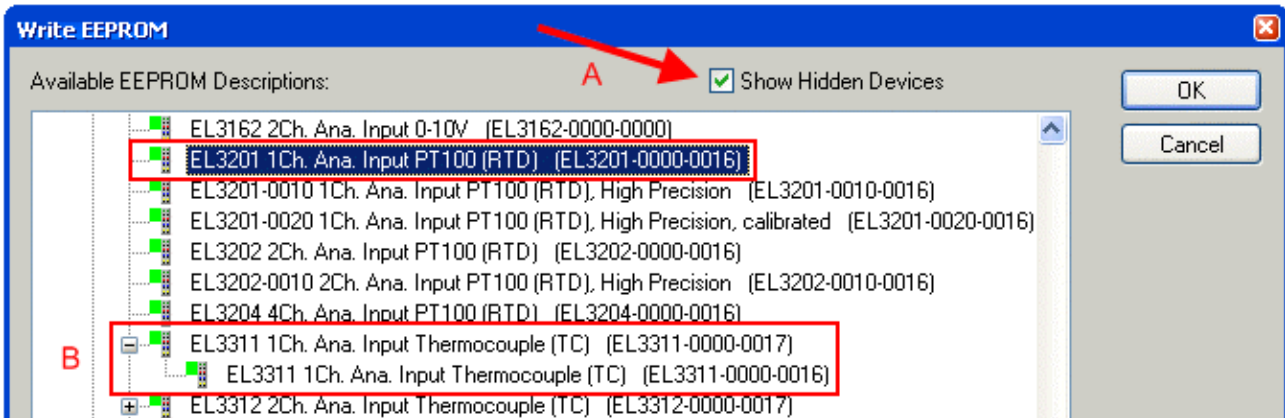


Fig. 167: Selecting the new ESI

A progress bar in the System Manager shows the progress. Data are first written, then verified.

**● The change only takes effect after a restart.**

**i** 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.

## 7.3.2 Firmware explanation

### Determining the firmware version

#### Determining the version via the 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

### i

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.

The screenshot displays the TwinCAT System Manager interface. On the left, a tree view shows the configuration structure, with 'Term 2 (EL3204)' selected. The main window shows the 'CoE - Online' tab. A table lists CoE entries with the following data:

Index	Name	Flags	Value
1000	Device type	RO	0x01401389 (20976521)
1008	Device name	RO	EL3204-0000
1009	Hardware version	RO	00
100A	Software version	RO	03
1011:0	Restore default parameters	RU	>   <

The 'Advanced Settings' dialog box is open, showing the 'Dictionary' section. The 'Online' radio button is selected under 'via SDO Information', and 'All Objects' is selected in the dictionary list.

Fig. 168: 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*.

## 7.3.3 Updating controller firmware \*.efw

#### ● CoE directory

### i

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*.

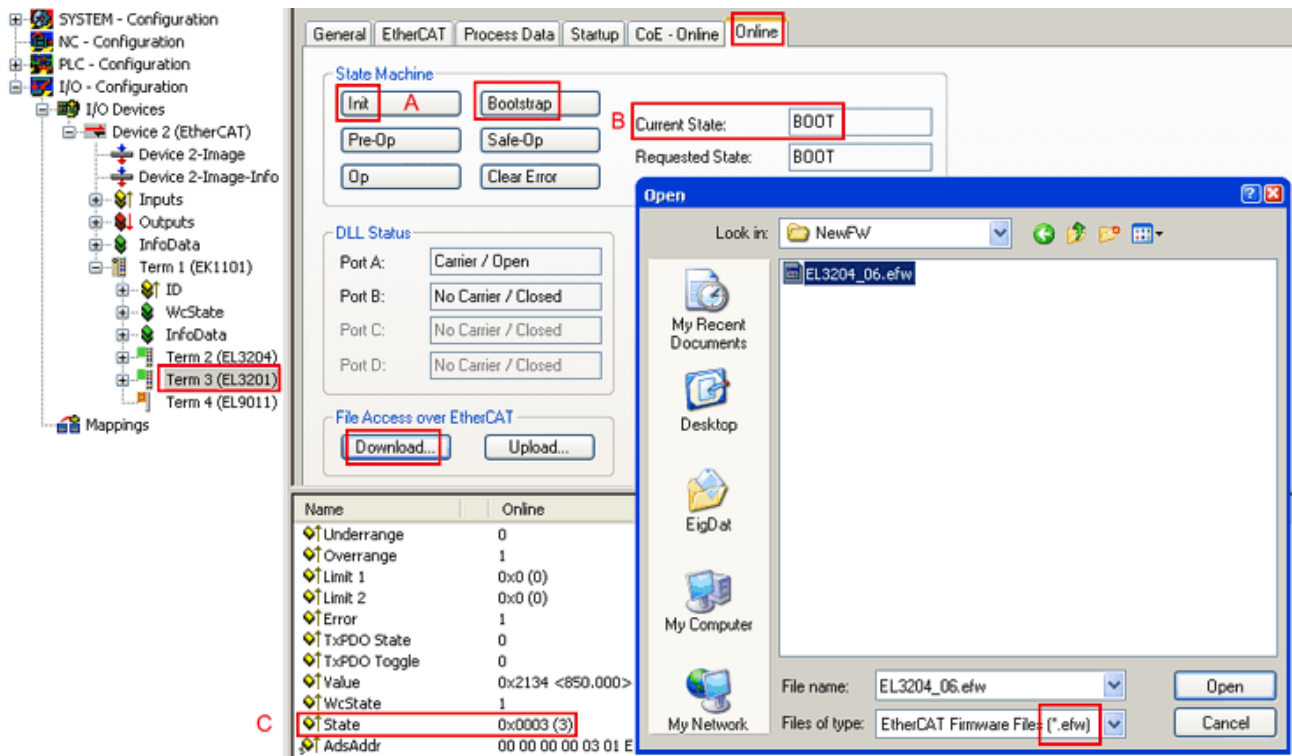
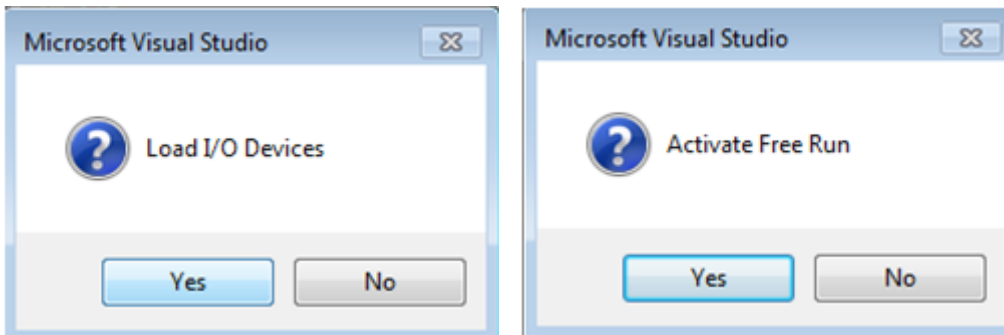


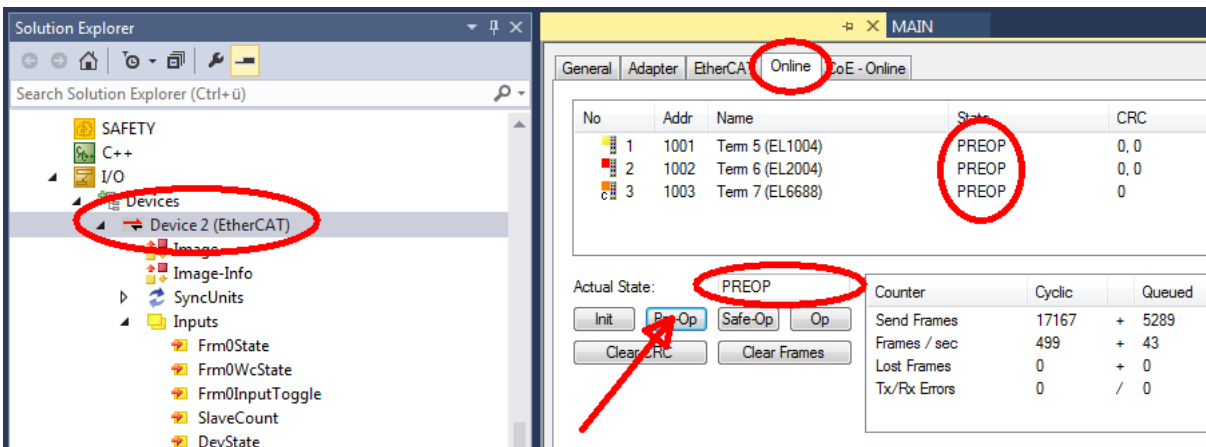
Fig. 169: Firmware Update

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  $\geq 1$  ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.

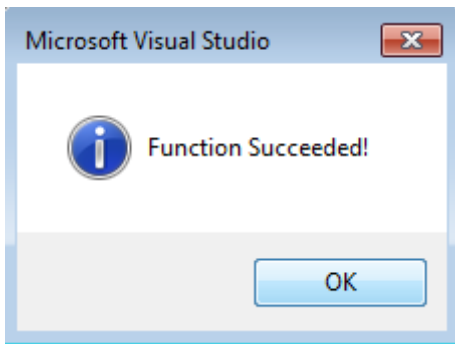


- Switch EtherCAT Master to PreOP



- 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.

### 7.3.4 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 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.

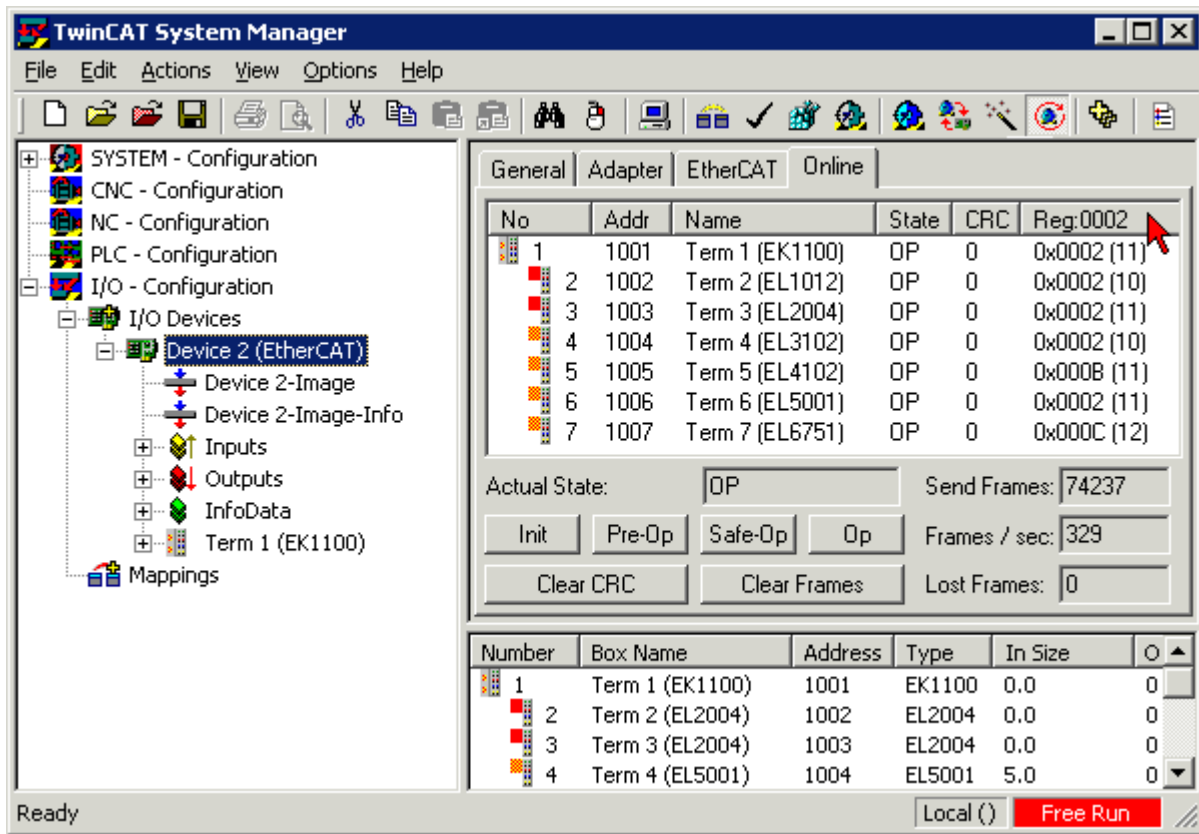


Fig. 170: FPGA firmware version definition

If the column *Reg:0002* is not displayed, right-click the table header and select *Properties* in the context menu.

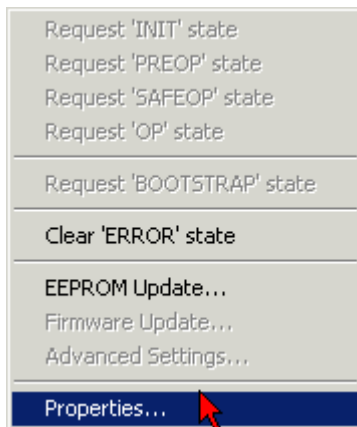


Fig. 171: Context menu *Properties*

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.



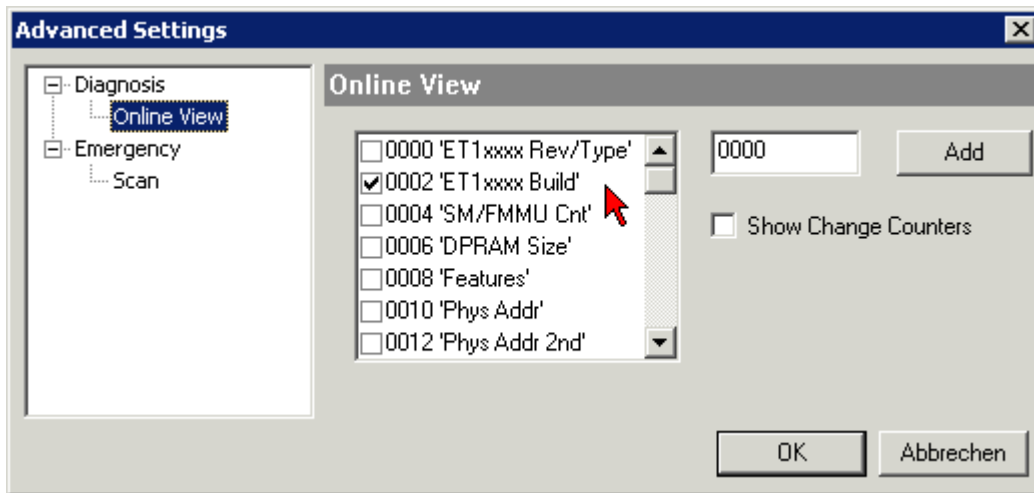


Fig. 172: Dialog *Advanced Settings*

### 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!

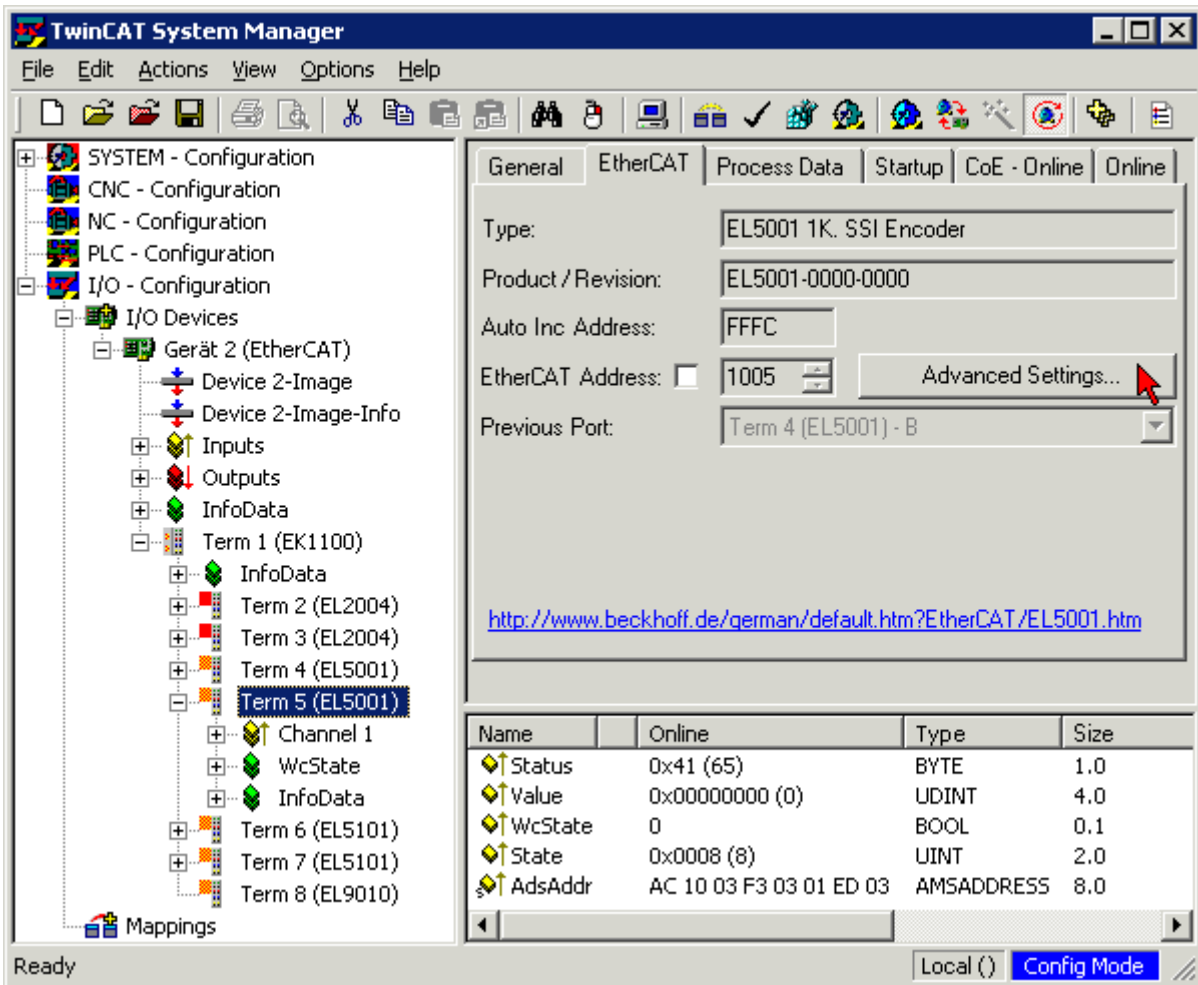
### Updating an EtherCAT device

The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

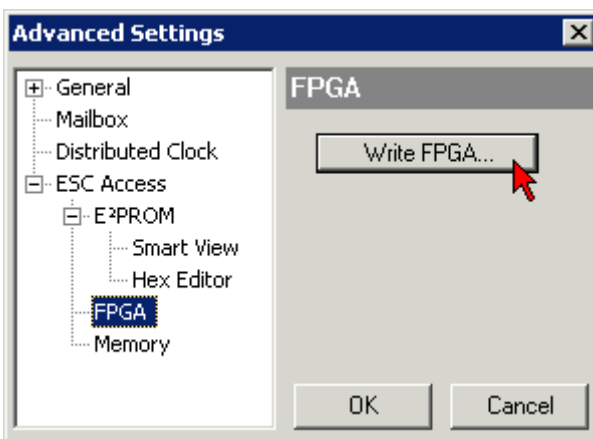
- Switch TwinCAT system to ConfigMode/FreeRun with cycle time  $\geq 1$  ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.



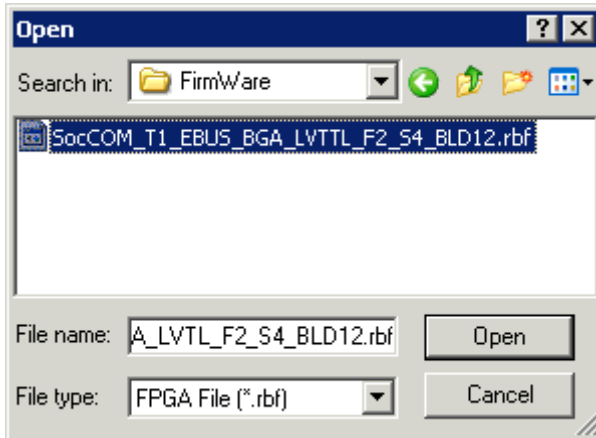
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



- The *Advanced Settings* dialog appears. Under *ESC Access/E<sup>2</sup>PROM/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!

## 7.3.5 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.

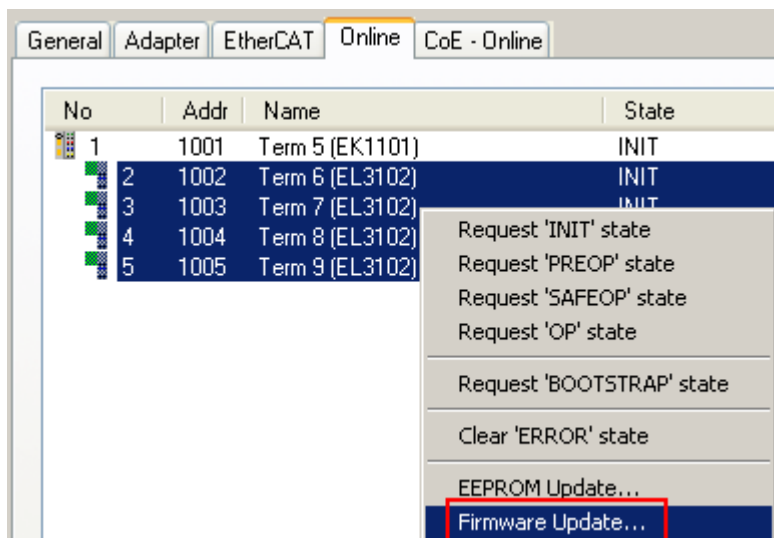


Fig. 173: Multiple selection and firmware update

Select the required slaves and carry out the firmware update in BOOTSTRAP mode as described above.

## 7.4 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*).

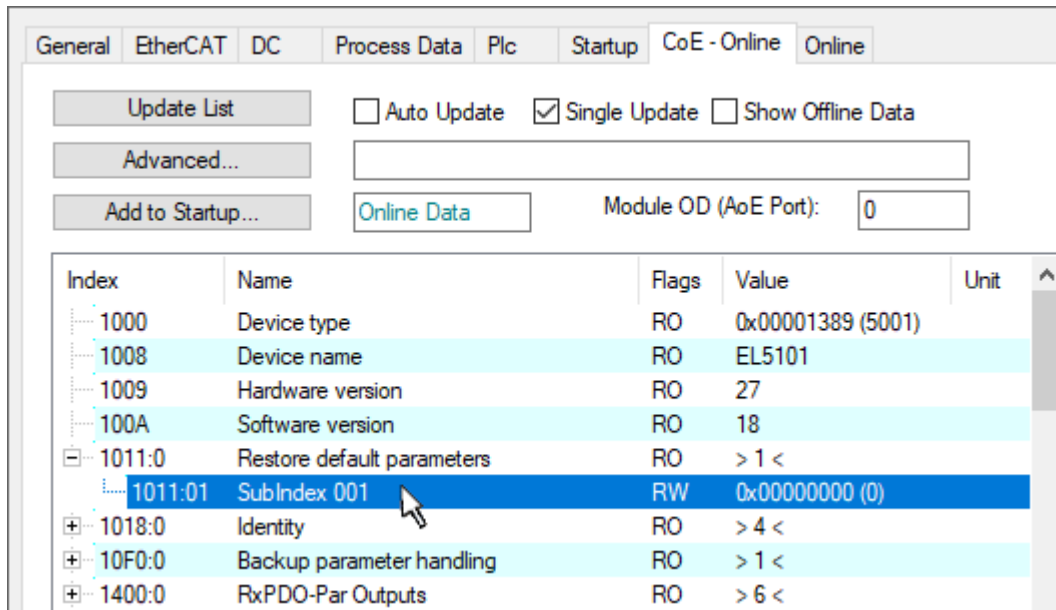


Fig. 174: Selecting the *Restore default parameters* PDO

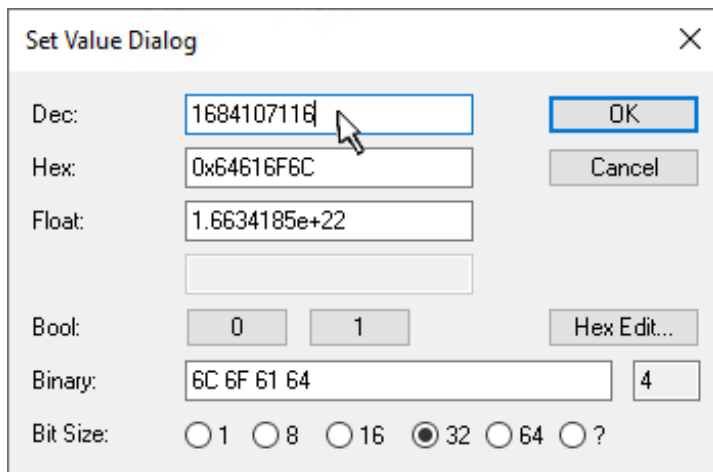


Fig. 175: Entering a restore value in the Set Value dialog

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.

---

**i 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.

---

## 7.5 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](http://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](mailto:support@beckhoff.com)  
web: [www.beckhoff.com/support](http://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](mailto:service@beckhoff.com)  
web: [www.beckhoff.com/service](http://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](mailto:info@beckhoff.com)  
web: [www.beckhoff.com](http://www.beckhoff.com)



More Information:  
**[www.beckhoff.com/EL5122](http://www.beckhoff.com/EL5122)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
Phone: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

