

**EtherNet/IP™**

**Documentation**

**EL6652-00x0**

**EtherNet/IP Master/Slave EtherCAT Terminal**

**Version: 2.5**  
**Date: 2019-10-22**

**BECKHOFF**



# Table of contents

<b>1 Foreword</b> .....	<b>5</b>
1.1 Notes on the documentation.....	5
1.2 Safety instructions .....	6
1.3 Documentation issue status .....	7
1.4 Version identification of EtherCAT devices .....	8
1.4.1 Beckhoff Identification Code (BIC).....	12
<b>2 Product overview</b> .....	<b>14</b>
2.1 EL6652-0000, EL6652-0010 - Introduction .....	14
2.2 EL6652-0000, EL6652-0010 - Technical data .....	16
<b>3 Basics communication</b> .....	<b>17</b>
3.1 EtherCAT basics.....	17
3.2 EtherCAT cabling – wire-bound.....	17
3.3 General notes for setting the watchdog.....	18
3.4 EtherCAT State Machine .....	20
3.5 CoE Interface.....	22
3.6 Distributed Clock .....	27
<b>4 Installation</b> .....	<b>28</b>
4.1 Instructions for ESD protection .....	28
4.2 Recommended mounting rails .....	28
4.3 Mounting and demounting - terminals with front unlocking .....	28
4.4 Positioning of passive Terminals .....	31
4.5 Installation positions .....	31
4.6 UL notice .....	33
<b>5 Commissioning</b> .....	<b>34</b>
5.1 TwinCAT Quick Start .....	34
5.1.1 TwinCAT 2 .....	37
5.1.2 TwinCAT 3 .....	47
5.2 TwinCAT Development Environment .....	59
5.2.1 Installation of the TwinCAT real-time driver.....	59
5.2.2 Notes regarding ESI device description.....	65
5.2.3 TwinCAT ESI Updater .....	69
5.2.4 Distinction between Online and Offline.....	69
5.2.5 OFFLINE configuration creation .....	70
5.2.6 ONLINE configuration creation .....	75
5.2.7 EtherCAT subscriber configuration.....	83
5.3 Basic function principles .....	93
5.4 Changing EtherNet/IP settings .....	94
5.5 EL6652-0000 Master (Scanner) .....	96
5.5.1 EL 6652-0000 configuration .....	96
5.5.2 EL6652-0000 - Configuration parameters .....	103
5.5.3 EL6652-0000 EDS file .....	104
5.6 EL6652-0010 Slave .....	110
5.6.1 EL 6652-0010 configuration .....	110

5.6.2	EL6652-0010 - Configuration parameters .....	115
5.6.3	Master (Scanner) configuration .....	116
<b>6</b>	<b>Configuration with the Twincat System Manager.....</b>	<b>121</b>
6.1	Object description and parameterization .....	121
<b>7</b>	<b>Diagnosis.....</b>	<b>126</b>
7.1	EL6652-0010 - LEDs .....	126
7.2	EL6652-0000, EL6652-0010 diagnostic history.....	128
<b>8</b>	<b>Appendix .....</b>	<b>130</b>
8.1	Firmware Update EL/ES/EM/ELM/EPxxxx .....	130
8.1.1	Device description ESI file/XML.....	131
8.1.2	Firmware explanation .....	134
8.1.3	Updating controller firmware *.efw.....	135
8.1.4	FPGA firmware *.rbf.....	136
8.1.5	Simultaneous updating of several EtherCAT devices.....	140
8.2	Firmware compatibility .....	141
8.3	Restoring the delivery state .....	141
8.4	Support and Service .....	142

# 1 Foreword

## 1.1 Notes on the documentation

### Intended audience

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

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

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

### Disclaimer

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

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

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

### Trademarks

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

### Patent Pending

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

The logo for EtherCAT, featuring the word "EtherCAT" in a bold, black, sans-serif font. A red arrow points from the top of the "A" towards the right, ending above the "T". A registered trademark symbol (®) is located to the right of the "T".

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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

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

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

## 1.2 Safety instructions

### Safety regulations

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

### Exclusion of liability

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

### Personnel qualification

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

### Description of instructions

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

#### **DANGER**

##### **Serious risk of injury!**

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

#### **WARNING**

##### **Risk of injury!**

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

#### **CAUTION**

##### **Personal injuries!**

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

#### **NOTE**

##### **Damage to environment/equipment or data loss**

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



##### **Tip or pointer**

This symbol indicates information that contributes to better understanding.

## 1.3 Documentation issue status

Version	Comment
2.5	<ul style="list-style-type: none"><li>• Update revision status</li><li>• Update structure</li></ul>
2.4	<ul style="list-style-type: none"><li>• Update chapter "Changing EthetNet/IP Setting"</li><li>• Update revision status</li><li>• Update structure</li></ul>
2.3	<ul style="list-style-type: none"><li>• Update Technical data</li><li>• Addenda chapter "Instructions for ESD protection"</li><li>• Addenda chapter "UL notice"</li></ul>
2.2	<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><li>• Update revision status</li></ul>
2.1	<ul style="list-style-type: none"><li>• Update Technical data</li></ul>
2.0	<ul style="list-style-type: none"><li>• Addenda and corrections, Public issue</li></ul>
1.0.1	<ul style="list-style-type: none"><li>• Addenda and corrections</li></ul>
1.0	<ul style="list-style-type: none"><li>• Addenda</li></ul>
0.1	<ul style="list-style-type: none"><li>• Preliminary documentation for EL6652</li></ul>

## 1.4 Version identification of EtherCAT devices

### Designation

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

- family key
- type
- version
- revision

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

### Notes

- The elements mentioned above result in the **technical designation**. EL3314-0000-0016 is used in the example below.
- EL3314-0000 is the order identifier, in the case of "-0000" usually abbreviated to EL3314. "-0016" is the EtherCAT revision.
- The **order identifier** is made up of
  - family key (EL, EP, CU, ES, KL, CX, etc.)
  - type (3314)
  - version (-0000)
- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.  
In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.  
Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.  
From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. "EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)".
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

### Identification number

Beckhoff EtherCAT devices from the different lines have different kinds of identification numbers:

#### Production lot/batch number/serial number/date code/D number

The serial number 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

Ser. no.: 12063A02: 12 - production week 12 06 - production year 2006 3A - firmware version 3A 02 - hardware version 02

Exceptions can occur in the **IP67 area**, where the following syntax can be used (see respective device documentation):

Syntax: D ww yy x y z u

D - prefix designation

ww - calendar week

yy - year

x - firmware version of the bus PCB

y - hardware version of the bus PCB

z - firmware version of the I/O PCB

u - hardware version of the I/O PCB

Example: D.22081501 calendar week 22 of the year 2008 firmware version of bus PCB: 1 hardware version of bus PCB: 5 firmware version of I/O PCB: 0 (no firmware necessary for this PCB) hardware version of I/O PCB: 1

### Unique serial number/ID, ID number

In addition, in some series each individual module has its own unique serial number.

See also the further documentation in the area

- IP67: [EtherCAT Box](#)
- Safety: [TwinSafe](#)
- Terminals with factory calibration certificate and other measuring terminals

### Examples of markings



Fig. 1: EL5021 EL terminal, standard IP20 IO device with serial/ batch number and revision ID (since 2014/01)



Fig. 2: EK1100 EtherCAT coupler, standard IP20 IO device with serial/ batch number



Fig. 3: CU2016 switch with serial/ batch number

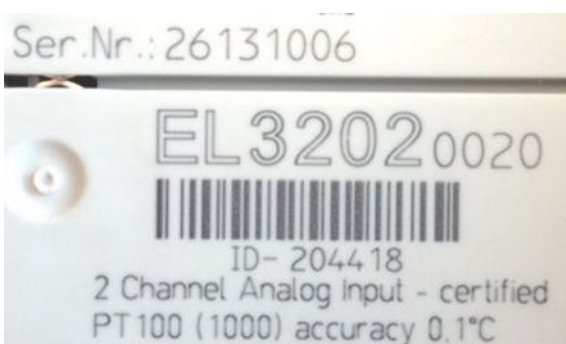


Fig. 4: EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418



Fig. 5: EP1258-0001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102

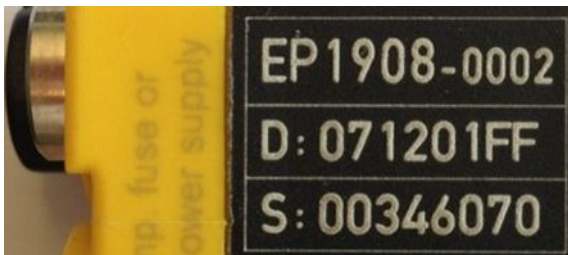


Fig. 6: EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070



Fig. 7: EL2904 IP20 safety terminal with batch number/ date code 50110302 and unique serial number 00331701



Fig. 8: ELM3604-0002 terminal with unique ID number (QR code) 100001051 and serial/ batch number 44160201

### 1.4.1 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.

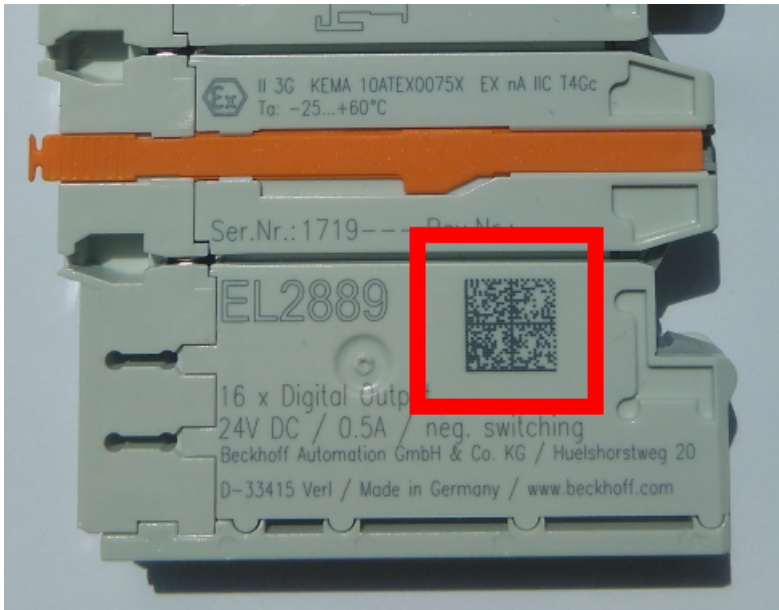


Fig. 9: BIC as data matrix code (DMC, code scheme ECC200)

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. The data under positions 1 to 4 are always available.

The following information is contained:

Item no.	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>	S	12	<b>S</b> BTNk4p562d7
3	Article description	<b>Beckhoff article description, e.g. EL1008</b>	1K	32	<b>1K</b> EL1809
4	Quantity	<b>Quantity in packaging unit, e.g. 1, 10, etc.</b>	Q	6	<b>Q</b> 1
5	Batch number	Optional: Year and week of production	2P	14	<b>2P</b> 401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	<b>51S</b> 678294104
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	<b>30P</b> F971, 2*K183
...					

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

**Structure of the BIC**

Example of composite information from item 1 to 4 and 6. The data identifiers are marked in red for better display:

**BTN**

An important component of the BIC is the Beckhoff Traceability Number (BTN, item no. 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.

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

## 2 Product overview

### 2.1 EL6652-0000, EL6652-0010 - Introduction

#### Ethernet/IP Master

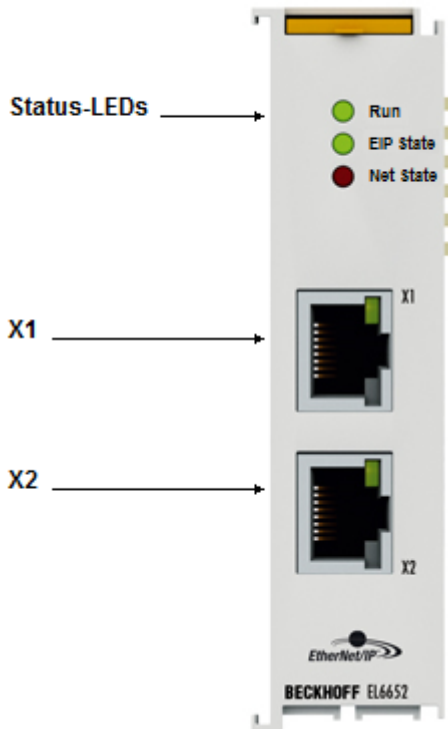


Fig. 10: EL6652-0000

The EL6652 EtherNet/IP-Master Terminal has a switched 2-port Ethernet connection and can thus be operated in a line with further Ethernet/IP nodes. The process data are configured by an EtherCAT master, allowing different process data and different sizes. The EL6652 supports both multicast and unicast connections. Up to 16 simple EtherNet/IP-Slave devices without configuration data can be connected via one "generic node". The import of EDS files is not supported (see [EDS file \[▶ 104\]](#)). The maximum process data size in both directions (input and output) is 1 kbyte.

#### Quick links

- [EtherCAT function principles](#)
- [Configuration \[▶ 96\]](#)
- [Parameter \[▶ 103\]](#)
- [Object description and parameterization \[▶ 121\]](#)

## Ethernet/IP Slave

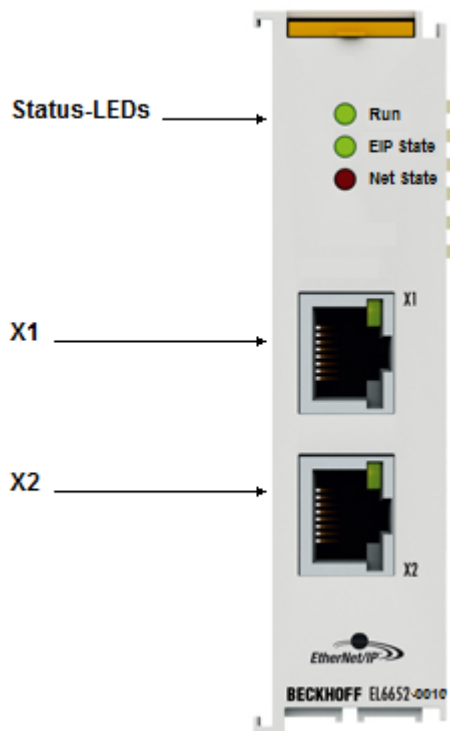


Fig. 11: *EL6652-0010*

The EtherNet/IP Slave terminal enables the exchange of data with an EtherNet/IP scanner or a master. Both Multicast and Unicast are supported. The terminal can behave like two EtherNet/IP slave devices. The second slave is a virtual slave. This enables you to connect two masters together or one master with two slaves and, for example, to transport more data or to operate with different polling times on the master. The maximum process data size in both directions (input and output) is 1 kbyte.

### Quick links

- [EtherCAT function principles](#)
- [Configuration \[► 110\]](#)
- [Parameter \[► 115\]](#)
- [Object description and parameterization \[► 121\]](#)

## 2.2 EL6652-0000, EL6652-0010 - Technical data

Technical data	EL6652	EL6652-0010
Description	Master (Scanner)	Slave
Number of possible slave devices	16 slaves	-
Number of ports/channels	2 (switched)	
Ethernet interface	100 MB Ethernet with 2 x RJ45	
Data width	max. 1 kByte input and 1 kByte output data	
Cable length	up to 100 m Twisted-Pair	
Hardware diagnosis	Status LEDs	
Housing	24 mm housing	
Power supply	via the E-Bus	
Current consumption from E-Bus	typ. 400 mA	
Electrical isolation	500 V (E-Bus/Ethernet)	
Configuration	via TwinCAT System Manager - from TwinCAT 2.11 Build 2248 - from TwinCAT 3.1 Build 4018.5	
Weight	ca. 75 g	
Permissible relative humidity	95%, no condensation	
Dimensions (W x H x D)	approx. 24 mm x 100 mm x 70 mm	
Mounting [ <a href="#">▶ 28</a> ]	on 35 mm mounting rail according to EN 60715	
Vibrations- / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27	
Permissible ambient temperature range during operation	- 0°C ... +55°C ( <a href="#">aligned in horizontal installation position ▶ 31</a> ) - 0°C ... +45°C ( <a href="#">all other installation positions ▶ 31</a> )	
Permissible ambient temperature range during storage	-25°C ... + 85°C	
Protection class	IP20	
Installation position	variable, see <a href="#">note ▶ 31</a> !	
Approval	CE cULus [ <a href="#">▶ 33</a> ]	



## 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

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

#### E-Bus supply

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

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

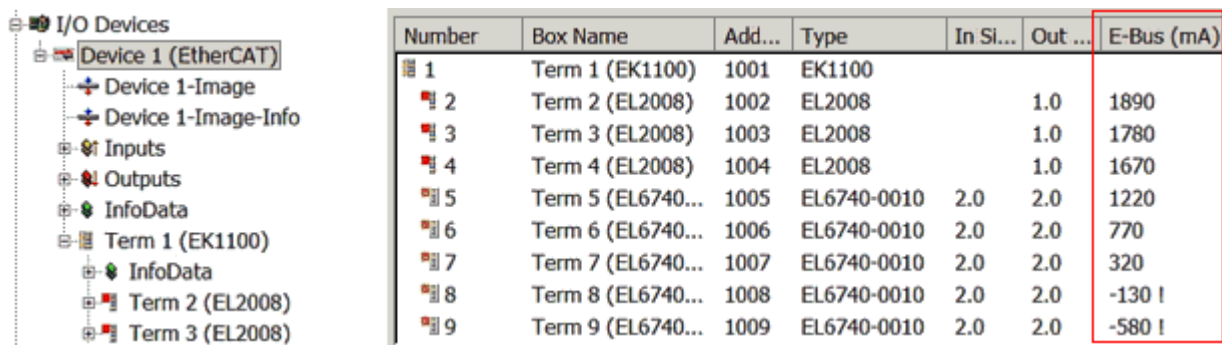


Fig. 12: System manager current calculation

**NOTE**

**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

ELxxxx terminals are equipped with a safety feature (watchdog) that switches off the outputs after a specifiable time e.g. in the event of an interruption of the process data traffic, depending on the device and settings, e.g. in OFF state.

The EtherCAT slave controller (ESC) in the EL2xxx terminals features 2 watchdogs:

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

#### SM watchdog (SyncManager Watchdog)

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

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

#### PDI watchdog (Process Data Watchdog)

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

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

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

The settings of the SM- and PDI-watchdog must be done for each slave separately in the TwinCAT System Manager.

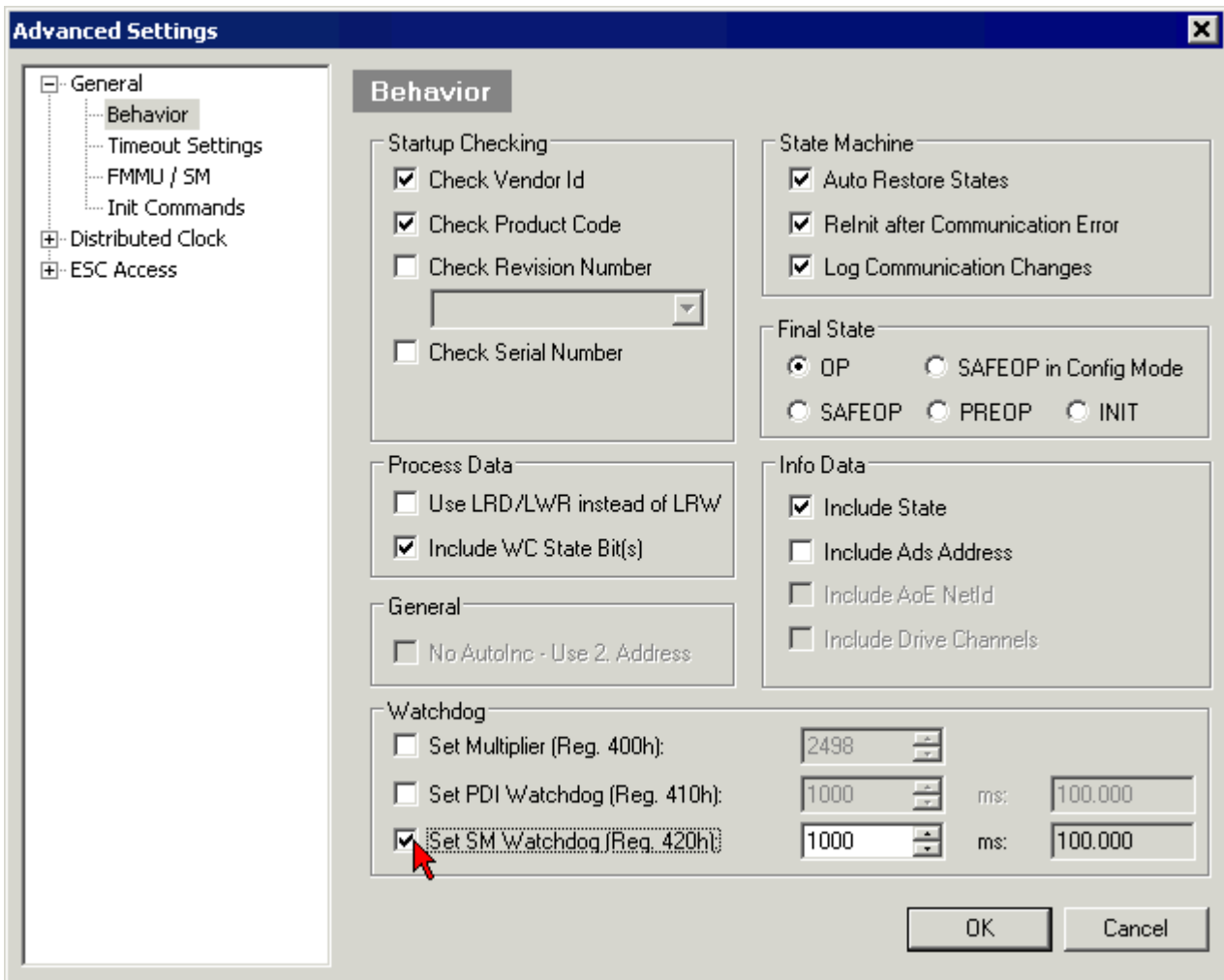


Fig. 13: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the multiplier is valid for both watchdogs.
- each watchdog has its own timer setting, the outcome of this in summary with the multiplier is a resulting time.
- Important: the multiplier/timer setting is only loaded into the slave at the start up, if the checkbox is activated.  
If the checkbox is not activated, nothing is downloaded and the ESC settings remain unchanged.

**Multiplier**

Multiplier

Both watchdogs receive their pulses from the local terminal cycle, divided by the watchdog multiplier:

$$1/25 \text{ MHz} * (\text{watchdog multiplier} + 2) = 100 \text{ } \mu\text{s} \text{ (for default setting of 2498 for the multiplier)}$$

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

The value in multiplier + 2 corresponds to the number of basic 40 ns ticks representing a watchdog tick. The multiplier can be modified in order to adjust the watchdog time over a larger range.

**Example "Set SM watchdog"**

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

**Calculation**

Multiplier = 2498 → watchdog base time =  $1 / 25 \text{ MHz} * (2498 + 2) = 0.0001 \text{ seconds} = 100 \mu\text{s}$   
 SM watchdog = 10000 →  $10000 * 100 \mu\text{s} = 1 \text{ second watchdog monitoring time}$

**⚠ CAUTION****Undefined state possible!**

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

**⚠ CAUTION****Damage of devices and undefined state possible!**

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

## 3.4 EtherCAT State Machine

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

A distinction is made between the following states:

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

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

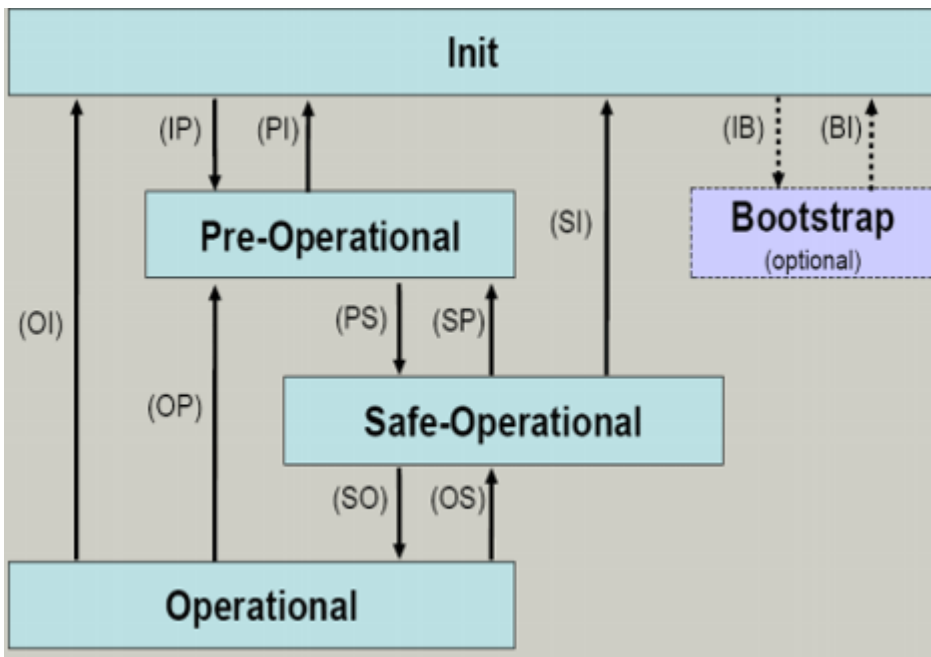


Fig. 14: States of the EtherCAT State Machine

**Init**

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

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

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

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

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

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

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

● **Outputs in SAFEOP state**

**i** The default set `watchdog [▶ 18]` monitoring sets the outputs of the module in a safe state - depending on the settings in `SAFEOP` and `OP` - e.g. in `OFF` state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the `SAFEOP` state.

**Operational (Op)**

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

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

**Boot**

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

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

## 3.5 CoE Interface

### General description

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

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

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

The order is specified in 2 levels via hexadecimal numbering: (main)index, followed by subindex. The value ranges are

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

Other important ranges are:

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

---

### ● Availability



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

---

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

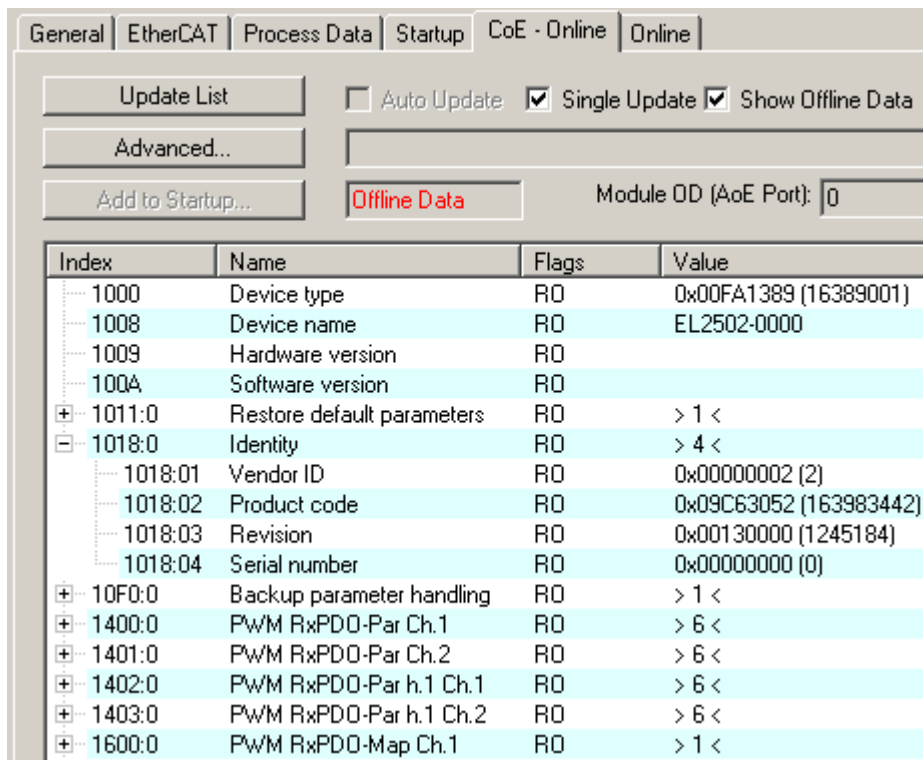


Fig. 15: "CoE Online " tab

The figure above shows the CoE objects available in device "EL2502", ranging from 0x1000 to 0x1600. The subindices for 0x1018 are expanded.

**Data management and function "NoCoeStorage"**

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

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

**i Data management**

If slave CoE parameters are modified online, Beckhoff devices store any changes in a fail-safe manner in the EEPROM, i.e. the modified CoE parameters are still available after a restart. The situation may be different with other manufacturers.

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

Please refer to the technical data in this documentation as to whether this applies to the respective device.

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

## **i** Startup list

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

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

### Recommended approach for manual modification of CoE parameters

- Make the required change in the System Manager  
The values are stored locally in the EtherCAT slave
- If the value is to be stored permanently, enter it in the Startup list.  
The order of the Startup entries is usually irrelevant.

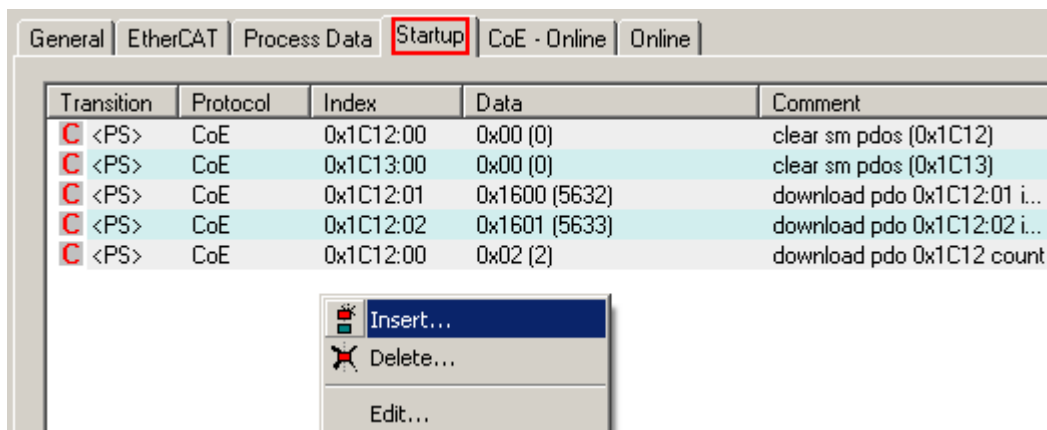


Fig. 16: Startup list in the TwinCAT System Manager

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

### Online/offline list

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

In both cases a CoE list as shown in Fig. "CoE online' tab" is displayed. The connectivity is shown as offline/online.

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



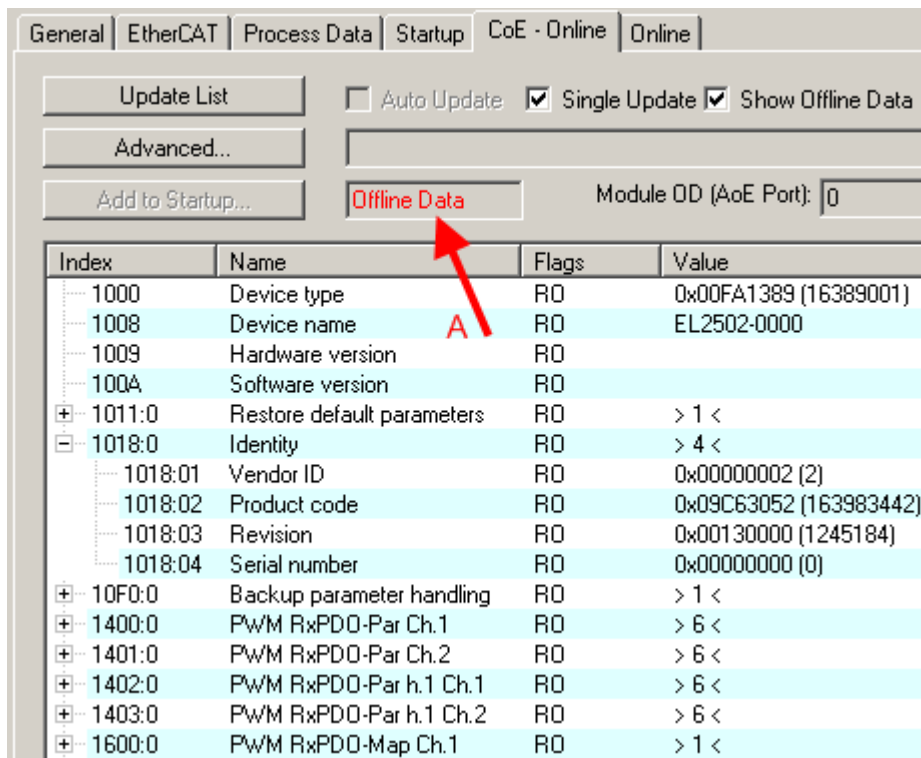


Fig. 17: Offline list

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

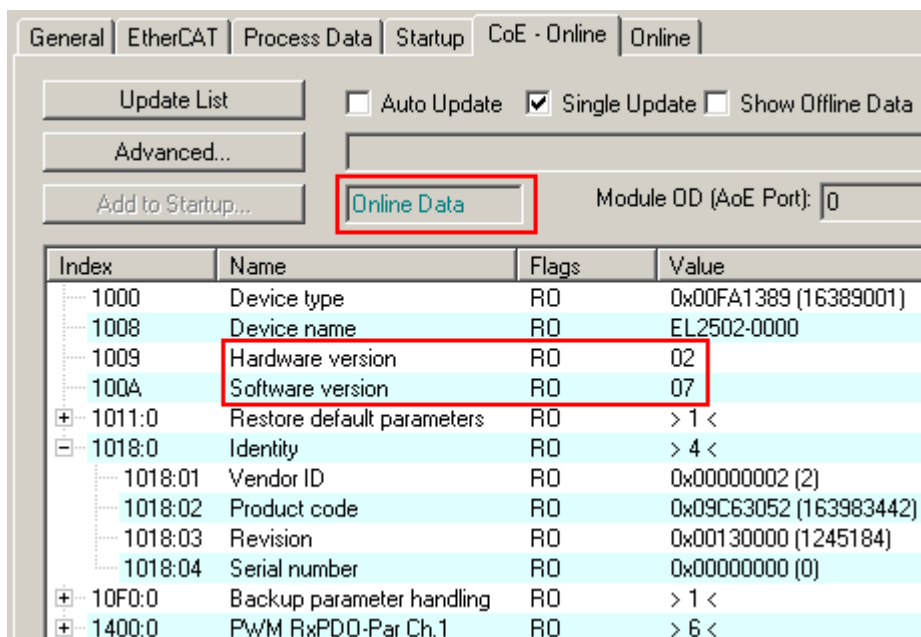


Fig. 18: Online list

**Channel-based order**

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

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

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

This is generally written as 0x80n0.

Detailed information on the CoE interface can be found in the [EtherCAT system documentation](#) on the Beckhoff website.

## 3.6 Distributed Clock

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

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

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

## 4 Installation

### 4.1 Instructions for ESD protection

#### NOTE

##### **Destruction of the devices by electrostatic discharge possible!**

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

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

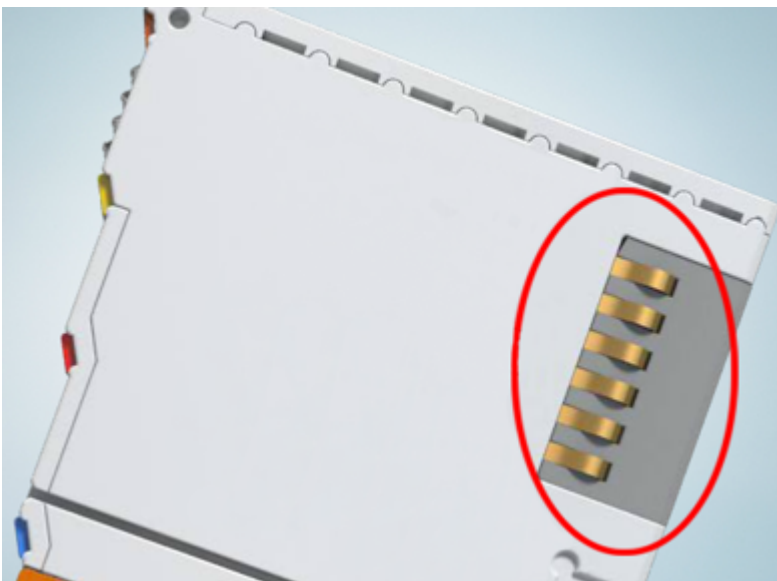


Fig. 19: Spring contacts of the Beckhoff I/O components

### 4.2 Recommended mounting rails

Terminal Modules und EtherCAT Modules of KMxxxx and EMxxxx series, same as the terminals of the EL66xx and EL67xx series can be snapped onto the following recommended mounting rails:

- DIN Rail TH 35-7.5 with 1 mm material thickness (according to EN 60715)
- DIN Rail TH 35-15 with 1,5 mm material thickness

#### **● Pay attention to the material thickness of the DIN Rail**

**i** Terminal Modules und EtherCAT Modules of KMxxxx and EMxxxx series, same as the terminals of the EL66xx and EL67xx series does not fit to the DIN Rail TH 35-15 with 2,2 to 2,5 mm material thickness (according to EN 60715)!

### 4.3 Mounting and demounting - terminals with front unlocking

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

### **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 recommended mounting rails under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

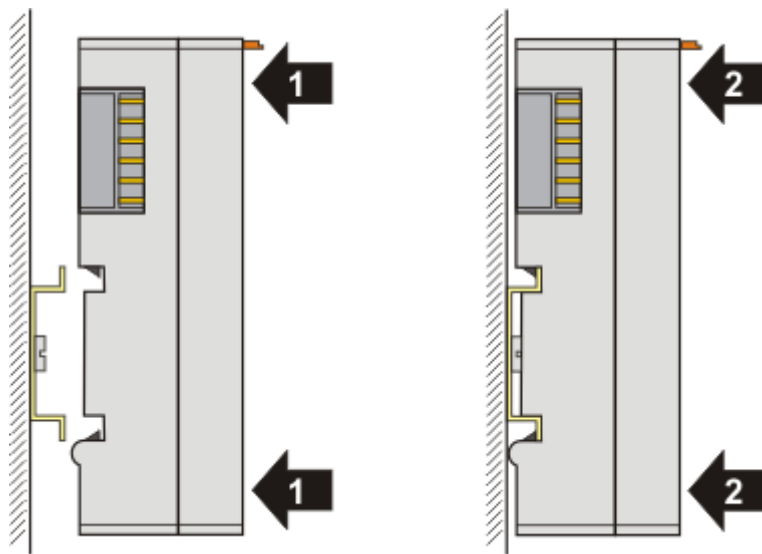
### **⚠ WARNING**

#### **Risk of electric shock and damage of device!**

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

### **Mounting**

- Fit the mounting rail to the planned assembly location.

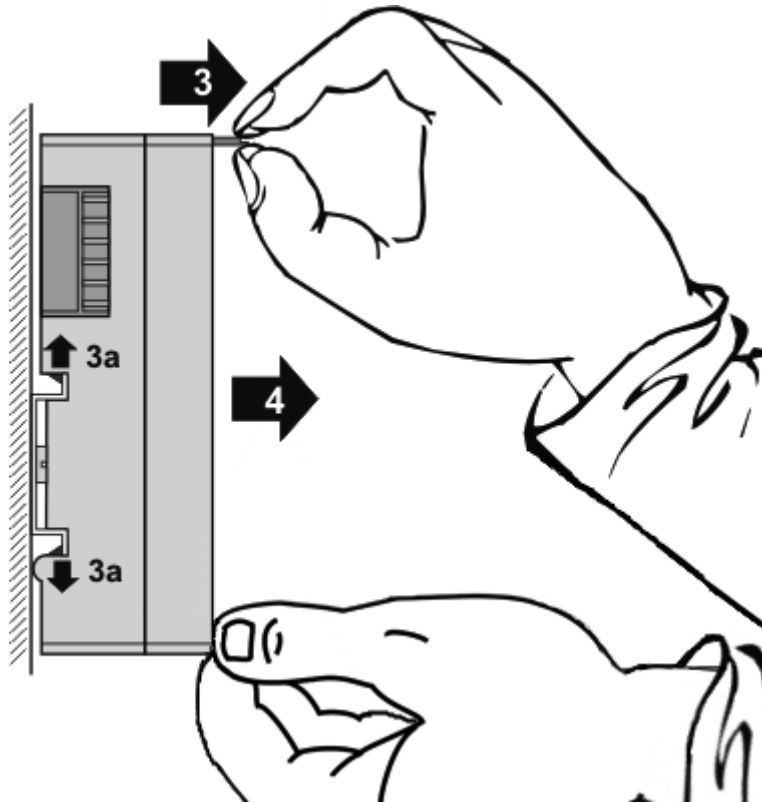


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

- Attach the cables.

### **Demounting**

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



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

## 4.4 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 2 passive terminals!

**Examples for positioning of passive terminals (highlighted)**

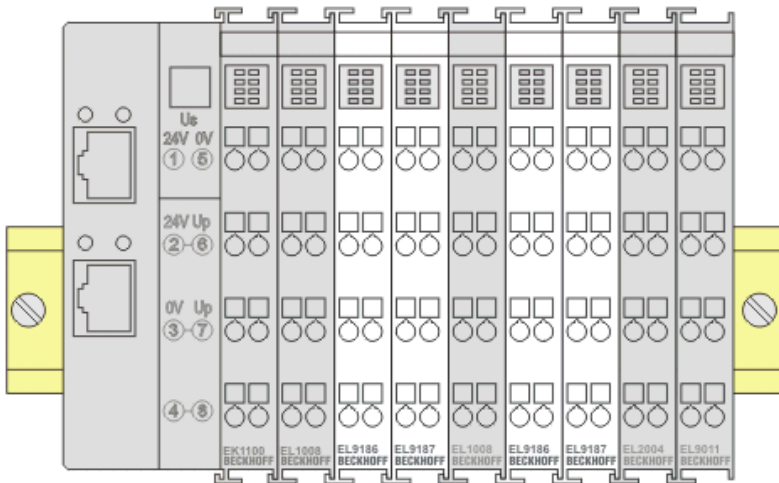


Fig. 20: Correct positioning

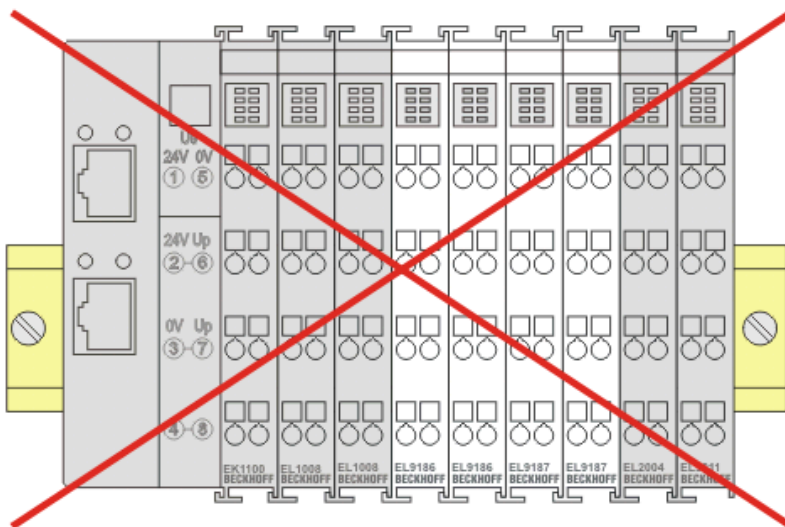


Fig. 21: Incorrect positioning

## 4.5 Installation positions

**NOTE**

**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.

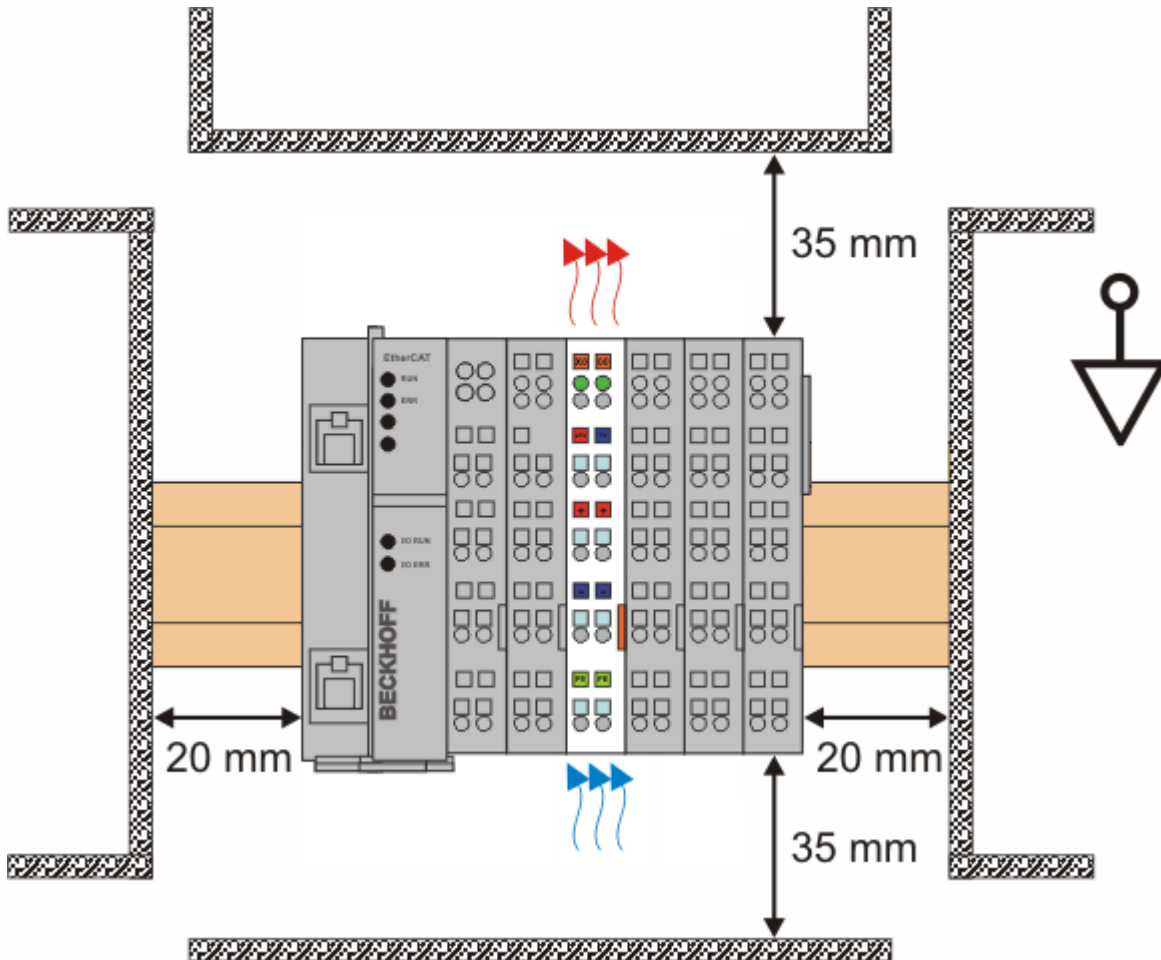


Fig. 22: Recommended distances for standard installation position

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.



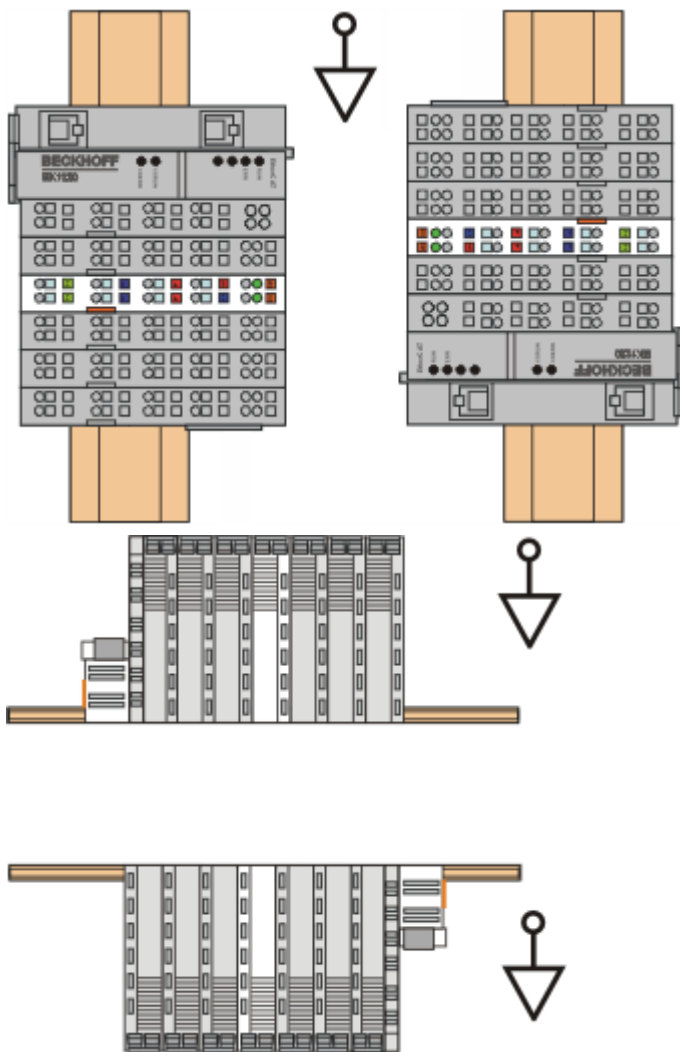





Fig. 23: Other installation positions

## 4.6 UL notice

	<p><b>Application</b> Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.</p>
	<p><b>Examination</b> 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).</p>
	<p><b>For devices with Ethernet connectors</b> Not for connection to telecommunication circuits.</p>

### Basic principles

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



## 5 Commissioning

### 5.1 TwinCAT Quick Start

TwinCAT is a development environment for real-time control including 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 Systemmanual:**  
Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- **TwinCAT 2** → TwinCAT System Manager → I/O - Configuration
- In particular, TwinCAT driver installation:  
**Fieldbus components** → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation

Devices contain the 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 offline mode can be found under <http://infosys.beckhoff.com>:  
**TwinCAT 2** → TwinCAT System Manager → IO - Configuration → Adding 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 from user PC to the individual control elements:

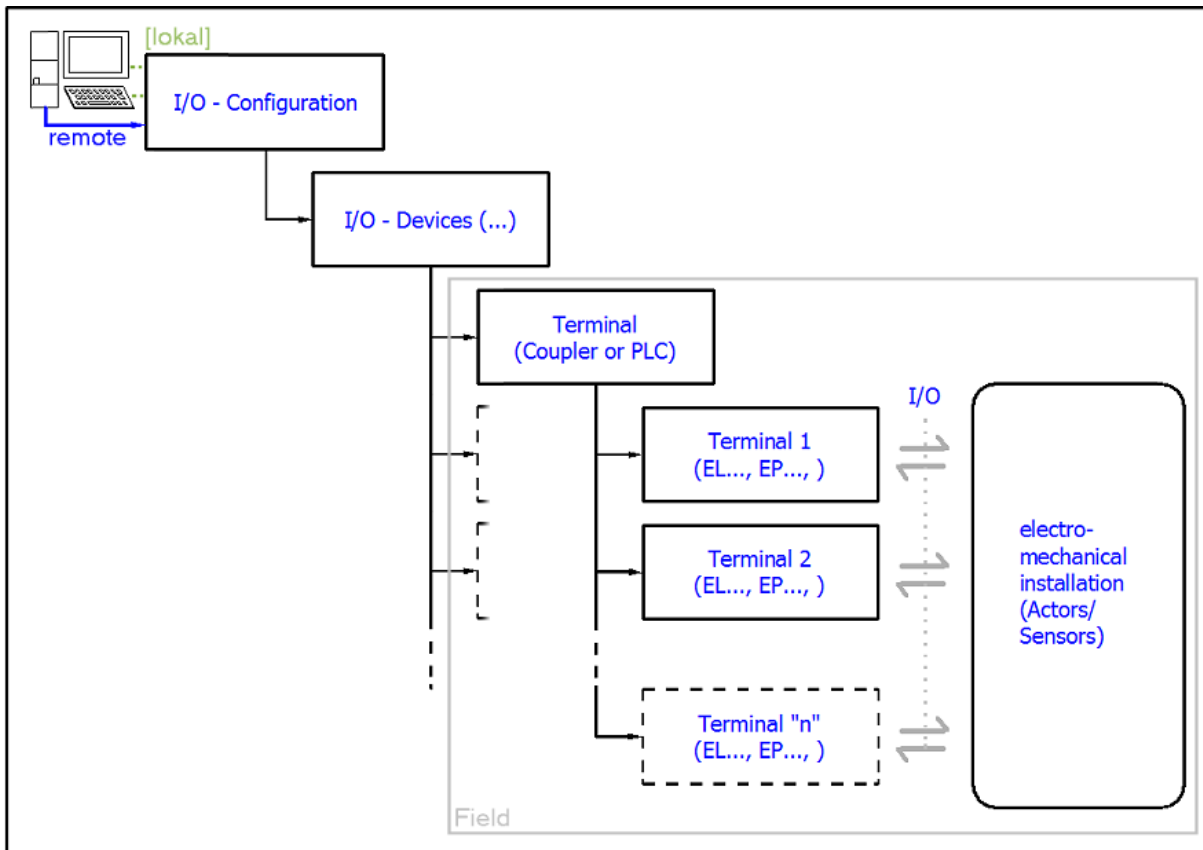


Fig. 24: Relationship between user side (commissioning) and installation

The user inserting of certain components (I/O device, terminal, box...) is the same in TwinCAT 2 and TwinCAT 3. The descriptions below relate to the online procedure.

### Sample configuration (actual configuration)

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

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

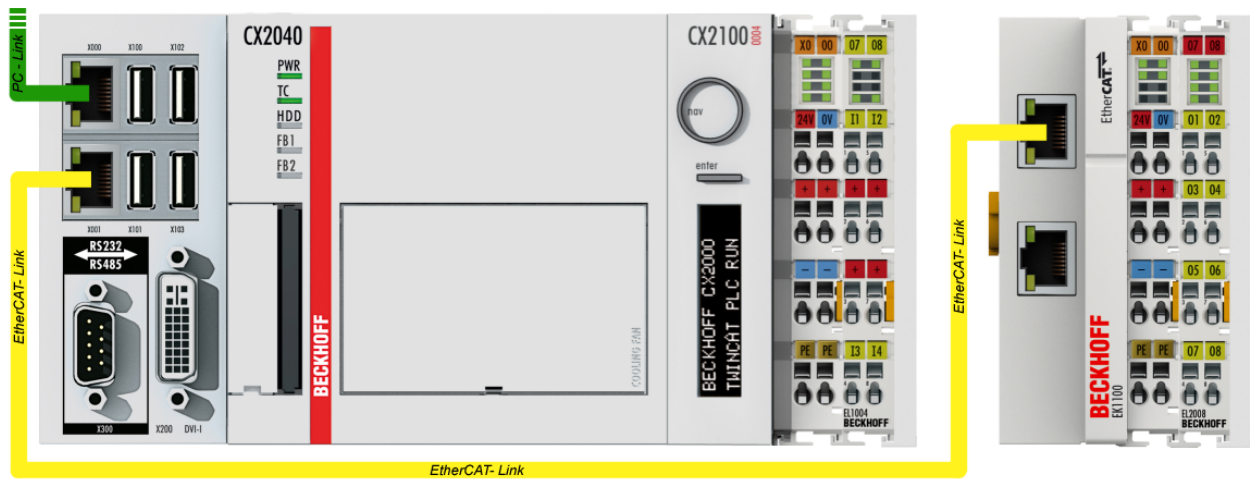


Fig. 25: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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

## 5.1.1 TwinCAT 2

### Startup

TwinCAT 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:

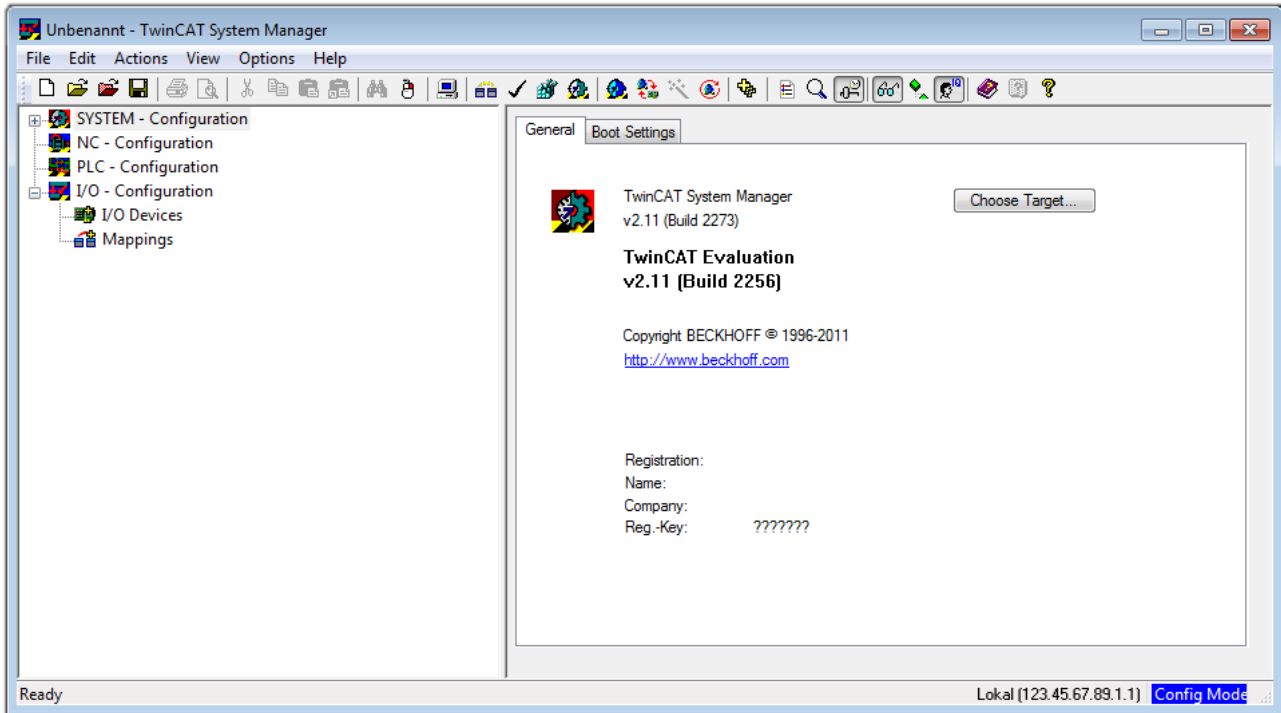



Fig. 26: Initial TwinCAT 2 user interface

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 thereby the next step is "Insert Device [▶ 39]".

If the intention is to address the TwinCAT runtime environment installed on a PLC as development environment remotely from another system, the target system must be made known first. In the menu under

"Actions" → "Choose Target System...", via the symbol "  " or the "F8" key, open the following window:

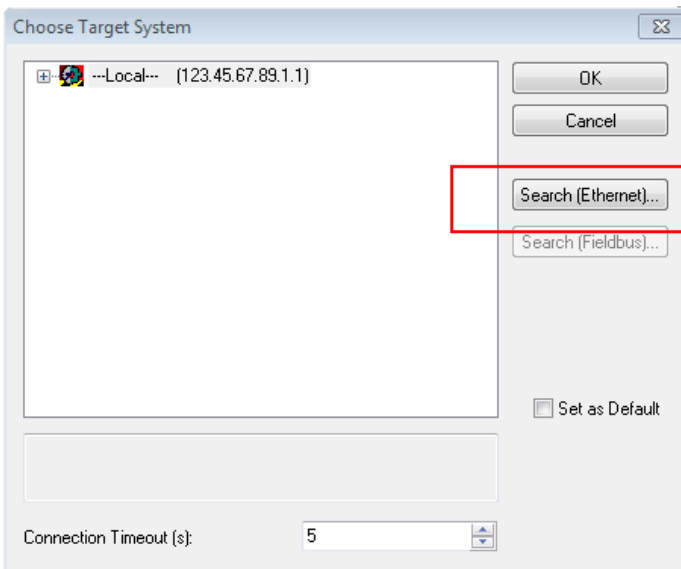


Fig. 27: Selection of the target system

Use "Search (Ethernet)..." to enter the target system. Thus a next 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.

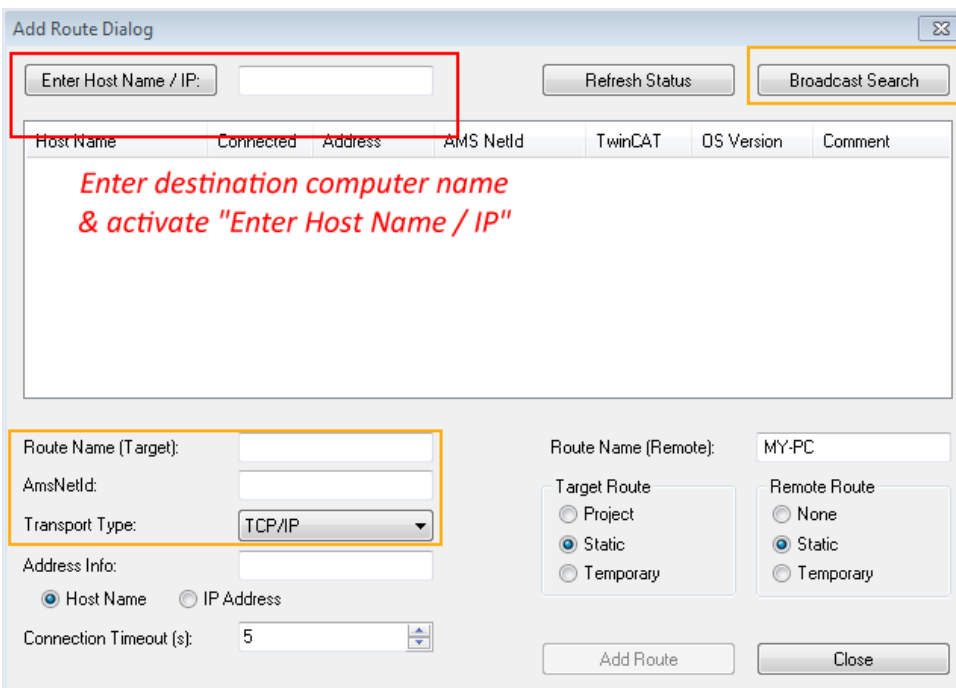
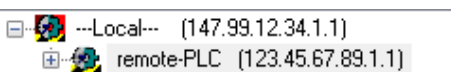


Fig. 28: Specify the PLC for access by the TwinCAT System Manager: selection of the target system



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



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 menu "Actions" → "Set/Reset TwinCAT to Config Mode..." (Shift + F4).

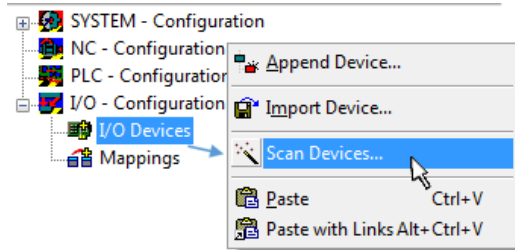


Fig. 29: Select "Scan Devices..."

Confirm the warning message, which follows, and select "EtherCAT" in the dialog:

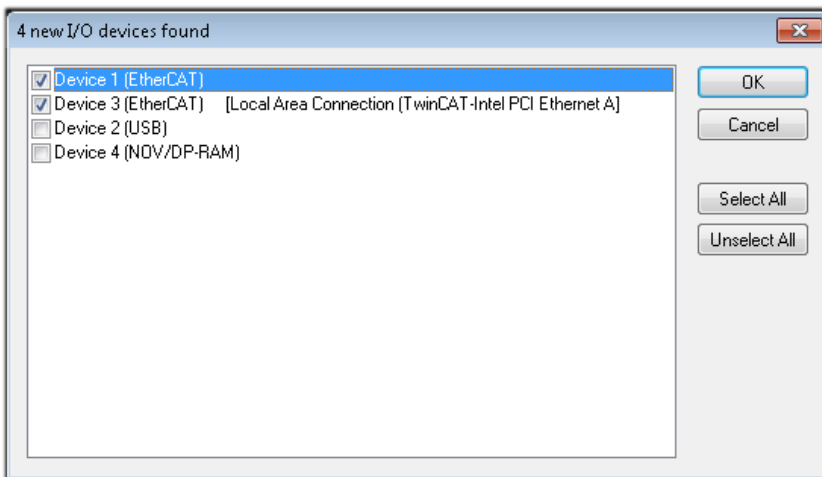


Fig. 30: Automatic detection of I/O devices: selection the devices to be integrated

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 [sample configuration](#) [▶ 35] described at the beginning of this section, the result is as follows:

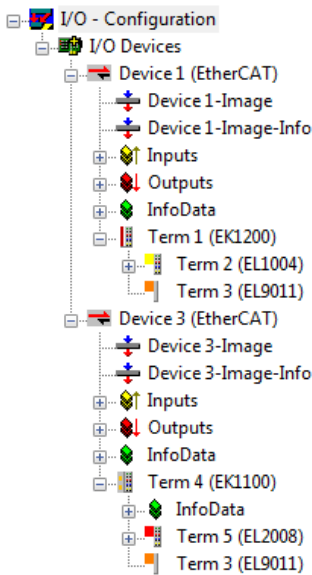


Fig. 31: Mapping of the configuration in the TwinCAT 2 System Manager

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

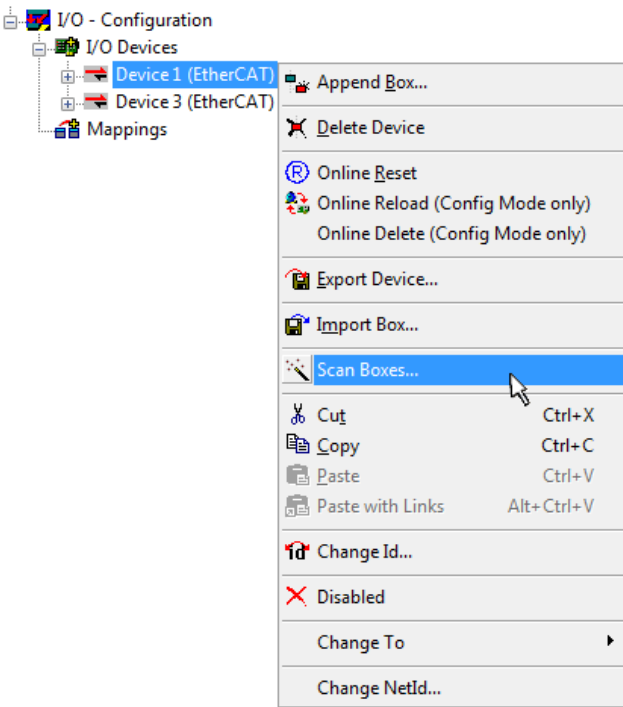


Fig. 32: Reading of individual terminals connected to a device

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 the creation of 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 to Structured Text (ST).

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

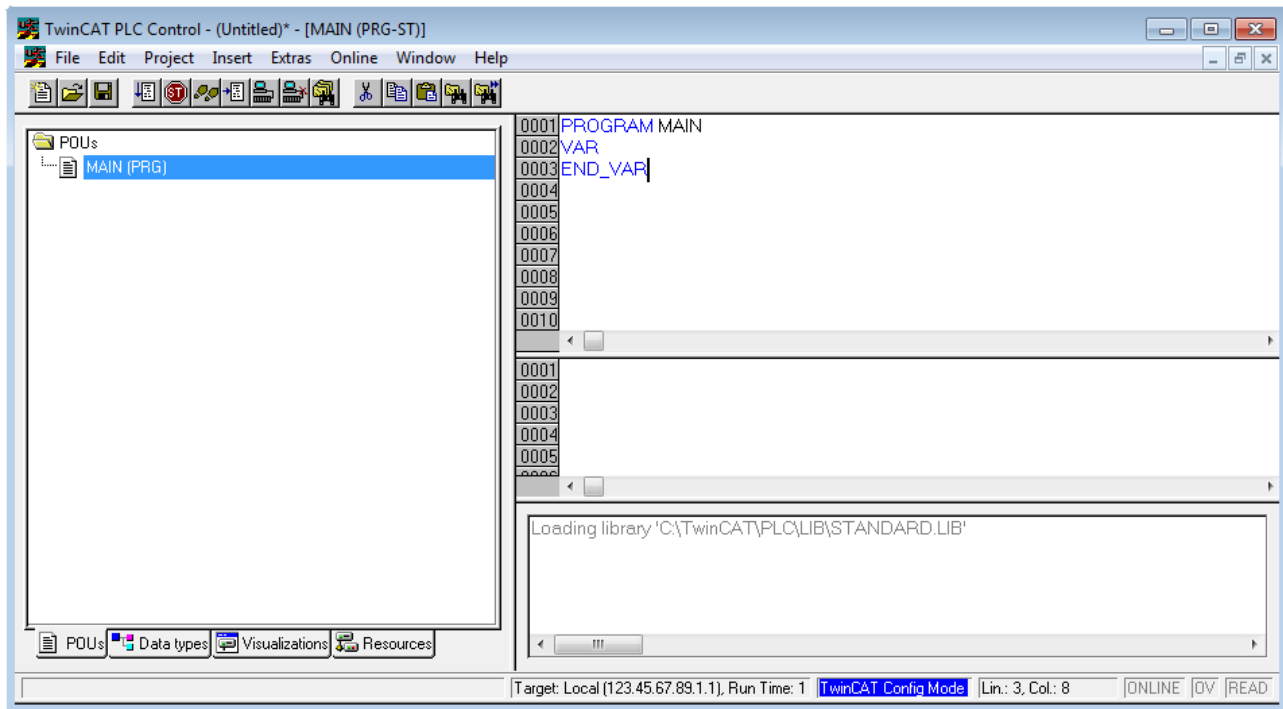


Fig. 33: TwinCAT PLC Control after startup

Sample variables and a sample program have been created and stored under the name "PLC\_example.pro":

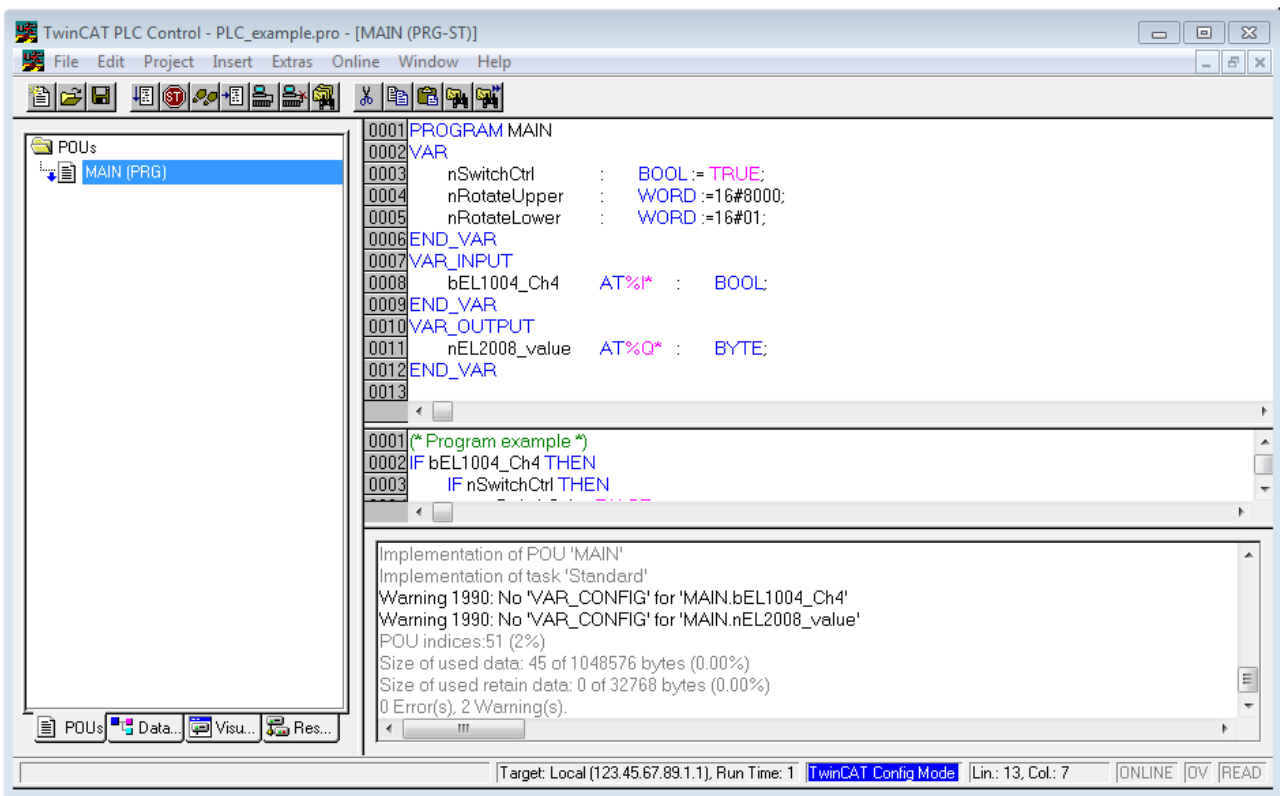


Fig. 34: Sample program with variables after a compile process (without variable integration)

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** via the context menu of the PLC configuration; right-click and select "Append PLC Project...":

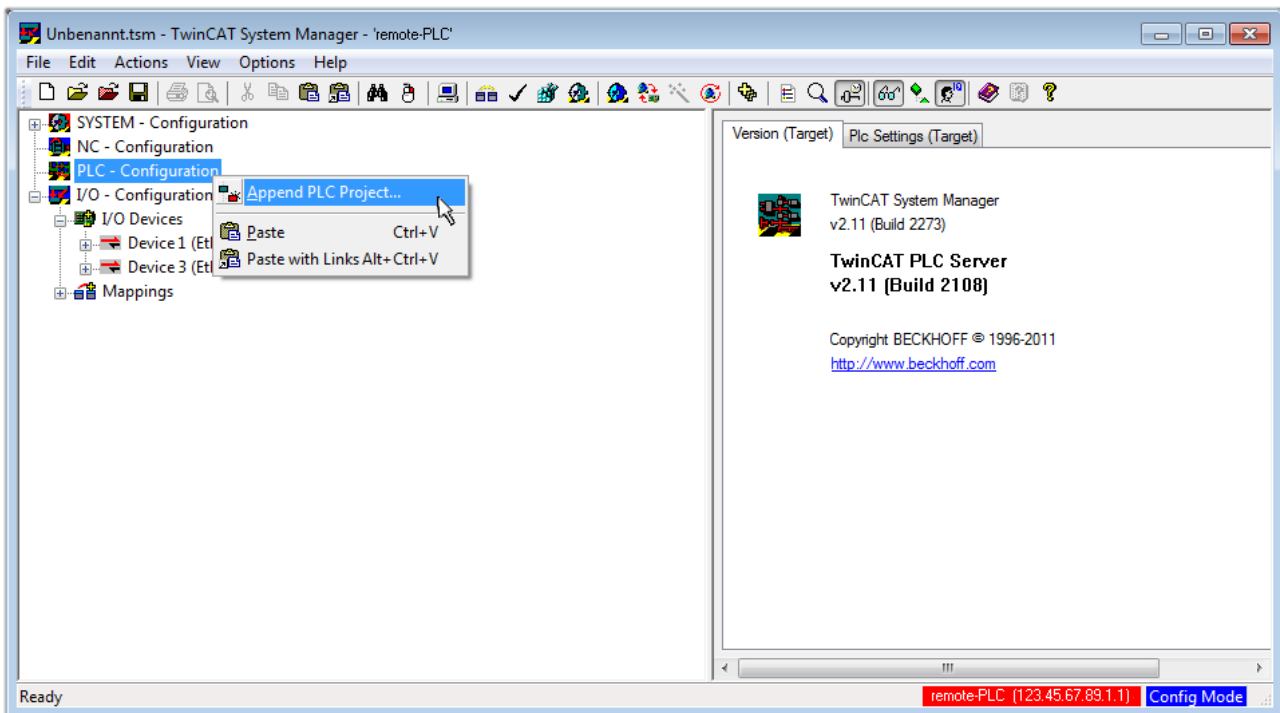


Fig. 35: Appending the TwinCAT PLC Control project

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:

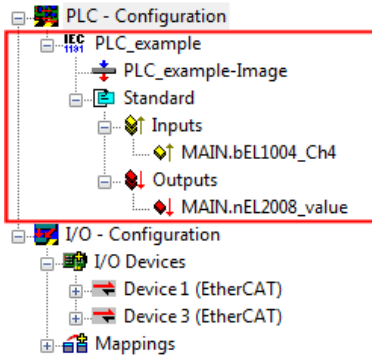


Fig. 36: PLC project integrated in the PLC configuration of the System Manager

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":

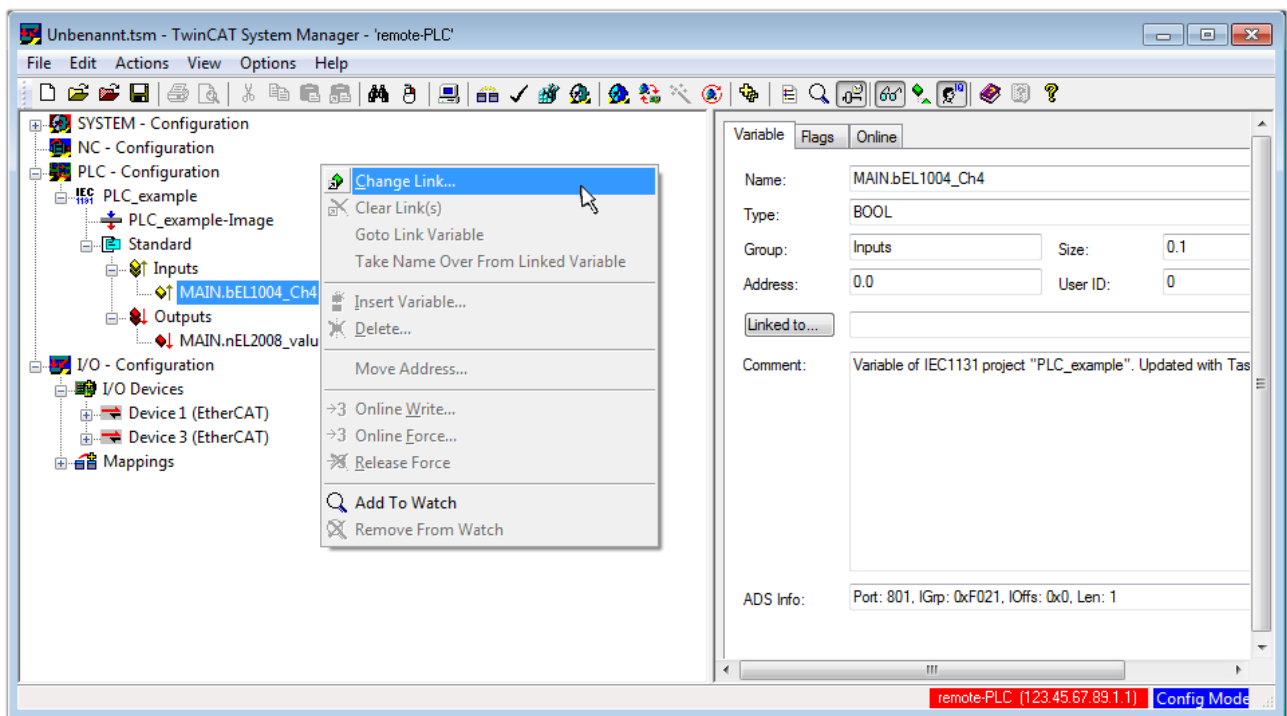


Fig. 37: Creating the links between PLC variables and process objects

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

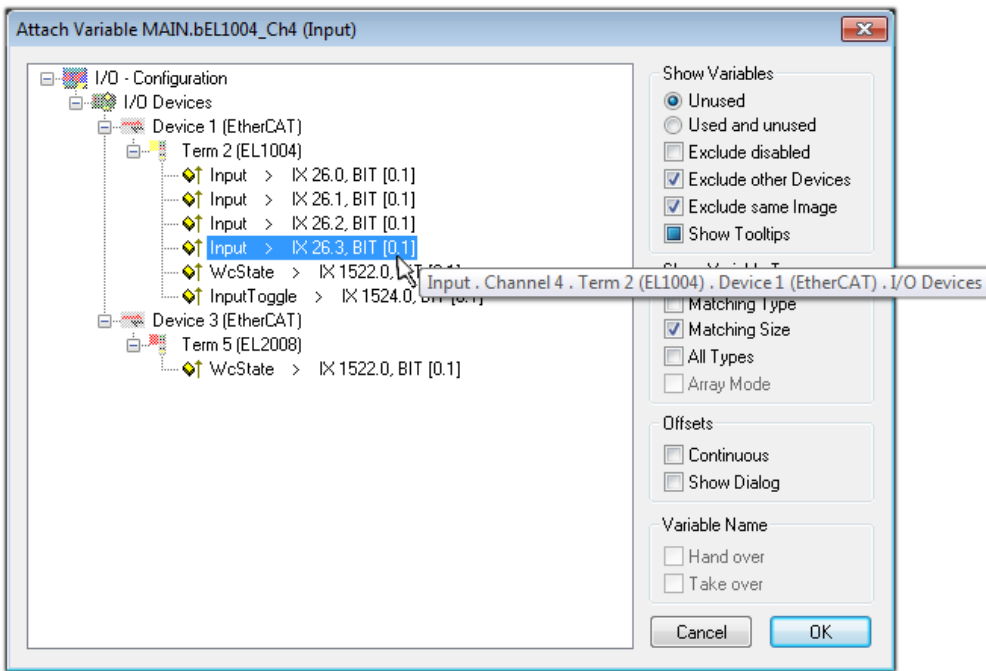


Fig. 38: Selecting PDO of type BOOL

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

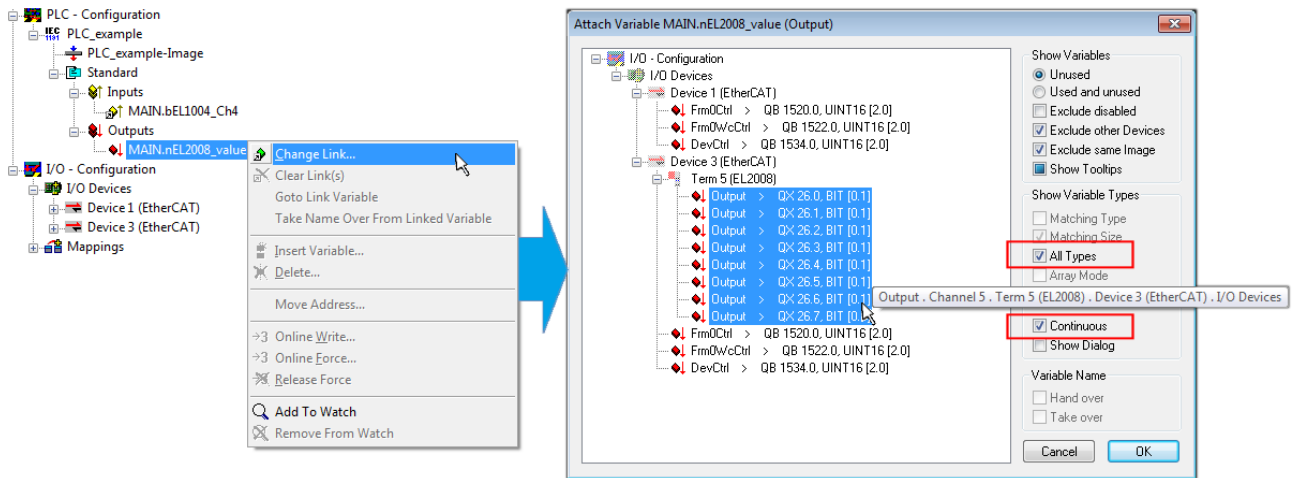



Fig. 39: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

Note that the "Continuous" checkbox was also activated. This is designed to allocate the bits contained in the byte of the variable "nEL2008\_value" sequentially to all eight selected output bits of the EL2008 terminal. In this way it is 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 (  ) at the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting a "Goto Link Variable" from the context menu of a variable. The object opposite, in this case the PDO, is automatically selected:

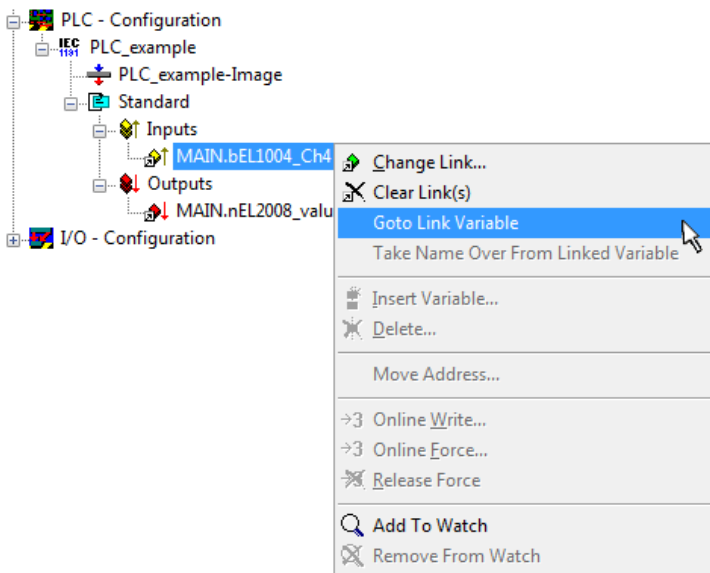

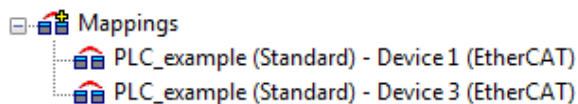


Fig. 40: Application of a "Goto Link" variable, using "MAIN.bEL1004\_Ch4" as a sample

The process of assigning variables to the PDO is completed via the menu selection "Actions" → "Generate

Mappings", key Ctrl+M or by clicking on the symbol  in the menu.


This can be visualized in the configuration:




The process of creating links can also take place in the opposite direction, i.e. starting with individual PDOs to variable. However, in this example it would then 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 possible to allocate this a set of bit-standardised variables (type "BOOL"). Here, too, a "Goto Link Variable" from the context menu of a PDO 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 are 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 Ethernet via "Online" → "Choose Run-Time System...":

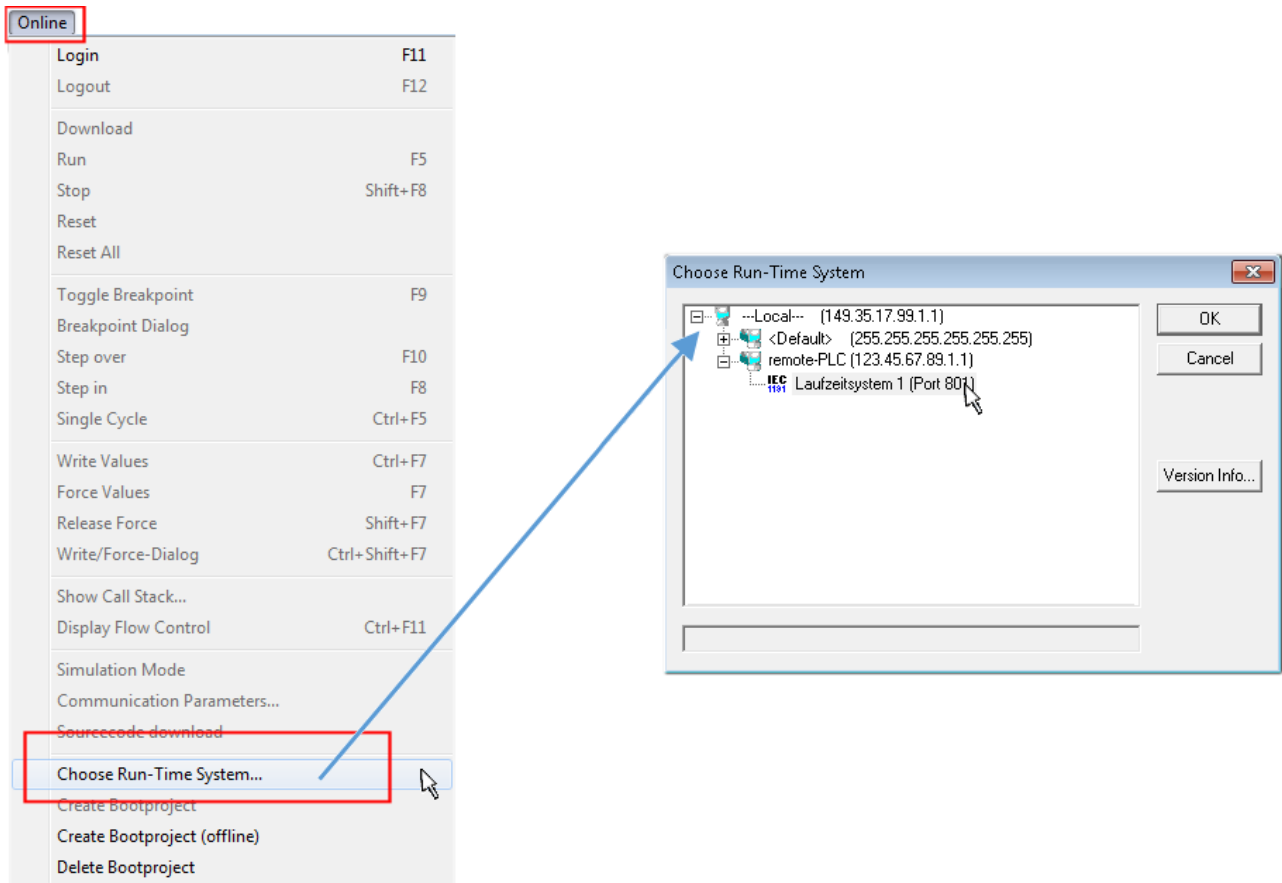



Fig. 41: Choose target system (remote)

In this sample "Runtime system 1 (port 801)" is selected and confirmed. Link the PLC with the real-time

system via 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 acknowledged with "Yes". The runtime environment is ready for the program start:

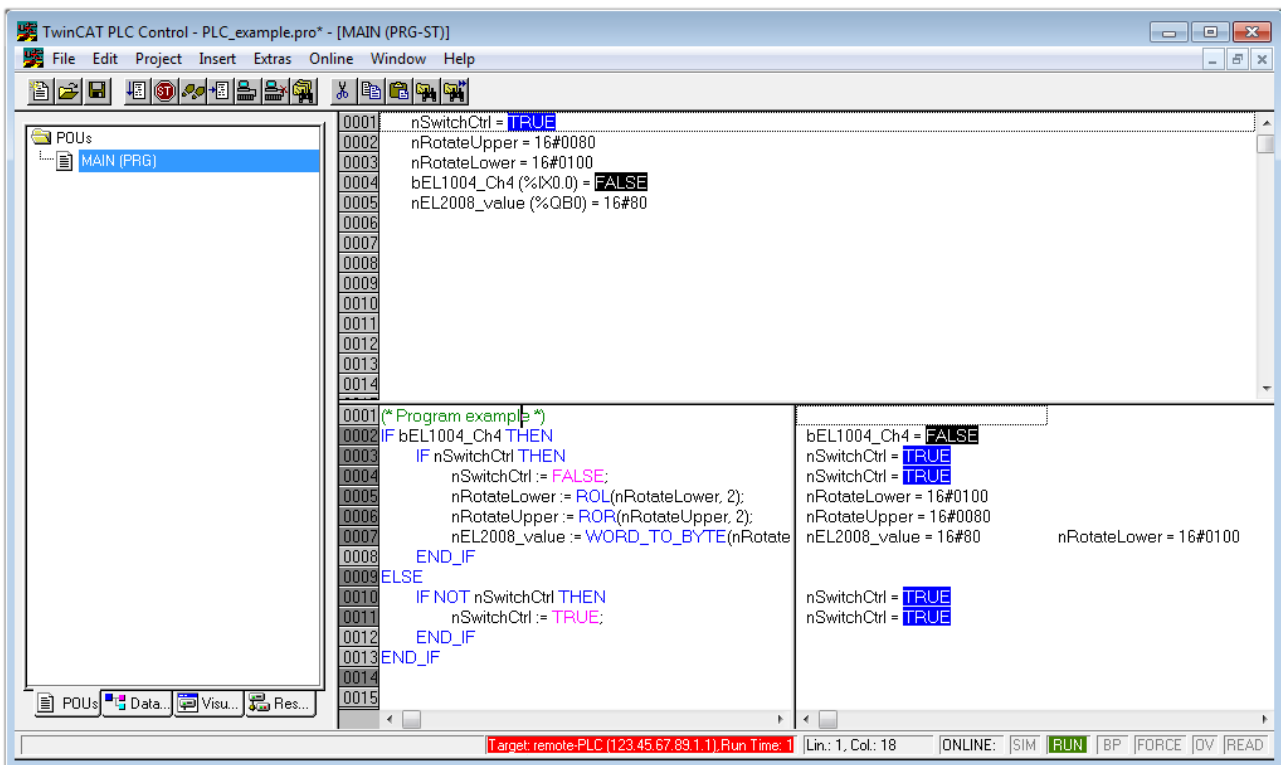


Fig. 42: PLC Control logged in, ready for program startup

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

### 5.1.2 TwinCAT 3

#### Startup

TwinCAT makes the development environment areas available together with Microsoft Visual Studio: after startup, the project folder explorer appears on the left in the general window area (cf. "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:

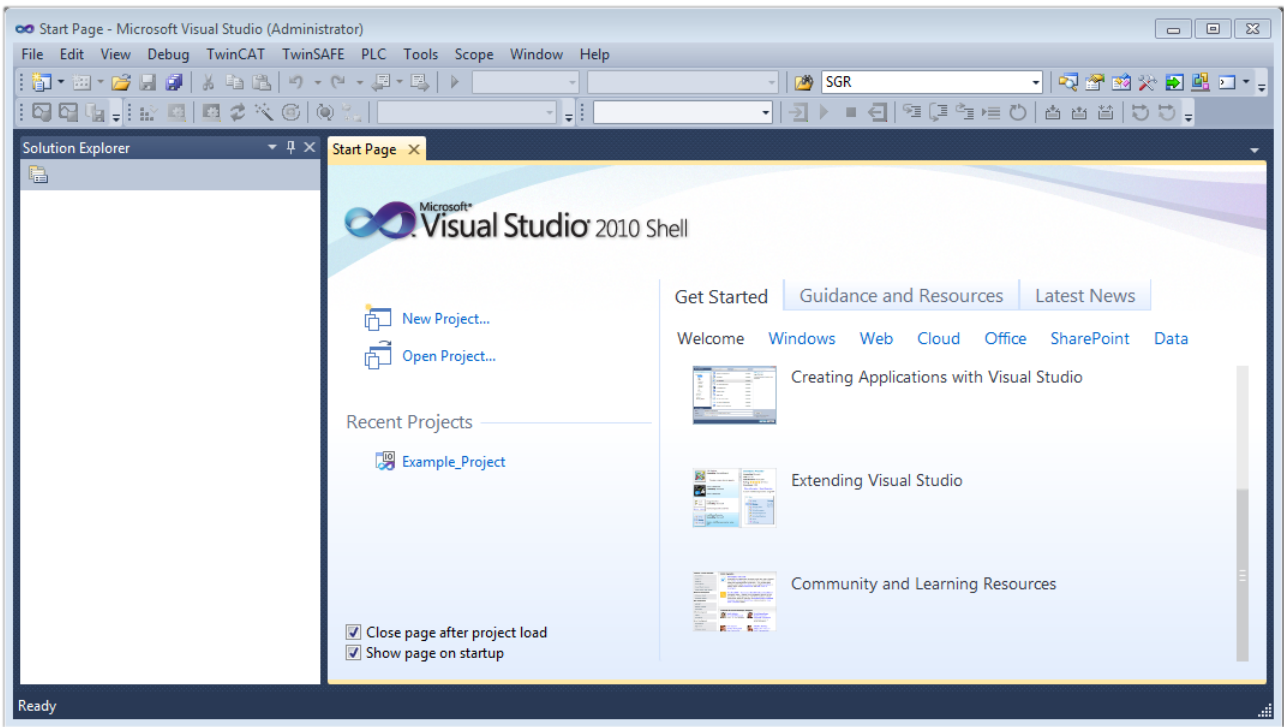



Fig. 43: Initial TwinCAT 3 user interface

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):

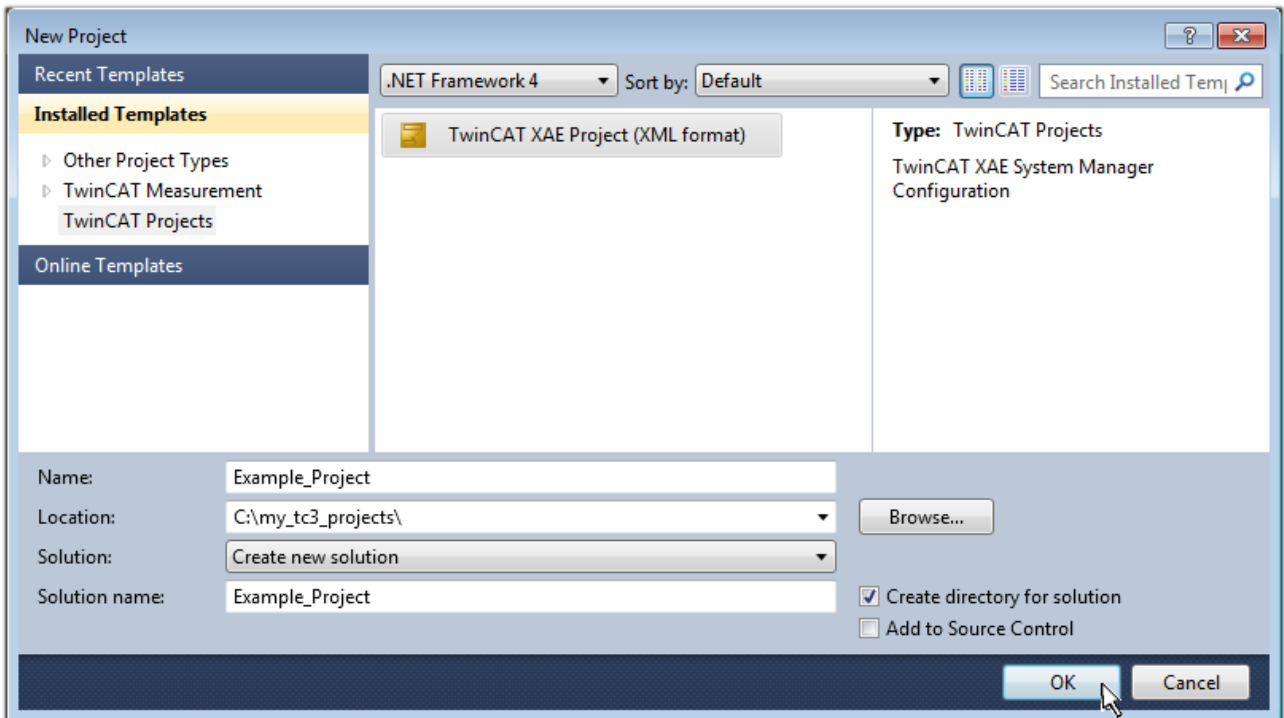


Fig. 44: Create new TwinCAT project

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



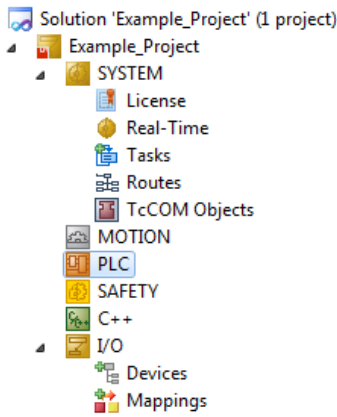
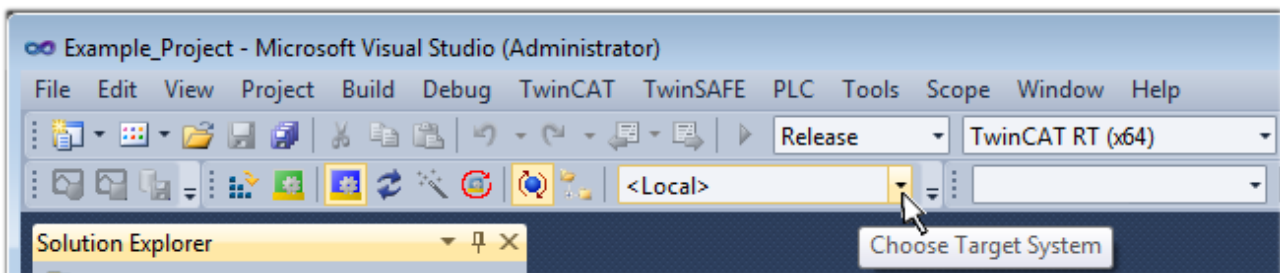


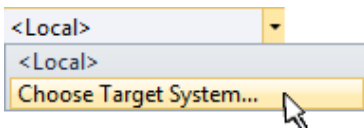
Fig. 45: New TwinCAT3 project in the project folder explorer

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, TwinCAT can be used in local mode and thereby the next step is "Insert Device [▶ 50]".

If the intention is to address the TwinCAT runtime environment installed on a PLC as development environment remotely from another system, 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:

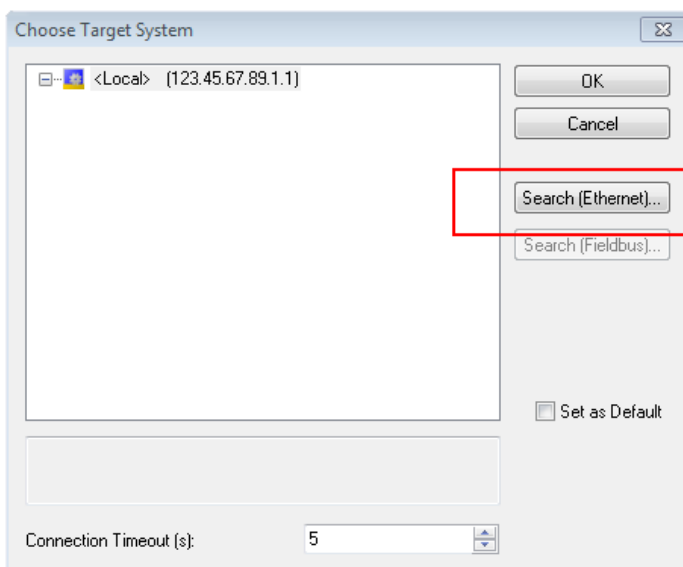


Fig. 46: Selection dialog: Choose the target system

Use "Search (Ethernet)..." to enter the target system. Thus a next 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.

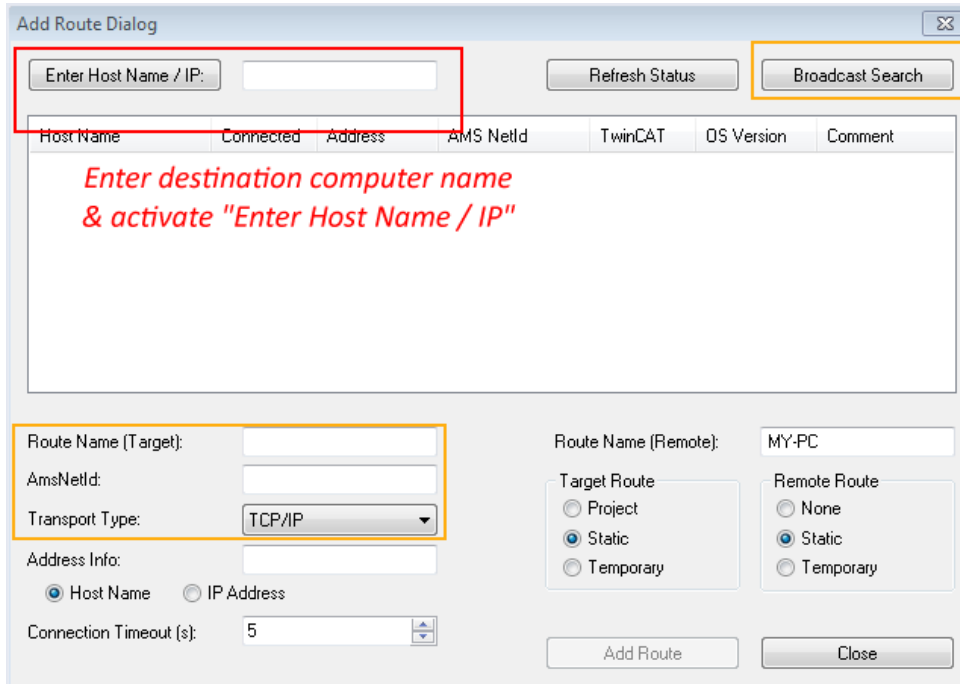
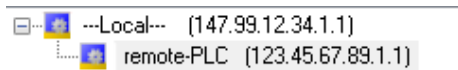


Fig. 47: Specify the PLC for access by the TwinCAT System Manager: selection of the target system


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




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

**Adding devices**

In the project folder explorer of the Visual Studio shell user interface on the left, select "Devices" within

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)".

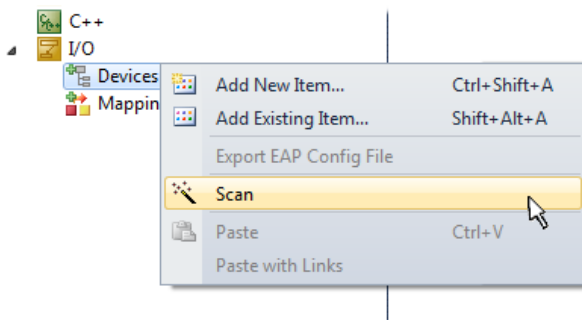


Fig. 48: Select "Scan"

Confirm the warning message, which follows, and select "EtherCAT" in the dialog:

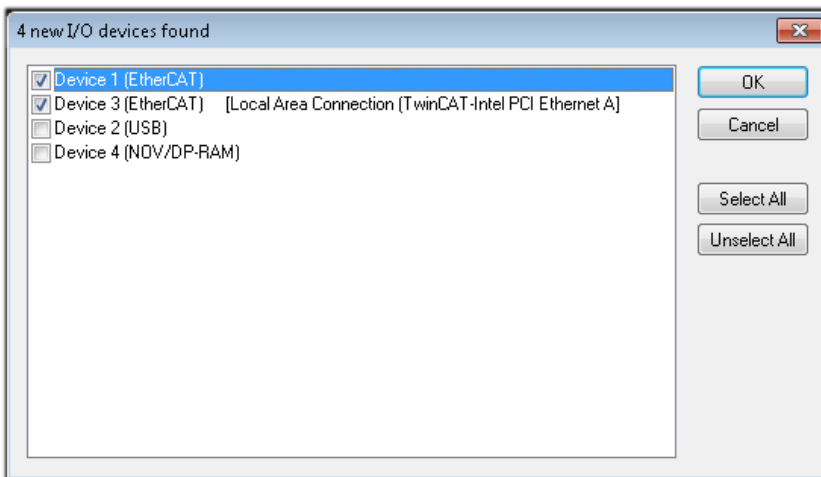


Fig. 49: Automatic detection of I/O devices: selection the devices to be integrated

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 [sample configuration \[▶\\_35\]](#) described at the beginning of this section, the result is as follows:

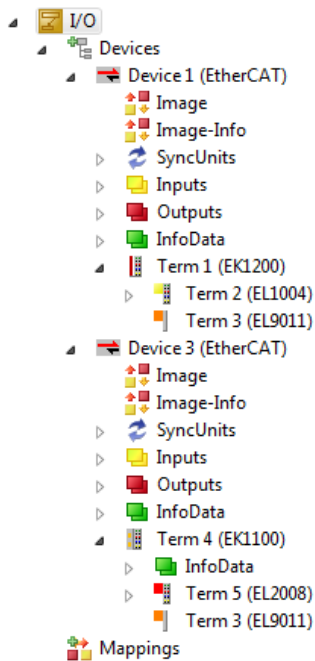


Fig. 50: Mapping of the configuration in VS shell of the TwinCAT3 environment

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

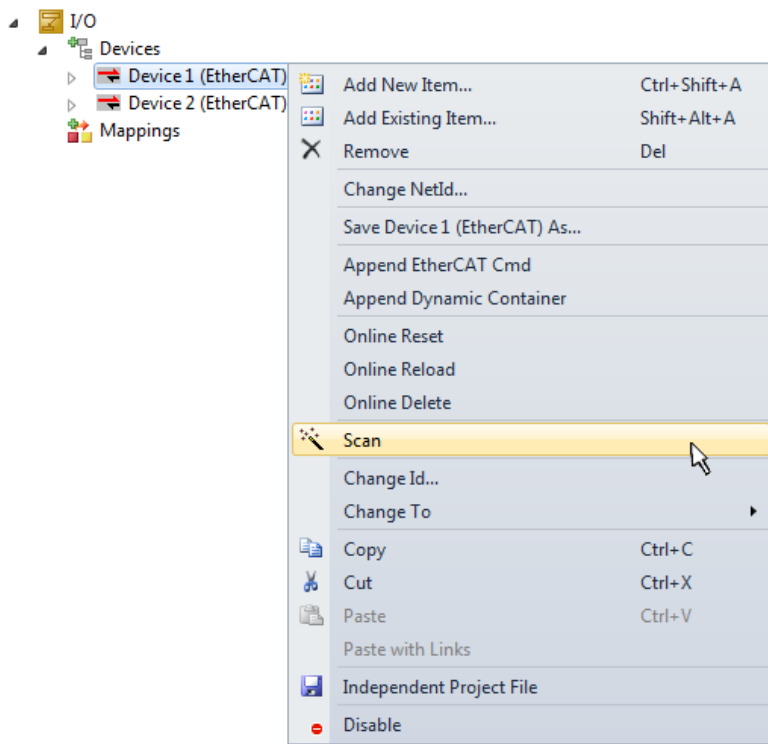


Fig. 51: Reading of individual terminals connected to a device

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

### Programming the PLC

TwinCAT PLC Control is the development environment for the creation of 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 to Structured Text (ST).

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

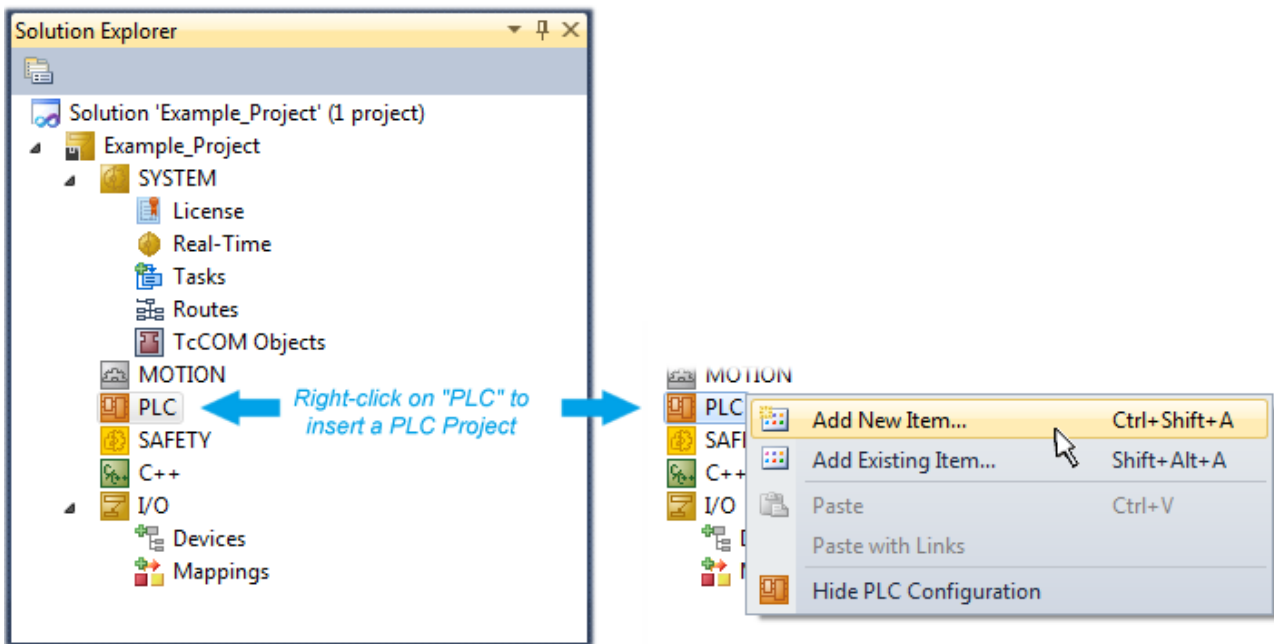


Fig. 52: Adding the programming environment in "PLC"

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

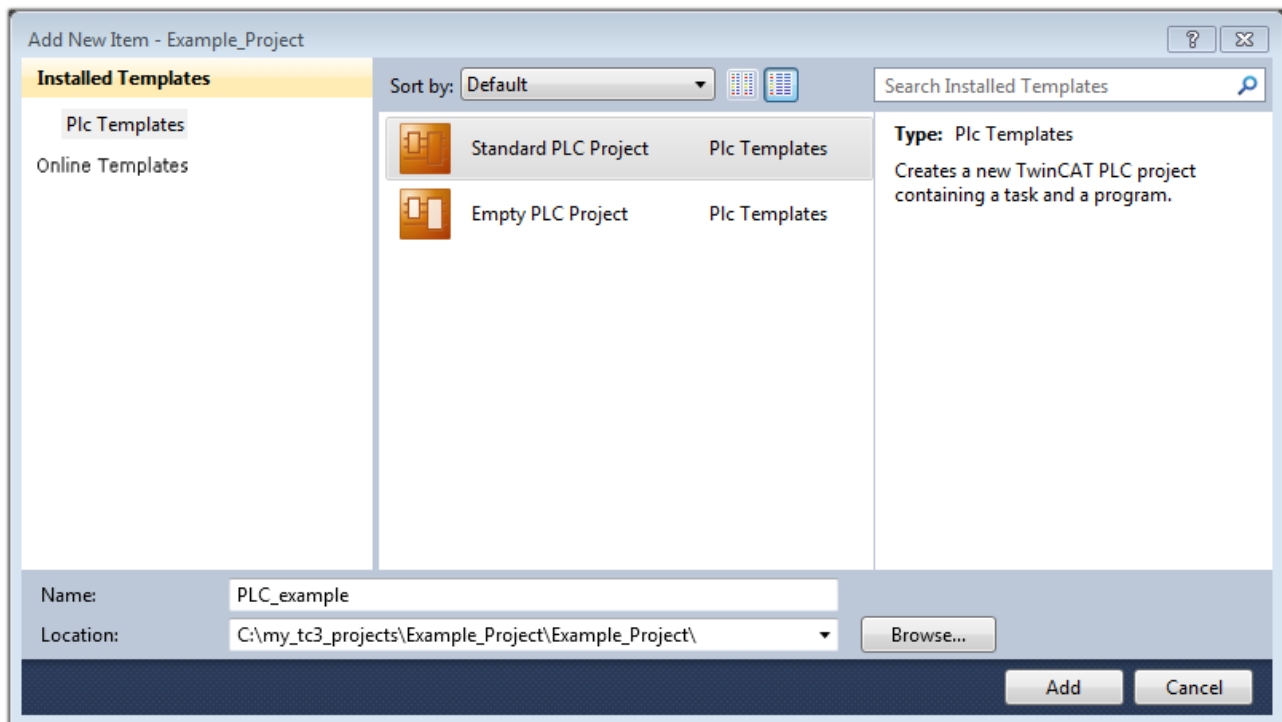


Fig. 53: Specifying the name and directory for the PLC programming environment

The "Main" program, which already exists by 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:

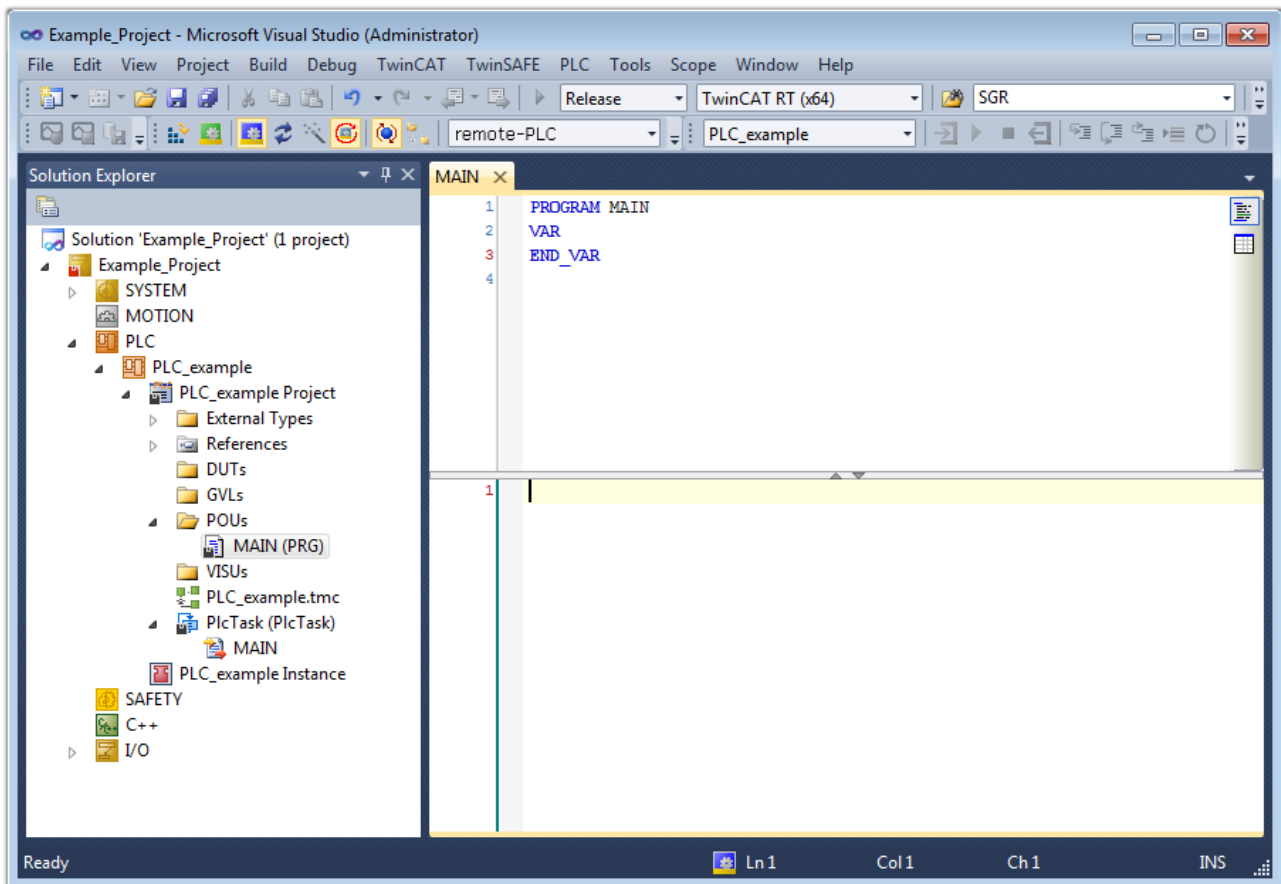


Fig. 54: Initial "Main" program of the standard PLC project

To continue, sample variables and a sample program have now been created:

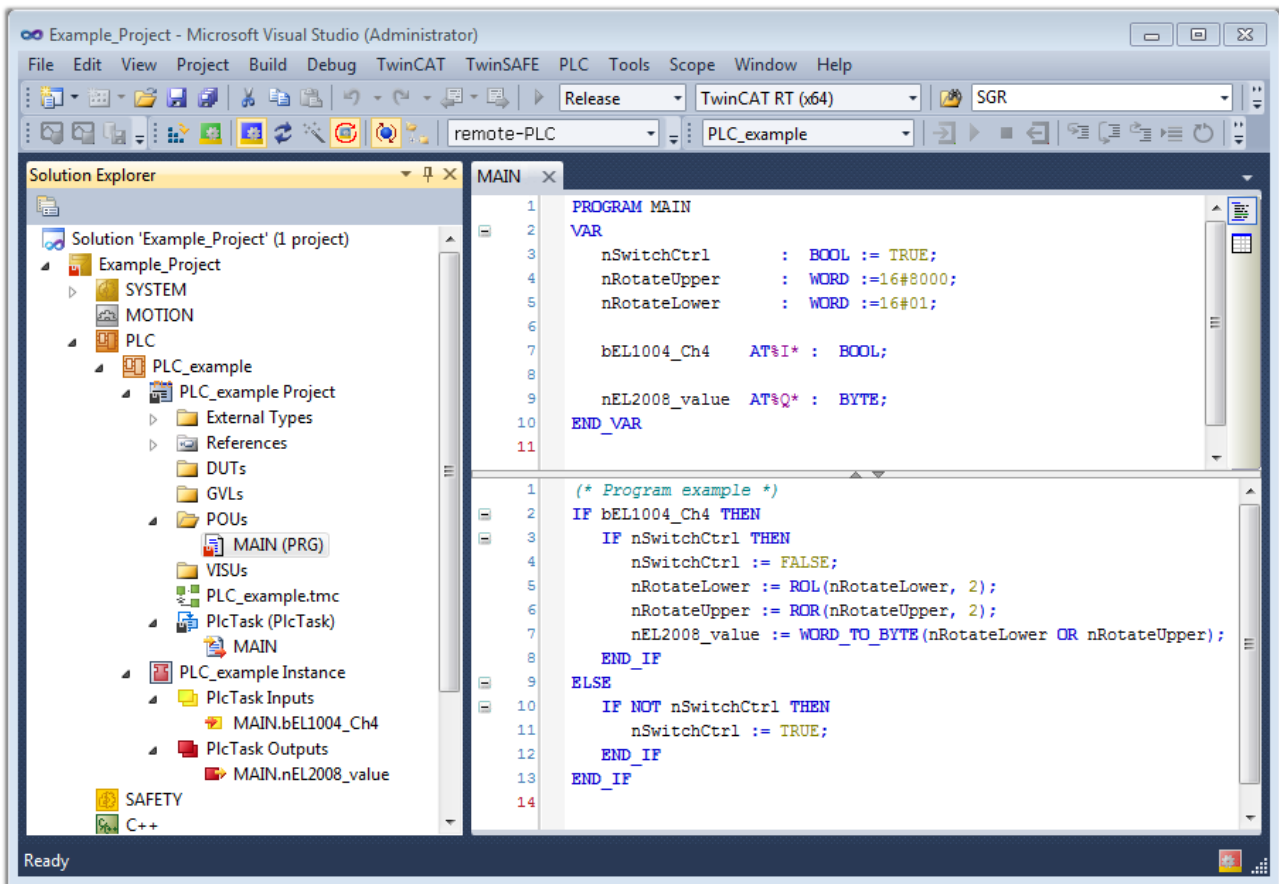


Fig. 55: Sample program with variables after a compile process (without variable integration)

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

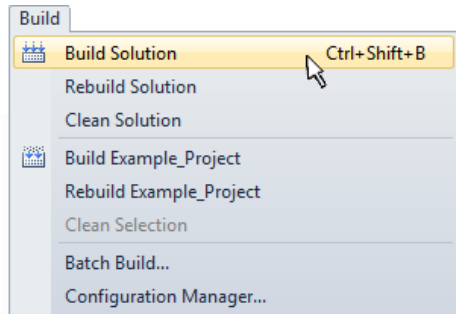
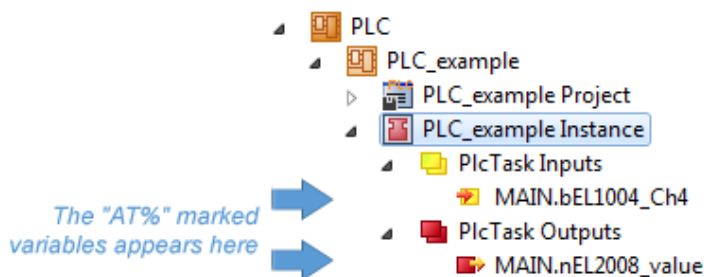


Fig. 56: Start program compilation

The following variables, identified in the ST/ PLC program with "AT%", are then available in 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 for selecting a suitable process object (PDO) for linking:

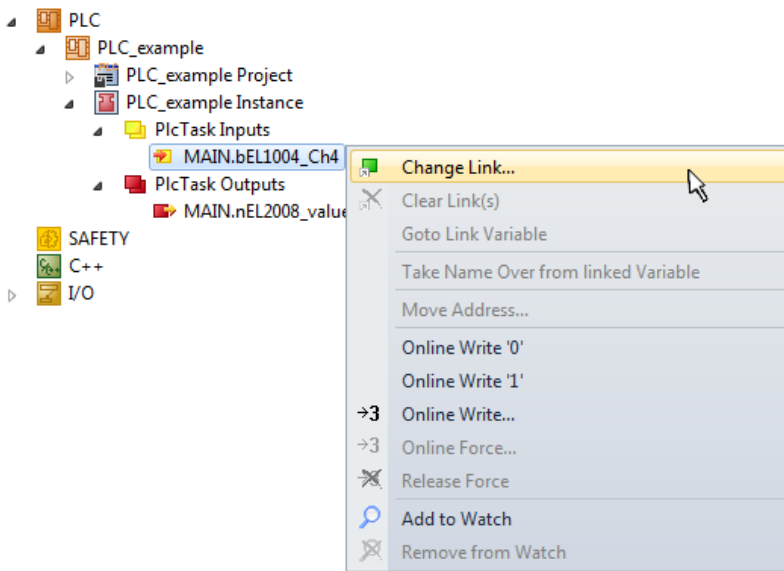


Fig. 57: Creating the links between PLC variables and process objects

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

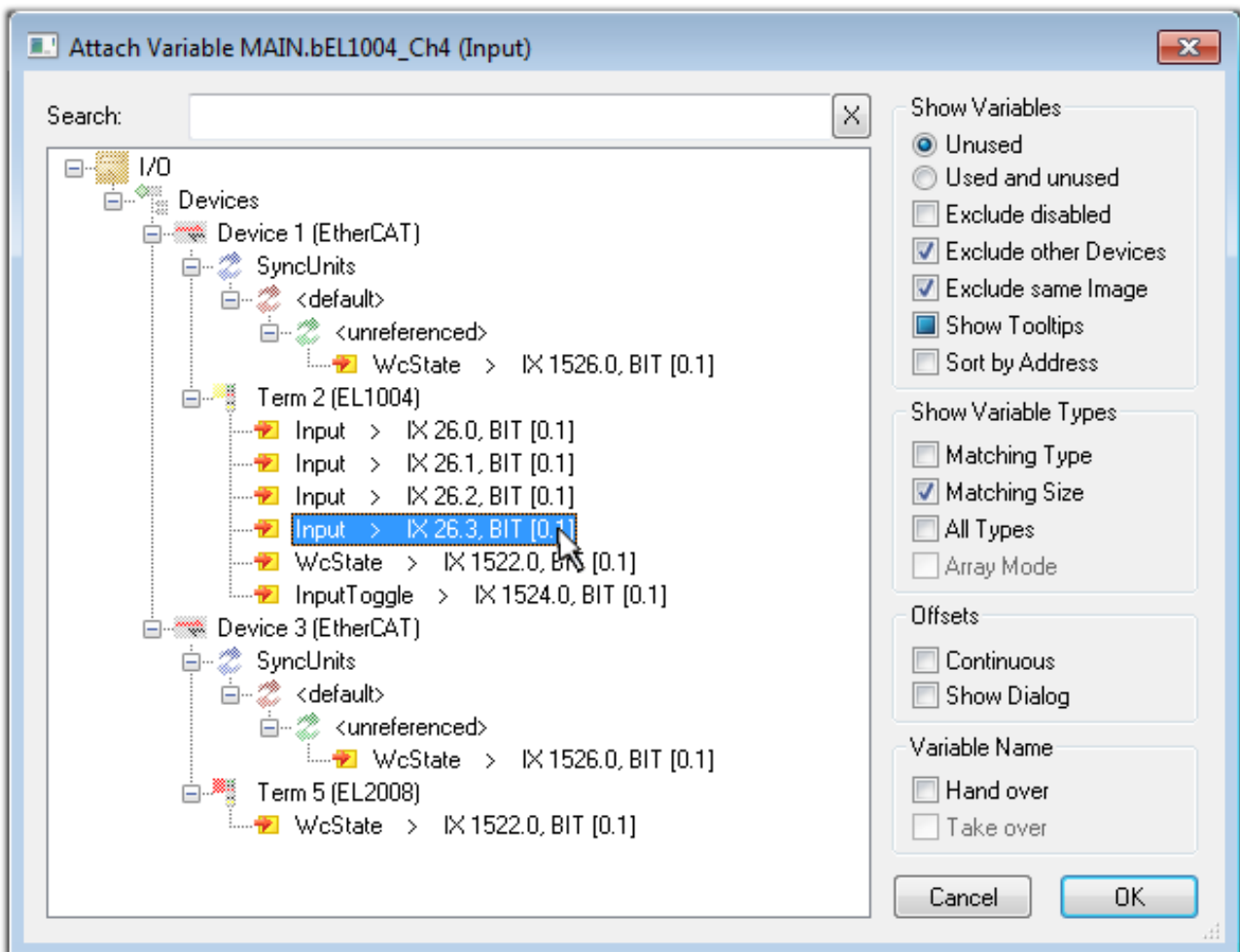


Fig. 58: Selecting PDO of type BOOL

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



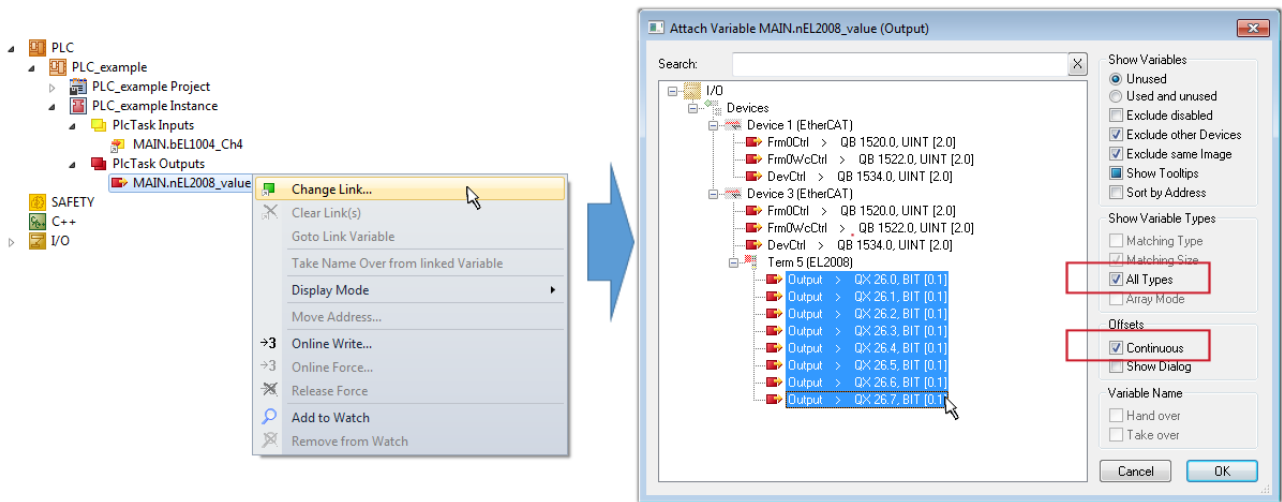



Fig. 59: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

Note that the "Continuous" checkbox was also activated. This is designed to allocate the bits contained in the byte of the variable "nEL2008\_value" sequentially to all eight selected output bits of the EL2008 terminal. In this way it is 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 (  ) at the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting a "Goto Link Variable" from the context menu of a variable. The object opposite, in this case the PDO, is automatically selected:

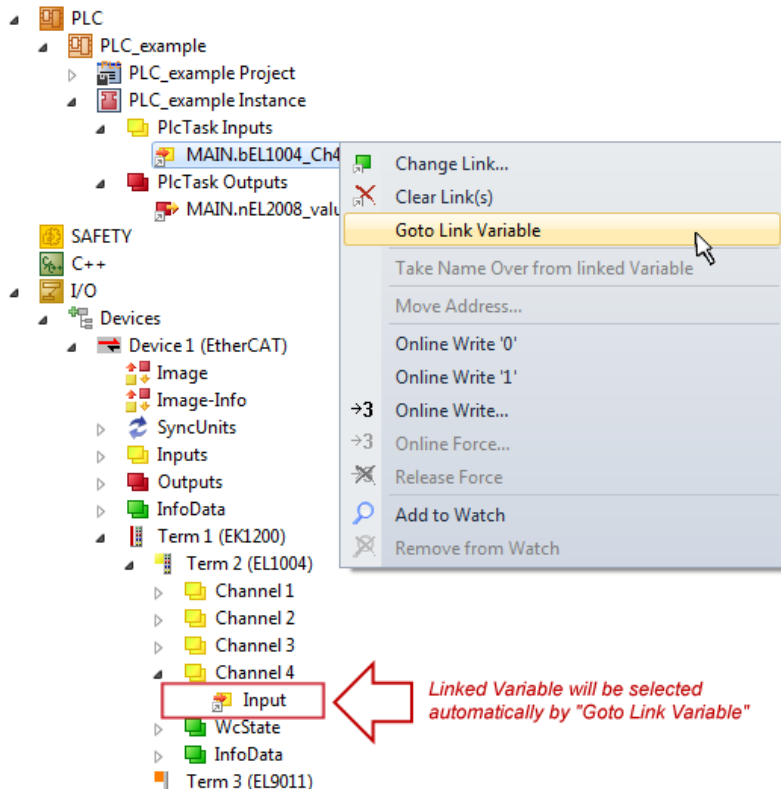


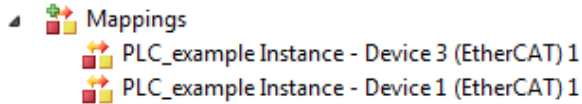
Fig. 60: Application of a "Goto Link" variable, using "MAIN.bEL1004\_Ch4" as a sample

The process of creating links can also take place in the opposite direction, i.e. starting with individual PDOs to variable. However, in this example it would then 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 possible to allocate this a set of bit-standardised variables (type "BOOL"). Here, too, a "Goto Link Variable" from the context menu of a PDO 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 with  or via the menu under "TwinCAT" in order to transfer settings of the development environment to the runtime system. Confirm the messages "Old configurations are 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

program start by click on symbol , the "F5" key or via "PLC" in the menu selecting "Start". The started programming environment shows the runtime values of individual variables:

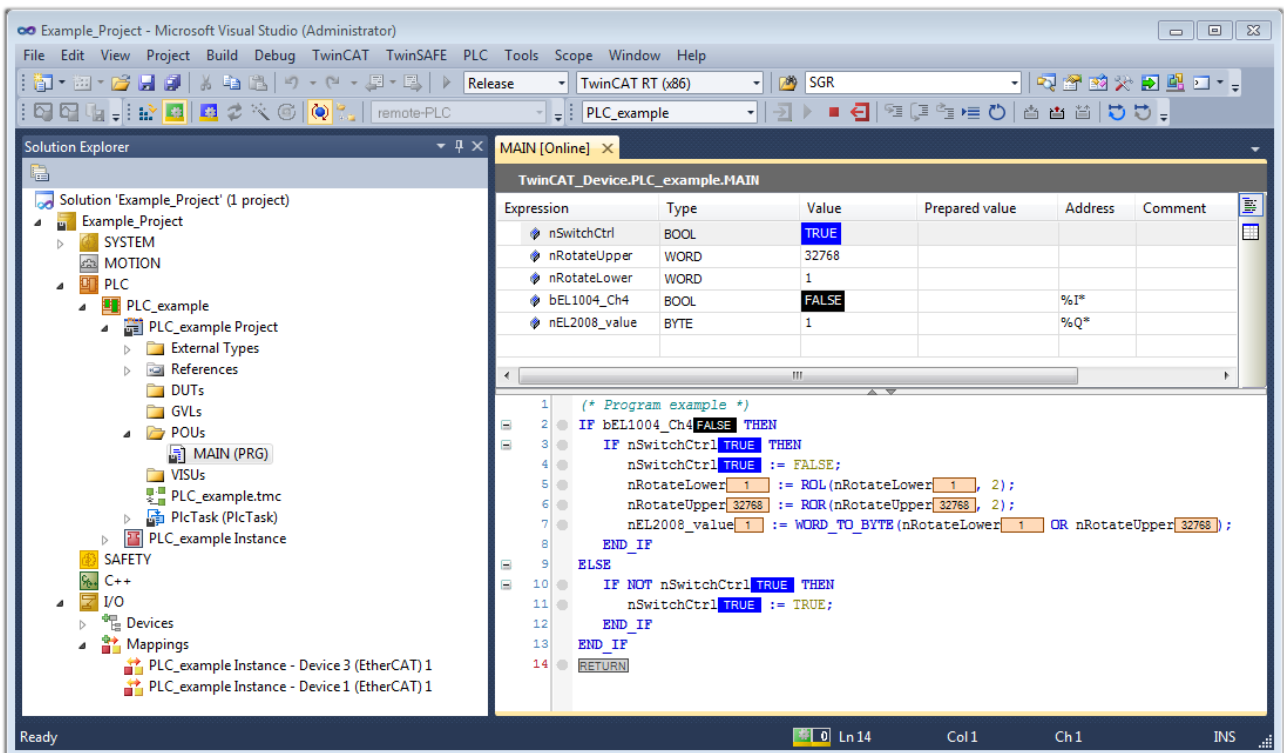


Fig. 61: TwinCAT development environment (VS shell): logged-in, after program startup

The two operator control elements for stopping  and logout  result in the required action (accordingly also for stop "Shift + F5", 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- und 64-Bit-Operatingsystem
  - 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. One option is described here.

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

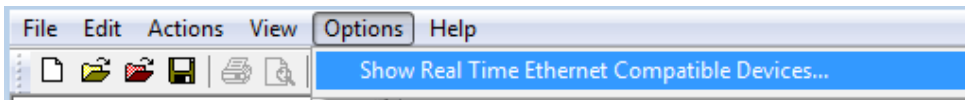


Fig. 62: System Manager "Options" (TwinCAT 2)

This has to be called up by the Menü "TwinCAT" within the TwinCAT 3 environment:

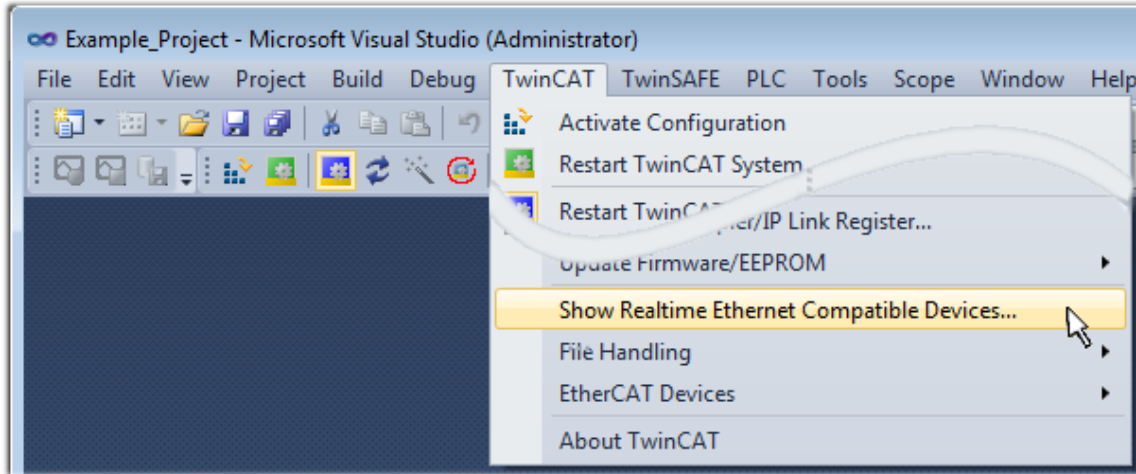


Fig. 63: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

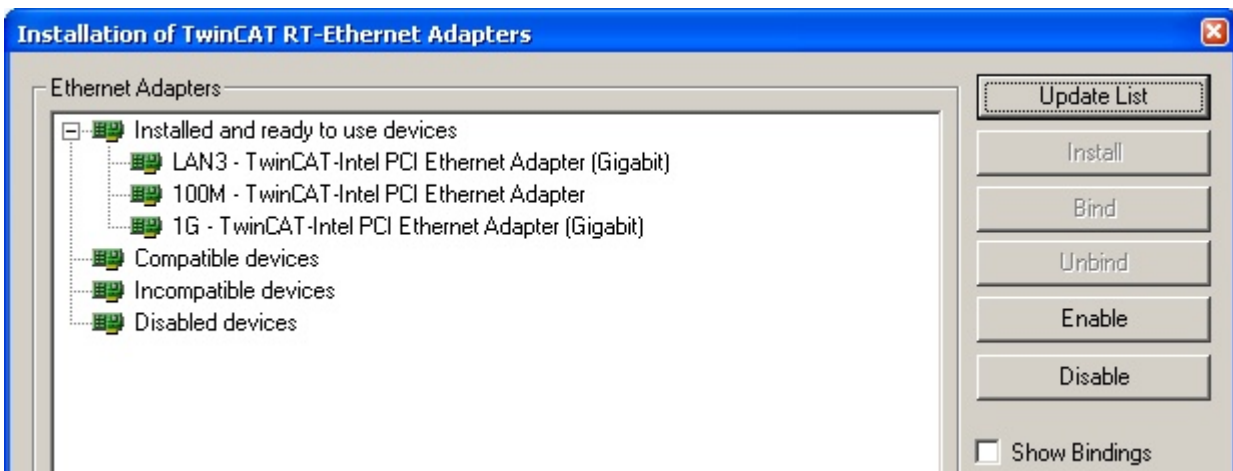


Fig. 64: Overview of network interfaces

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

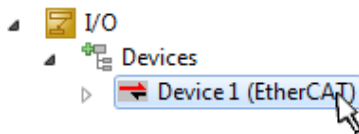
A Windows warning regarding the unsigned driver can be ignored.

**Alternatively** an EtherCAT-device can be inserted first of all as described in chapter [Offline configuration creation, section "Creating the EtherCAT device" \[► 70\]](#) in order to view the compatible ethernet ports via its EtherCAT properties (tab „Adapter“, button „Compatible Devices...“):



Fig. 65: 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)

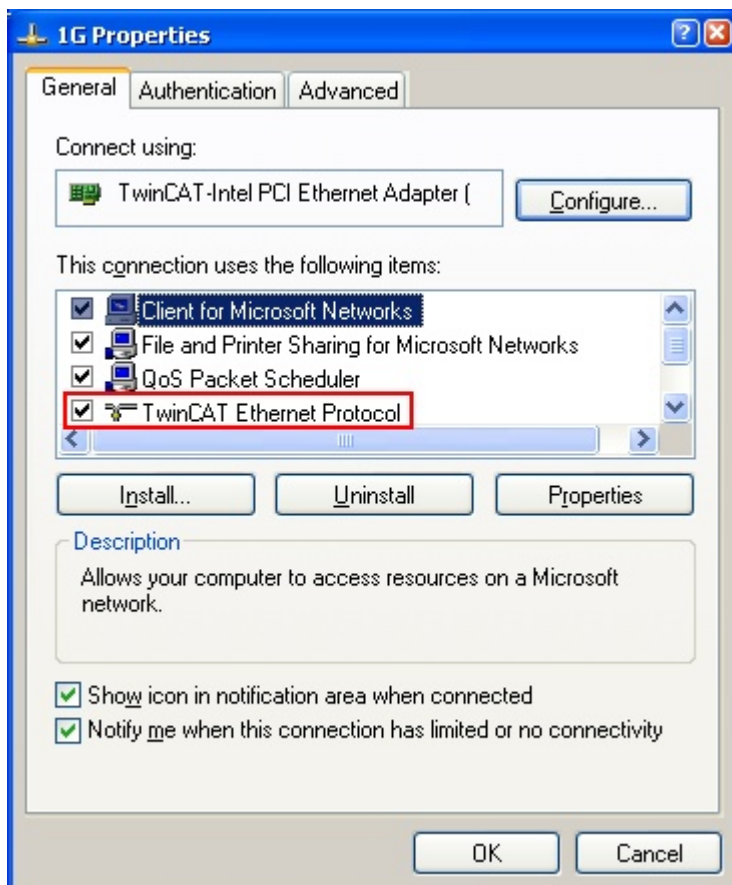


Fig. 66: Windows properties of the network interface

A correct setting of the driver could be:

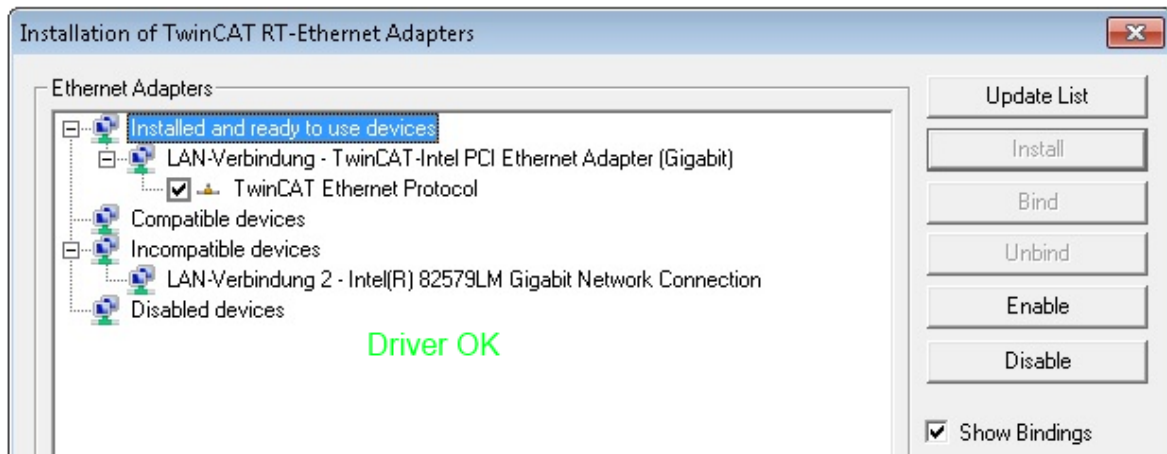


Fig. 67: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

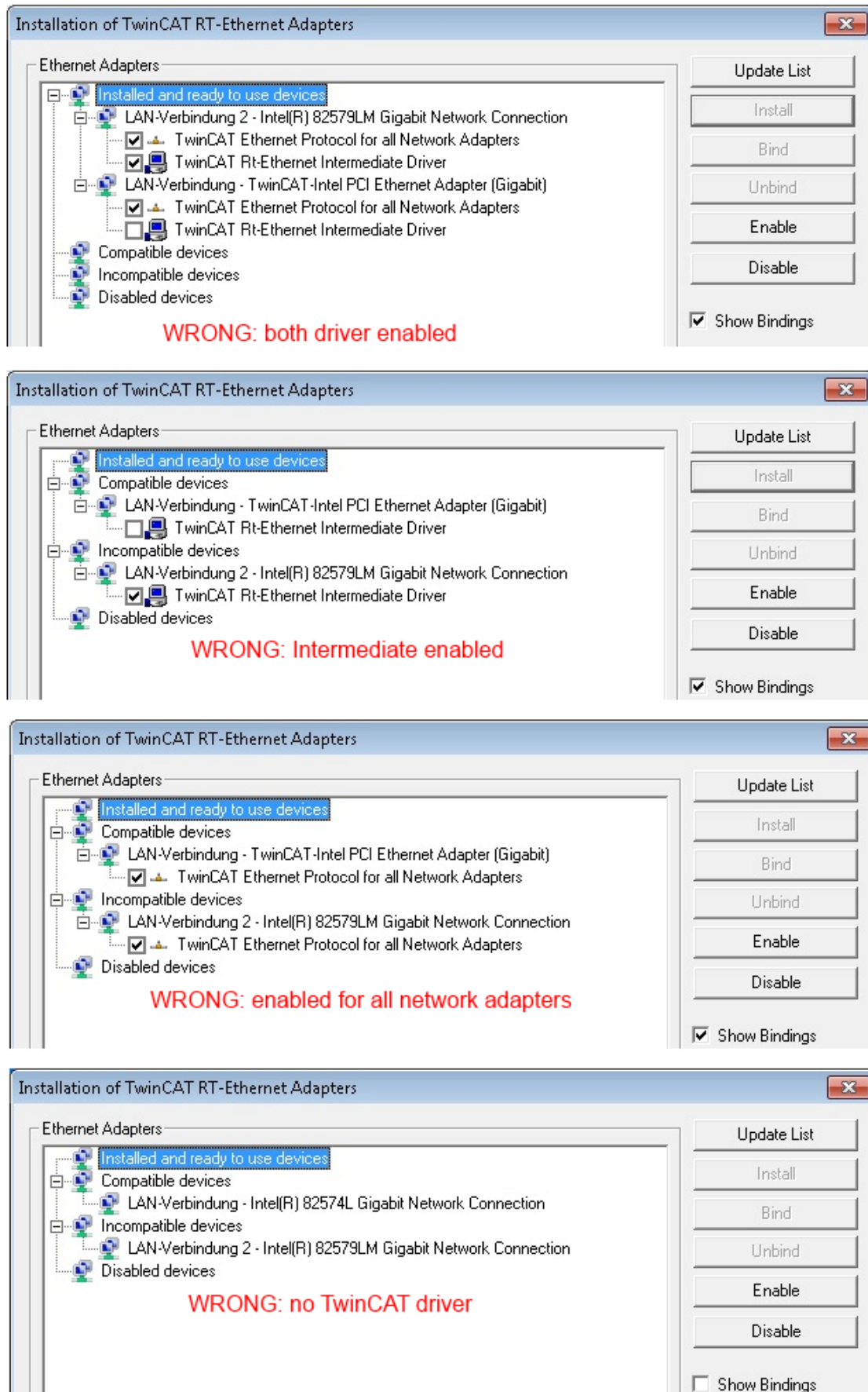


Fig. 68: Incorrect driver settings for the Ethernet port

## IP address of the port used

### **i** IP address/DHCP

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

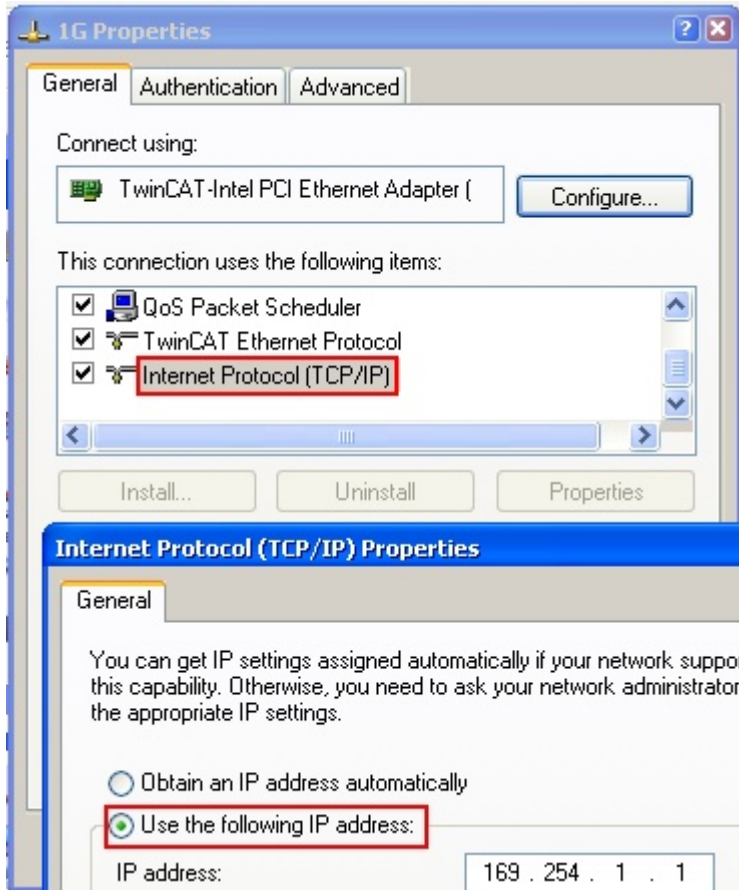


Fig. 69: TCP/IP setting for the Ethernet port



## 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 \[► 69\]](#) 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”

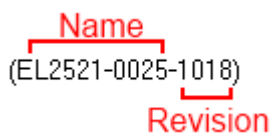


Fig. 70: Identifier structure

The order identifier consisting of name + type (here: EL2521-0010) 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 \[► 8\]](#).

## 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.

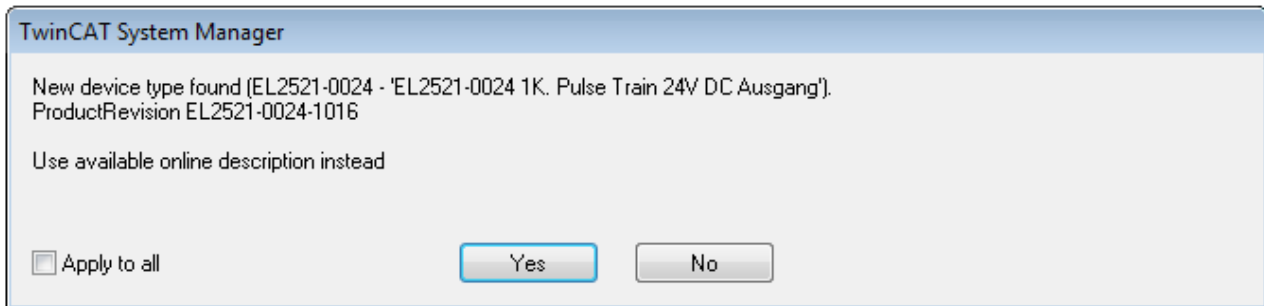


Fig. 71: *OnlineDescription information window (TwinCAT 2)*

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

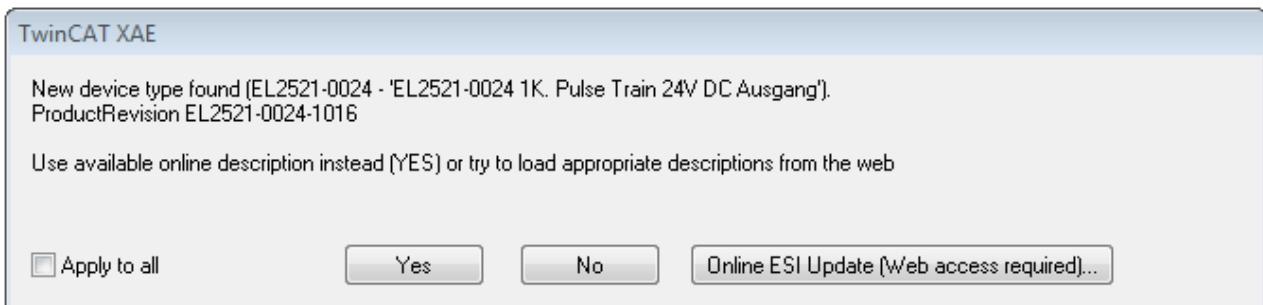


Fig. 72: *Information window OnlineDescription (TwinCAT 3)*

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.

### NOTE

#### 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](#)' [[▶ 70](#)].

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. 73: 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").

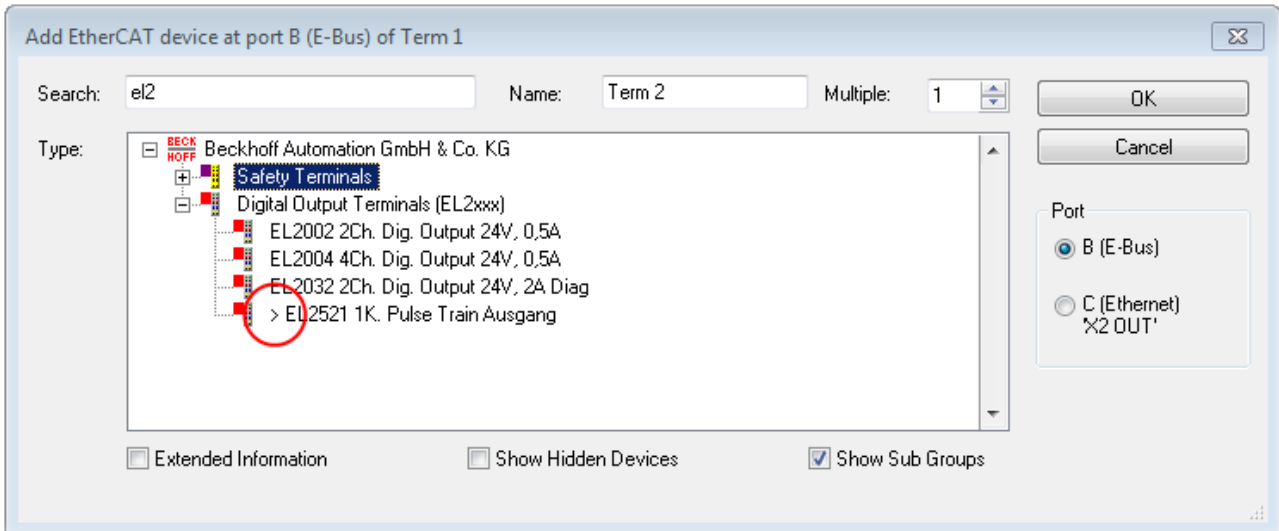


Fig. 74: Indication of an online recorded ESI of EL2521 as an example

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.

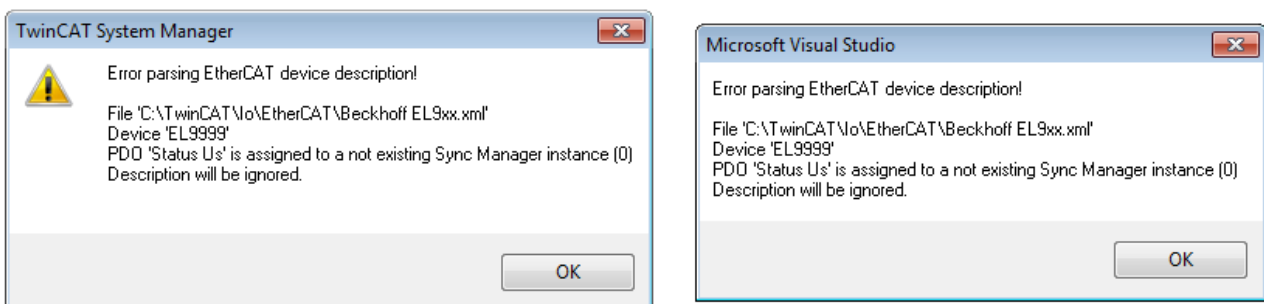


Fig. 75: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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:

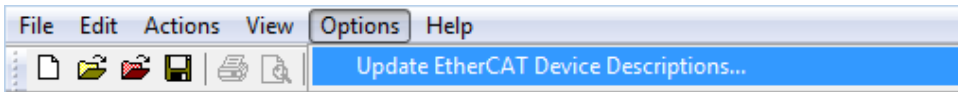


Fig. 76: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

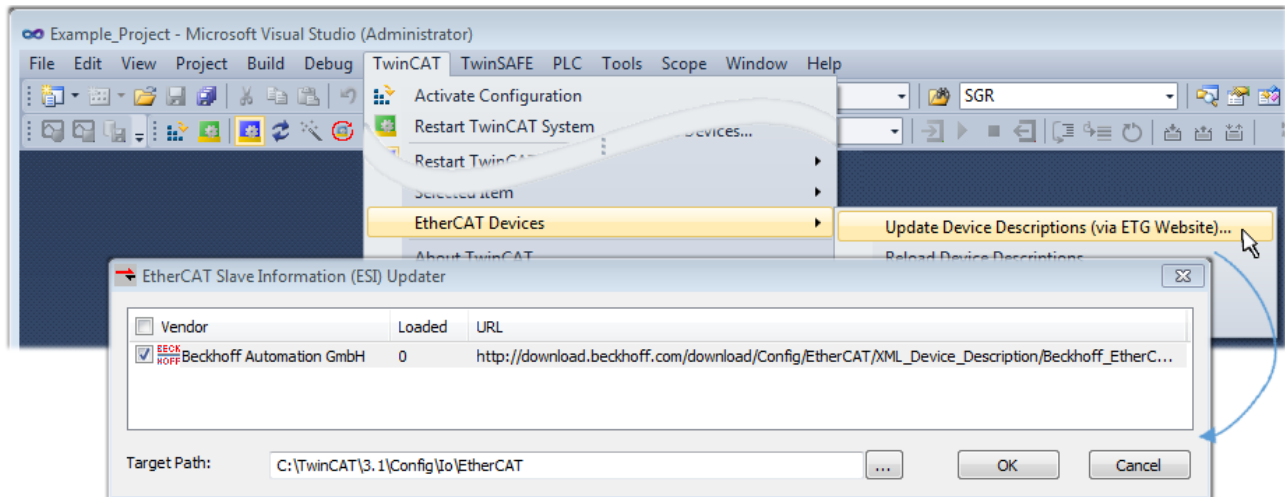


Fig. 77: Using the ESI Updater (TwinCAT 3)

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” [▶ 65].

**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 [▶ 75] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 76]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 79]

The scan with existing configuration [▶ 80] 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.

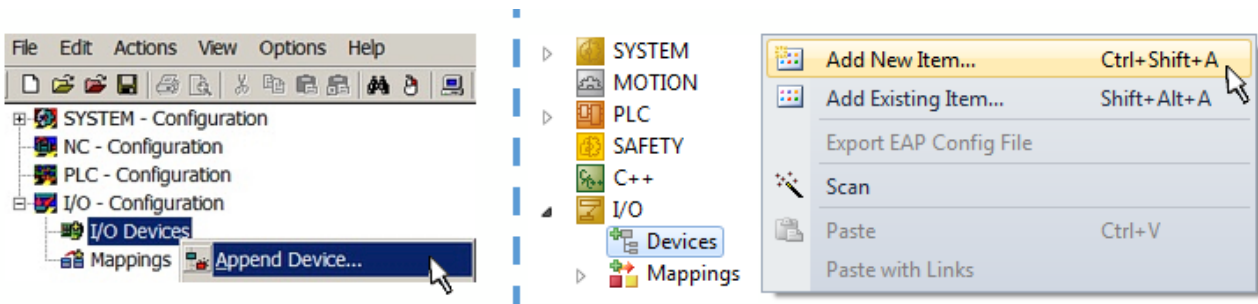


Fig. 78: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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".

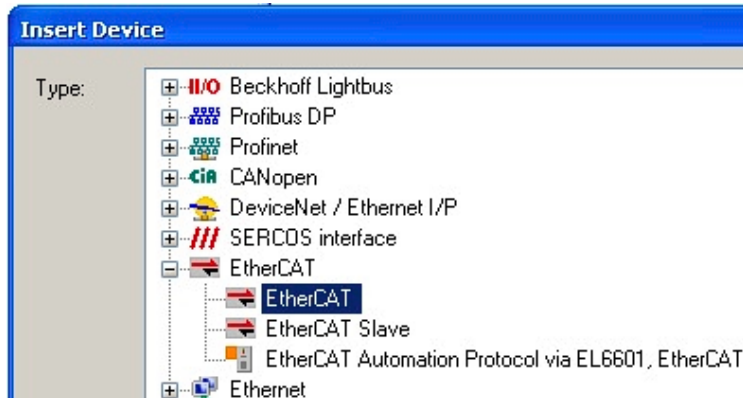


Fig. 79: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

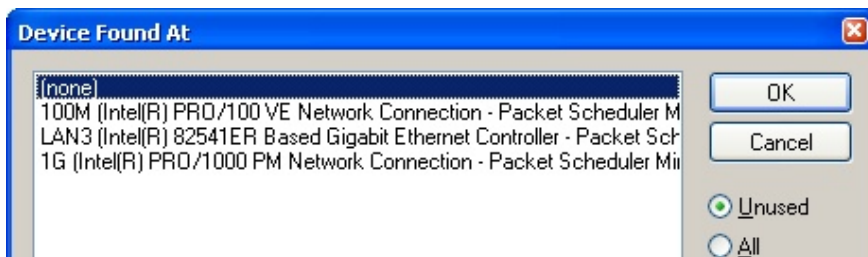


Fig. 80: Selecting the Ethernet port

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)”.

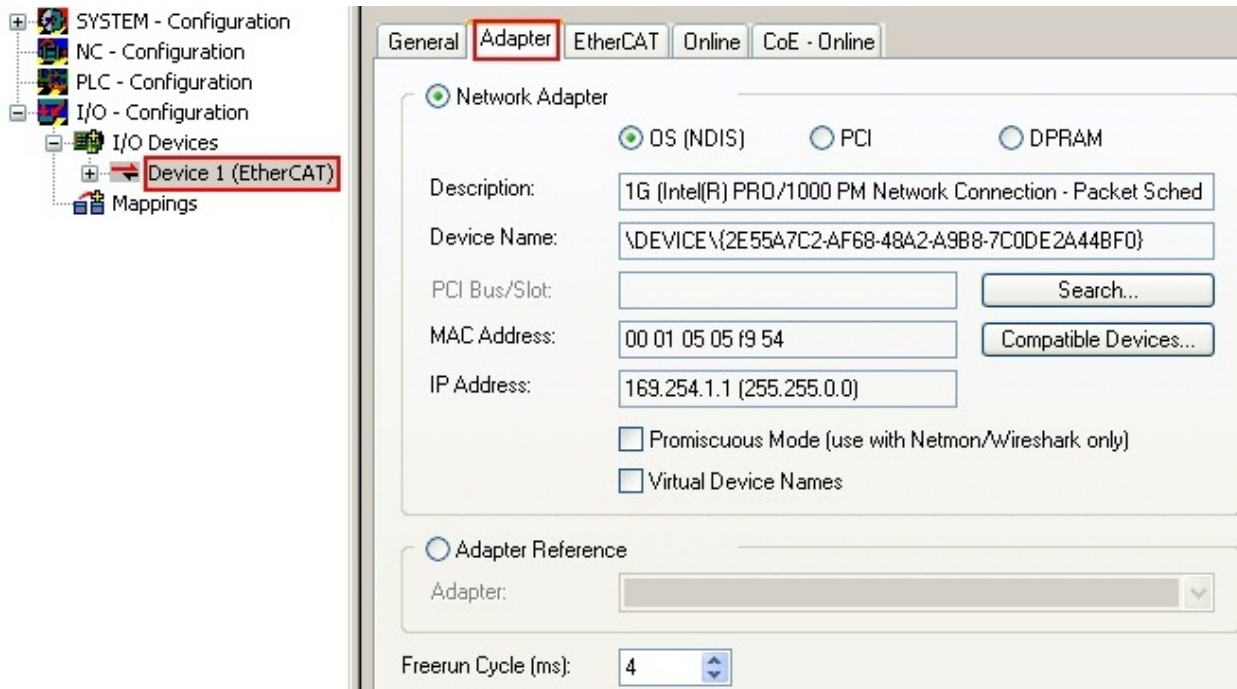
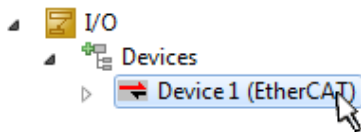


Fig. 81: EtherCAT device properties (TwinCAT 2)

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 \[▶ 59\]](#).

**Defining EtherCAT slaves**

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

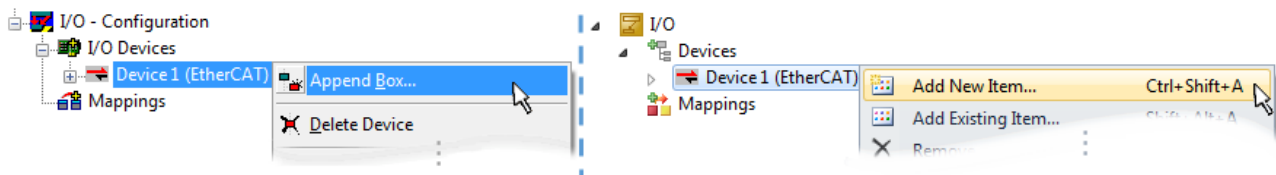


Fig. 82: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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: EK couplers, EP boxes, devices with RJ45/M8/M12 connector

- “E-Bus”: LVDS “terminal bus”, “EJ-module”: EL/ES terminals, various modular modules

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

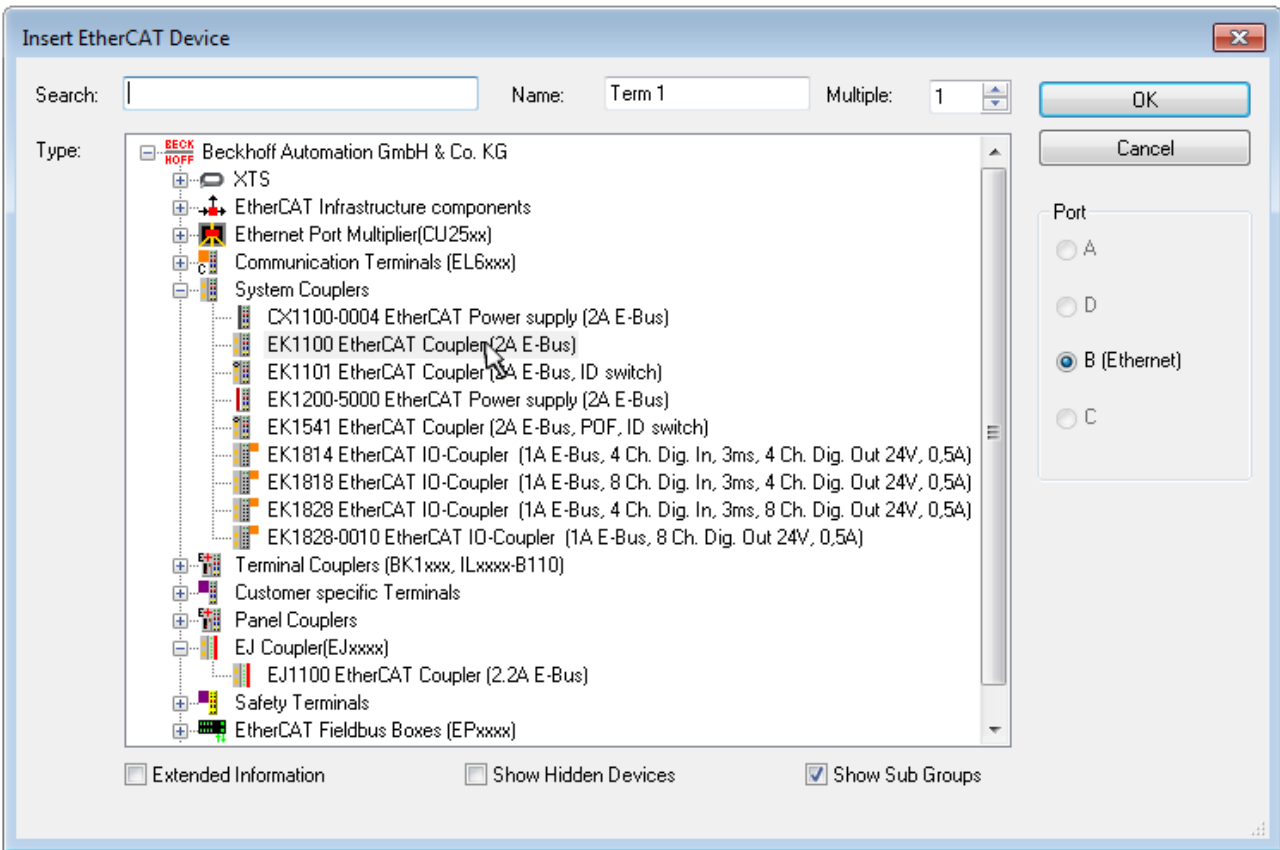


Fig. 83: Selection dialog for new EtherCAT device

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”.

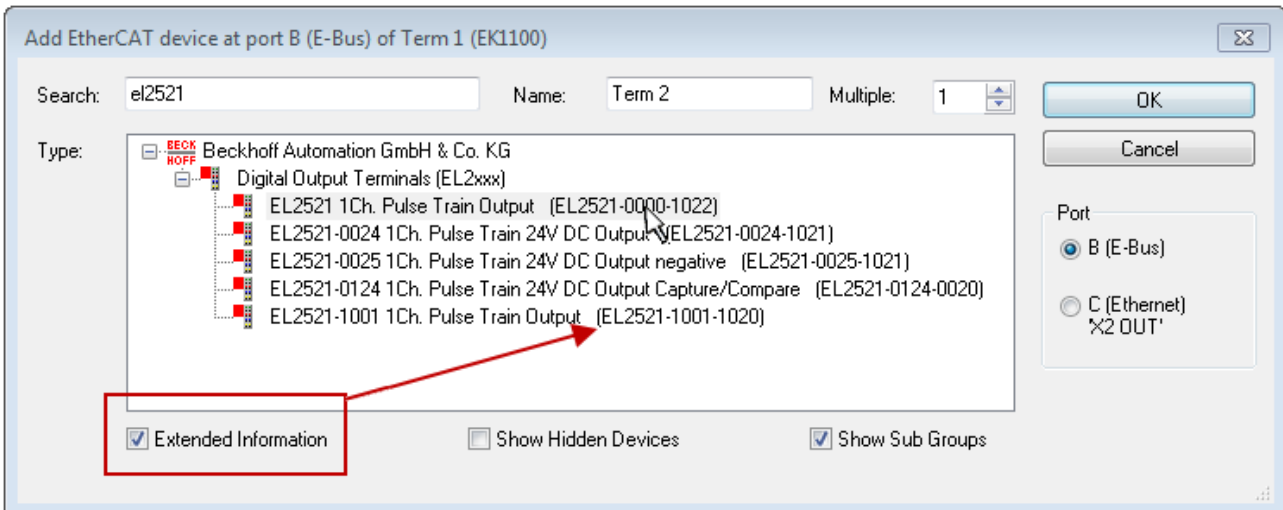


Fig. 84: Display of device revision

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”.



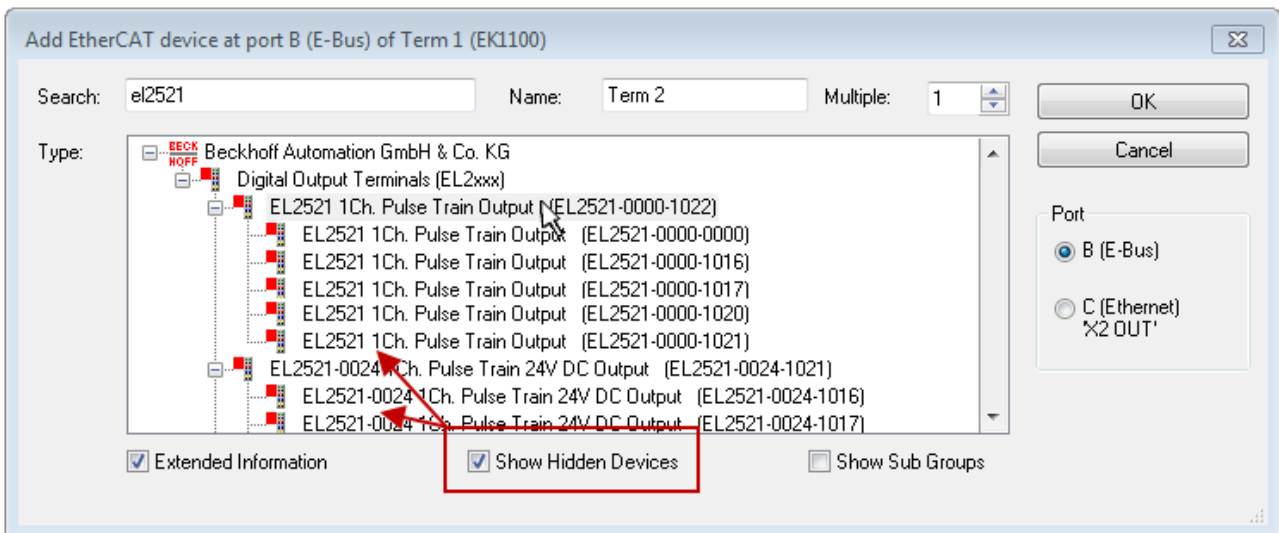


Fig. 85: Display of previous revisions

**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.

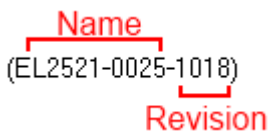


Fig. 86: Name/revision of the terminal

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 parameterised as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

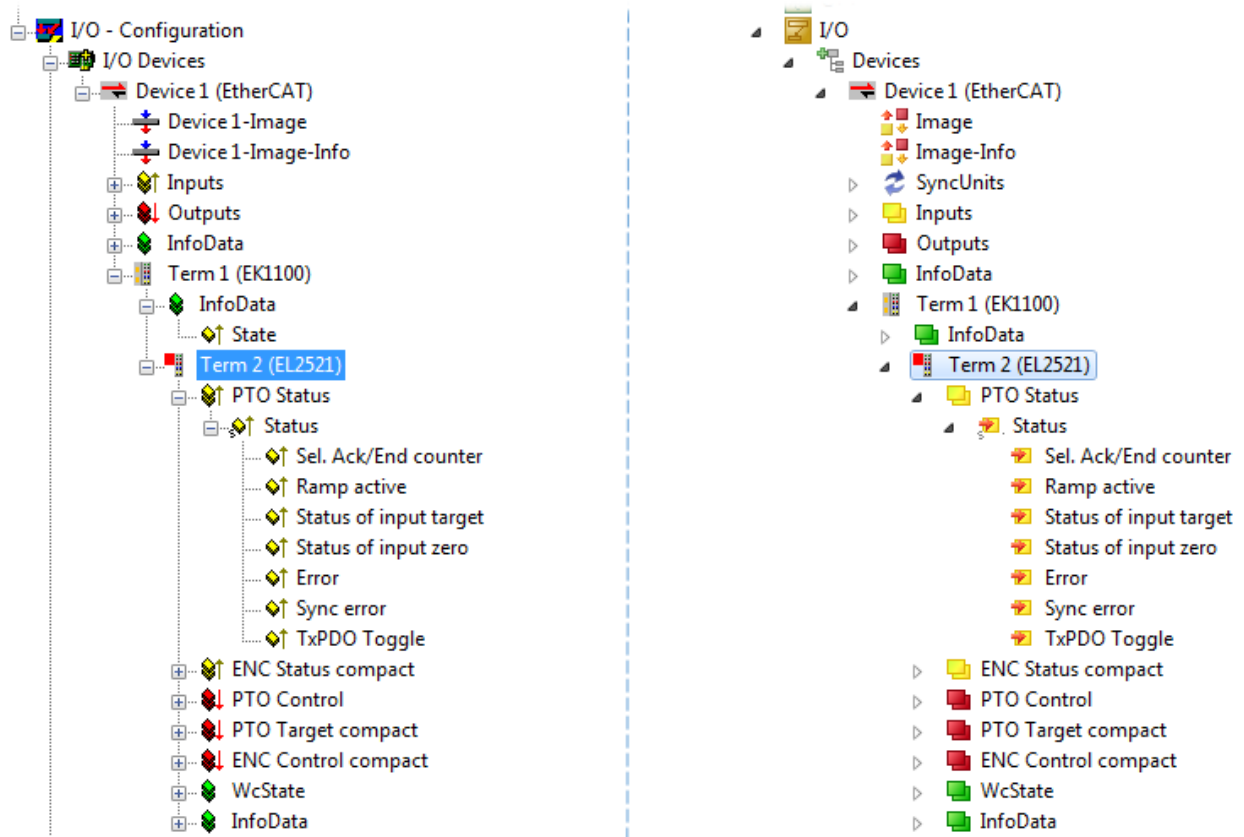




Fig. 87: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



## 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.

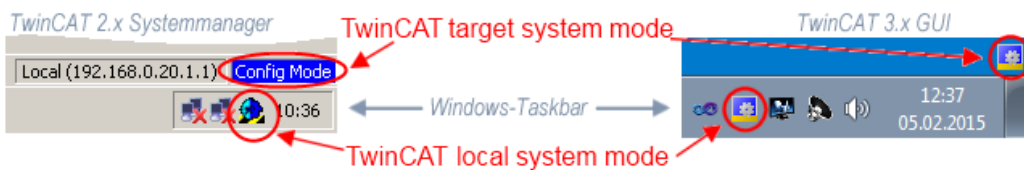


Fig. 88: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

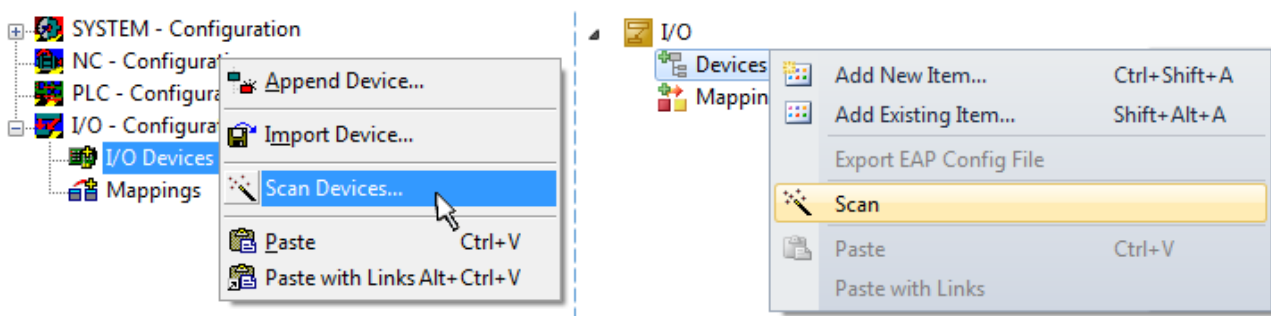


Fig. 89: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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

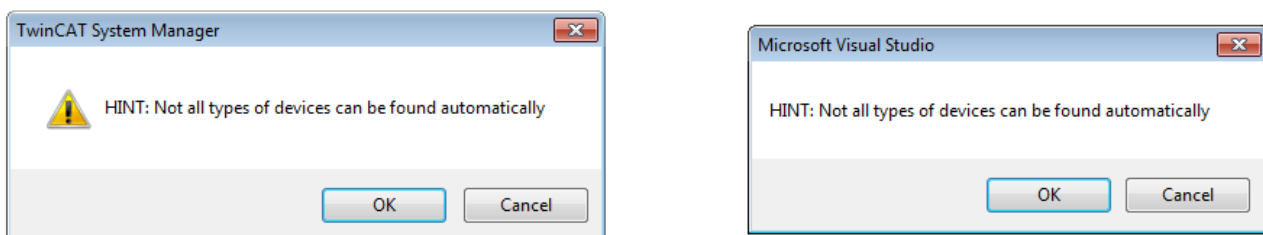


Fig. 90: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

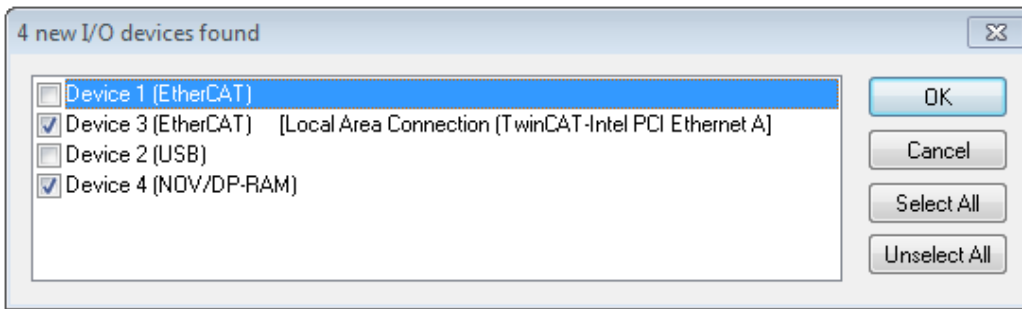


Fig. 91: Detected Ethernet devices

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](#) [▶ 59].

## 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. 92: Example default state

## NOTE

### 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](#) [▶ 80] 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:

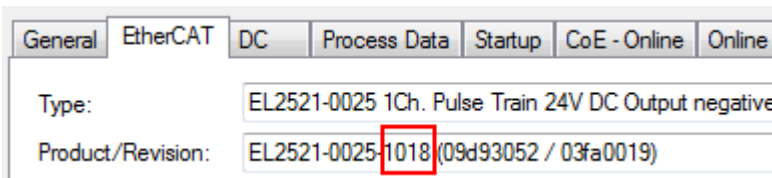


Fig. 93: Installing EtherCAT terminal with revision -1018

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

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

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and a **new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of 'B.tsm' or even 'B.pro' is therefore unnecessary. The series-produced machines can continue to be built with 'B.tsm' and 'B.pro'; it makes sense to perform a comparative scan [▶ 80](#) 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:

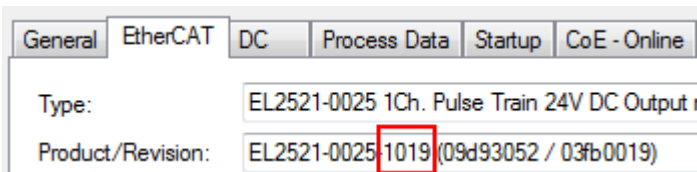


Fig. 94: Detection of EtherCAT terminal with revision -1019

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since virtually a new configuration is 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. 95: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

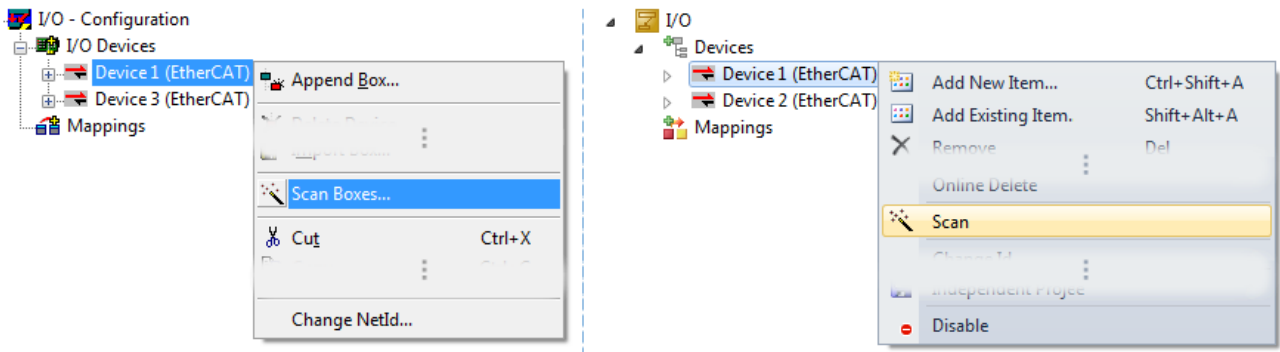


Fig. 96: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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. 97: Scan progress exemplarily by TwinCAT 2

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



Fig. 98: 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. 99: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 100: 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”.

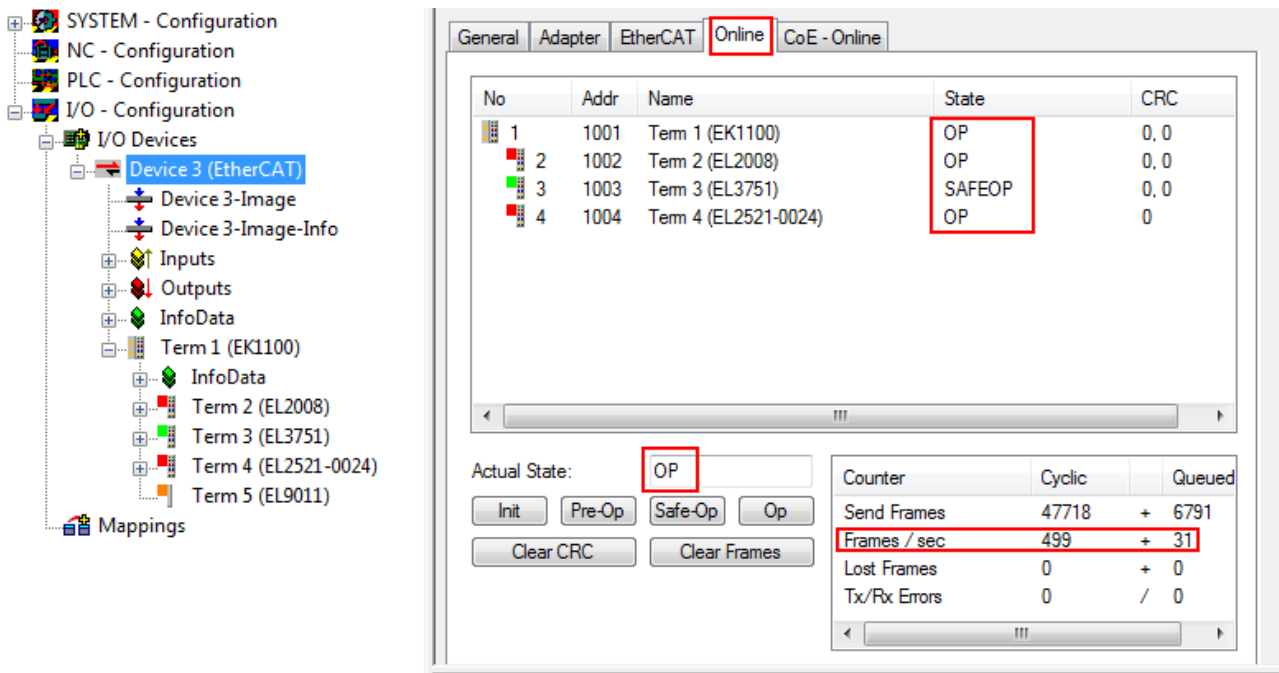


Fig. 101: Online display example

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 \[► 70\]](#).

### 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.

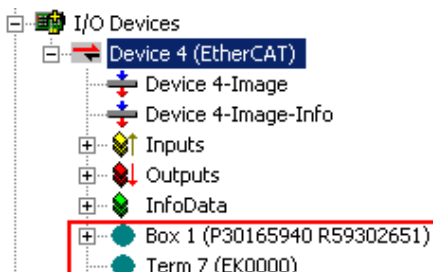


Fig. 102: Faulty identification

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**

**NOTE**

**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. 103: 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.

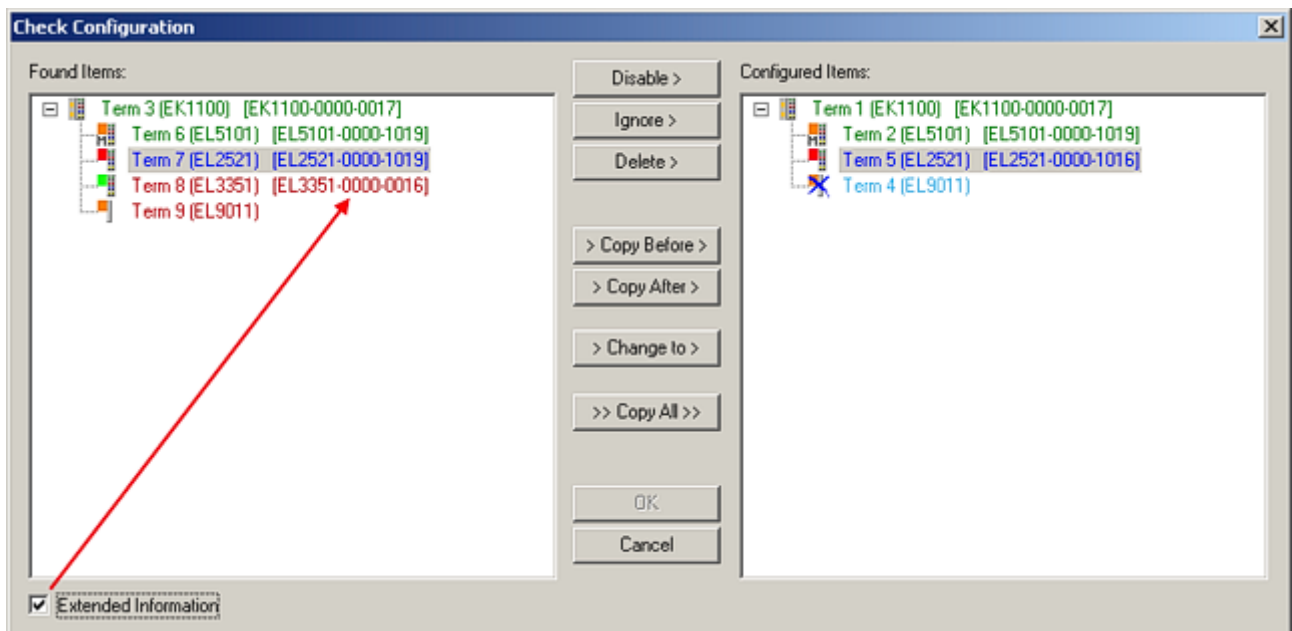


Fig. 104: Correction dialog

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



Colour	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>

**● Device selection based on revision, compatibility**

**i** 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.

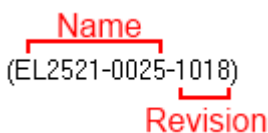


Fig. 105: Name/revision of the terminal

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 parameterised as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

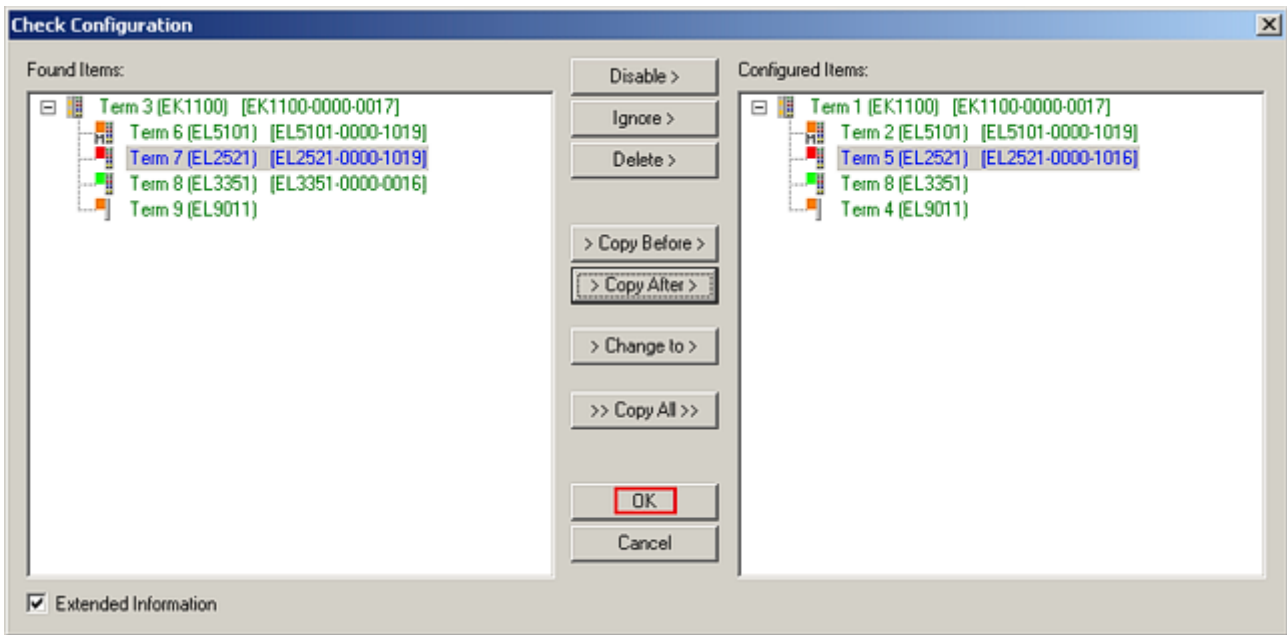


Fig. 106: Correction dialog with modifications

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.

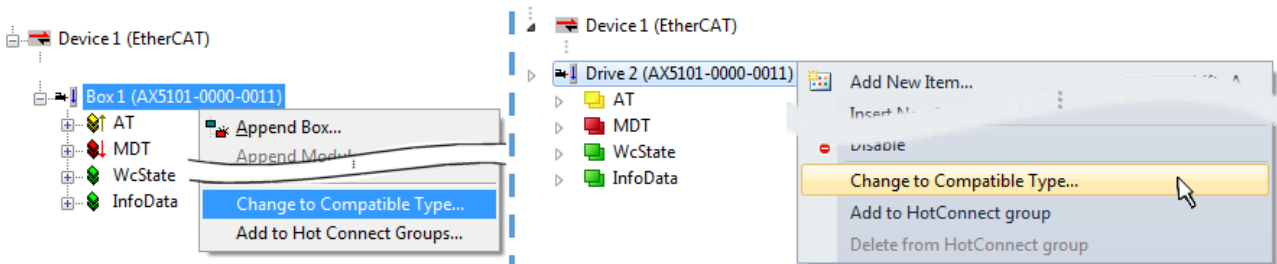


Fig. 107: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

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*

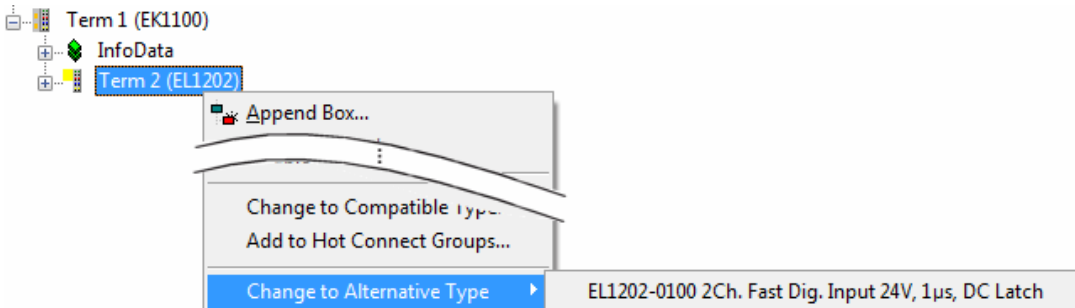


Fig. 108: TwinCAT 2 Dialog Change to Alternative Type

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).

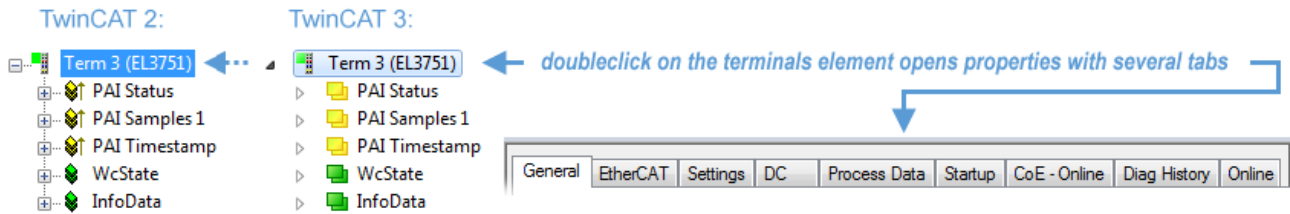


Fig. 109: Branch element as terminal EL3751

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

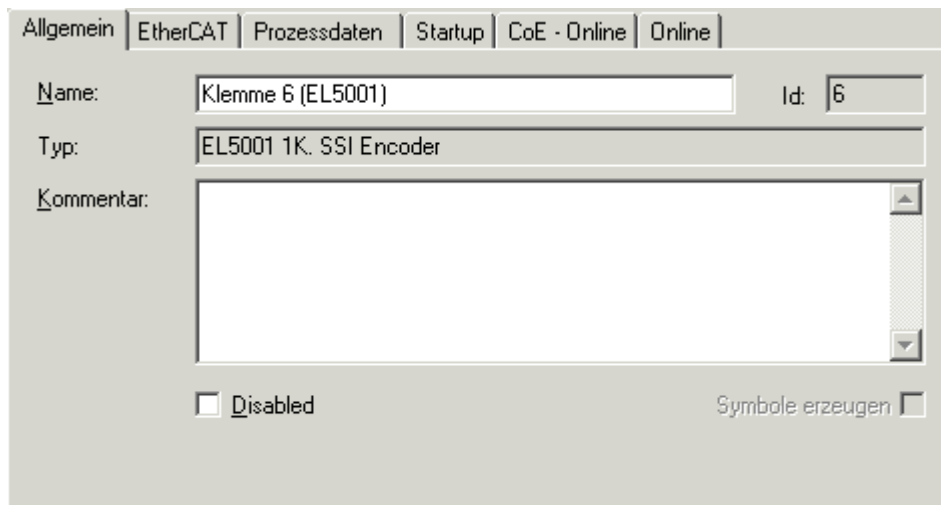


Fig. 110: "General" tab

<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**

The screenshot shows the 'EtherCAT' configuration tab with the following fields and values:

- Typ:** EL5001 1K. SSI Encoder
- Produkt / Revision:** EL5001-0000-0000
- Auto-Inc-Adresse:** FFF8
- EtherCAT-Adresse:**  1006 Weitere Einstellungen...
- Vorgänger-Port:** Klemme 5 (EL5001) - B

At the bottom of the tab, there is a link: <http://www.beckhoff.de/german/default.htm?EtherCAT/EL5001.htm>

Fig. 111: „EtherCAT“ tab

**Type**

EtherCAT device type

**Product/Revision**

Product and revision number of the EtherCAT device

**Auto Inc Addr.**

Auto increment address of the EtherCAT device. The auto increment address can be used for addressing each EtherCAT device in the communication ring through its physical position. Auto increment addressing is used during the start-up phase when the EtherCAT master allocates addresses to the EtherCAT devices. With auto increment addressing the first EtherCAT slave in the ring has the address  $0000_{\text{hex}}$ . For each further slave the address is decremented by 1 ( $FFFF_{\text{hex}}$ ,  $FFFE_{\text{hex}}$  etc.).

**EtherCAT Addr.**

Fixed address of an EtherCAT slave. This address is allocated by the EtherCAT master during the start-up phase. Tick the control box to the left of the input field in order to modify the default value.

**Previous Port**

Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combination field is activated and the EtherCAT device to which this device is to be connected can be selected.

**Advanced Settings**

This button opens the dialogs for advanced settings.

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

**“Process Data” tab**

Indicates the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (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.

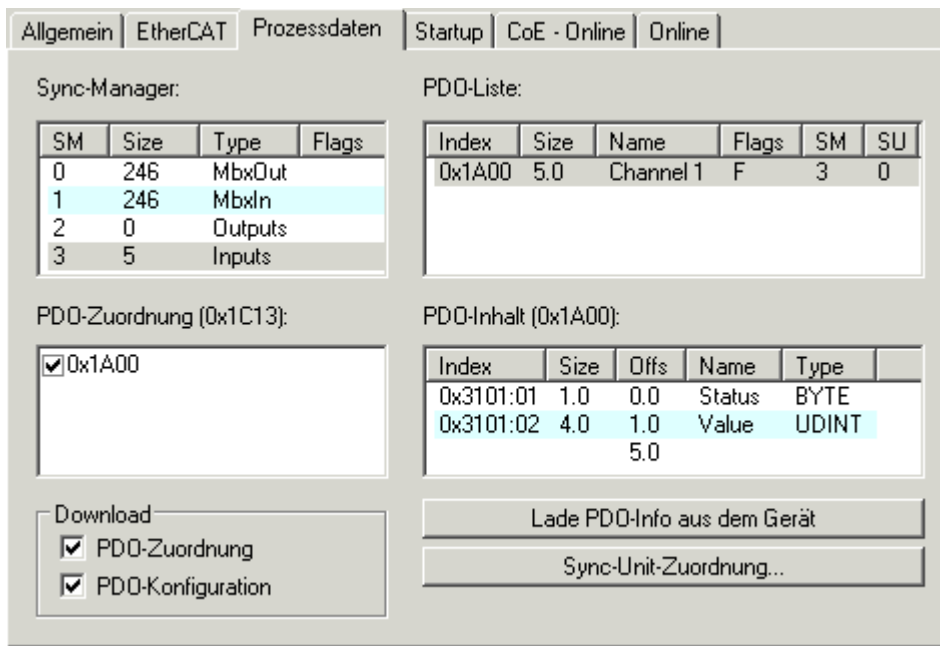


Fig. 112: "Process Data" tab

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").

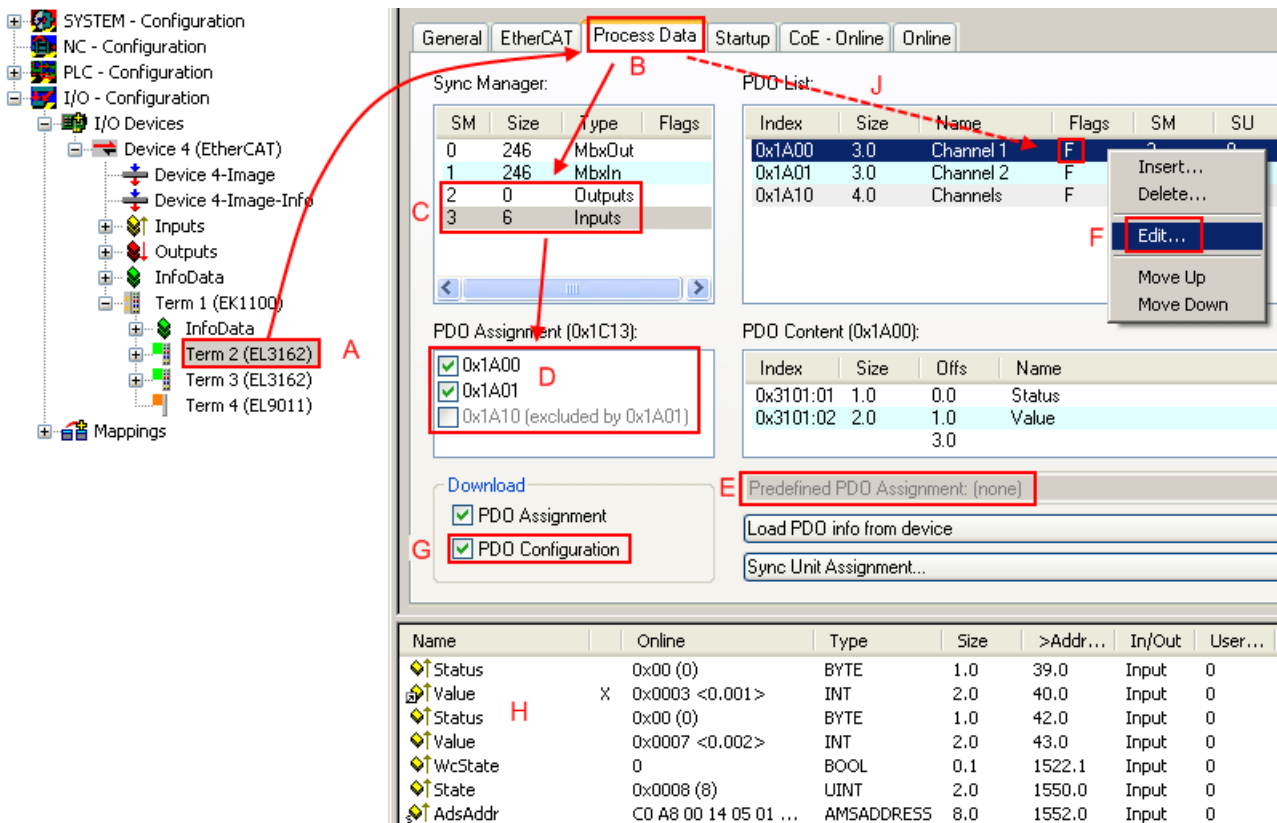


Fig. 113: Configuring the process data

**Manual modification of the process data**

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

A detailed description [▶ 91] 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.

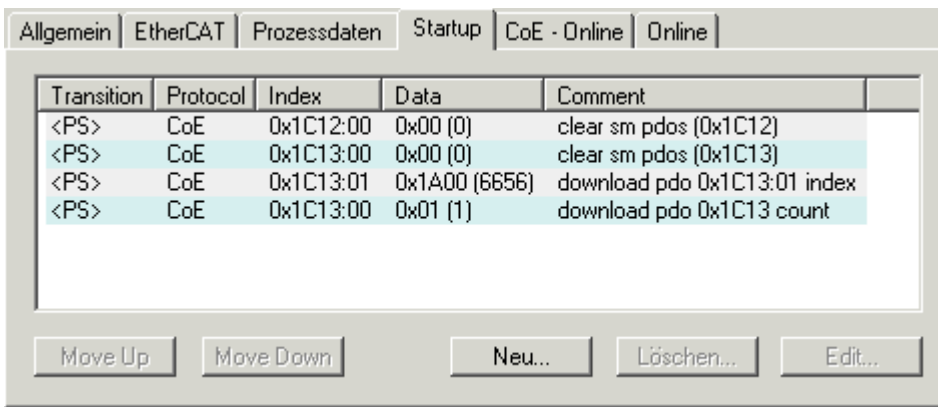


Fig. 114: „Startup“ tab

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.

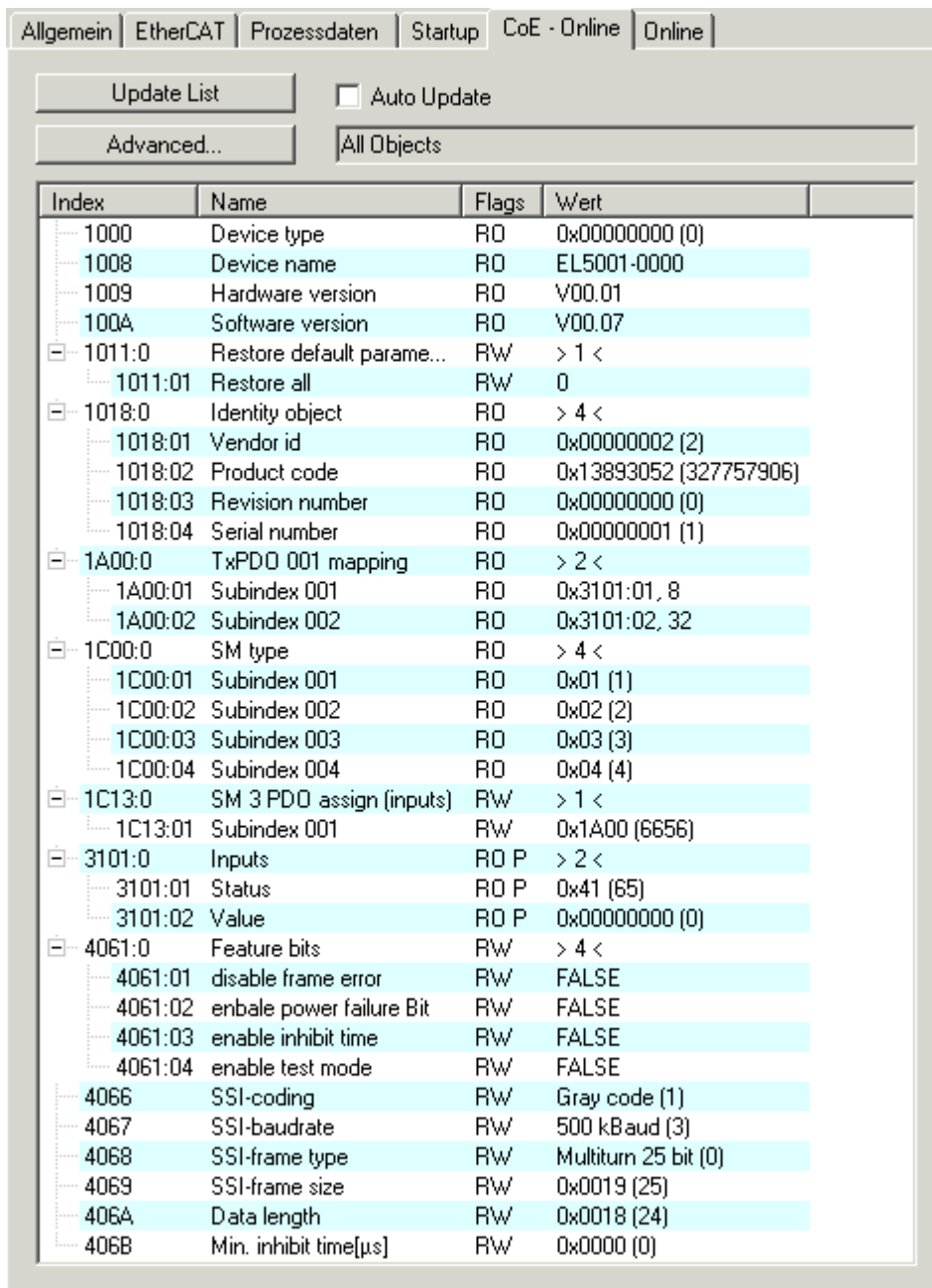


Fig. 115: "CoE – Online" tab

### 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.



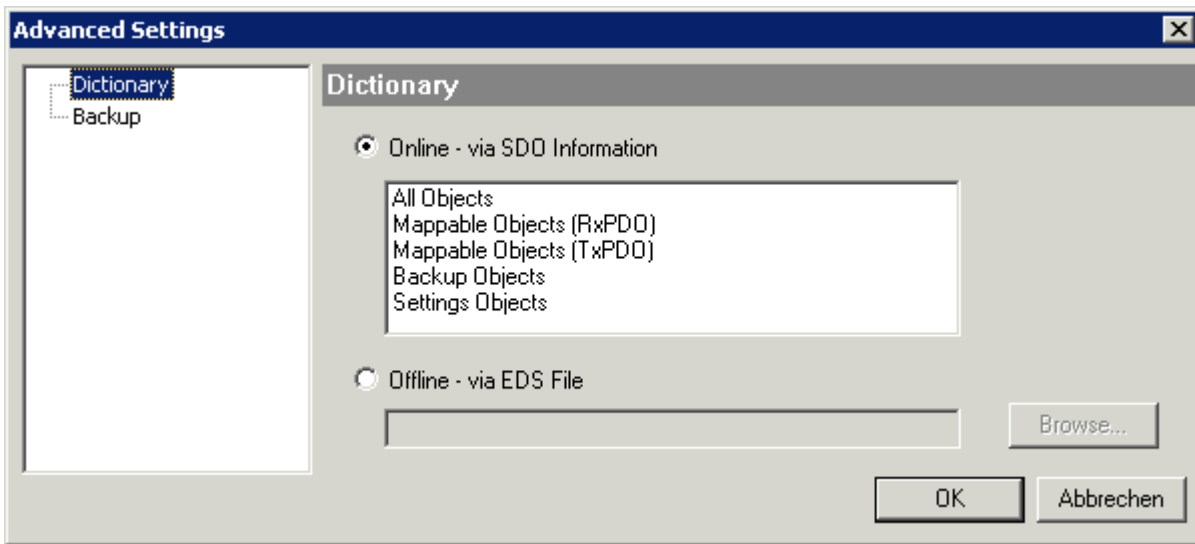


Fig. 116: Dialog “Advanced settings”

**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**

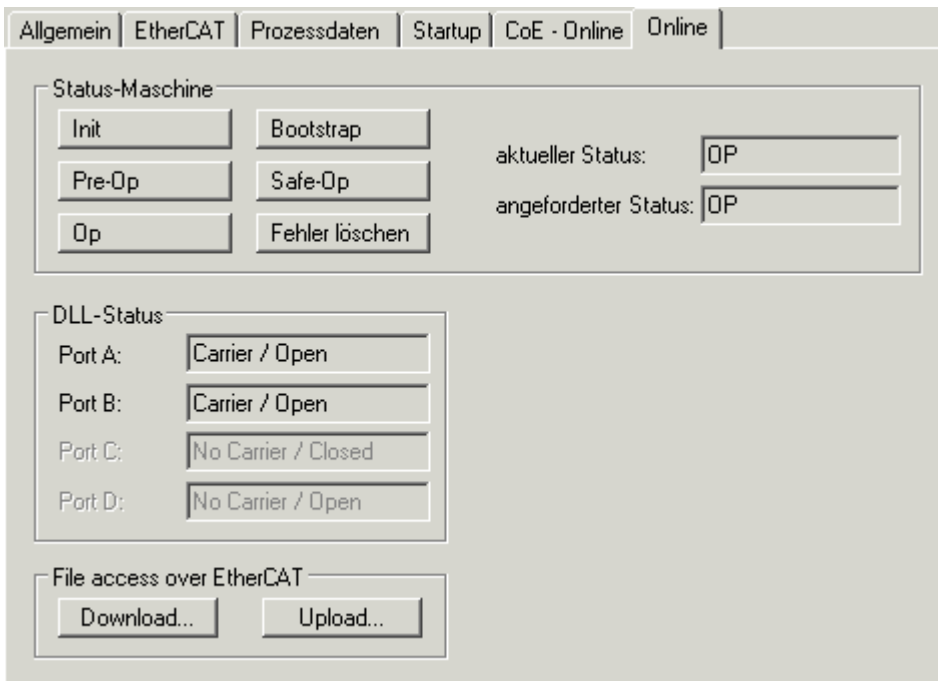


Fig. 117: „Online“ tab

**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)**

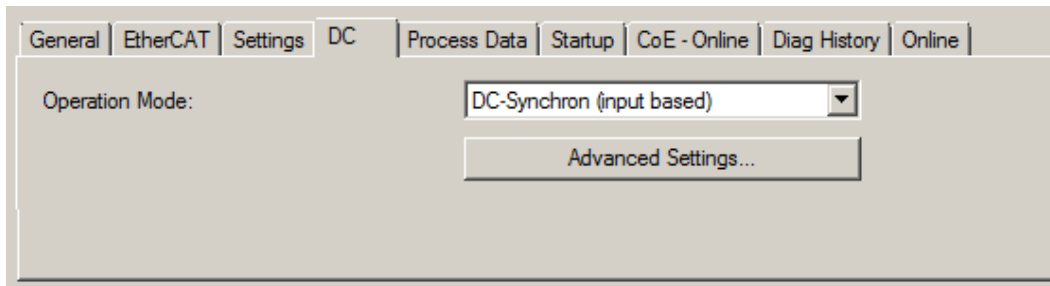


Fig. 118: "DC" tab (Distributed Clocks)

- 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 are 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.

#### ● **i** 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 \[▶ 89\]](#)),
  - 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 \[► 86\]](#) 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 Basic function principles

### EL6652(-0000) EtherNet/IP Master

The EL6652 EtherNet/IP-Master Terminal has a switched 2-port Ethernet connection and can thus be operated in a line with further Ethernet/IP nodes. The process data are configured by an EtherCAT master, allowing different process data and sizes. The EL6652 supports both multicast and unicast connections. Up to 16 simple EtherNet/IP-Slave devices can be connected via one generic node. The EL6652 is optionally also available as a slave version.

#### Configuration

The configuration is carried out in several steps. The terminal must be contained in the EtherCAT master (e.g. TwinCAT) as EL6652 in the EtherCAT tree. Furthermore an additional device must be created in order to provide this terminal with the parameters and with configuration data ([see Configuration \[▶ 96\]](#) and [see EDS file \[▶ 104\]](#)).

### EL6652-0010 EtherNet/IP Slave

The EtherNet/IP Slave terminal enables the exchange of data with an EtherNet/IP scanner or a master. Both Multicast and Unicast are supported. In addition, the terminal can behave like two EtherNet/IP slave devices. The second slave is a virtual slave. This enables you to connect two masters together or one master with two slaves and, for example, to transport more data or to operate with different polling times on the master. The maximum process data in both directions is 1 kbyte.

The terminal is an EtherCAT slave device and is configured via the EtherCAT master. The IP settings as well as the number and types of data items are defined here. The terminal behaves on the EtherNet/IP side as a slave device in the EtherNet/IP network. No further configuration is necessary via the EtherNet/IP master. A connection can only be successfully established if the EtherCAT master is running and in data exchange mode; only then can the EtherNet/IP master also establish a connection.

#### Configuration

The configuration takes place in several steps. The terminal must be contained in the EtherCAT master (e.g. TwinCAT) as EL6652-0010 in the EtherCAT tree. Furthermore an additional device must be created in order to provide this terminal with the parameters and with configuration data ([see Configuration \[▶ 110\]](#) and [Parameters \[▶ 115\]](#)). Once this is completed and the EtherCAT master is in data exchange mode, the configuration takes place in the EtherNet/IP Master/Scanner.



#### **EtherCAT XML device description and configuration files**

We recommend downloading the latest [EtherCAT Device Description](#) from the download area on the Beckhoff website and installing it according to the installation instructions.

---

The process image is freely configurable. In this way it is possible to reduce the process data for internal application to a minimum.

## 5.4 Changing EtherNet/IP settings

With the following products the settings can also be changed by ADS.

- TS6280-xxxx | TwinCAT EtherNet/IP Slave (excluding Legacy Device), from TwinCAT Build 2249
- TS6281-xxxx | TwinCAT EtherNet/IP Master, from TwinCAT Build 2249
- EL6652 | EtherNet/IP Master Terminal (from software version 01V0.36)
- EL6652-0010 | EtherNet/IP Slave Terminal (from software version 01V0.36)

The **Store Category** must be defined in the System Manager for the setting. This is entered in the object 0xF800:2B "Advanced options" for all EtherNet/IP devices in the scanner or, for the slave interface, in object 0x8000:2B or 0x8010:2B for the optional 2<sup>nd</sup> slave.

If the corresponding flag is set, the IP address from the memory is used; if nothing is entered there the flag is ignored and parameters from the System Manager are used.

Index	Name	Flags	Value
[-] F800:0	Master Settings	M RO	> 43 <
F800:01	Number	M RO	0x0002 (2)
F800:03	Product Name	M RW	Device 2 (EtherNet/IP Scanner)
F800:04	Device Type	M RO	0x000C (12)
F800:05	Vendor ID	M RO	0x006C (108)
F800:06	Product Code	M RO	0xAF12 (44818)
F800:07	Revision	M RO	1.1
F800:08	Serial Number	M RO	0x00000000 (0)
F800:20	MAC Address	M RO	02 1B 21 65 0B 6A
F800:21	IP Address	M RW	192.254.20.100
F800:22	Network Mask	M RW	255.255.255.0
F800:23	Gateway Address	M RW	0.0.0.0
F800:24	DHCP Max Retries	M RW	0
F800:25	TCP/IP TTL	M RW	128
F800:26	TCP/IP UDP Checksum	M RW	TRUE
F800:27	TCP/IP TCP Timeout	M RW	300 Seconds
F800:28	MultiCast TTL	M RW	1
F800:29	MultiCast UDP Checksum	M RW	FALSE
F800:2A	Forward Class3 to PLC	M RW	FALSE
F800:2B	Advanced Options	M RW	0x0100 (256)
[+] F900:0	Master Info	RO	> 43 <

Fig. 119: Object 0xF800, Master Settings

### ADS-Write command

AMSNetId:

In the case of the EL6652(-0010) this is the same as the EtherCAT AmsNetId; in the case of the supplement the AMSNetId is to be taken from the System Manager.

Port:

When using the EL6652(-0010) the port number is the corresponding EtherCAT Slave address of the terminal. In the case of the EtherNet/IP supplement the port number is to be permanently set to 0xFFFF.

IDXGRP: 0x0001F480

IDXOFFS: 0x00000000

After adjusting the settings a TwinCAT restart is to be carried out in the case of the supplement, as well as switching once to *PreOp* mode in the case of the EL6652. The new settings are then accepted on transition to the *OP* mode.

**Setting for setting (4 bytes + object size (256 bytes))**

Byte Offset 0: 0x45  
 Byte Offset 1: 0x23  
 Byte Offset 2: ObjIndex LoByte (e.g. 0x8000 for Slave 1 and 0x8010 for Slave 2 and 0xF800 for the Master)  
 Byte Offset 3: ObjIndex HiByte  
 Byte Offset 4-260: Data of the object (see object description below)

**Setting for resetting (4 bytes)**

Byte Offset 0: 0x00  
 Byte Offset 1: 0x00  
 Byte Offset 2: ObjIndex LoByte (e.g. 0x8000 for Slave 1 and 0x8010 for Slave 2 and 0xF800 for the Master)  
 Byte Offset 3: ObjIndex HiByte

**Object description**

Offset	Name	Data Type	SubIndex	Store Category	
				1	2
0x00..0x01	Id	UINT16	1		
0x02..0x03	Reserved	UINT16	-		
0x04..0x23	Product Name	BYTE[32], STRING(31)	3		X
0x24..0x27	Device Type	UINT32	4		
0x28..0x2B	Verndor ID	UINT32	5		
0x2C..0x2F	Product Code	UINT32	6		X
0x30..0x33	Revision	UINT32	7		
0x34..0x37	Serial Number	UINT32	8		
0x38..0x7D	Reserved	BYTE[70]	-		
0x7E..0x83	MAC Address	BYTE[6]	32		
0x84..0x87	IP Address	UINT32	33	X	
0x88..0x8B	Network Mask	UINT32	34	X	
0x8C..0x8F	Gateway Address	UINT32	35	X	
0x90..0x91	DHCP Max Retries	UINT16	36		
0x92..0x93	TCP/IP TTL	UINT16	37		
0x94..0x95	TCP/IP UDP Checksum	UINT16	38		
0x96..0x97	TCP/IP TCP Time-out	UINT16	39		
0x98..0x99	Multicast TTL	UINT16	40		
0x9A..0x9B	Multicast Checksum	UINT16	41		
0x9C..0x9D	Forwaord Class3 to PLC	UINT16	42		
0x9E..0x9F	Flags	UINT16	43		
0xA0..0xFF	Reserved	Byte[96]	-		

**Store Category**

The "Store Category" defines, which \*.tsm settings should be overwritten with settings from the remanent memory. To do this, bits 9 & 8 must be set accordingly under "Flags" in the System Manager project. In order to modify both, both bits must be set.  
 (Bit9=Cat2, Bit8=Cat1)

**ADS-Read command**

AMSNetId:

In the case of the EL6652(-0010) this is the same as the EtherCAT AmsNetId; in the case of the supplement the AMSNetId is to be taken from the System Manager.

Port:

When using the EL6652(-0010) the port number is the corresponding EtherCAT Slave address of the terminal. In the case of the EtherNet/IP supplement the port number is to be permanently set to 0xFFFF.

Slave:

IDXGRP: 0x44818000

IDXOFFS:ID of the slave interface

LEN: 256

The ID of the slave interface is taken from the setting in the System Manager, the ID is set by the system and can only be read.

Index	Name	Flags	Wert	Einheit
8000:0	Slave Settings (Box 4)	M RO	> 43 <	
8000:01	Slave Number	M RO	0x0004 (4)	
8000:03	Product Name	M RW	Box 1 (TC EtherNet/IP Slave)	
8000:04	Device Type	M RO	0x000C (12)	
8000:05	Vendor ID	M RO	0x006C (108)	
8000:06	Product Code	M RO	0x7FFE (32766)	
8000:07	Revision	M RO	1.1	
8000:08	Serial Number	M RO	0x00000000 (0)	
8000:20	MAC Address	M RO	02 00 04 00 00 00	
8000:21	IP Address	M RW	0.0.0.0	
8000:22	Network Mask	M RW	0.0.0.0	
8000:23	Gateway Address	M RW	0.0.0.0	
8000:24	DHCP Max Retries	M RW	0	
8000:25	TCP/IP TTL	M RW	128	
8000:26	TCP/IP UDP Checksum	M RW	TRUE	
8000:27	TCP/IP TCP Timeout	M RW	300 Seconds	
8000:28	MultiCast TTL	M RW	1	

Fig. 120: Object 0x8000: slave settings

Master:


IDXGRP: 0x4481F800

IDXOFFS:0x00000000

LEN: 256

The data are saved in the data array as described above.

Example for TwinCAT 2.11 R3 and EL6652 or EL6652-0010, reading and writing the IP setting:

 <https://infosys.beckhoff.com/content/1033/el6652/Resources/zip/1405400587.zip>

## 5.5 EL6652-0000 Master (Scanner)

### 5.5.1 EL 6652-0000 configuration

The EL6652 EtherCAT Terminal is a simple Ethernet/IP Master and supports the exchange of process data without configuration data. Only Ethernet/IP Slaves that require no configuration can be used. The feature corresponds to the "Generic Node" in the case of Rockwell. Before using the EL6652, you must check whether the slave can enter into process data exchange mode without configuration.

The import of an EDS file is currently not supported. The EDS file [► 104] chapter is intended to show how the data from the EDS file are to be interpreted and how the corresponding values for the configuration of the slave are to be manually extracted from the EDS file.

#### Commissioning: Insertion of the EL6652-0000

Insert the EL6652 into your EtherCAT system. You can perform the online scan as usual or manually add the terminal.



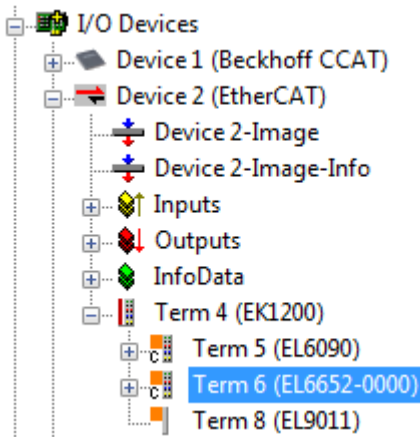


Fig. 121: Inserting the EL6652-0000 in TwinCAT 2.1x

As soon as you have appended the terminal you must also create the "EtherNet/IP" device in the System Manager. Go onto "I/O devices" and append a further device.

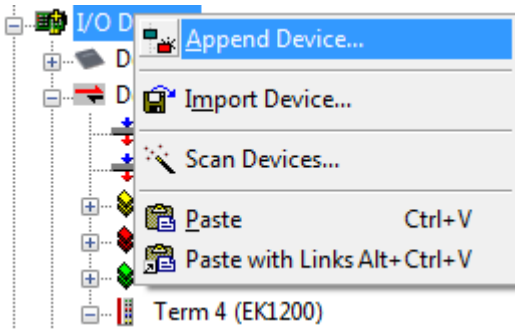


Fig. 122: Insert a device "EtherNet/IP" in "I/O devices"

Under "EtherNet/IP" you will find the adapter "EtherNet/IP adapter (EL6652)". Select this.

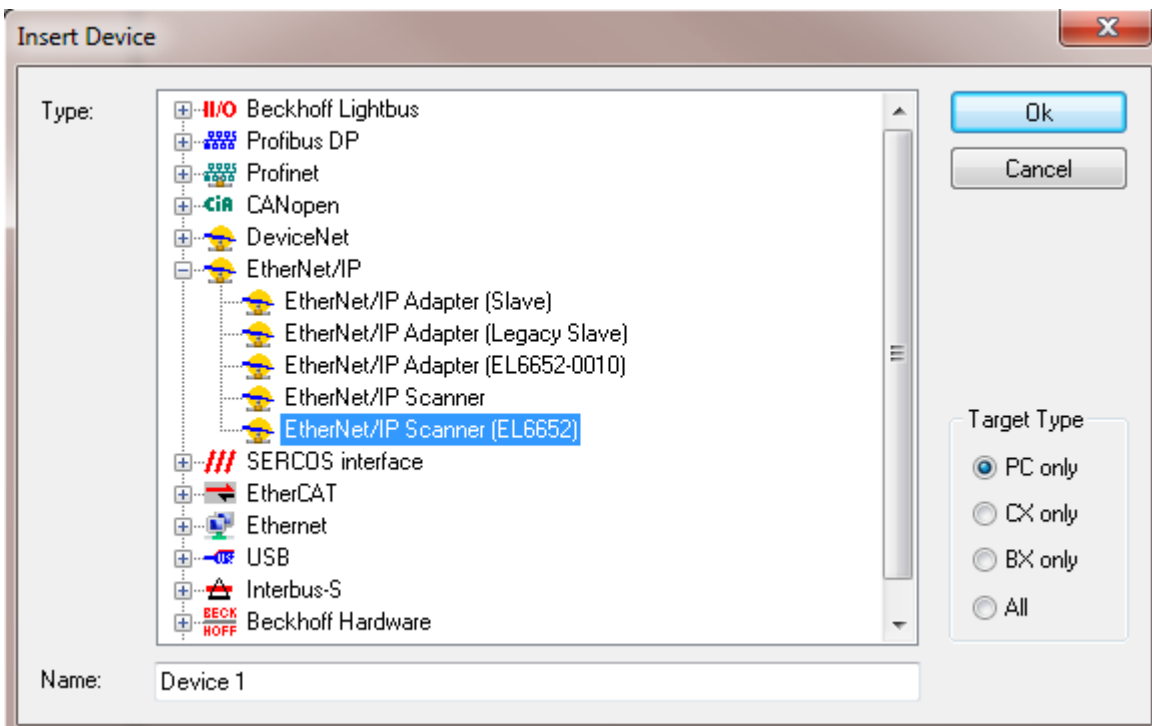


Fig. 123: Selection of adapter "EtherNet/IP Scanner (EL6652)"

If you have only one EL6652 in your system, the System Manager links it automatically. You should nevertheless check this (see fig. "Searching for the EL6652 terminal") by going onto the "EtherNet/IP Device" (1), "Adapter" (2) and then finding the terminal in the "Device Name". If that is not the case, go onto "Search" (3) and then select the correct terminal (4).

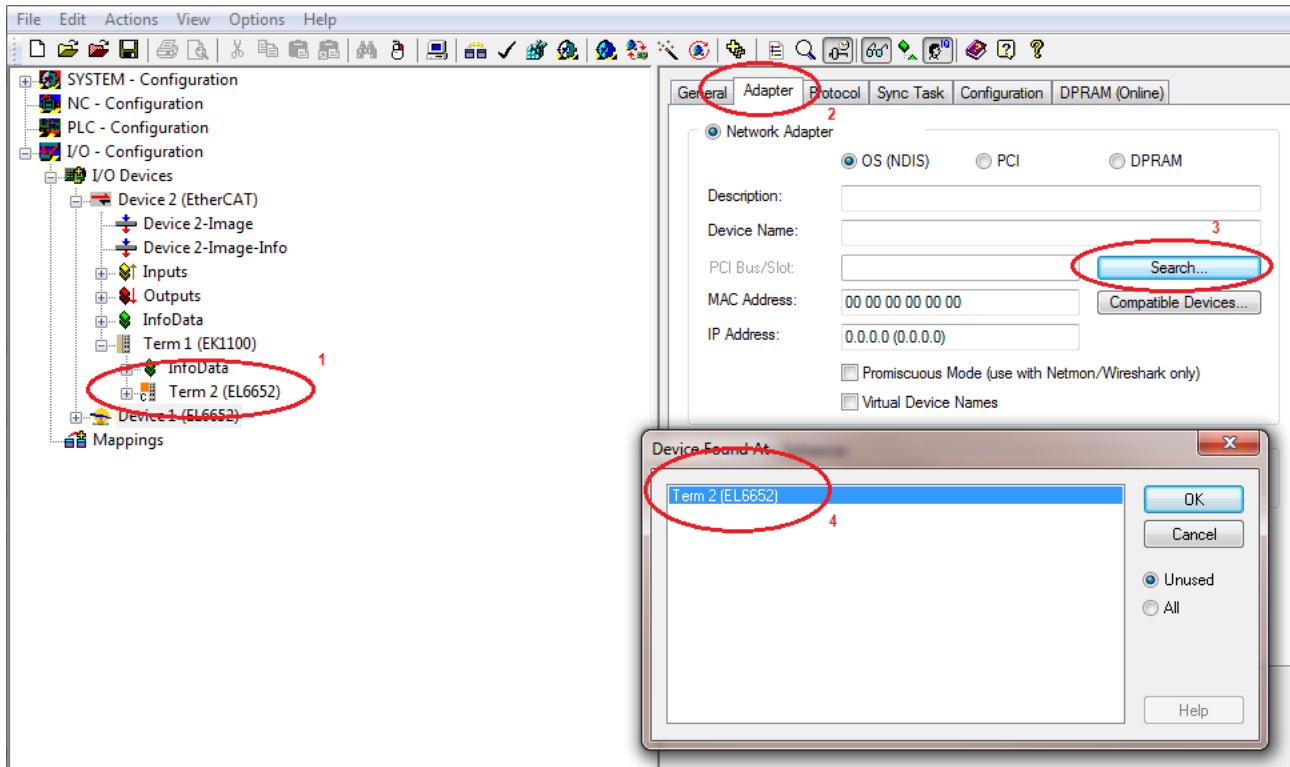


Fig. 124: Searching for the EL6652 terminal

The MAC address and IP address are not updated in this dialog box. These can be found on the EtherNet/IP box under "Configuration".

**Configuration**

Now create a configuration; the master must be configured with an IP address, SubNetMask and optionally with a gateway.

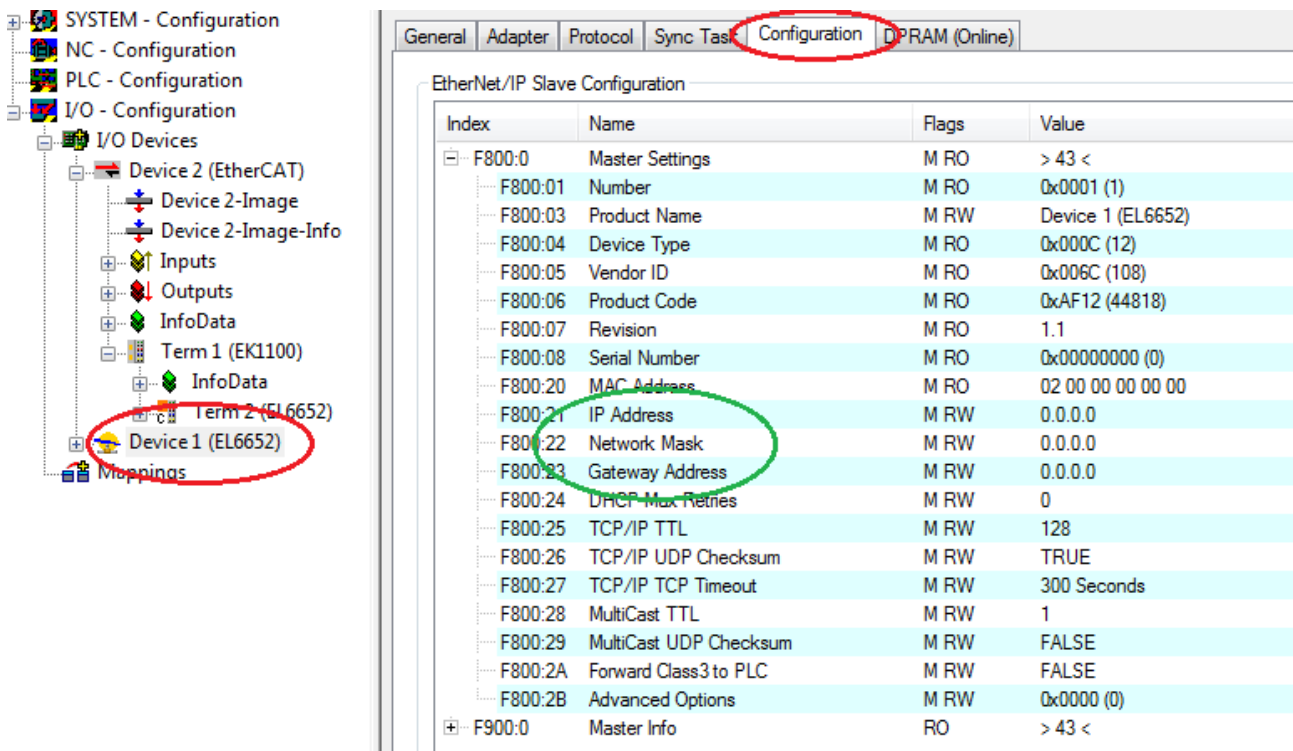


Fig. 125: Configuration of the IP address, network mask and gateway address

Now click on "Device1 (EL6652)" with the right mouse button and add a "Generic Ethernet/IP adapter". Alternatively you can search for Ethernet/IP devices (Scan Boxes...).

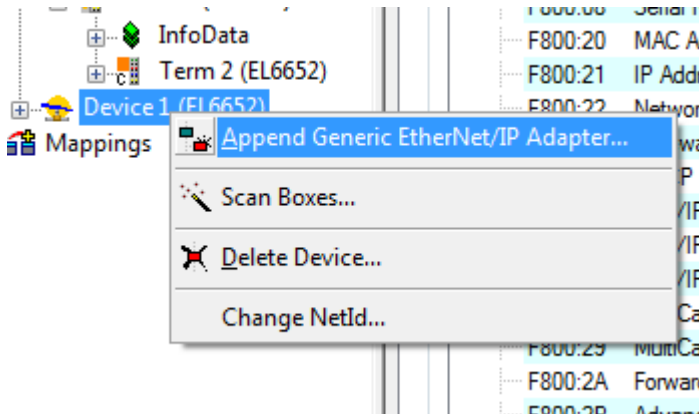


Fig. 126: Add "Generic EtherNet/IP Adapter"

You will now be requested to enter the IP address. After that you can append a "Connection" (the "Message Object" is not currently supported).

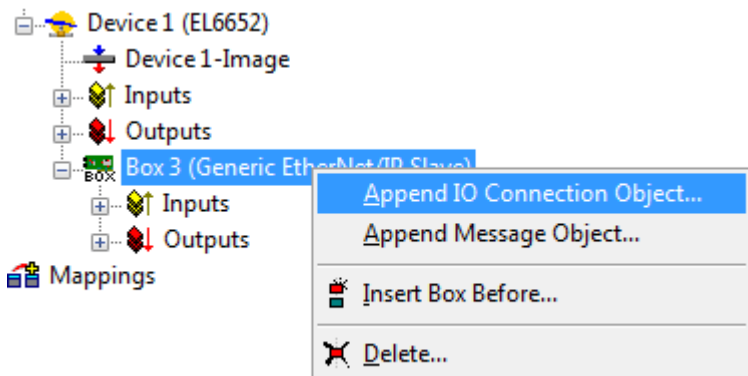


Fig. 127: Append "Connection Object"

The following dialog box appears (see the chapter EDS file [▶ 104] for the settings for this).

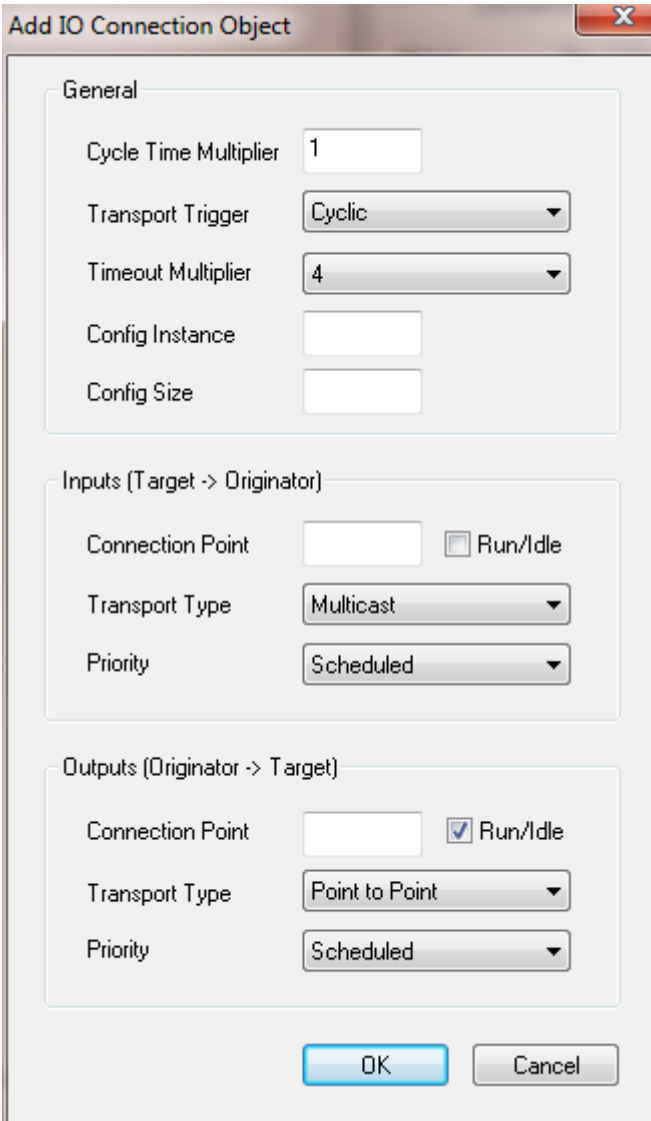


Fig. 128: Dialog box "Add I/O Connection Object"

Now append the variables that you want to use for the cyclic data exchange (these do not have to correspond to the EDS file in terms of type; only the overall length of the data is important here).

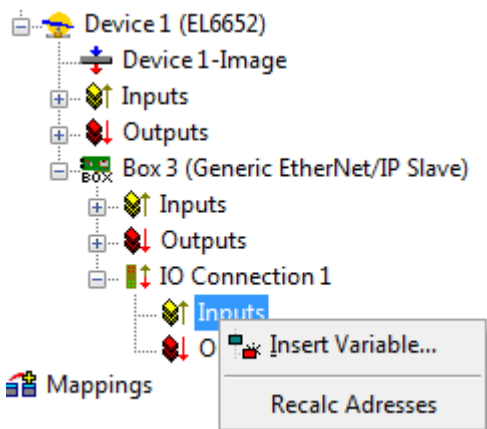


Fig. 129: Appending variables

**Task time**

The time with which the EtherNet/IP is to operate is specified with the SyncTask. Two methods are available here:

**SyncTask via Mapping** – in this case the task is used with which the variables are linked. That is usually the PLC task. However, if the PLC is to be stopped, for example by breakpoints, the task is no longer processed. The consequence of this is that the EtherNet/IP is also no longer triggered and the EtherNet/IP master goes into the error state.

**SyncTask via Special Sync Task** - in this case an additional task is used that is automatically started with TwinCAT. This runs in its own cycle and is therefore also independent of other tasks that are linked with the variables.

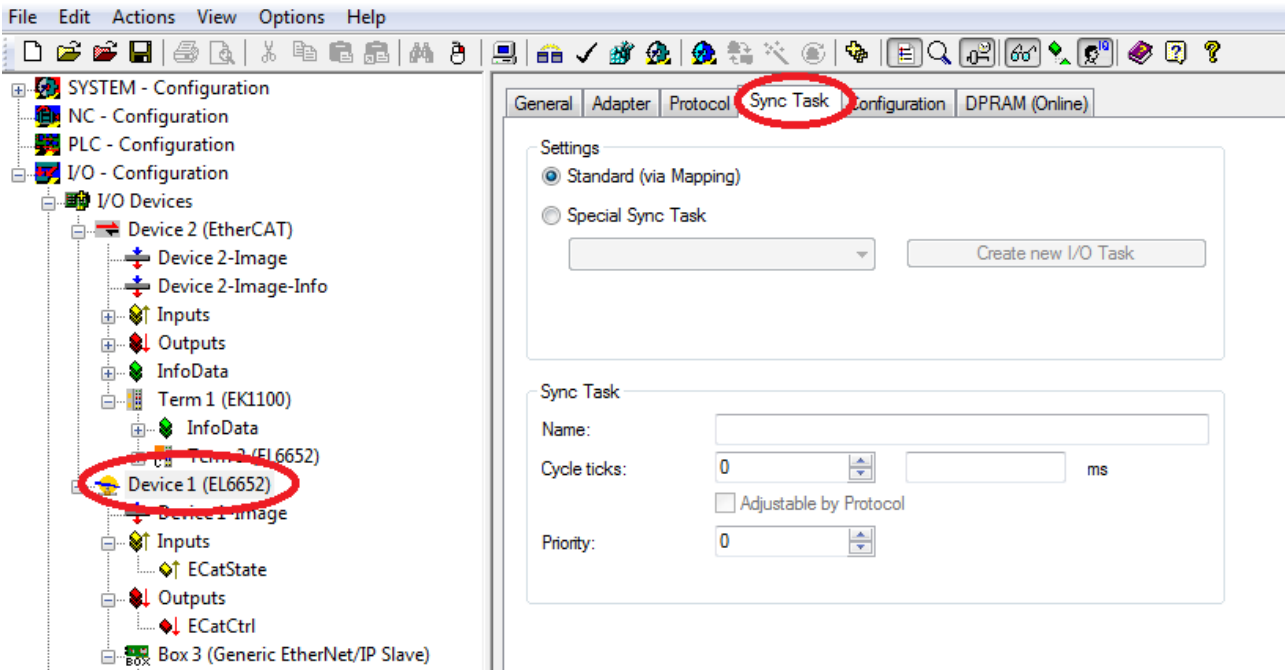


Fig. 130: Selection of the task time

**Diagnostics**

Two steps are necessary for the diagnosis. Firstly the monitoring of the EtherCAT status with "ECatState" (red), which must be equal to "0". Secondly the monitoring of the Ethernet/IP status in the State (green). State = 0x8000 is a configuration error; in this case the I/O connection must be checked. If State = 0x0001 an error has occurred in the Ethernet/IP data exchange. If the value has already been "0", this is usually a watchdog error. This means that the data did not arrive at the Ethernet/IP master within the set watchdog time. The Ethernet/IP cycle time should be increased if this error occurs frequently. If the State is already 0x0001 at the start of communication you should check and read out the "DiagHistory" (tab on the EL6652. See chapter Diagnostic History [▶ 128]).

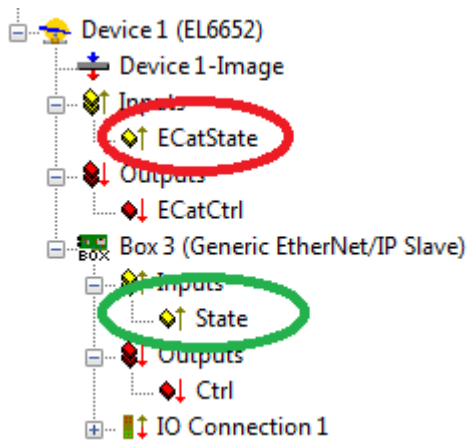


Fig. 131: *Diagnosis of "ECatState" and "State"*

## 5.5.2 EL6652-0000 - Configuration parameters

### Index 0xF800 Scanner Settings

Index	Name	Access	Meaning
F800:0	Master settings		
F800:1	Number	RO	Box Id
F800:3	Product Name	RW	Name of the device
F800:4	Device Type	RO	Device type
F800:5	Vendor ID	RO	Vendor number
F800:6	Product Code	RO	Product code
F800:7	Revision	RO	Version
F800:8	Serial Number	RO	Serial number (see object 0xF900)
F800:20	MAC address	RO	MAC address (see object 0xF900)
F800:21	IP address	RW	= 0 means DHCP enabled, any other value results in fixed IP Address
F800:22	Network Mask	RW	= 0 means DHCP enabled, any other value results in fixed SubNetMask
F800:23	Gateway Adress	RW	= 0 means DHCP enabled, any other value results in fixed Gateway Address
F800:24	DHCP Max Retries	RW	0-infinite (number of dhcp retries (actually only infinite retries implemented))
F800:25	TCP/IP TTL	RW	Time to live for Unicast TCP/UDP Communication
F800:26	TCP/IP UDP Checksum	RW	0-UDP Checksum disabled, 1-UDP Checksum enabled
F800:27	TCP/IP TCP Timeout	RW	Tcp Idle Timout in seconds (0-Timout disabled)
F800:28	MultiCast TTL	RW	Time to live for Multicast UDP Communication
F800:29	MultiCast UDP Checksum	RW	0-UDP Checksum disabled, 1-UDP Checksum enabled
F800:2A	Forward Class3 to PLC	RW	Forward Explicit Messaging to PLC (actually not implemented)
F800:2B	Advanced Slave Options	RW	Store Category (Bit9=Cat2, Bit8=Cat1) see <a href="#">Writing the IP address from the PLC [►_94]</a>


### Index 0xF900 Scanner Info

The current valid settings are displayed here; these can differ from the object 0xF800. The object 0xF900 displays the active parameters to you.

### 5.5.3 EL6652-0000 EDS file

The following chapter is intended to explain to you, by way of an example, how an EDS file is interpreted and how you can use this information for the configuration of the slave in the System Manager.

The example uses an EDS file of the type Endress+Hauser: Promass 100 EDS

 <https://infosys.beckhoff.com/content/1033/el6652/Resources/zip/1405402763.zip>

#### Selection of a connection

In the EDS file there are usually several connection options; in the yellow line you can see that the slave next to 'connection 1' requires 398 bytes of configuration data. Since the EL6652 does not support this, this connection cannot be used.

```
[Connection Manager]
Connection1 =
Param10000,,Assem102, $ O->T RPI, size, format
Param10000,,Assem100, $ T->O RPI, size, format
.. $ proxy config size, format
398,Assem104, $ target config size, format
"Fix Input/Output + Config Assembly", $ Connection Name
"Fix Input/Output + Config Assembly", $ help string
"20 04 24 68 2C 66 2C 64"; $ Path
```

There are further connection options in the EDS, in this case Connection 7. The configuration data are set to "0" in this connection. This can be used for communication with the EL6652.

```
Connection7 =
Param10000,,Assem102, $ O->T RPI, size, format
Param10000,,Assem101, $ T->O RPI, size, format
.. $ proxy config size, format
0,, $ target config size, format
"Configurable Input + Fix Output", $ Connection Name
"Configurable Input + Fix Output", $ help string
"20 04 24 69 2C 66 2C 65"; $ Path
```

#### Config, input and output instance

The config, input and output instances must now be read.

```
Connection7 =
Param10000,,Assem102, $ O->T RPI, size, format
Param10000,,Assem101, $ T->O RPI, size, format
.. $ proxy config size, format
0,, $ target config size, format
"Configurable Input + Fix Output", $ Connection Name
"Configurable Input + Fix Output", $ help string
"20 04 24 69 2C 66 2C 65"; $ Path
```

Blue corresponds to the config instance, red to the output instance and yellow to the input instance; these are usually represented in the EDS file as hex values and must be entered in the System Manager as decimal numbers.

```
69hex -> 105dec Config
66hex -> 102dec Output
65hex -> 101dec Input
```



**Add IO Connection Object**

**General**

Cycle Time Multiplier: 1

Transport Trigger: Cyclic

Timeout Multiplier: 4

Config Instance: 105

Config Size: 0

**Inputs (Target -> Originator)**

Connection Point: 101  Run/Idle

Transport Type: Multicast

Priority: Scheduled

**Outputs (Originator -> Target)**

Connection Point: 102  Run/Idle

Transport Type: Point to Point

Priority: Scheduled

OK Cancel

Fig. 132: Enter the values in the "Add IO Connection Object" dialog box

Register the values according to the EDS file in the System Manager.

### Cycle Time Multiplier

The EL6652 always operates internally with a cycle time of 1 millisecond. Some Ethernet/IP slaves are not designed for this time. The "Cycle Time Multiplier" can increase the time. Check the EDS file to see which minimum value your slave can handle. If the slave is not able to process the cycle time of 1 ms, the time must be increased by a factor via the multiplier.

```
Connection7 =
Param10000,Assem102, $ O->T RPI, size, format
Param10000,Assem101, $ T->O RPI, size, format
,, $ proxy config size, format
0,, $ target config size, format
"Configurable Input + Fix Output", $ Connection Name
"Configurable Input + Fix Output", $ help string
"20 04 24 69 2C 66 2C 65"; $ Path
```

Theoretically, different RPI times can be used for the transmitting and receiving directions; however, the EL6652 always uses the same time for both directions. In this EDS file the parameters for the RPI are located in Param10000.

```

Param10000 =
0,      $ reserved, shall equal 0
,,      $ Link Path Size, Link Path
0x0000, $ Descriptor
0xC8,   $ Data Type (UDINT / UINT32)
4,      $ Data Size in bytes
"RPI Range", $ Name
",      $ units
",      $ help string
5000,10000000,20000, $ min, max, default data values
,,,,   $ mult, div, base, offset scaling
,,,,   $ mult, div, base, offset links
;      $ decimal places

```

This Ethernet/IP slave can handle a minimum of 5 ms (5000  $\mu$ s); the manufacturer specifies a default value of 20 ms. It is advisable to follow the manufacturer's specification and to use the 20 ms. This means that a "Cycle Time Multiplier" of 20 is to be entered (1 ms \* 20 = 20 ms).

### Specifying the data length

As the final step the data length must be entered; here again, the values are taken from the EDS file.

```

Connection7 =
Param10000,,Assem102, $ O->T RPI, size, format
Param10000,,Assem101, $ T->O RPI, size, format
,, $ proxy config size, format
0,, $ target config size, format
"Configurable Input + Fix Output", $ Connection Name
"Configurable Input + Fix Output", $ help string
"20 04 24 69 2C 66 2C 65"; $ Path

```

Assem102 corresponds to the output data, in the EDS file with 64 bytes; the green data correspond to the bits, in total 64 bytes again. These values can be taken in the System Manager as a basis in order to use the data just as they are represented in the EDS file. A byte array of 64 bytes can also be used, whose data are broken down in the PLC.

```

Assem102 =
"Output Assembly Fix", $ Assembly Name
"20 04 24 66 30 03", $ Path
64, $ Size in bytes
0x0000, $ Descriptor, default = 0
,, $ reserved fields
1.Param20600, $ - / Totalizer Control 1 Activation
1.Param20601, $ - / Totalizer Control 2 Activation
1.Param20602, $ - / Totalizer Control 3 Activation
1.Param20603, $ - / Pressure Compensation Activation
1.Param20604, $ - / Reference Density Compensation Activation
1.Param20605, $ - / Temperature Compensation Activation
1.Param20606, $ - / Verification Activation
1,
8,
8,
8,
16.Param93, $ TIC_Control_1 / Control Totalizer 1
16,
16.Param94, $ TIC_Control_2 / Control Totalizer 2
16,
16.Param95, $ TIC_Control_3 / Control Totalizer 3
16,
32 Param121, $ C4W_InputPressure_1 / External pressure
16.Param85, $ SU_Pressure_1 / Pressure unit
16,
32 Param122, $ C4W_InputRefDensity_1 / External reference density
16.Param86, $ SU_RefDensity_1 / Reference density unit
16,
32 Param123, $ C4W_InputTemperature_1 / External temperature
16.Param87, $ SU_Temperature_1 / Temperature unit
16,
16.Param40, $ HBT_PerformVerification_1 / Start verification
16,
32,
32,
32,
32,
32,
32,

```

The output data in the System Manager are illustrated below; first of all 8 bits are inserted, then 3 x byte, 6 x INT, 1 x DWord, 2 x INT, etc., until the length corresponds to 64 bytes.

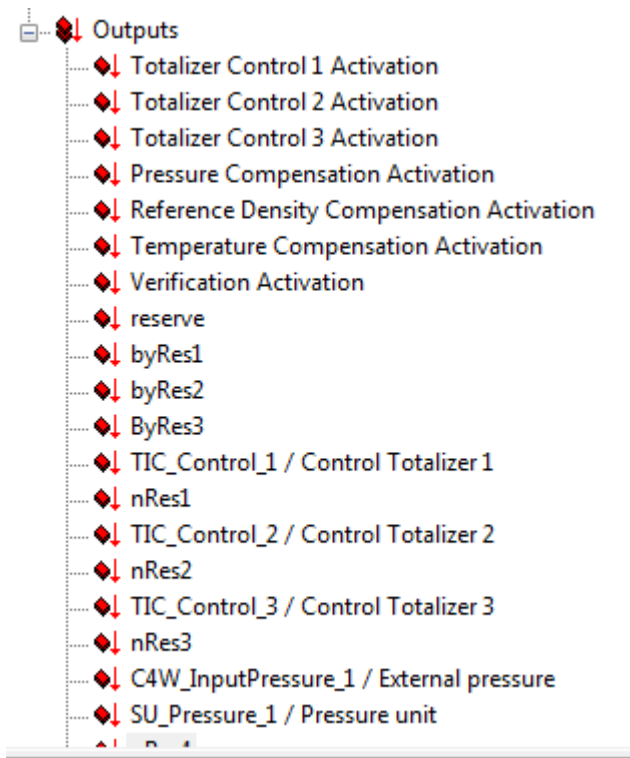


Fig. 133: *Output data in the System Manager*

The inputs Assem101 are handled analogously:

```

Assem101 =
"Input Assembly Configurable", $ Assembly Name
"20 04 24 65 30 03", $ Path
88, $ Size in bytes
0x0000, $ Descriptor, default = 0
,, $ reserved fields
32,, $ Header
32,, $ Device status (16) / Diagnostic (8) / PadByte (8)
32 Param20000
32 Param20001
32 Param20002
32 Param20003
32 Param20004
32 Param20005
32 Param20006
32 Param20007
32 Param20008
32 Param20009
32 Param20010
32 Param20011
32 Param20012
32 Param20013
32 Param20014
32 Param20015
32 Param20016
32 Param20017
32 Param20018
32 Param20019
    
```

On completion the configuration must look like the following:

Fig. 134: Configuration "IO Connection Object"

**Summary:**

Green: cycle time of the slave

Blue: config parameter (size always 0) config instance 105

Yellow: input instance Len 88 connection point 101

Red: output instance Len 64 connection point 102

## 5.6 EL6652-0010 Slave

### 5.6.1 EL 6652-0010 configuration

The most important settings for establishing a connection with an EtherNet/IP master or a scanner are IP address, the assembly instance number and thus the length of the data and the correct task cycle time.

The IP address can be freely assigned and is transferred to the terminal during the EtherCAT start-up. The assembly instance numbers are permanently assigned and must be correctly set in the master. This also always includes the number of data or the size of the process image. The maximum length of the data can differ from master to master. Enquire to the manufacturer of the master for this information. The task time with which the EtherNet/IP is to be operated must be at least just as large on the EtherCAT terminal as on the master side. However, it can also be many times smaller. For example, you can set 10 ms on the master side. The task of the EtherNet/IP slave may then be 10, 5, 2, and 1 ms for example.

In terms of communication, "Unicast" or "Multicast" can be used on the master side – the EL6652-0010 supports both modes of operation. In the case of Multicast, pay attention to the high network load that this causes, especially in systems with many or short cycle times. A high network load may possibly impair communication.

No configuration data are necessary for the EL6652-0010; the configuration must usually be specified, but the length of configuration data is to be set to "zero".

#### Commissioning: Insertion of the EL6652-0010

Insert the EL6652-0010 into your EtherCAT system. You can perform the online scan as usual or manually add the terminal.

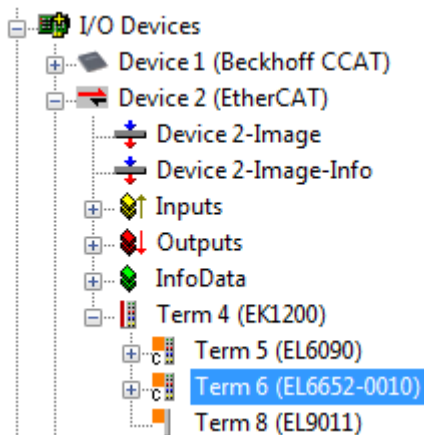


Fig. 135: Inserting the EL6652-0010 in TwinCAT 2.1x

As soon as you have appended the terminal you must also create the "EtherNet/IP" device in the System Manager. Go onto "I/O devices" and append a further device.

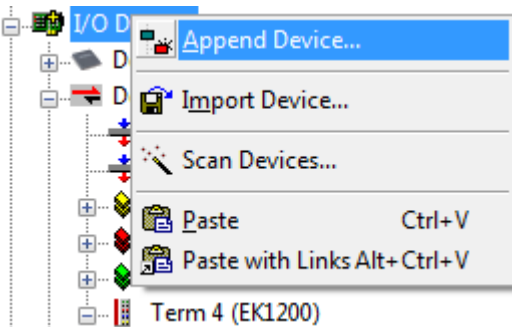


Fig. 136: Insert a device "EtherNet/IP" in "I/O devices"

Under "EtherNet/IP" you will find the adapter "EtherNet/IP adapter (EL6652-0010)". Select this.

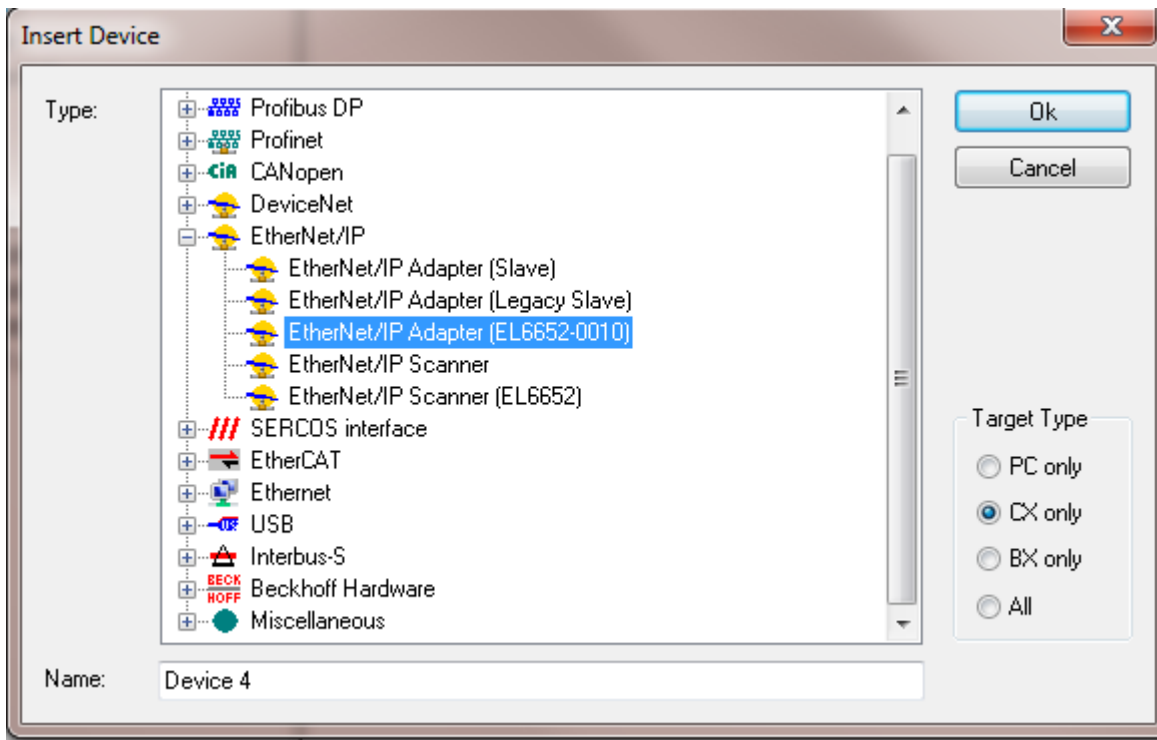


Fig. 137: Selection of adapter "EtherNet/IP adapter (EL6652-0010)"

If you have only one EL6652-0010 in your system, the System Manager links it automatically.

You should nevertheless check this (see fig. "Searching for the EL6652-0010 terminal") by going onto the "EtherNet/IP Device" (1), "Adapter" (2) and then finding the terminal in the "Device Name". If that is not the case, go onto "Search" (3) and then select the correct terminal (4).

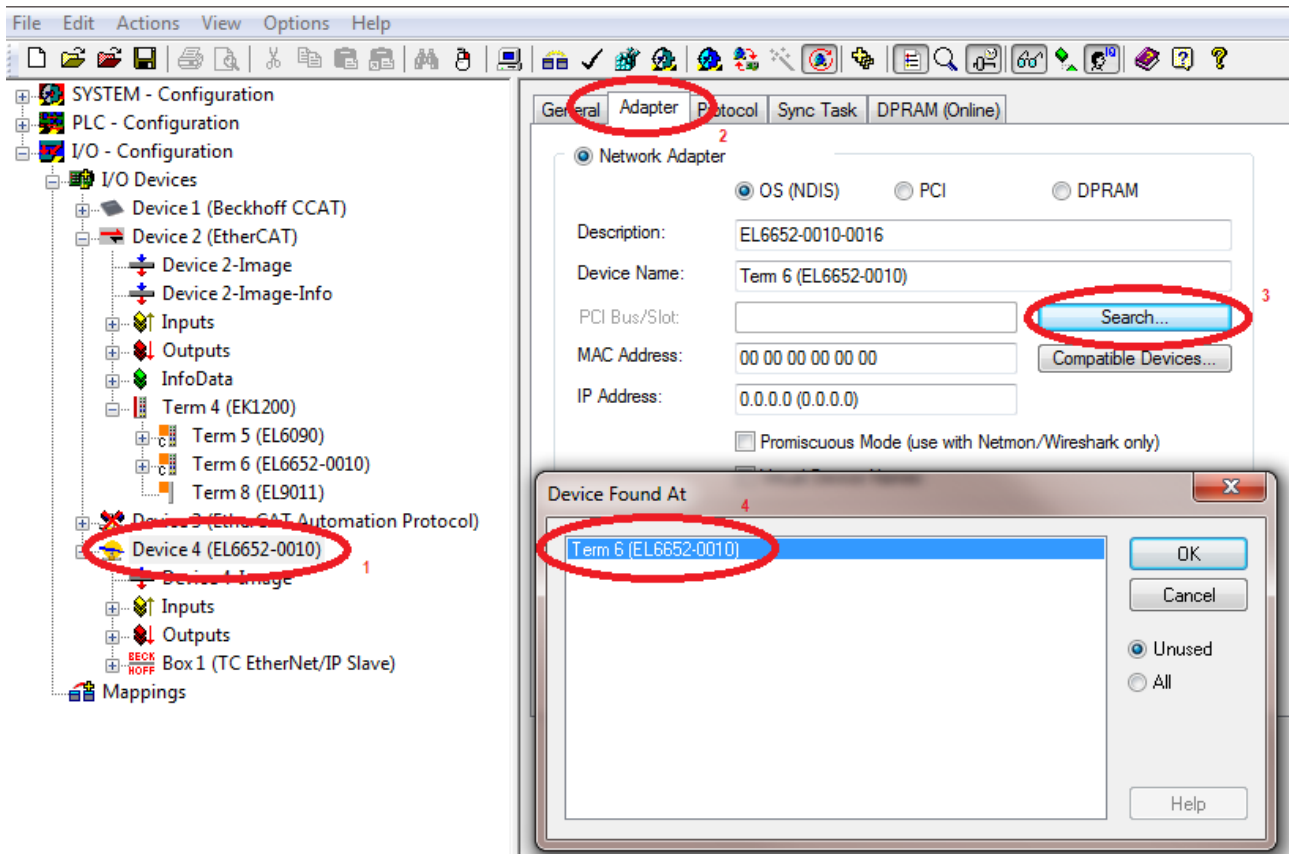


Fig. 138: Searching for the EL6652-0010 terminal

The MAC address and IP address are not updated in this dialog box. These can be found on the EtherNet/IP box under "Configuration".

**Configuration**

Now create a configuration; the master must be configured with an IP address, SubNetMask and the data which are to be exchanged with the EtherNet/IP Master.

Select the box and enter the IP address (0x8000:21) and SubNetMask (0x8000:22) (green marking), that you wish to use on the slave.

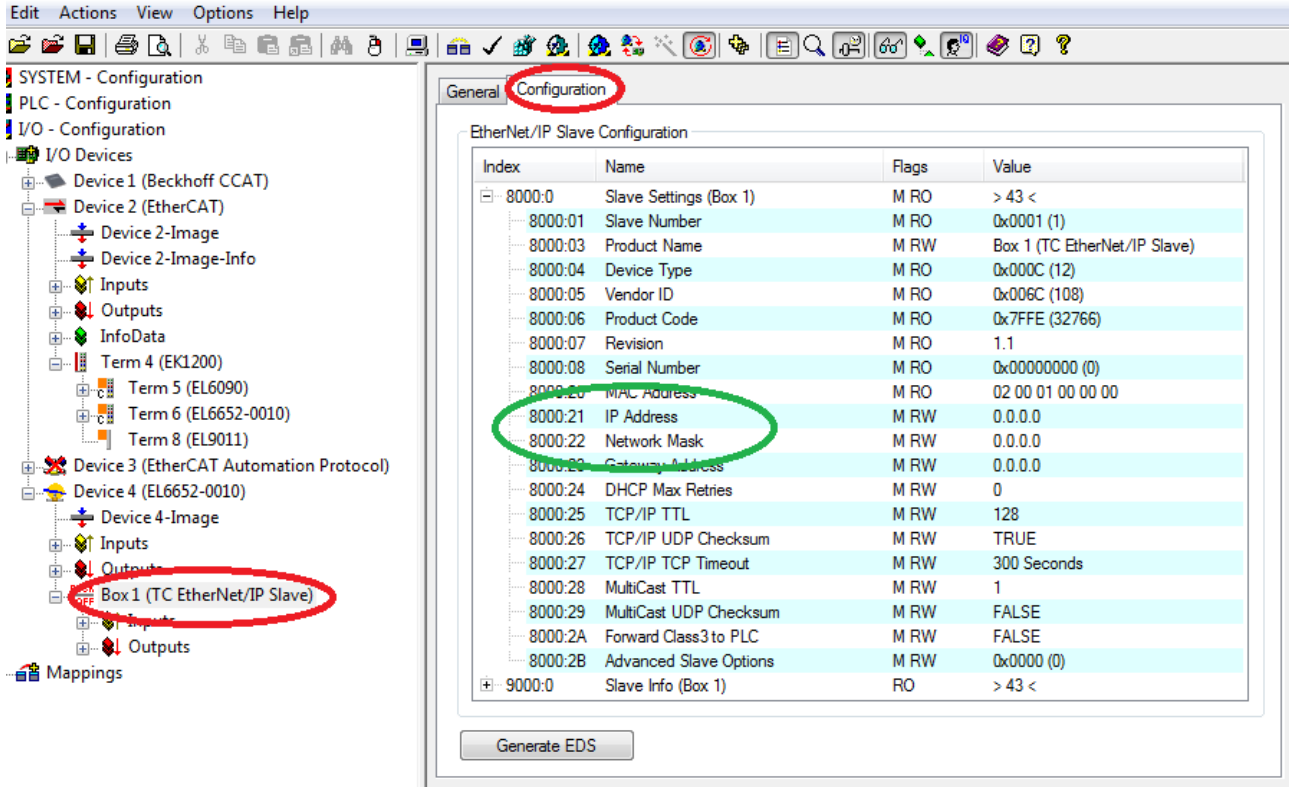


Fig. 139: Configuration of the IP address, network mask

Then append the IO connection.

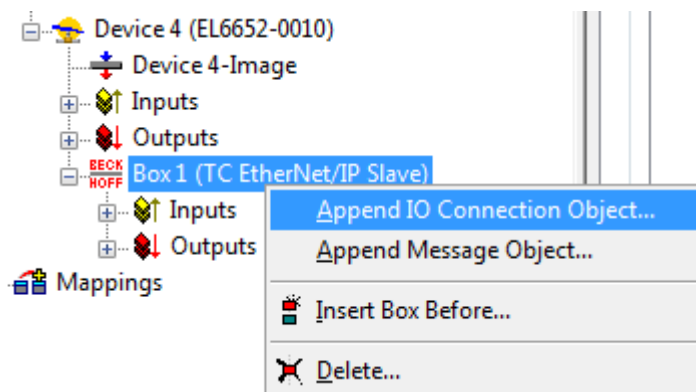


Fig. 140: Append "Connection Object"

The IO assembly already has inputs and outputs; here you insert any desired types of data and their quantity. These are then linked with the PLC or the task. In the next step, insert a data word (2 bytes) under the inputs (right-click on Inputs) and also append a data word (2 bytes) for the outputs (right-click on Outputs).



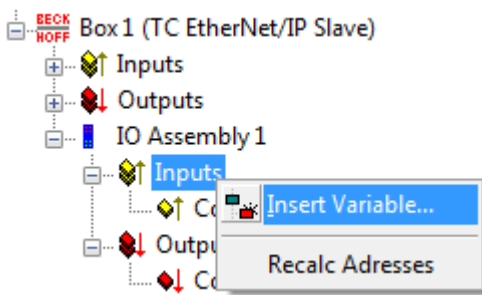


Fig. 141: Append variables

An IO assembly (0x8001) is now present in the configuration. The instance numbers must be observed for the entry in the master.

- Value 128 for the configuration; as already mentioned, this is always to be used with "zero" bytes.
- Value 129 for the input data; the length is 6 bytes (4 bytes which are automatically created in the system and 2 bytes for the word created by the user)
- Value 130 for the output data; the length is 6 bytes (4 bytes which are automatically created in the system and 2 bytes for the word created by the user)

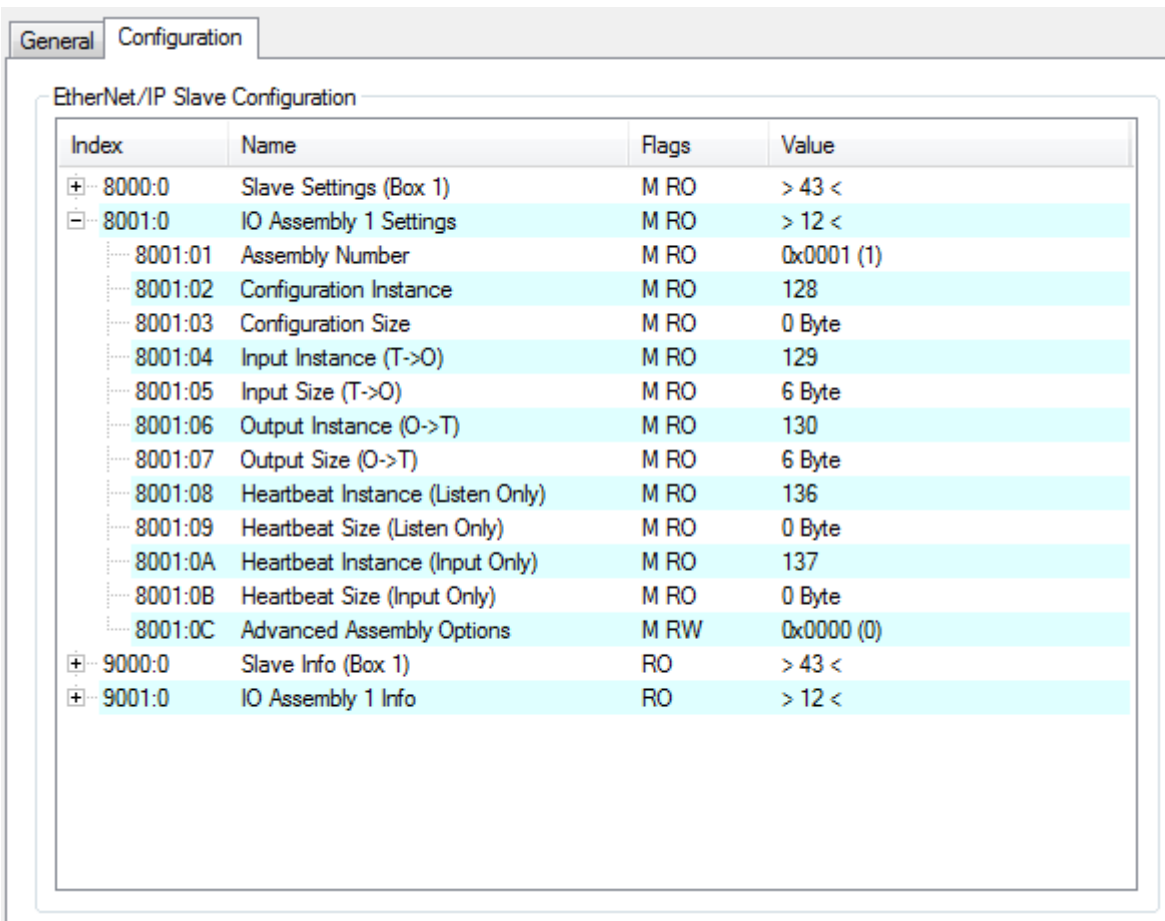


Fig. 142: Configuration of the slave settings

**Task time**

The time with which the EtherNet/IP is to operate is specified with the SyncTask. Two methods are available here:

**SyncTask via Mapping** – in this case the task is used with which the variables are linked. That is usually the PLC task. However, if the PLC is to be stopped, for example by breakpoints, the task is no longer processed. The consequence of this is that the EtherNet/IP is also no longer triggered and the EtherNet/IP master goes into the error state.

**SyncTask via Special Sync Task** - in this case an additional task is used that is automatically started with TwinCAT. This runs in its own cycle and is therefore also independent of other tasks that are linked with the variables.

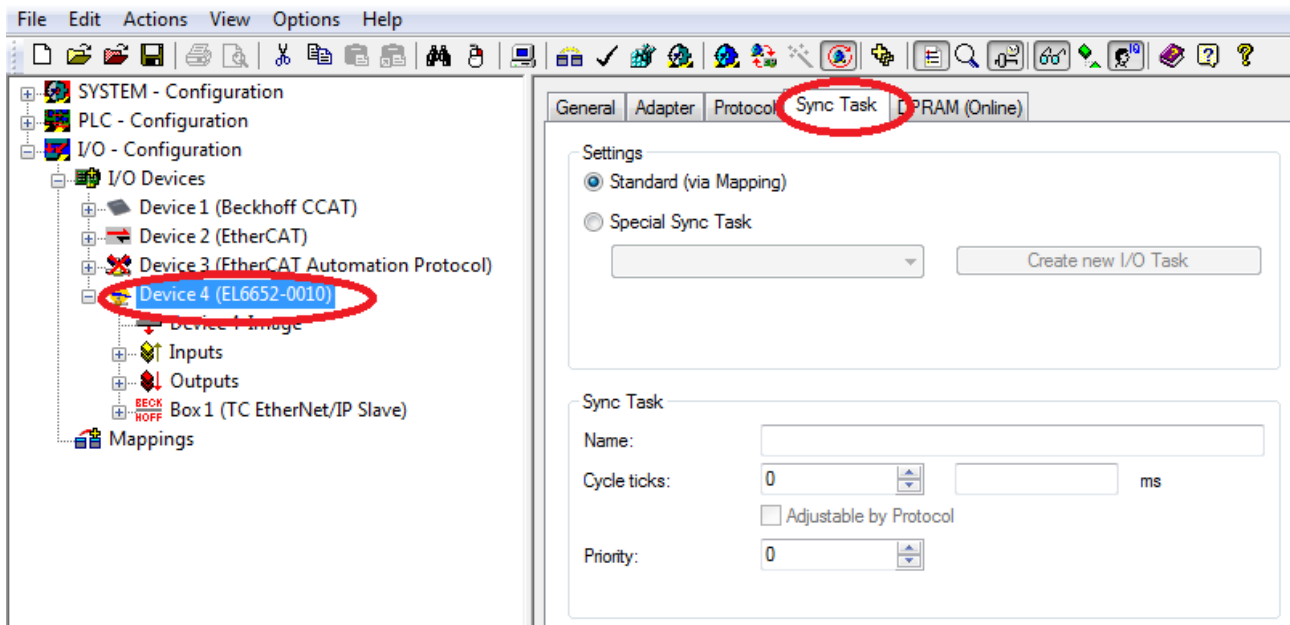


Fig. 143: Selection of the task time

### Virtual slave

As a special feature, the EL6652-0010 offers the possibility to create a 2nd slave with its own MAC address and IP address, which is called a virtual slave. Note here that the virtual interface must share the maximum process data with the real slave. The advantage with this option is that you can simply connect two EtherNet/IP controllers together or that you can get around limitations in the bus communication with the slave without using additional terminals.

A second box is created in the System Manager and the procedure is exactly the same as when configuring a real slave.

In this case the status LEDs are interpreted as follows: if one of the two slaves (real or virtual) has an error, this is displayed even if the other slave is error-free. If both slaves have an error the error of the real slave is always displayed.

## 5.6.2 EL6652-0010 - Configuration parameters

### Index 0x8000 Slave Settings

Index	Name	Meaning
8000:0	slave settings	
8000:1	Slave Number	Slave Box ID
8000:3	Product Name	Name of the device
8000:4	Device Type	Device type
8000:5	Vendor ID	Vendor number
8000:6	Product Code	Product code
8000:7	Revision	Version
8000:8	Serial Number	Serial number (see object 0x9000)
8000:20	MAC address	MAC address (see object 0x9000)
8000:21	IP address	= 0 means DHCP enabled, any other value results in fixed IP Address
8000:22	Network Mask	= 0 means DHCP enabled, any other value results in fixed SubNetMask
8000:23	Gateway Adress	= 0 means DHCP enabled, any other value results in fixed Gateway Address
8000:24	DHCP Max Retries	0-infinite (number of dhcp retries (actually only infinite retries implemented)
8000:25	TCP/IP TTL	Time to live for Unicast TCP/UDP Communication
8000:26	TCP/IP UDP Checksum	0-UDP Checksum disabled, 1-UDP Checksum enabled
8000:27	TCP/IP TCP Timeout	Tcp Idle Timeout in seconds (0-Timeout disabled)
8000:28	MultiCast TTL	Time to live for Multicast UDP Communication
8000:29	MultiCast UDP Checksum	0-UDP Checksum disabled, 1-UDP Checksum enabled
8000:2A	Forward Class3 to PLC	Forward Explicit Messaging to PLC (actually not implemented)
8000:2B	Advanced Slave Options	Store Category (Bit9=Cat2, Bit8=Cat1) see <a href="#">Writing the IP address from the PLC</a> [► 94]

**Index 0x8001 IO Assembly Settings**

Index	Name	Meaning
8001:0	IO Assembly Settings	
8001:1	Assembly Number	Assembly Id
8001:1	Configuration Instance	Config Instance
8001:3	Configuration Size	Config Size (always 0)
8001:4	Input Instance (T->O)	Connection Point for Input Data (Target->Originator)
8001:5	Input Size (T->O)	Size of Input Data (in Bytes)
8001:6	Output Instance (O->T)	Connection Point for Output Data (Originator->Target)
8001:7	Output Size (O-T)	Size of Output Data (in Bytes)
8001:8	Heartbeat Instance (Listen Only)*	Heartbeat Connection Point for Listen Only Connections
8001:9	Heartbeat Size (Listen Only)*	Always 0
8001:A	Heartbeat Instance (Input Only)**	Heartbeat Connection Point for Input Only Connections
8001:B	Heartbeat Size (Input Only)**	Always 0
8001:C	Advanced Assembly Options	Bit 14: 0x4000 hex 0 = default 1 = disable Mapping of "ConnCtrl" and „ConnState“ to EtherNet/IP IO Connection  All other bits always 0 (reserve)

\* Heartbeat Instance (Listen Only): allows monitoring of the input data (the output data in the case of the EL6652-0010) if a connection exists. The "listen only" connection is also terminated when the normal connection is terminated.

\*\* Heartbeat Instance (Input Only): allows monitoring of the input data (the output data in the case of the EL6652-0010). This connection is independent of the actual communication.

The heartbeat is necessary for the monitoring of both connection types (Listen Only and Input Only).

**Index 0x9000 Slave Info**

The current valid settings are displayed here; these can differ from the object 0x8000. The object 0x9000 displays the active parameters.

**Index 0x9001 IO Assembly Info**

The current valid assembly settings are displayed here; these can differ from the object 0x8001. The object 0x9001 displays the active parameters.

**5.6.3 Master (Scanner) configuration****5.6.3.1 EL6652-0010 - Configuration on an RSLogix5000****Example of an EtherNet/IP Master/Scanner configuration**

Open the RSLogix and create a new project. In this example a CompactLogix (L32E) is used.



Fig. 144: *Creating a new controller*

Append a new module.

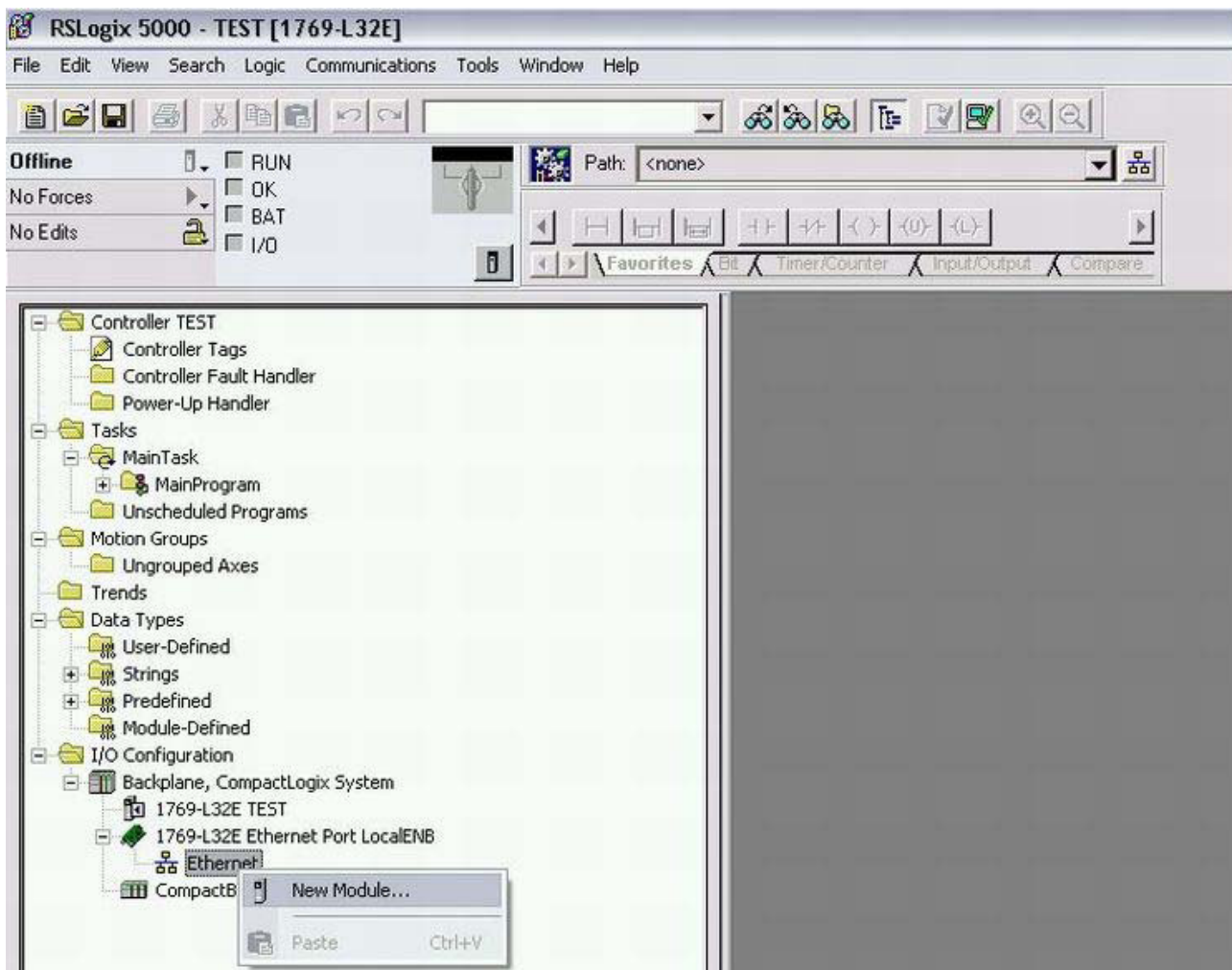


Fig. 145: Appending a new module

Select ETHERNET MODULE "Generic Ethernet Module".

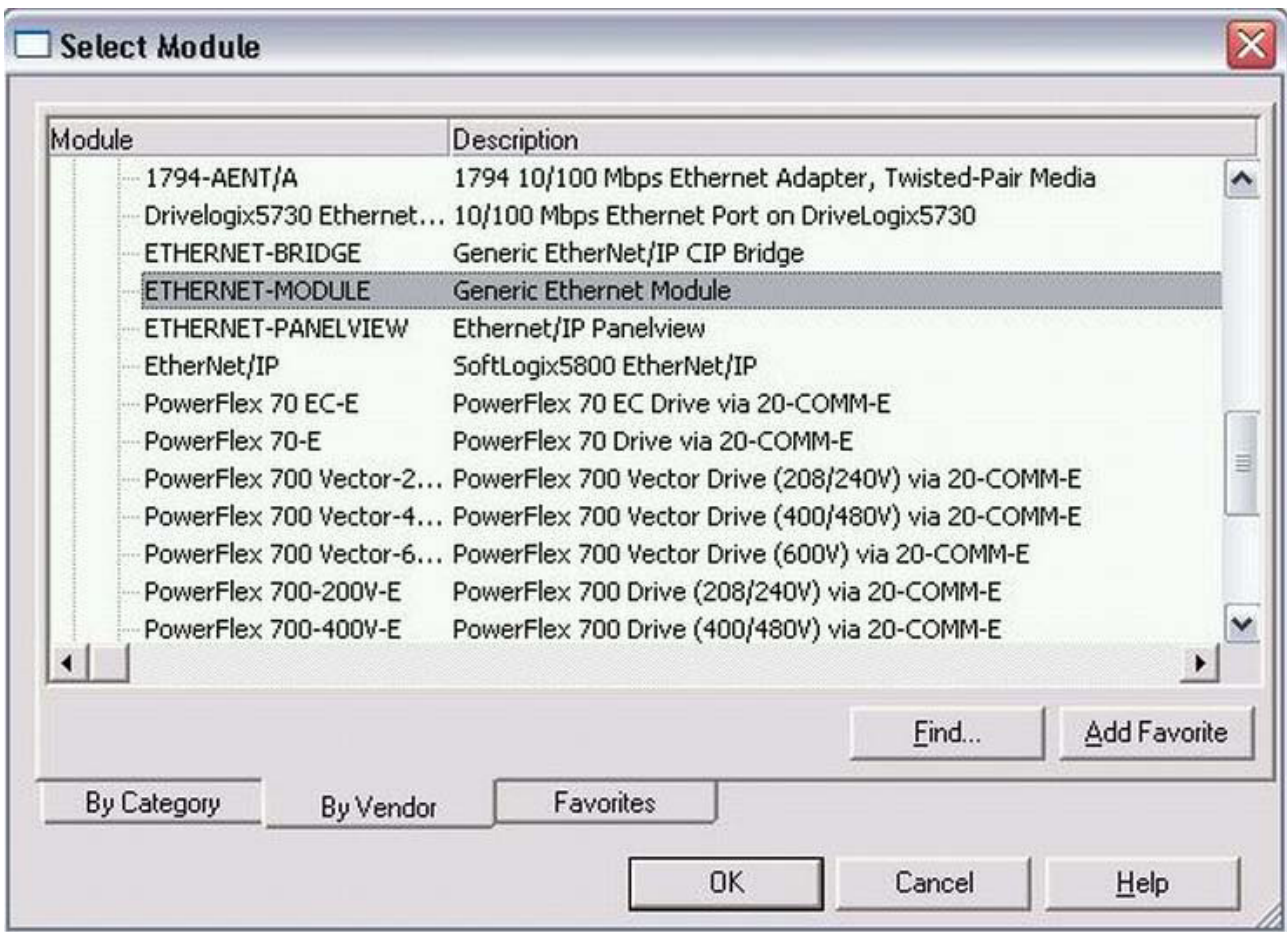


Fig. 146: Selection of the ETHERNET MODULE "Generic Ethernet Module"

Enter the IP address from object 0x8000:21. The input instance is to be entered as 129<sub>dec</sub>, the output instance as 130<sub>dec</sub> and the config instance as 128<sub>dec</sub>. The data length is dependent on the Comm format. INT was selected in the illustration "Transferring the parameters into the "New modules" mask", therefore the number of data items from objects 0x8001:05 and 0x8001:07 must be divided once again by 2, since these are specified in TwinCAT in bytes and on the RSLogix in Word length (INT). If an odd number of bytes is created you must round up; this also applies even if the Comm format is set to DINT, in which case you must round up to the next whole number.

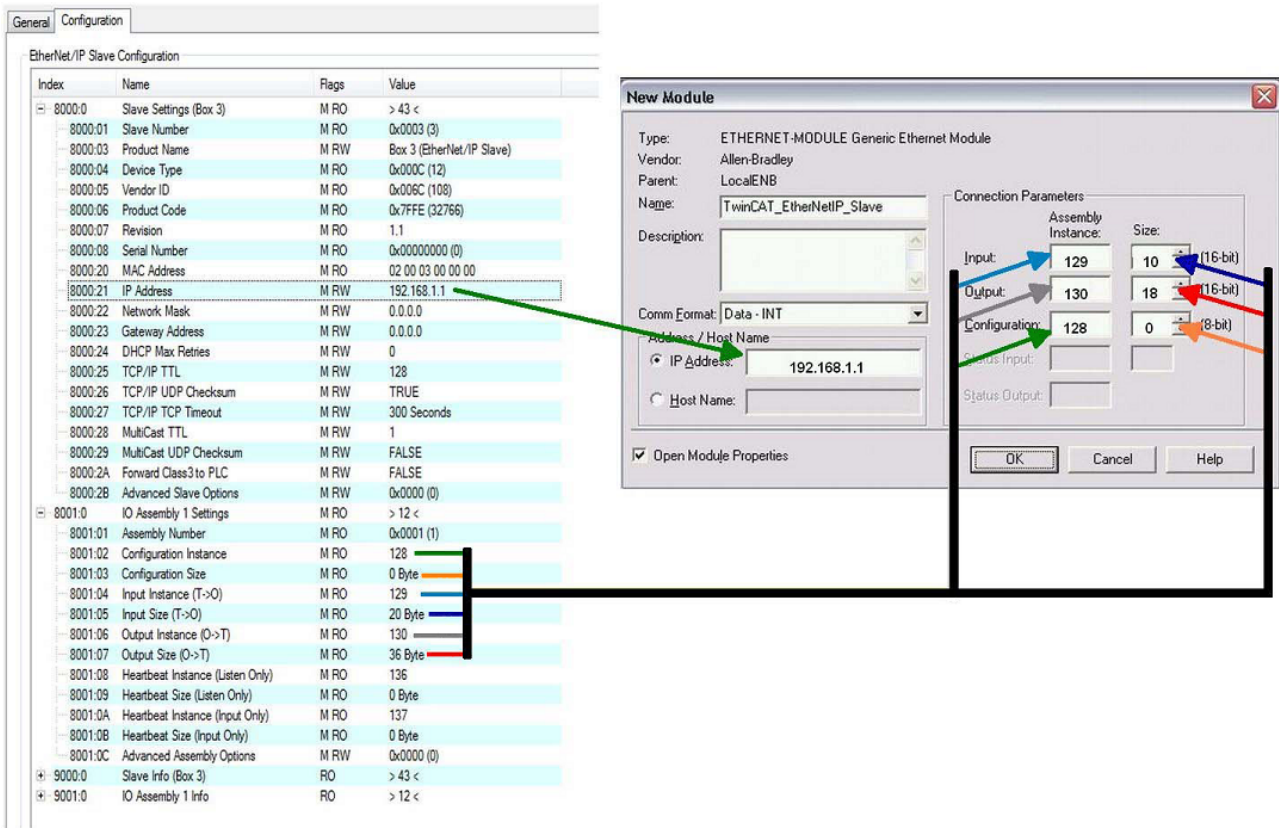


Fig. 147: Transferring the parameters into the "New Modules" mask



# 6 Configuration with the Twincat System Manager

## 6.1 Object description and parameterization

### ● EtherCAT XML device description and configuration files

**i** We recommend downloading the latest [EtherCAT Device Description](#) from the download area on the Beckhoff website and installing it according to the installation instructions.

### ● Parameterization

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

### Introduction

All objects necessary for the commissioning of the EL6652 and EL6652-0010 are configured by the TwinCAT system. Nothing needs to be set or changed in the objects.

### Objects for commissioning

#### Index 1011 Restore default parameters

Index	Name	Meaning	Format	Flags	Default value
1011:0	Restore default parameters	Restore default parameters	UINT8	RO	0x01 (1 <sub>dec</sub> )
1011:01	SubIndex 001	If this object is set to "0x64616F6C" in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0 <sub>dec</sub> )

### Complete overview

### Standard objects

#### Index 1000 Device type

Index (hex)	Name	Meaning	Format	Flags	Default value
1000:0	Device type	Device type of the EtherCAT slave: the Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x00001389 (5001 <sub>dec</sub> )

#### Index 1008 Device name

Index (hex)	Name	Meaning	Format	Flags	Default value
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL6652

#### Index 1009 Hardware version

Index (hex)	Name	Meaning	Format	Flags	Default value
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	variable

#### Index 100A Software version

Index (hex)	Name	Meaning	Format	Flags	Default value
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	variable

**Index 1018 Identity**

Index (hex)	Name	Meaning	Format	Flags	Default value
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4 <sub>dec</sub> )
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2 <sub>dec</sub> )
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x19FC3052 (43597842 <sub>dec</sub> )
1018:03	Revision	Revision number of the EtherCAT slave; the low word (bit 0-15) indicates the special terminal number, the high word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1018:04	Serial number	Serial number of the EtherCAT slave; the low byte (bit 0-7) of the low word contains the year of production, the high byte (bit 8-15) of the low word contains the week of production, the high word (bit 16-31) is 0	UINT32	RO	variable

**Index 10F0 Backup parameter handling**

Index (hex)	Name	Meaning	Format	Flags	Default value
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries	UINT8	RO	0x01 (1 <sub>dec</sub> )
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	variable

**Index 1600 IO RxPDO-Map**

Index	Name	Meaning	Format	Flags	Default value
1600:0	IO RxPDOPDO-Map	PDO Mapping RxPDO 1	UINT8	RO	0x02 (2 <sub>dec</sub> )
1600:01	SubIndex 001	1. PDO Mapping entry (16 bits align)	UINT32	RO	0x7000:11, 16

**Index 1C13 TxPDO assign**

Index (hex)	Name	Meaning	Format	Flags	Default value
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x05 (5 <sub>dec</sub> )
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656 <sub>dec</sub> )

**Index 1C32 SM output parameter**

Index (hex)	Name	Meaning	Format	Flags	Default value
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32 <sub>dec</sub> )
1C32:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> <li>• 0: Free Run</li> <li>• 1: Synchronous with SM 2 event</li> <li>• 2: DC-Mode - Synchronous with SYNC0 Event</li> <li>• 3: DC-Mode - Synchronous with SYNC1 event</li> </ul>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C32:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> <li>• Free Run: Cycle time of the local timer</li> <li>• Synchronous with SM 2 event: Master cycle time</li> <li>• DC-Mode: SYNC0/SYNC1 Cycle Time</li> </ul>	UINT32	RW	0x000186A0 (100000 <sub>dec</sub> )
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> <li>• Bit 0 = 1: free run is supported</li> <li>• Bit 1 = 1: Synchronous with SM 2 event is supported</li> <li>• Bit 2-3 = 01: DC mode is supported</li> <li>• Bit 4-5 = 10: Output shift with SYNC1 event (only DC mode)</li> <li>• Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▶ 123])</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dec</sub> )
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000186A0 (100000 <sub>dec</sub> )
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:08	Command	<ul style="list-style-type: none"> <li>• 0: Measurement of the local cycle time is stopped</li> <li>• 1: Measurement of the local cycle time is started</li> </ul> <p>The entries 0x1C32:03 [▶ 123], 0x1C32:05 [▶ 123], 0x1C32:06 [▶ 123], 0x1C32:09 [▶ 123], 0x1C33:03 [▶ 124], 0x1C33:06 [▶ 123], and 1C33:09 [▶ 124] are updated with the maximum measured values. For a subsequent measurement the measured values are reset</p>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C32:09	Delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

**Index 1C33 SM input parameter**

Index (hex)	Name	Meaning	Format	Flags	Default value
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32 <sub>dec</sub> )
1C33:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> <li>• 0: Free Run</li> <li>• 1: Synchronous with SM 3 event (no outputs available)</li> <li>• 2: DC - Synchronous with SYNC0 Event</li> <li>• 3: DC - Synchronous with SYNC1 Event</li> <li>• 34: Synchronous with SM 2 event (outputs available)</li> </ul>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C33:02	Cycle time	as <a href="#">0x1C32:02</a> [ <a href="#">▶ 123</a> ]	UINT32	RW	0x000186A0 (100000 <sub>dec</sub> )
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> <li>• Bit 0: free run is supported</li> <li>• Bit 1: synchronous with SM 2 event is supported (outputs available)</li> <li>• Bit 1: synchronous with SM 3 event is supported (no outputs available)</li> <li>• Bit 2-3 = 01: DC mode is supported</li> <li>• Bit 4-5 = 01: input shift through local event (outputs available)</li> <li>• Bit 4-5 = 10: input shift with SYNC1 event (no outputs available)</li> <li>• Bit 14 = 1: dynamic times (measurement through writing of <a href="#">0x1C32:08</a> [<a href="#">▶ 123</a>] or <a href="#">0x1C33:08</a> [<a href="#">▶ 124</a>])</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dec</sub> )
1C33:05	Minimum cycle time	as <a href="#">0x1C32:05</a> [ <a href="#">▶ 123</a> ]	UINT32	RO	0x000186A0 (100000 <sub>dec</sub> )
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:08	Command	as <a href="#">0x1C32:08</a> [ <a href="#">▶ 123</a> ]	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C33:09	Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:0B	SM event missed counter	as <a href="#">0x1C32:11</a> [ <a href="#">▶ 123</a> ]	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:0C	Cycle exceeded counter	as <a href="#">0x1C32:12</a> [ <a href="#">▶ 123</a> ]	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:0D	Shift too short counter	as <a href="#">0x1C32:13</a> [ <a href="#">▶ 123</a> ]	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:20	Sync error	as <a href="#">0x1C32:32</a> [ <a href="#">▶ 123</a> ]	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

**Profile-specific objects**

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

**Index F000 Modular device profile**

Index (hex)	Name	Meaning	Format	Flags	Default value
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2 <sub>dec</sub> )
F000:01	Module index distance	Index spacing of the objects of the individual channels	UINT16	RO	0x0010 (16 <sub>dec</sub> )
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0005 (5 <sub>dec</sub> )

**Index F008 Code word**

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

**Index F010 Module list**

Index (hex)	Name	Meaning	Format	Flags	Default value
F010:0	Module list	Max. subindex	UINT8	RW	0x05 (5 <sub>dec</sub> )
F010:01	SubIndex 001	-	UINT32	RW	0x00000320 (0800 <sub>dec</sub> )

# 7 Diagnosis

## 7.1 EL6652-0010 - LEDs

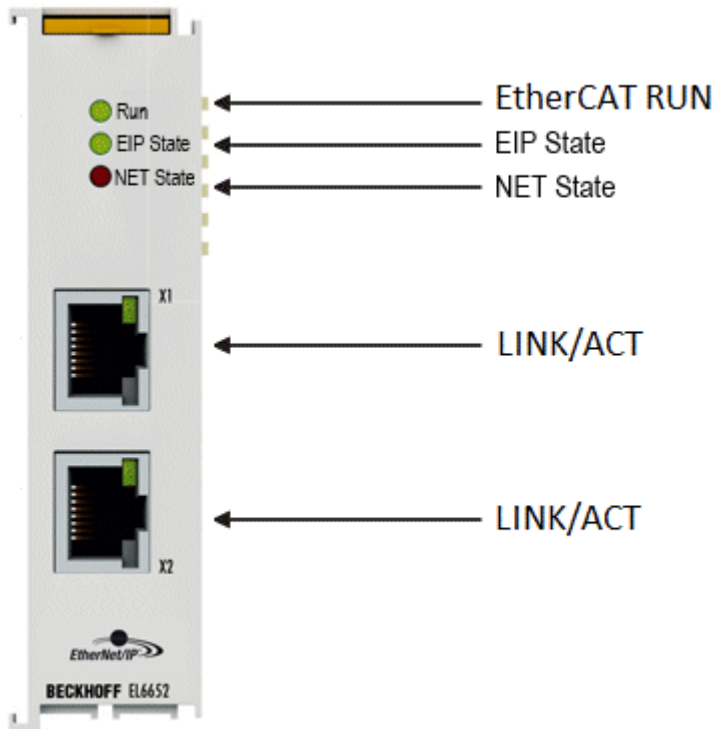


Fig. 148: EL6652 LEDs

### LEDs for EtherCAT diagnosis

LED	Display	Description
RUN	green	off
		flashes (200 ms)
		off (1 s) on (200 ms)
		on
		State of the EtherCAT state machine: <b>INIT</b> = initialization of the terminal; <b>BOOTSTRAP</b> = function for terminal firmware updates State of the EtherCAT state machine: <b>PREOP</b> = function for mailbox communication and different standard-settings set State of the EtherCAT state machine: <b>SAFEOP</b> = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state State of the EtherCAT state machine: <b>OP</b> = normal operating state; mailbox and process data communication is possible

**LED EIP State**

Color green	Color red	Meaning
off	off	no EtherNet/IP slave configuration on the EL6652-0010
on	off	All configured IO assemblies are in data exchange with the EtherNet/IP master. All connections are in the run state (cyclic exchange of valid process data)
off (1 s) on (200 ms)	off	EtherNet/IP slave and IO assembly are correctly parameterized
flashes (400 ms)	off	One of the EtherNet/IP slaves has no valid IO assembly configuration
off	off (1 s) on (200 ms)	A general error occurred with the EtherNet/IP slave
off	on	Internal error. Replace the EtherCAT Terminal

**LED NET State**

Color green	Color red	Meaning
off	off	No link detected
on	off	EL terminal has detected a link and was configured correctly
flashes (400 ms)	off	At least one Ethernet port has an active link and a configured EtherNet/IP Slave interface has no valid IP address configured
off (1 s) on (200 ms)	off	All configured EtherNet/I slaves have a valid IP address configuration. UDP and TCP Layer was initialized
off	on	Internal error. Replace the EtherCAT Terminal
off	off (1 s) on (200 ms)	A general error occurred with the EtherNet/IP slave

**LEDs starting up**

Run	EIP State	NET State	Meaning
off	off	off	No electrical voltage connected to E-bus. The EL6652-0010 must be exchanged if EtherCAT Terminals behind it need to function.
off	off	red on	EL terminal is starting up; after approx. 10 seconds, the LED should go out. If this does not happen, the EL6652-0010 module must be exchanged.
on	on	red on	EL terminal starts up; the terminal should be ready after approx. 2 - 3 s; the startup is completed as soon as the red "NET State" LED goes out.

## 7.2 EL6652-0000, EL6652-0010 diagnostic history

The diagnostic history is an aid for monitoring the status of the EtherNet/IP interface, the status of the terminal and the Ethernet port and to display the diagnostic messages in plain text with a time stamp.

Furthermore, information/errors that occurred in the past or only briefly are also logged in order to also allow precise fault finding at a later time.

The diagnostic history is a component of the System Manager and can be found there under the EL6652 in the "Diag History" tab.

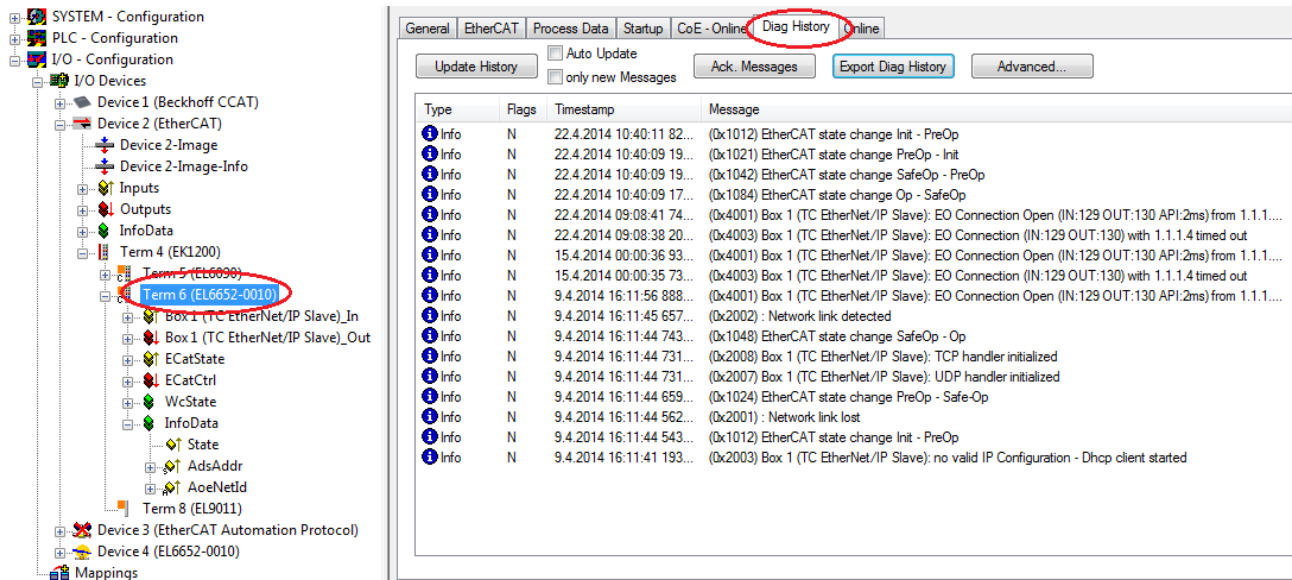


Fig. 149: "Diag History" tab



**Error Codes EL6652**

Error	Code	Description	Remedy
CN_ORC_ALREADY_USED	0x100 hex / 256 dec	Connection already in use	The connection is already established; use another connection or close this one
CN_ORC_BAD_TRANSPORT	0x103 hex / 259 dec	Transport type not supported	
CN_ORC_OWNER_CONFLICT	0x106 hex / 262 dec	More than one guy configuring	
CN_ORC_BAD_CONNECTION	0x107 hex / 263 dec	Trying to close inactive conn	
CN_ORC_BAD_CONNECTION_TYPE	0x108 hex / 264 dec	Unsupported connection type	The connection type is not supported, check your settings
CN_ORC_BAD_CONNECTION_SIZE	0x109 hex / 265 dec	Connection size mismatch	The connection size does not match, check your settings
CN_ORC_CONN_UNCONFIGURED	0x110 hex / 272 dec	Connection unconfigured	
CN_ORC_BAD_RPI	0x111 hex / 273 dec	Unsupportable RPI	The task time usually doesn't match; make sure that the EL6652 operates internally with 1 ms and that you can adjust this with the Cycle Time Multiplier. Otherwise, adapt the task time to suit.
CN_ORC_NO_CM_RESOURCES	0x113 hex / 275 dec	Conn Mgr out of connections	
CN_ORC_BAD_VENDOR_PRODUCT	0x114 hex / 276 dec	Mismatch in electronic key	
CN_ORC_BAD_DEVICE_TYPE	0x115 hex / 277 dec	Mismatch in electronic key	
CN_ORC_BAD_REVISION	0x116 hex / 278 dec	Mismatch in electronic key	
CN_ORC_BAD_CONNECTION_POINT	0x117 hex / 279 dec	Nonexistent instance number	
CN_ORC_BAD_CONFIGURATION	0x118 hex / 280 dec	Bad config instance number	
CN_ORC_CONNECTION_REQUEST_FAILURES	0x119 hex / 281 dec	No controlling connection open	
CN_ORC_NO_APPLICATION_RESOURCES	0x11A hex / 282 dec	App out of connections	No further free connection available

If you cannot fix this error yourself, Support will require the following information:

EDS file of the slave, TwinCAT Build, firmware version of the EL6652 and a Wireshark recording (please make this using a switch with port mirroring or a hub).

## 8 Appendix

### 8.1 Firmware Update EL/ES/EM/ELM/EPxxxx

This section describes the device update for Beckhoff EtherCAT slaves from the EL/ES, ELM, EM, EK and EP series. A firmware update should only be carried out after consultation with Beckhoff support.

#### Storage locations

An EtherCAT slave stores operating data in up to 3 locations:

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in \*.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with \*.rbf firmware.
- In addition, each EtherCAT slave has a memory chip, a so-called **ESI-EEPROM**, for storing its own device description (ESI: EtherCAT Slave Information). On power-up this description is loaded and the EtherCAT communication is set up accordingly. The device description is available from the download area of the Beckhoff website at (<https://www.beckhoff.de>). All ESI files are accessible there as zip files.

Customers can access the data via the EtherCAT fieldbus and its communication mechanisms. Acyclic mailbox communication or register access to the ESC is used for updating or reading of these data.

The TwinCAT System Manager offers mechanisms for programming all 3 parts with new data, if the slave is set up for this purpose. Generally the slave does not check whether the new data are suitable, i.e. it may no longer be able to operate if the data are unsuitable.

#### Simplified update by bundle firmware

The update using so-called **bundle firmware** is more convenient: in this case the controller firmware and the ESI description are combined in a \*.efw file; during the update both the firmware and the ESI are changed in the terminal. For this to happen it is necessary

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxx-xxx\_REV0016\_SW01.efw
- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

- ESI/Revision: e.g. by means of an online scan in TwinCAT ConfigMode/FreeRun – this is a convenient way to determine the revision
- Firmware: e.g. by looking in the online CoE of the device

#### NOTE

##### Risk of damage to the device!

Note the following when downloading new device files

- Firmware downloads to an EtherCAT device must not be interrupted
- Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.
- The power supply must adequately dimensioned. The signal level must meet the specification.

In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

### 8.1.1 Device description ESI file/XML

**NOTE**

**Attention regarding update of the ESI description/EEPROM**

Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.

The ESI device description is stored locally on the slave and loaded on start-up. Each device description has a unique identifier consisting of slave name (9 characters/digits) and a revision number (4 digits). Each slave configured in the System Manager shows its identifier in the EtherCAT tab:

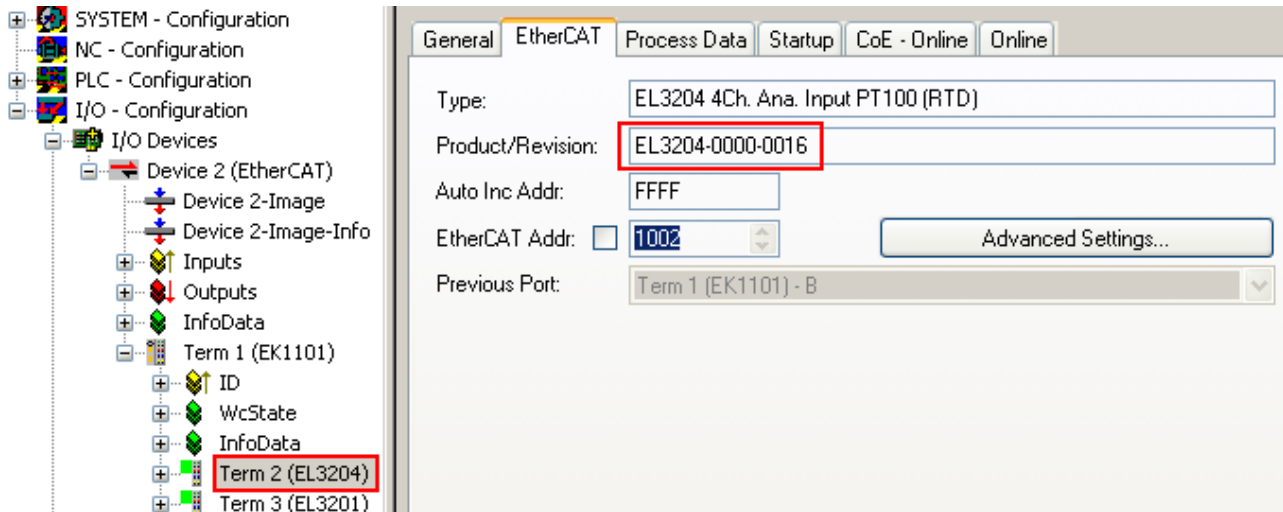


Fig. 150: Device identifier consisting of name EL3204-0000 and revision -0016

The configured identifier must be compatible with the actual device description used as hardware, i.e. the description which the slave has loaded on start-up (in this case EL3204). Normally the configured revision must be the same or lower than that actually present in the terminal network.

For further information on this, please refer to the [EtherCAT system documentation](#).

**● Update of XML/ESI description**

**i** The device revision is closely linked to the firmware and hardware used. Incompatible combinations lead to malfunctions or even final shutdown of the device. Corresponding updates should only be carried out in consultation with Beckhoff support.

**Display of ESI slave identifier**

The simplest way to ascertain compliance of configured and actual device description is to scan the EtherCAT boxes in TwinCAT mode Config/FreeRun:

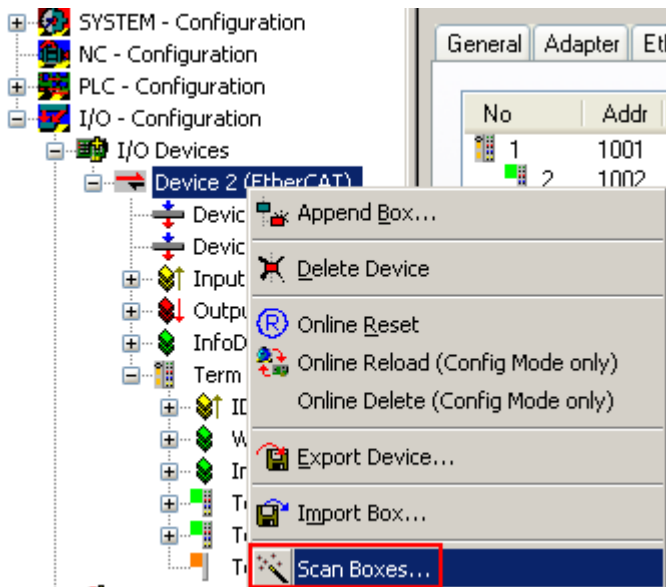


Fig. 151: Scan the subordinate field by right-clicking on the EtherCAT device

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



Fig. 152: Configuration is identical

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

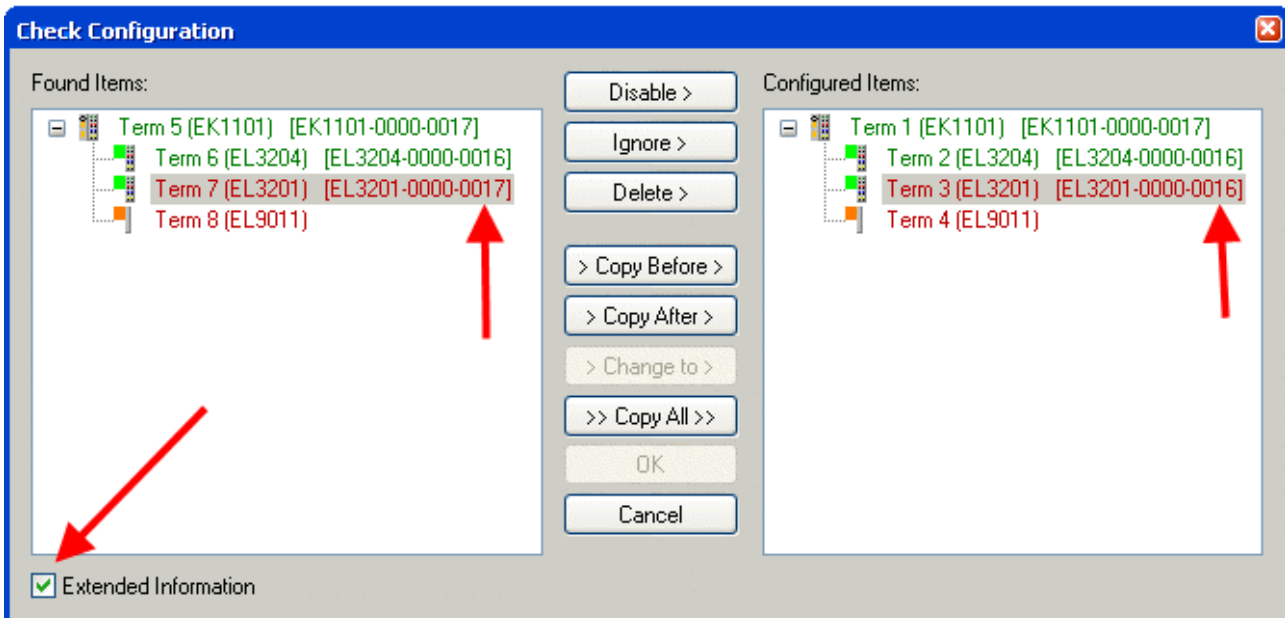


Fig. 153: Change dialog

In this example in Fig. *Change dialog*, an EL3201-0000-0017 was found, while an EL3201-0000-0016 was configured. In this case the configuration can be adapted with the *Copy Before* button. The *Extended Information* checkbox must be set in order to display the revision.

### Changing the ESI slave identifier

The ESI/EEPROM identifier can be updated as follows under TwinCAT:

- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*

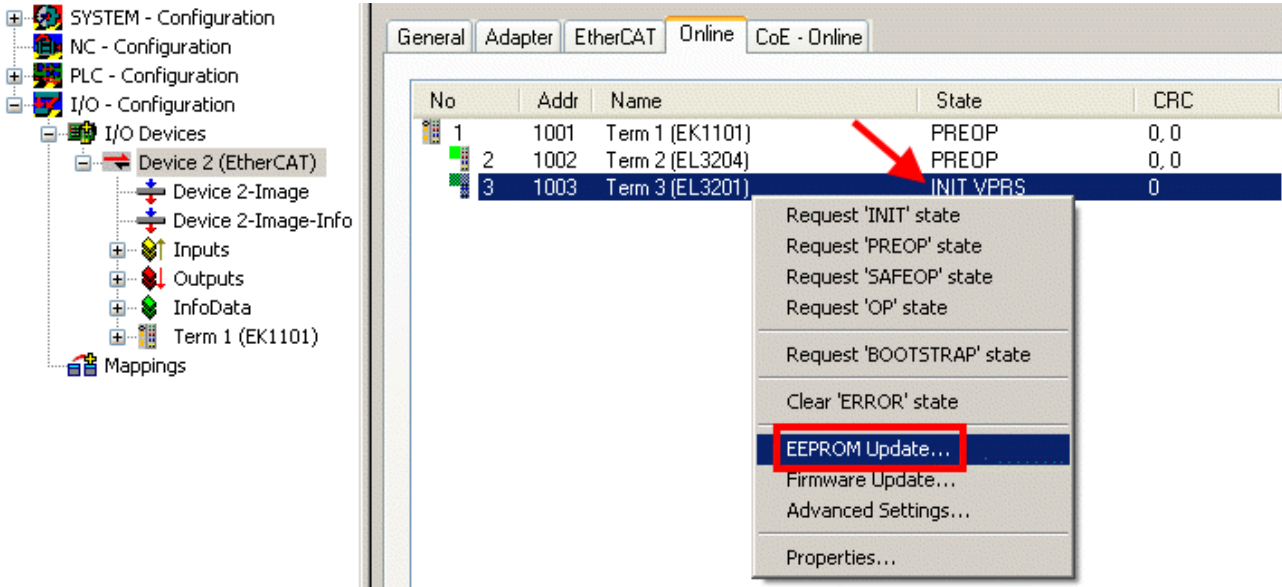


Fig. 154: EEPROM Update

The new ESI description is selected in the following dialog, see Fig. *Selecting the new ESI*. The checkbox *Show Hidden Devices* also displays older, normally hidden versions of a slave.

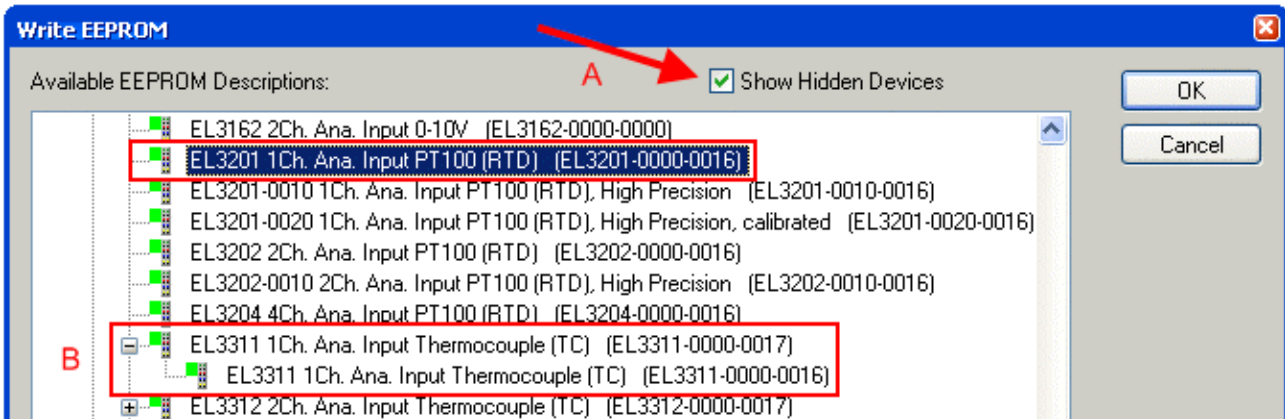


Fig. 155: Selecting the new ESI

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

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

Most EtherCAT devices read a modified ESI description immediately or after startup from the INIT. Some communication settings such as distributed clocks are only read during power-on. The EtherCAT slave therefore has to be switched off briefly in order for the change to take effect.

## 8.1.2 Firmware explanation

### Determining the firmware version

#### Determining the version on laser inscription

Beckhoff EtherCAT slaves feature serial numbers applied by laser. The serial number has the following structure: **KK YY FF HH**

KK - week of production (CW, calendar week)  
 YY - year of production  
 FF - firmware version  
 HH - hardware version

Example with ser. no.: 12 10 03 02:

12 - week of production 12  
 10 - year of production 2010  
 03 - firmware version 03  
 02 - hardware version 02

#### Determining the version via the System Manager

The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

#### ● CoE Online and Offline CoE

**i** Two CoE directories are available:

- **online**: This is offered in the EtherCAT slave by the controller, if the EtherCAT slave supports this. This CoE directory can only be displayed if a slave is connected and operational.
- **offline**: The EtherCAT Slave Information ESI/XML may contain the default content of the CoE. This CoE directory can only be displayed if it is included in the ESI (e.g. "Beckhoff EL5xxx.xml").

The Advanced button must be used for switching between the two views.

In Fig. *Display of EL3204 firmware version* the firmware version of the selected EL3204 is shown as 03 in CoE entry 0x100A.

The screenshot displays the TwinCAT System Manager interface. On the left, a tree view shows the configuration structure, with 'Term 2 (EL3204)' selected and highlighted in red. The main window shows the 'CoE - Online' tab. The 'Advanced Settings' section is expanded, and the 'Online Data' button is highlighted in red. Below this, a table lists CoE entries:

Index	Name	Flags	Value
1000	Device type	RO	0x01401389 (20976521)
1008	Device name	RO	EL3204-0000
1009	Hardware version	RO	00
100A	Software version	RO	03
1011:0	Restore default parameters	HU	> 1 <

The '100A Software version' entry is highlighted in red. Below the table, the 'Dictionary' section is visible, with the 'All Objects' button highlighted in red. The 'Online' radio button is selected in the 'Dictionary' section.

Fig. 156: Display of EL3204 firmware version

In (A) TwinCAT 2.11 shows that the Online CoE directory is currently displayed. If this is not the case, the Online directory can be loaded via the *Online* option in Advanced Settings (B) and double-clicking on *AllObjects*.

### 8.1.3 Updating controller firmware \*.efw

**i** **CoE directory**

The Online CoE directory is managed by the controller and stored in a dedicated EEPROM, which is generally not changed during a firmware update.

Switch to the *Online* tab to update the controller firmware of a slave, see Fig. *Firmware Update*.

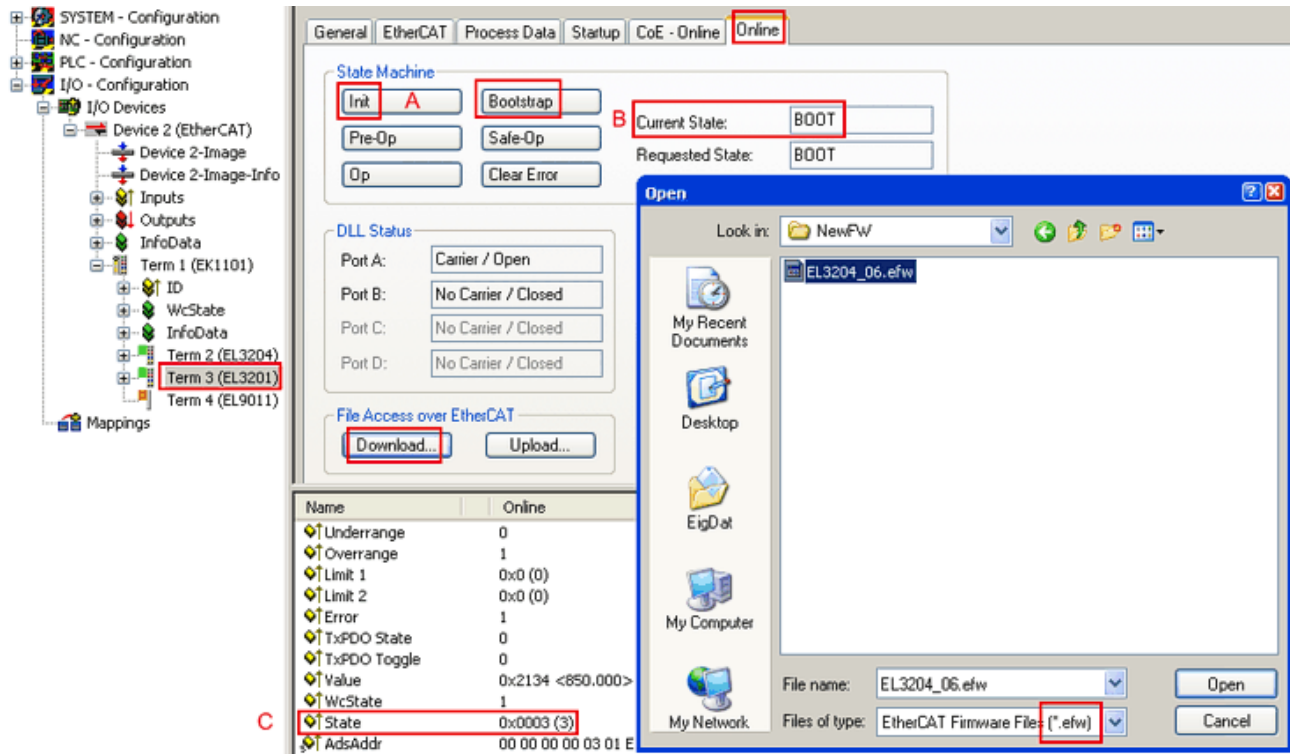
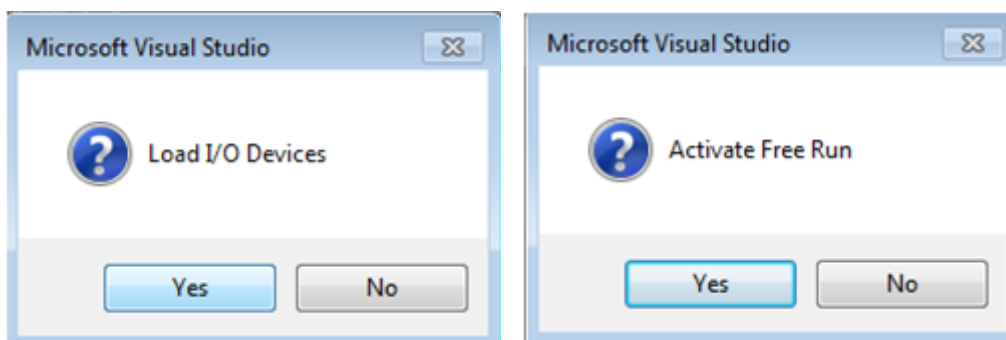


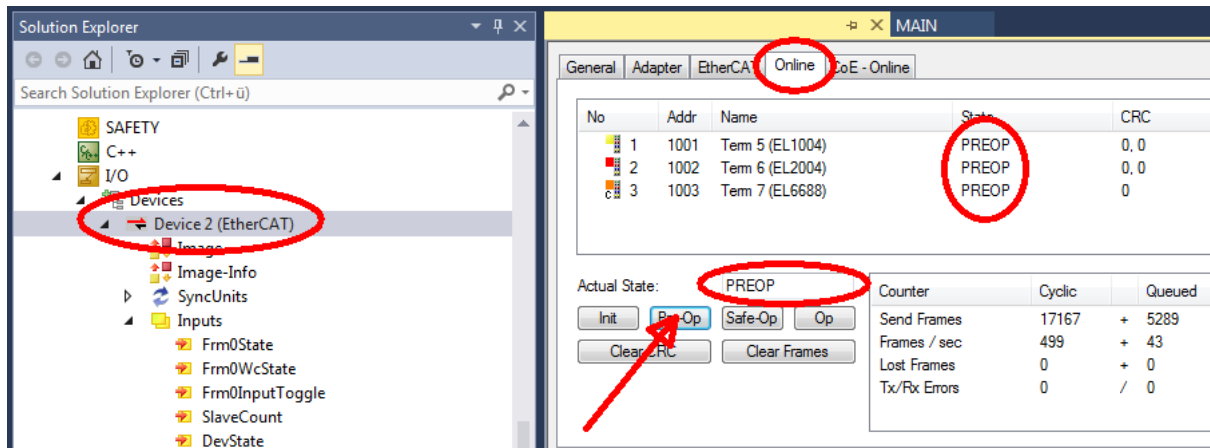
Fig. 157: Firmware Update

Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

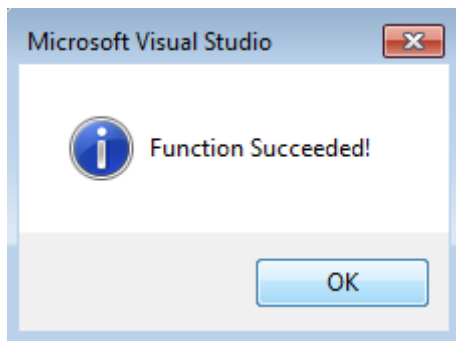
- Switch TwinCAT system to ConfigMode/FreeRun with cycle time  $\geq 1$  ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.



- Switch EtherCAT Master to PreOP



- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP
- Check the current status (B, C)
- Download the new \*.efw file (wait until it ends). A pass word will not be necessary usually.



- After the download switch to INIT, then PreOP
- Switch off the slave briefly (don't pull under voltage!)
- Check within CoE 0x100A, if the FW status was correctly overtaken.

### 8.1.4 FPGA firmware \*.rbf

If an FPGA chip deals with the EtherCAT communication an update may be accomplished via an \*.rbf file.

- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

The firmware version number included in the terminal serial number contains both firmware components. If one of these firmware components is modified this version number is updated.

#### Determining the version via the System Manager

The TwinCAT System Manager indicates the FPGA firmware version. Click on the Ethernet card of your EtherCAT strand (Device 2 in the example) and select the *Online* tab.

The *Reg:0002* column indicates the firmware version of the individual EtherCAT devices in hexadecimal and decimal representation.



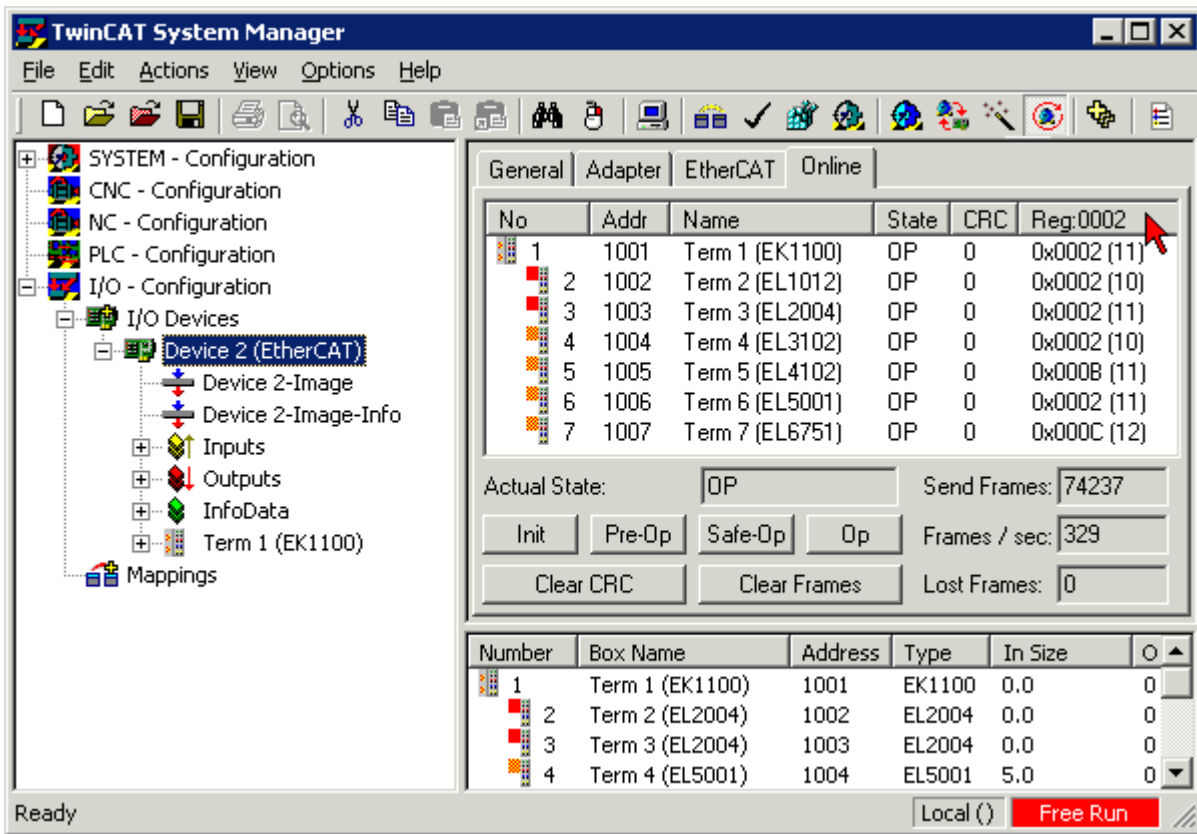


Fig. 158: FPGA firmware version definition

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

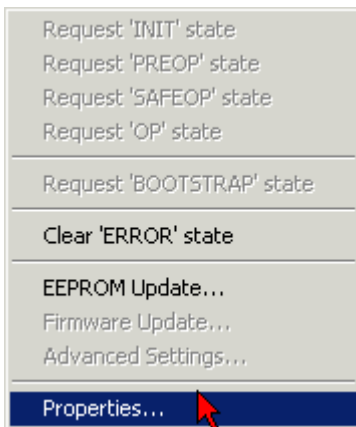


Fig. 159: Context menu *Properties*

The *Advanced Settings* dialog appears where the columns to be displayed can be selected. Under *Diagnosis/Online View* select the *'0002 ETxxxx Build'* check box in order to activate the FPGA firmware version display.

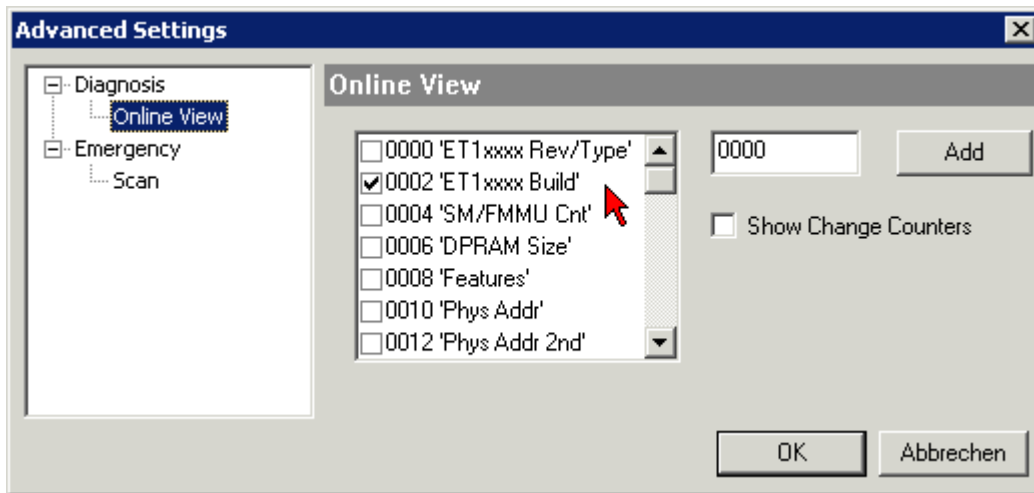


Fig. 160: Dialog *Advanced Settings*

### Update

For updating the FPGA firmware

- of an EtherCAT coupler the coupler must have FPGA firmware version 11 or higher;
- of an E-Bus Terminal the terminal must have FPGA firmware version 10 or higher.

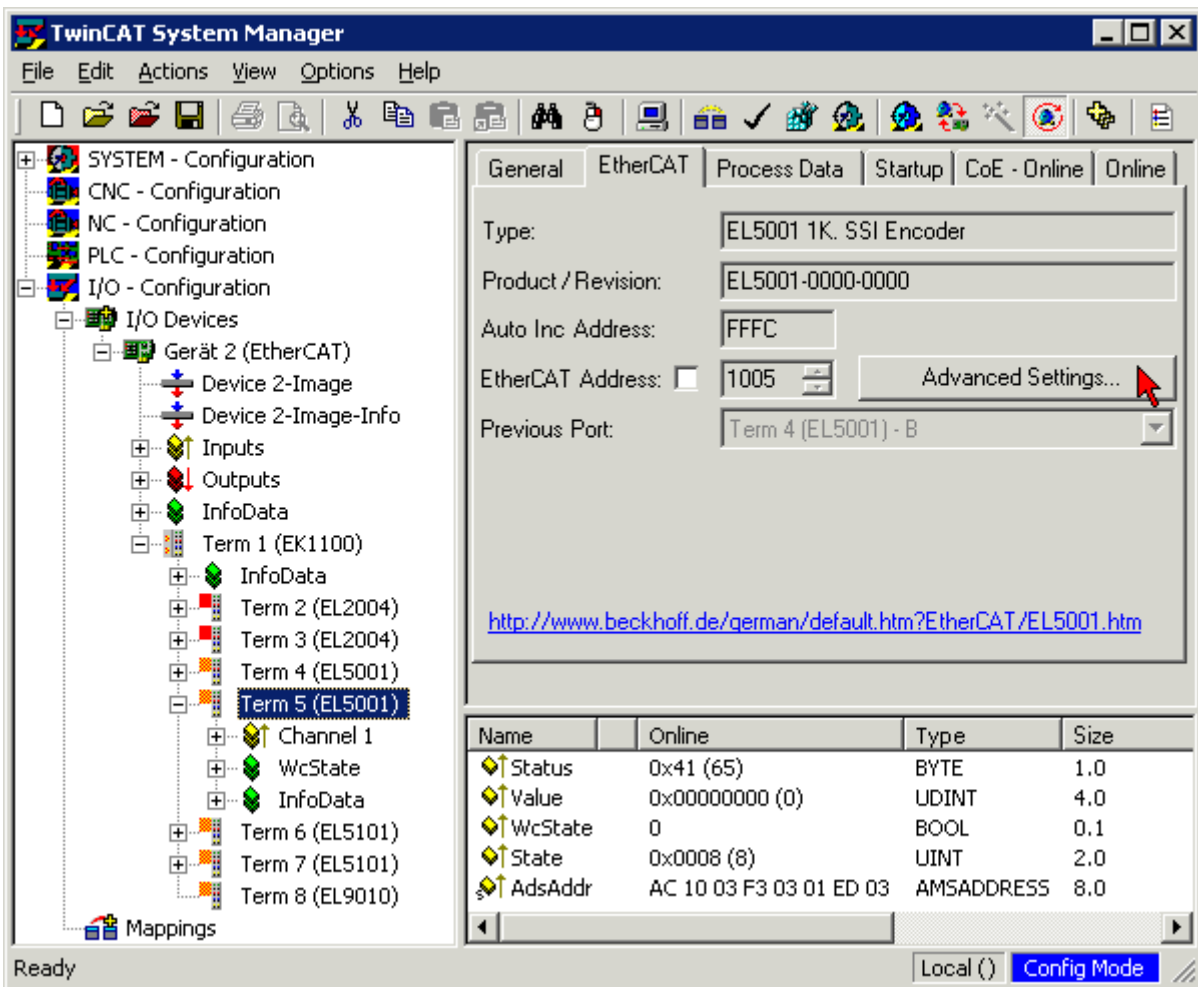
Older firmware versions can only be updated by the manufacturer!

### Updating an EtherCAT device

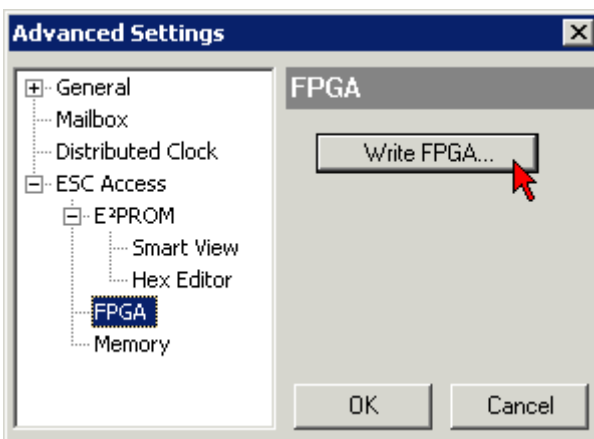
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

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

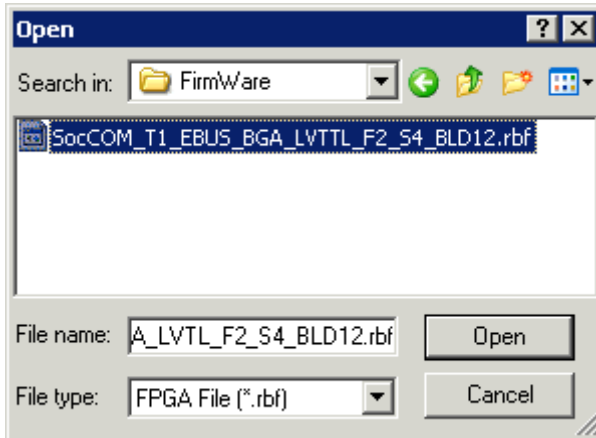
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



- The *Advanced Settings* dialog appears. Under *ESC Access/E<sup>2</sup>PROM/FPGA* click on *Write FPGA* button:



- Select the file (\*.rbf) with the new FPGA firmware, and transfer it to the EtherCAT device:



- Wait until download ends
- Switch slave current less for a short time (don't pull under voltage!). In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.
- Check the new FPGA status

### NOTE

#### **Risk of damage to the device!**

A download of firmware to an EtherCAT device must not be interrupted in any case! If you interrupt this process by switching off power supply or disconnecting the Ethernet link, the EtherCAT device can only be recommissioned by the manufacturer!

## 8.1.5 Simultaneous updating of several EtherCAT devices

The firmware and ESI descriptions of several devices can be updated simultaneously, provided the devices have the same firmware file/ESI.

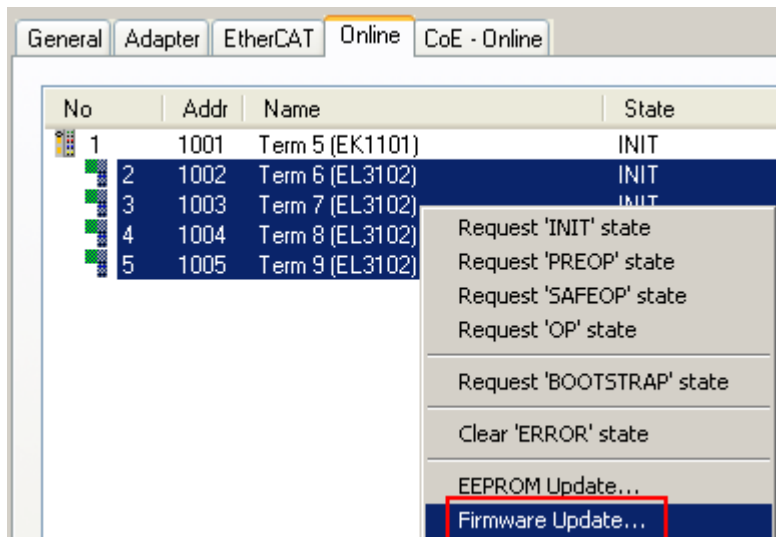


Fig. 161: Multiple selection and firmware update

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

## 8.2 Firmware compatibility

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

**Note**

- It is recommended to use the newest possible firmware for the respective hardware.
- Beckhoff is not under any obligation to provide customers with free firmware updates for delivered products.

<b>NOTE</b>			
<b>Risk of damage to the device!</b>			
Pay attention to the instructions for firmware updates on the separate page. If a device is placed in BOOT-STRAP mode for a firmware update, it does not check when downloading whether the new firmware is suitable. This can result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!			

<b>EL6652-0000</b>			
Hardware (HW)	Firmware (FW)	Revision No.	Release date
10 – 14*	01*	EL6652-0000-0017	2015/04

<b>EL6652-0010</b>			
Hardware (HW)	Firmware (FW)	Revision No.	Release date
10 – 14*	01*	EL6652-0010-0017	2015/04

\*) This is the current compatible firmware/hardware version at the time of the preparing this documentation. Check on the Beckhoff web page whether more up-to-date [documentation](#) is available.

## 8.3 Restoring the delivery state

To restore the delivery state for backup objects in ELxxx terminals, the CoE object Restore default parameters, *SubIndex 001* can be selected in the TwinCAT System Manager (Config mode) (see Fig. *Selecting the Restore default parameters PDO*)

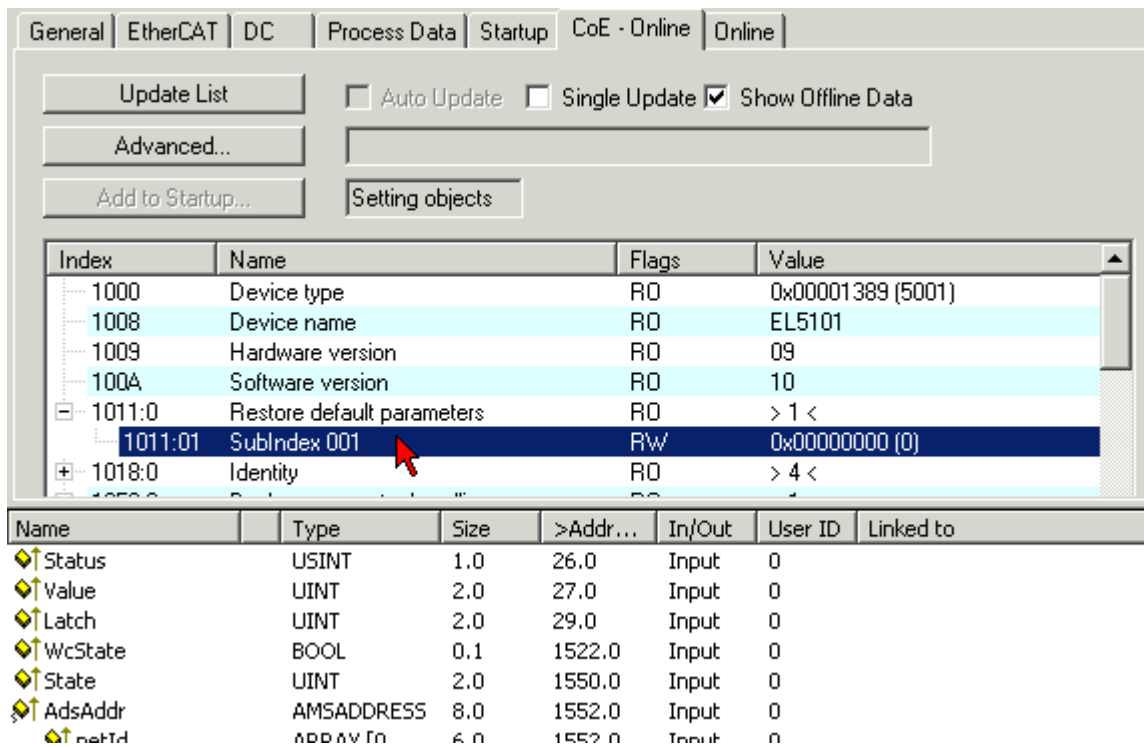


Fig. 162: Selecting the "Restore default parameters" PDO

Double-click on SubIndex 001 to enter the Set Value dialog. Enter the value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* and confirm with *OK* (Fig. *Entering a restore value in the Set Value dialog*). All backup objects are reset to the delivery state.

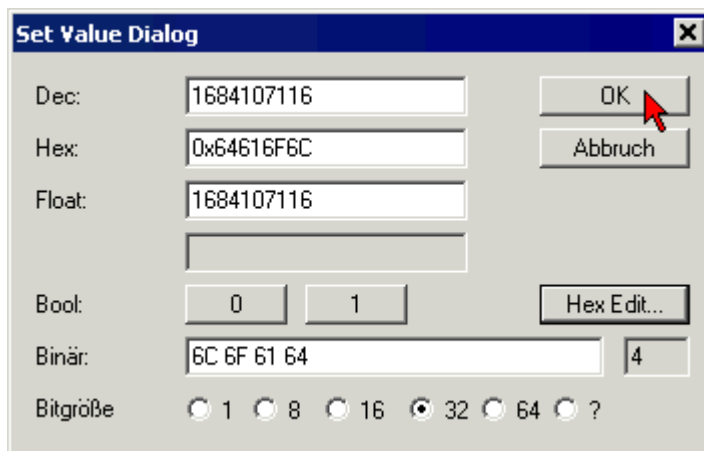


Fig. 163: Entering a restore value in the Set Value dialog

### ● Alternative restore value

**I** In some older terminals 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.

## 8.4 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:

<http://www.beckhoff.com>

You will also find further documentation for Beckhoff components there.

**Beckhoff Headquarters**

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20  
33415 Verl  
Germany

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

**Beckhoff Support**

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

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

Hotline: +49 5246 963 157  
Fax: +49 5246 963 9157  
e-mail: [support@beckhoff.com](mailto:support@beckhoff.com)

**Beckhoff Service**

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

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

Hotline: +49 5246 963 460  
Fax: +49 5246 963 479  
e-mail: [service@beckhoff.com](mailto:service@beckhoff.com)

## Table of figures

Fig. 1	EL5021 EL terminal, standard IP20 IO device with serial/ batch number and revision ID (since 2014/01).....	9
Fig. 2	EK1100 EtherCAT coupler, standard IP20 IO device with serial/ batch number.....	10
Fig. 3	CU2016 switch with serial/ batch number.....	10
Fig. 4	EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418 .....	10
Fig. 5	EP1258-00001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102.....	11
Fig. 6	EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070 .....	11
Fig. 7	EL2904 IP20 safety terminal with batch number/ date code 50110302 and unique serial number 00331701.....	11
Fig. 8	ELM3604-0002 terminal with unique ID number (QR code) 100001051 and serial/ batch number 44160201.....	11
Fig. 9	BIC as data matrix code (DMC, code scheme ECC200).....	12
Fig. 10	EL6652-0000 .....	14
Fig. 11	EL6652-0010 .....	15
Fig. 12	System manager current calculation .....	18
Fig. 13	EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog .....	19
Fig. 14	States of the EtherCAT State Machine.....	21
Fig. 15	"CoE Online " tab .....	23
Fig. 16	Startup list in the TwinCAT System Manager .....	24
Fig. 17	Offline list.....	25
Fig. 18	Online list .....	25
Fig. 19	Spring contacts of the Beckhoff I/O components.....	28
Fig. 20	Correct positioning.....	31
Fig. 21	Incorrect positioning.....	31
Fig. 22	Recommended distances for standard installation position .....	32
Fig. 23	Other installation positions .....	33
Fig. 24	Relationship between user side (commissioning) and installation.....	35
Fig. 25	Control configuration with Embedded PC, input (EL1004) and output (EL2008) .....	36
Fig. 26	Initial TwinCAT 2 user interface.....	37
Fig. 27	Selection of the target system .....	38
Fig. 28	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	38
Fig. 29	Select "Scan Devices..." .....	39
Fig. 30	Automatic detection of I/O devices: selection the devices to be integrated.....	39
Fig. 31	Mapping of the configuration in the TwinCAT 2 System Manager.....	40
Fig. 32	Reading of individual terminals connected to a device.....	40
Fig. 33	TwinCAT PLC Control after startup .....	41
Fig. 34	Sample program with variables after a compile process (without variable integration).....	42
Fig. 35	Appending the TwinCAT PLC Control project .....	42
Fig. 36	PLC project integrated in the PLC configuration of the System Manager .....	43
Fig. 37	Creating the links between PLC variables and process objects.....	43
Fig. 38	Selecting PDO of type BOOL .....	44
Fig. 39	Selecting several PDOs simultaneously: activate "Continuous" and "All types".....	44
Fig. 40	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample .....	45
Fig. 41	Choose target system (remote) .....	46



Fig. 42	PLC Control logged in, ready for program startup .....	47
Fig. 43	Initial TwinCAT 3 user interface .....	48
Fig. 44	Create new TwinCAT project .....	48
Fig. 45	New TwinCAT3 project in the project folder explorer .....	49
Fig. 46	Selection dialog: Choose the target system .....	49
Fig. 47	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	50
Fig. 48	Select "Scan" .....	50
Fig. 49	Automatic detection of I/O devices: selection the devices to be integrated .....	51
Fig. 50	Mapping of the configuration in VS shell of the TwinCAT3 environment .....	51
Fig. 51	Reading of individual terminals connected to a device .....	52
Fig. 52	Adding the programming environment in "PLC" .....	53
Fig. 53	Specifying the name and directory for the PLC programming environment .....	53
Fig. 54	Initial "Main" program of the standard PLC project .....	54
Fig. 55	Sample program with variables after a compile process (without variable integration) .....	55
Fig. 56	Start program compilation .....	55
Fig. 57	Creating the links between PLC variables and process objects .....	56
Fig. 58	Selecting PDO of type BOOL .....	56
Fig. 59	Selecting several PDOs simultaneously: activate "Continuous" and "All types" .....	57
Fig. 60	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample .....	57
Fig. 61	TwinCAT development environment (VS shell): logged-in, after program startup .....	58
Fig. 62	System Manager "Options" (TwinCAT 2) .....	60
Fig. 63	Call up under VS Shell (TwinCAT 3) .....	60
Fig. 64	Overview of network interfaces .....	60
Fig. 65	EtherCAT device properties(TwinCAT 2): click on „Compatible Devices...“ of tab “Adapter” .....	61
Fig. 66	Windows properties of the network interface .....	61
Fig. 67	Exemplary correct driver setting for the Ethernet port .....	62
Fig. 68	Incorrect driver settings for the Ethernet port .....	63
Fig. 69	TCP/IP setting for the Ethernet port .....	64
Fig. 70	Identifier structure .....	65
Fig. 71	OnlineDescription information window (TwinCAT 2) .....	66
Fig. 72	Information window OnlineDescription (TwinCAT 3) .....	66
Fig. 73	File OnlineDescription.xml created by the System Manager .....	67
Fig. 74	Indication of an online recorded ESI of EL2521 as an example .....	67
Fig. 75	Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3) .....	67
Fig. 76	Using the ESI Updater (>= TwinCAT 2.11) .....	69
Fig. 77	Using the ESI Updater (TwinCAT 3) .....	69
Fig. 78	Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3) .....	70
Fig. 79	Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3) .....	70
Fig. 80	Selecting the Ethernet port .....	70
Fig. 81	EtherCAT device properties (TwinCAT 2) .....	71
Fig. 82	Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3) .....	71
Fig. 83	Selection dialog for new EtherCAT device .....	72
Fig. 84	Display of device revision .....	72
Fig. 85	Display of previous revisions .....	73
Fig. 86	Name/revision of the terminal .....	73
Fig. 87	EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3) .....	74

Fig. 88	Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3).....	75
Fig. 89	Scan Devices (left: TwinCAT 2; right: TwinCAT 3).....	75
Fig. 90	Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3).....	75
Fig. 91	Detected Ethernet devices .....	76
Fig. 92	Example default state .....	76
Fig. 93	Installing EthetCAT terminal with revision -1018 .....	77
Fig. 94	Detection of EtherCAT terminal with revision -1019 .....	77
Fig. 95	Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: Twin- CAT 3) .....	77
Fig. 96	Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3).....	78
Fig. 97	Scan progressexemplary by TwinCAT 2 .....	78
Fig. 98	Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3).....	78
Fig. 99	Displaying of "Free Run" and "Config Mode" toggling right below in the status bar .....	78
Fig. 100	TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: Twin- CAT 3) .....	78
Fig. 101	Online display example .....	79
Fig. 102	Faulty identification .....	79
Fig. 103	Identical configuration (left: TwinCAT 2; right: TwinCAT 3).....	80
Fig. 104	Correction dialog .....	80
Fig. 105	Name/revision of the terminal .....	81
Fig. 106	Correction dialog with modifications .....	82
Fig. 107	Dialog "Change to Compatible Type..." (left: TwinCAT 2; right: TwinCAT 3).....	82
Fig. 108	TwinCAT 2 Dialog Change to Alternative Type .....	82
Fig. 109	Branch element as terminal EL3751 .....	83
Fig. 110	"General" tab.....	83
Fig. 111	„EtherCAT“ tab.....	84
Fig. 112	"Process Data" tab.....	85
Fig. 113	Configuring the process data .....	86
Fig. 114	„Startup“ tab.....	87
Fig. 115	"CoE – Online" tab .....	88
Fig. 116	Dialog "Advanced settings" .....	89
Fig. 117	„Online“ tab .....	89
Fig. 118	"DC" tab (Distributed Clocks).....	90
Fig. 119	Object 0xF800, Master Settings .....	94
Fig. 120	Object 0x8000: slave settings.....	96
Fig. 121	Inserting the EL6652-0000 in TwinCAT 2.1x .....	97
Fig. 122	Insert a device "EtherNet/IP" in "I/O devices" .....	97
Fig. 123	Selection of adapter "EtherNet/IP Scanner (EL6652)" .....	97
Fig. 124	Searching for the EL6652 terminal .....	98
Fig. 125	Configuration of the IP address, network mask and gateway address.....	99
Fig. 126	Add "Generic EtherNet/IP Adapter" .....	99
Fig. 127	Append "Connection Object" .....	99
Fig. 128	Dialog box "Add I/O Connection Object" .....	100
Fig. 129	Appending variables .....	100
Fig. 130	Selection of the task time.....	101
Fig. 131	Diagnosis of "ECatState" and "State" .....	102

Fig. 132	Enter the values in the "Add IO Connection Object" dialog box .....	105
Fig. 133	Output data in the System Manager .....	108
Fig. 134	Configuration "IO Connection Object" .....	109
Fig. 135	Inserting the EL6652-0010 in TwinCAT 2.1x .....	110
Fig. 136	Insert a device "EtherNet/IP" in "I/O devices" .....	110
Fig. 137	Selection of adapter "EtherNet/IP adapter (EL6652-0010)" .....	111
Fig. 138	Searching for the EL6652-0010 terminal .....	111
Fig. 139	Configuration of the IP address, network mask .....	112
Fig. 140	Append "Connection Object" .....	112
Fig. 141	Append variables .....	113
Fig. 142	Configuration of the slave settings.....	113
Fig. 143	Selection of the task time.....	114
Fig. 144	Creating a new controller.....	117
Fig. 145	Appending a new module .....	118
Fig. 146	Selection of the ETHERNET MODULE "Generic Ethernet Module".....	119
Fig. 147	Transferring the parameters into the "New Modules" mask .....	120
Fig. 148	EL6652 LEDs.....	126
Fig. 149	"Diag History" tab.....	128
Fig. 150	Device identifier consisting of name EL3204-0000 and revision -0016 .....	131
Fig. 151	Scan the subordinate field by right-clicking on the EtherCAT device .....	132
Fig. 152	Configuration is identical .....	132
Fig. 153	Change dialog .....	132
Fig. 154	EEPROM Update .....	133
Fig. 155	Selecting the new ESI.....	133
Fig. 156	Display of EL3204 firmware version .....	134
Fig. 157	Firmware Update .....	135
Fig. 158	FPGA firmware version definition .....	137
Fig. 159	Context menu Properties .....	137
Fig. 160	Dialog Advanced Settings .....	138
Fig. 161	Multiple selection and firmware update .....	140
Fig. 162	Selecting the "Restore default parameters" PDO .....	142
Fig. 163	Entering a restore value in the Set Value dialog.....	142