

# BECKHOFF New Automation Technology

Documentation | EN

## EL2202, EL2252

2-Channel Digital Output Terminals, 24 V DC, 0.5 A



EtherCAT®



# Table of contents

<b>1</b>	<b>Foreword</b> .....	<b>5</b>
1.1	Product overview Digital Output Terminals 24 V DC, 0.5 A .....	5
1.2	Notes on the documentation .....	6
1.3	Guide through documentation .....	7
1.4	Safety instructions .....	8
1.5	Documentation issue status .....	9
1.6	Version identification of EtherCAT devices .....	10
1.6.1	General notes on marking .....	10
1.6.2	Version identification of EL terminals .....	11
1.6.3	Beckhoff Identification Code (BIC) .....	12
1.6.4	Electronic access to the BIC (eBIC) .....	14
<b>2</b>	<b>Product description</b> .....	<b>16</b>
2.1	EL2202 .....	16
2.1.1	EL2202 - Introduction .....	16
2.1.2	EL2202 - Technical data .....	17
2.1.3	EL2202 - LEDs and connection .....	18
2.2	EL2252 .....	19
2.2.1	EL2252 - Introduction .....	19
2.2.2	EL2252 - Technical data .....	20
2.2.3	EL2252 - LEDs and connection .....	21
<b>3</b>	<b>Basics communication</b> .....	<b>22</b>
3.1	EtherCAT basics .....	22
3.2	EtherCAT cabling – wire-bound .....	22
3.3	General notes for setting the watchdog .....	23
3.4	EtherCAT State Machine .....	25
3.5	CoE - Interface: notes .....	27
3.6	Distributed Clock .....	28
<b>4</b>	<b>Mounting and wiring</b> .....	<b>29</b>
4.1	Instructions for ESD protection .....	29
4.2	Explosion protection .....	30
4.2.1	ATEX - Special conditions (standard temperature range).....	30
4.2.2	IECEX - Special conditions .....	31
4.2.3	Continuative documentation for ATEX and IECEX .....	32
4.3	Output protection during DC switching of inductive loads (overvoltage limitation).....	33
4.4	UL notice .....	36
4.5	Installation on mounting rails .....	37
4.6	Installation positions .....	40
4.7	Positioning of passive Terminals .....	42
4.8	Installation instructions for enhanced mechanical load capacity .....	43
4.9	Connection .....	44
4.9.1	Connection system .....	44
4.9.2	Wiring .....	46
4.9.3	Shielding .....	47

4.10	Note - power supply .....	48
4.11	Disposal .....	49
<b>5</b>	<b>Commissioning .....</b>	<b>50</b>
5.1	TwinCAT Quick Start.....	50
5.1.1	TwinCAT 2 .....	53
5.1.2	TwinCAT 3 .....	63
5.2	TwinCAT Development Environment.....	77
5.2.1	Installation of the TwinCAT real-time driver .....	77
5.2.2	Notes regarding ESI device description .....	83
5.2.3	TwinCAT ESI Updater.....	87
5.2.4	Distinction between Online and Offline .....	87
5.2.5	OFFLINE configuration creation.....	88
5.2.6	ONLINE configuration creation .....	93
5.2.7	EtherCAT subscriber configuration .....	101
5.3	General Commissioning Instructions for an EtherCAT Slave .....	111
<b>6</b>	<b>EL2202 - Basic function principles.....</b>	<b>119</b>
6.1	Delivery state .....	119
6.2	Switching characteristics .....	119
6.3	Tristate mode (high-resistance outputs).....	119
6.4	Watchdog .....	120
6.5	Operating modes.....	120
6.6	EL2202-0100 Advanced Settings.....	121
6.7	Switching from EL2202-0000 to EL2202-0100 and back.....	122
<b>7</b>	<b>EL2252 - Basic function principles.....</b>	<b>124</b>
7.1	Switching characteristics .....	124
7.2	Watchdog .....	124
7.3	Functioning.....	124
7.4	Application - brief overview .....	125
7.5	Detailed description of application with TwinCAT .....	125
<b>8</b>	<b>Example programs .....</b>	<b>130</b>
8.1	Sample program TwinCAT 2.....	131
8.2	Sample program TwinCAT 3.....	134
<b>9</b>	<b>Appendix.....</b>	<b>136</b>
9.1	EtherCAT AL Status Codes .....	136
9.2	Firmware compatibility .....	136
9.3	Restoring the delivery state.....	136
9.4	Support and Service.....	138

# 1 Foreword

## 1.1 Product overview Digital Output Terminals 24 V DC, 0.5 A

[EL2202 \[► 16\]](#) – 2 channel digital output terminal, 24 V<sub>DC</sub>, T<sub>ON</sub>/T<sub>OFF</sub> 1 μs, push-pull outputs with tristate

[EL2252 \[► 19\]](#) – 2 channel digital output terminal, 24 V<sub>DC</sub>, with time stamp, with tristate

## 1.2 Notes on the documentation

### Intended audience

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

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.

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

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

### Disclaimer

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

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

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

### Trademarks

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

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.



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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

### Third-party brands

Trademarks of third parties may be used in this documentation. You can find the trademark notices here:

<https://www.beckhoff.com/trademarks>

## 1.3 Guide through documentation

### NOTICE



#### Further components of documentation

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

Title	Description
<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>Explosion Protection for Terminal Systems</b> ( <a href="#">PDF</a> )	Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx
<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](#).

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

## 1.4 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.

## 1.5 Documentation issue status

Version	Comment
3.9.0	<ul style="list-style-type: none"> <li>• Update chapter "Technical Data"</li> <li>• Update chapter "Mounting and wiring"</li> <li>• Update structure</li> </ul>
3.8	<ul style="list-style-type: none"> <li>• Update chapter "Technical Data"</li> <li>• Update structure</li> </ul>
3.7	<ul style="list-style-type: none"> <li>• Update chapter "UL notice"</li> <li>• Update structure</li> </ul>
3.6	<ul style="list-style-type: none"> <li>• Update chapter "EL2252 - sample programs"</li> <li>• Update structure</li> </ul>
3.5	<ul style="list-style-type: none"> <li>• Update chapter "EL2252 - sample programs"</li> </ul>
3.4	<ul style="list-style-type: none"> <li>• Update chapter "Technical data"</li> <li>• Update chapter "EL2252 - sample programs"</li> <li>• Update structure</li> </ul>
3.3	<ul style="list-style-type: none"> <li>• Update chapter "Technical data"</li> <li>• Addenda chapter "Instructions for ESD protection"</li> </ul>
3.2	<ul style="list-style-type: none"> <li>• Update chapter "Detailed description of application with TwinCAT"</li> <li>• Update structure</li> </ul>
3.1	<ul style="list-style-type: none"> <li>• Update chapter "Notes on the documentation"</li> <li>• Correction of Technical data</li> <li>• Update chapter "TwinCAT 2.1x" -&gt; "TwinCAT Development Environment" and "TwinCAT Quick Start"</li> </ul>
3.0	<ul style="list-style-type: none"> <li>• Migration</li> <li>• Update structure</li> </ul>
2.1	<ul style="list-style-type: none"> <li>• Update chapter "Technical data"</li> <li>• Addenda chapter "Installation instructions for enhanced mechanical load capacity"</li> <li>• Update structure</li> </ul>
2.0	<ul style="list-style-type: none"> <li>• New structure</li> <li>• Update chapter "Watchdog"</li> </ul>
1.4	<ul style="list-style-type: none"> <li>• Notes on firmware update added; notes on trademarks added</li> </ul>
1.3	<ul style="list-style-type: none"> <li>• Description of sample program added</li> </ul>
1.2	<ul style="list-style-type: none"> <li>• Technical data added, new sample program added</li> </ul>
1.1	<ul style="list-style-type: none"> <li>• Technical data added, sample program added</li> </ul>
1.0	<ul style="list-style-type: none"> <li>• Technical data added</li> </ul>
0.1	<ul style="list-style-type: none"> <li>• Provisional documentation for EL2202, EL2252</li> </ul>

## 1.6 Version identification of EtherCAT devices

### 1.6.1 General notes on marking

#### Designation

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

- family key
- type
- version
- revision

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

#### Notes

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

## 1.6.2 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

### 1.6.3 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.

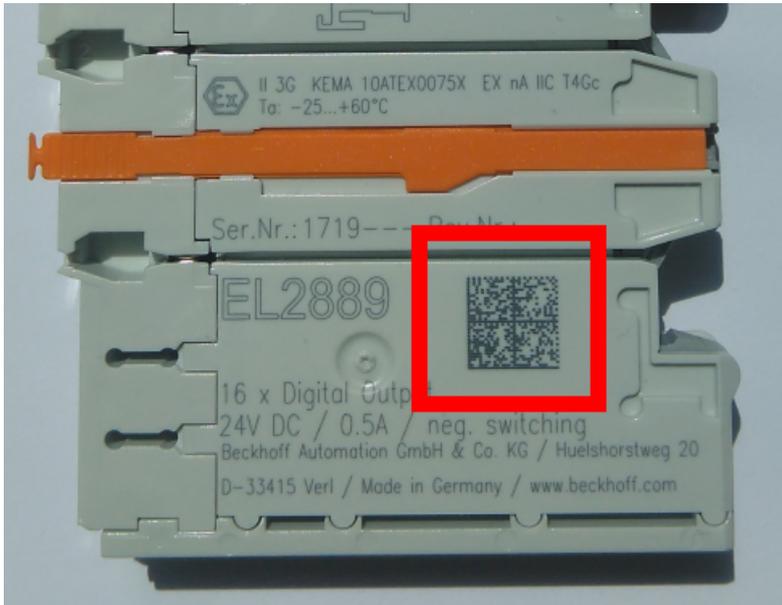


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

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>SBTN</b> k4p562d7
3	Article description	<b>Beckhoff article description, e.g. EL1008</b>	1K	32	<b>1KEL</b> 1809
4	Quantity	<b>Quantity in packaging unit, e.g. 1, 10, etc.</b>	Q	6	<b>Q1</b>
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	12	<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**SBTN**k4p562d7**1KEL**1809 **Q1** **51S**678294

Accordingly as DMC:



Fig. 3: Example DMC **1P**072222**SBTN**k4p562d7**1KEL**1809 **Q1** **51S**678294

**BTN**

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

**NOTICE**

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

## 1.6.4 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.

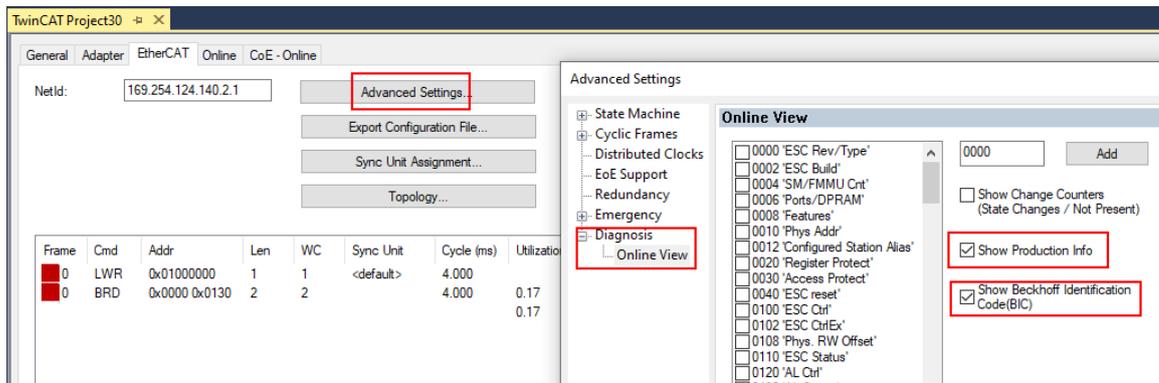
### EtherCAT devices (IP20, IP67)

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	1P158442SBTN0008jckp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

- The object 0x10E2 will be preferentially introduced into stock products in the course of necessary firmware revision.
- From TwinCAT 3.1. build 4024.24, the functions *FB\_EcCoEReadBIC* and *FB\_EcCoEReadBTN* for reading into the PLC are available in the *Tc2\_EtherCAT* library from v3.3.19.0
- The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2\_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 EL2202

#### 2.1.1 EL2202 - Introduction

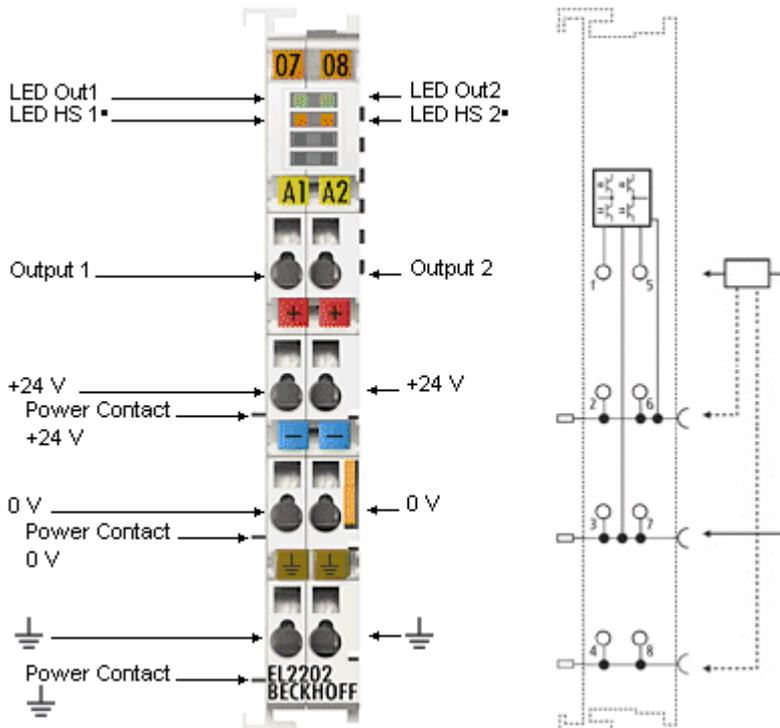


Fig. 4: EL2202

#### 2 channel digital output terminal, 24 V<sub>DC</sub>, T<sub>ON</sub>/T<sub>OFF</sub> 1 μs, push-pull outputs with tristate

The EL2202 digital output terminal connects the binary control signals from the automation device on to the actuators at the process level with electrical isolation. This terminal benefits from very small output delay and is therefore suitable for signals requiring particularly fast output. The EtherCAT terminal supports distributed clocks, i.e. the output data can be monitored synchronous with other data from terminals with distributed clock support. The accuracy across the system is  $\ll 1 \mu\text{s}$ . The EtherCAT terminal has a push-pull output that enables the output to be connected actively to 24 V, 0 V or high-resistance. The EL2202 contains 2 channels, the signal state of which is indicated via LEDs. The state "high-resistance" (tristate) is indicated through a further LED for each channel.

The EL2202 can be used in 2 operating modes that can be changed online via the Beckhoff TwinCAT System Manager:

- EL2202-0000 The outputs are switched as soon as the EtherCAT frame supplying process data was verified as valid.
- EL2202-0100 The outputs are cyclically switched through SYNC0 that runs synchronous with the task that supplies the EL2202-0100 with process data.

The delivery state is EL2202-0000.

#### Quick links

- [EtherCAT basics \[► 22\]](#)
- [Creation of the configuration \[► 77\]](#)
- [Basics \[► 119\]](#)

## 2.1.2 EL2202 - Technical data

Technical data	EL2202
Digital outputs	2
Rated voltage of the outputs	24 V <sub>DC</sub> (-15%/+20%)
Load type	ohmic, inductive, lamp load
Output current per channel	max. 0.5 A (short-circuit-proof in push mode)
Current limitation	typ. 4 A / 150 µs
Output switching time T <sub>ON</sub> /T <sub>OFF</sub>	< 1 µs
Output stage	Push-pull
Supply voltage for electronic	via the E-bus
Current consumption via E-bus	typ. 130 mA
Current consumption via power voltage	typ. 30 mA + load
Electrical isolation	500 V (E-bus/field voltage)
Distributed Clocks (DC)	Yes
Accuracy Distributed Clocks (DC)	<< 1 µs
Accuracy output signal	EL2202-0100: accuracy Distributed Clocks + output switching time
Bit width in process image	4 bit
Configuration	no address or configuration settings required
Weight	approx. 60 g
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting <a href="#">[▶ 37]</a>	on 35 mm mounting rail conforms to EN 60715
Enhanced mechanical load capacity	yes, see also <a href="#">installation instructions [▶ 43]</a> for enhanced mechanical load capacity
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Marking / Approval <sup>*)</sup>	CE, EAC, UKCA, CCC cULus <a href="#">[▶ 36]</a> , ATEX <a href="#">[▶ 30]</a> , IECEx <a href="#">[▶ 31]</a>

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

### Ex markings

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

### 2.1.3 EL2202 - LEDs and connection

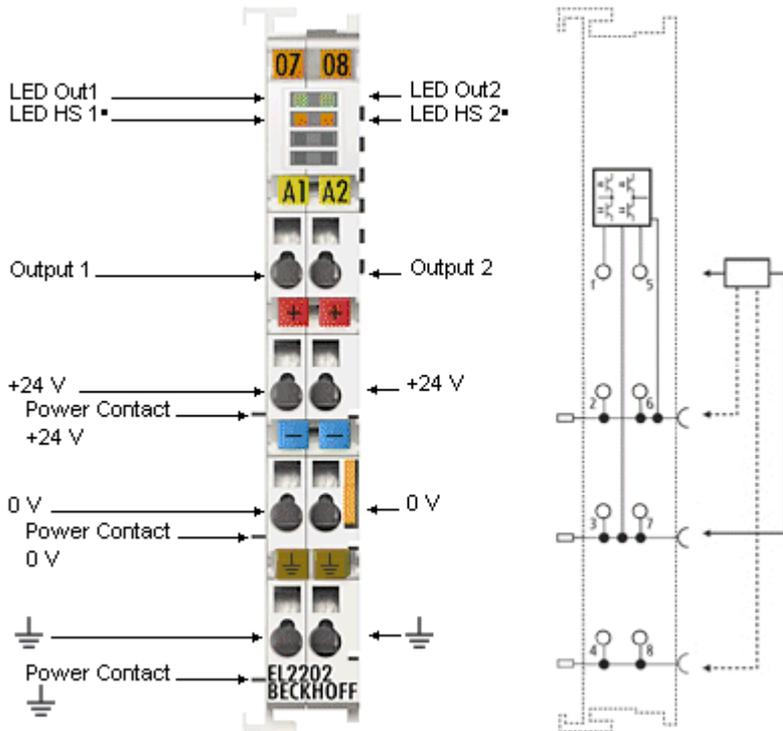


Fig. 5: EL2202

#### LEDs

LED	Color	Meaning	
OUT 1	green	off	No output signal is present at the respective output
OUT 2		on	+24 V output signal at the corresponding output
HS 1•	orange	off	Output switched according to LED OUT1 or LED OUT2
HS 2•		on	Output 1 or 2 switched to high-resistance

#### Connection EL2202

Terminal point		Description
Name	No.	
Output 1	1	Output 1
+ 24 V	2	+24 V (internally connected to terminal point 6 and positive power contact)
0 V	3	0 V (internally connected to terminal point 7 and negative power contact)
⏏	4	Ground
Output 2	5	Output 2
+ 24 V	6	24 V (internally connected to terminal point 2 and positive power contact)
0 V	7	0 V (internally connected to terminal point 3 and negative power contact)
⏏	8	Ground

## 2.2 EL2252

### 2.2.1 EL2252 - Introduction

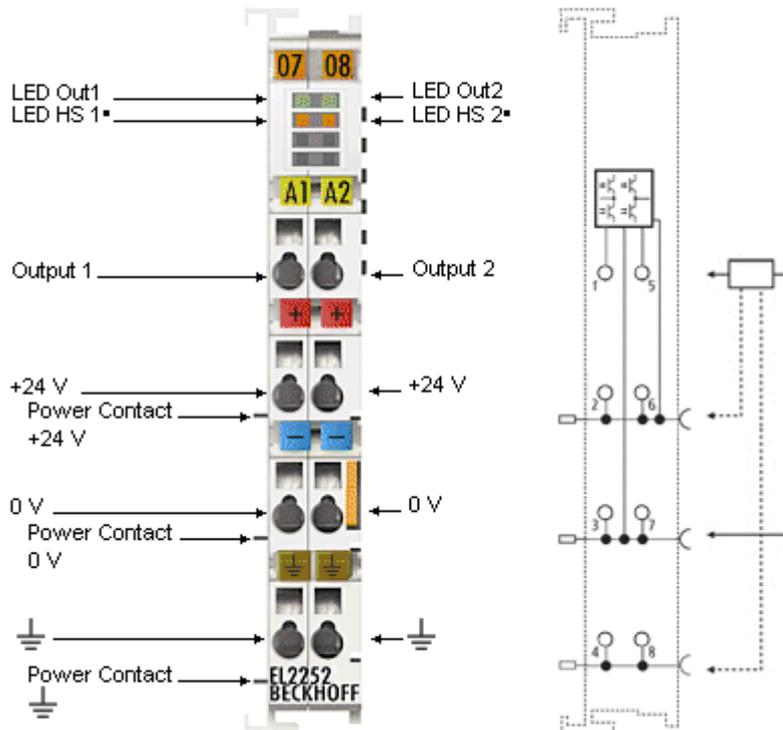


Fig. 6: EL2252

#### 2 channel digital output terminal with time stamp

The EL2252 digital output terminal connects the binary control signals from the automation device on to the actuators at the process level with electrical isolation. This terminal benefits from very small output delay and is therefore suitable for signals requiring particularly fast output. The EtherCAT terminal supports distributed clocks, i.e. the output data can be monitored synchronous with other data from terminals with distributed clock support. The accuracy across the system is  $\ll 1 \mu\text{s}$ . The EtherCAT terminal has a push-pull output that enables the output to be connected actively to 24 V, 0 V or high-resistance. The EL2252 contains 2 channels, the signal state of which is indicated via LEDs.

The time when the EL2252 switches its outputs must be specified. The time is transferred as process data into the terminal and activated through a bit sequence. The terminal then waits without further user input until the switching event occurs. The specified time is based on the distributed clock time in ns from 1 January 2000, 00:00 hours. During each EtherCAT cycle only *one* switching event can occur in the EL2252.

#### Quick links

- [EtherCAT basics \[► 22\]](#)
- [Creation of the configuration \[► 77\]](#)
- [Basics \[► 124\]](#)

**2.2.2 EL2252 - Technical data**

Technical data	EL2252
Digital outputs	2
Rated voltage of the outputs	24 V <sub>DC</sub> (-15%/+20%)
Load type	ohmic, inductive, lamp load
Output current per channel	max. 0.5 A (short-circuit-proof in push mode)
Current limitation	typ. 4 A / 150 µs
Output switching time T <sub>ON</sub> /T <sub>OFF</sub>	< 1 µs
Output stage	Push-pull
Supply voltage for electronic	via the E-bus
Current consumption via E-bus	typ. 130 mA
Current consumption via power voltage	typ. 30 mA + load
Electrical isolation	500 V (E-bus/field voltage)
Distributed Clocks (DC)	Yes
Accuracy Distributed Clocks (DC)	<< 1 µs
Time stamp resolution in the terminal	10 ns
Accuracy output signal	accuracy Distributed Clocks + output switching time
Bit width in process image	max. 17 byte + 4 bit
Configuration	no address or configuration settings required
Weight	approx. 60 g
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting [ <a href="#">▶ 37</a> ]	on 35 mm mounting rail conforms to EN 60715
Enhanced mechanical load capacity	yes, see also <a href="#">installation instructions ▶ 43</a> for enhanced mechanical load capacity
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Marking / Approval <sup>*)</sup>	CE, EAC, UKCA, CCC cULus [ <a href="#">▶ 36</a> ], ATEX [ <a href="#">▶ 30</a> ], IECEx [ <a href="#">▶ 31</a> ]

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

**Ex markings**

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

### 2.2.3 EL2252 - LEDs and connection

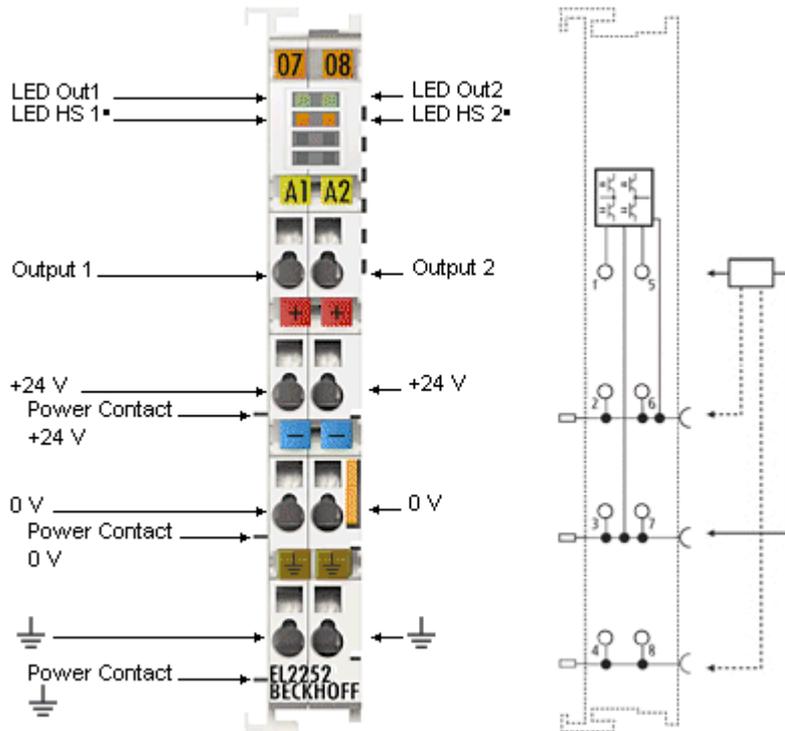


Fig. 7: EL2252

#### LEDs

LED	Color	Meaning	
OUT 1	green	off	No output signal is present at the respective output
OUT 2		on	+24 V output signal at the corresponding output
HS 1•	orange	off	Output switched according to LED OUT1 or LED OUT2
HS 2•		on	Output 1 or 2 switched to high-resistance

#### Connection EL2252

Terminal point		Description
Name	No.	
Output 1	1	Output 1
+ 24 V	2	+24 V (internally connected to terminal point 6 and positive power contact)
0 V	3	0 V (internally connected to terminal point 7 and negative power contact)
⏏	4	Ground
Output 2	5	Output 2
+ 24 V	6	24 V (internally connected to terminal point 2 and positive power contact)
0 V	7	0 V (internally connected to terminal point 3 and negative power contact)
⏏	8	Ground

## 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.

#### ● Recommended cables



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

Suitable cables for the connection of EtherCAT devices can be found on the [Beckhoff website!](#)

#### E-Bus supply

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

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

Number	Box Name	Addr...	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 features two watchdogs:

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

Their times are individually parameterized in TwinCAT as follows:

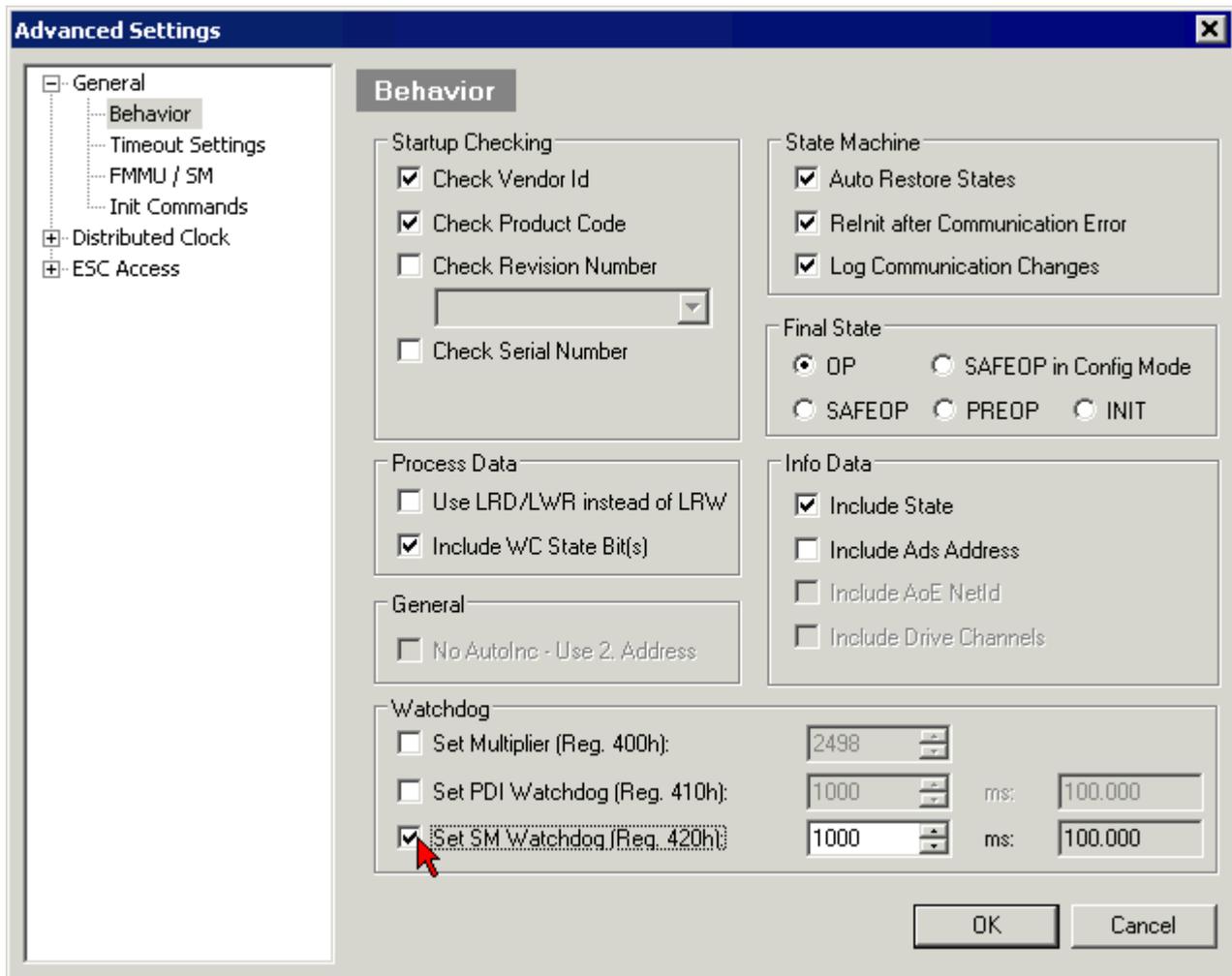


Fig. 9: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

#### Notes:

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

#### SM watchdog (SyncManager Watchdog)

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

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

The maximum possible watchdog time depends on the device. For example, for "simple" EtherCAT slaves (without firmware) with watchdog execution in the ESC it is usually up to 170 seconds. For complex EtherCAT slaves (with firmware) the SM watchdog function is usually parameterized via register 400h/420h

but executed by the microcontroller ( $\mu\text{C}$ ) and can be significantly lower. In addition, the execution may then be subject to a certain time uncertainty. Since the TwinCAT dialog may allow inputs up to 65535, a test of the desired watchdog time is recommended.

### PDI watchdog (Process Data Watchdog)

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

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

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

### Calculation

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

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

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

#### **⚠ CAUTION**

##### **Undefined state possible!**

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

#### **⚠ CAUTION**

##### **Damage of devices and undefined state possible!**

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

## 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
- Operational
- Bootstrap

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

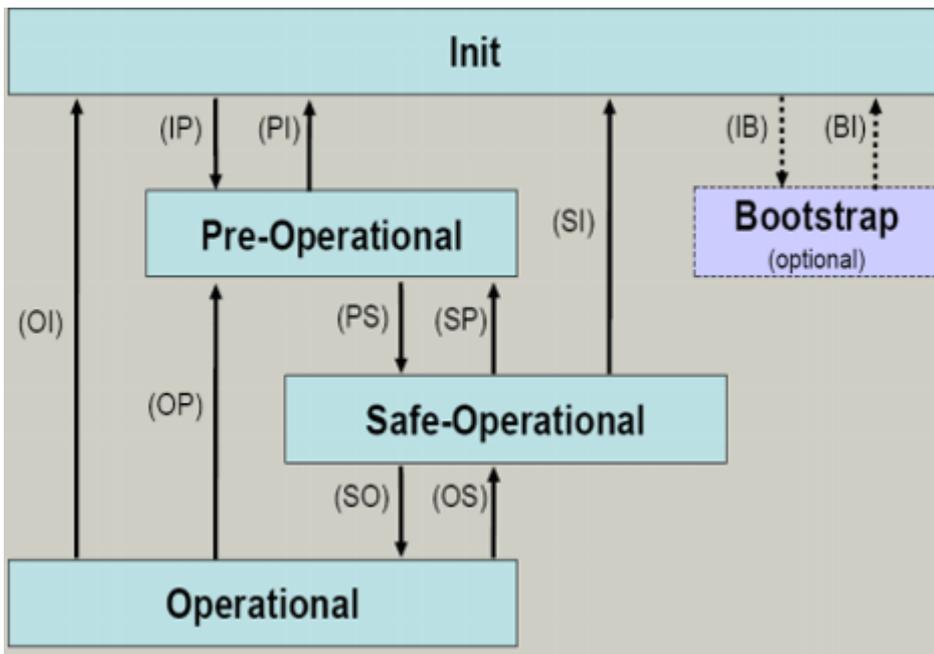


Fig. 10: States of the EtherCAT State Machine

**Init**

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

**Pre-Operational (Pre-Op)**

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

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

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

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

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

**● Outputs in SAFEOP state**

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

**Operational (Op)**

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

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

## Boot

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

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

## 3.5 CoE - Interface: notes

This device has no CoE.

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.

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

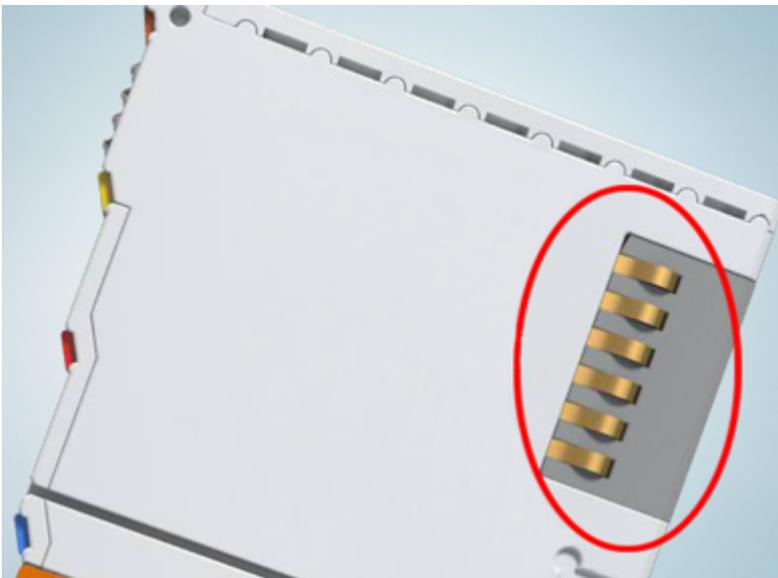


Fig. 11: Spring contacts of the Beckhoff I/O components

## 4.2 Explosion protection

### 4.2.1 ATEX - Special conditions (standard temperature range)

#### WARNING

**Observe the special conditions for the intended use of Beckhoff fieldbus components with standard temperature range in potentially explosive areas (directive 2014/34/EU)!**

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60079-15! The environmental conditions during use are thereby to be taken into account!
- For dust (only the fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9): The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-31 for group IIIA or IIIB and IP6X for group IIIC, taking into account the environmental conditions under which the equipment is used!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of 0 to 55°C for the use of Beckhoff fieldbus components standard temperature range in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

#### Standards

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

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010
- EN 60079-31:2013 (only for certificate no. KEMA 10ATEX0075 X Issue 9)

#### Marking

The Beckhoff fieldbus components with standard temperature range certified according to the ATEX directive for potentially explosive areas bear one of the following markings:



**II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: 0 ... +55°C**

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C  
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

or



**II 3G KEMA 10ATEX0075 X Ex nA nC IIC T4 Gc Ta: 0 ... +55°C**

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C  
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

## 4.2.2 IECEx - Special conditions

**⚠ WARNING**

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

- For gas: The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to IEC 60079-15, taking into account the environmental conditions under which the equipment is used!
- For dust (only the fieldbus components of certificate no. IECEx DEK 16.0078X Issue 3): The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-31 for group IIIA or IIIB and IP6X for group IIIC, taking into account the environmental conditions under which the equipment is used!
- The equipment shall only be used in an area of at least pollution degree 2, as defined in IEC 60664-1!
- Provisions shall be made to prevent the rated voltage from being exceeded by transient disturbances of more than 119 V!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range for the use of Beckhoff fieldbus components in potentially explosive areas!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The front hatch of certified units may only be opened if the supply voltage has been switched off or a non-explosive atmosphere is ensured!

### Standards

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

- EN 60079-0:2011
- EN 60079-15:2010
- EN 60079-31:2013 (only for certificate no. IECEx DEK 16.0078X Issue 3)

### Marking

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

Marking for fieldbus components of certificate no. IECEx DEK 16.0078X Issue 3:	<b>IECEx DEK 16.0078 X</b> <b>Ex nA IIC T4 Gc</b> <b>Ex tc IIIC T135°C Dc</b>
--	---

Marking for fieldbus components of certificates with later issues:	<b>IECEx DEK 16.0078 X</b> <b>Ex nA IIC T4 Gc</b>
--	--

### 4.2.3 Continulative documentation for ATEX and IECEx

#### NOTICE



#### Continulative documentation about explosion protection according to ATEX and IECEx

Pay also attention to the continuative documentation

#### **Ex. Protection for Terminal Systems**

Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx,

that is available for [download](#) within the download area of your product on the Beckhoff homepage [www.beckhoff.com](http://www.beckhoff.com)!

## 4.3 Output protection during DC switching of inductive loads (overvoltage limitation)

The digital outputs can be used to switch inductive loads, provided that the energy generated during shutdown is dissipated outside the output. The focus of this consideration is on DC-excited loads (typically 5-24 V DC in industry).

Two fields need to be considered.

- Selection of the protective circuit
- Dimensioning the protective circuit

Inductive loads, such as electromechanical relays, contactors or coils, want to remain in their steady state. If the driver (in our example, the digital output from Beckhoff) stops supplying (by being shutdown), the coil will attempt to allow the current to continue flowing due to its stored magnetic energy. It will then "draw" current from the switching output until the magnetic energy is used by parasitic effects. Without additional measures, a high negative voltage (ranging from several tens to thousands of volts, depending on the load) will occur at the switch output due to the self-induction of the coil. This voltage may result in immediate destruction, or repeated occurrences may cause damage. This applies to semiconductor switching outputs (e.g. EL20xx, EL28xx) as well as relay switching outputs (e.g. EL26xx).

Another disadvantage is that this short, high voltage (also known as a "transient") in the kV/ $\mu$ s range generates significant electromagnetic radiation, causing capacitive interference with neighboring devices and lines. Any damping/suppression components should therefore preferably be placed close to the load (i.e., at the point of origin) so that the transients do not even reach the line.

The following measures can be considered, among others. The attached figures show a relay as an example of a common inductive load.

### 1. Resistance parallel to inductance

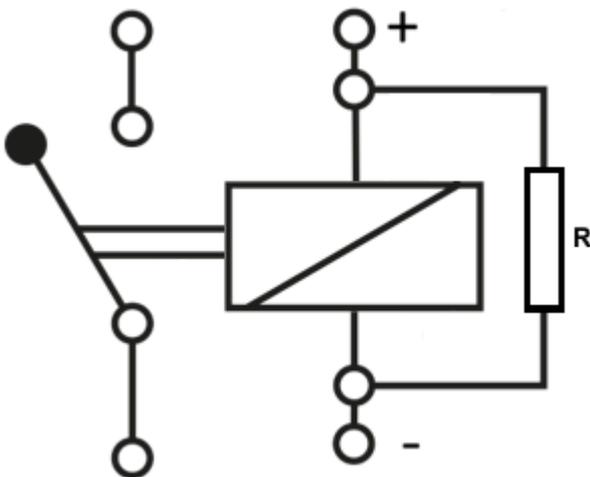


Fig. 12: Resistance parallel to inductance

It is a very simple method, but one in which current constantly flows uselessly past the coil. This circuit is also known as snubbers.

Note: An RC element is common for AC excitation. However, it must be designed very precisely to ensure optimum operation (time behavior, power loss).

### 2. Solution with semiconductors

- Short-circuit diode on the inductance, free-wheeling diode

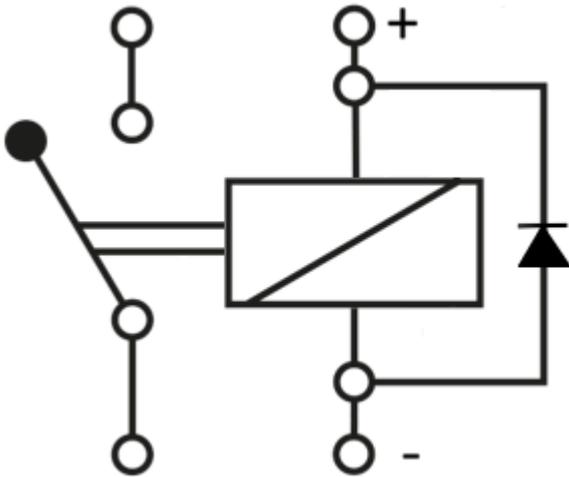


Fig. 13: Short-circuit diode on the inductance, free-wheeling diode

This solution is often the most appropriate for DC excitation and should be considered first. It reliably protects the output as the self-induction voltage is limited to approx. -1 V (even less if a Schottky diode is used). In addition, it is usually easy to dimension, as hardly any power is dissipated via the diode with small DC inductances.

However, the induction voltage only drops very slowly (a few ms), which can be disadvantageous for relays as a switching load or if fast shutdown is desired.

Note: In the event of expected power failures, however, this latching effect may be desirable to prevent the relay from dropping too early.

- Varistor or Zener diode, ideally a bidirectional suppressor diode (TVS= Transient Voltage Suppressor)

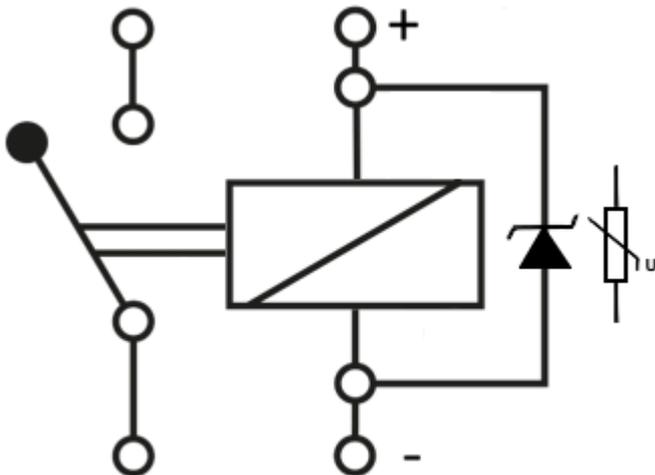


Fig. 14: Example circuit with Zener diode

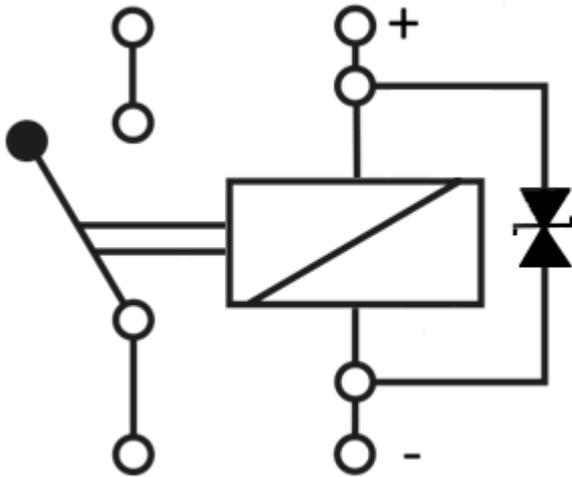


Fig. 15: Example circuit with bidirectional suppressor diode (Transient Voltage Suppressor)

This solution ensures that the load drops relatively quickly (a relay may be required). The coil is not short-circuited, but must dissipate its energy at a higher voltage. This means that the magnetic field can collapse quickly. It should be noted that the semiconductor can become very warm.

The reverse voltage of the Zener diode must be significantly higher than the regular operating voltage, otherwise normal operation is not possible. In a 24 V industrial system (usually up to approx. 29 V are permissible), for example, it is 35 V. Such diodes are also often used in 24 V DC power supplies to temporarily short-circuit transients!

Unfortunately, designing a semiconductor-based overvoltage limitation is not trivial. The following points must be observed:

1. How much (magnetic) energy must be dissipated?  
If the inductance of the load and the nominal current are known, the energy stored in the coil can be calculated using the formula  $W [J] = 1/2 * L [H] * I^2$ . Unfortunately, many manufacturers do not specify the inductance, so it may have to be determined by measurement.
2. What breaking energy [J, Ws, VAs] can the switching output withstand, and is this value sufficient?  
See Technical data.
3. How often must the energy be dissipated (switching frequency)?
4. Unfortunately, it is difficult to determine the achievable performance of the protective component because it depends on several factors, including how the generated heat is dissipated (mechanical connection = thermal resistance -> installation costs), the pulse frequency and the tolerable residual voltage level.
5. Therefore, the design typically requires an experimental, system-specific approach. This approach can begin with commercially available standard diodes, such as 1N4007/1000 V or TVS 600 W. Alternatively, you can select normative tests, for which a protective component will then be selected experimentally.

Corresponding components are commercially available

- as discrete, wired components from various electronic distributors/shops.
- as ready-to-install DIN rail accessories.
- as a special accessory for sensitive products (e.g. contactors).

## 4.4 UL notice

### ⚠ CAUTION



#### Application

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

### ⚠ CAUTION



#### Examination

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

### ⚠ CAUTION



#### For devices with Ethernet connectors

Not for connection to telecommunication circuits.

### Basic principles

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



## 4.5 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

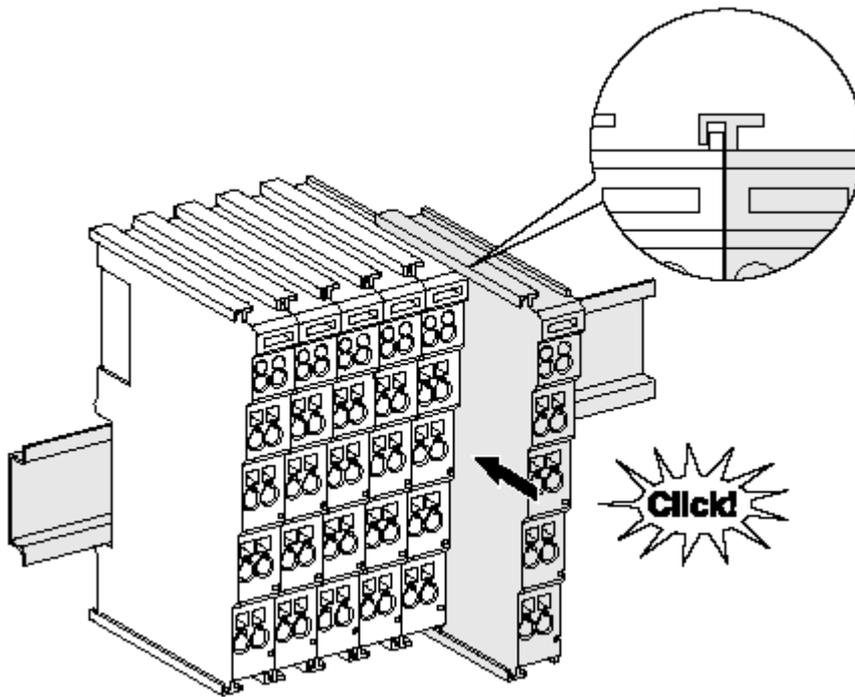


Fig. 16: Attaching on mounting rail

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

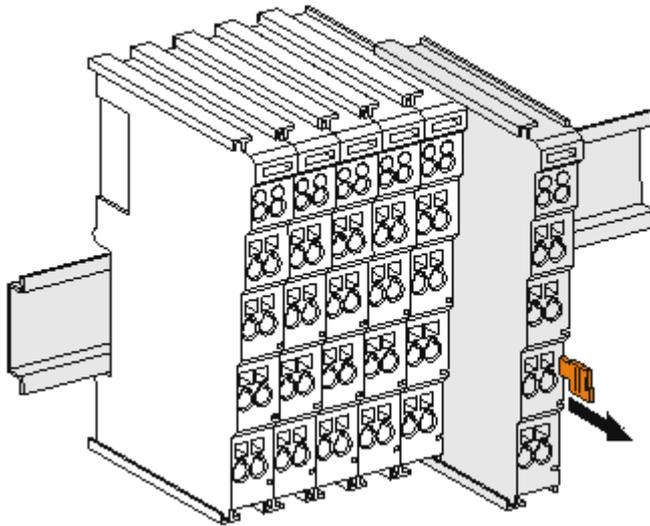


Fig. 17: Disassembling of terminal

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 points 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.

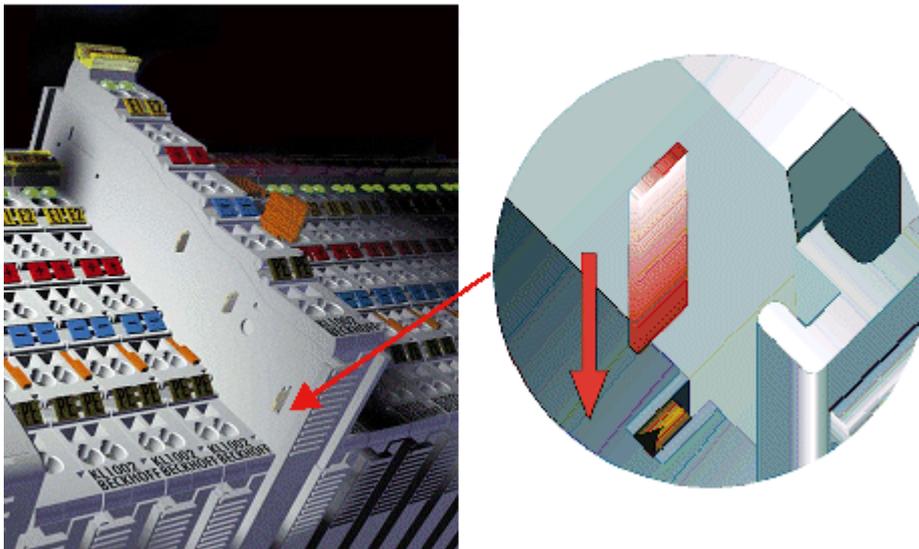


Fig. 18: Power contact on left side

### 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!

## 4.6 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.

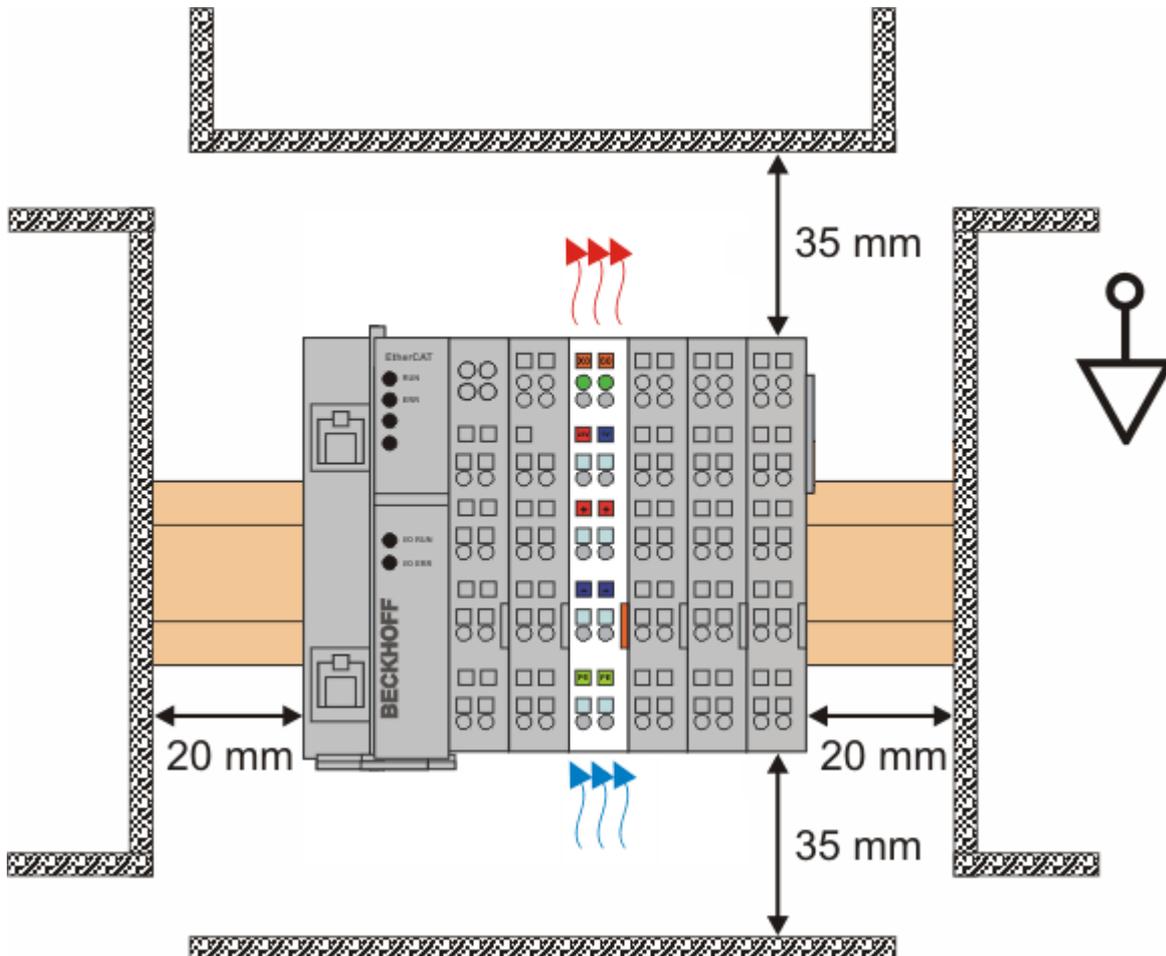


Fig. 19: Recommended distances for standard installation position

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.

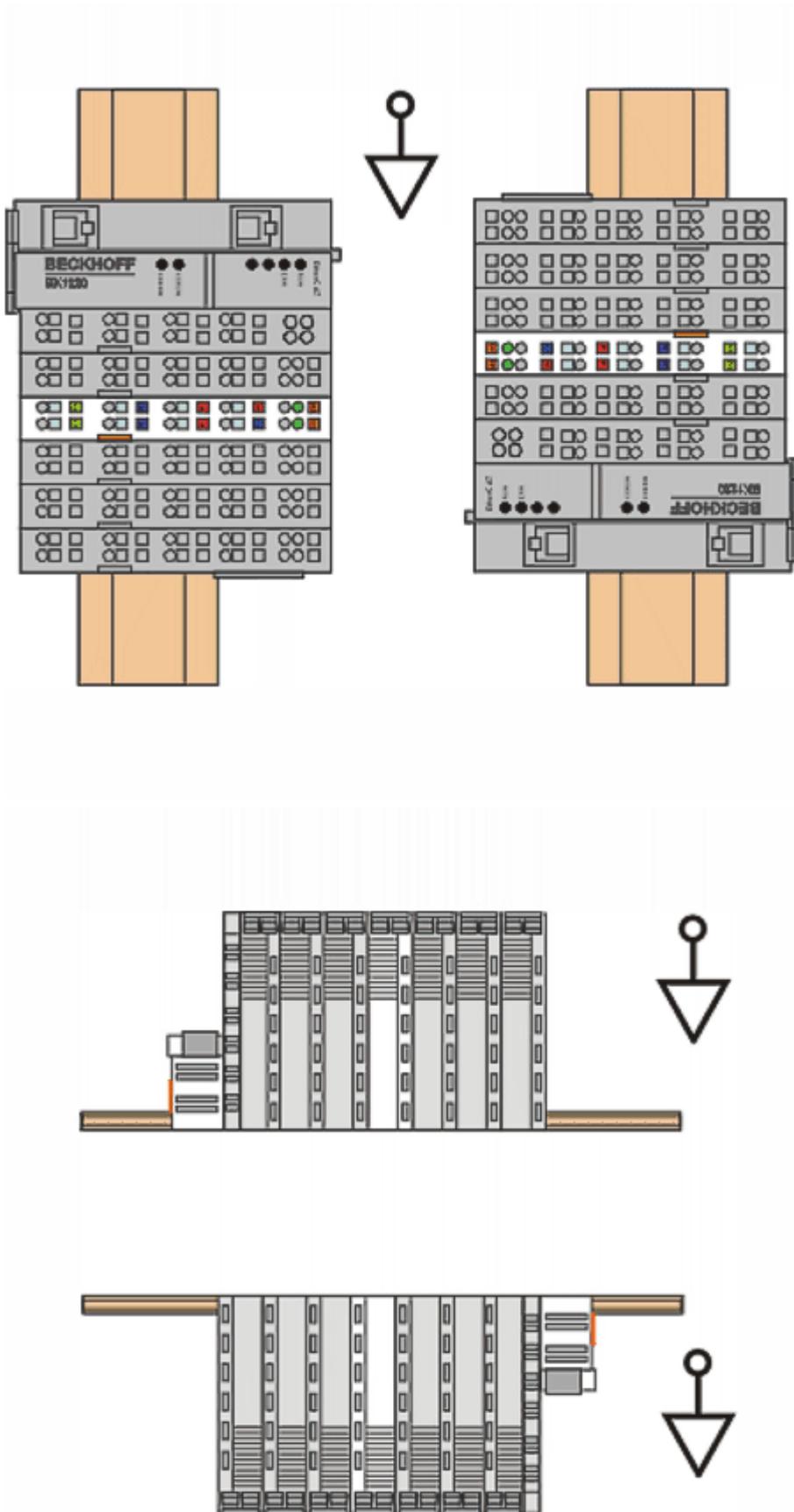


Fig. 20: Other installation positions

## 4.7 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)**

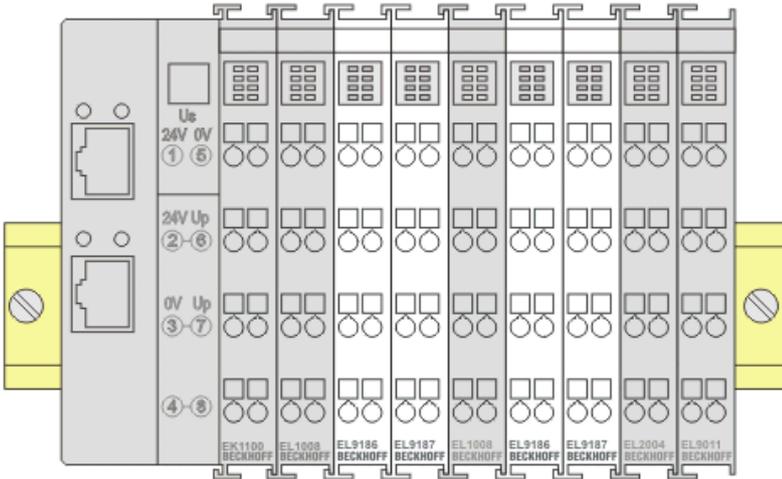


Fig. 21: Correct positioning

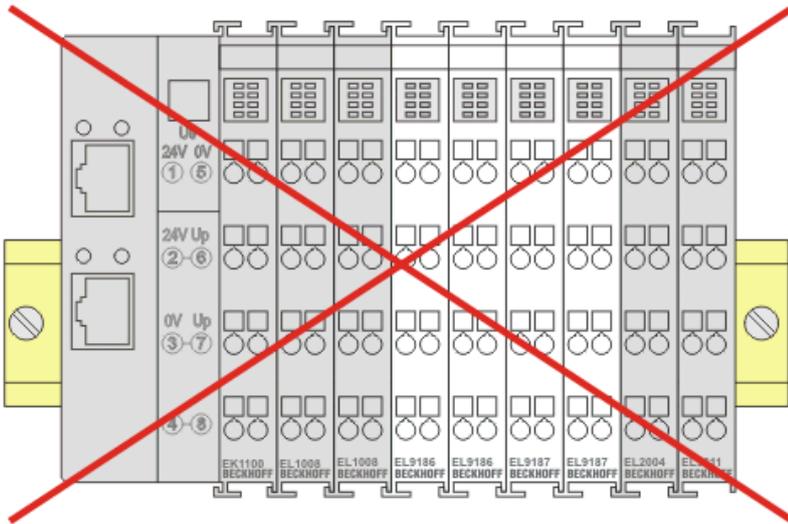


Fig. 22: Incorrect positioning

## 4.8 Installation instructions for enhanced mechanical load capacity

**⚠ WARNING**

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

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

**Additional checks**

The terminals have undergone the following additional tests:

Verification	Explanation
Vibration	10 frequency runs in 3 axes
	6 Hz < f < 60 Hz displacement 0.35 mm, constant amplitude
	60.1 Hz < f < 500 Hz acceleration 5 g, constant amplitude
Shocks	1000 shocks in each direction, in 3 axes
	25 g, 6 ms

**Additional installation instructions and notes**

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

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

## 4.9 Connection

### 4.9.1 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. 23: Standard wiring

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

#### Pluggable wiring (ESxxxx / KSxxxx)



Fig. 24: 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. 25: High Density Terminals

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

---

#### ● **Wiring HD Terminals**

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

---

### Ultrasonically compacted (ultrasonically welded) strands

---

#### ● **Ultrasonically compacted (ultrasonically welded) strands**

**i** Ultrasonically compacted (ultrasonically welded) strands can also be connected to the standard and high-density terminals. In this case, please note the tables concerning the wire-size width [► 47]!

---

## 4.9.2 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

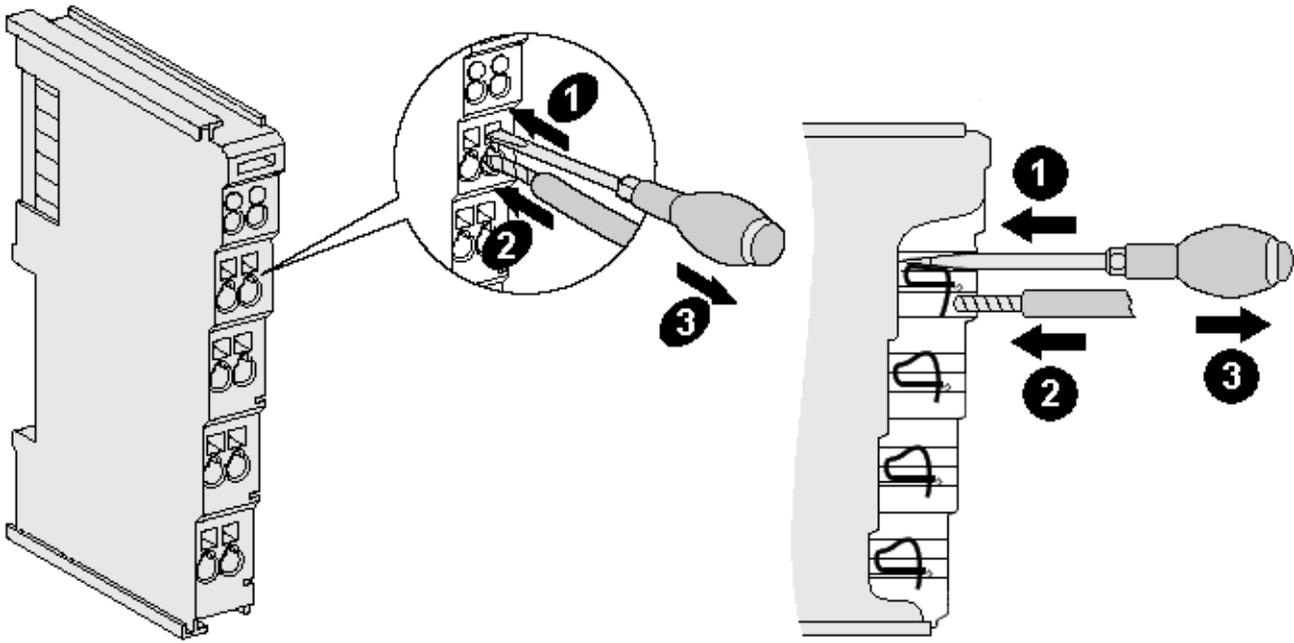


Fig. 26: Connecting a cable on a terminal point

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

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

See the following table for the suitable wire size width:

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width (single core wires)	0.08 ... 2.5 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](#) [[▶ 45](#)]) with 16/32 terminal points

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

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 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 compacted [ultrasonically welded] strands)	only 1.5 mm <sup>2</sup> (see <a href="#">notice</a> [ <a href="#">▶ 45</a> ])
Wire stripping length	8 ... 9 mm

## 4.9.3 Shielding

### ● Shielding

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

## 4.10 Note - power supply

### WARNING

#### **Power supply from SELV / PELV power supply unit!**

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

Notes:

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

## 4.11 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:

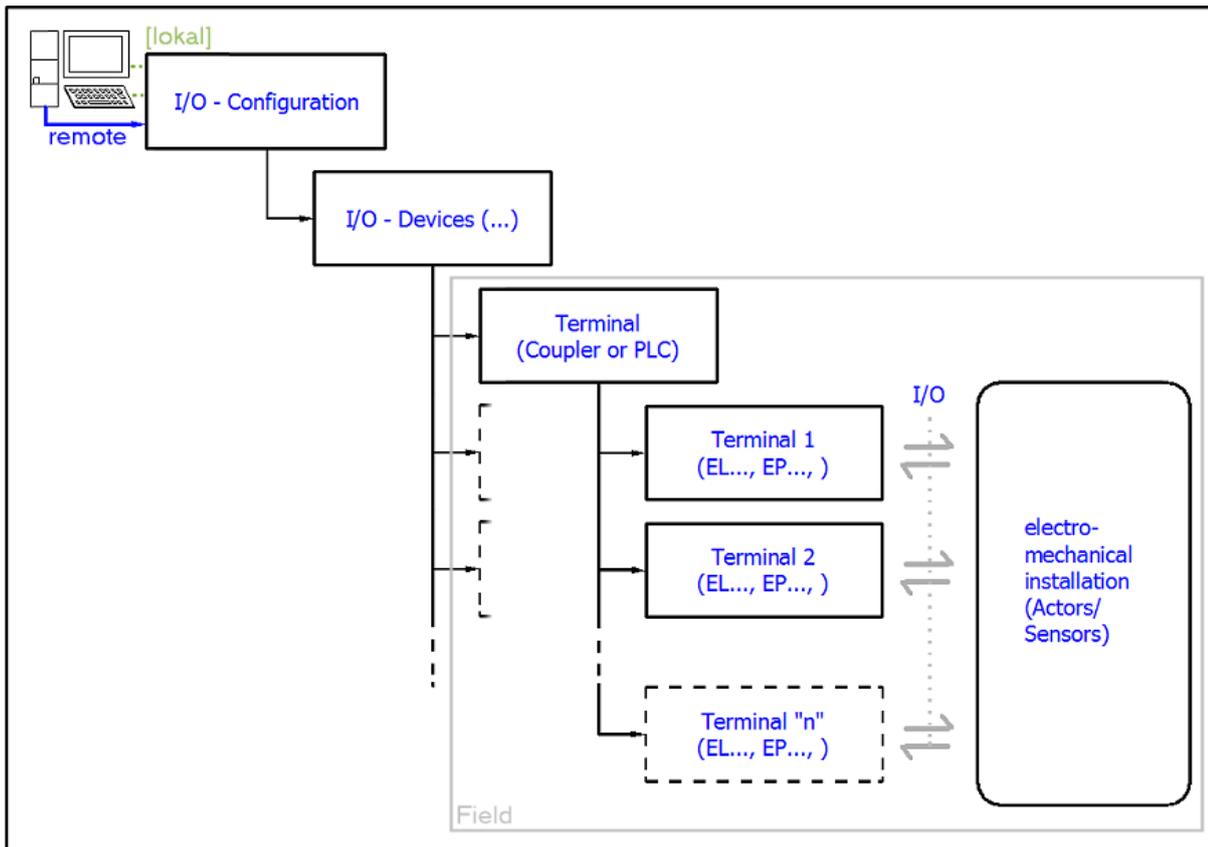


Fig. 27: Relationship between user side (commissioning) and installation

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:

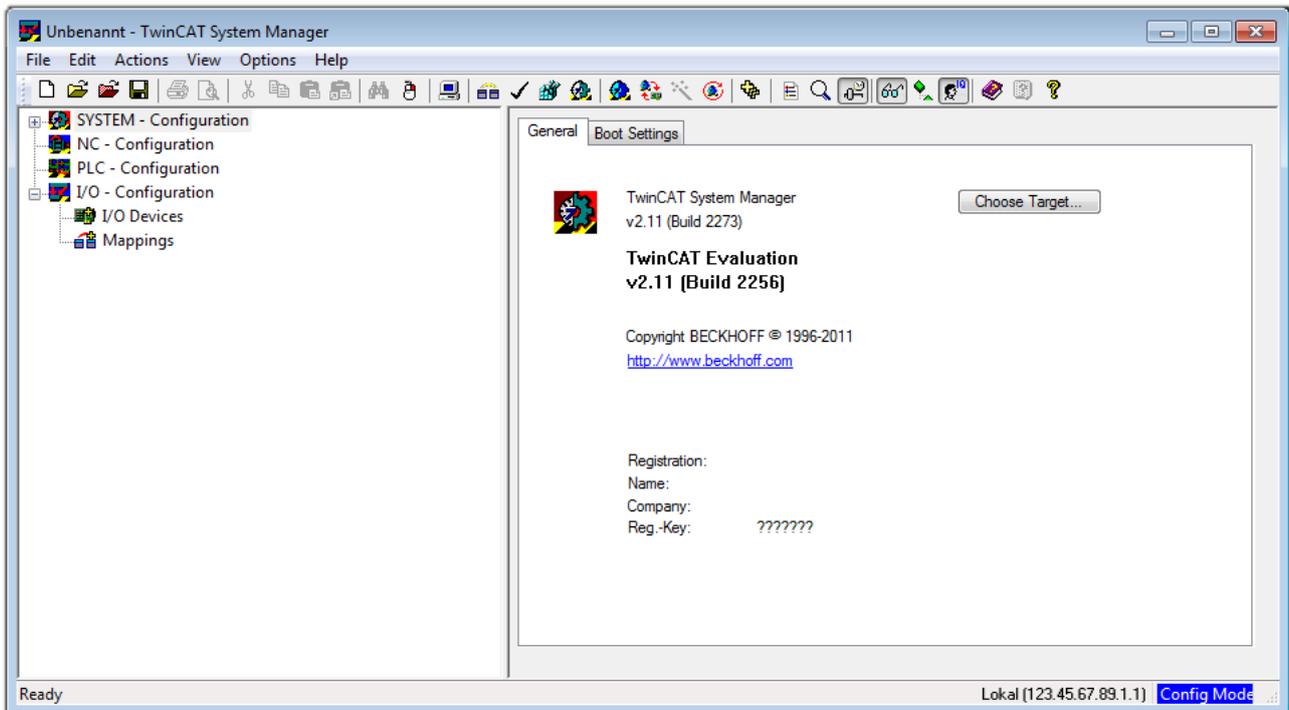


Fig. 29: Initial TwinCAT 2 user interface

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 \[▶ 55\]](#)”.

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:

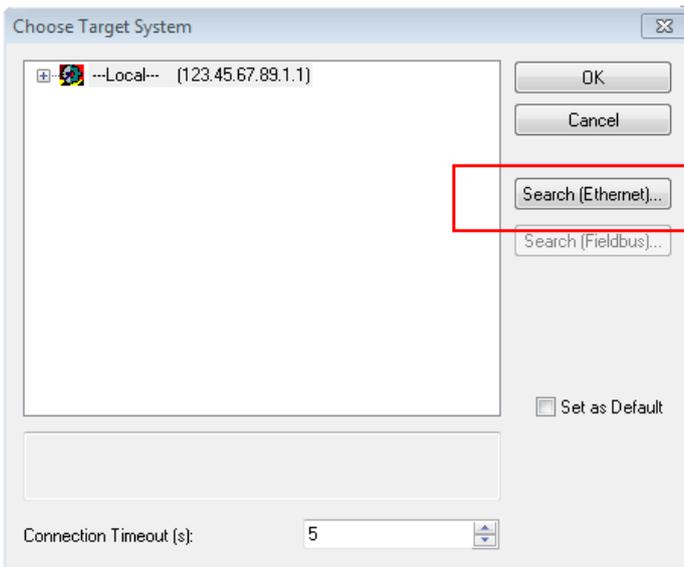


Fig. 30: Selection of the target system

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

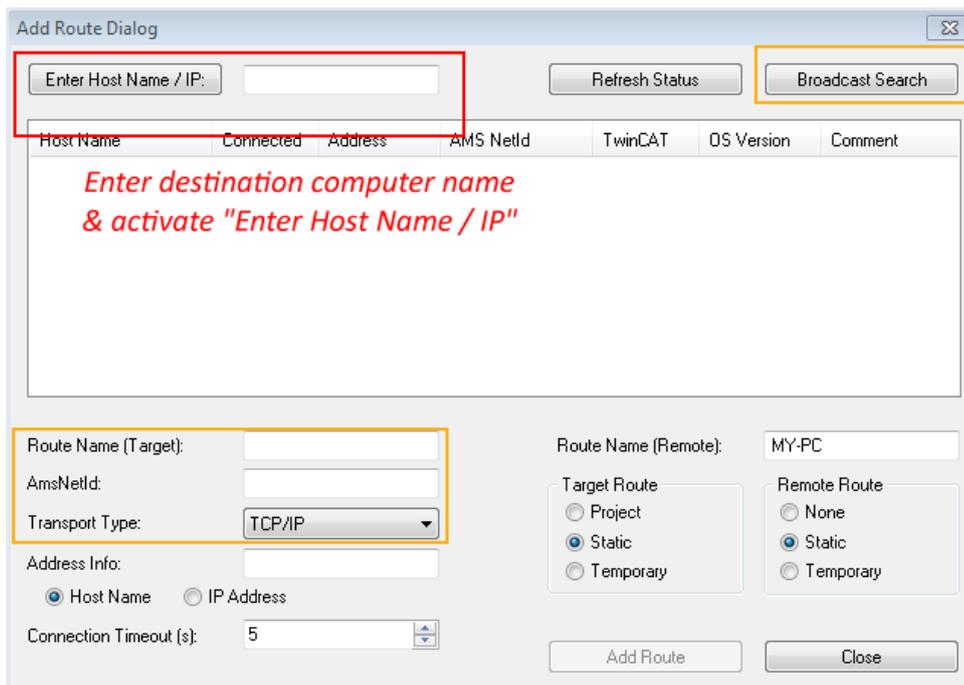
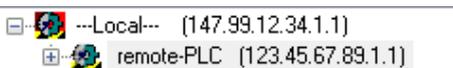


Fig. 31: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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).

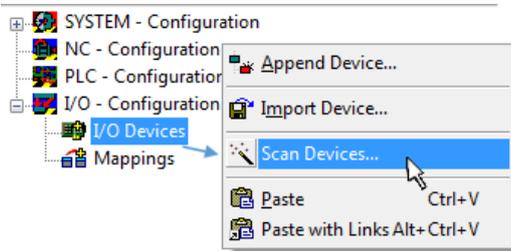


Fig. 32: Select “Scan Devices...”

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

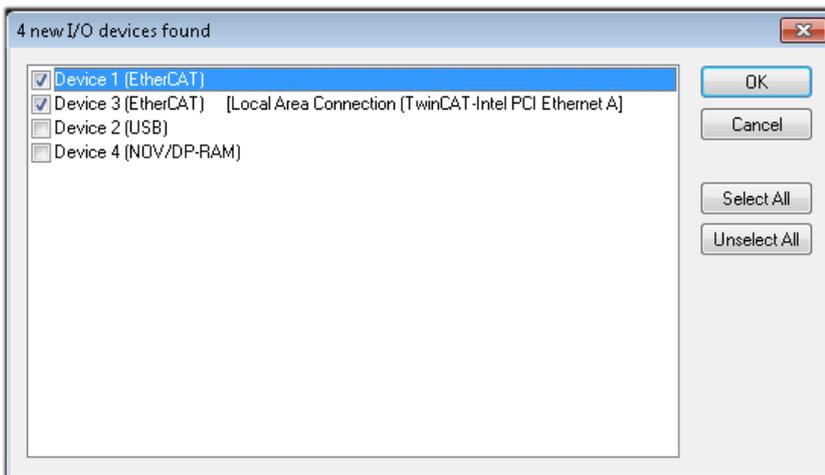


Fig. 33: Automatic detection of I/O devices: selection of the devices to be integrated

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 \[▶ 51\]](#) described at the beginning of this section, the result is as follows:

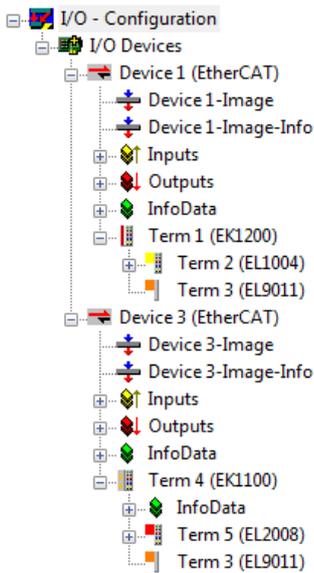


Fig. 34: Mapping of the configuration in the TwinCAT 2 System Manager

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:

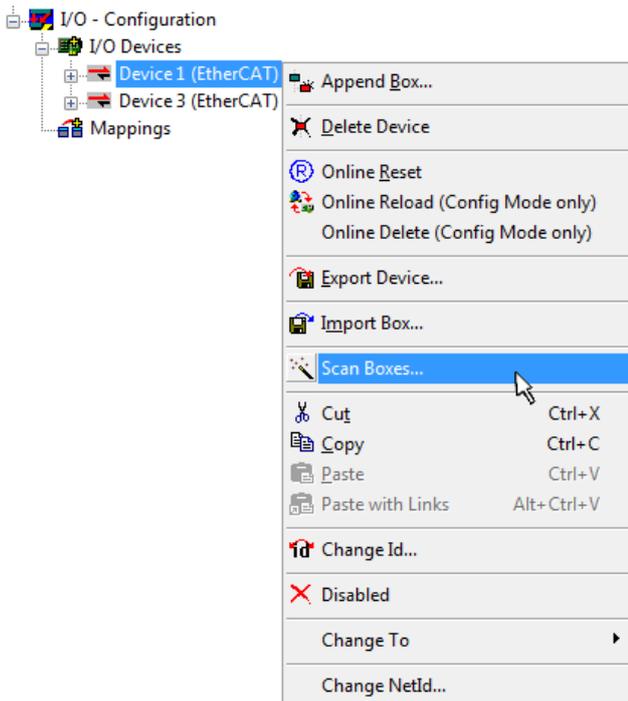


Fig. 35: Reading of individual terminals connected to a device

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:

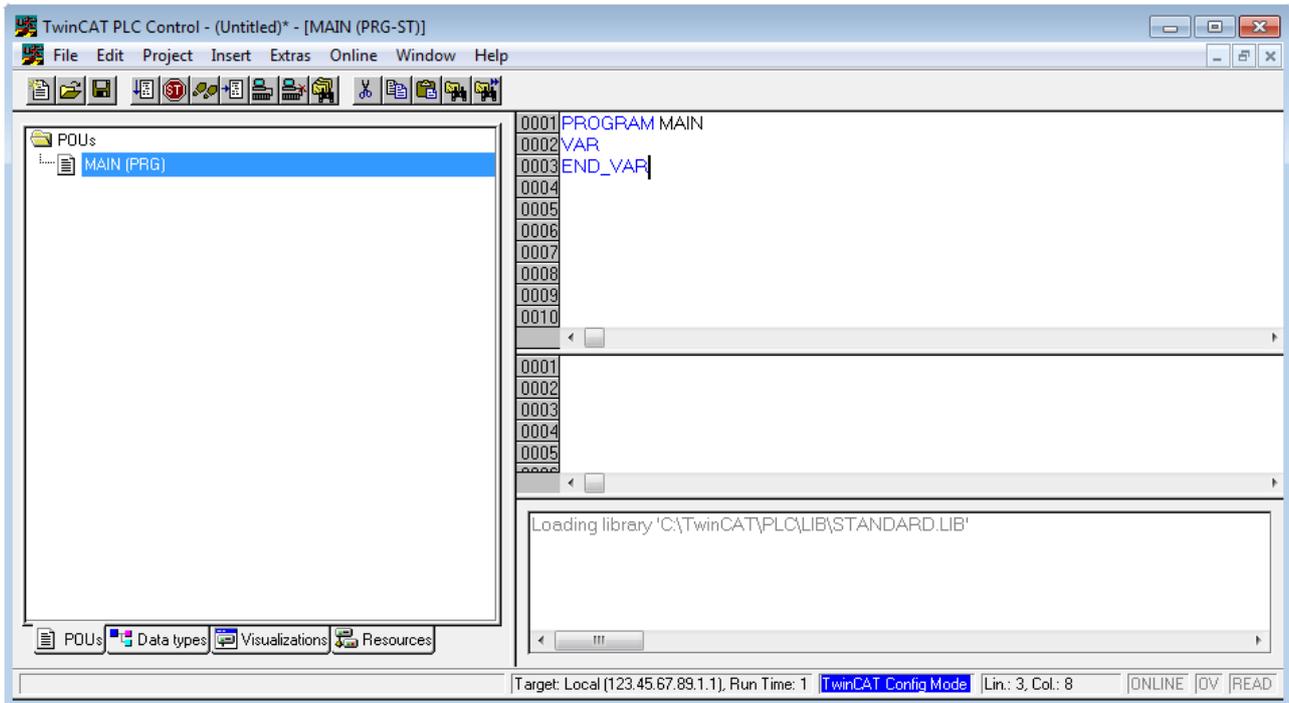


Fig. 36: TwinCAT PLC Control after startup

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

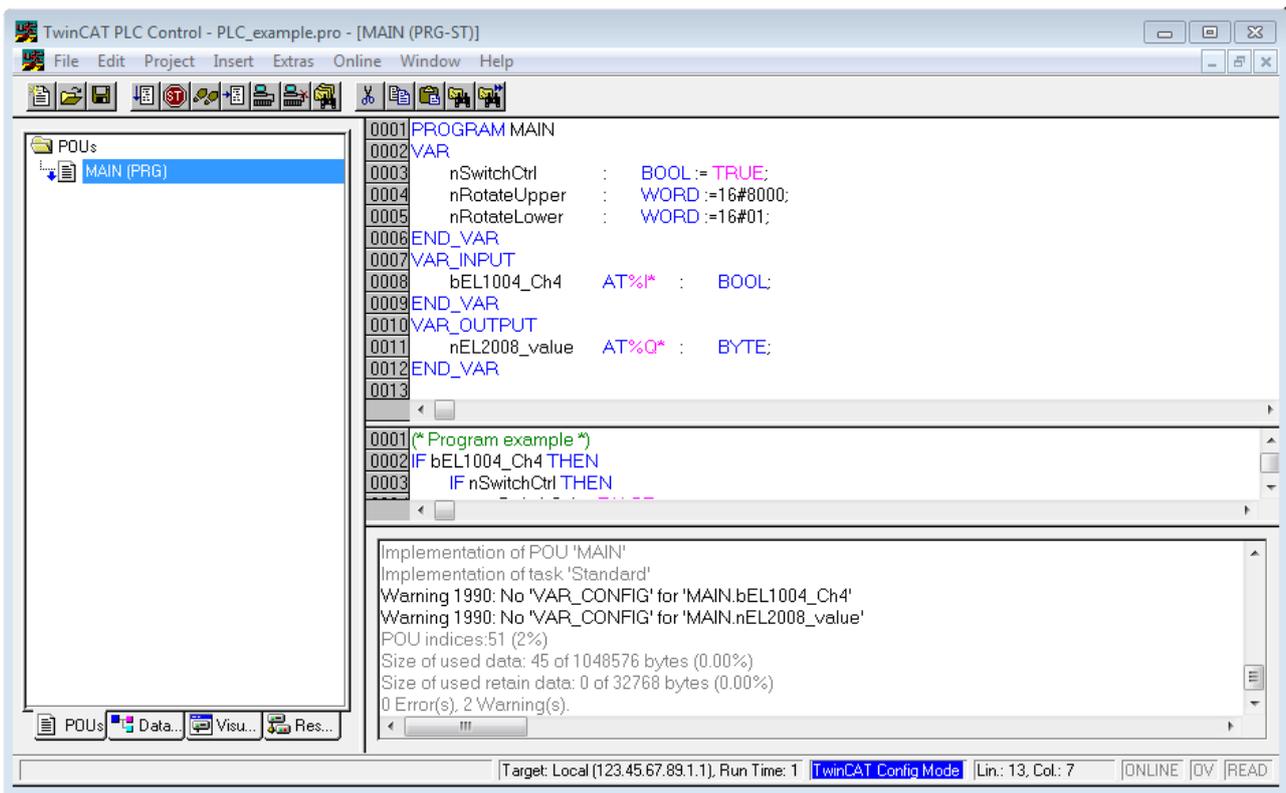


Fig. 37: Example program with variables after a compile process (without variable integration)

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...”:

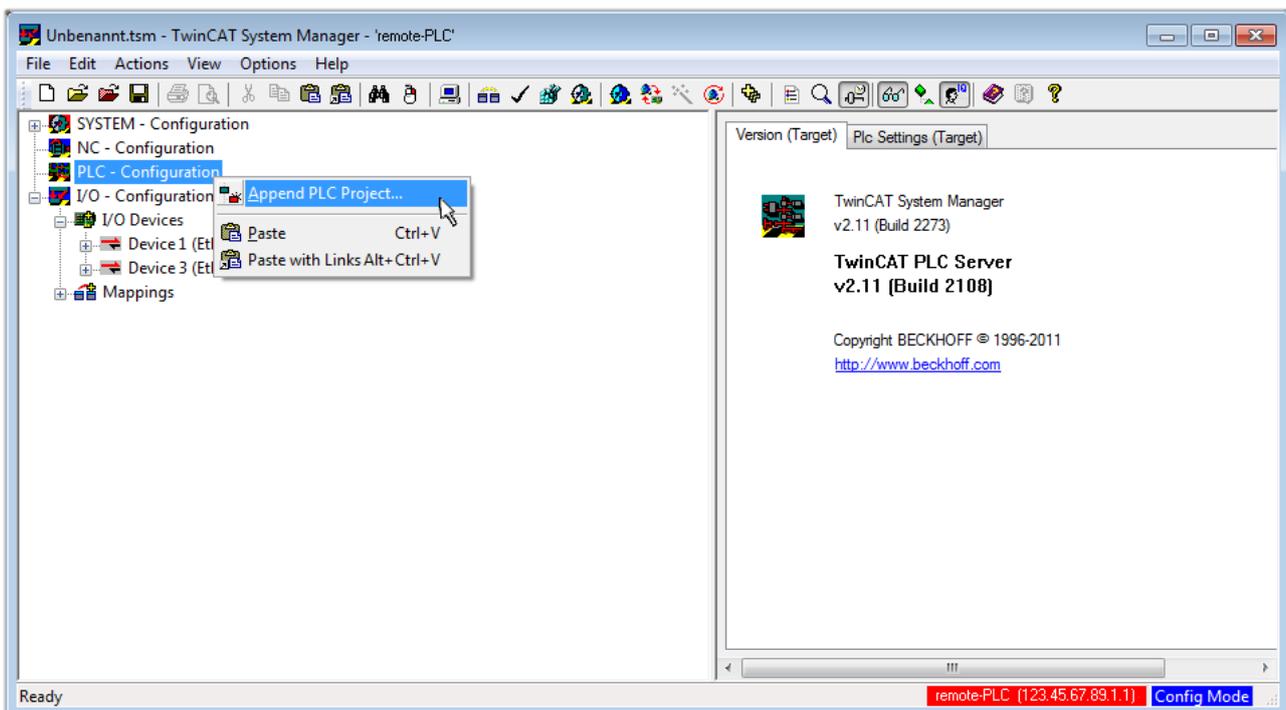


Fig. 38: Appending the TwinCAT PLC Control project

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:

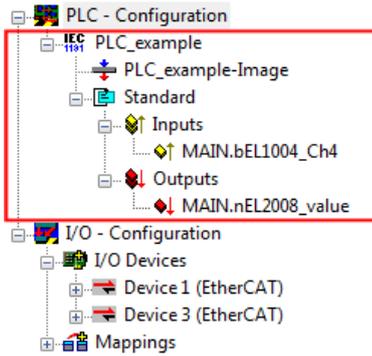


Fig. 39: PLC project integrated in the PLC configuration of the System Manager

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”:

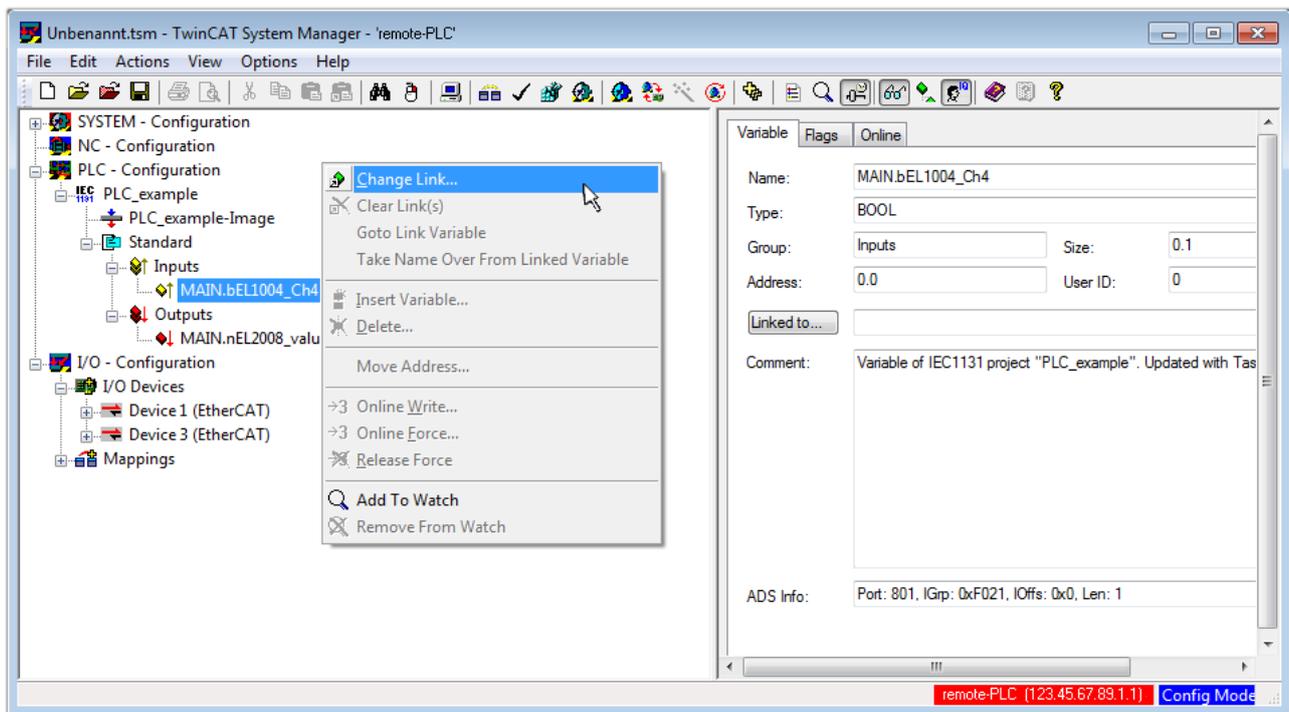


Fig. 40: Creating the links between PLC variables and process objects

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

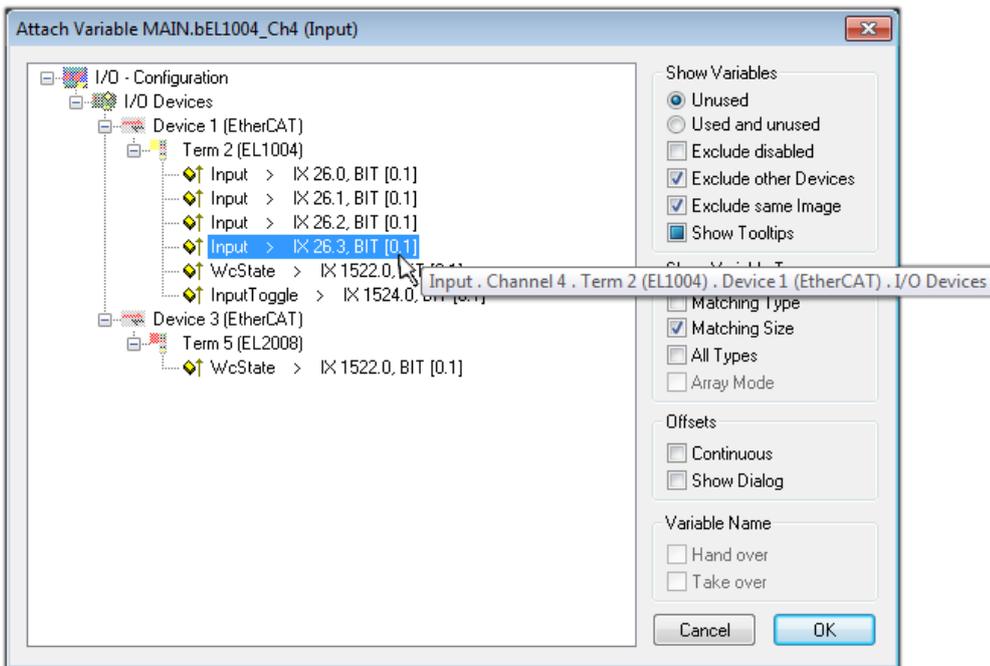


Fig. 41: Selecting BOOL-type PDO

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:

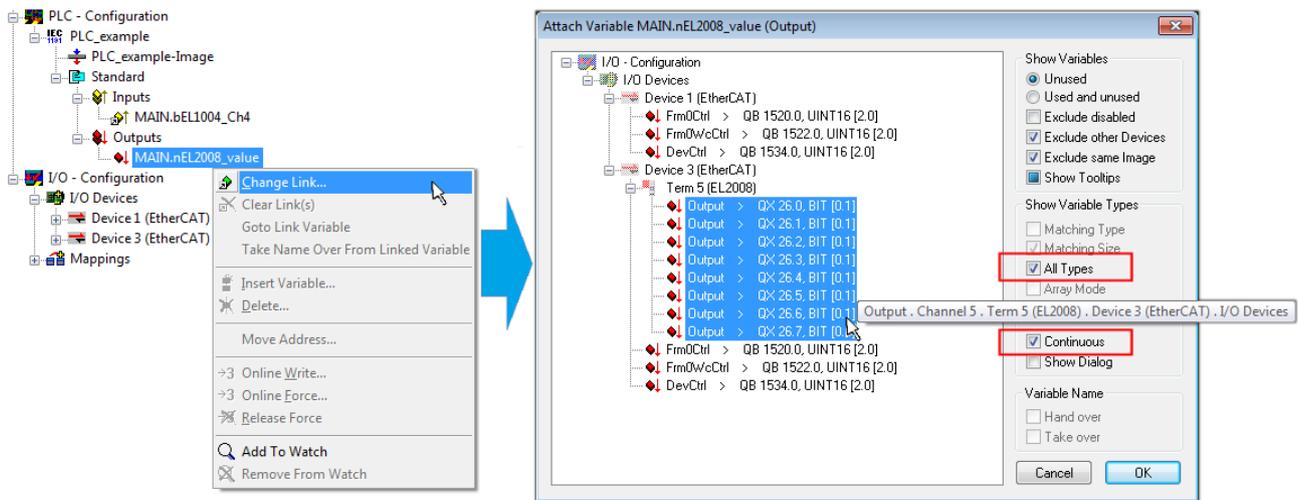


Fig. 42: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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:

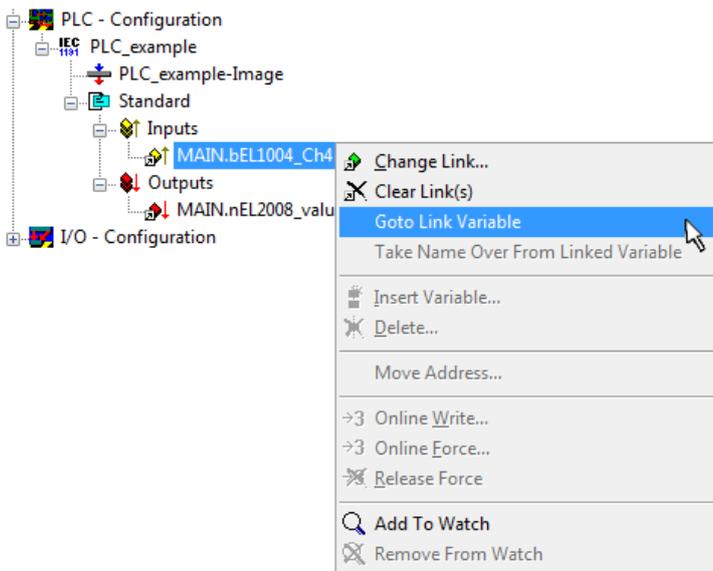


Fig. 43: Application of a “Goto Link Variable”, using “MAIN.bEL1004\_Ch4” as an example

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...”:

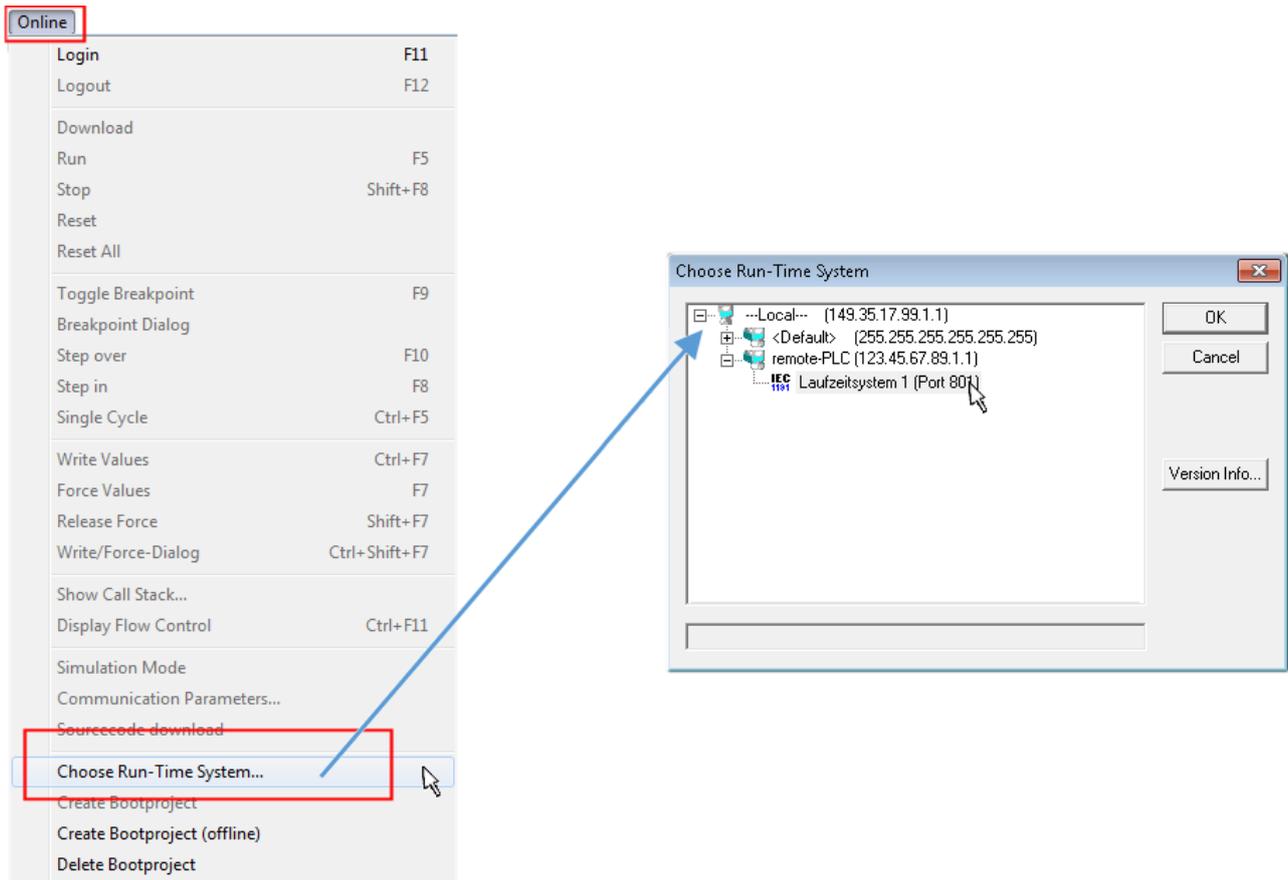
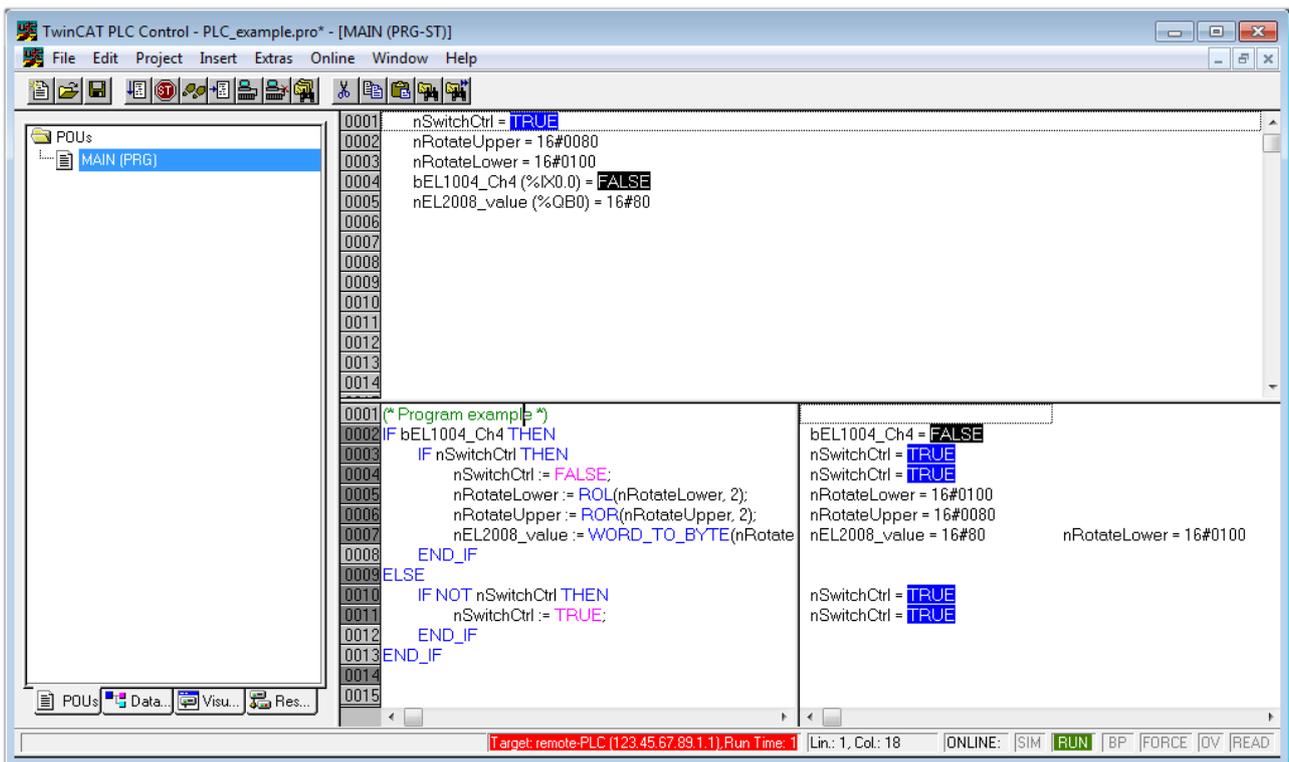


Fig. 44: Choose target system (remote)

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:



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. 46: 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):

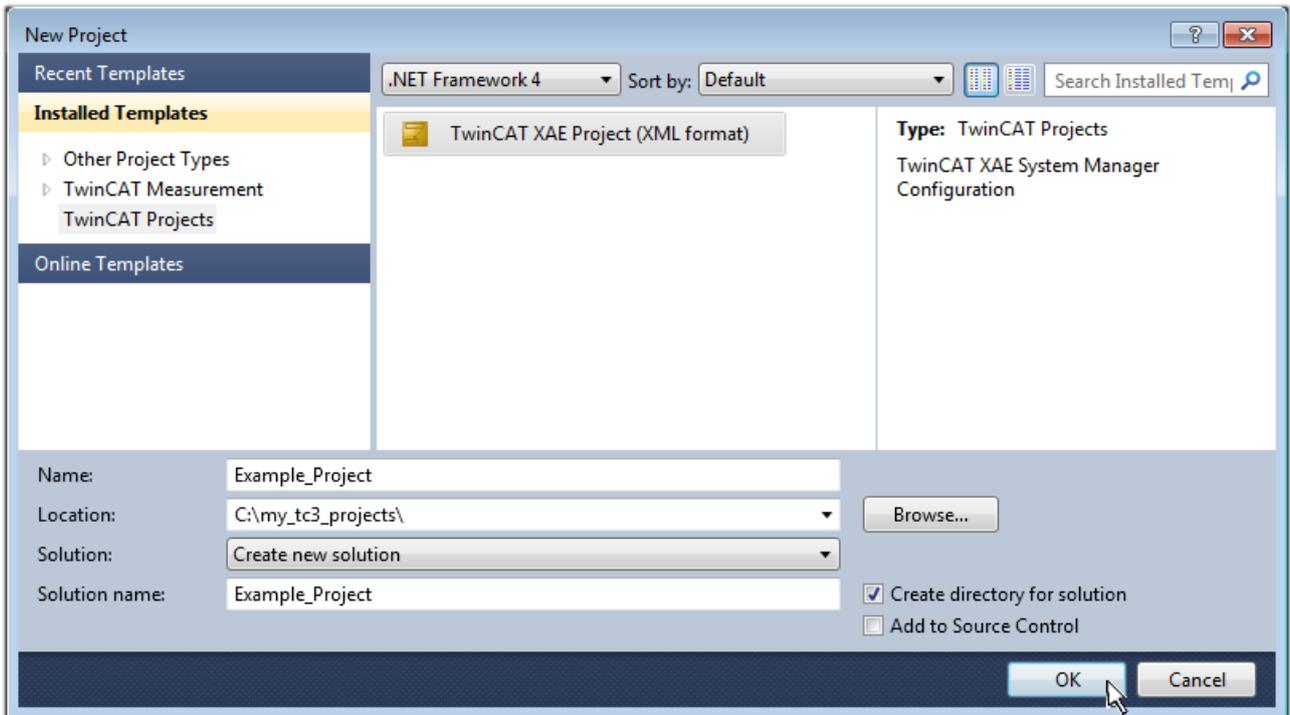


Fig. 47: Create new TwinCAT 3 project

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

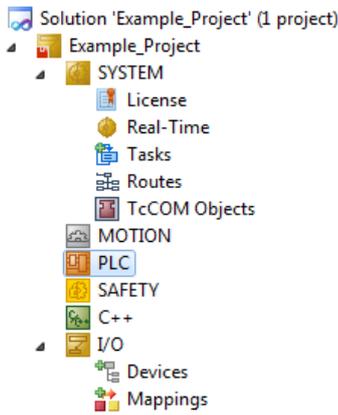
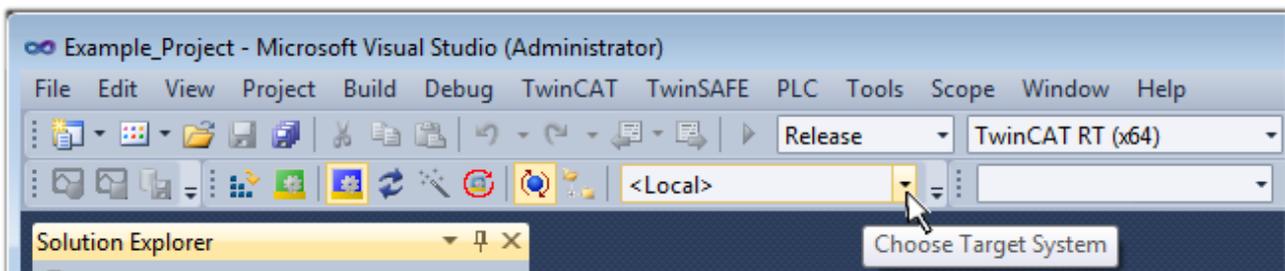


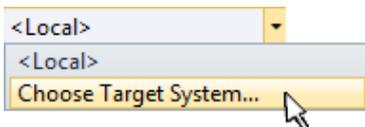
Fig. 48: New TwinCAT 3 project in the project folder explorer

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 [▶ 66]”.

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:

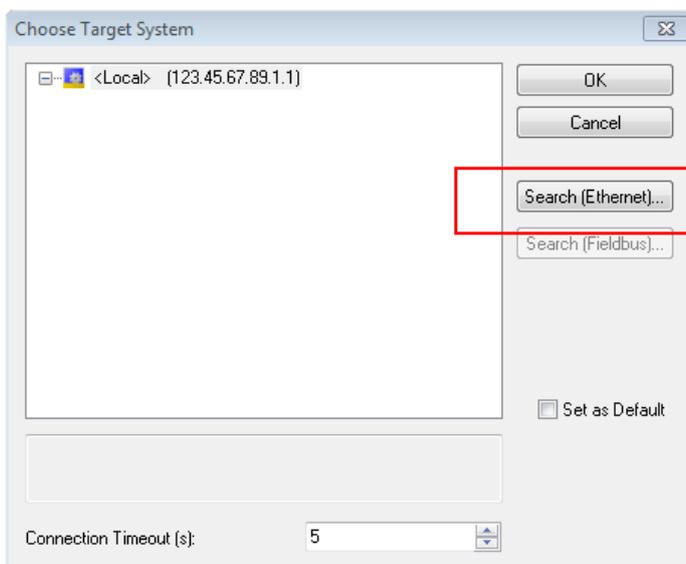


Fig. 49: Selection dialog: Choose the target system

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

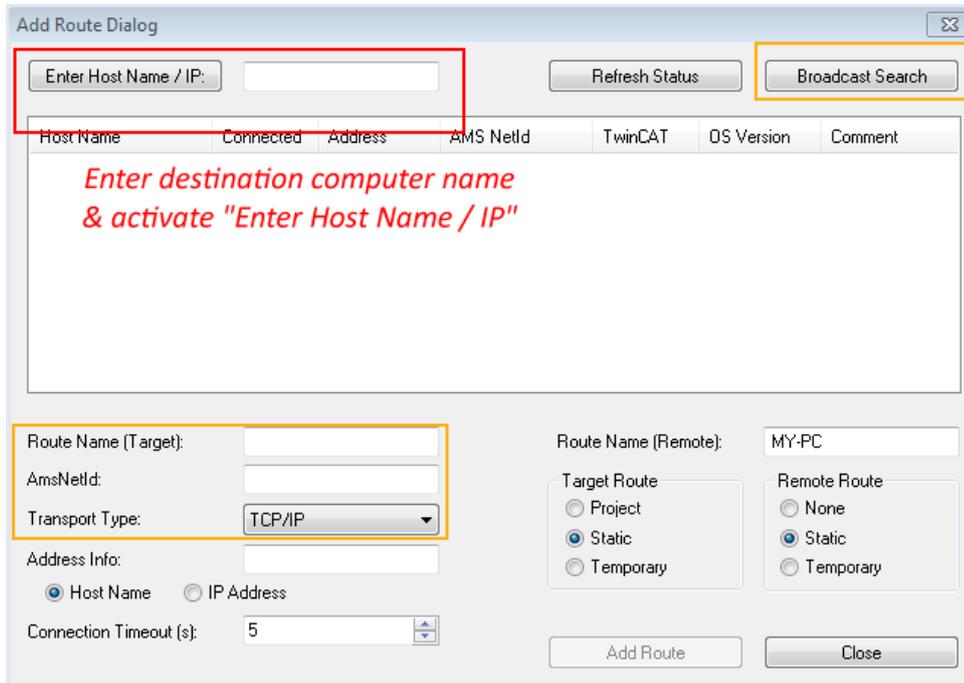
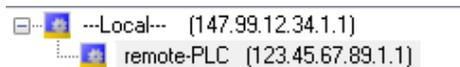


Fig. 50: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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)”.

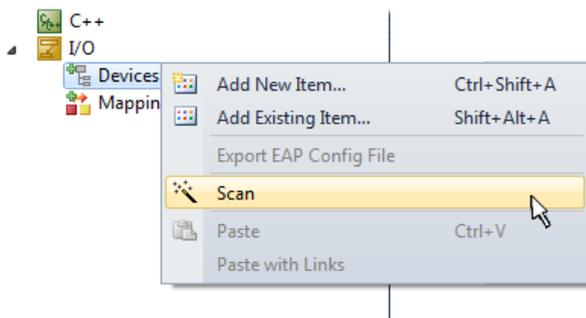


Fig. 51: Select “Scan”

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

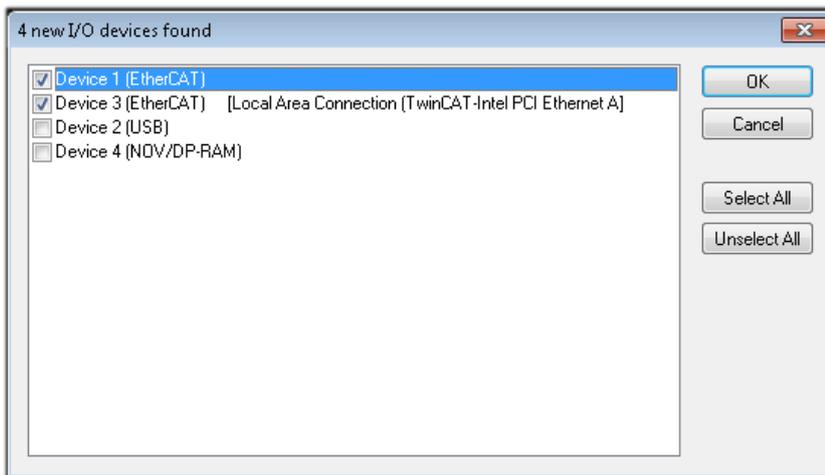


Fig. 52: Automatic detection of I/O devices: selection of the devices to be integrated

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 \[▶ 51\]](#) described at the beginning of this section, the result is as follows:

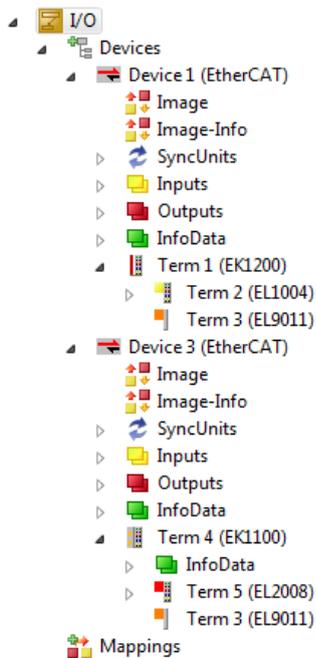


Fig. 53: Mapping of the configuration in VS shell of the TwinCAT 3 environment

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:

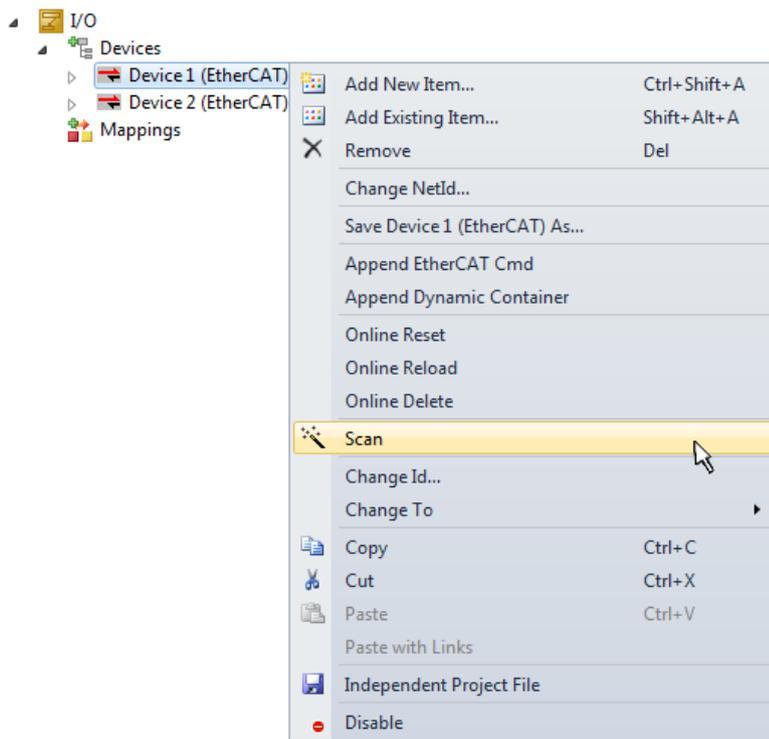


Fig. 54: Reading of individual terminals connected to a device

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....":

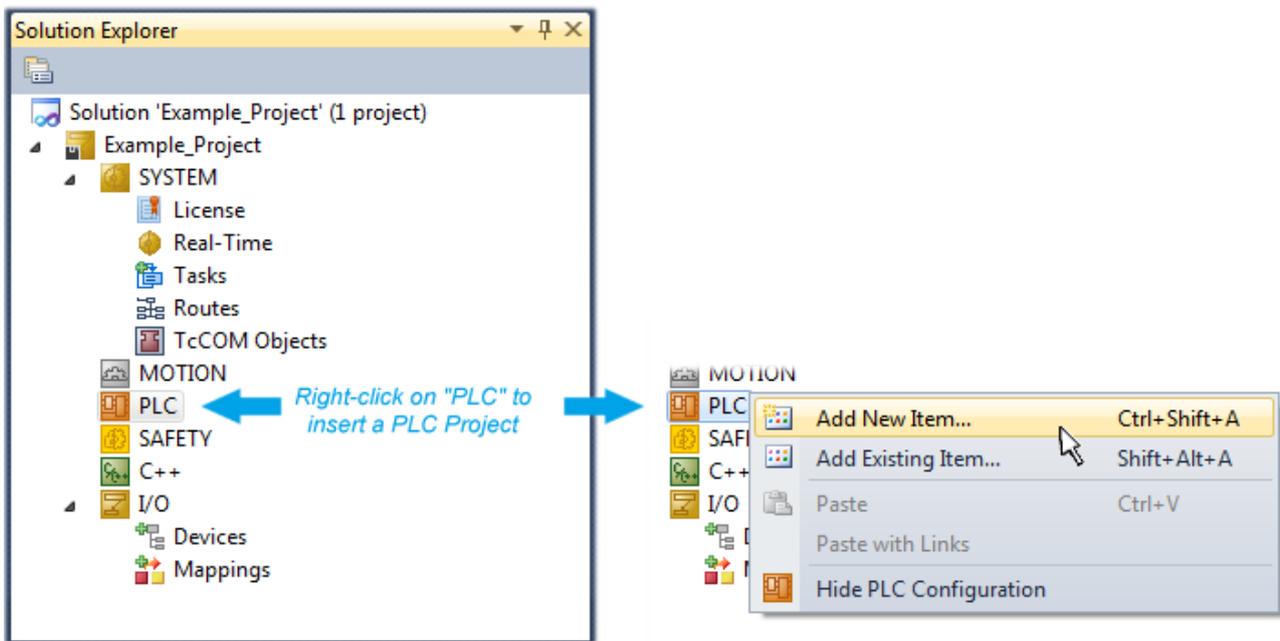


Fig. 55: Adding the programming environment in “PLC”

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

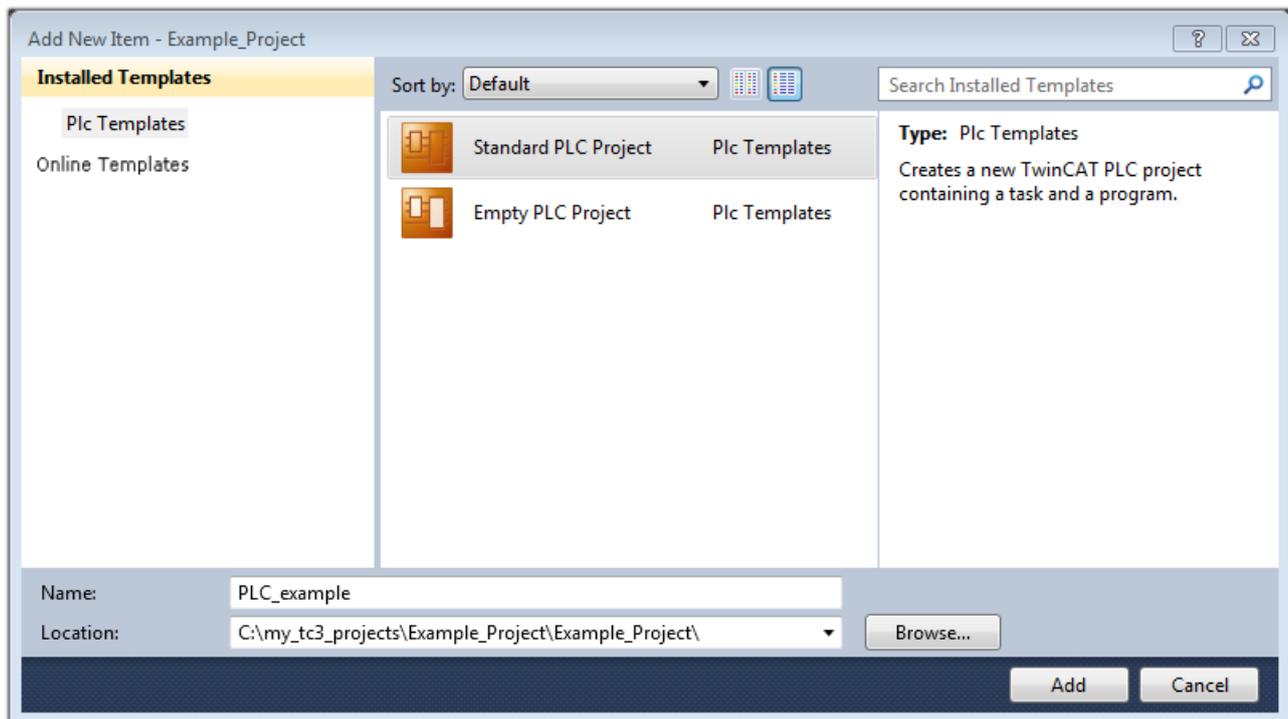


Fig. 56: Specifying the name and directory for the PLC programming environment

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:

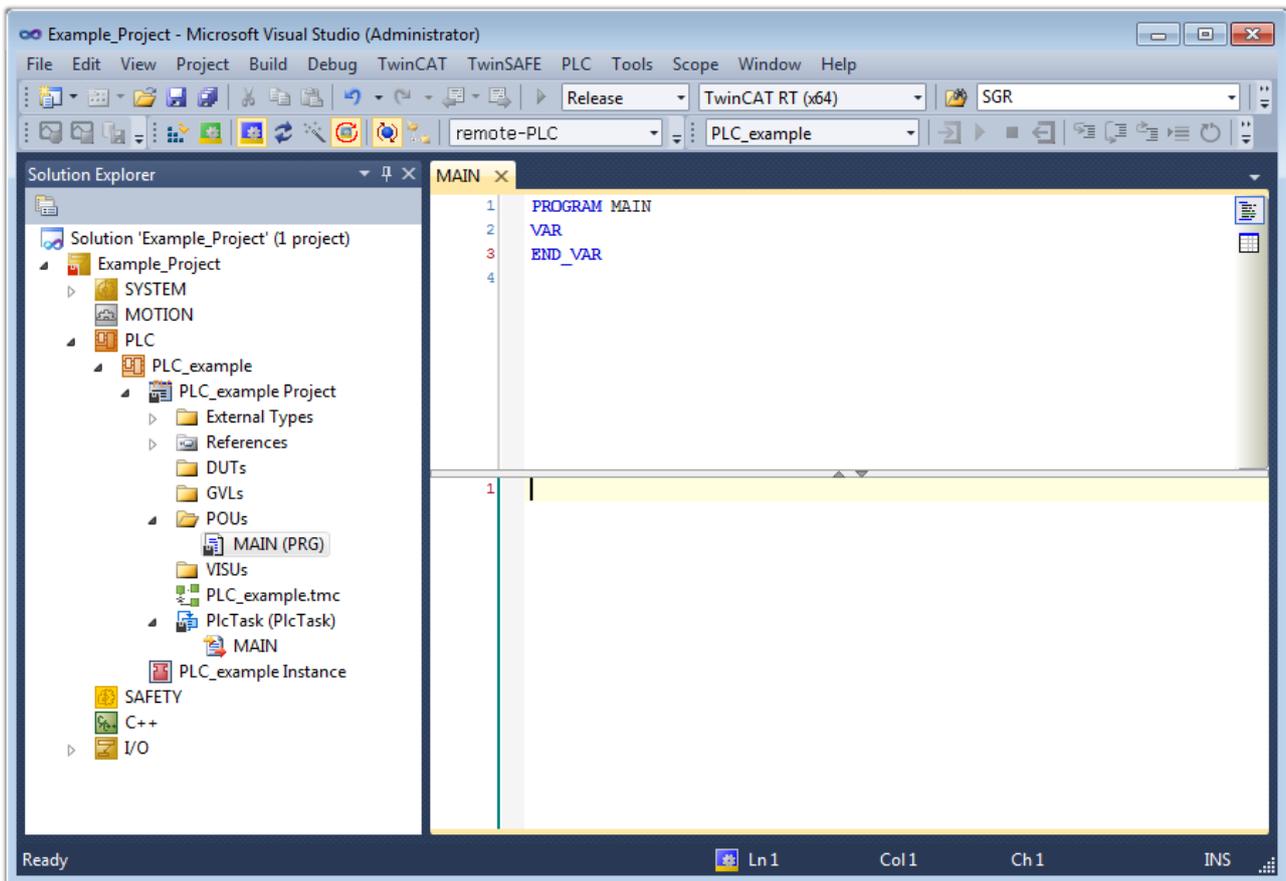


Fig. 57: Initial “Main” program for the standard PLC project

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

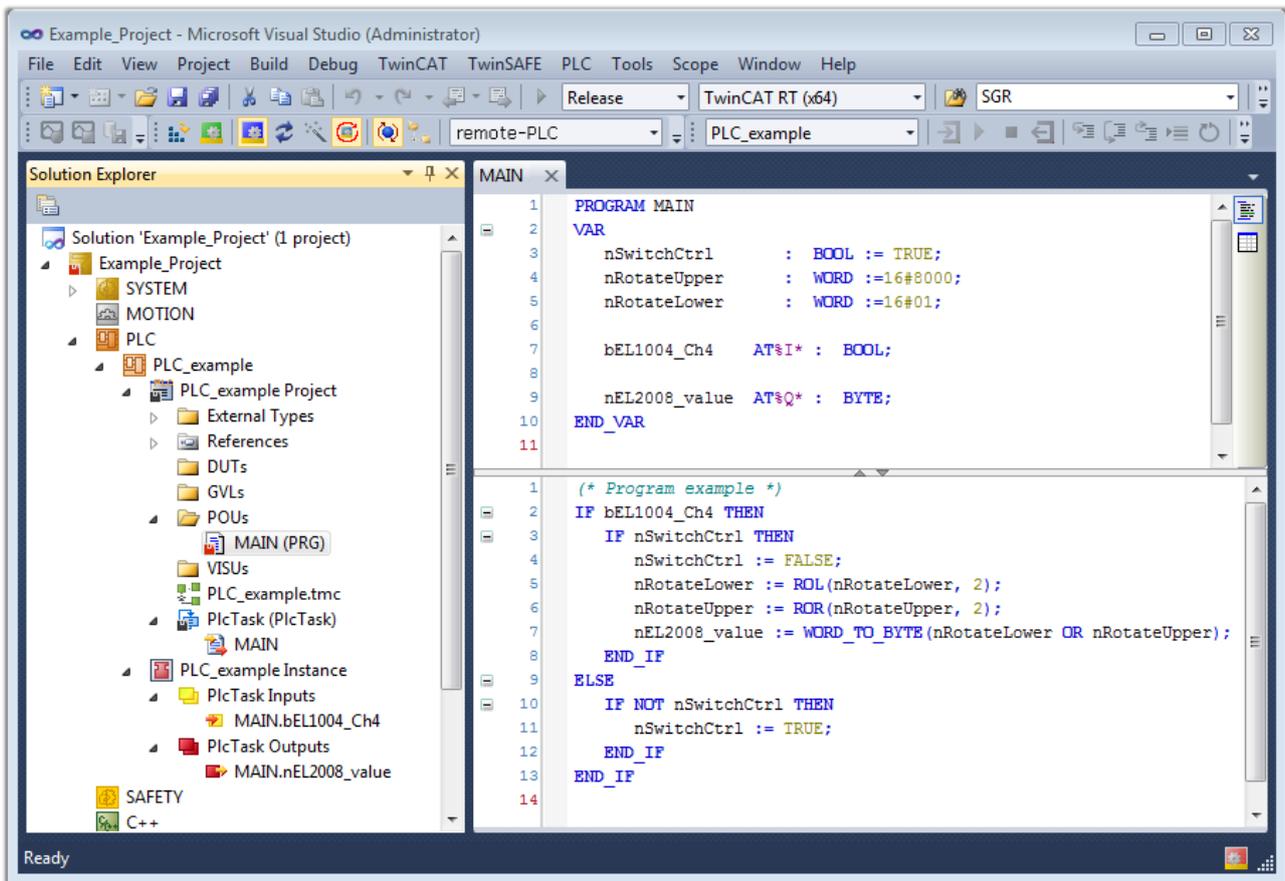


Fig. 58: Example program with variables after a compile process (without variable integration)

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

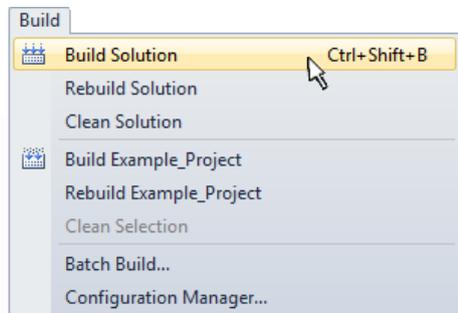
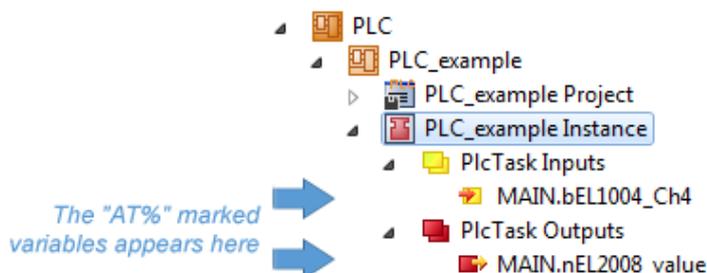


Fig. 59: Start program compilation

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:

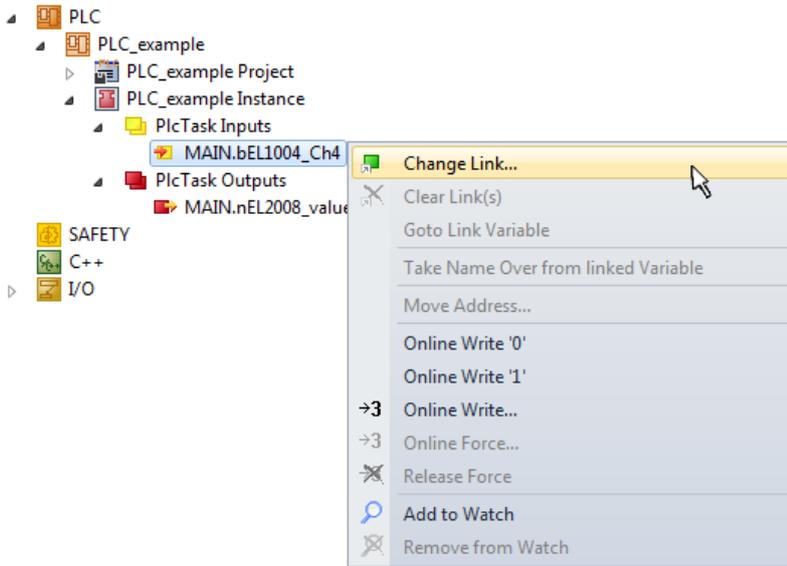


Fig. 60: Creating the links between PLC variables and process objects

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

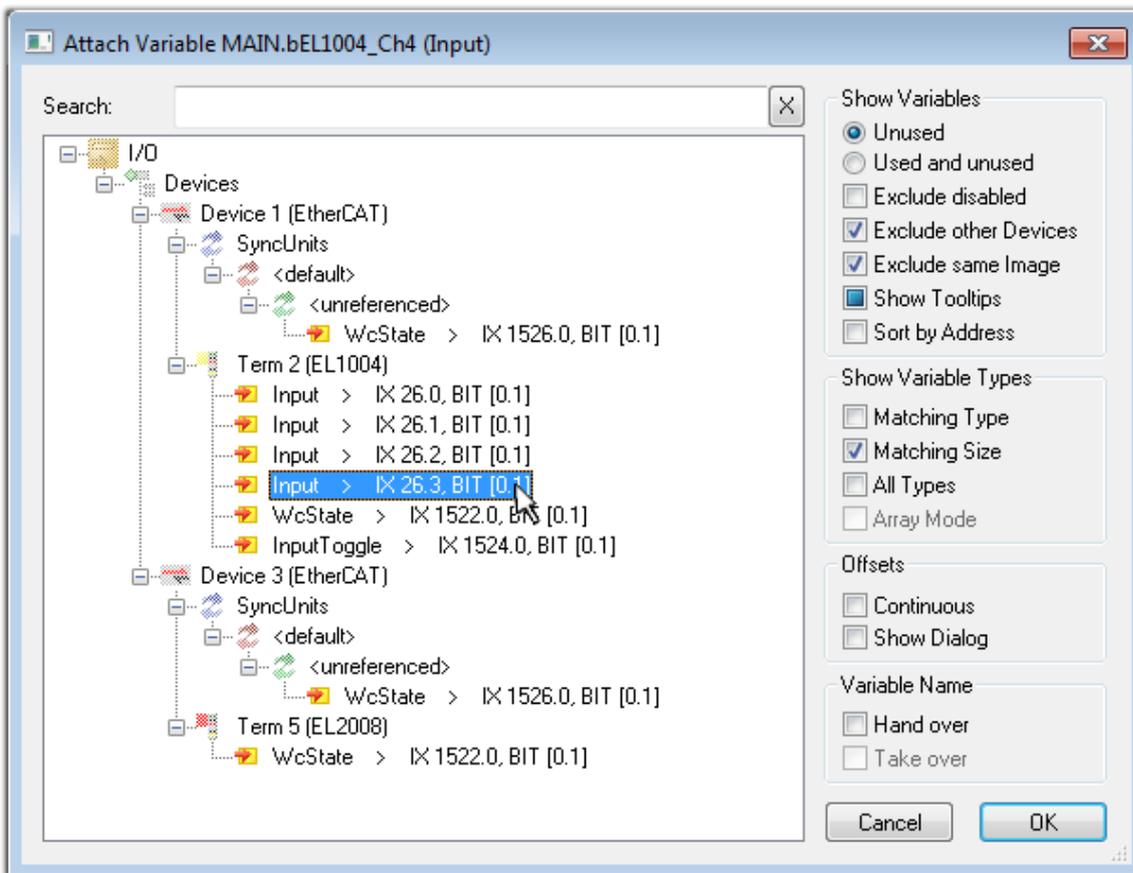


Fig. 61: Selecting BOOL-type PDO

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:

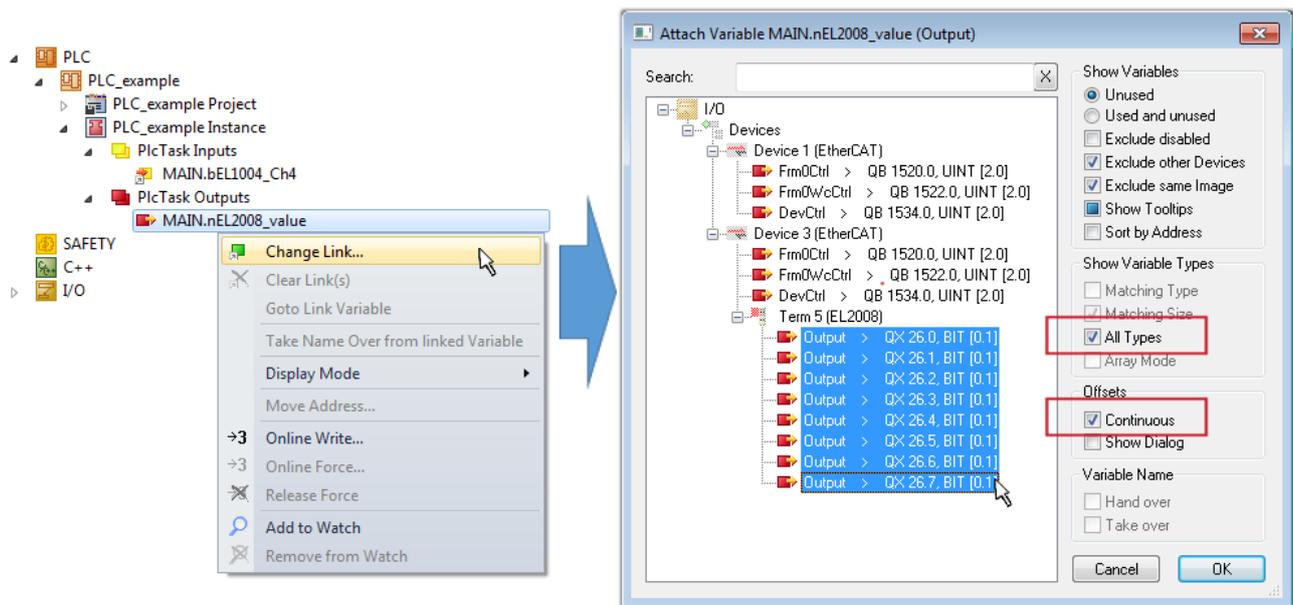


Fig. 62: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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:

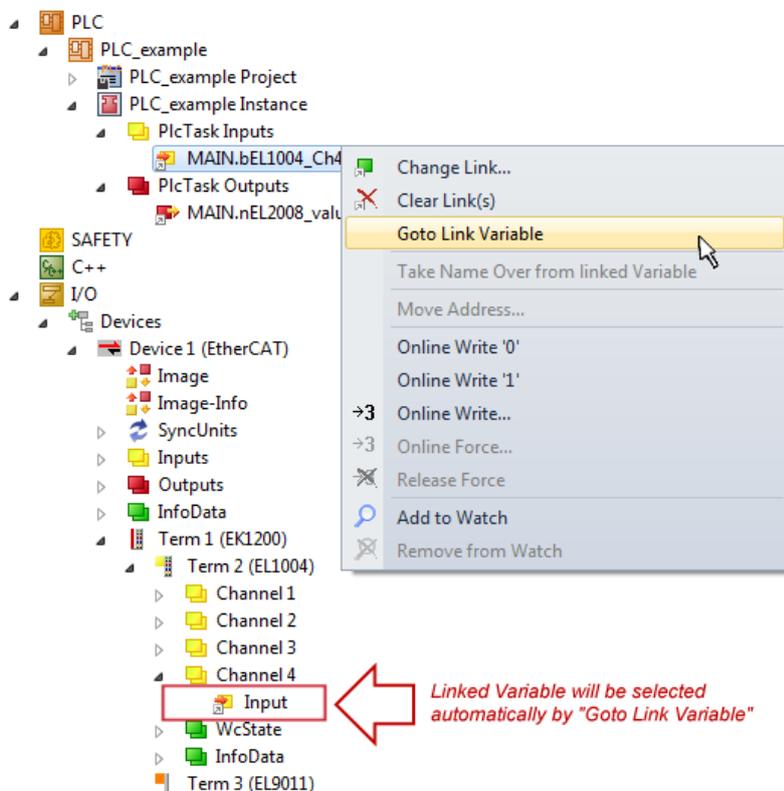


Fig. 63: Application of a “Goto Link Variable”, using “MAIN.bEL1004\_Ch4” as an example

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.

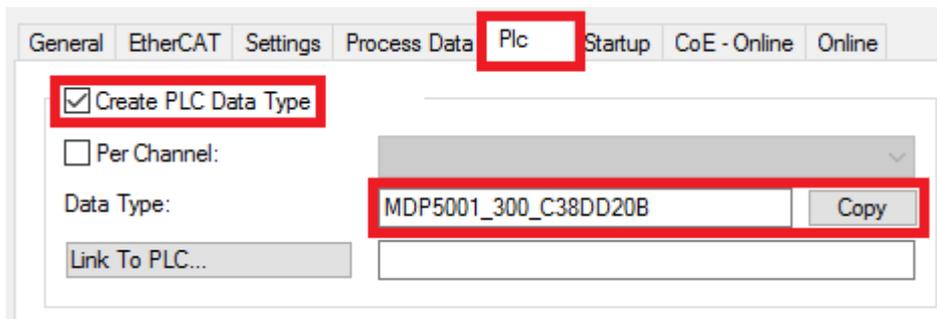


Fig. 64: Creating a PLC data type

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

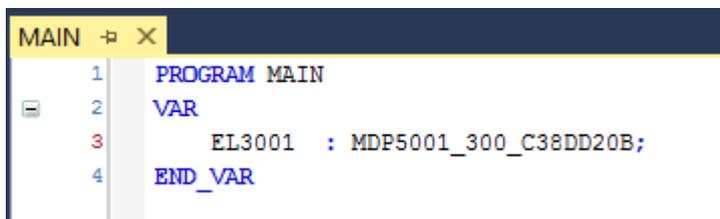


Fig. 65: Instance\_of\_struct

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.

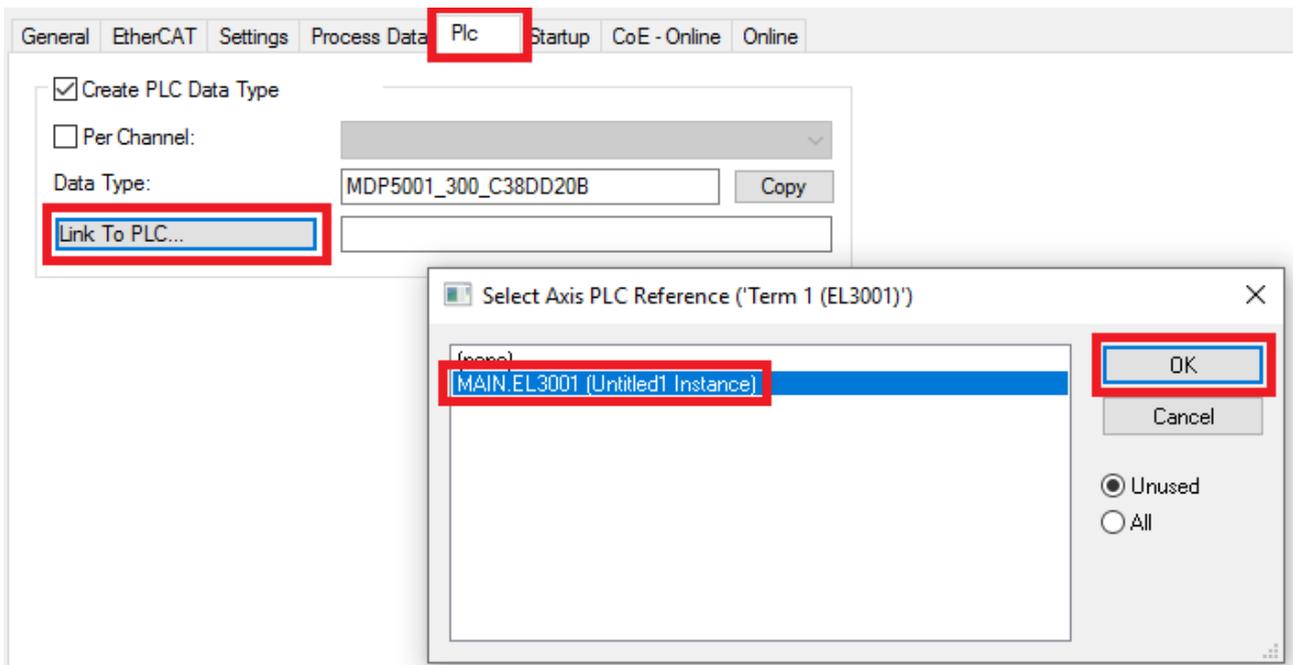


Fig. 66: Linking the structure

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

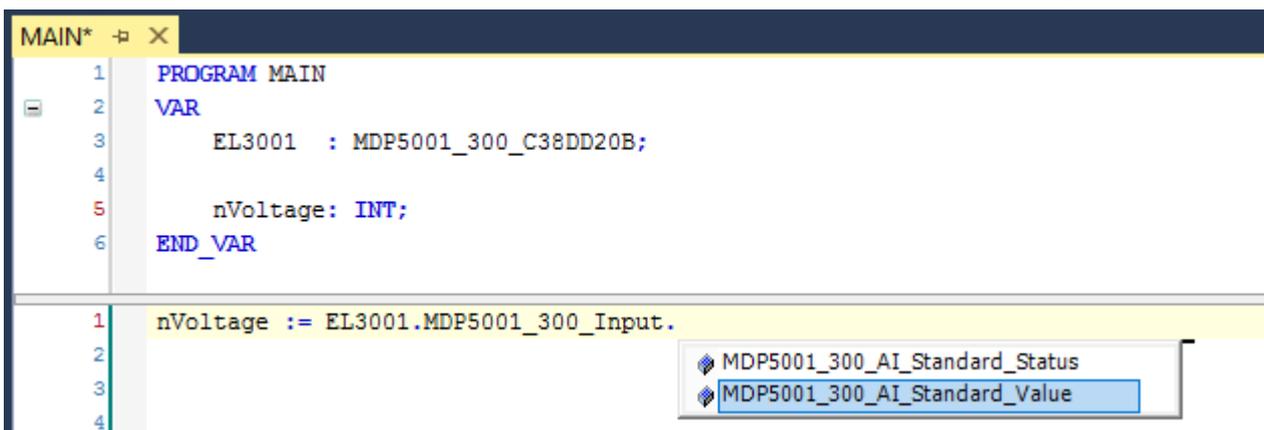
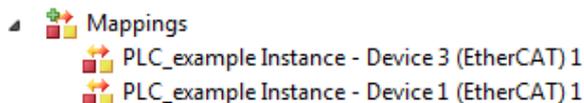


Fig. 67: Reading a variable from the structure of the process data

**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:

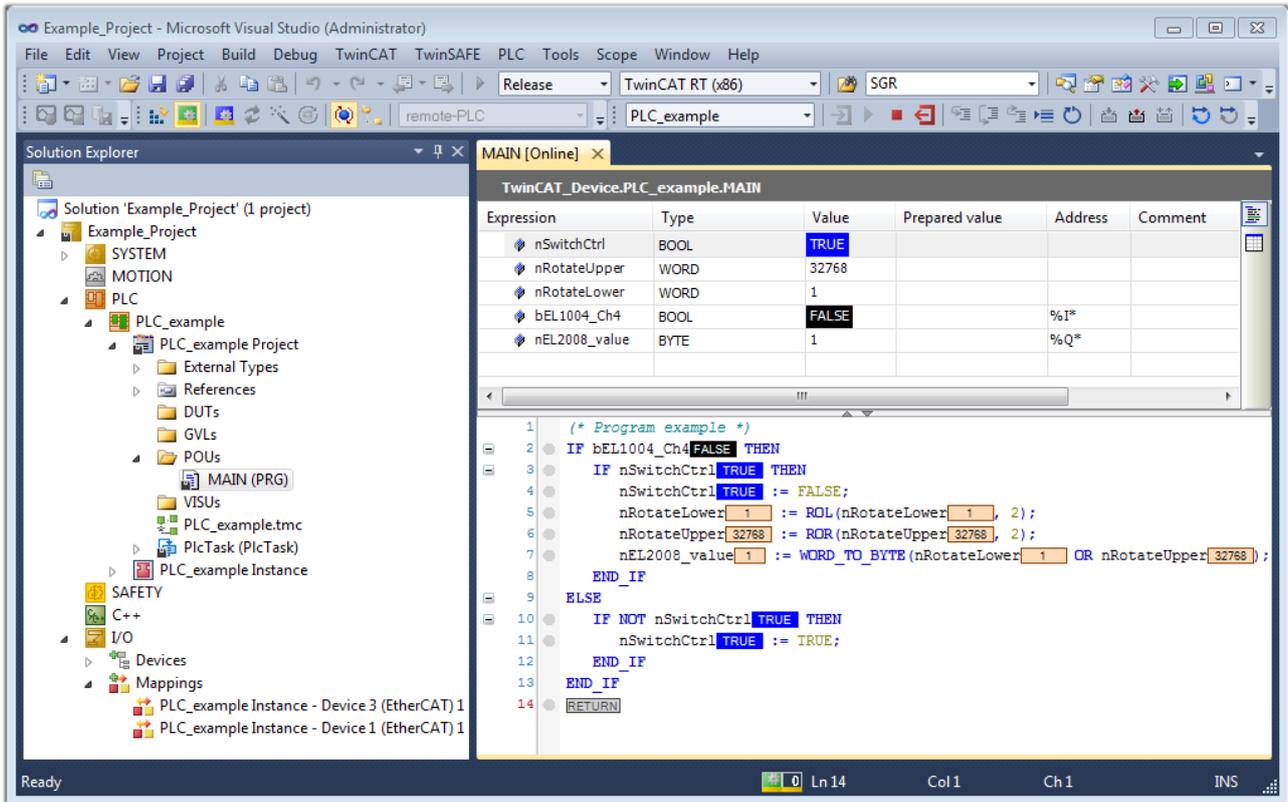


Fig. 68: TwinCAT 3 development environment (VS shell): logged-in, after program startup

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.

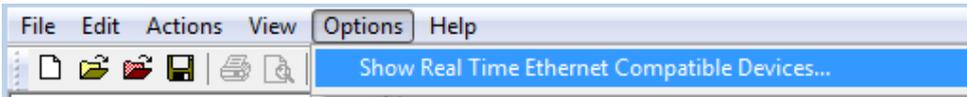


Fig. 69: System Manager “Options” (TwinCAT 2)

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

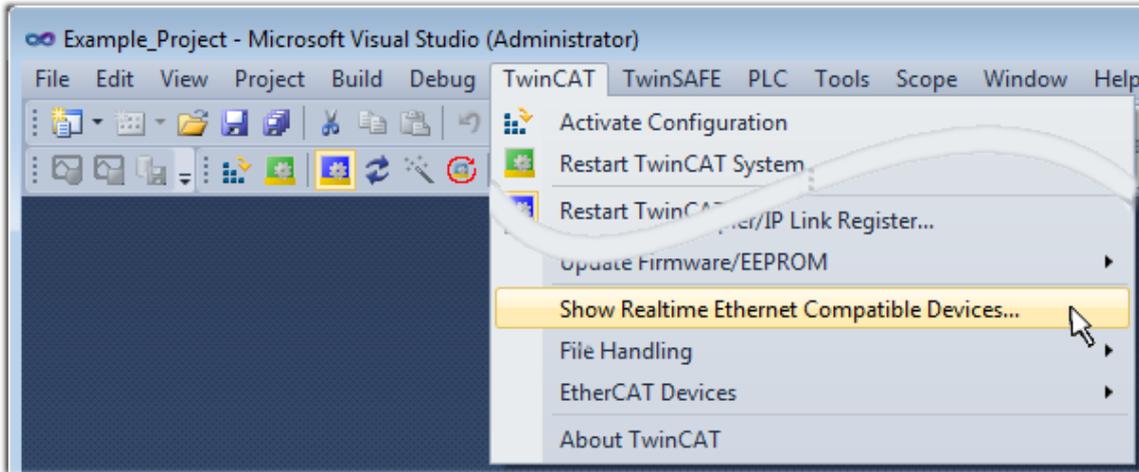


Fig. 70: Call up under VS Shell (TwinCAT 3)

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

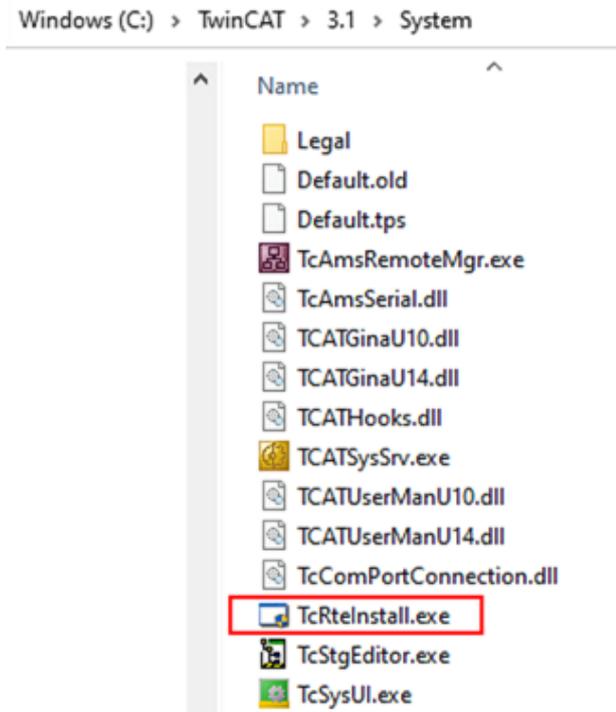


Fig. 71: TcRtelInstall in the TwinCAT directory

In both cases, the following dialog appears:

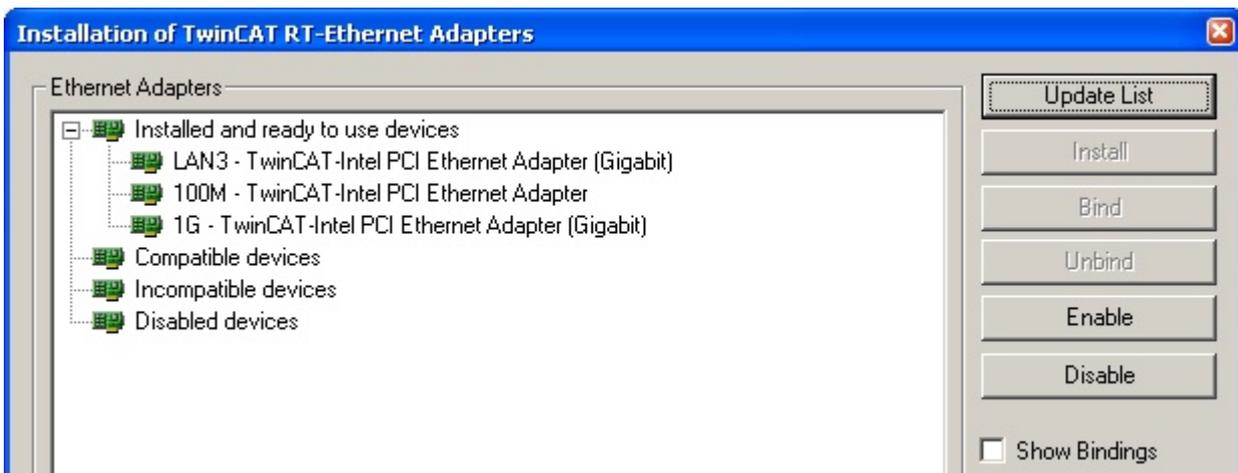


Fig. 72: Overview of network interfaces

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” [▶ 88] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices...”):



Fig. 73: 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)

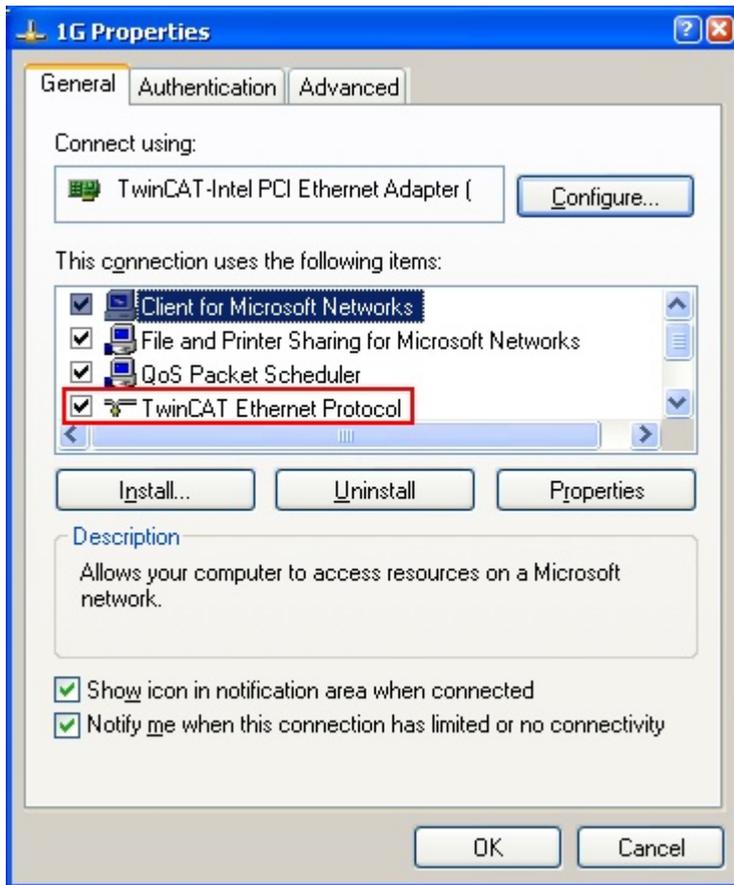


Fig. 74: Windows properties of the network interface

A correct setting of the driver could be:

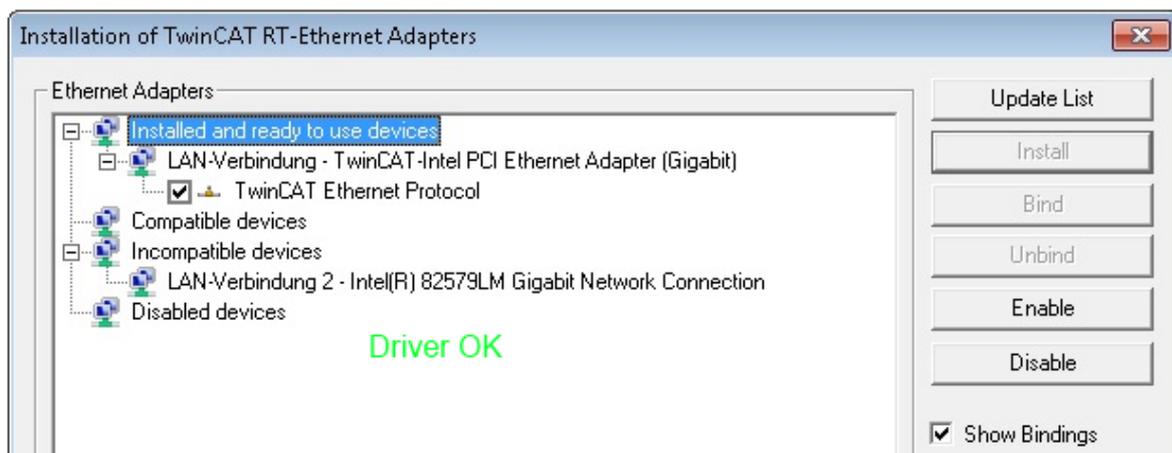


Fig. 75: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

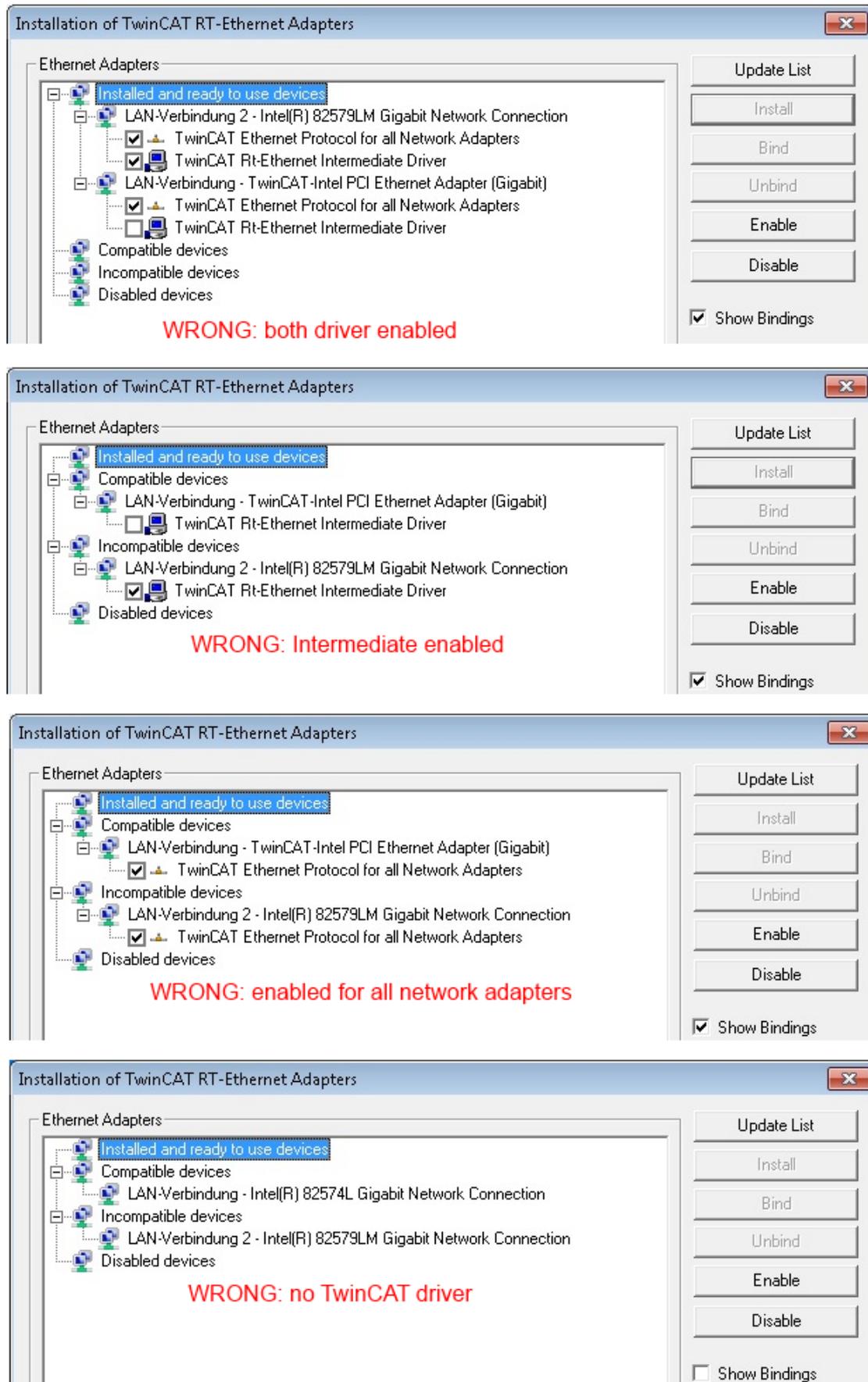


Fig. 76: Incorrect driver settings for the Ethernet port

## IP address of the port used

### ● IP address/DHCP

**i** 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.

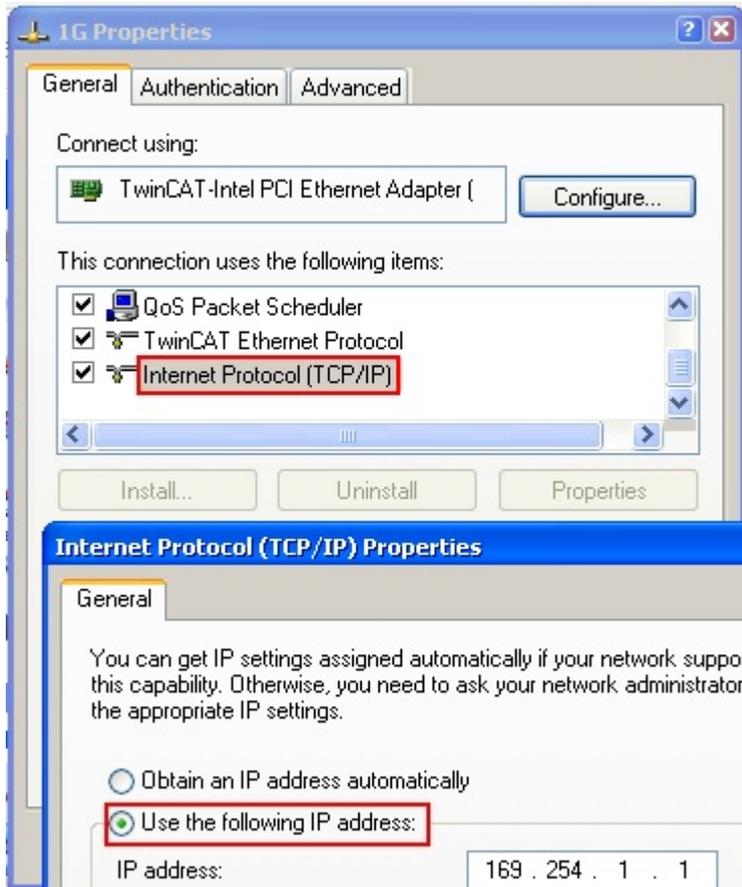


Fig. 77: TCP/IP setting for the Ethernet port

## 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](#) [▶ 87] 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”

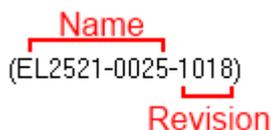


Fig. 78: Identifier structure

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](#) [▶ 10].

## 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.

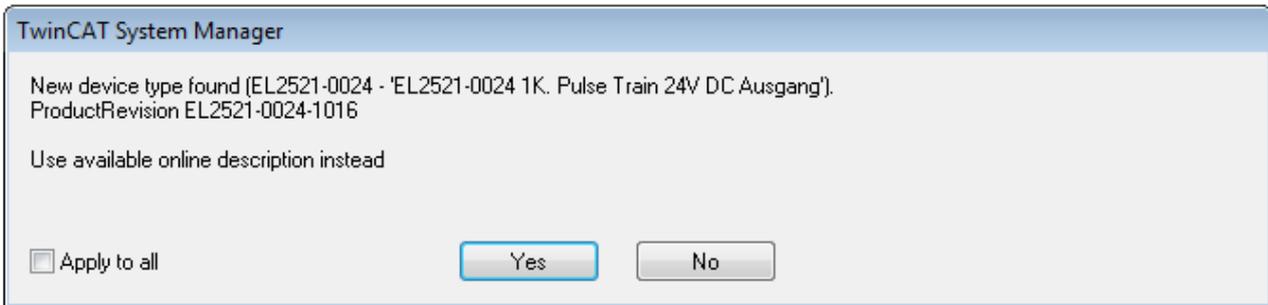


Fig. 79: OnlineDescription information window (TwinCAT 2)

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

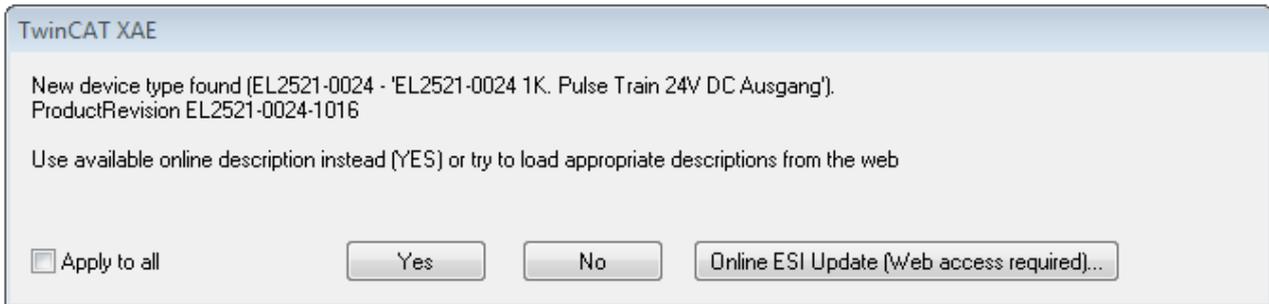


Fig. 80: Information window OnlineDescription (TwinCAT 3)

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 \[▶ 88\]](#)”.

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. 81: 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*).

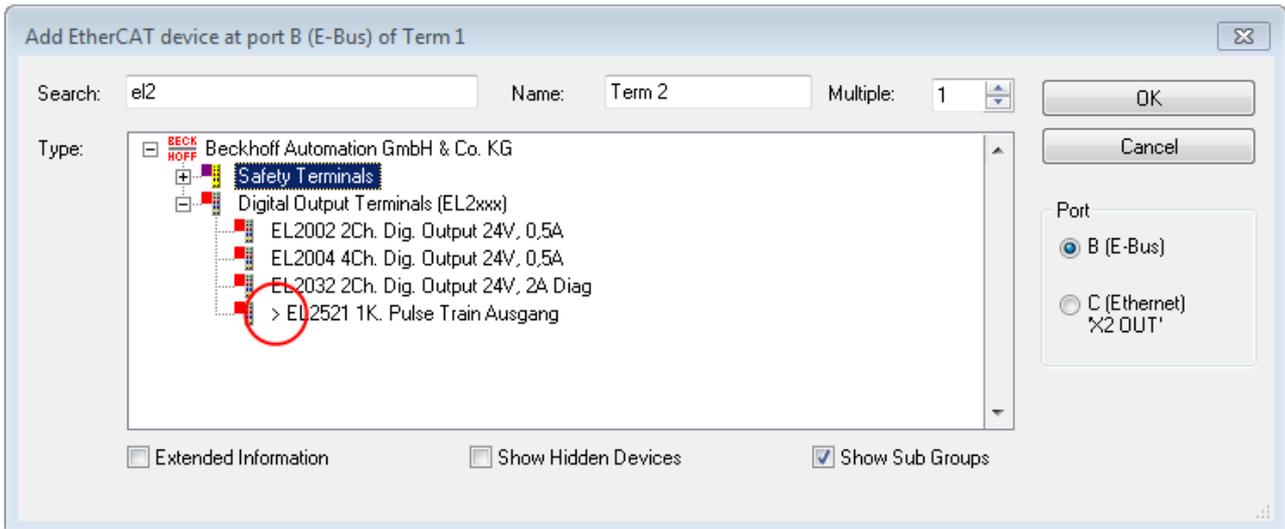


Fig. 82: Indication of an online recorded ESI of EL2521 as an example

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.

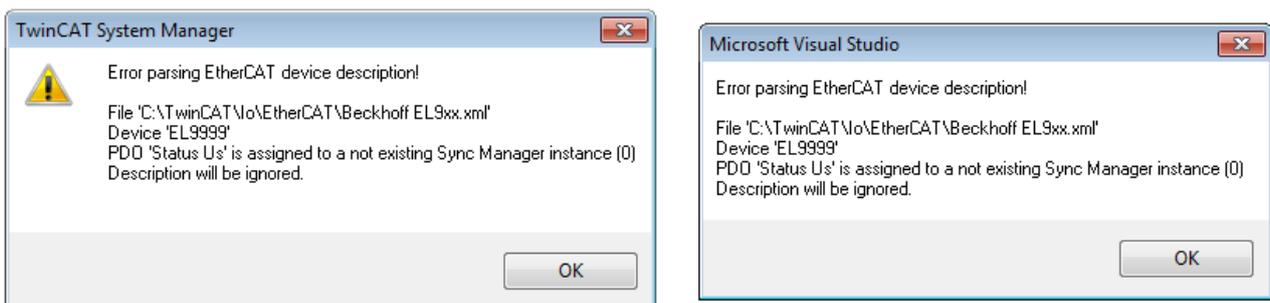


Fig. 83: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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:

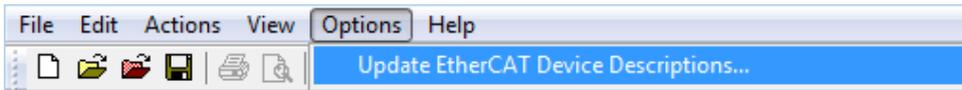


Fig. 84: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

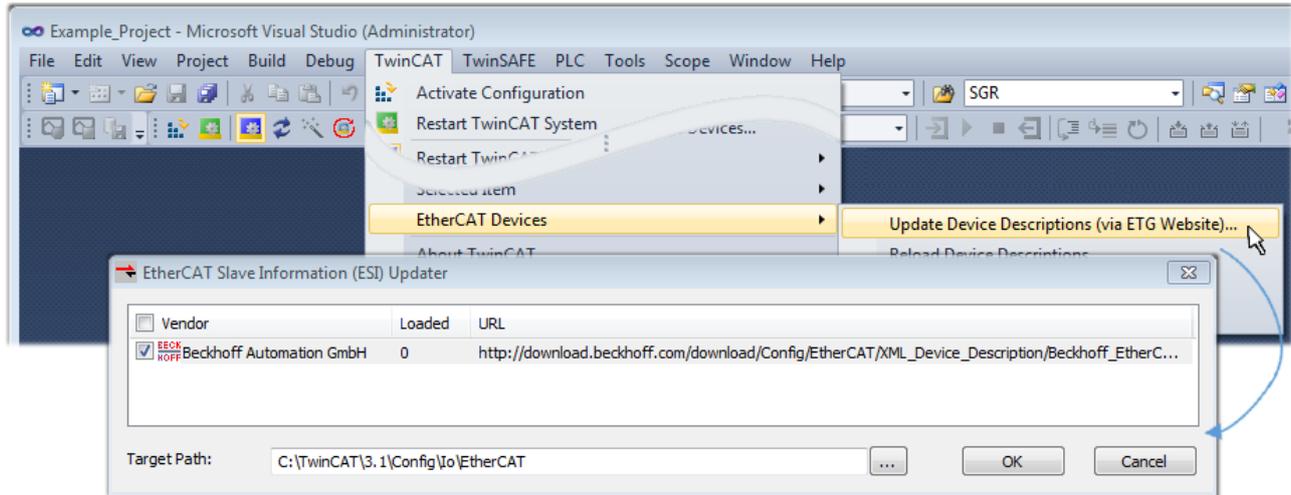


Fig. 85: Using the ESI Updater (TwinCAT 3)

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” \[▶ 83\]](#).

**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 [▶ 93] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 94]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 97]

The scan with existing configuration [▶ 98] 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.

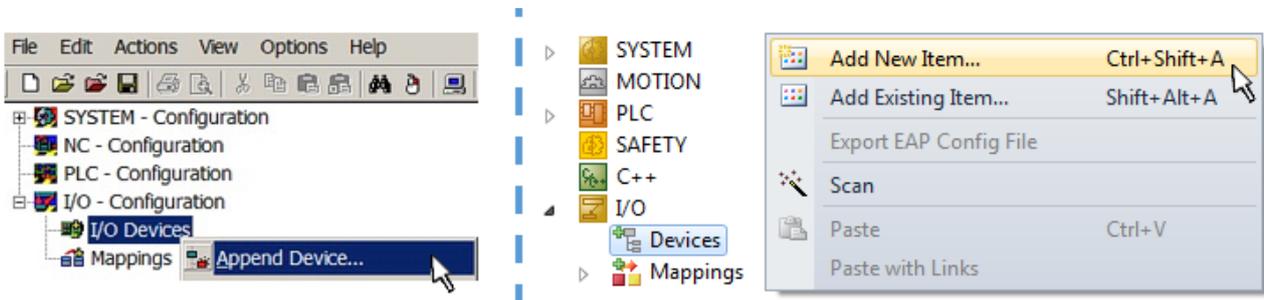


Fig. 86: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

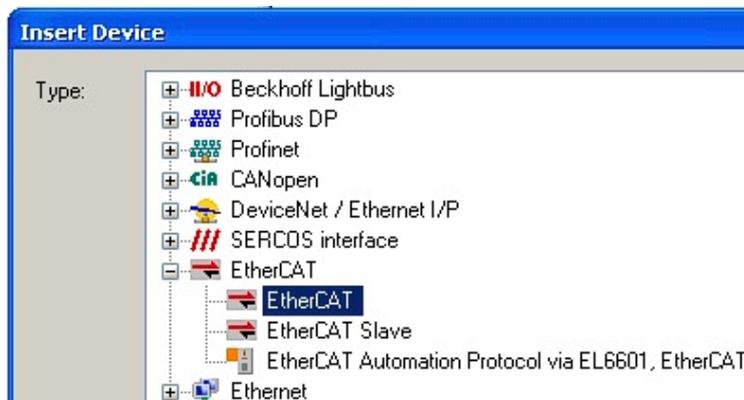


Fig. 87: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

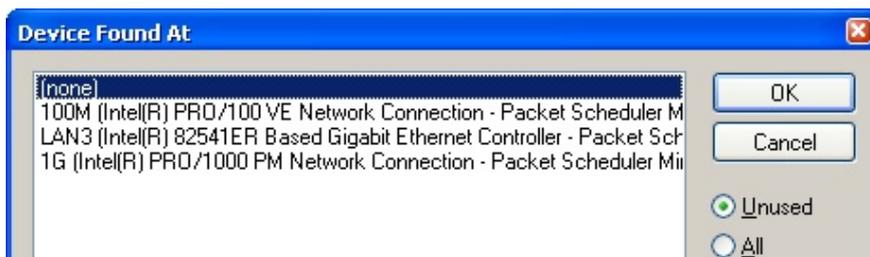


Fig. 88: Selecting the Ethernet port

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)”.

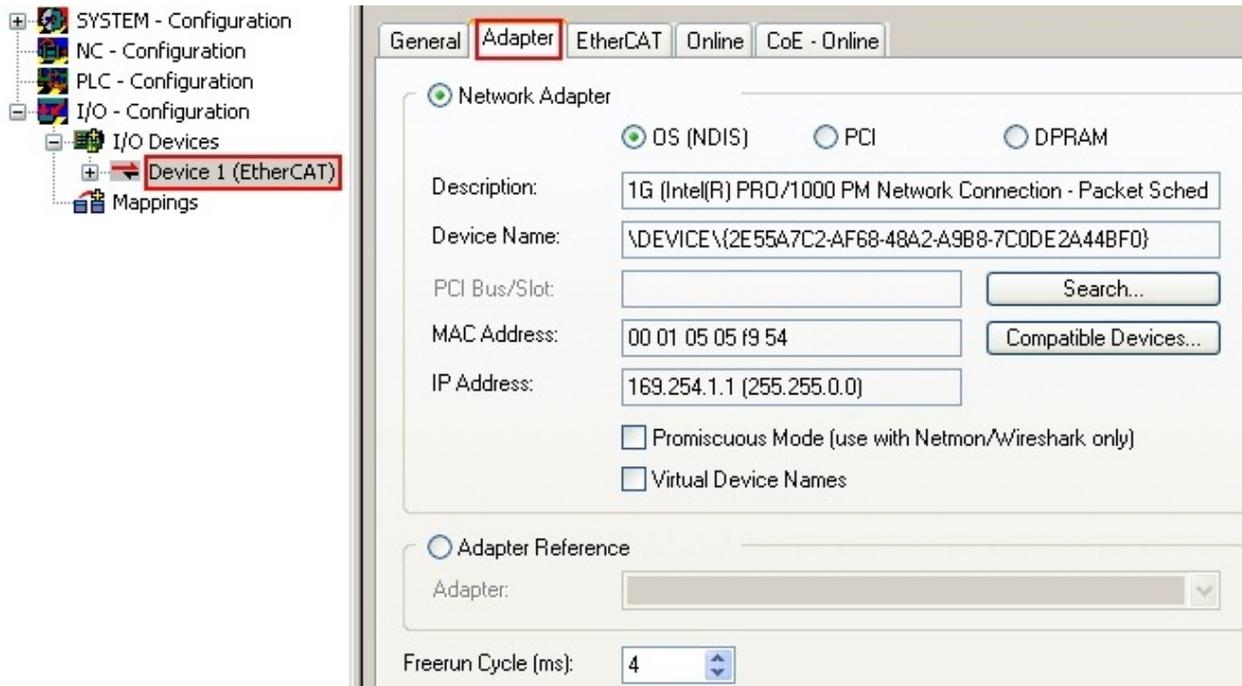


Fig. 89: EtherCAT device properties (TwinCAT 2)

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 \[▶ 77\]](#).

**Defining EtherCAT slaves**

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

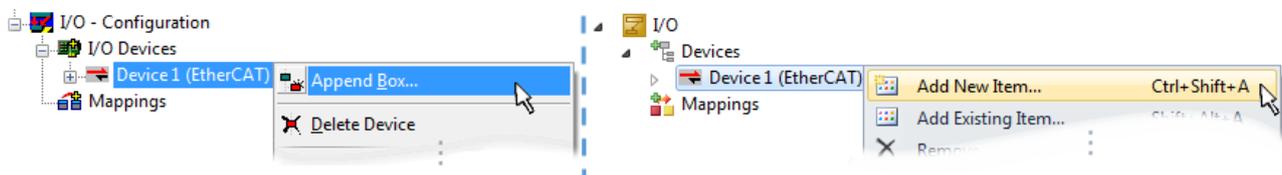


Fig. 90: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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).

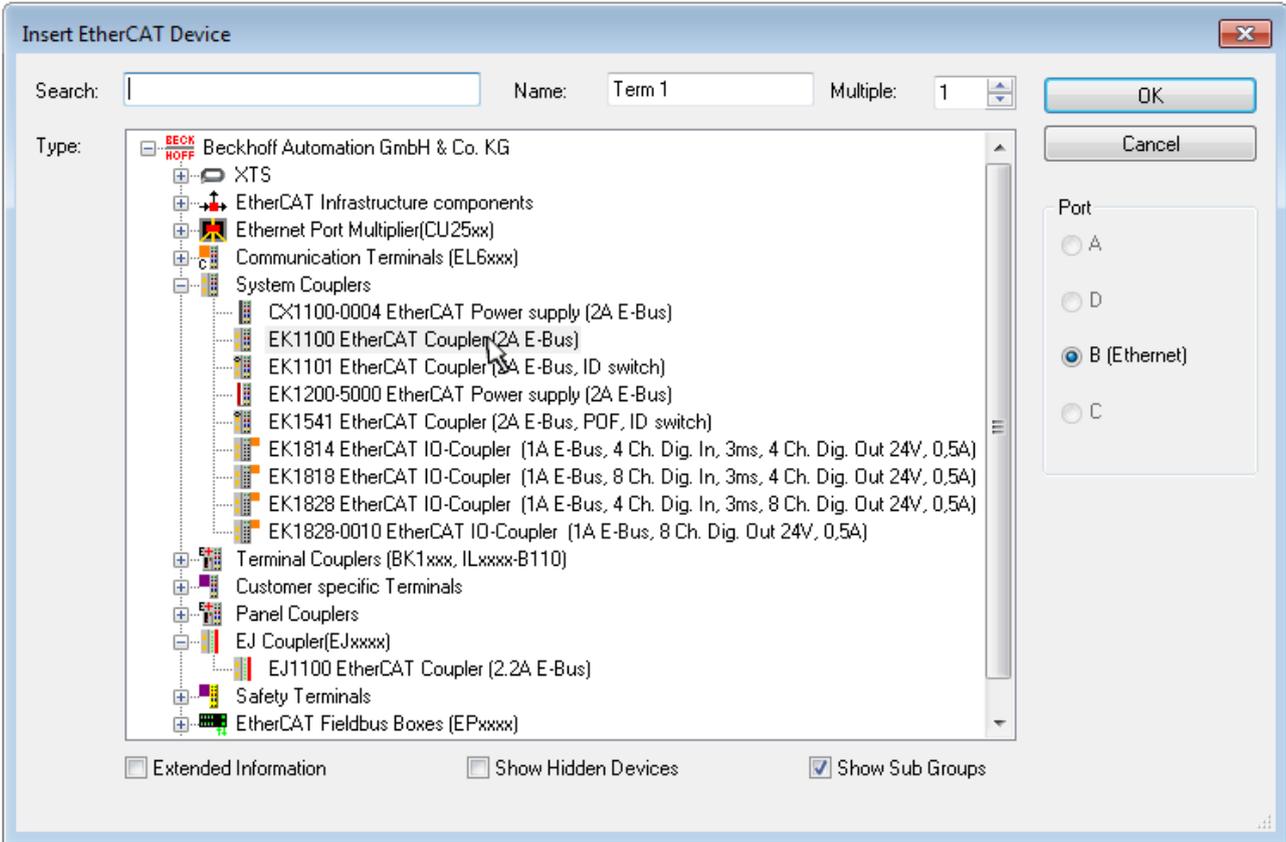


Fig. 91: Selection dialog for new EtherCAT device

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”.

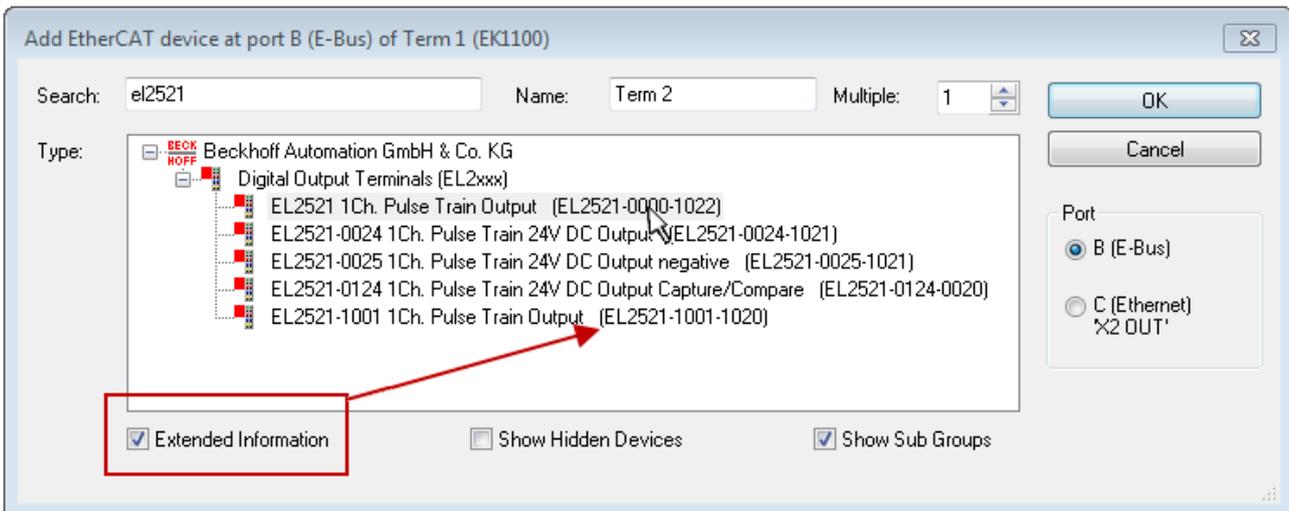


Fig. 92: Display of device revision

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”.

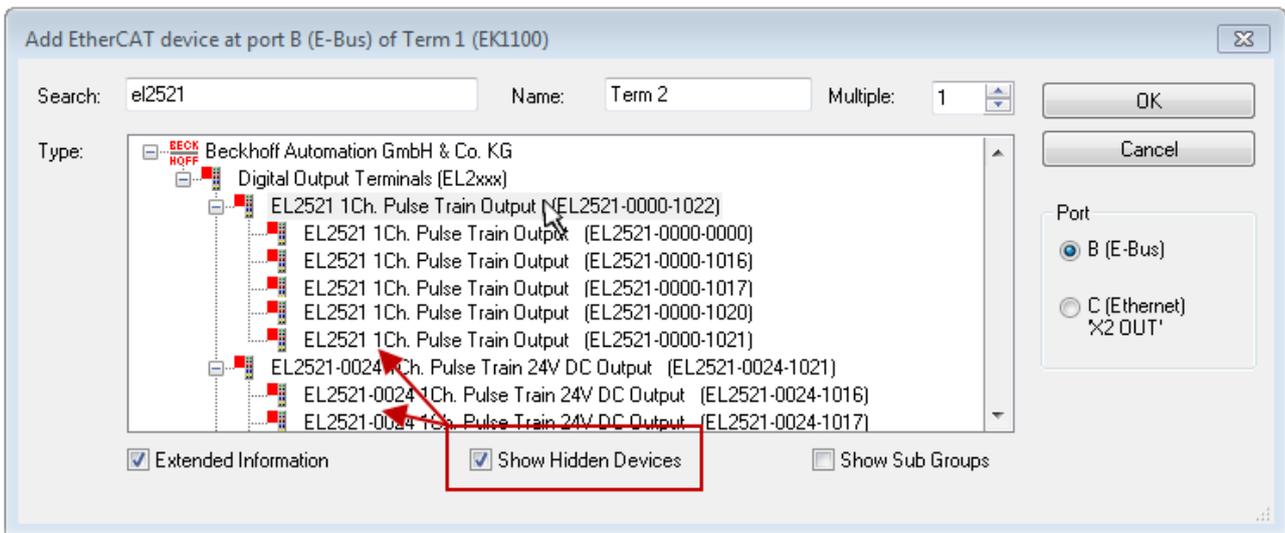


Fig. 93: Display of previous revisions

**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.

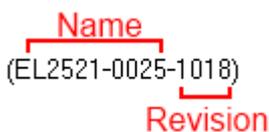


Fig. 94: Name/revision of the terminal

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, ...

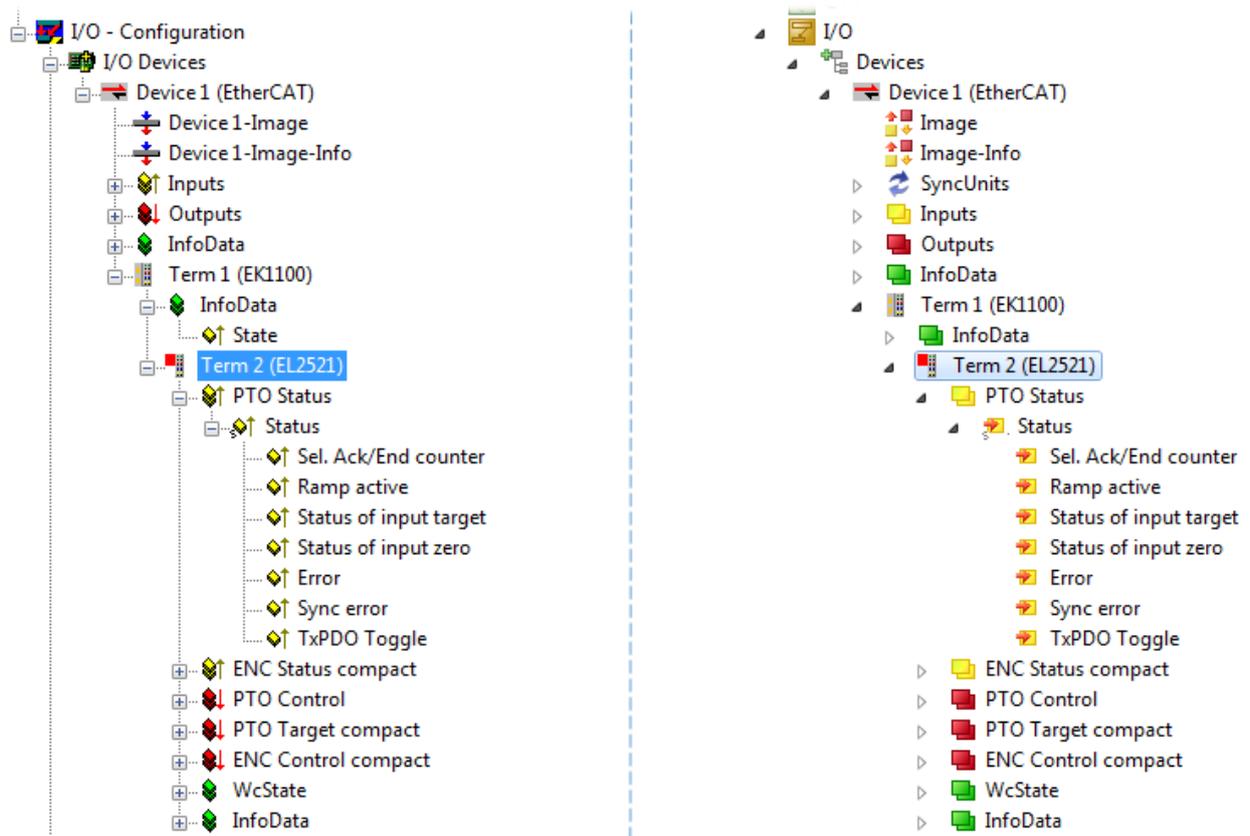


Fig. 95: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

## 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. 96: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

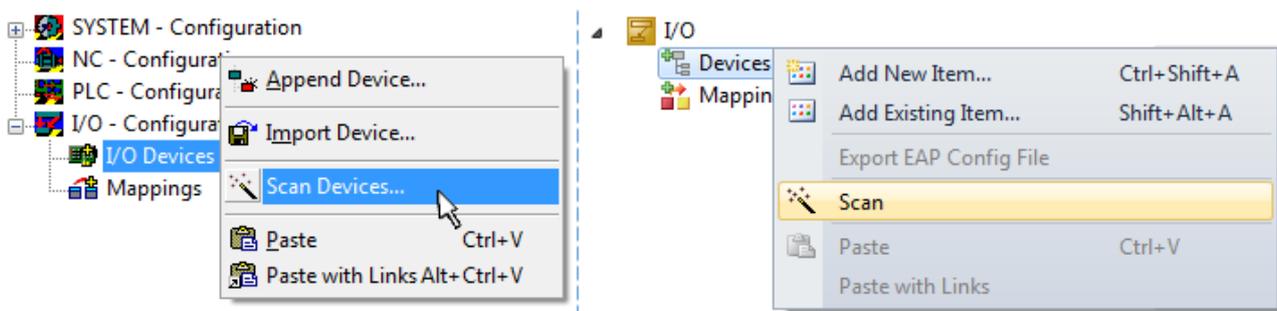


Fig. 97: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

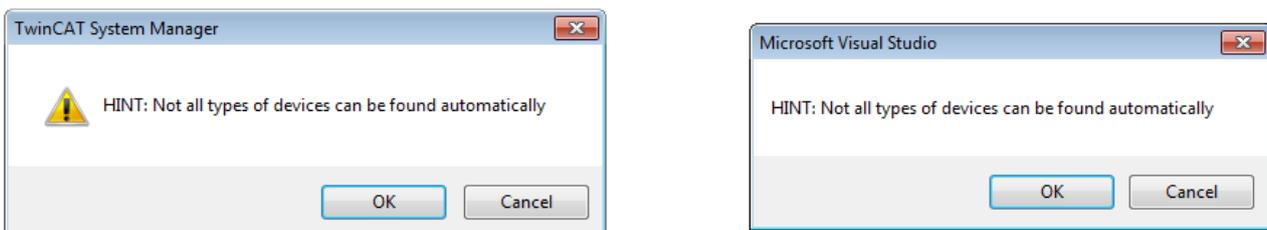


Fig. 98: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

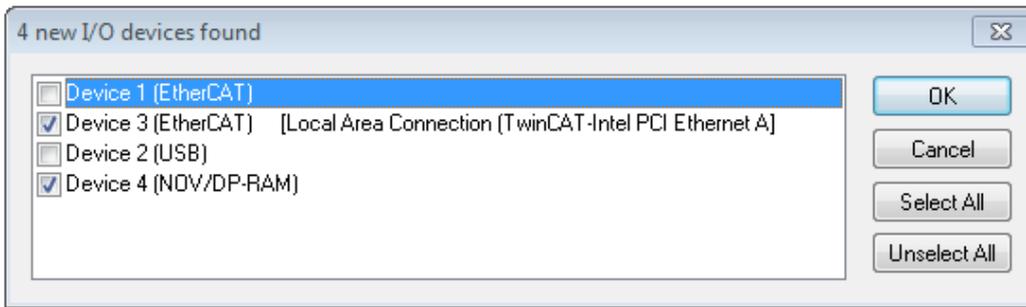


Fig. 99: Detected Ethernet devices

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 \[▶ 77\]](#).

## 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.

**Name**  
(EL2521-0025-1018)  
**Revision**

Fig. 100: 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 \[▶ 98\]](#) 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:

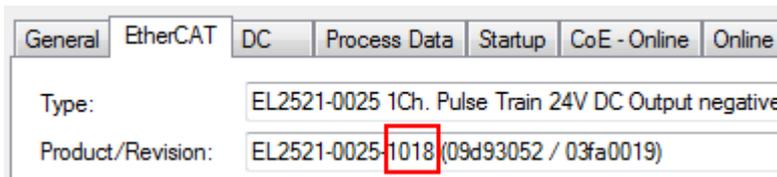


Fig. 101: Installing EthetCAT terminal with revision -1018

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

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

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and a **new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of “B.tsm” or even “B.pro” is therefore unnecessary. The series-produced machines can continue to be built with “B.tsm” and “B.pro”; it makes sense to perform a comparative scan [► 98] 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:

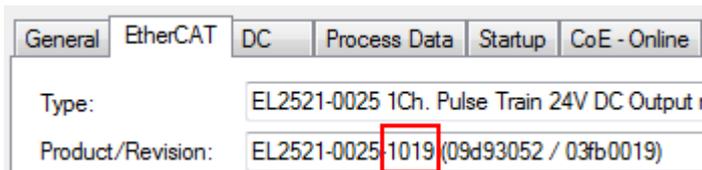


Fig. 102: Detection of EtherCAT terminal with revision -1019

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.

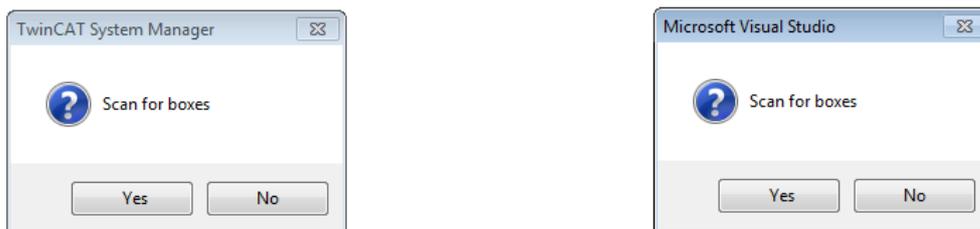


Fig. 103: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

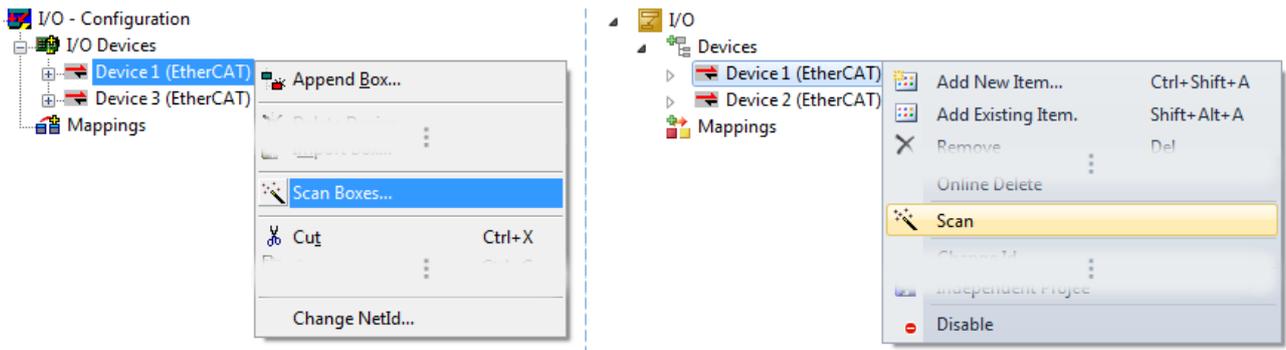


Fig. 104: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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.

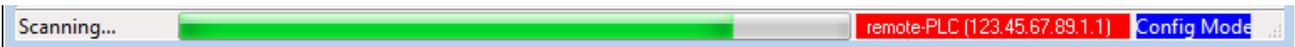


Fig. 105: Scan progress example by TwinCAT 2

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



Fig. 106: 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. 107: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 108: 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*.

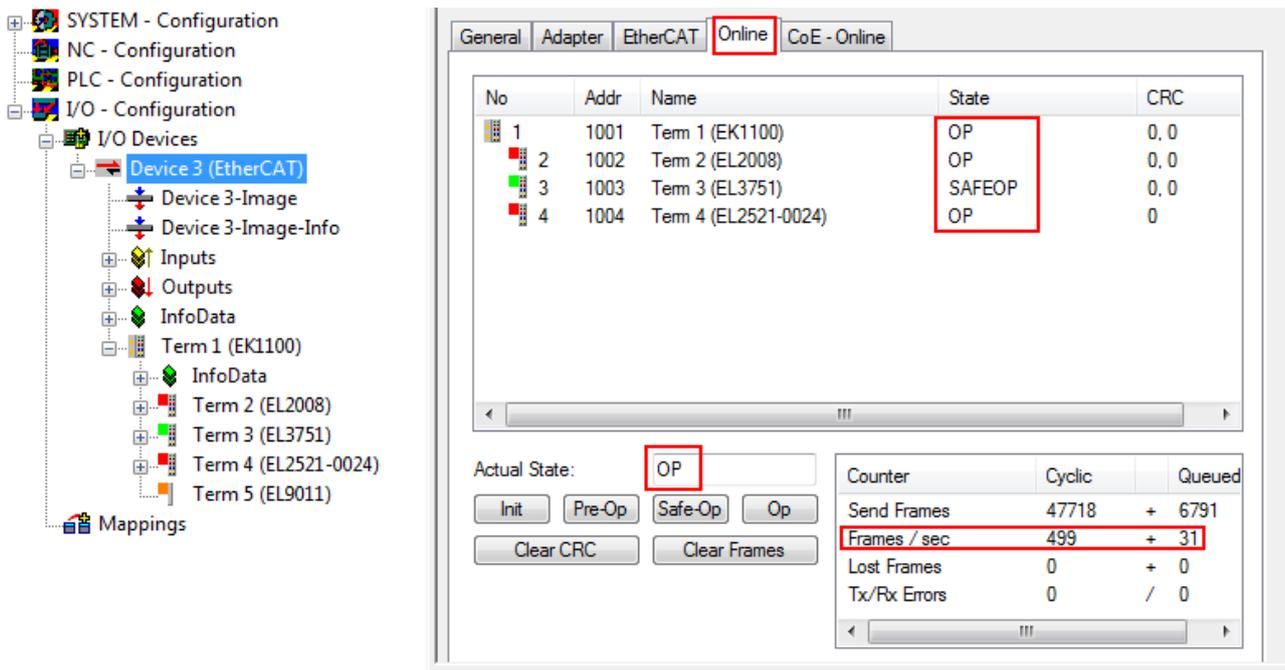


Fig. 109: Online display example

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 \[► 88\]](#).

### 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.

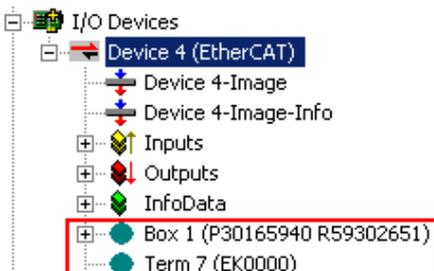


Fig. 110: Faulty identification

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. 111: 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.

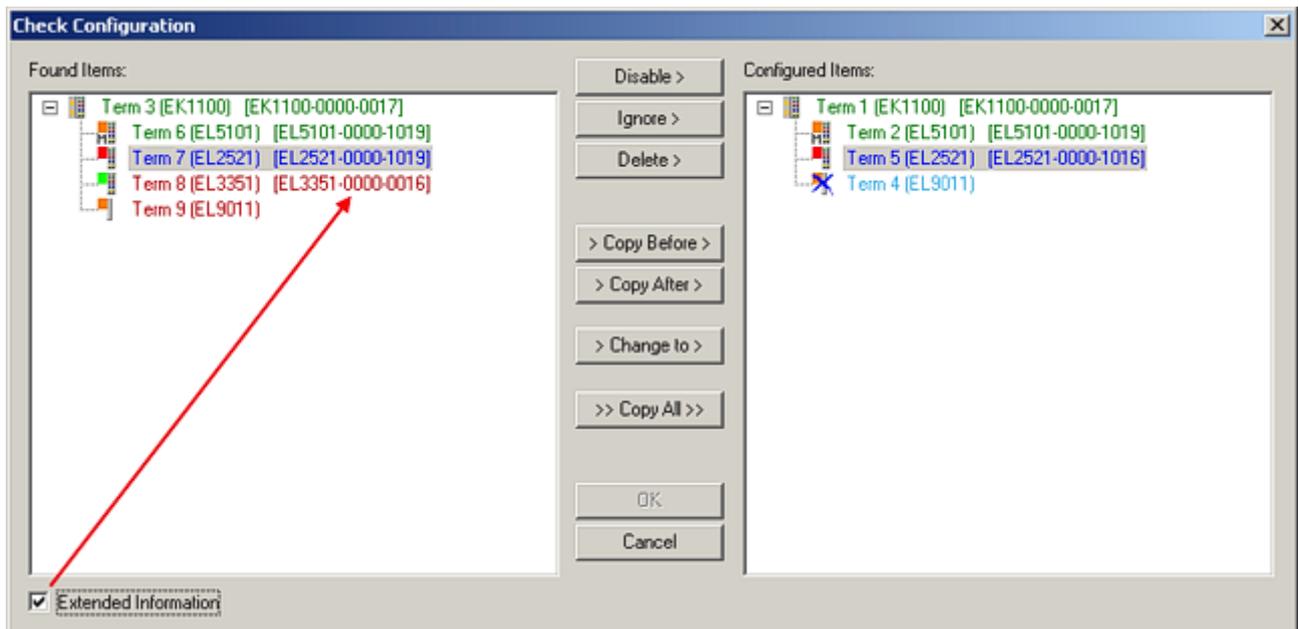


Fig. 112: Correction dialog

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 >= 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.

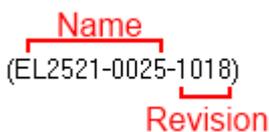


Fig. 113: Name/revision of the terminal

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, ...

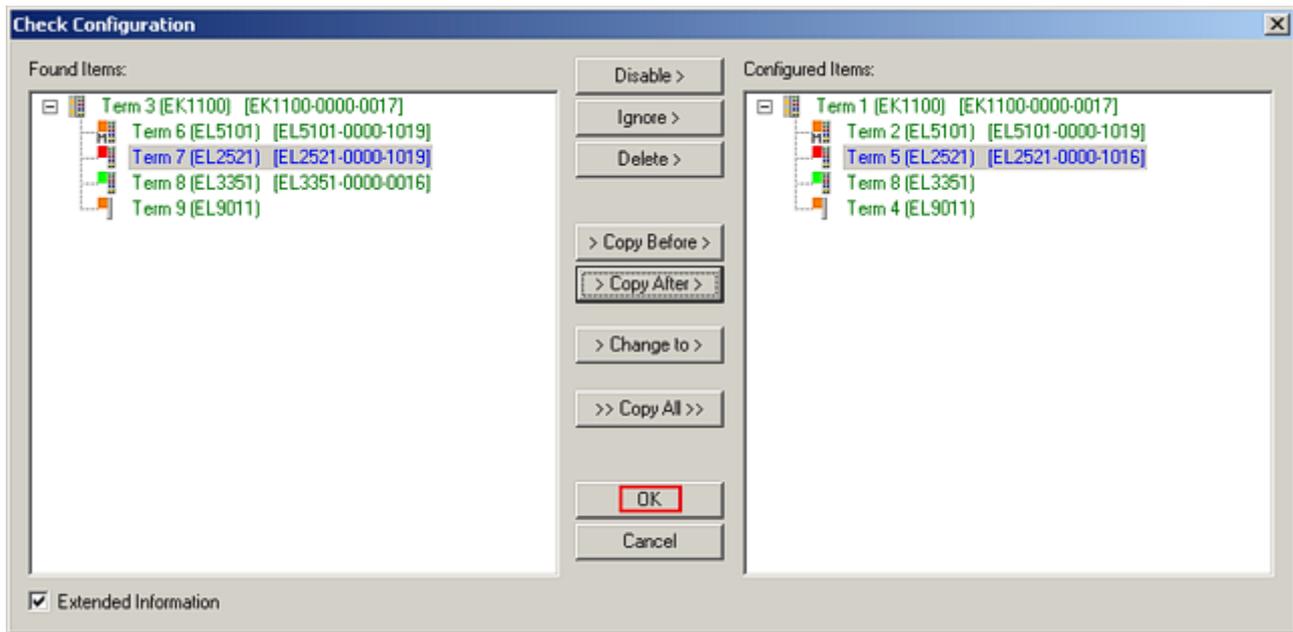


Fig. 114: Correction dialog with modifications

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.

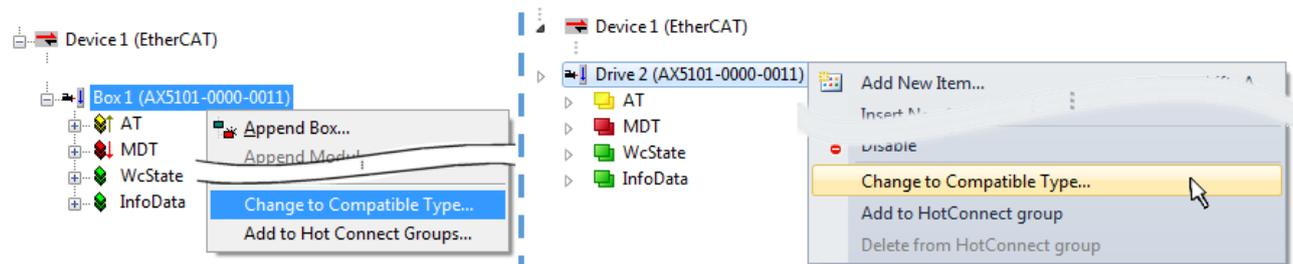


Fig. 115: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

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, Entry.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

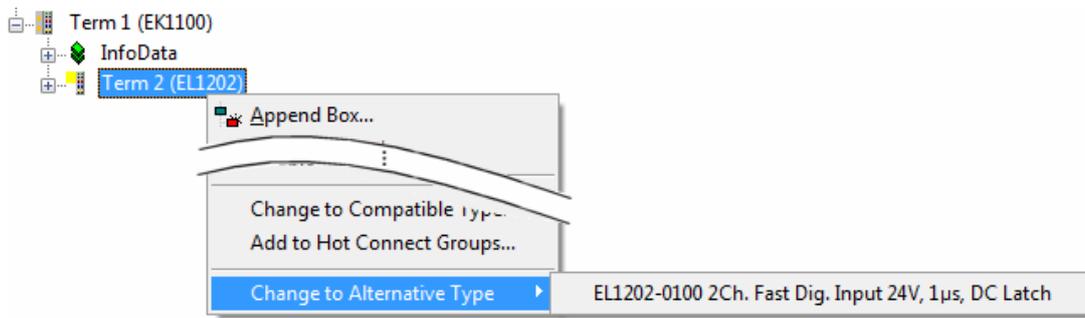


Fig. 116: TwinCAT 2 Dialog Change to Alternative Type

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).

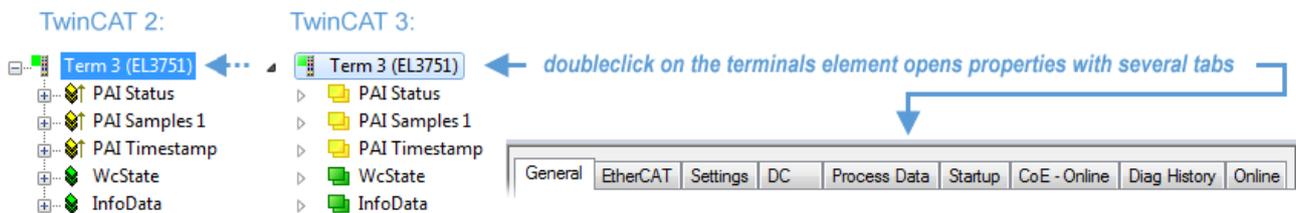


Fig. 117: Branch element as terminal EL3751

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

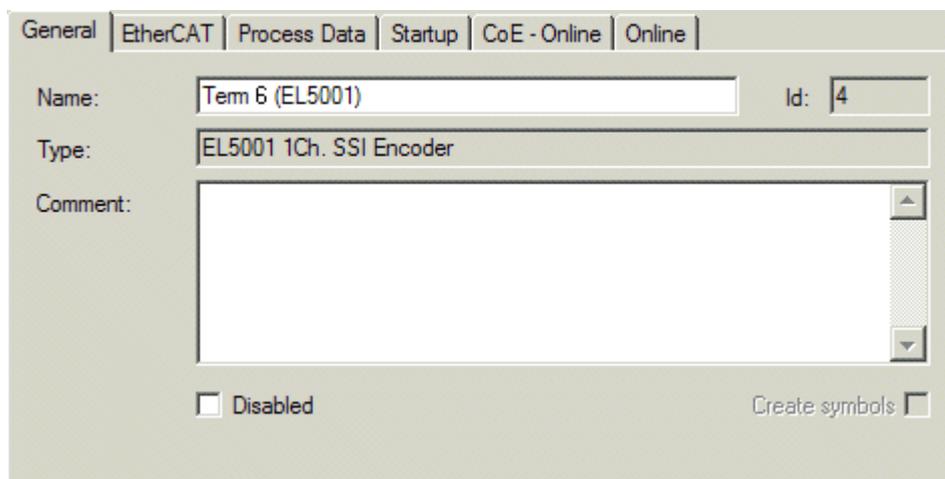


Fig. 118: “General” tab

<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. 119: “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.

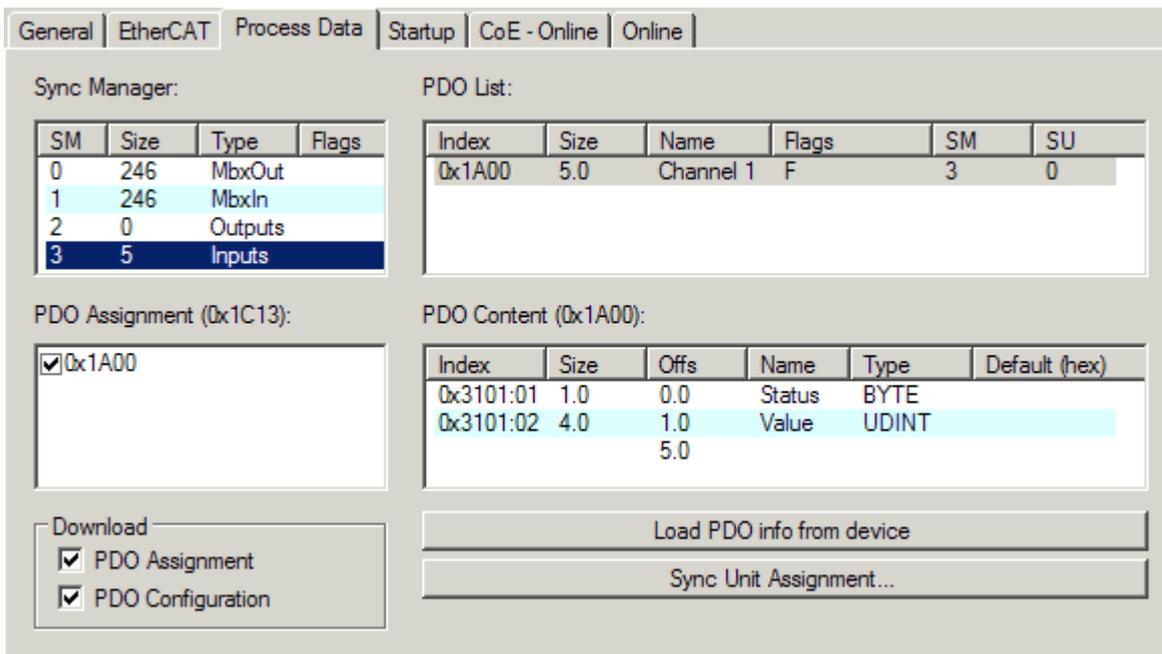


Fig. 120: "Process Data" tab

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").

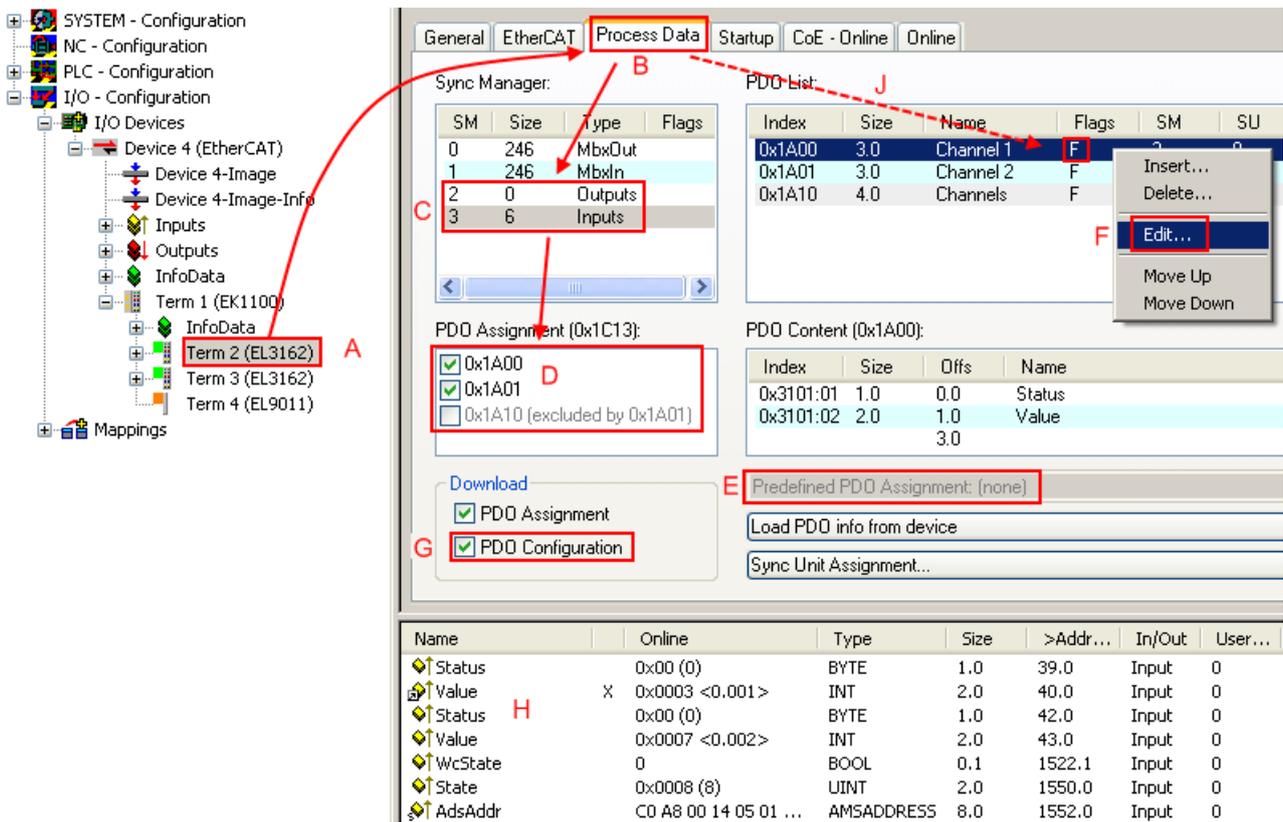


Fig. 121: Configuring the process data

**i 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 [► 109] 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.

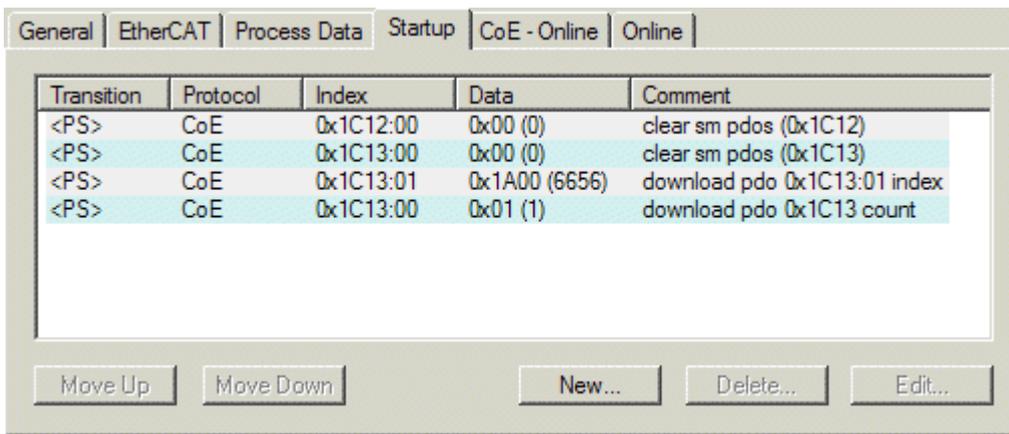


Fig. 122: “Startup” tab

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.

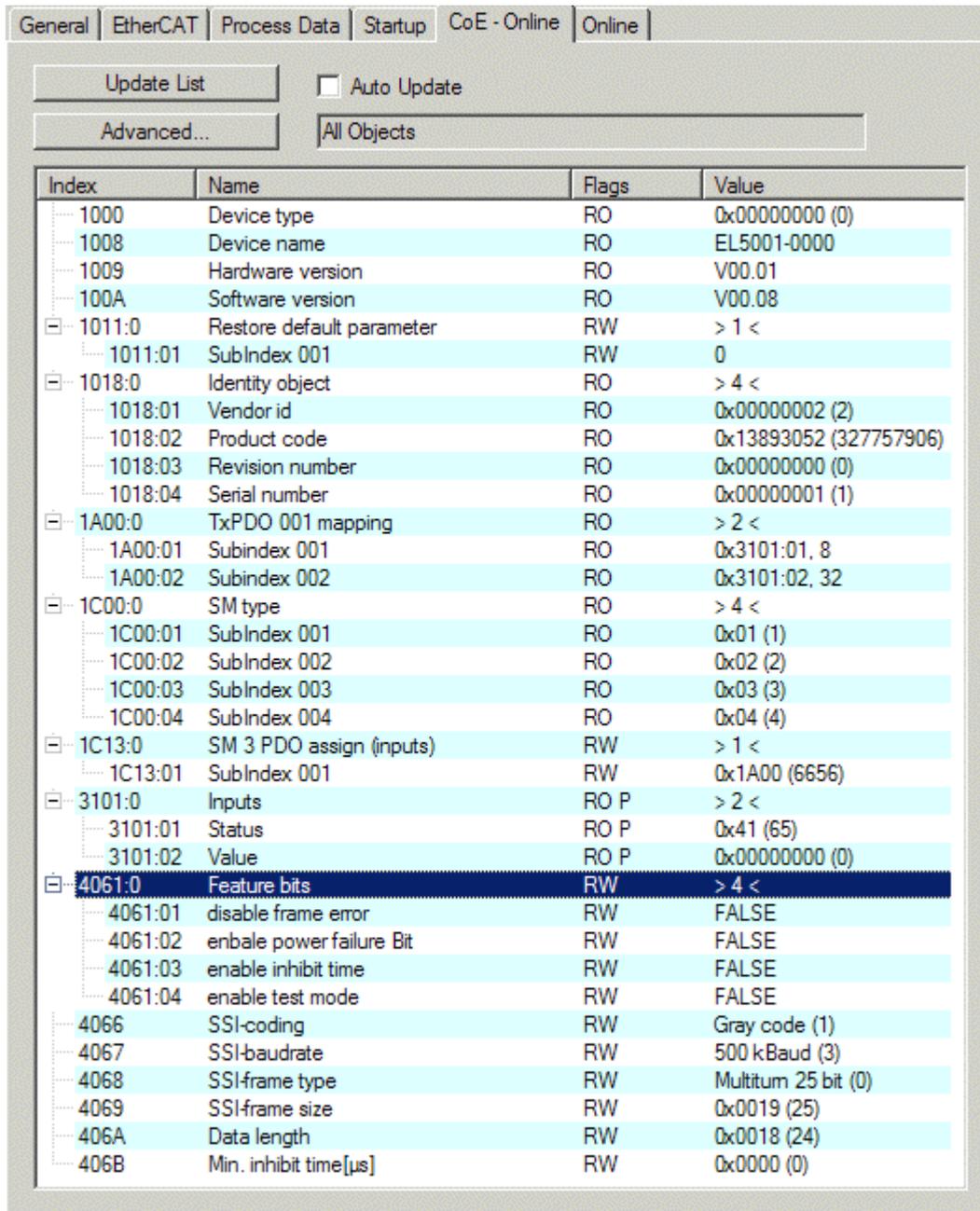


Fig. 123: "CoE - Online" tab

**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.

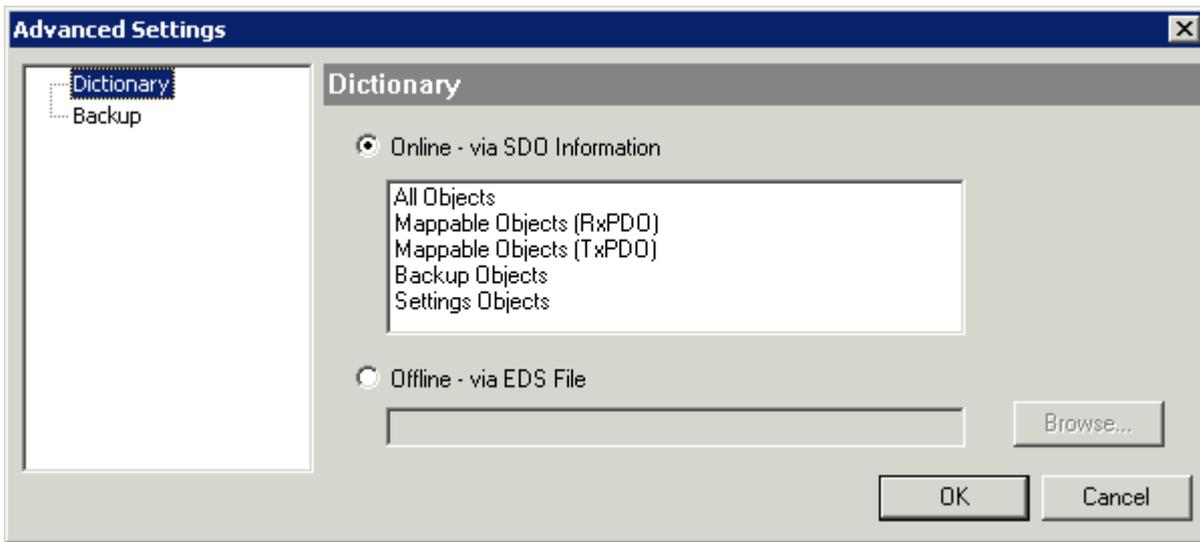


Fig. 124: Dialog “Advanced settings”

**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

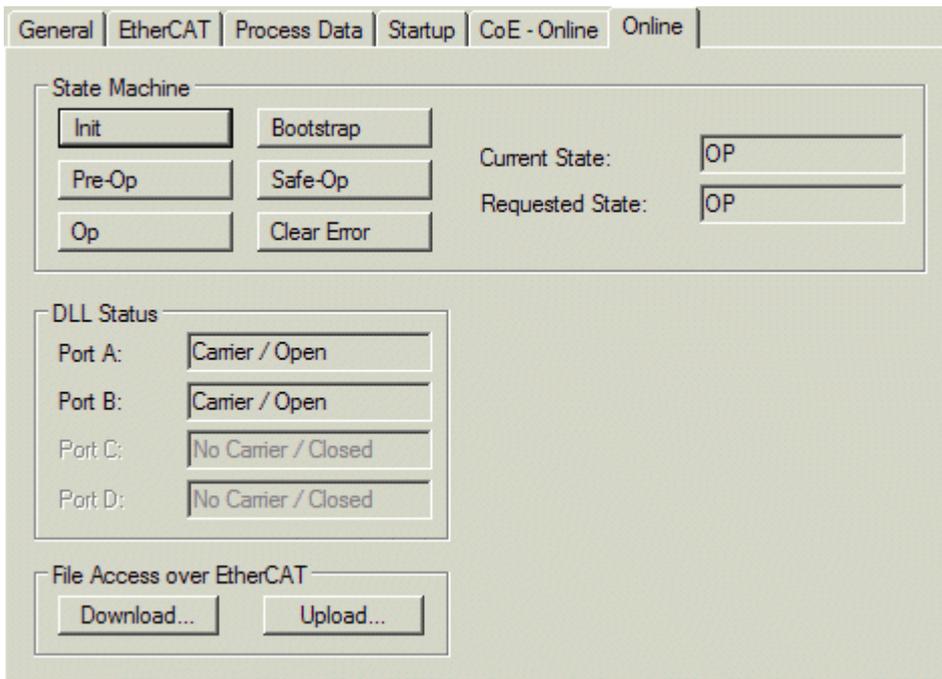


Fig. 125: “Online” tab

**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)**

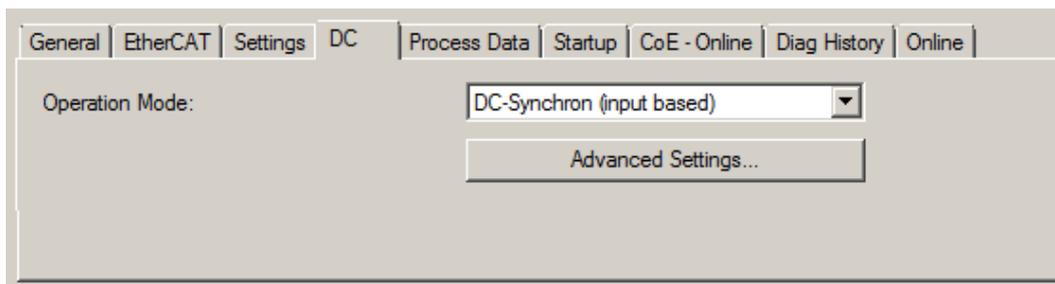


Fig. 126: “DC” tab (Distributed Clocks)

- 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 \[▶ 107\]](#)),
  - 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 \[► 104\]](#) 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.

### 5.3 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.

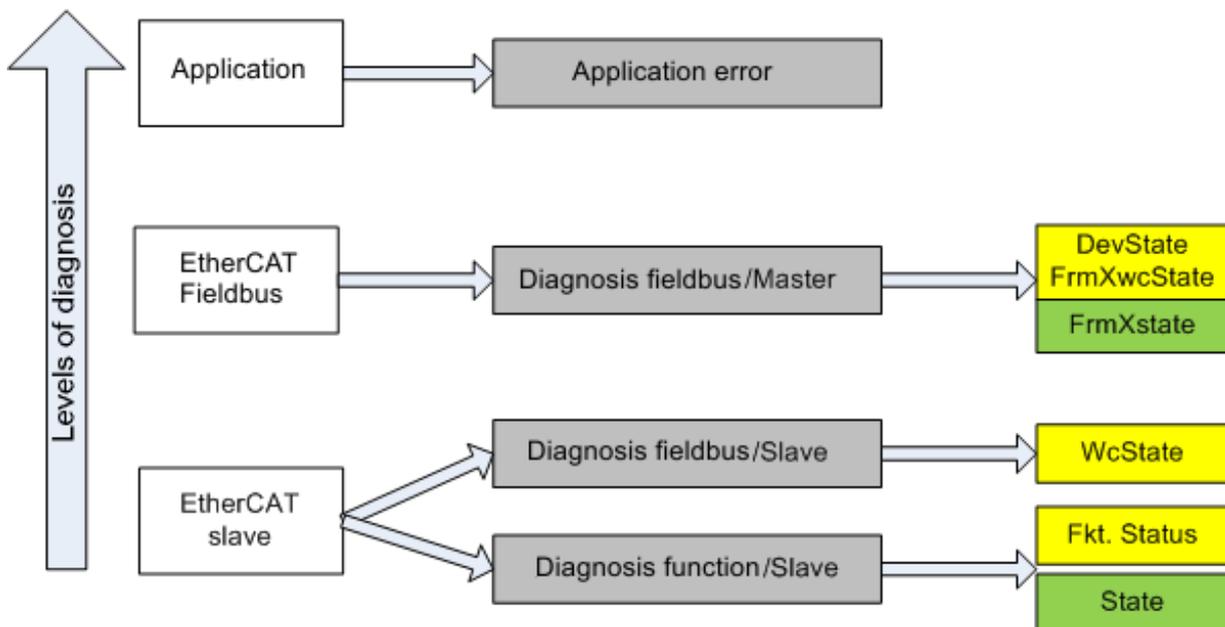


Fig. 127: Selection of the diagnostic information of an EtherCAT Slave

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.

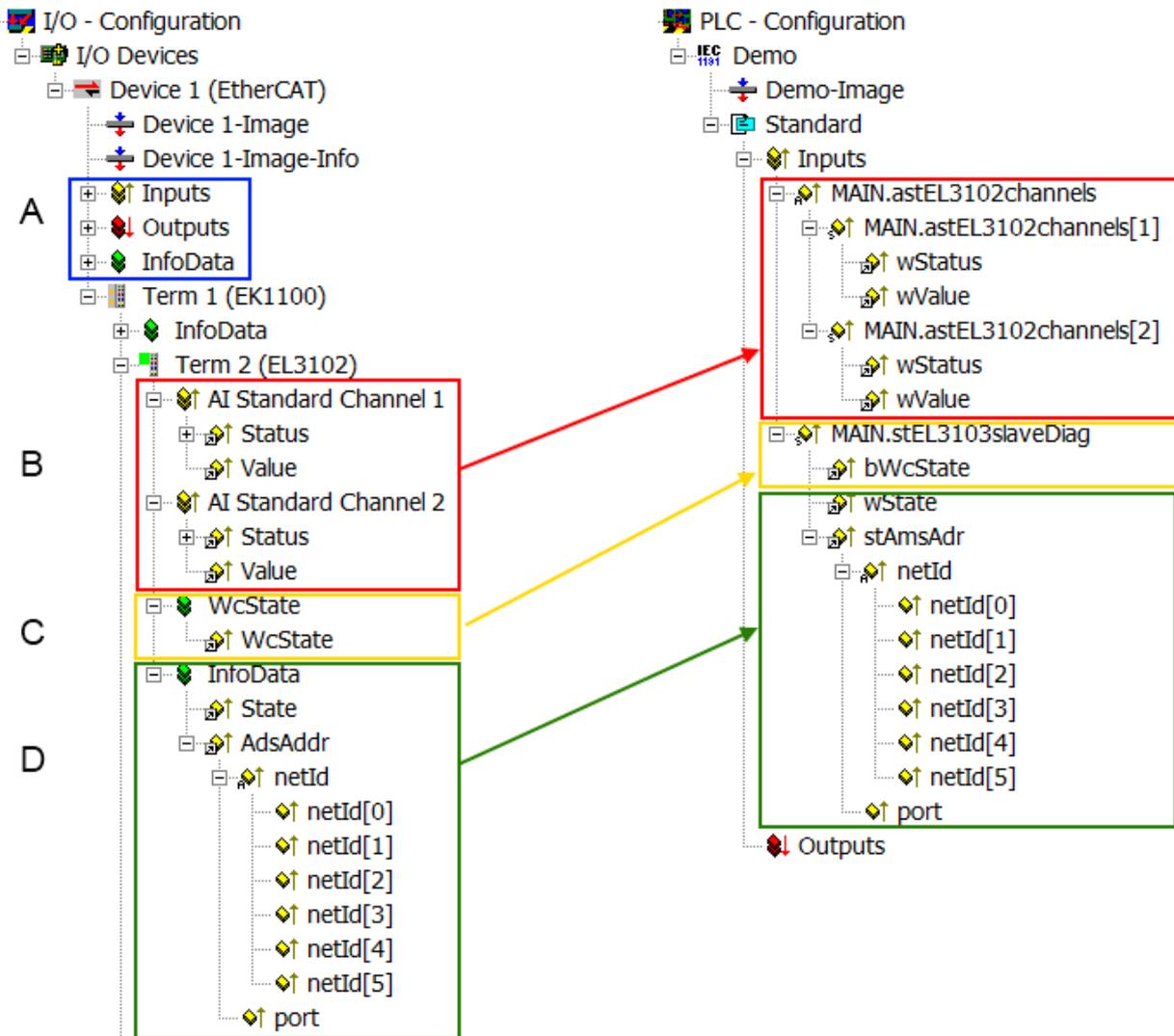


Fig. 128: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated cyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: <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*:

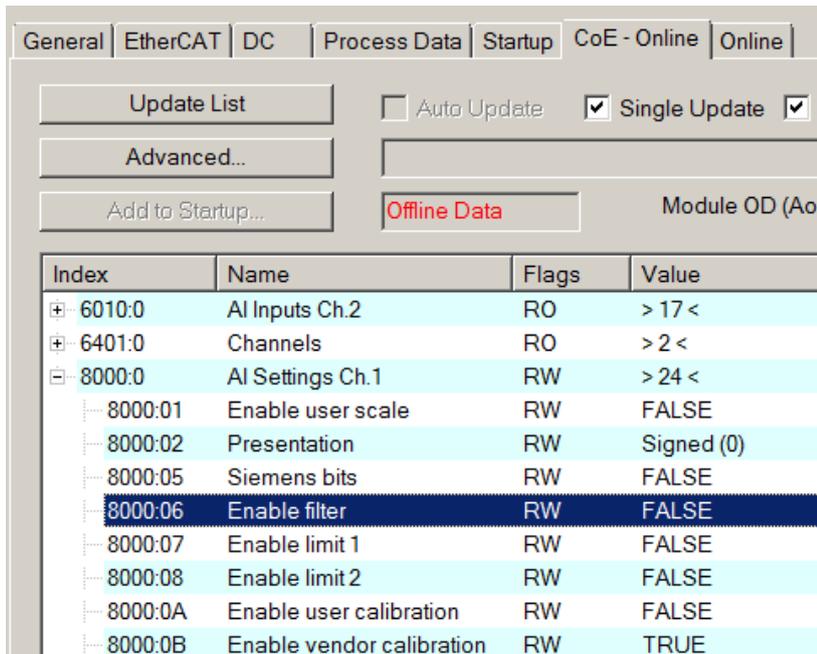


Fig. 129: EL3102, CoE directory

**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.

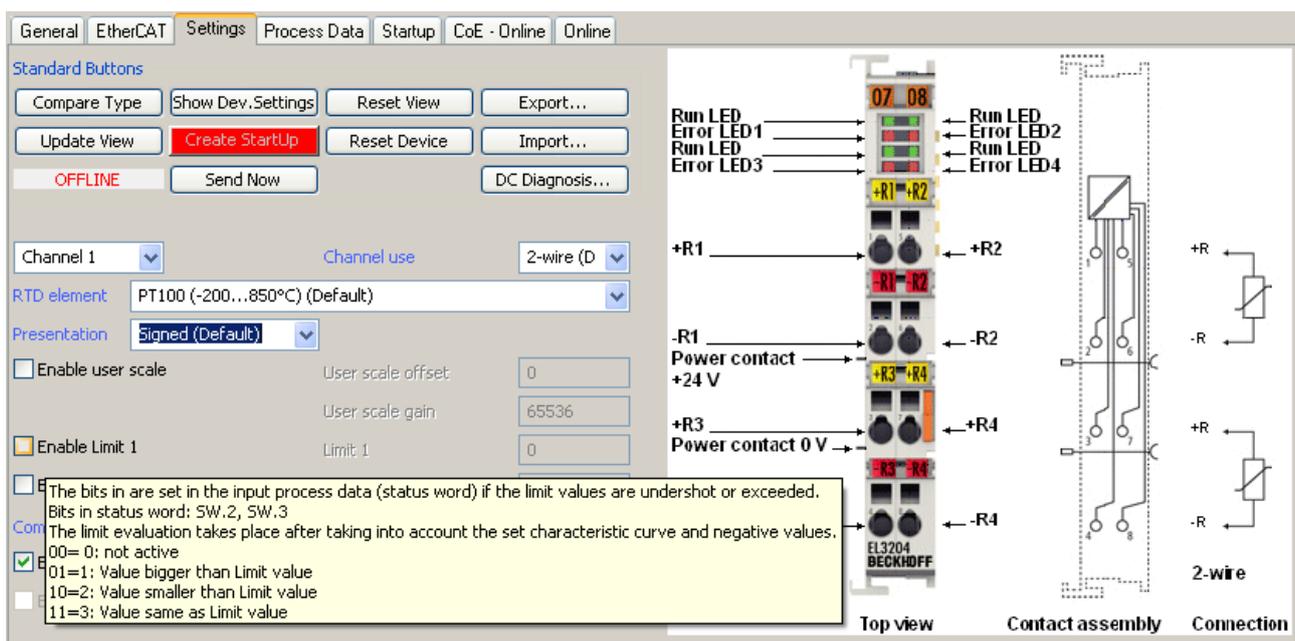


Fig. 130: Example of commissioning aid for a EL3204

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 [▶ 25]" 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.

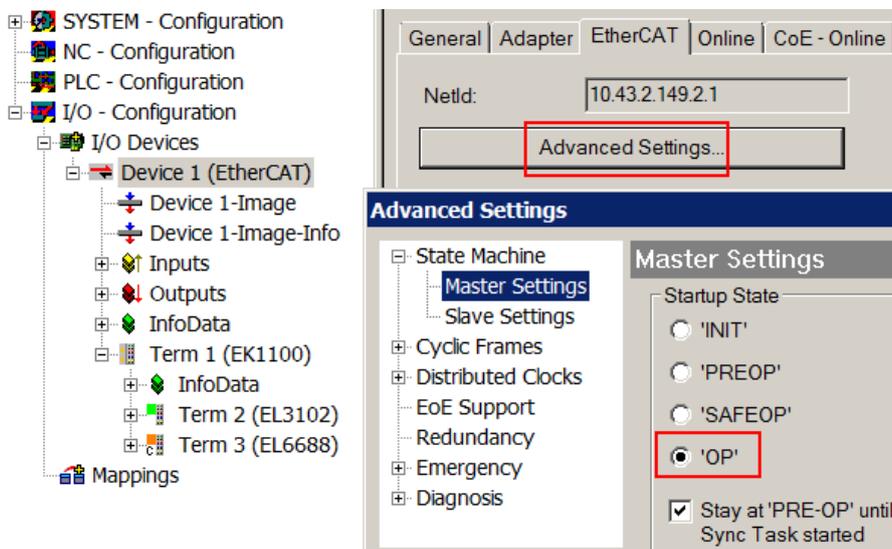


Fig. 131: Default behaviour of the System Manager

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

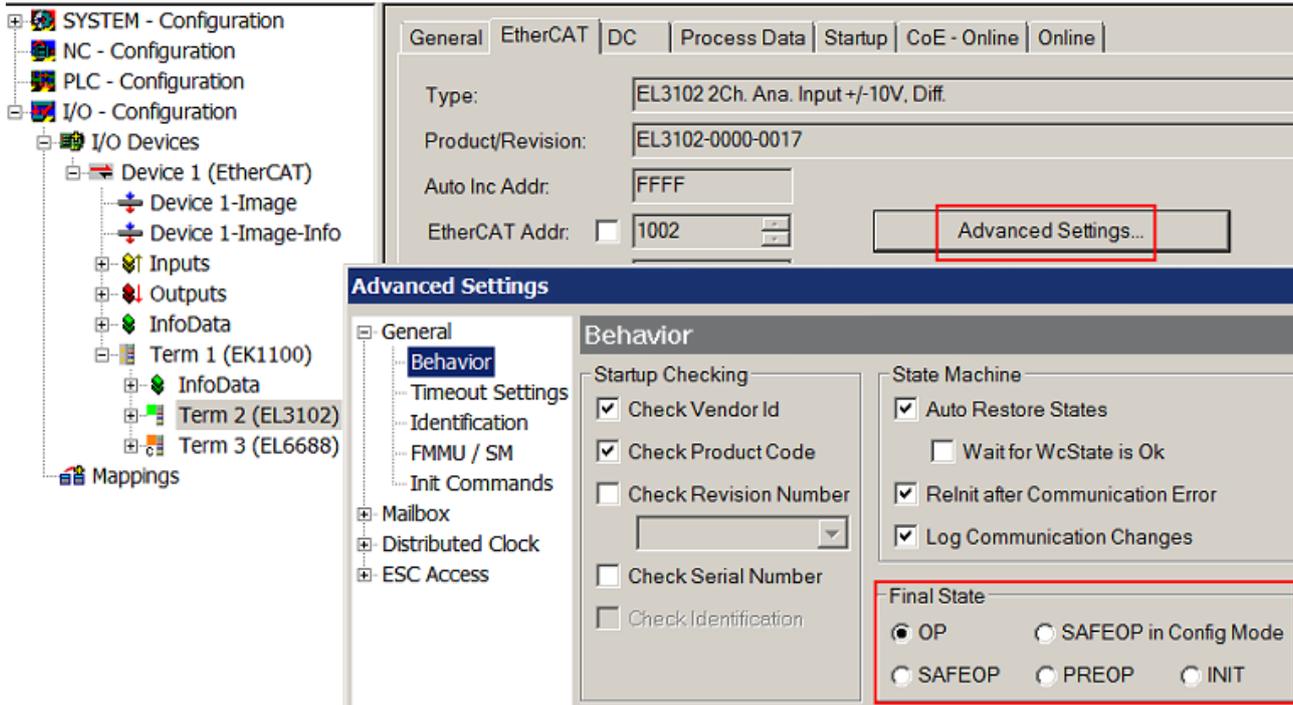


Fig. 132: Default target state in the Slave

### 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.

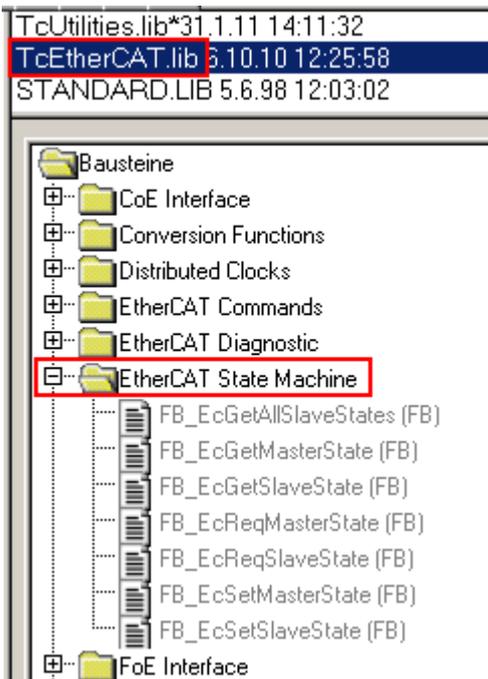


Fig. 133: PLC function blocks

**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. 134: 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:

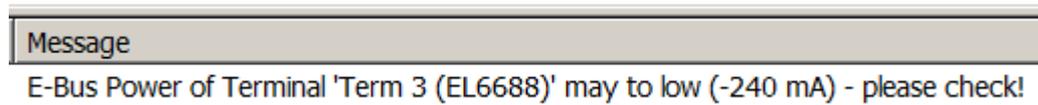


Fig. 135: Warning message for exceeding E-Bus current

<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 EL2202 - Basic function principles

### 6.1 Delivery state

No special measures are required for commissioning the EL2202-0000.

Current configuration files should be downloaded from [www.beckhoff.de](http://www.beckhoff.de) --> Download --> Configuration files --> EtherCAT XML Device Description, if they are not included in the TwinCAT version used.

### 6.2 Switching characteristics

The output circuit of the EL22xx is optimized for fast signal output, even at higher currents. To this end the EL22xx features a push/pull output stage that actively switches to 24 V and 0 V. The push stage is overcurrent- and short circuit-protected, the pull stage (switching to 0 V) is not. This may have to be taken into account when inductive loads are switched off. The time required by a signal change as a rising/falling edge from the central controller (ESC, EtherCAT slave controller) to the connection at the front of the terminal is specified as  $T_{ON}/T_{OFF} < 1 \mu s$  for the EL22xx range, both for a rising ( $T_{ON}$ ) and a falling edge ( $T_{OFF}$ ). Due to this short absolute signal cycle time the temperature drift during the cycle time is also very low.

The edge steepness of the output drivers is  $< 1 \mu s/24 V$ . The power supply of the EL22xx should be dimensioned according to the power requirements of the connected actuators. The power supply lines, power supply unit and actuator lines should be suitably short or generous respectively.

---

#### ● Measurement of time delay

**i** While optimized electronic components make the delay between the physically real edge at the ESC output and the edge at the terminal point very short ( $< 1 \mu s$ ), it is not negligible. If the user has a particular need for precision, calibration must be carried out, and the precise time delay measured in accordance with the environmental conditions.

---

### 6.3 Tristate mode (high-resistance outputs)

In Tristate mode each channel can be switched to high-resistance. This state ensures that the respective output behaves as if it was not connected, so that it does not influence the outputs of other outputs/devices connected in parallel. The associated output takes on the same output voltage as the other active devices.

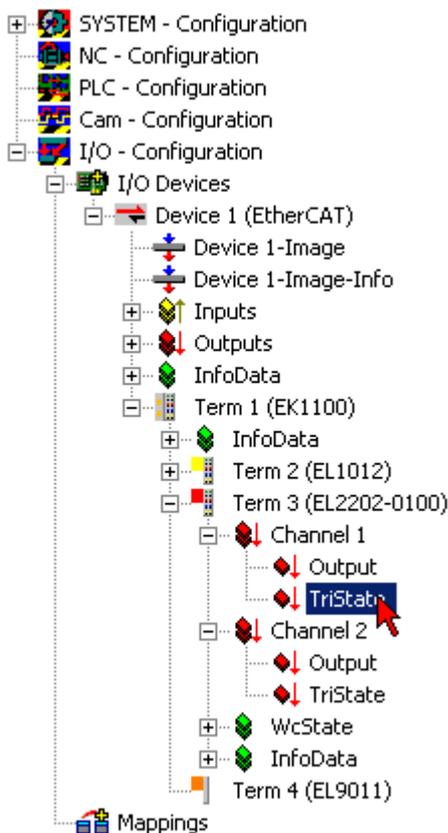


Fig. 136: EL2202 in the TwinCAT tree, Tristate mode

## 6.4 Watchdog

### ● Watchdog

For using the [watchdog](#) [▶ 23] in the EL2202 please refer to the [associated page](#) [▶ 23].

## 6.5 Operating modes

The EL2202 series offers two different operating modes:

### EL2202-0000 (delivery state)

Fast 2-channel 24 V output terminal - writing of outputs is triggered by the EtherCAT frame. The output data are written as soon as an EtherCAT frame has transferred data for the respective EL2202-0000 and this frame was verified as valid after it has passed through the EL2202-0000. This means that the output time depends on the low jitter of the EtherCAT frame. No other settings are required at the terminal.

#### Example:

Two EL2202-0000 are operated side by side. Both are cyclically supplied with output data by the same task every 1 ms. This means that the outputs from the second EL2202-0000 would always be a little later than the outputs of the first EL2202-0000. The difference arises from the cumulative cycle times of the EtherCAT frames through the EtherCAT slaves positioned between the two EL2202-0000 and the frame length in  $\mu\text{s}$ .

### EL2202-0100

Fast 2-channel 24 V input terminal - writing of outputs is triggered in a highly stable cycle by the local DC clock (distributed clock).

The output data are transferred to the EtherCAT frame once it was verified as valid. The data are only transferred to the power stage with the SYNC0 signal of the ESC, which is cyclically generated by the local

distributed clock. The output cycle of the EL2202-0100 is therefore totally independent of the EtherCAT jitter. The variability of the output cycle is within the range of the distributed clock accuracy ( $< 100$  ns). The task supplying the EL2202-0100 with process data determines the SYNC0 signal cycle of the EL2202-0100. Therefore, if several (PLC) tasks with different cycle times are operated on the *same* EtherCAT strand, the task supplying the EL2202-0100 with process data determines the output cycle. If two tasks running at different speeds each write to one EL2202-0100 channel, the faster cycle time determines the output cycle.

### **i** EL2202-0100 operating mode with distributed clocks

In contrast to the EL2202-0000 the EL2202-0100 can only be used with EtherCAT masters that support distributed clocks (DC) (e.g. Beckhoff TwinCAT). To change the mode the fail-safe configuration in the terminal has to be changed.

## 6.6 EL2202-0100 Advanced Settings

### **⚠ CAUTION**

#### **CAUTION! Risk of device damage!**

The entries in these dialogs are not monitored or corrected. Incorrect or invalid values can lead to malfunction of the terminal. The settings in this area should only be changed by persons who have the required basic knowledge!

The SYNC0 signal is cyclically generated internally in the ESC in the EL2202-0100 from the local distributed clock as described above and therefore determines the timing of the data output to the power stage. This SYNC0 interrupt can now be manipulated manually by the user based on an additive constant ("shift time"). This means that in an EL2202-0100 the SYNC0 signal can locally be shifted by a constant value relative to other terminals / EtherCAT devices. This setting cannot be implemented on a channel-by-channel basis, i.e. it applies on a terminal by terminal basis and therefore concerns all channels in an EL2202-0100.

#### **Example:**

Two EL2202-0100 are operated side by side. Both are cyclically supplied with output data every 1 ms by the same task and by default would therefore write their outputs at the *same time* (within the distributed clock accuracy, i.e.  $< 100$  ns difference). However, the aim is for the second EL2202-0100 to write its outputs 345  $\mu$ s later than the first EL2202-0100, as required by the material flow in the application, for example. To this end a constant shift time of 345  $\mu$ s for the SYNC0 signal is entered in the second EL2202-0100.

Beckhoff TwinCAT supports these configuration options through the following dialog (*Terminal 2202 -> "DC" tab -> Advanced Settings... -> Distributed Clock*) :

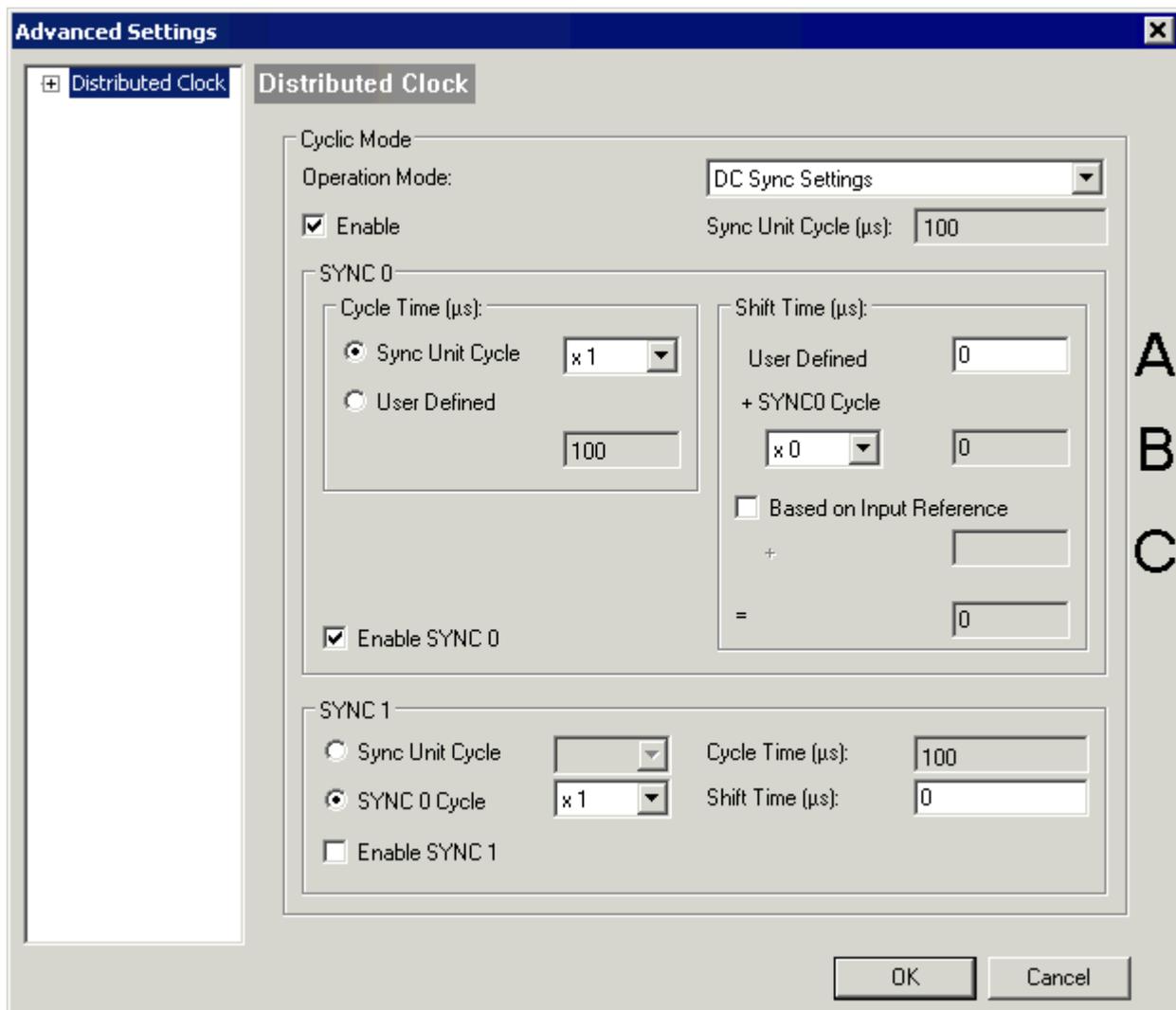


Fig. 137: Manipulation of the SYNC0 signal in the EL2202-0100

The user can enter a freely selected value (A), a multiple of the local SYNC0 Cycle Time (B), and an “Input reference” (C). Based on the example, a value of 345 should be entered manually at (A) (see Fig. *Manipulation of the SYNC0 signal in the EL2202-0100*).

### ⚠ CAUTION

#### CAUTION! Risk of device damage!

The entries in these dialogs are not monitored or corrected. Incorrect or invalid values can lead to malfunction of the terminal. The settings in this area should only be changed by persons who have the required basic knowledge!

## 6.7 Switching from EL2202-0000 to EL2202-0100 and back

To switch the EL2202 from one mode to the other the configuration has to be changed via a write access to the EEPROM in the terminal. In the Beckhoff TwinCAT System Manager this can be done conveniently via a dialog.

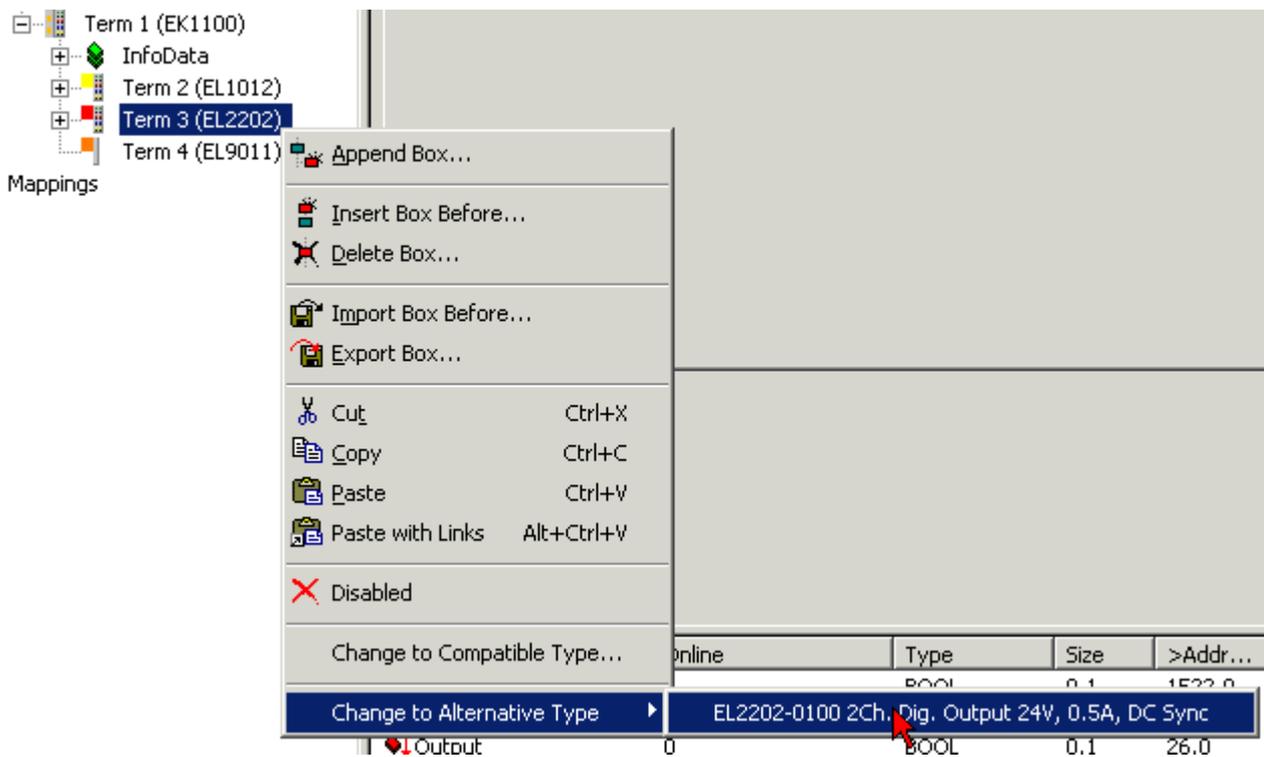


Fig. 138: EL2202 change dialog

A right-click on the EL2202 in TwinCAT configuration tree, allows change to an “Alternative Type”, and selection of the other EL2202.

**i Switching of operating mode EL2202-0000 -> EL2202-0100**

During the write access (several seconds) the power supply for the EL2202 and the EtherCAT communication must not be interrupted. The EL2202 then has to be restarted.

## 7 EL2252 - Basic function principles

### Delivery state

No special measures are required for commissioning the EL2252.

#### ● EtherCAT XML Device Description



If the XML description of the EL2252 is not available in your system you can download the latest XML file from the download area of the [Beckhoff website](#) and install it according to the installation instructions.

This chapter comprises the following sections:

- [Switching characteristics \[▶ 124\]](#)
- [Functioning \[▶ 124\]](#)
- [Application - brief overview \[▶ 125\]](#)
- [Detailed description of application with TwinCAT \[▶ 125\]](#)
- [Example Program \[▶ 129\]](#)

### 7.1 Switching characteristics

The output circuit of the EL22xx is optimized for fast signal output, even at higher currents. To this end the EL22xx features a push/pull output stage that actively switches to 24 V and 0 V. The push stage is overcurrent- and short circuit-protected, the pull stage (switching to 0 V) is not. This may have to be taken into account when inductive loads are switched off. The time required by a signal change as a rising/falling edge from the central controller (ESC, EtherCAT slave controller) to the connection at the front of the terminal is specified as  $T_{ON}/T_{OFF} < 1 \mu s$  for the EL22xx range, both for a rising ( $T_{ON}$ ) and a falling edge ( $T_{OFF}$ ). Due to this short absolute signal cycle time the temperature drift during the cycle time is also very low.

The edge steepness of the output drivers is  $< 1 \mu s/24 V$ . The power supply of the EL22xx should be dimensioned according to the power requirements of the connected actuators. The power supply lines, power supply unit and actuator lines should be suitably short or generous respectively.

In Tristate mode each channel can be switched to high-resistance.

#### ● Measurement of time delay



While optimized electronic components make the delay between the physically real edge at the ESC output and the edge at the terminal point very short ( $< 1 \mu s$ ), it is not negligible. If the user has a particular need for precision, calibration must be carried out, and the precise time delay measured in accordance with the environmental conditions.

### 7.2 Watchdog

#### ● Watchdog



For using the [watchdog \[▶ 23\]](#) in the EL2252 please refer to the [associated page \[▶ 23\]](#).

### 7.3 Functioning

All processes in the EL2252 are controlled by the SYNC0 pulse of the EtherCAT slave controller (ESC). In this EL2252 the ESC is a special IC that handles the EtherCAT communication. In contrast to many other applications, in the EL2252 the SYNC0 signal is not used cyclically free-running, but the process of writing data to the outputs is triggered when it occurs for the first time. To this end the user specifies (via the process data) the start time of SYNC0 and the required output states in which the EL2252 is written to and activated.

The ESC features an integrated local clock (distributed clock) that is continuously synchronized system-wide by the EtherCAT master with the master itself and with all other distributed clocks in the EtherCAT strand with a relative tolerance of < 100 ns.

The ESC in the EL2252 continuously and independently compares the start time with the current local distributed clock time. If the times match it immediately triggers the SYNC0 signal, which in turn internally leads to writing of the required output states.

When fully configured the distributed clock time is a 64 bit value with a resolution of 1 ns and a start time of 1.1.2000, 00:00 hours. When the EtherCAT strand starts up the EtherCAT master takes the time from the local PC and transfers it to the EtherCAT distributed clock network.



### EtherCAT and Distributed Clocks

A basic introduction to the subject of EtherCAT and Distributed Clocks can be downloaded from the [Beckhoff website](#): the "Distributed Clocks system description".

## 7.4 Application - brief overview

The EL2252 must be used based on the following procedure:

- Determining the current distributed clock time in 64 bit format, e.g. through the active [SysTime \[▶ 126\]](#) of the EL2252
- Calculation of the required [StartTime \[▶ 126\]](#) (output time) with 64-bit-operations
- process data transfer:
  - [StartTime \[▶ 126\]](#)
  - Output data [Output \[▶ 126\]](#) and [Tristate \[▶ 126\]](#) of channel 1 and 2
  - At the same time set [Activate \[▶ 126\]](#) = 0
- In the following cycle set [Activate \[▶ 126\]](#) = 3; this activates the start time in the terminal  
 The EL2252 now waits as described above until the start time occurs and then send the output data to the power stage  
 Until the switching event occurs, [Activate \[▶ 126\]](#) and the output bits [Output \[▶ 126\]](#)/[Tristate \[▶ 126\]](#) must be maintained in accordance with the target state  
 In the event of a voltage loss in the terminal the start time is lost and must be set again by the controller

Therefore, only *one* switching event can be defined for each EL2252, which affects all 4 output variables *Output* and *Tristate* of channel 1 and 2 at the same time. For each EtherCAT cycle only one switching event can be defined.

Once a switching task has been issued it cannot be cancelled. However, since a switching task that has already been issued can always be overwritten by a new one. Switching tasks that have already been issued can effectively be cancelled through overwriting with an invalid switching task (e.g. time in the past).

The selectable PDO "Feedback" can be used to check the success of the switching task output. It maps the connected outputs bit by bit, and the tristate circuit in a byte:

- Bit 0: Output channel 1
- Bit 1: Tristate channel 1
- Bit 2: Output channel 2
- Bit 3: Tristate channel 2

## 7.5 Detailed description of application with TwinCAT

### Process data

After reading/manual insertion in the System Manager, the EL2252 appears as follows (Fig. Default representation of the EL2252):

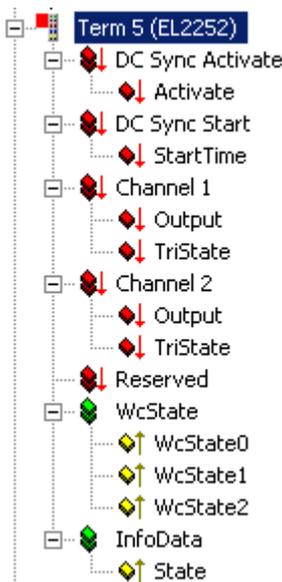


Fig. 139: Default representation of the EL2252

### Description of the process data

- **Activate**  
This byte activates a new start time in the terminal through the transition 0 --> 3
- **StartTime**  
64-bit value of the next desired switching event
- **Output**  
Output 0 V/24 V of channel 1 or 2
- **Tristate**  
Switch channel 1 or 2 to high-resistance
- **WcState operational information**  
Each datagram of the EL2252 indicates its processing state here. This allows the EL2252 to be monitored for correct process data communication.
- **State operational information**  
Status display of the “EtherCAT state machine
- **SysTime**  
64 bit system time, read out of the terminal. Available since revision 0016.  
The PDO system time can't be transferred simultaneously with the PDO Feedback. Generally it's recommended to determine the current system time via an alternative way, e.g. by the EtherCAT Master or dedicated Function Blocks (FB) of the EtherCAT library (Tc2\_EtherCAT.lib).
- **Feedback**  
1 byte feedback contains the current output state (output and tristate) of the terminal. Available since revision 0018.  
The PDO Feedback can't be transferred simultaneously with the PDO system time.

### ● Data type with 64 bit width

**i** A data type with a width of 64 bit is defined in TcEthercat.lib as T\_DCTIME or in TcUtilities.lib as T\_LARGE\_INTEGER.

The process data of the EL2252 can be extended with an input variable representing the local distributed clock time in this EL2252. To this end select the bottom option, i.e. “Inputs” (SM3), in the “Process Data” tab of the EL2252 in the “Sync Manager” field. In the “PDO Assignment” field below, the entry 0x1A00 can now be activated (Fig. *Activation of the input variable SysTime*). After reloading of the configuration (F4 key) or start in config mode distributed clock time of the EL2252 is now cyclically displayed as *SysTime* under the process data.

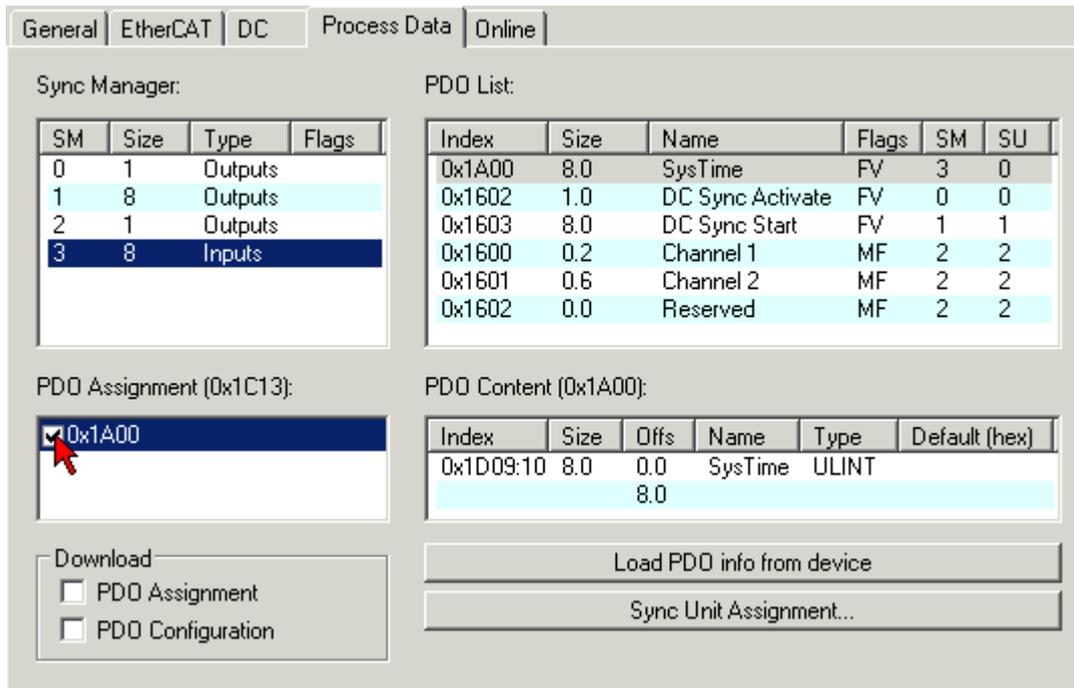


Fig. 140: Activation of the SysTime input variable

**i** **EL2252-0000-0018**

If TwinCAT is configured offline with the EL2252-0000-0018 or through scanning a corresponding EL2252, the PDO feedback is also available as a byte. This can be used to control the triggering of the switching order (see [sample programs \[► 129\]](#)).

This SysTime variable can now be linked cyclically as a 64-bit process data with the higher-level controller (Fig. *EL2252 process image in full configuration*).

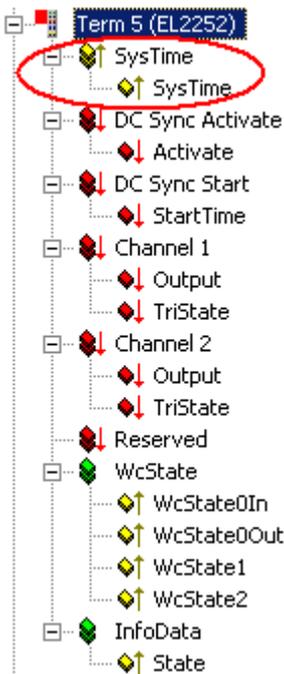


Fig. 141: EL2252 process image in full configuration

## Determining a current distributed clock time

If a PLC cycle requires the current distributed clock time, this can be obtained via an input variable of the EtherCAT master or directly from the EL2252. These methods are basically equivalent, since all local clocks in all slaves maintain the same distributed clock time (within the specified accuracy) based on the distributed clock synchronization within the EtherCAT network.

### ● Functions for data types with 64-bit width



A selection of functions for handling 64-bit numbers is available under Beckhoff TwinCAT in the TcUtilities.lib library. Longer execution times are required here than is the case with standard, 32-bit data types.

A data type with a width of 64 bit is defined in TcEthercat.lib as T\_DCTIME or in TcUtilities.lib as T\_LARGE\_INTEGER.

**Transfer from EL2252:** The *SysTime* process data has to be activated in the EL2252 as specified above and can then be linked with the PLC. The value can also be obtained from another EtherCAT slave that makes its local distributed clock time available as a process data, e.g. an EL1252.

**Transfer from the EtherCAT master:** The EtherCAT master can display a copy of the current master distributed clock. To this end activate the setting "Show DC System Time (64 bit)" in the *EtherCAT device* --> *EtherCAT tab* --> *Advanced Settings* --> *Distributed Clocks* (Fig. *Activation of the master distributed clock display*).

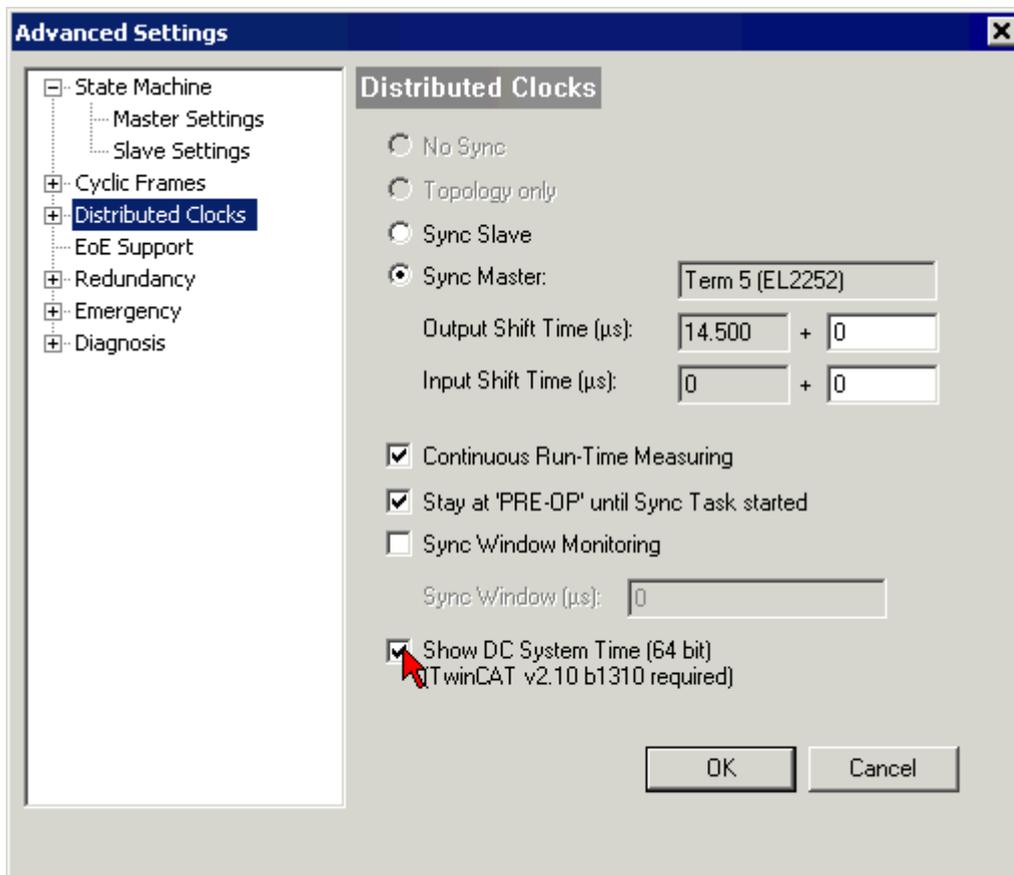


Fig. 142: Activation of the master distributed clock display

The process image of the EtherCAT master now looks as shown in Fig. *Extended process image of the EtherCAT master*.

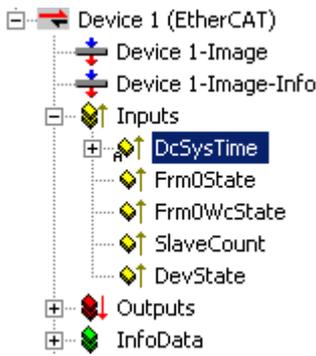


Fig. 143: Extended process image of the EtherCAT master

**i** **SYSTIME**

The SysTime value of a terminal or the DcSysTime value of an EtherCAT master should be used advisedly and only as a rough guideline for the current time range (order of magnitude: 1-2 task cycles) of the distributed clock system. This is due to the fact that the local time in the terminal is read into associated buffers several  $\mu$ s before retrieving frame. The EtherCAT strand then requires some time for transporting the process data to the master. The accuracy of the online display in the TwinCAT System Manager (see Fig. *Extended process image of the EtherCAT master*) is in the 3-figure millisecond range at best due to the visualization character. Within a PLC the SysTime process data may also be outdated to a certain degree: depending on when the analysis takes place in the program code, the process data may be several  $\mu$ s/ms older. If responses of an EtherCAT strand (e.g. an EL2252 output terminal) are to be based on a SysTime read in this way, 2-3 PLC cycles should be taken into account as a buffer. It is more appropriate to use an EL2252 with other EtherCAT slaves that generate a time stamp based on ambient influences, such as the EL1252.

**Calculation of the required output time**

**i** **Functions for data types with 64-bit width**

A selection of functions for handling 64-bit numbers is available under Beckhoff TwinCAT in the TcUtilities.lib library. Longer execution times are required here than is the case with standard, 32-bit data types. A data type with a width of 64 bit is defined in TcEthercat.lib as T\_DCTIME or in TcUtilities.lib as T\_LARGE\_INTEGER.

The switching time of the EL2252 must be at least 2 task cycles in the future, because otherwise the process data sequence cannot be adhered to.

**Process data transfer sequence**

The process data of the EL2252 have to be transferred in two cycles:

Cycle n: *StartTime* based on calculation, *Activate* = 0 and output bits *Output/Tristate* based on target state

Cycle n+1: *Activate* = 3

Cycle n+2 and following: *Activate* = 3 and output bits to be held until the target state is reached. A new switching time with associated target state can then be transferred.

**Sample Program**

A sample program demonstrates the use of EL2252 and EL1252 for outputting pulses. It includes a version with feedback and version without feedback.

## 8 Example programs

### **i** Using the sample programs

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

#### Example 1: Output and evaluation of pulses

In this sample program an EL2252 generates periodically different pulses on both channels, which can be measured in real-time with an EL1252.

Connection diagram:

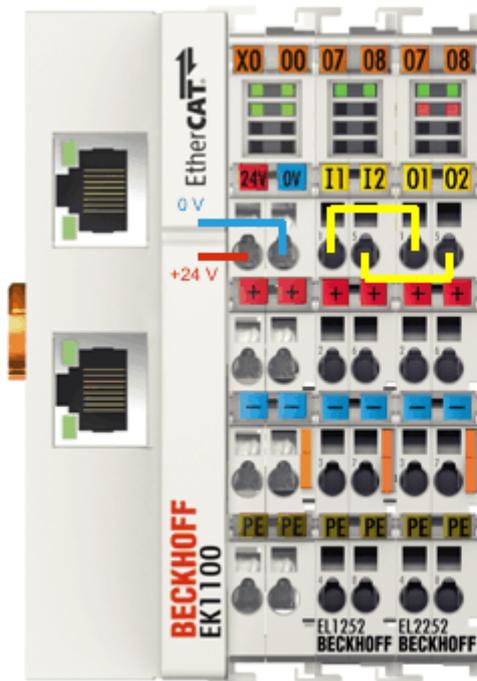


Fig. 144: Wiring for example program 1

#### Starting the example program

The application examples have been tested with a test configuration and are described accordingly. Certain deviations when setting up actual applications are possible.

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

- TwinCAT 2
  - TwinCAT master PC with Windows XP Professional SP 3, TwinCAT version 2.10 (Build 1335) and INTEL PRO/100 VE Ethernet adapter
- TwinCAT 3
  - TwinCAT-Master-PC with Windows 7 (64 Bit) SP 1, TwinCAT version 3.1 (Build 4022) and TwinCAT-Intel PCI Ethernet adapter (Gigabit)
- Beckhoff EK1100 EtherCAT coupler, EL1252, EL2252 and EL9011 terminals.

## 8.1 Sample program TwinCAT 2

 [https://infosys.beckhoff.com/content/1033/el2202\\_el2252/Resources/4318611851.zip](https://infosys.beckhoff.com/content/1033/el2202_el2252/Resources/4318611851.zip)

Please continue with further descriptions in section [TwinCAT Quickstart](#), [TwinCAT 2.TwinCAT Quickstart](#), [TwinCAT 2 \[▶ 53\]](#).

### Procedure for starting the program

- After clicking the Download button, save the zip file locally on your hard disk, and unzip the \*.TSM (configuration) and the \*.PRO (PLC program) files into a temporary working folder.
- The \*.pro file can be opened by double click or by the TwinCAT PLC Control application with menu selection “File/ Open”. The \*.tsm file is provided for the TwinCAT System Manager (to review or overtake configurations).
- Connect the hardware in accordance with [fig. Connection for sample program \[▶ 130\]](#) and connect the Ethernet adapter of your PC to the EtherCAT coupler (further information on this can be found in the corresponding coupler manuals)
- Select the local Ethernet adapter (with real-time driver, if applicable) under System configuration, I/O configuration, I/O devices, Device (EtherCAT); then on the “Adapter” tab choose “Search...”, select the appropriate adapter and confirm (see [Fig. Searching the Ethernet adapter + Selection and confirmation of the Ethernet adapter](#)).

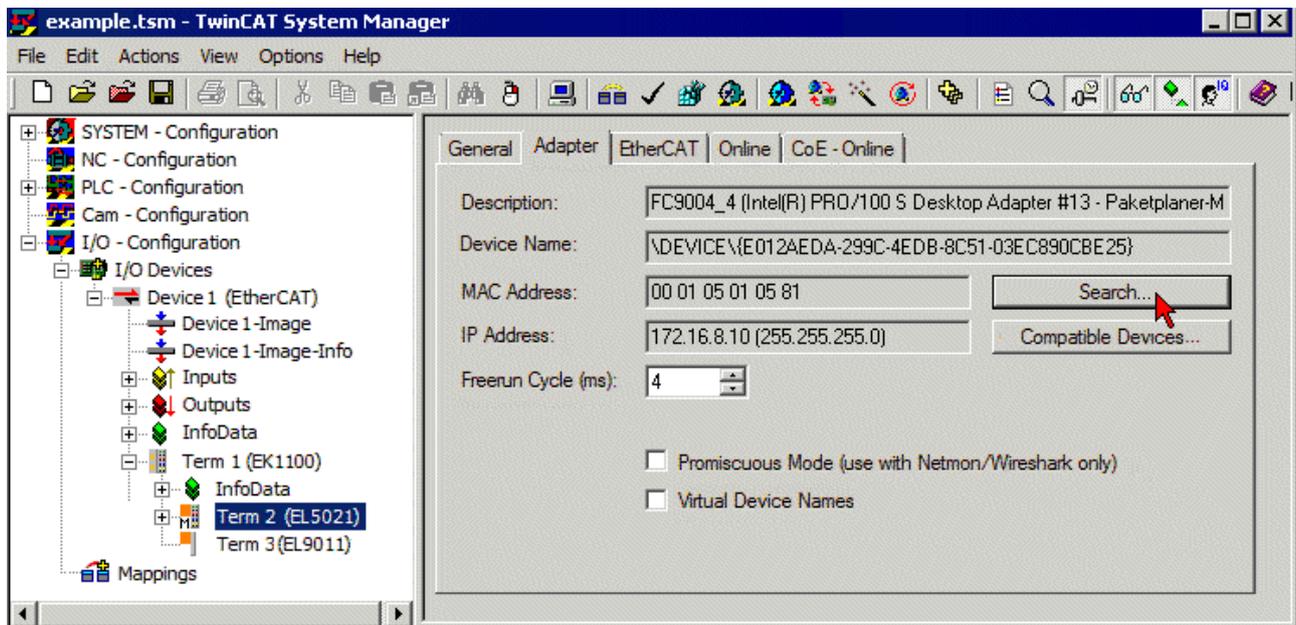


Fig. 145: Searching the Ethernet adapter

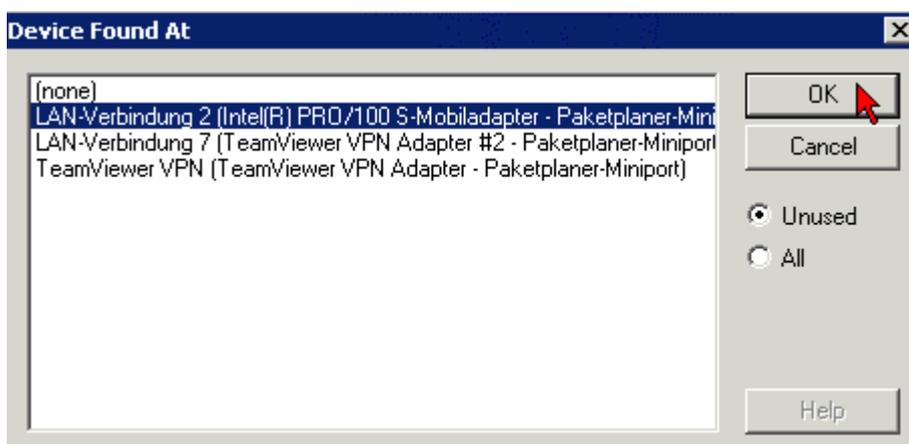


Fig. 146: Selection and confirmation of the Ethernet adapter

- Activate and confirm the configuration (Fig. *Activation of the configuration + Confirming the activation of the configuration*)



Fig. 147: Activation of the configuration



Fig. 148: Confirming the activation of the configuration

- Confirm new variable mapping, restart in RUN mode (Fig. *Generate variable mapping + Restarting TwinCAT in RUN mode*)

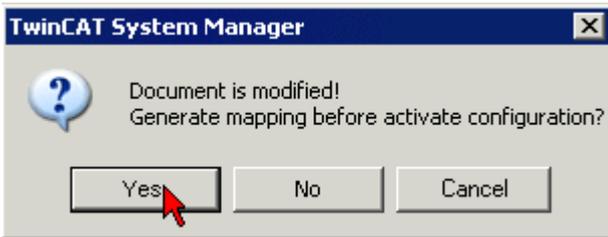


Fig. 149: Generating variable mapping



Fig. 150: Restarting TwinCAT in RUN mode

- In TwinCAT PLC, under the “Project” menu, select “Rebuild all” to compile the project (Fig. *Compile project*)

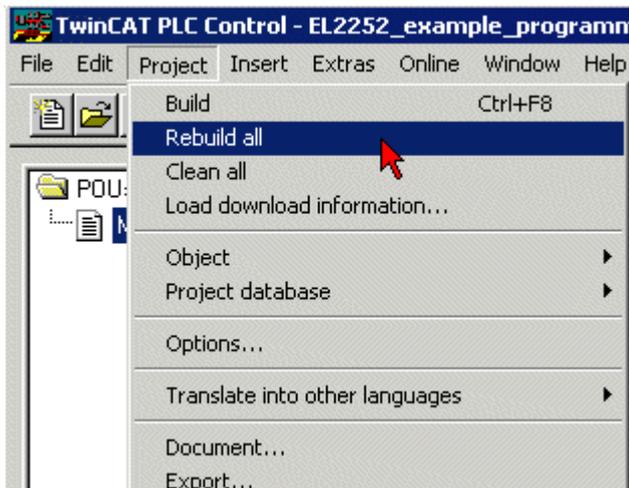


Fig. 151: Compile project

- In TwinCAT PLC: log in with the "F11" button, confirm loading the program (Fig. *Confirming program start*), run the program with the "F5" button

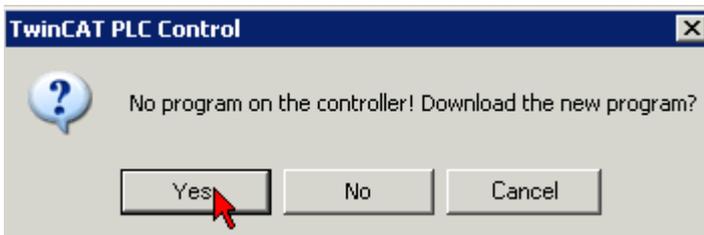


Fig. 152: Confirming program start

## 8.2 Sample program TwinCAT 3

 [https://infosys.beckhoff.com/content/1033/el2202\\_el2252/Resources/5510184331.zip](https://infosys.beckhoff.com/content/1033/el2202_el2252/Resources/5510184331.zip)

### Preparations for starting the sample programs (tzip file / TwinCAT 3)

- Click on the download button to save the Zip archive locally on your hard disk, then unzip the \*.tzip archive file in a temporary folder.

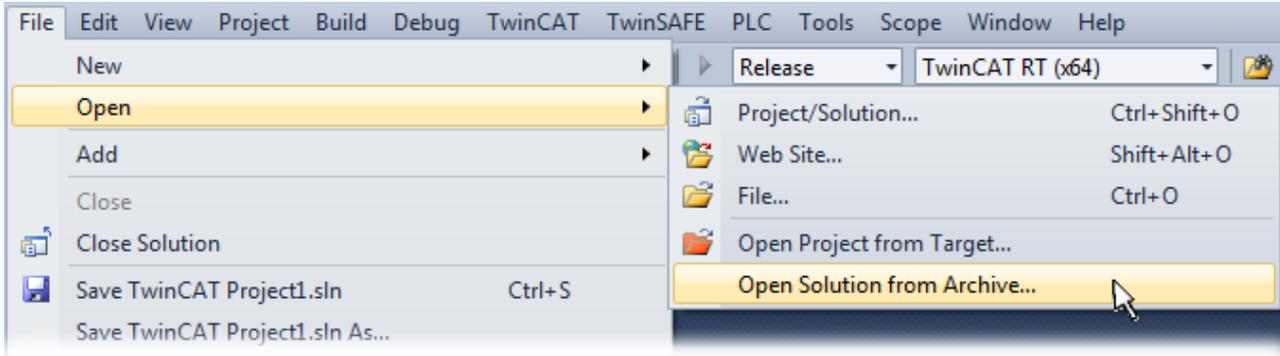


Fig. 153: Opening the \*. tzip archive

- Select the .tzip file (sample program).
- A further selection window opens. Select the destination directory for storing the project.
- For a description of the general PLC commissioning procedure and starting the program please refer to the terminal documentation or the EtherCAT system documentation.
- The EtherCAT device of the example should usually be declared your present system. After selection of the EtherCAT device in the “Solutionexplorer” select the “Adapter” tab and click on “Search...”:

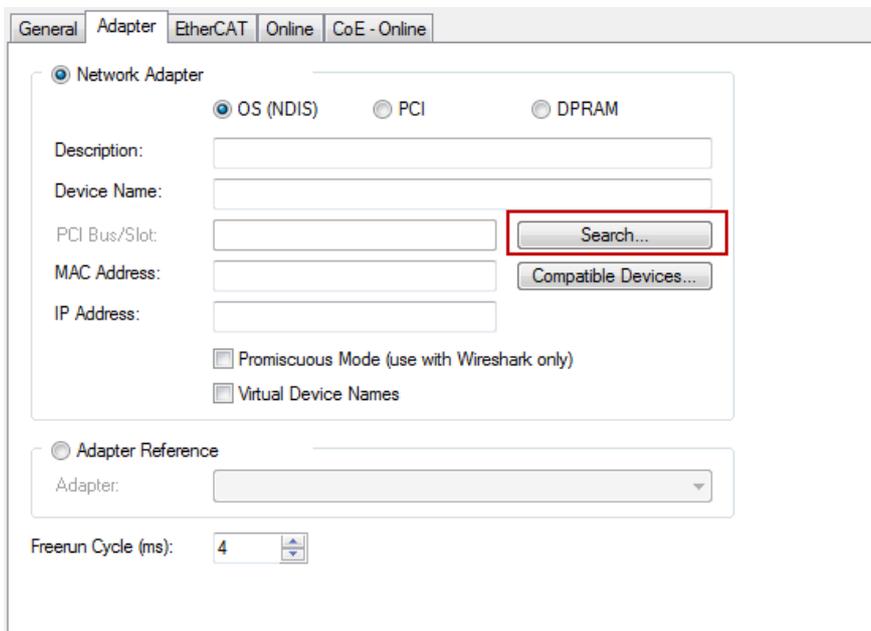


Fig. 154: Search of the existing HW configuration for the EtherCAT configuration of the example

- Checking NetId: the “EtherCAT” tab of the EtherCAT device shows the configured NetId:



The first four numbers must be identical with the project NetId of the target system. The project NetId

can be viewed within the TwinCAT environment above, where a pull down menu can be opened to choose a target system (by clicking right in the text field). The number blocks are placed in brackets there next to each computer name of a target system.

- Modify the NetId: By right clicking on “EtherCAT device” within the solution explorer a context menu opens where “Change NetId...” have to be selected. The first four numbers of the NetId of the target computer must be entered; both last values are 4.1 usually.

Example:

- NetId of project: myComputer (123.45.67.89.1.1)
- Entry via „Change NetId...“: 123.45.67.89.4.1

Also see more hints in section:

[Commissioning, TwinCAT Quickstart, TwinCAT 3, Startup](#) [► 63].

## 9 Appendix

### 9.1 EtherCAT AL Status Codes

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

### 9.2 Firmware compatibility

The terminals of the EL2202 and EL2252 series have no firmware.

### 9.3 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*).

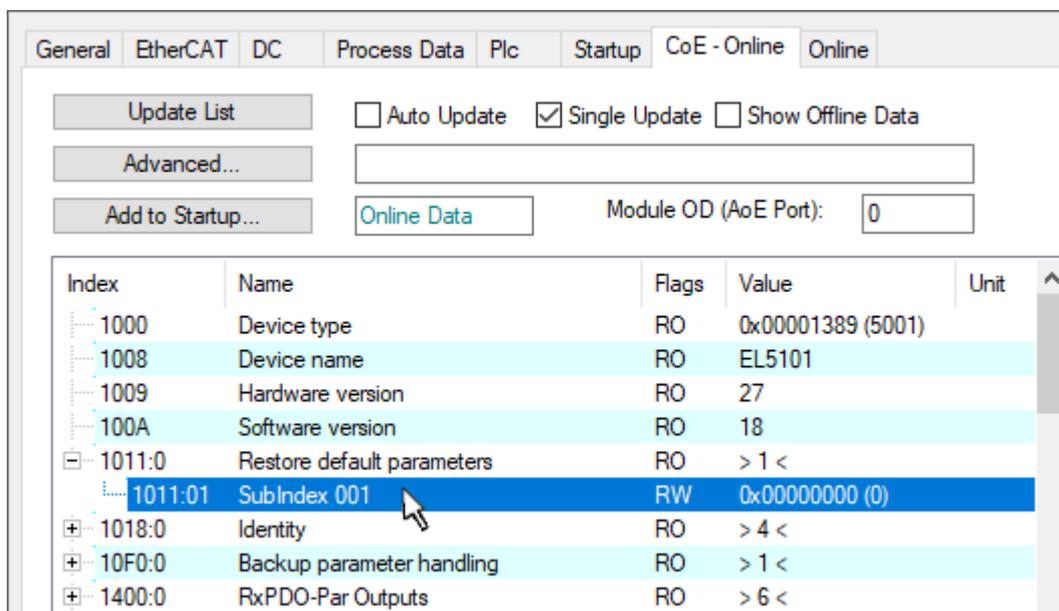


Fig. 155: Selecting the *Restore default parameters* PDO

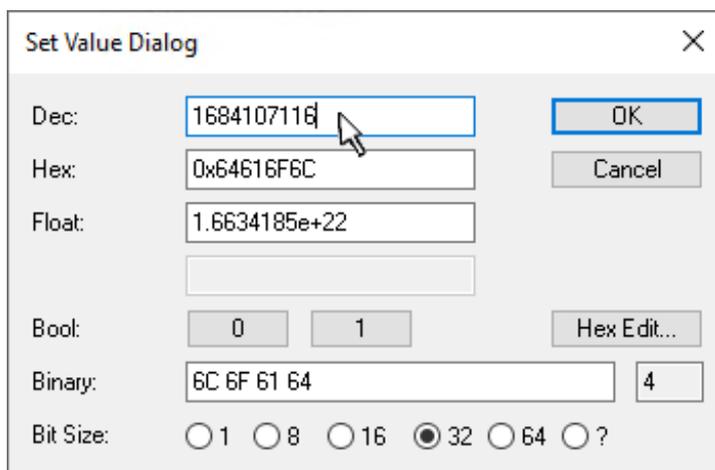


Fig. 156: Entering a restore value in the Set Value dialog

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.

---

## 9.4 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)

## **Trademark statements**

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.

## **Third-party trademark statements**

DeviceNet and EtherNet/IP are trademarks of ODVA, Inc.

DSP System Toolbox, Embedded Coder, MATLAB, MATLAB Coder, MATLAB Compiler, MathWorks, Predictive Maintenance Toolbox, Simscape, Simscape™ Multibody™, Simulink, Simulink Coder, Stateflow and ThingSpeak are registered trademarks of The MathWorks, Inc.

Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

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

More Information:  
[www.beckhoff.com/EL2xxx](http://www.beckhoff.com/EL2xxx)

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)

