

Documentation | EN

EL6601, EL6614

Ethernet Switchport Terminals

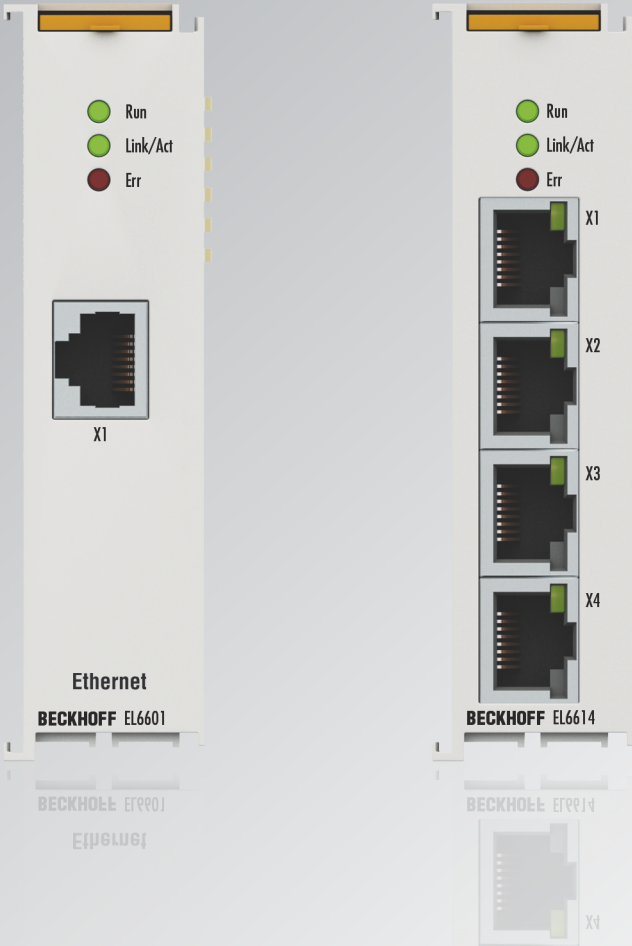


Table of contents

1 Foreword	5
1.1 Product overview Ethernet Switchport Terminals	5
1.2 Notes on the documentation	6
1.3 Guide through documentation	7
1.4 Notes on information security.....	8
1.5 Safety instructions.....	9
1.6 Documentation issue status	10
1.7 Version identification of EtherCAT devices	12
1.7.1 General notes on marking.....	12
1.7.2 Version identification of EL terminals.....	13
1.7.3 Beckhoff Identification Code (BIC).....	14
1.7.4 Electronic access to the BIC (eBIC).....	16
2 Product description	18
2.1 Introduction	18
2.2 Technical data	20
2.3 Basic function principles.....	21
2.4 EL66xx - Non Realtime	24
2.5 EL66xx and Beckhoff network variables	30
2.5.1 Explanation network variables	30
2.5.2 Settings in the System Manager	31
2.5.3 Notes.....	32
2.5.4 Suppress publisher	33
2.5.5 Filter subscribers.....	33
2.5.6 Setting up TwinCAT 2.10	34
2.5.7 Setting up TwinCAT 2.11	36
2.6 Configuration in the CX20x0 & CX50x0 system.....	38
3 Basics communication	41
3.1 EtherCAT basics	41
3.2 EtherCAT cabling – wire-bound	41
3.3 General notes for setting the watchdog	43
3.4 EtherCAT State Machine	44
3.5 CoE Interface	46
3.6 Distributed Clock	51
4 Mounting and wiring	52
4.1 Instructions for ESD protection	52
4.2 Explosion protection.....	53
4.2.1 ATEX - Special conditions (extended temperature range).....	53
4.2.2 IECEx - Special conditions.....	54
4.2.3 Continuative documentation for ATEX and IECEx.....	55
4.2.4 cFMus - Special conditions	56
4.2.5 Continuative documentation for cFMus.....	57
4.3 UL notice	58
4.4 Note - power supply	59

4.5	Mounting and demounting - top front unlocking	60
4.6	Positioning of passive Terminals	62
4.7	Note on shielding	63
4.8	Installation positions	64
4.9	Disposal	66
5	Commissioning	67
5.1	TwinCAT Development Environment	67
5.1.1	Installation of the TwinCAT real-time driver	67
5.1.2	Notes regarding ESI device description	73
5.1.3	OFFLINE configuration creation	77
5.1.4	ONLINE configuration creation	82
5.1.5	EtherCAT subscriber configuration	90
5.2	General Commissioning Instructions for an EtherCAT Slave	99
5.3	Object description and parameterization	107
5.3.1	Objects for commissioning	107
5.3.2	Objects for regular operation	108
5.3.3	Standard objects (0x1000-0x1FFF)	108
5.3.4	Profile-specific objects (0x6000-0xFFFF)	111
5.4	Beckhoff network variables	114
5.4.1	Introduction	114
5.4.2	Configuration of the Publisher	115
5.4.3	Configuration of the Subscriber	118
5.4.4	Beckhoff network variables - Settings	122
6	Application samples	130
6.1	Sample programs	130
6.2	Application sample - network printer	131
6.3	Application sample - Service interface with remote desktop	137
6.4	Application sample - Lower-level control system	147
6.5	Application sample – setting up an EtherCAT Master PC as a network bridge	152
6.6	Application sample - Flexible Ethernet Port	157
7	Appendix	162
7.1	Firmware compatibility	162
7.2	Firmware Update EL/ES/EM/ELM/EP/EPP/ERPxxxx	164
7.2.1	Device description ESI file/XML	165
7.2.2	Firmware explanation	168
7.2.3	Updating controller firmware *.efw	169
7.2.4	FPGA firmware *.rbf	171
7.2.5	Simultaneous updating of several EtherCAT devices	175
7.3	Restoring the delivery state	176
7.4	Support and Service	178

1 Foreword

1.1 Product overview Ethernet Switchport Terminals

[EL6601 \[► 18\]](#)

1-port communication interface, Ethernet switch port

[EL6614 \[► 18\]](#)

4-port communication interface, Ethernet switch port

1.2 Notes on the documentation

Intended audience

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

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

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

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

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

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

1.3 Guide through documentation

NOTICE



Further components of documentation

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

Title	Description
EtherCAT System Documentation (PDF)	<ul style="list-style-type: none"> • System overview • EtherCAT basics • Cable redundancy • Hot Connect • EtherCAT devices configuration
Explosion Protection for Terminal Systems (PDF)	Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx
Control Drawing I/O, CX, CPX (PDF)	Connection diagrams and Ex markings (conform to cFMus)
Infrastructure for EtherCAT/Ethernet (PDF)	Technical recommendations and notes for design, implementation and testing
Software Declarations I/O (PDF)	Open source software declarations for Beckhoff I/O components

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

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

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

1.4 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our <https://www.beckhoff.com/secguide>.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at <https://www.beckhoff.com/secinfo>.

1.5 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

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

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product



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

1.6 Documentation issue status

Version	Comment
4.8	<ul style="list-style-type: none"> • Chapter "Recommended mounting rails" removed • Update chapter "Technical data" • Update revisions status
4.7	<ul style="list-style-type: none"> • Update chapter "Technical data" • Update structure • Update revisions status
4.6	<ul style="list-style-type: none"> • Chapter "Notes on information security" added • Update structure
4.5	<ul style="list-style-type: none"> • Update chapter "Technical data" • Update chapter "Version identification of EtherCAT devices" • Update structure • Update Notes • Update revisions status • Chapter Disposal added
4.4	<ul style="list-style-type: none"> • Update chapter "Application sample" • Update structure
4.3	<ul style="list-style-type: none"> • Update chapter "UL notice" • Update chapter "Technical data" • Update chapter "Firmware compatibilität" • Update structure
4.2	<ul style="list-style-type: none"> • Update chapter "Object description" • Update structure • Revision status updated
4.1	<ul style="list-style-type: none"> • Note in chapter "address assignment" added • Update structure • Revision status updated
4.0	<ul style="list-style-type: none"> • Migration • Update structure • Revision status updated
3.4	<ul style="list-style-type: none"> • Update structure • "Technical data" section updated • Revision status updated
3.3	<ul style="list-style-type: none"> • Update structure • "Technical data" section updated • Chapter "Introduction" updated • Chapter "Configuration on the CX20x0 & CX50x0 System" inserted • Revision status updated
3.2	<ul style="list-style-type: none"> • Update structure • "Technical data" section updated • Chapter "EtherCAT-PC as Network Bridge" updated • Revision status updated
3.1	<ul style="list-style-type: none"> • Update structure • "Technical data" section updated • Update chapter "Mounting and wiring"
3.0	<ul style="list-style-type: none"> • Notes on cable redundancy added

Version	Comment
2.9	<ul style="list-style-type: none">• Notes Subscriber filter, diagnostic data added
2.8	<ul style="list-style-type: none">• Technical notes added
2.7	<ul style="list-style-type: none">• Technical notes added
2.6	<ul style="list-style-type: none">• Technical notes added
2.5	<ul style="list-style-type: none">• Chapter Firmware updated
2.4	<ul style="list-style-type: none">• Technical notes network variables added
2.3	<ul style="list-style-type: none">• Application sample added
2.2	<ul style="list-style-type: none">• Technical notes added
2.1	<ul style="list-style-type: none">• Technical notes (Subscriber, Publisher) added
2.0	<ul style="list-style-type: none">• Technical notes and CoE objects added
1.9	<ul style="list-style-type: none">• Note on installation position added
1.8	<ul style="list-style-type: none">• Technical notes added
1.7	<ul style="list-style-type: none">• Technical notes network variables added
1.6	<ul style="list-style-type: none">• LED and port description added
1.5	<ul style="list-style-type: none">• EL6614 added
1.4	<ul style="list-style-type: none">• Application sample added
1.3	<ul style="list-style-type: none">• Technical data added (object description)
1.2	<ul style="list-style-type: none">• Technical data completed, explanations on mailbox communication and network variables added
1.1	<ul style="list-style-type: none">• Technical data added, UL labelling added
1.0	<ul style="list-style-type: none">• Technical data added, first public issue
0.1	<ul style="list-style-type: none">• Preliminary documentation for EL6601

1.7 Version identification of EtherCAT devices

1.7.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

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

1.7.2 Version identification of EL terminals

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

Structure of the serial number: **KK YY FF HH**

KK - week of production (CW, calendar week)

YY - year of production

FF - firmware version

HH - hardware version

Example with serial number 12 06 3A 02:

12 - production week 12

06 - production year 2006

3A - firmware version 3A

02 - hardware version 02



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

1.7.3 Beckhoff Identification Code (BIC)

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

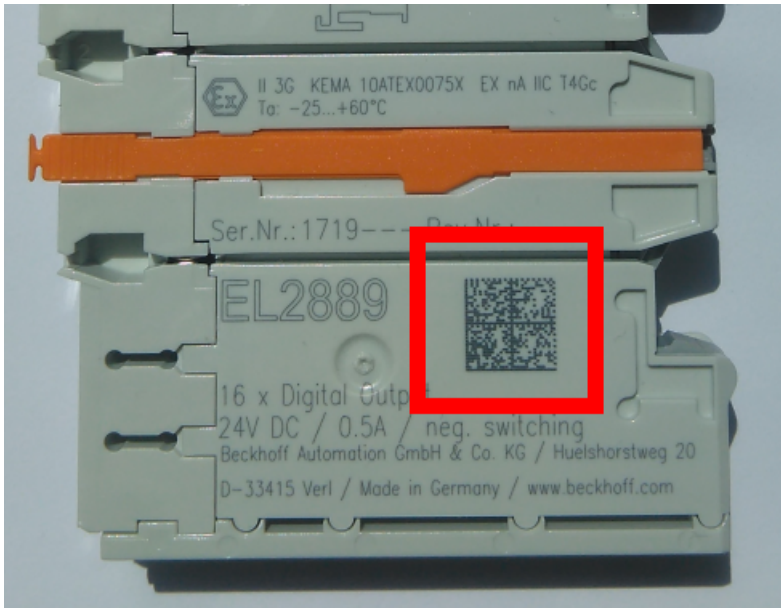


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

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

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

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

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

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

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

Position	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	Beckhoff order number	1P	8	1P 072222
2	Beckhoff Traceability Number (BTN)	Unique serial number, see note below	SBTN	12	SBTN k4p562d7
3	Article description	Beckhoff article description, e.g. EL1008	1K	32	1K EL1809
4	Quantity	Quantity in packaging unit, e.g. 1, 10, etc.	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P 401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S 678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30P F971, 2*K183
...					

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

Structure of the BIC

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

1P072222**SBTN**k4p562d7**1K**EL1809 **Q1** **51S**678294

Accordingly as DMC:



Fig. 3: Example DMC **1P**072222**SBTN**k4p562d7**1K**EL1809 **Q1** **51S**678294

BTN

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

NOTICE

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

1.7.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

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

K-bus devices (IP20, IP67)

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

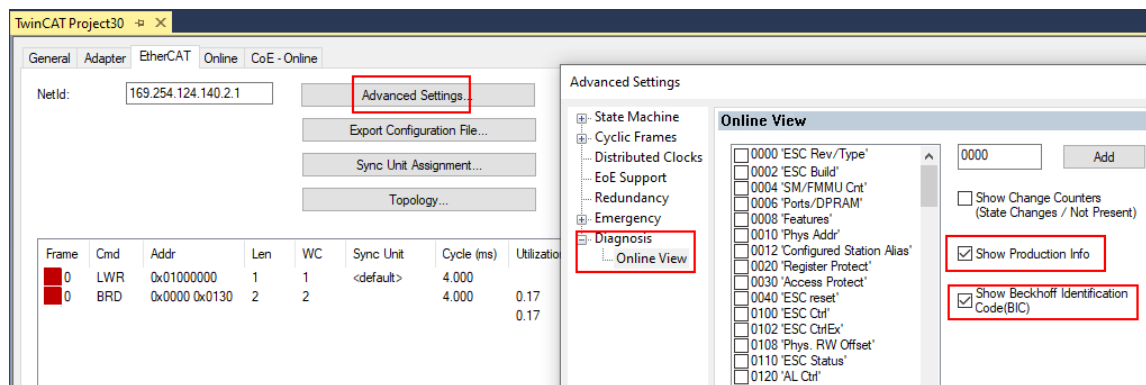
EtherCAT devices (IP20, IP67)

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

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

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

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



- The BTN and its contents are then displayed:

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

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

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

Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	SubIndex 001	RO	1P158442SBTN0008jckp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

- The object 0x10E2 will be preferentially introduced into stock products in the course of necessary firmware revision.
- From TwinCAT 3.1. build 4024.24, the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC are available in the *Tc2_EtherCAT* library from v3.3.19.0
- The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2_Uilities* as of TwinCAT 3.1 build 4024.24
 - *F_SplitBIC*: The function splits the Beckhoff Identification Code (BIC) *sBICValue* into its components using known identifiers and returns the recognized substrings in the *ST_SplittedBIC* structure as a return value
 - *BIC_TO_BTN*: The function extracts the BTN from the BIC and returns it as a return value
- Note: If there is further electronic processing, the BTN is to be handled as a *string(8)*; the identifier "SBTN" is not part of the BTN.
- Technical background
 The new BIC information is written as an additional category in the ESI-EEPROM during device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored using a category in accordance with the ETG.2010. ID 03 tells all EtherCAT masters that they may not overwrite these data in the event of an update or restore the data after an ESI update.
 The structure follows the content of the BIC, see here. The EEPROM therefore requires approx. 50..200 bytes of memory.
- Special cases
 - If multiple hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC information.
 - If multiple non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC information.
 - If the device consists of several sub-devices which each have their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

PROFIBUS; PROFINET, and DeviceNet devices

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

2 Product description

2.1 Introduction

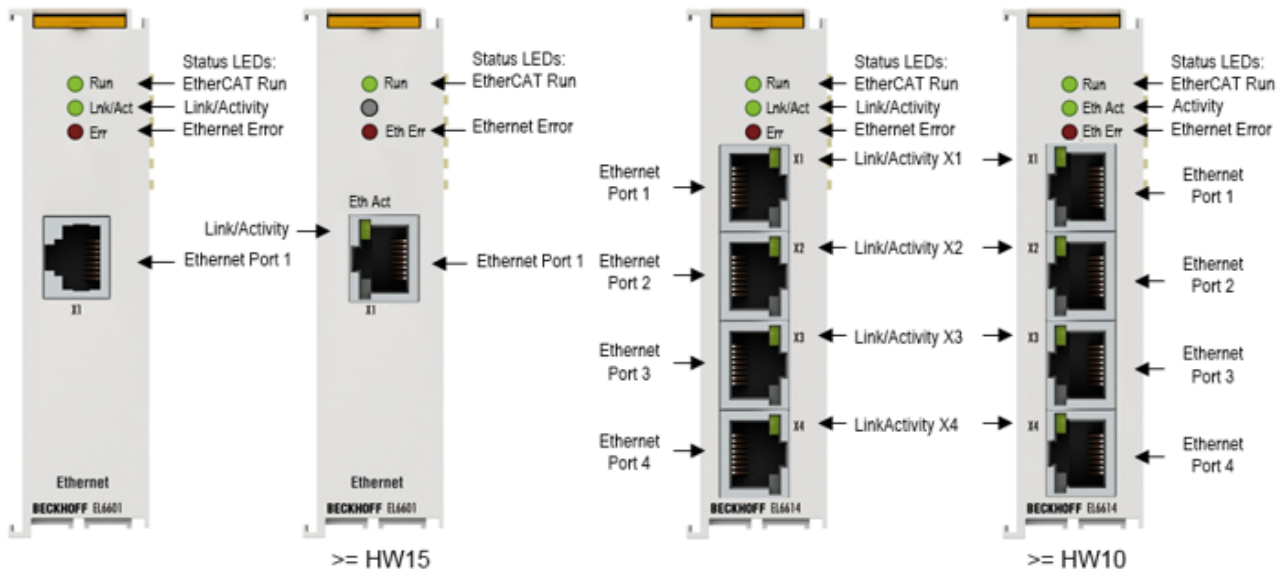


Fig. 4: EL6601, EL6614

Switch terminals for Ethernet

The switch terminals for Ethernet are used for decentralized connection of random Ethernet devices to the EtherCAT terminal network. The EtherCAT system relays the Ethernet communication of the connected devices fully transparent and collision-free.

The 4 port Ethernet switch terminal EL6614 purposefully forwards the incoming frames from the ports to the destination ports. In full duplex mode, it thus enables collision-free communication of the connected devices with each other.

Any number of EL6601/EL6614 can be used simultaneously and at any position in the EtherCAT terminal network. No configuration is required. In conjunction with the network port at the EtherCAT master the EL6601/EL6614 devices operate like a virtual switch whose ports are distributed in the field. The EtherCAT fieldbus is the backbone of this switch.

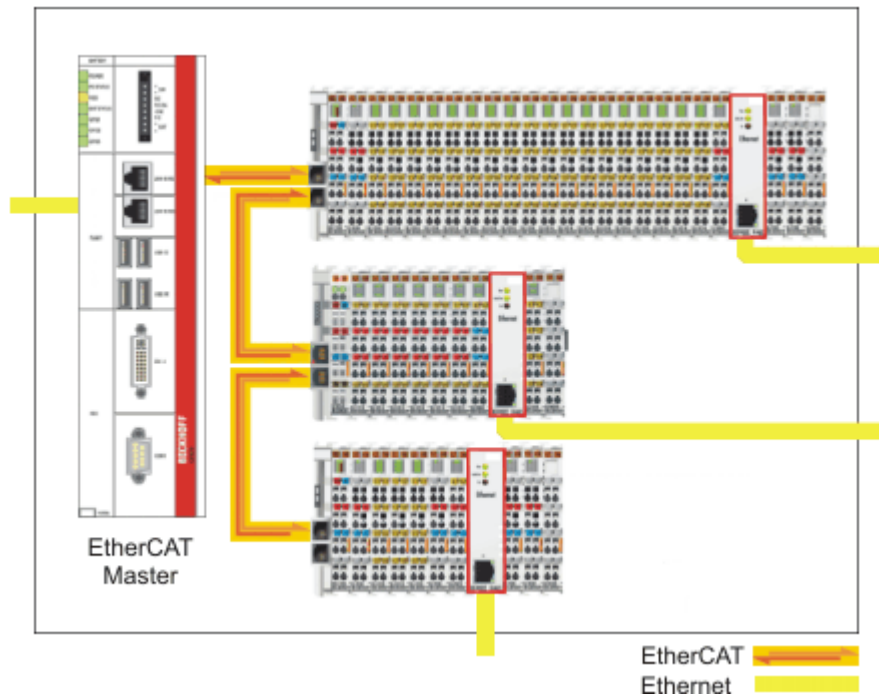


Fig. 5: EL6601 as a virtual, field-distributed switch

Further benefits underline the particular suitability for the application in industrial environments:

- Compact design in EtherCAT terminal housing
- 10/100 MBaud, half or full duplex, with automatic baud rate detection
- Autocrossing (automatic detection of crossed lines)

LEDs

LED	Color	Meaning	
RUN	green	These LEDs indicate the terminal's operating state:	
		off	State of the <u>EtherCAT State Machine</u> [▶ 90]: INIT = initialization of the terminal or BOOTSTRAP = function for <u>firmware updates</u> [▶ 164] of the terminal
		flashing	State of the EtherCAT State Machine: PREOP = function for mailbox communication and different standard-settings set
		single flash	State of the EtherCAT State Machine: SAFEOP = verification of the <u>Sync Manager</u> [▶ 91] channels and the distributed clocks. Outputs remain in safe state
on	State of the EtherCAT State Machine: OP = normal operating state; mailbox and process data communication is possible		
Link/Act	green	Connection / data exchange field bus	
*Link/Act X1 - X4	green	Connection / data exchange Ethernet port X1- X4	
Eth Err	red	Error message EtherCAT (see <u>Diagnostics</u> [▶ 22])	

* only EL6614

Connections

- 1 x RJ45 with 10BASE-T/100BASE-TX Ethernet (EL6601)
- 4 x RJ45 with 10BASE-T/100BASE-TX Ethernet (EL6614)

2.2 Technical data

Technical data	EL6601	EL6614
Bus system	all Ethernet (IEEE 802.3) based protocols	
Number of Ethernet ports	1	4
Ethernet interface	10BASE-T/100BASE-TX Ethernet with 1 x RJ45	10BASE-T/100BASE-TX Ethernet with 4 x RJ45
Cable length	up to 100 m twisted pair	
Data transfer rate	10/100 Mbit/s, IEEE 802.3u Auto negotiation, half or full duplex at 10 and 100 Mbit/s possible, automatic settings	
Network variables	EL6601 as of Firmware 07, EL6614 as of Firmware 03: max 32 Publishers with total of max. 1024 bytes <u>total data</u> [▶ 30] max 32 Subscriber with total of max. 1024 bytes <u>total data</u> [▶ 30]	
Distributed Clocks	no	
Diagnostics	Status-LEDs, CoE data about ADS	
Power supply	via the E-bus	
Current consumption via E-bus	typ. 260 mA	typ. 310 mA
Electrical isolation	500 V (E-Bus/Ethernet)	
Bit width in process image	-	
Configuration	TwinCAT System Manager/EtherCAT Master	
Weight	approx. 75 g	approx. 85 g
Permissible ambient temperature range during operation	-25°C ... +60°C (extended temperature range)	Horizontal installation position: -25°C ... +60°C (extended temperature range) <i>all other installation positions:</i> -25°C ... +45°C, see note [▶ 64]
Permissible ambient temperature range during storage	-40°C ... +85°C	-40°C ... +85°C
Permissible relative humidity	95 %, no condensation	
Dimensions (W x H x D)	approx. 26 mm x 100 mm x 52 mm (width aligned: 23 mm)	
Mounting	on 35 mm mounting rail conforms to EN 60715	
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27	
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4	
Protection class	IP20	
Installation position	variable	see note [▶ 64]
Approvals / markings*	CE, cULus [▶ 58], UKCA, EAC, ATEX [▶ 53], IECEx [▶ 54], cFMus [▶ 56]	CE, cULus [▶ 58], UKCA, EAC, ATEX [▶ 53], IECEx [▶ 54]

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

Ex markings

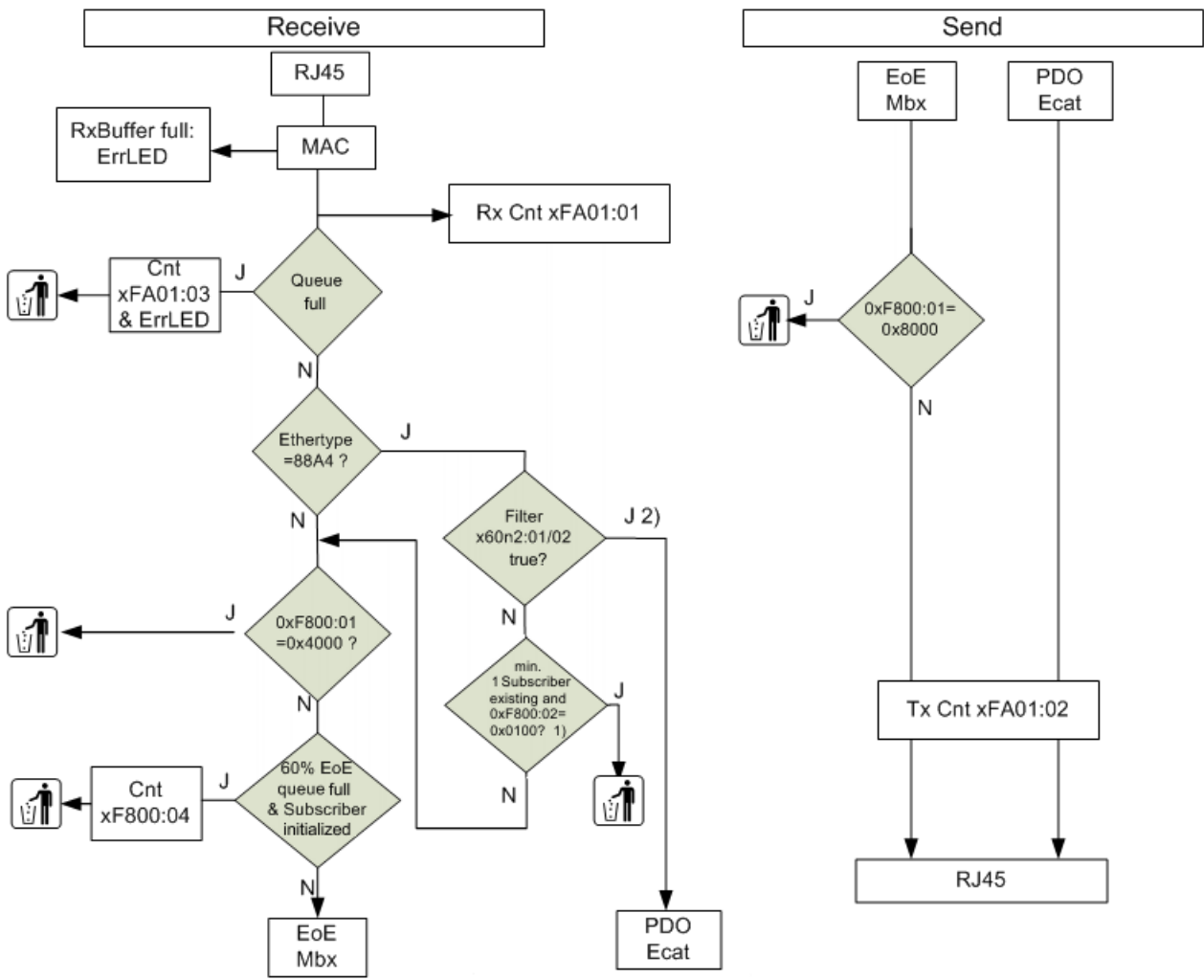
Standard	Marking
ATEX	II 3 G Ex nA IIC T4 Gc
IECEX	Ex nA IIC T4 Gc
cFMus	Class I, Division 2, Groups A, B, C, D Class I, Zone 2, AEx/Ex ec IIC T4 Gc

2.3 Basic function principles

The EL66xx Ethernet Switchport terminals have 2 different operating modes, ideal for the tasks required for Ethernet connectivity. The two operating modes, which can be active simultaneously, provide both the real-time-critical transmission and reception of configured network variables as well as the transport of standard Ethernet traffic, which, while it is not real-time-critical, does involve large data flows using, for instance, the IP protocol:

- **Real-time data exchange:** Publisher/subscriber, Beckhoff network variables, EtherCAT Automation Protocol
The TwinCAT configuration file *.tsm configures an EL66xx when EtherCAT starts up with CoE parameters in such a way that it
 - transmits, as the publisher, data delivered through the cyclical data transfer in the real-time cycle.
 - transmits subscribers received in the same way to the EtherCAT Master over the cyclical EtherCAT data exchange.
Cyclical data exchange with the EL66xx is configured in the PDO settings of the EL66xx when EtherCAT starts up, and cannot be changed online.
- **Non-real-time data exchange**
In parallel with this, the EL66xx can transfer Ethernet frames through the acyclic mailbox exchange (EoE = Ethernet over EtherCAT) between the terminal and the EtherCAT Master/TwinCAT. This data exchange is optimized for throughput, and may involve automatic fragmentation - by default, all telegrams that are not transferred in the PDO context are transported through the acyclic channel by means of EoE.

The flow of data in the EL66xx can be represented schematically as follows:



1) From EL6614 FW06 (bug in FW05, here 0xF800:03)
 2) There are only the number of bytes copied into the PDO, as configured in the PDO. Therefore it is possible, that incomplete data come to the controller, if the data volume is higher. In the CoE x60nn (for each Subscriber), the individual components of the Ignore Item (x60nn) can be switched on the Subscriber Filter: x60n0: filter values, x60n2: corresponding filter value activated, default: VariableID is activated.

Fig. 6: EL66xx data diagram

The EL6601/EL6614 cannot transport an *EtherNet Industrial Protocol (EtherNet/IP)*.

Diagnostics

Online diagnostics

The following objects are available for initial diagnostic in the CoE directory:

- 0xFA01, subindex 01: Frame Counter Rx (incoming to RJ45 socket).
- 0xFA01, subindex 02: Frame Counter Tx (outgoing from RJ45 socket).

The values can be read from the controller using PLC function blocks (FB_EcCoeSdoRead in TcEtherCAT.lib).

This and further diagnostic information from the CoE of the EL66xx are accessible via https://infosys.beckhoff.com/content/1033/el6601_el6614/Resources/2349552907.zip

Error LED

The red Error LED lights up 250 ms in the event of

- Ethernet Receive Overrun --> in general, more Ethernet frames are received at the RJ45 connection than can be transported away via EtherCAT (PDO or mailbox). The telegrams are discarded.
- Ethernet EoE Overrun --> more non-real-time frames are being received at the RJ45 connector than can be transported away by EtherCAT/EoE. The data are discarded.
- Ethernet Frame Error

If the occurrence of an overrun causes data to be lost, higher protocol layers in an Ethernet network are responsible for repeating the transmission.

i Overruns

The following measures can be used to counter overruns:

- activating the Subscriber Filter [▶ 30] in the EL66xx concerned
- Increasing/decelerating the cycle time of the publisher
- Suppressing temporarily publisher transmission or modulo in the System Manager
- Reducing/accelerating the EtherCAT cycle time of the subscriber, so that more data are fetched by the EL66xx

Cable redundancy

If the EL66xx is operated in a system with cable redundancy, please keep the following in mind:

- real-time operation with network variables is possible
- in the event of non-real-time operation with IP transfer the IP traffic is routed via the primary EtherCAT port. Therefore the Windows IP settings of this port are also used.

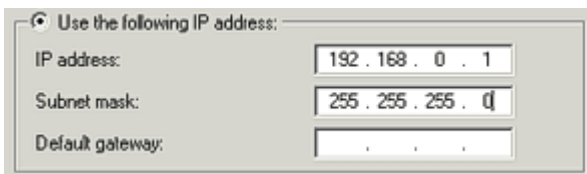


Fig. 7: IP settings EtherCAT port

If there is no longer a link to this port, from Windows under TwinCAT 2 or 3 there is also no IP communication to this port currently. For this reason, do not let the Ethernet connection between the primary EtherCAT port and the first EtherCAT slave fail, since otherwise IP communication is no longer possible via the EL66xx.

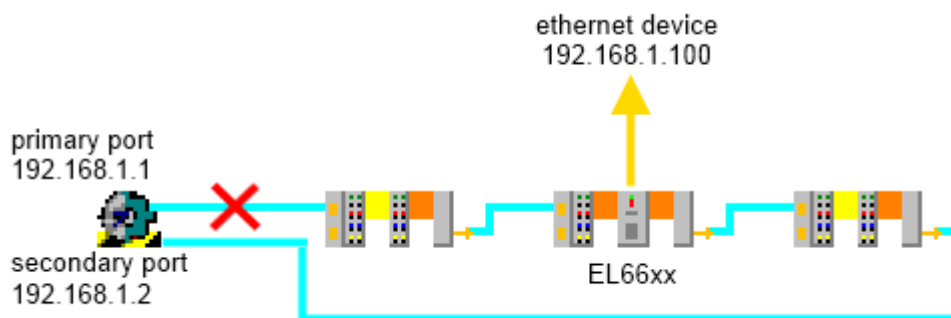


Fig. 8: Connection failure between primary EtherCAT port and 1st slave (X)

2.4 EL66xx - Non Realtime

EL66xx and Ethernet transport via mailbox communication

In addition to regular cyclical process data exchange an EtherCAT master offers a further mechanism for transporting data to an EtherCAT slave or reading data from it. This mechanism is used for one-time or dynamically alternating Data Exchange, such as e.g. the parameterization of an EtherCAT slave. Mailbox communication can also be used for transporting large data blocks acyclically on request from master or slave. This additional communication takes place between the cyclical process data frames (the conventional EtherCAT frames) on the EtherCAT bus.

● Data throughput in mailbox communication

i Since mailbox communication can only take place between the regular process data frames, data throughput with this communication method depends on the load of the EtherCAT bus. This means that the Ethernet throughput of the EL6601 also depends on the load of the underlying EtherCAT fieldbus.

The EoE method (Ethernet over EtherCAT) is used for the EL66xx. Dedicated settings are available for this in the System Manager.

Data throughput

The data throughput of the EL66xx in Ethernet frames or bytes/second depends on

- The EtherCAT cycle time on the fieldbus: The shorter the EtherCAT cycle used for the process data, the more acyclical mailbox queries can be completed. If several different EtherCAT cycle times are used in an EtherCAT strand the fastest cycle time is the relevant time
- The time between the process data frames that is available for mailbox communication: The longer the Ethernet line is free for acyclical mailbox communication, the higher the Ethernet data throughput of the EL6601.
- The mailbox size [▶ 26] in bytes: The larger the mailbox, the more Ethernet frames the EL6601 can send to the EtherCAT master or received from it simultaneously.
- The number of terminals in the EtherCAT system that use mailbox communication at the same time.
- The EoE settings [▶ 28] in the TwinCAT System Manager, see the EoE section.

The following values were determined as samples (TwinCAT 2.10, 2.11)

- > 5 Mbit/s from the EL6601 to the Ethernet device
- > 2 Mbit/s from the Ethernet device to the EL6601

with an EtherCAT cycle time of 100 µs and a mailbox size of 1024 bytes.

● Tips for shortening the response times

i We recommend the following procedure to shorten the response times in your application (e.g. to ping requests): Significantly lower the EtherCAT cycle time currently being used or insert a new task with a lower cycle time, e.g.: 500 µs if up to this point you have been using 2.5 ms EtherCAT cycle. Important: This task must access genuine IO process data from the EtherCAT slaves and be recognizable under Device EtherCAT -> Tab EtherCAT, see Fig. *Real frame structure from the TwinCAT System Manager*

Fra...	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (...)	Size / Duration ...
0	LRD	0x00010000	1	1	<default>	0.500		
0	LRD	0x00080000	1	1		0.500	1.34	42 / 6.72
1	LRD	0x00020000	3	12	<default>	2.500		
1	BRD	0x0000 0x0130	2	15		2.500	0.27	45 / 6.72




Fig. 9: Real frame structure from the TwinCAT System Manager

Note regarding the specified values

i These values are typical values without warranty. Throughput rates may differ in different applications depending on boundary conditions.

Address assignment

From FW03 onwards, the EL6601/6614 can also assign IP addresses to connected devices and works as a DHCP or BOOTP server for *one* device. The following settings are required in the System Manager (EL66xx --> Advanced Settings --> Mailbox --> EoE):

- Setting "Switch Port", Fig. *Default setting of the EL66xx as switch port without IP address assignment*. The EL66xx works like a normal switch and passes Ethernet frames transparently through to TwinCAT/Windows
- Setting for "IP Port", Fig. *From FW03: Settings for dynamically assigned IP address*. The EL66xx works with address assignment to *one* connected Ethernet device. A DHCP or a BootP Client must be activated in the device (refer to the network adaptor settings in the operating system). The EL66xx responds to the device's corresponding DHCP/BootP query by assigning the specified IP address/subnet mask to the device. In the DHCP method this address is regularly queried by the client and assigned to the server/EL66xx.

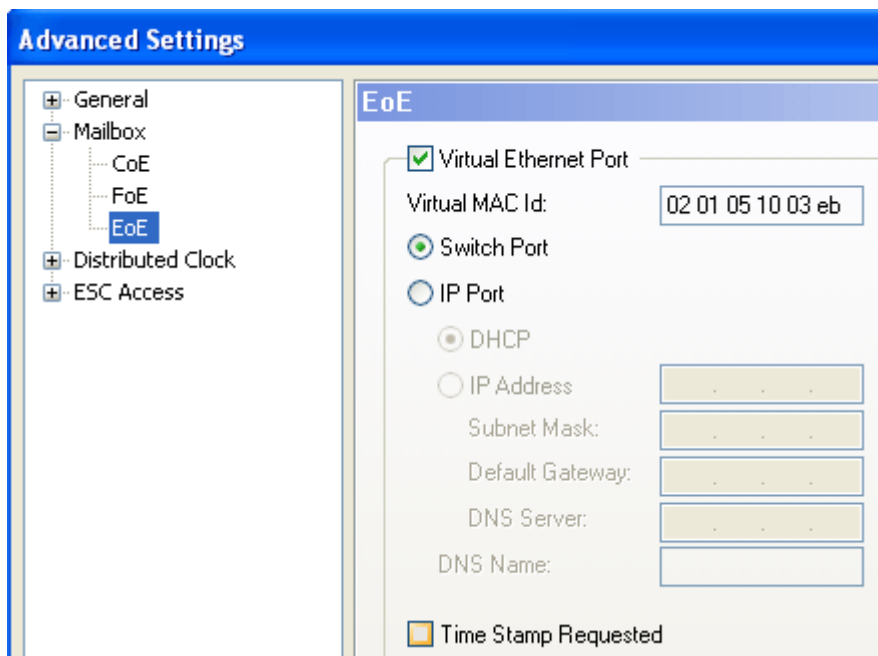


Fig. 10: Default setting of the EL66xx as switch port without IP address assignment

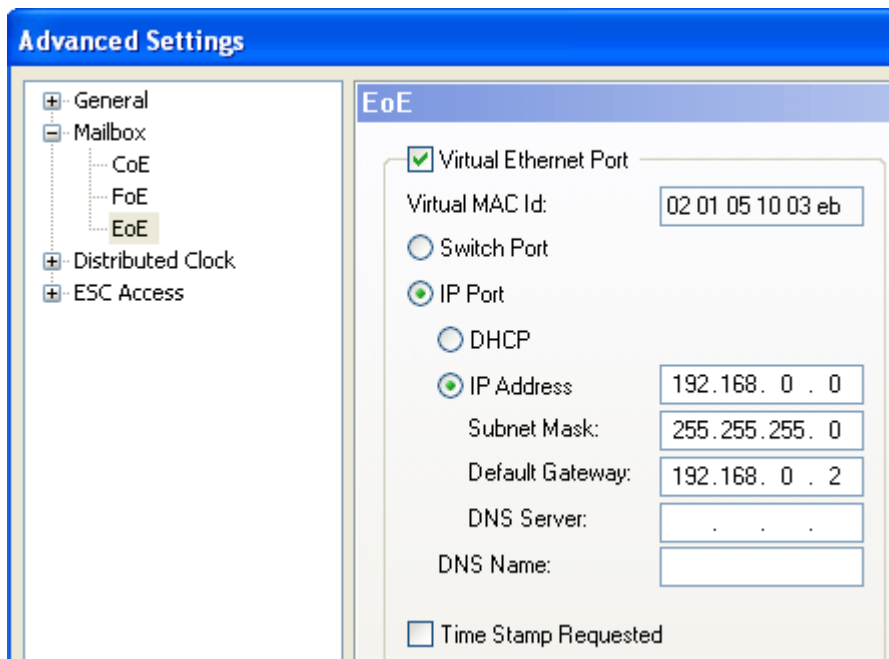


Fig. 11: From FW03: Settings for dynamically assigned IP address

Please note:

- The “DHCP” checkbox must not be used - the “IP address” checkbox activates the DHCP/BootP function in the EL66xx.
- The Gateway, Mask and Server settings are likewise communicated to the client/the device
- Only *one* address can be assigned, i.e. no switch with connected subscribers may follow.
- the address range must be identical to that of the EtherCAT adapter.
- DHCP Server Identifier: several DHCP Servers need a ServerID in the response telegram. Solution for the EL6601 from firmware 15: the value 0x1000 has to be entered in the object 0xF800:01. If a Default Gateway is registered in the EL6601, it is used as a DHCP Server Identifier.

Mailbox settings

The mailbox size can be modified in the Beckhoff TwinCAT System Manager:

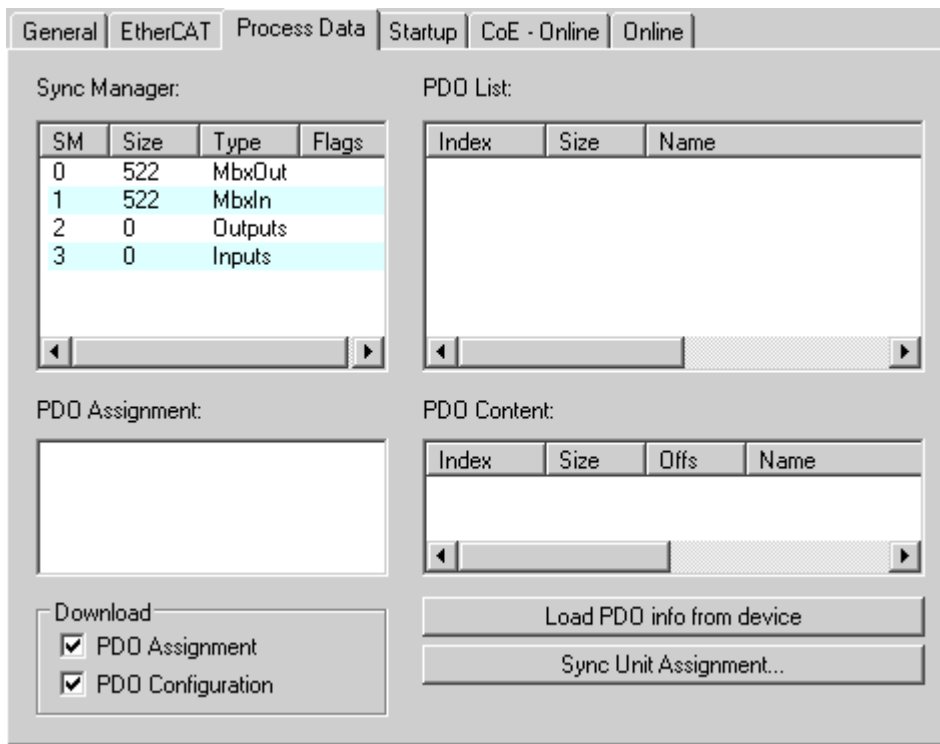


Fig. 12: Default mailbox settings

By default the mailbox is set to 522 Byte Input and 522 Byte Output (20 A_{hex}), see Fig. *Default Settings of the Mailbox*, Entries for SyncManager 0 and 1. To increase the data throughput the size of the mailbox can be increased to 1024 Byte, see Fig. *Increasing the Size of the Mailbox*.

Default mailbox size

i As of Revision EL66xx-0000-0018 the mailbox is already set to 1024 Byte by default in both directions, therefore it cannot be further enlarged. The previous statements apply for terminals with Revision -0000, -0016 or -0017.

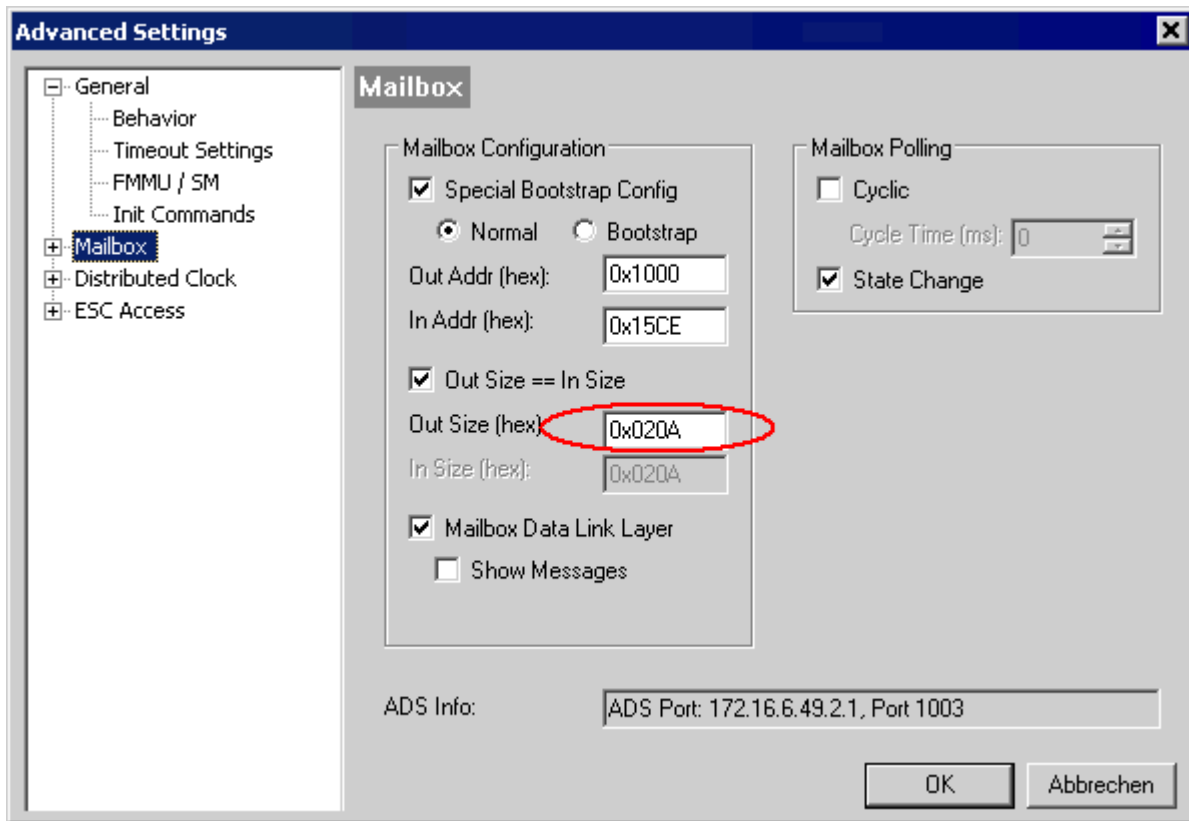


Fig. 13: Increasing the mailbox

Under EL6601 -> EtherCAT tab -> "Advanced Settings..." -> "Mailbox" the "Out Size" can be set to hexadecimal values between $42_{dec}/2A_{hex}$ and $1024_{dec}/400_{hex}$ bytes. Ethernet frames that are larger than the EL6601 mailbox are fragmented by the EL6601 or the EtherCAT master and reassembled after passing through the EtherCAT system.

Virtual switch setting

The EL66xx devices in the TwinCAT system generally appear as virtual switches, with the EtherCAT system as the "backbone".

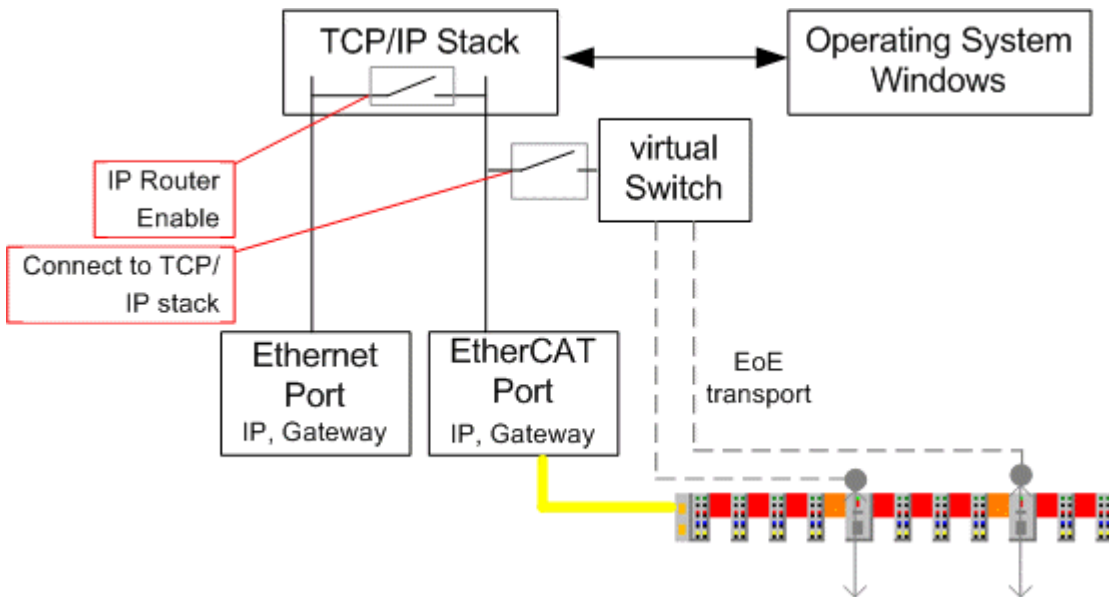


Fig. 14: TwinCAT 2.11, virtual TwinCAT switch

The required settings will be found under TwinCAT | EtherCAT device | Advanced settings

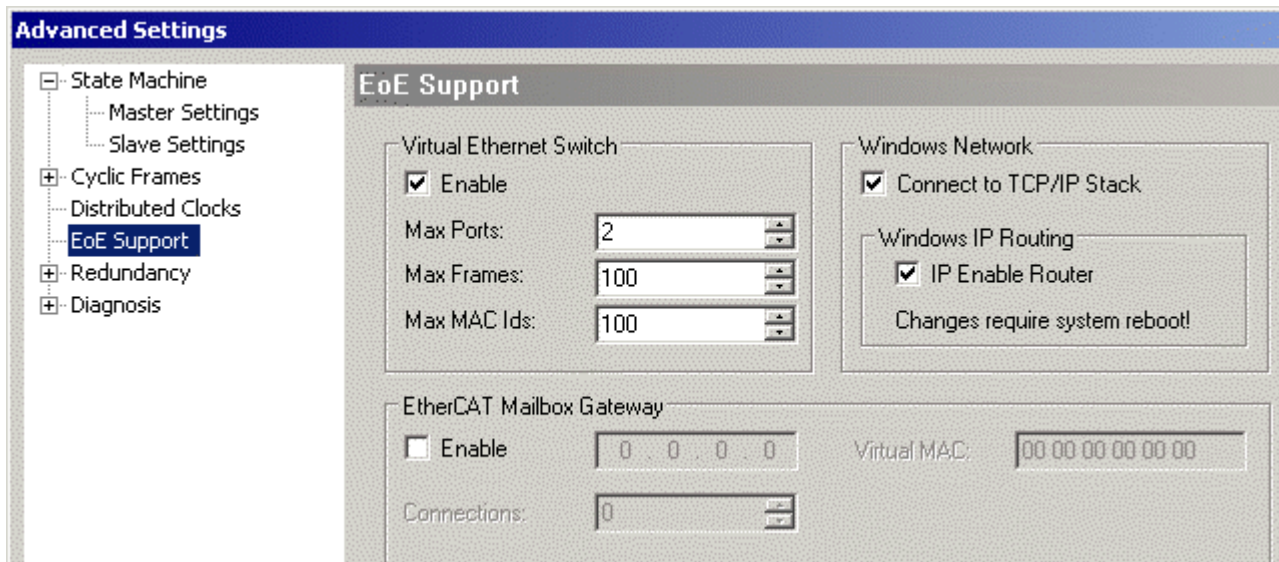


Fig. 15: TwinCAT 2.11, virtual TwinCAT switch

Notes

- If a large number of EL66xx devices are used along the EtherCAT strand it may be helpful to increase the value of *MaxFrames*
- If the EL66xx is used exclusively to transfer network variables, *ConnectToTcpStack* should be deactivated
- IP-routing is active by default. This can also be checked by entering "ipconfig /all" on the command line (Windows)

2.5 EL66xx and Beckhoff network variables

2.5.1 Explanation network variables

Network variables

The EL66xx support sending/receiving network variables. This applies for the EL6601 as of Firmware 07, for the EL6614 as of Firmware 03.

A maximum of 32 for each, publishers and subscriber, are permitted per EL66xx.

● Hardware replacement

i If the system was designed with a previous EL6601 version (EL6601-0000-0000), this can be replaced with versions from EL6601-0000-0017 without problem. If the system was designed for version EL6601-0000-0017 or higher, replacement with a previous version is not possible due to unsupported network variables.

Network variables are specially configured Ethernet frames that enable Beckhoff devices to communicate with each other in real-time via Ethernet. Such device can send (publisher) or receive (subscriber) messages.

An Ethernet frame is sent for each publisher (Ethernet-based). A maximum of 1500 bytes of data can thus be sent per publisher. Within a publisher/subscriber several variables (publisher and subscriber variables) can be created.

Generally, several publishers/subscribers can be configured for each sending/receiving device (e.g. IPC or EL6601).

Based on the sample of a data sender the hierarchy therefore consists of

- the sending device with a minimum of one Ethernet interface: IPC, CX, FC9011, EL6601, ...
 - FastEthernet/100 MBit and 1 GBit are supported
 - This Ethernet interface is configured in the local TwinCAT System Manager as a real-time Ethernet device
- 1..n configured publishers - each publisher is sent as an independent Ethernet frame and can therefore contain a maximum of 1500 bytes
- 1..n publisher variables contained therein for linking with the task/PLC
 - For each publisher variable the user data and [diagnostic data \[► 30\]](#) are transferred

On the receiver side the configuration is mirrored.

The EL66xx can also process publishers and subscribers which are frame data

- Max. 32 publishers and/or subscribers
- For each transmit direction (publisher or subscriber) the following maxima apply:
 - all publishers: 1024 bytes [total data \[► 30\]](#)
 - all subscribers: 1024 bytes [total data \[► 30\]](#)


● Update of the terminal

i The values above apply for a EL6601/6614-0000-0018. Version -0017 only supports a maximum of 300 bytes per publisher/subscriber. If a -0017 terminal is used, the values specified above can be achieved by an update to revision -0018. Please contact our technical support.

With appropriate EtherCAT cycle time and depending on the scale and number of the publishers/subscribers configured in the EL66xx, real-time cycle times down to 500 µs or below are possible.

Typical throughput values for EL6601, FW08, Rev. EL6601-0000-0018 are

- 1 publisher with 1000 bytes, 1 subscriber with 1000 bytes, simultaneous bidirectional operation: 2 ms
- 1 publisher with 100 bytes, 1 subscriber with 100 bytes, simultaneous bidirectional operation: 300 µs

Both characteristic values were determined with this sample program . TwinCAT from version 2.11 is required for the *.tsm System Manager file.

The EL6601 is used as a sample to explain configuration as publisher or subscriber for network variables. The dialogs under TwinCAT 2.10 and TwinCAT 2.11 here are slightly different.

The following descriptions of the dialogs of the EL6601 in the TwinCAT System Manager can be applied equally to the EL6614.

Note regarding the term total data

For each data direction the EL6601/EL6614 from Rev. -0018 can transfer a maximum of 1024 byte total data. The total data consist of the user data (e.g. a UDINT) and the diagnostic data for the EL66xx.

Formula for number of diagnostic data bytes

- Publisher direction: 2 + ((number of publishers) * 2)
- Subscriber direction: 2 + ((number of subscriber variables) * 4)

If the configured data quantity exceeds 1024 bytes, a corresponding message window appears when activation is attempted:

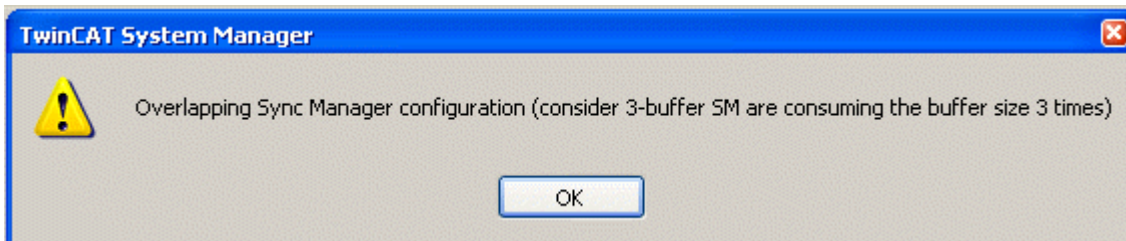


Fig. 16: Notice on exceeding configured data volume

Note regarding the data quantity

The EL66xx (EL6601 from FW07, EL6614 from FW03) has an 8 kbyte data memory with the following default allocation

Type	Usable extent	Operation mode	Allocated memory
Mailbox Out	1024 bytes		1024 bytes (fixed)
Mailbox in	1024 bytes		1024 bytes (fixed)
Publisher	1024 bytes	3-buffer mode	3072 bytes
Subscriber	1024 bytes	3-buffer mode	3072 bytes

If more publisher or subscriber data are required for an application, the SyncManagers can be modified accordingly. The mailbox cannot be modified.

Documents about this

 https://infosys.beckhoff.com/content/1033/el6601_el6614/Resources/17350688139.zip

2.5.2 Settings in the System Manager

● Appearance of the variables

i Depending on the platform used (PC or EL66xx), the publisher/subscriber will appear differently. A publisher/subscriber can be created:

- on a PC network interface, see [Beckhoff network variables - Settings \[▶ 122\]](#)
- on an EL66xx

The following sample illustrates the setup for a publisher and a subscriber variable (each with a size of a 16-bit word) on an EL6601 under TwinCAT 2.10.

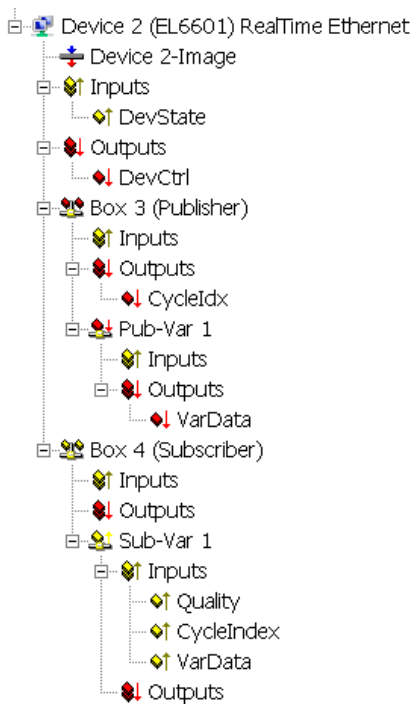


Fig. 17: Network variable sample configuration on an EL6601

Process data:

- "CycleIdx": must be served by the application in order to be evaluated on the subscriber side
- "CycleIndex": CycleIdx counterpart on the subscriber side.
- "VarData": the data to be sent.

2.5.3 Notes

- The RT statistics displays are not supported under TwinCAT for an EL66xx-RT device.
Solution: As an alternative, corresponding CoE parameters can be read for diagnostic purposes.
- The publisher features of "OnChangeOnly" and "DataExchange (divider/modulo)" are not supported together with the EL66xx.
Solution: [from FW08] Transmitting the configured publisher variables can be cyclically suppressed by DevCtrl.
- If a publisher is set up on an EL66xx, the publisher's CycleIndex [► 123] must be taken care of by the user. On a PC, on the other hand, they are incremented by TwinCAT.
- The following is recommended for diagnosis of a network variable connection:
 1. Monitor the link status in the "DevState" of the RT device (Device --> Inputs --> DevState). The expected state is *DevState* = 0.
 2. Monitor the *Quality* and *CycleIndex* in the subscriber.
- The link LED in the EL66xx only indicates the status of the cable connection, not that of any network variable connection that may exist.
- If the EL66xx is used exclusively to transfer network variables, ConnectToTcpStack [► 29] should be deactivated.
- A maximum of 32 for each, publishers and subscriber, are permitted per EL66xx.

2.5.4 Suppress publisher

Applicable: TwinCAT from version 2.11, EL6601 from FW08, EL6614 from FW04

If the EL66xx is operated with a short cycle time and with publishers configured, this can place a high loading on the connected network. For this reason, the EL66xx can be configured in such a way that the transmission of individual publishers can be blocked through the *DevCtrl* variable. The object 0xF800:02 must be occupied in the CoE (CanOpenOverEtherCAT) for this purpose.

Groups of publisher boxes can be blocked by setting appropriate bits (publisher frames). The topmost 4 bits (the high nibble of high byte) from 0xF800:02 specify the granularity of the groups 1..15, i.e. how many publisher frames are handled together as one group:

The upper 8 bits of *DevCtrl* (format: 16 bits) then block the transmission of the publisher frames located in the corresponding group in the current cycle.

High byte of *DevCtrl* :

- 0 = no blocking
- n = each bit in *DevCtrl* corresponds to a group of n publishers, where n has a value in the range [1..31]

It follows that a maximum of 8 groups of publishers can be blocked.

Sample:

DevCtrl.10 = true and 0xF800:02 = 0x2000 signifies that the third group will be blocked in this PLC cycle. One group consists of 2 publisher frames, which means that in this case all the publisher variables that are located in publisher frames 5 and 6 will not be transmitted.

NOTICE

Suppressing individual publishers

The structure of a "publisher" as a publisher box in the System Manager is

- an Ethernet frame containing
- n publishers

The individual bits in DevCtrl each block a group of publisher frames.

The success achieved in this way can be observed using, for instance, a network monitor such as Wireshark.

● Changes in the CoE

i The CoE contents can, if writable, be changed online using the TwinCAT System Manager. However, after the terminal or the EtherCAT system is restarted, this change will no longer be present; default values will apply. As a result, any permanent change must be stored in the terminal's CoE startup list.

Note: In this documentation, bit counting starts from 0: value.0, value.1, ...

2.5.5 Filter subscribers

Applicable: TwinCAT from version 2.11, EL6601 from FW08, EL6614 from FW06

Depending on how the Ethernet network is configured, large or small numbers of the publisher telegrams being used there arrive at the EL66xx devices included in the network. At the start, the EL 66xx is configured by the EtherCAT Master to the subscriber variables that it is to receive: source AMS Net ID and ID of the variables are loaded into the CoE for each subscriber. The CoE objects 0x60n0:01 and 0x60n0:02 then respectively contain the AmsNetId and Variables ID to be checked. The EL66xx devices can therefore filter according to the incoming publisher IDs, and compare them with their own subscriber IDs. For this purpose the publisher variables contained in the Ethernet frames received are disassembled and checked individually.

If an incoming subscriber

- corresponds to a configured AMS Net ID and Variables ID, then the contents are transferred to EtherCAT via PDO.
- does NOT correspond to the above, then the contents are transferred as standard to the acyclic mailbox interface for transmission to the Master.

This is the standard setting of the EL66xx.

The second way generates a high acyclic EtherCAT transport load, because subscribers received by the EL66xx are transported that should not be transported by this EL66xx at all. For this reason the subscriber filter can be activated by the **CoE entry 0xF800:02 = 0x0100 (bit 8 = TRUE)**. The subscriber data that do not correspond to the AmsNetID/Variables ID filter are then discarded in the terminal and are not transferred to the mailbox.

● Filter subscribers



Activation of the subscriber filter is recommended.

Since the EL66xx needs to be re-initialized with each INIT-OP transition, it is essential to set the named CoE entry in the startup list.

Note: In this documentation, bit counting starts from 0: value.0, value.1, ...

2.5.6 Setting up TwinCAT 2.10

Once the EtherCAT bus and its devices have been configured, the EL6601 is appended as a separate device in the configuration tree.

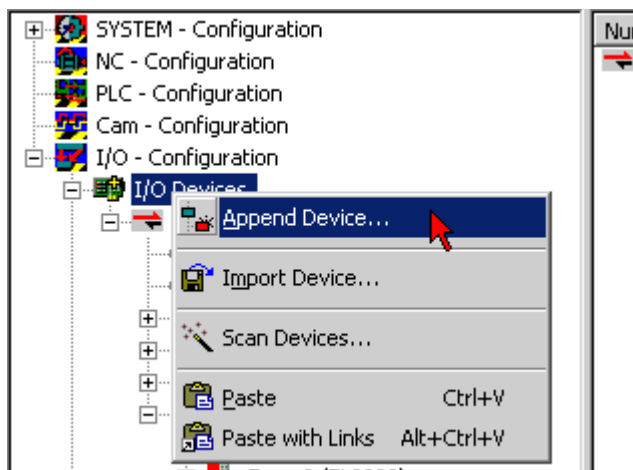


Fig. 18: Append device

In the selection dialog an EL6601 is offered as a real-time Ethernet device. The EL6601 must also be selected here when an EL6614 is being used.

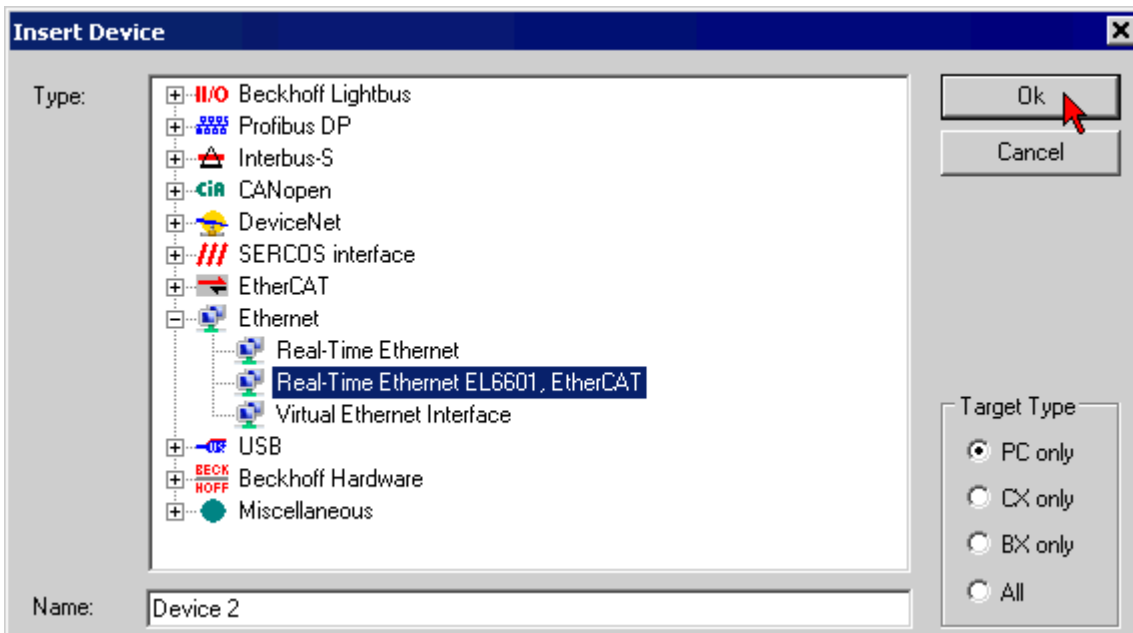


Fig. 19: Select EL6601

An imaginary box is now appended to the EL6601 as publisher or subscriber.

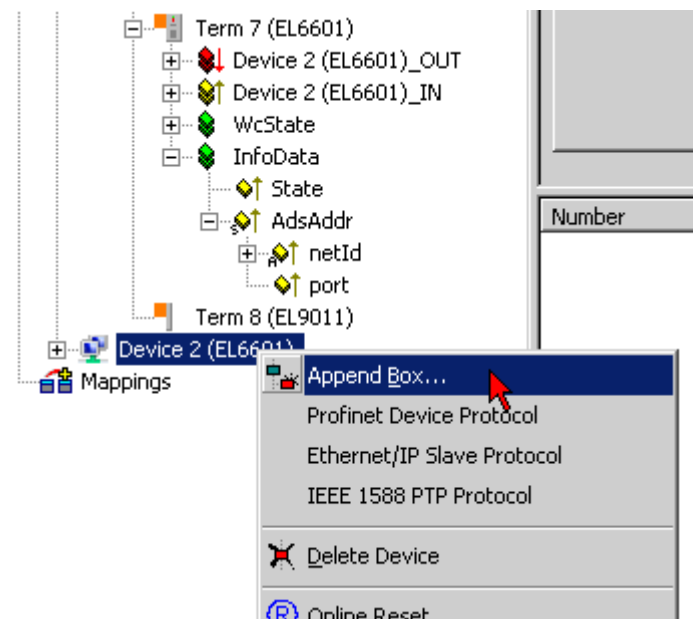


Fig. 20: Append box

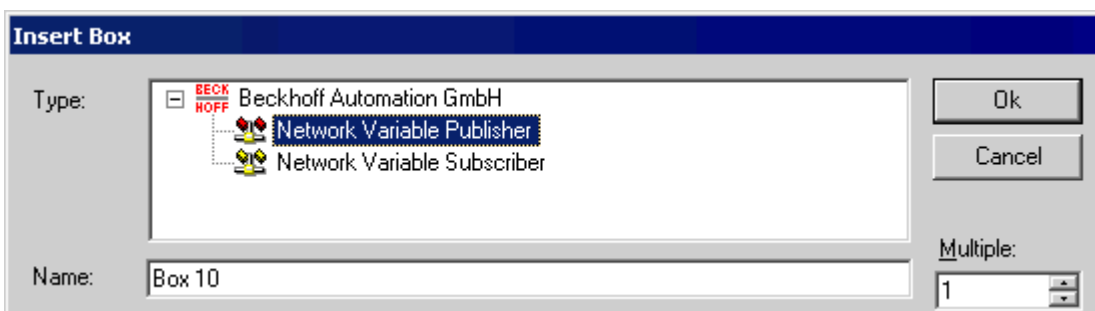


Fig. 21: Append network variable

The "EL6601 device" is now linked to the actual EL6601 or EL6614 in the selection dialog ("Adapter" tab -> "Search...").

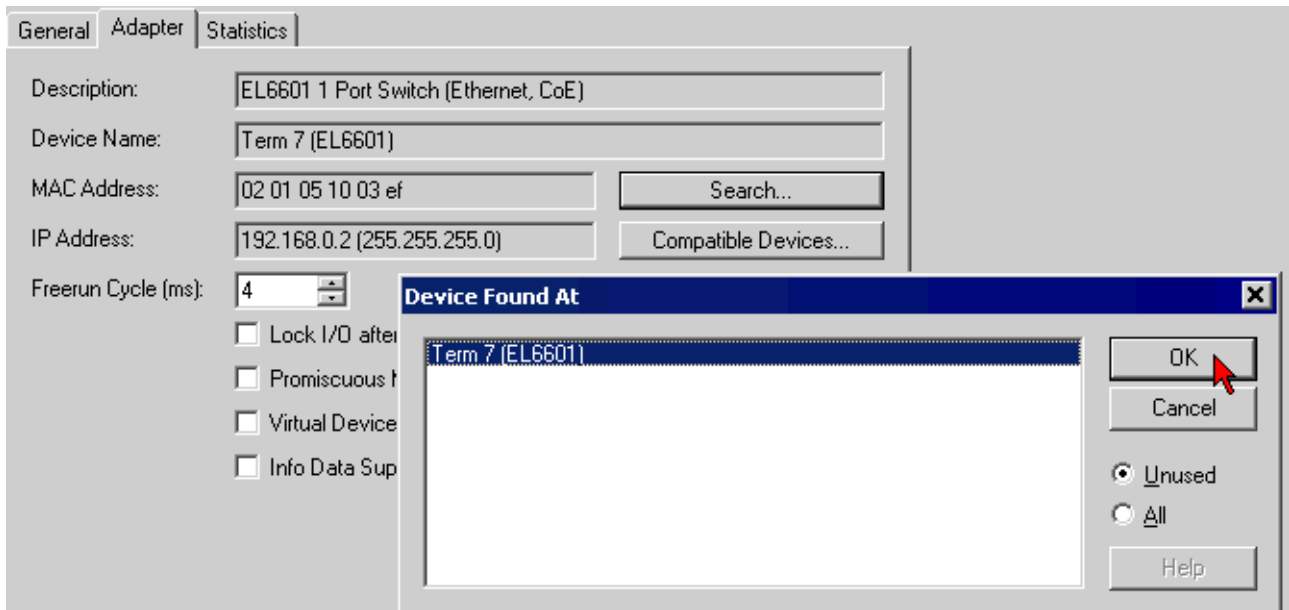


Fig. 22: Link device with EL6601

All further steps are done as described in the preceding sections.

2.5.7 Setting up TwinCAT 2.11

If the EtherCAT configuration has been created manually or scanned from the field itself you can now configure an EL66xx as a transmitter/receiver of network variables.

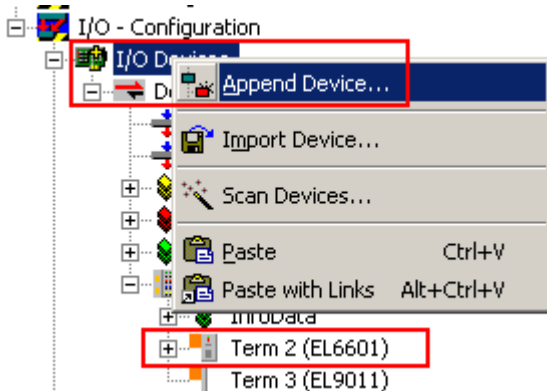


Fig. 23: Append new device

Select the EtherCAT Automation Protocol in the device dialog:



Fig. 24: Select EtherCAT Automation Protocol

The new device is automatically assigned to an available EL66xx, or this can also be done manually:

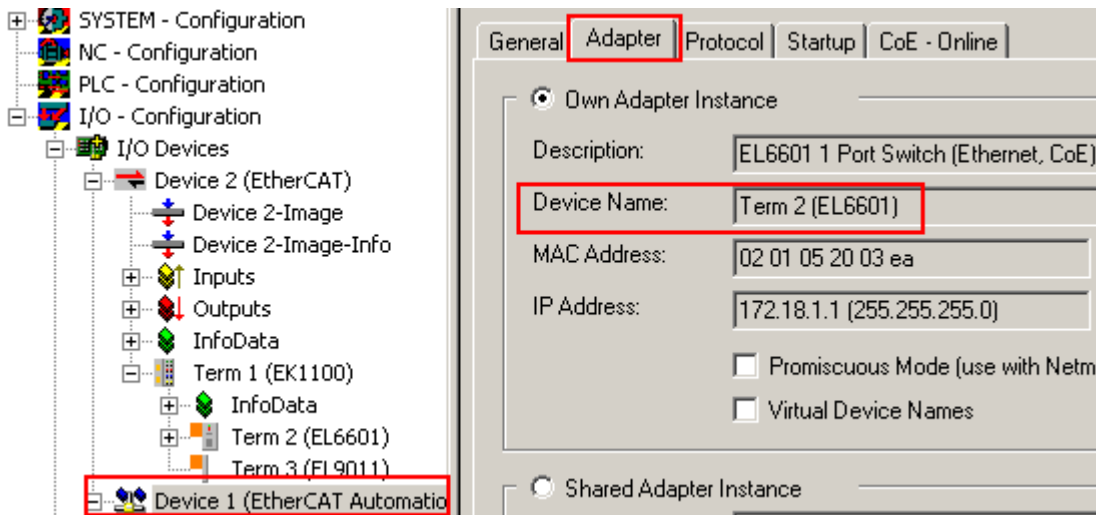


Fig. 25: Device assignment to the EL66xx

Transmitter/receiver variables must now be created:

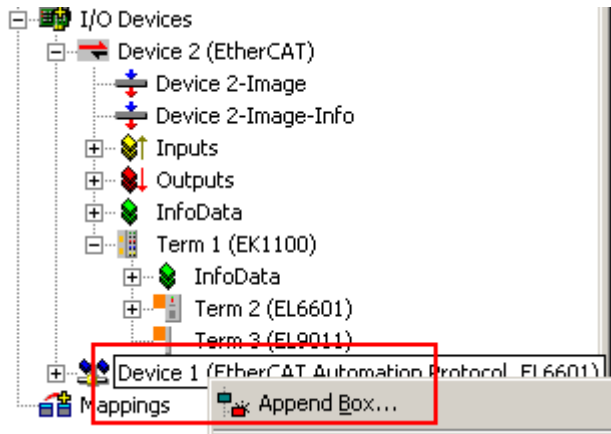


Fig. 26: Append box

Multiple publishers and subscribers can be created for each EtherCAT Automation Protocol device.



Fig. 27: Publisher/Subscriber

An EtherCAT Automation Protocol device appears as follows in the topology view:

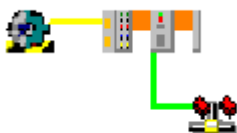


Fig. 28: Topology view

All further steps are done as described in the preceding sections.

2.6 Configuration in the CX20x0 & CX50x0 system

The embedded PCs of CX20x0 and CX50x0 series feature a special integrated I/O interface for E-bus and K-bus with automatic switching. The EL66xx devices in the TwinCAT system generally appear as virtual switch, with the EtherCAT system as the "backbone". In the CX20x0 and CX50x0 system, the internal interface connection is not implemented through a network interface, but through an FPGA.

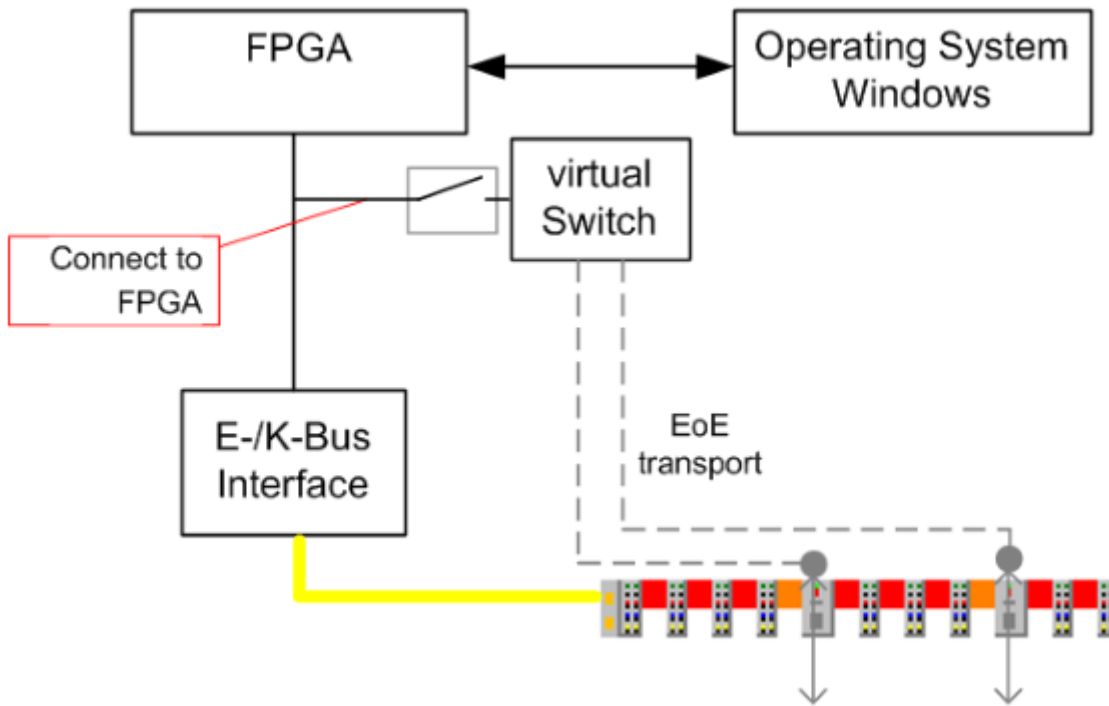


Fig. 29: Virtual TwinCAT switch in the CX20x0 & CX50x0 system

Due to the internal connection via FPGA and the automatic E-bus and K-bus detection, with offline configuration the Ethernet port only becomes visible when the configuration is activated. To configure the Ethernet port offline, proceed as follows:

- Due to the automatic E-bus and K-bus switching, any terminal should be connected with the appropriate bus
- The internal PCI port is detected during offline configuration and must be selected

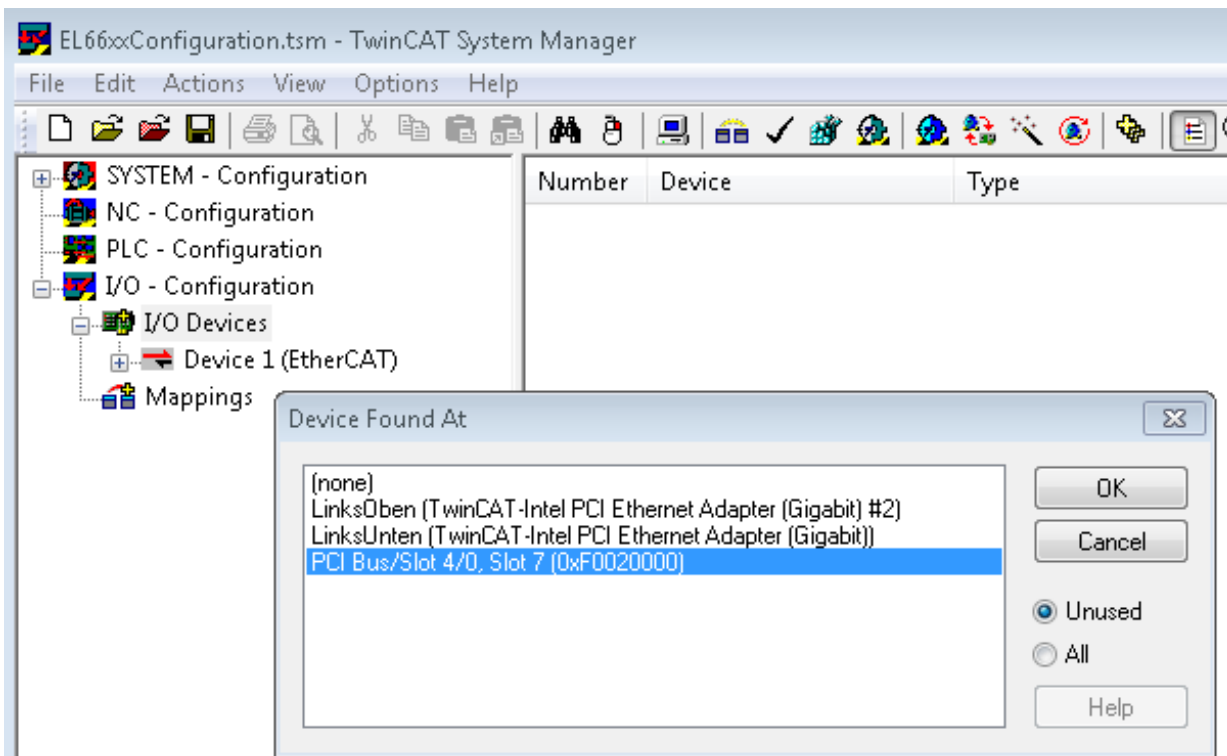


Fig. 30: Dialog for selection of the PCI port

- The customer specified configuration can be created, and the EL66xx can be inserted in the configuration

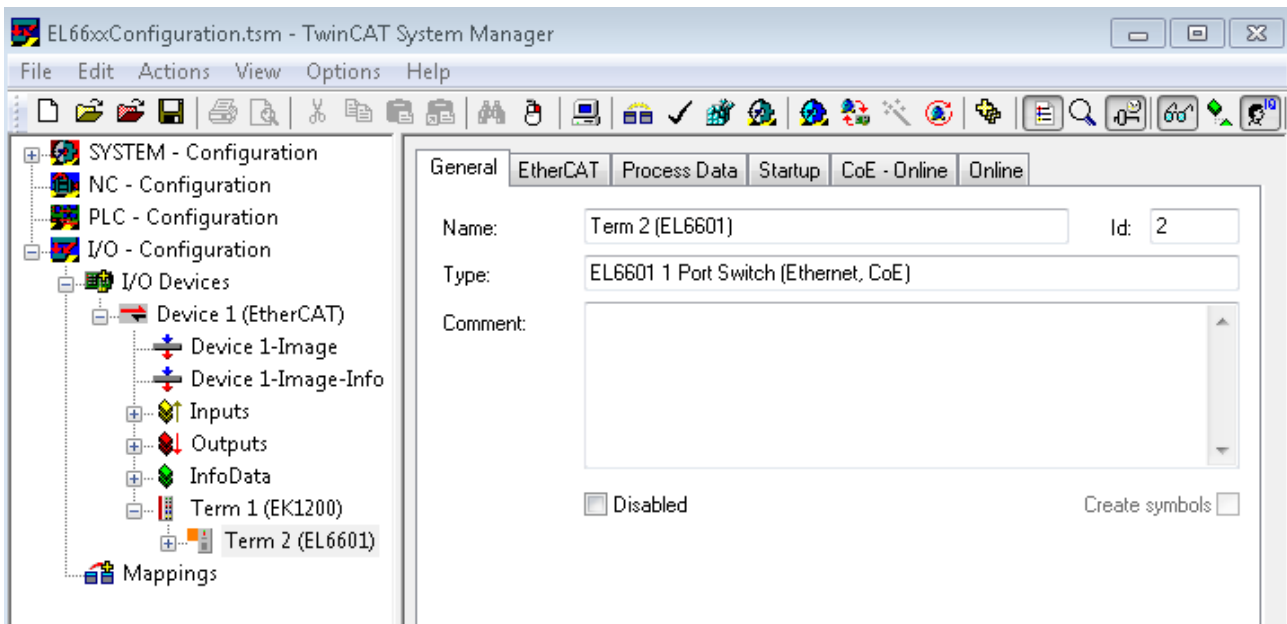


Fig. 31: Insertion of the EL66xx in the Configuration

- The Ethernet port is detected after "Reload I/O devices" (F4) and then appears under network connections
- as "Local Area Connection 4"

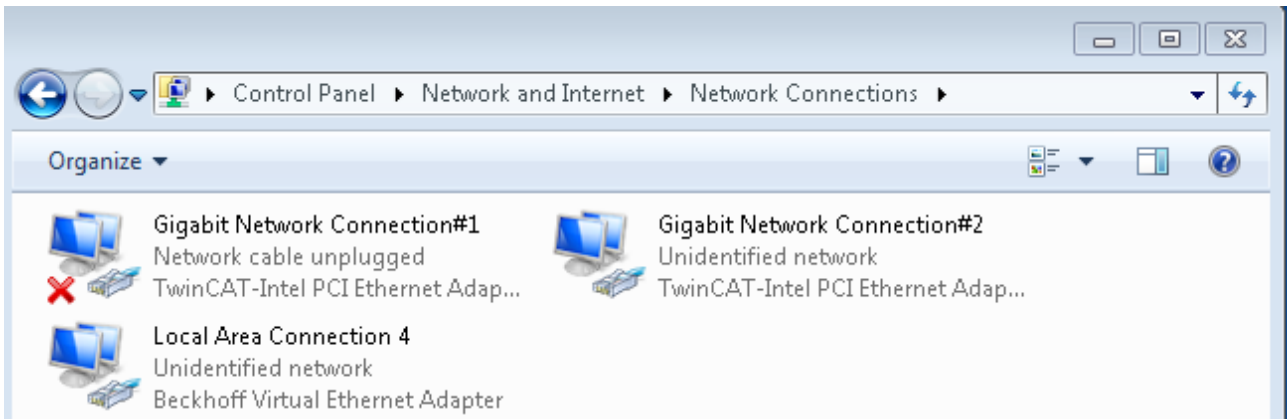


Fig. 32: New Network “Local Area Connection” in the Windows network connections

- The port can now be configured as required. The settings are applied and saved. Even if the port disappears again, the settings are retained for subsequent commissioning.

If the problem persists, i.e. if the Ethernet port of the EL66xx still fails to show up in the network connection, see troubleshooting tips below. Follow these tips and the countermeasures listed.

Prerequisites

Check the following:	
Virtual Ethernet switch is not enabled	Check the virtual switch settings [p. 28] and the corresponding notes
TwinCAT2 and TwinCAT3 are installed simultaneously	Possible driver conflict, please contact Beckhoff support

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

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

E-Bus supply

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

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

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

Fig. 33: System manager current calculation

NOTICE

Malfunction possible!

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

3.3 General notes for setting the watchdog

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

The EtherCAT slave controller features two watchdogs:

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

Their times are individually parameterized in TwinCAT as follows:

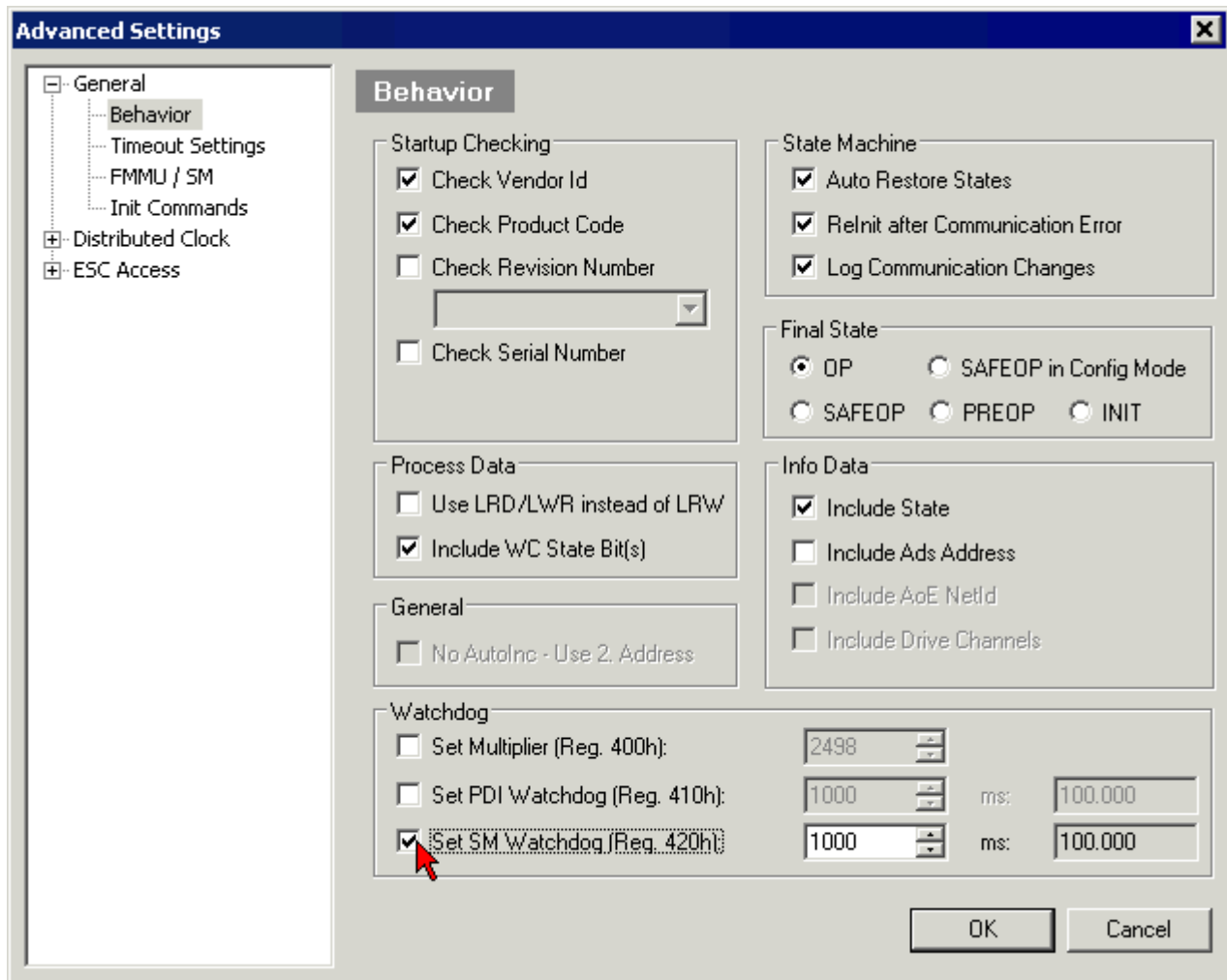


Fig. 34: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

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

SM watchdog (SyncManager Watchdog)

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

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

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

PDI watchdog (Process Data Watchdog)

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

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

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

Calculation

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

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

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

⚠ CAUTION

Undefined state possible!

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

⚠ CAUTION

Damage of devices and undefined state possible!

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

3.4 EtherCAT State Machine

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

A distinction is made between the following states:

- Init
- Pre-Operational
- Safe-Operational
- Operational

- Bootstrap

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

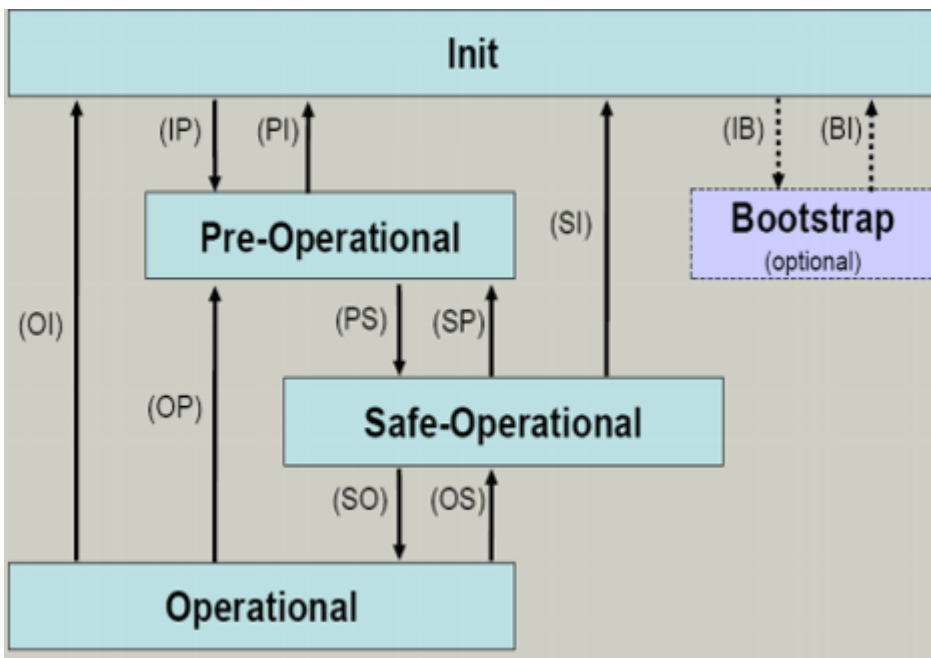


Fig. 35: States of the EtherCAT State Machine

Init

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

Pre-Operational (Pre-Op)

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

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

Safe-Operational (Safe-Op)

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

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

● **Outputs in SAFEOP state**

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

Operational (Op)

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

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

Boot

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

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

3.5 CoE Interface

General description

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

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

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

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

The value ranges are

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

Other important ranges are:

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

● Availability

i 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:

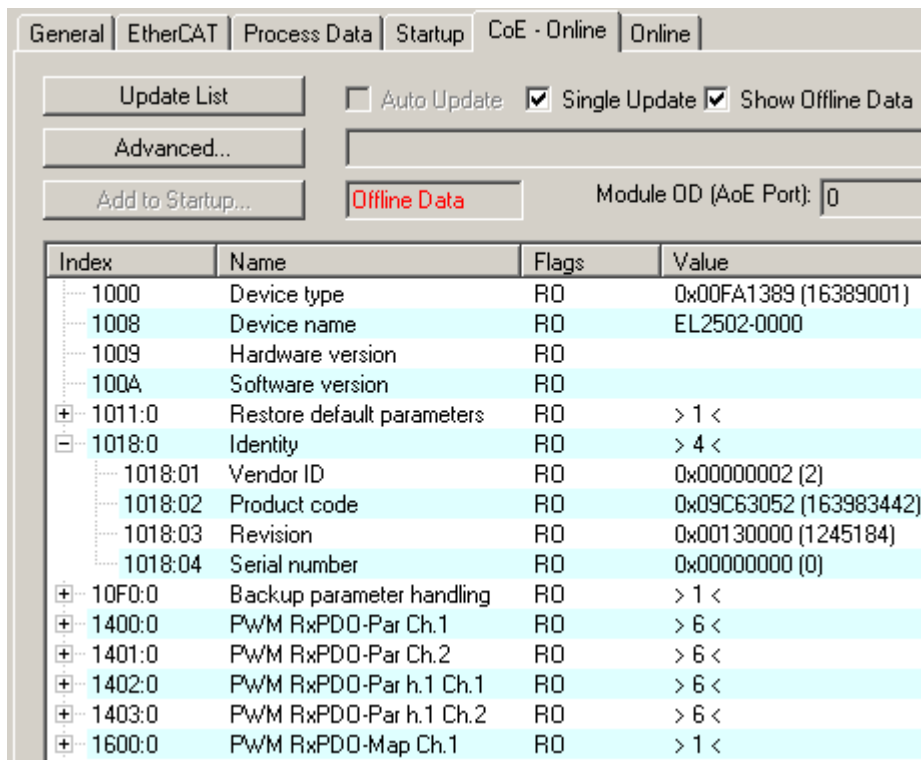


Fig. 36: “CoE Online” tab

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

NOTICE

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

When using/manipulating the CoE parameters observe the general CoE notes in chapter "[CoE interface](#)" of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the [Beckhoff website](#)),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see [TwinCAT3 | PLC Library: Tc2_EtherCAT](#) and [Example program R/W CoE](#))

Data management and function “NoCoeStorage”

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

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

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

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.

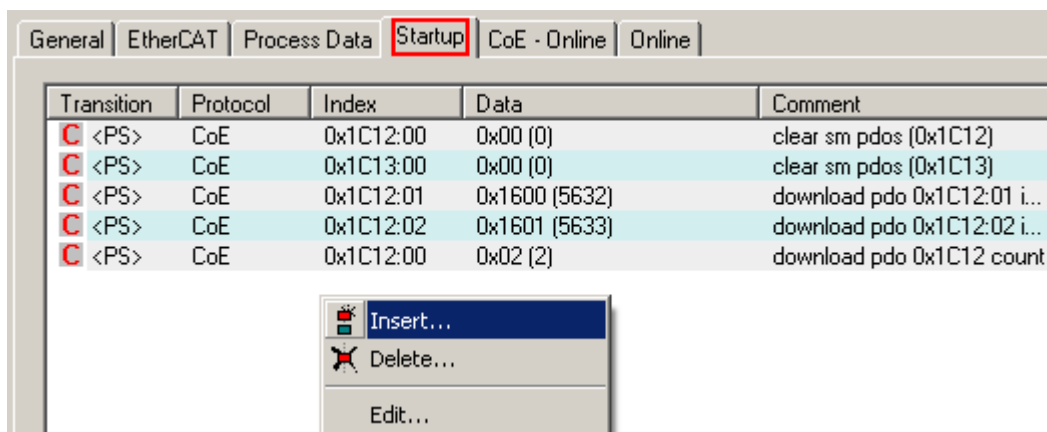


Fig. 37: Startup list in the TwinCAT System Manager

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

Online / offline list

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

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

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

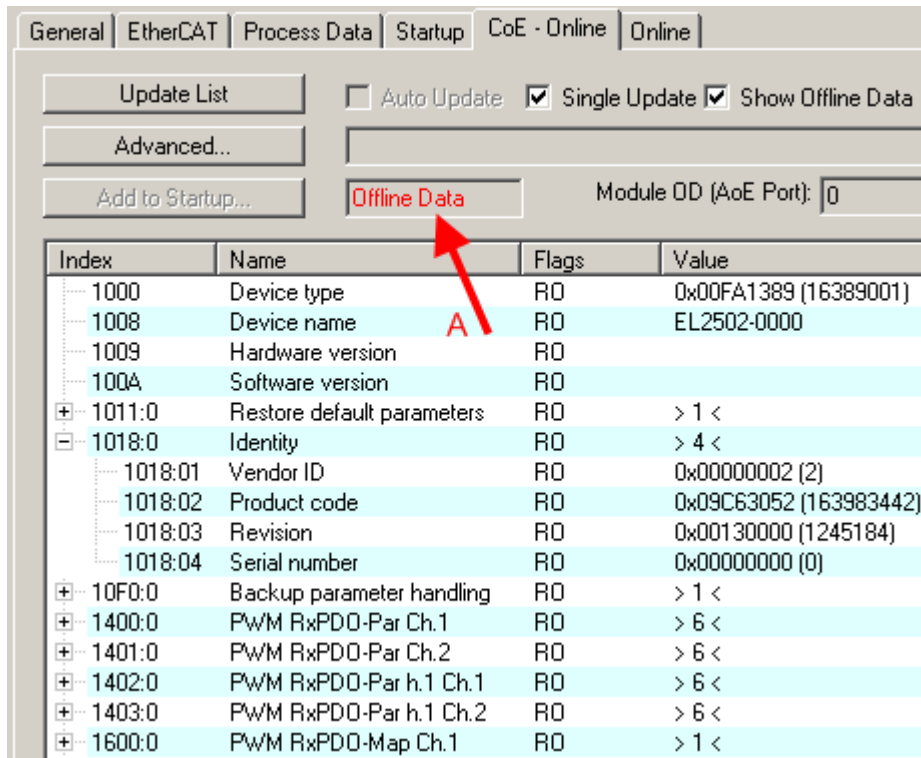


Fig. 38: Offline list

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

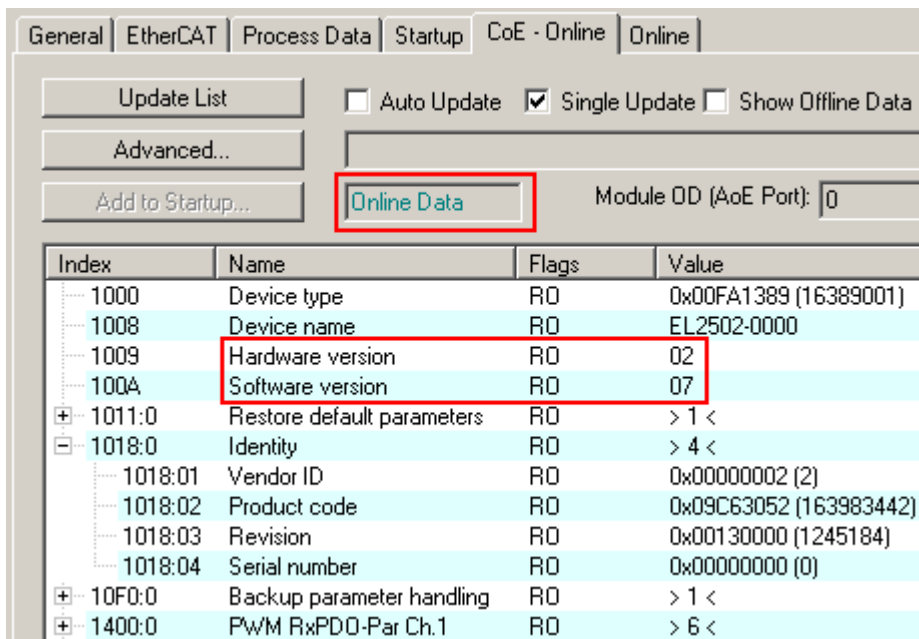


Fig. 39: Online list

Channel-based order

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

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

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

This is generally written as 0x80n0.

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

3.6 Distributed Clock

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

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

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

4 Mounting and wiring

4.1 Instructions for ESD protection

NOTICE

Destruction of the devices by electrostatic discharge possible!

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

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

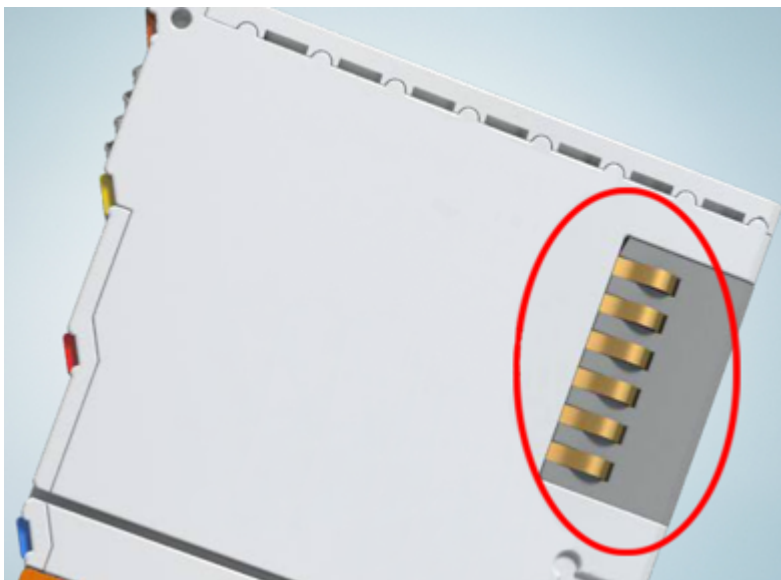


Fig. 40: Spring contacts of the Beckhoff I/O components

4.2 Explosion protection

4.2.1 ATEX - Special conditions (extended temperature range)

⚠ WARNING

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

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60079-15! The environmental conditions during use are thereby to be taken into account!
- For dust (only the fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9): The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-31 for group IIIA or IIIB and IP6X for group IIIC, taking into account the environmental conditions under which the equipment is used!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of -25 to 60°C for the use of Beckhoff fieldbus components with extended temperature range (ET) in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

Standards

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

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

Marking

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



II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: -25 ... +60°C
 II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: -25 ... +60°C
 (only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

or



II 3G KEMA 10ATEX0075 X Ex nA nC IIC T4 Gc Ta: -25 ... +60°C
 II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: -25 ... +60°C
 (only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

4.2.2 IECEx - Special conditions

WARNING

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

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

Standards

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

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

Marking

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

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

Marking for fieldbus components of certificates with later issues:	IECEx DEK 16.0078 X
	Ex nA IIC T4 Gc

4.2.3 Continulative documentation for ATEX and IECEx

NOTICE



Continulative documentation about explosion protection according to ATEX and IECEx

Pay also attention to the continuative documentation

Ex. Protection for Terminal Systems

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

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

4.2.4 cFMus - Special conditions

WARNING

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

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

Standards

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

M20US0111X (US):

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

FM20CA0053X (Canada):

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

Marking

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

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

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

4.2.5 Continulative documentation for cFMus

NOTICE



Continulative documentation about explosion protection according to cFMus

Pay also attention to the continuative documentation

Control Drawing I/O, CX, CPX

Connection diagrams and Ex markings,

that is available for download within the download area of your product on the Beckhoff homepage www.beckhoff.com!

4.3 UL notice

⚠ CAUTION



Application

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

⚠ CAUTION



Examination

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

⚠ CAUTION



For devices with Ethernet connectors

Not for connection to telecommunication circuits.

Basic principles

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



4.4 Note - power supply

WARNING

Power supply from SELV / PELV power supply unit!

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

Notes:

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

4.5 Mounting and demounting - top front unlocking

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

● Fixing of mounting rails

i 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).

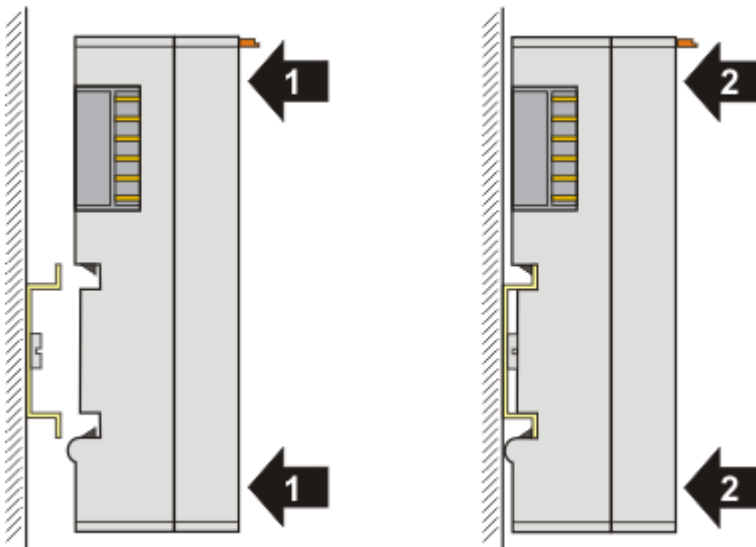
⚠ WARNING

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.

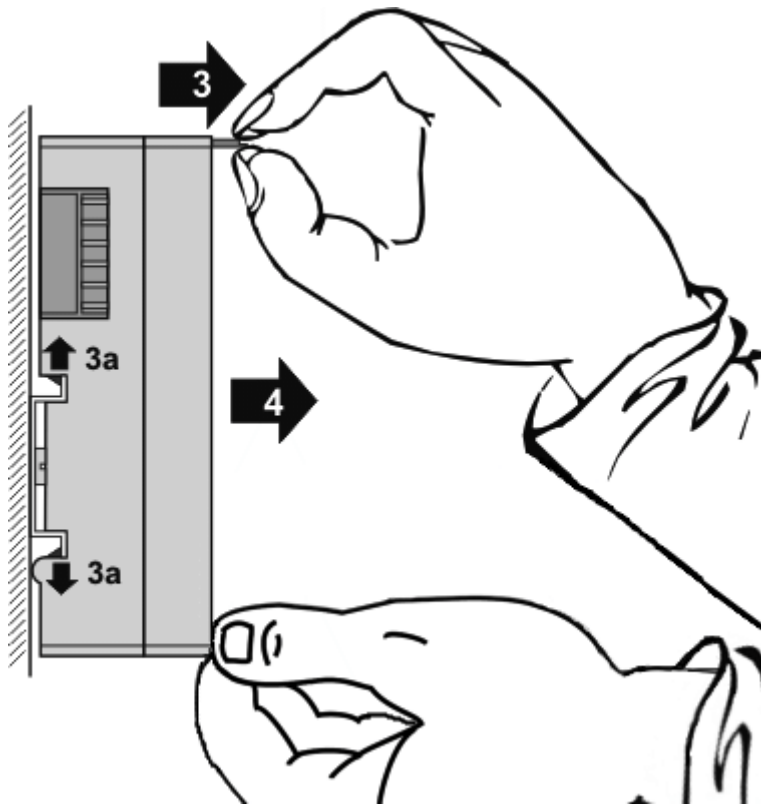


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

- 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.

4.6 Positioning of passive Terminals

i Hint for positioning of passive terminals in the bus terminal block

EtherCAT Terminals (ELxxxx / ESxxxx), which do not take an active part in data transfer within the bus terminal block are so called passive terminals. The passive terminals have no current consumption out of the E-Bus.

To ensure an optimal data transfer, you must not directly string together more than two passive terminals!

Examples for positioning of passive terminals (highlighted)

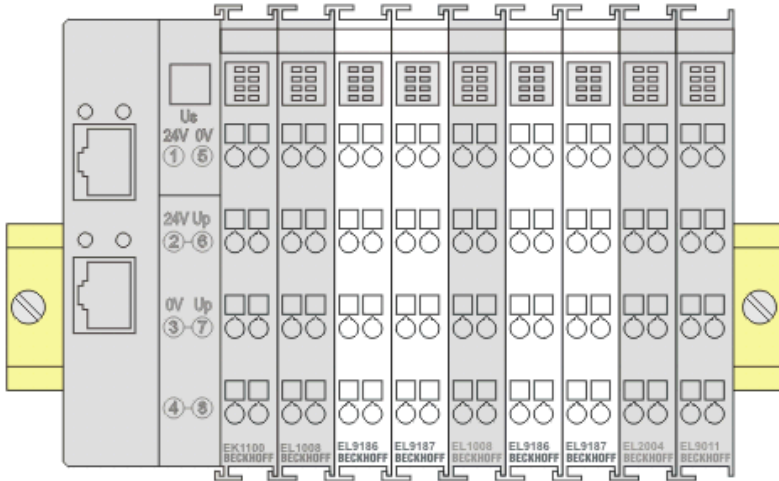


Fig. 41: Correct positioning

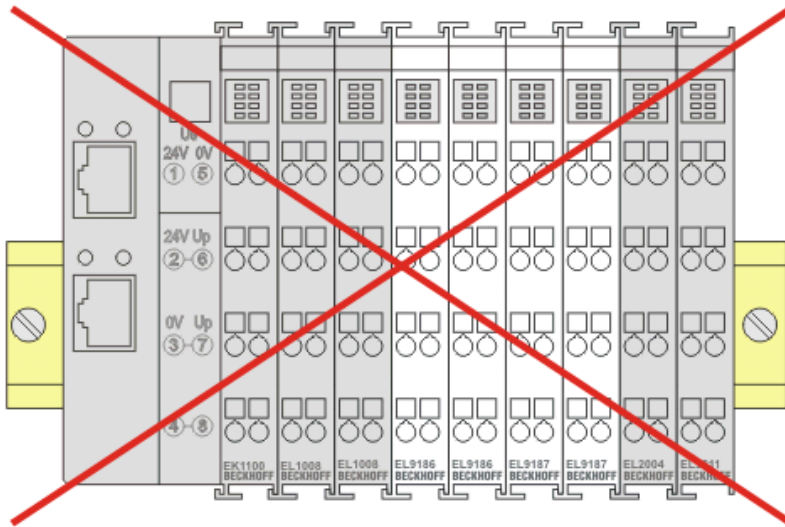


Fig. 42: Incorrect positioning

4.7 Note on shielding

NOTICE

Low-resistance shielding through external shield connection

The shielding of the EL66xx is capacitively connected to the mounting rail.

If low-resistance shielding is desired or required, the shield must be connected to an external shield support (e.g. with the clamp strap for shield connection [ZB5800](#)).

4.8 Installation positions

NOTICE

Constraints regarding installation position and operating temperature range

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

Optimum installation position (standard)

The optimum installation position requires the mounting rail to be installed horizontally and the connection surfaces of the EL- / KL terminals to face forward (see Fig. "Recommended distances for standard installation position"). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. "From below" is relative to the acceleration of gravity.

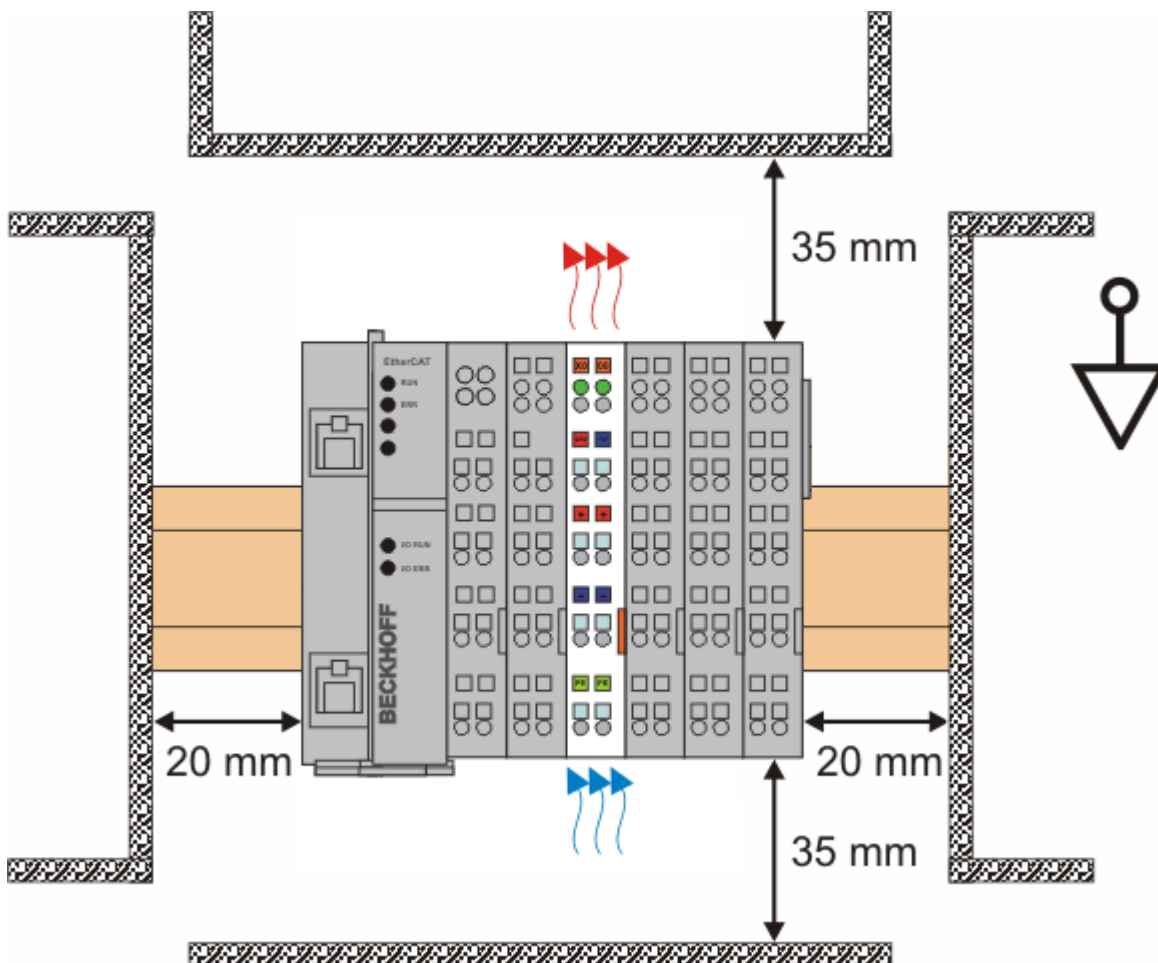


Fig. 43: Recommended distances for standard installation position

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.

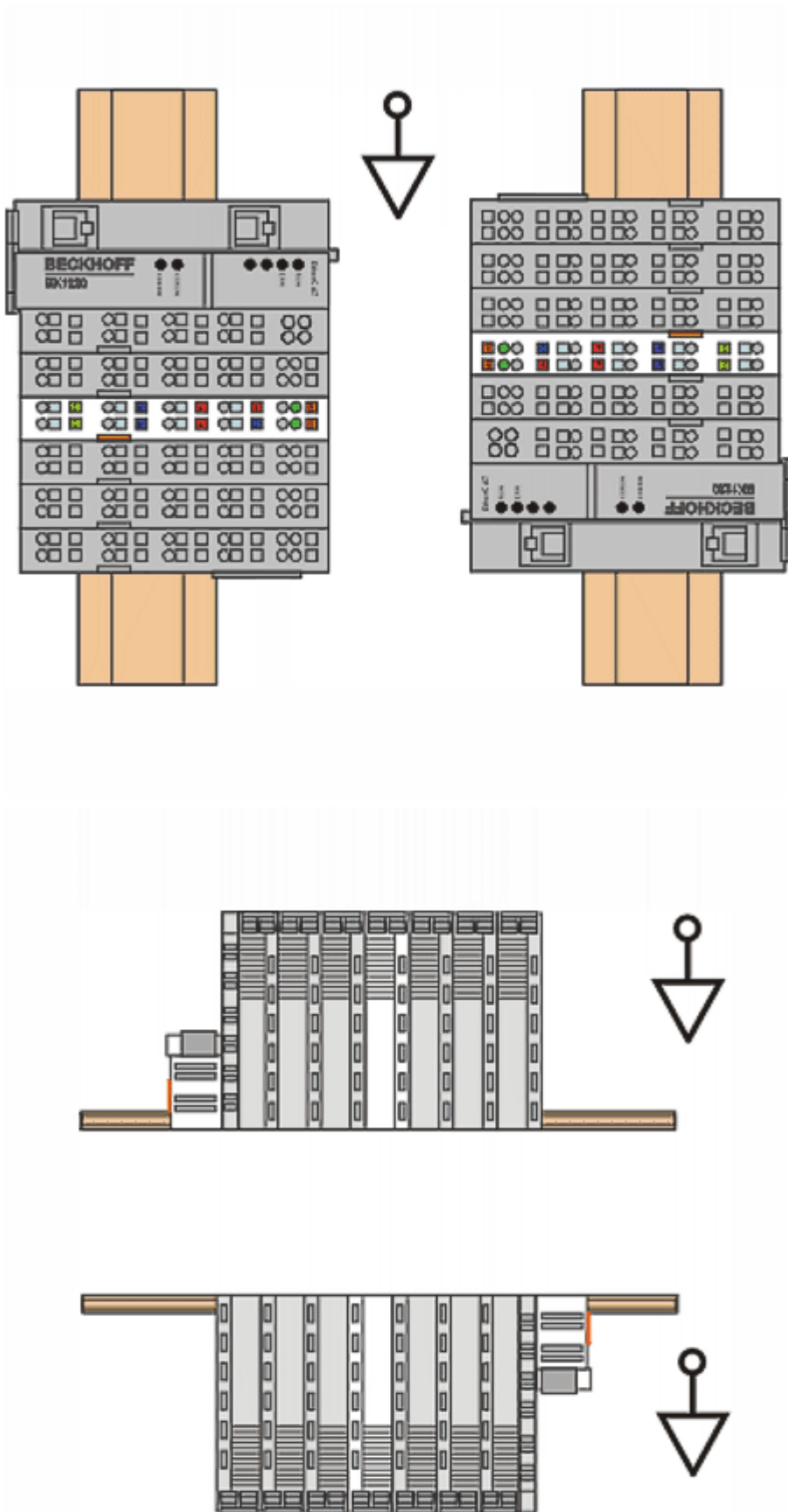


Fig. 44: Other installation positions

4.9 Disposal



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

5 Commissioning

5.1 TwinCAT Development Environment

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

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

Details:

- **TwinCAT 2:**
 - Connects I/O devices to tasks in a variable-oriented manner
 - Connects tasks to tasks in a variable-oriented manner
 - Supports units at the bit level
 - Supports synchronous or asynchronous relationships
 - Exchange of consistent data areas and process images
 - Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)
 - Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/2000/XP/Vista, Windows 7, NT/XP Embedded, CE
 - Interconnection to all common fieldbusses
 - More...

Additional features:

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

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

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

5.1.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways.

A: Via the TwinCAT Adapter dialog

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

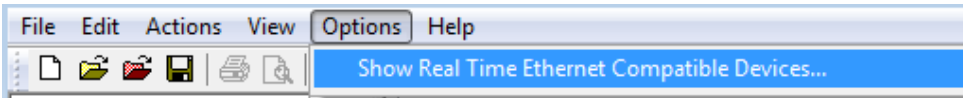


Fig. 45: System Manager “Options” (TwinCAT 2)

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

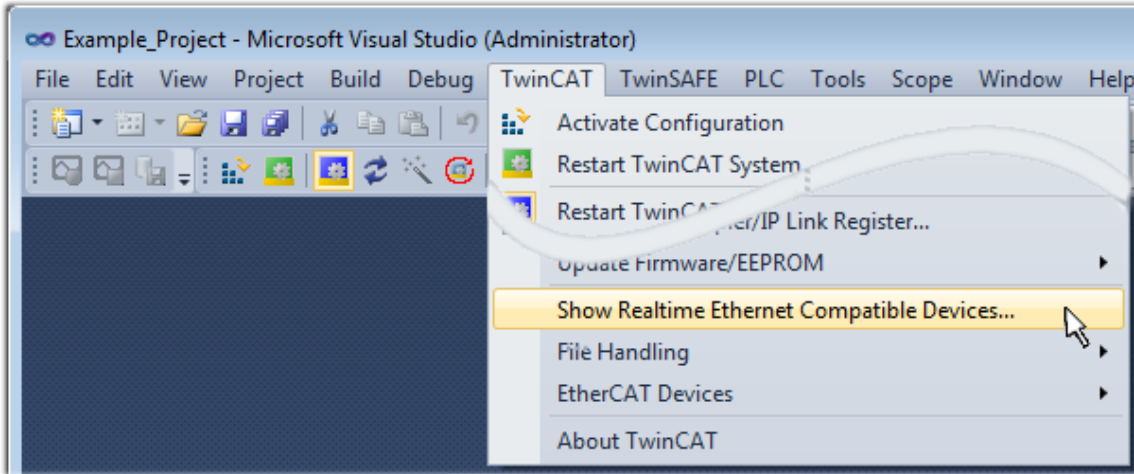


Fig. 46: Call up under VS Shell (TwinCAT 3)

B: Via TcRtelInstall.exe in the TwinCAT directory

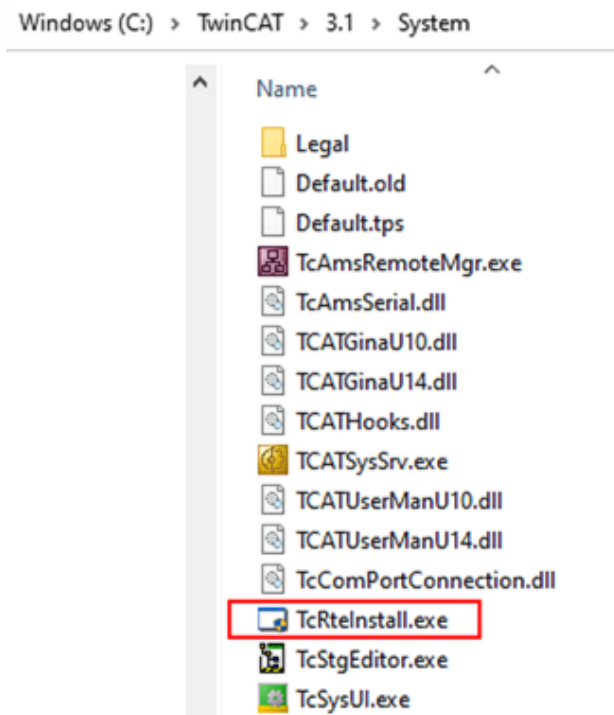


Fig. 47: TcRtelInstall in the TwinCAT directory

In both cases, the following dialog appears:

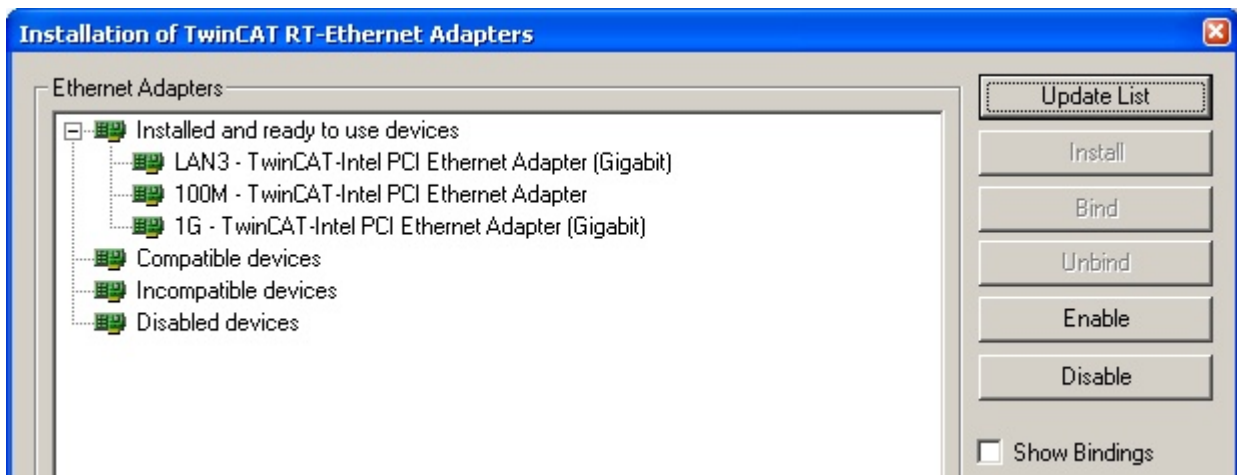


Fig. 48: Overview of network interfaces

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

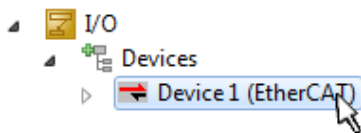
A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter [Offline configuration creation](#), section “Creating the EtherCAT device” [▶ 77] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices...”):



Fig. 49: 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)

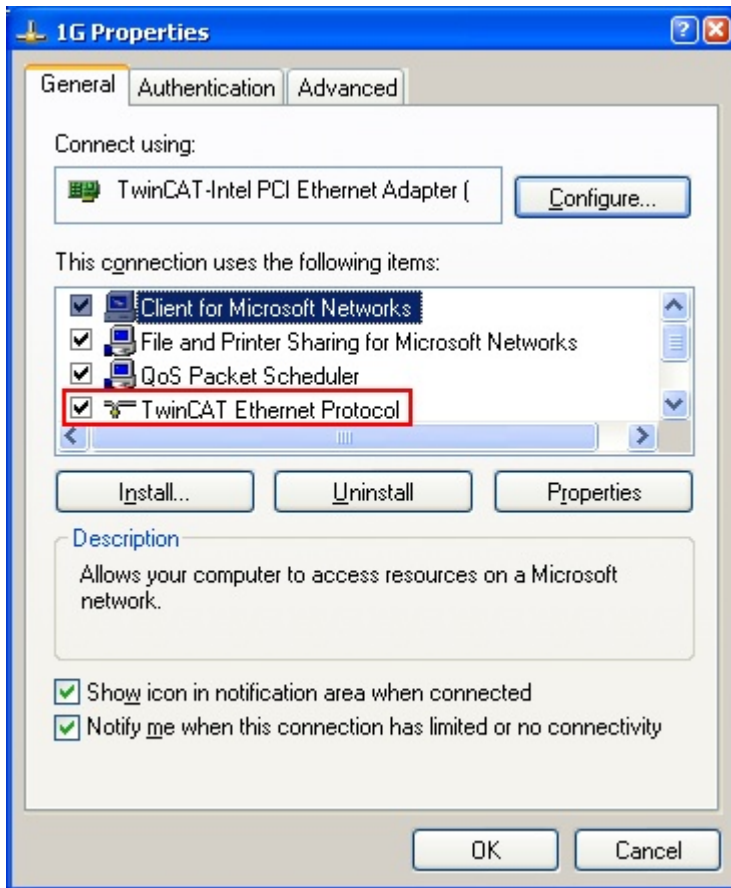


Fig. 50: Windows properties of the network interface

A correct setting of the driver could be:

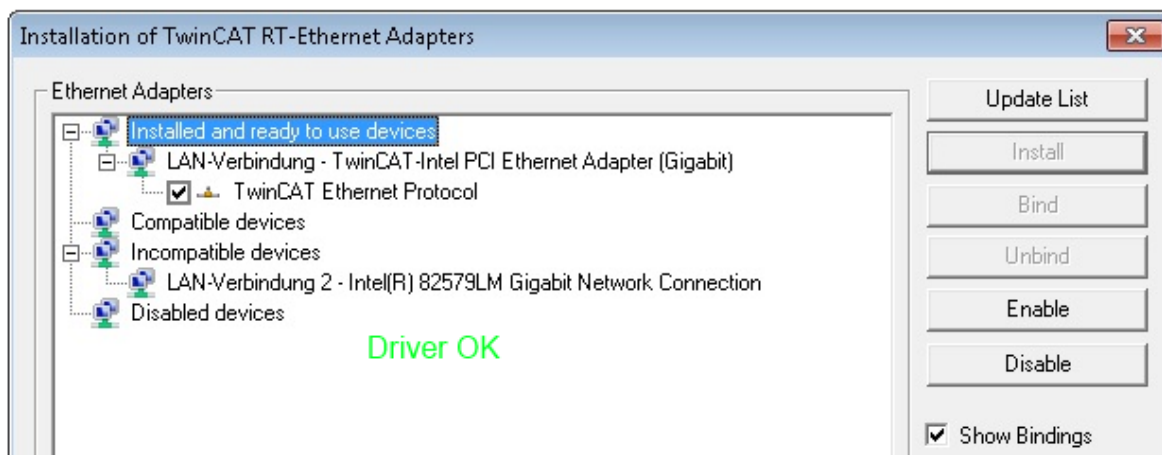


Fig. 51: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

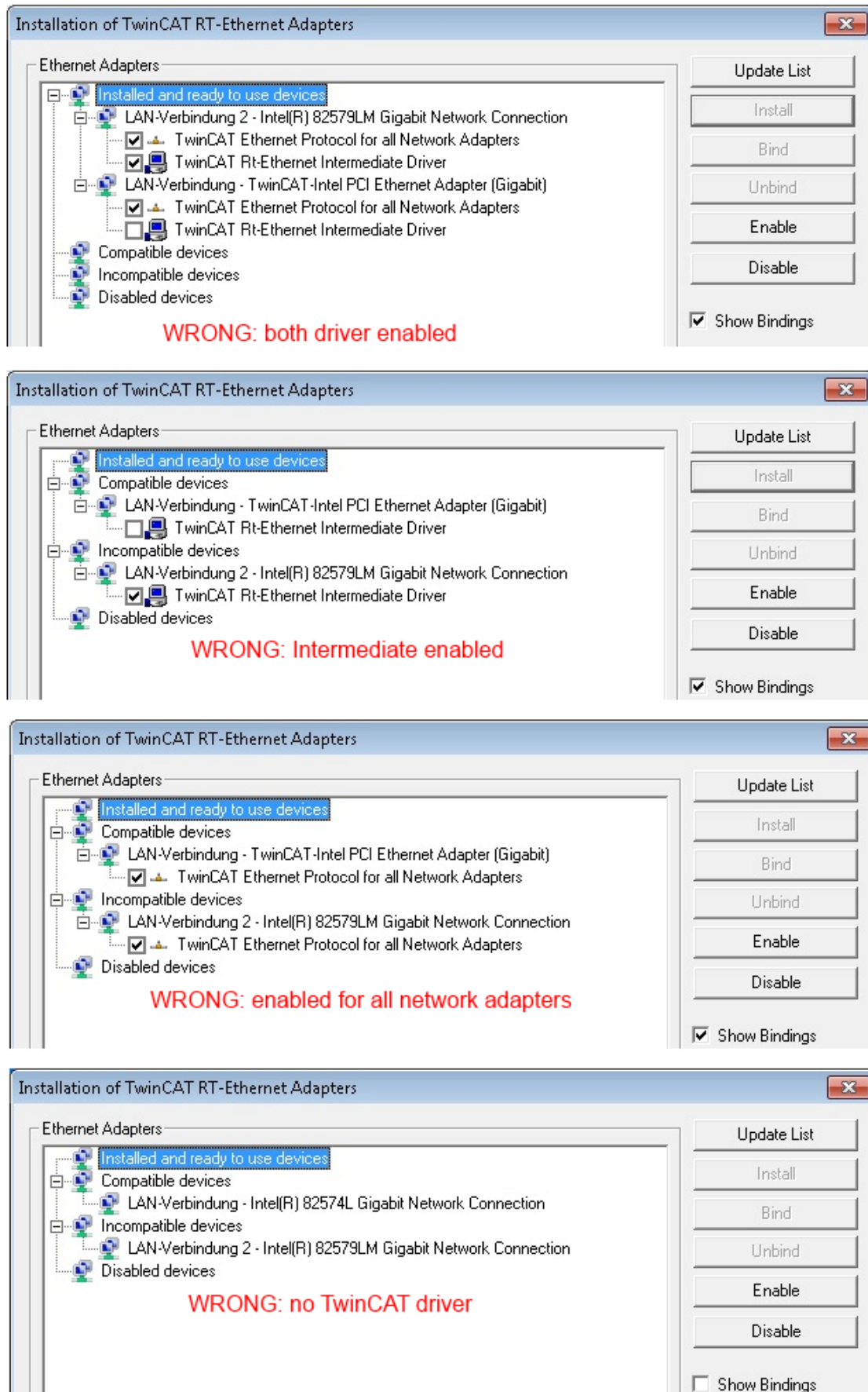


Fig. 52: Incorrect driver settings for the Ethernet port

IP address of the port used

● IP address/DHCP

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

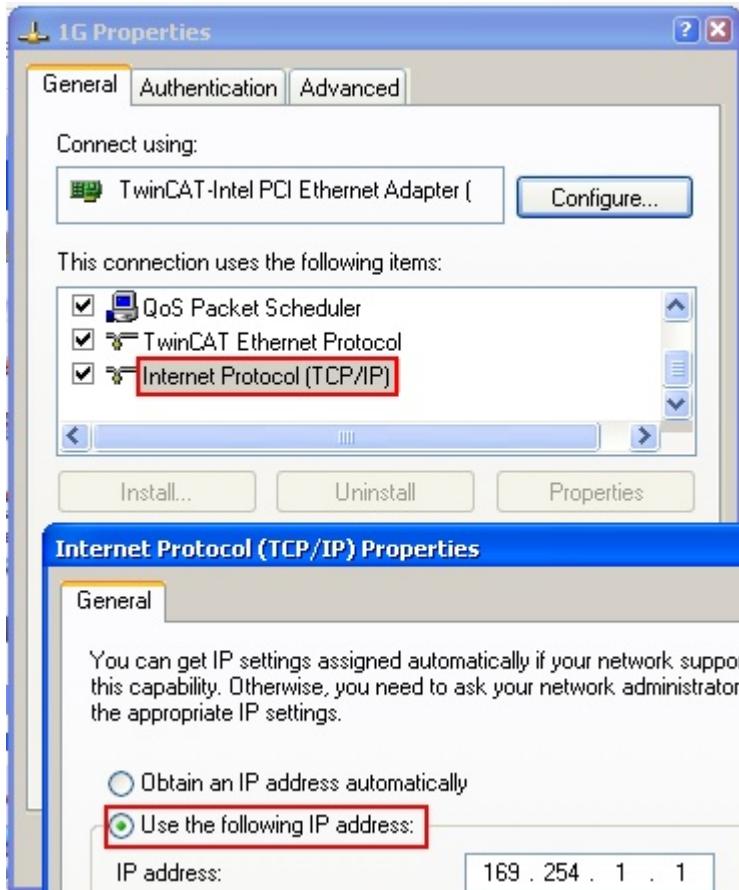


Fig. 53: TCP/IP setting for the Ethernet port

5.1.2 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 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”

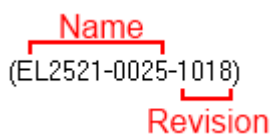


Fig. 54: Identifier structure

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

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.

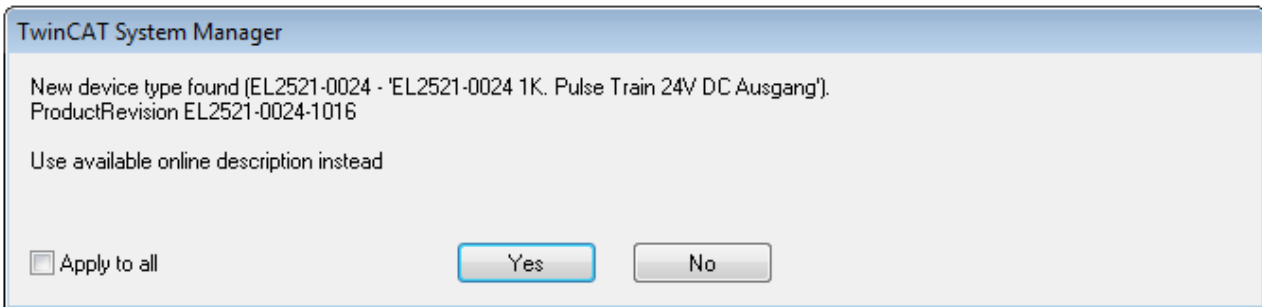


Fig. 55: OnlineDescription information window (TwinCAT 2)

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

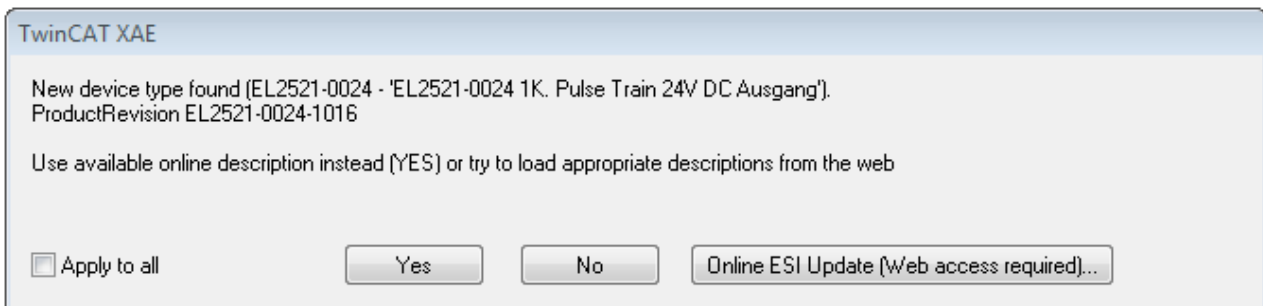


Fig. 56: Information window OnlineDescription (TwinCAT 3)

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

NOTICE

Changing the “usual” configuration through a scan

- ✓ If a scan discovers a device that is not yet known to TwinCAT, distinction has to be made between two cases. Taking the example here of the EL2521-0000 in the revision 1019
 - a) no ESI is present for the EL2521-0000 device at all, either for the revision 1019 or for an older revision. The ESI must then be requested from the manufacturer (in this case Beckhoff).
 - b) an ESI is present for the EL2521-0000 device, but only in an older revision, e.g. 1018 or 1017. In this case an in-house check should first be performed to determine whether the spare parts stock allows the integration of the increased revision into the configuration at all. A new/higher revision usually also brings along new features. If these are not to be used, work can continue without reservations with the previous revision 1018 in the configuration. This is also stated by the Beckhoff compatibility rule.

Refer in particular to the chapter “[General notes on the use of Beckhoff EtherCAT IO components](#)” and for manual configuration to the chapter “[Offline configuration creation \[► 77\]](#)”.

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. 57: 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*).

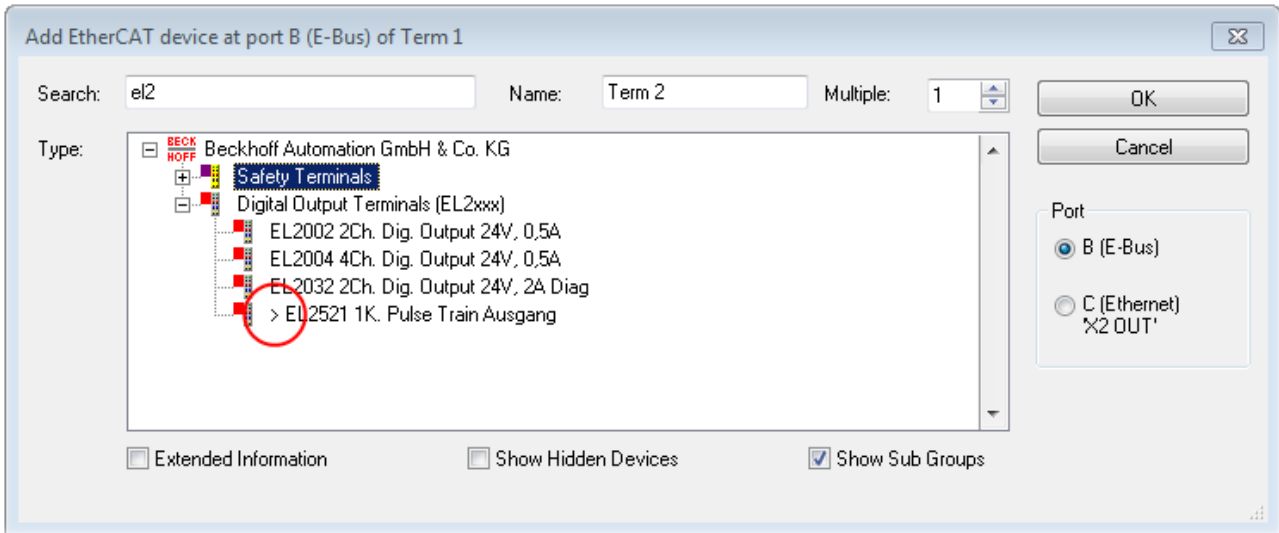


Fig. 58: Indication of an online recorded ESI of EL2521 as an example

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.

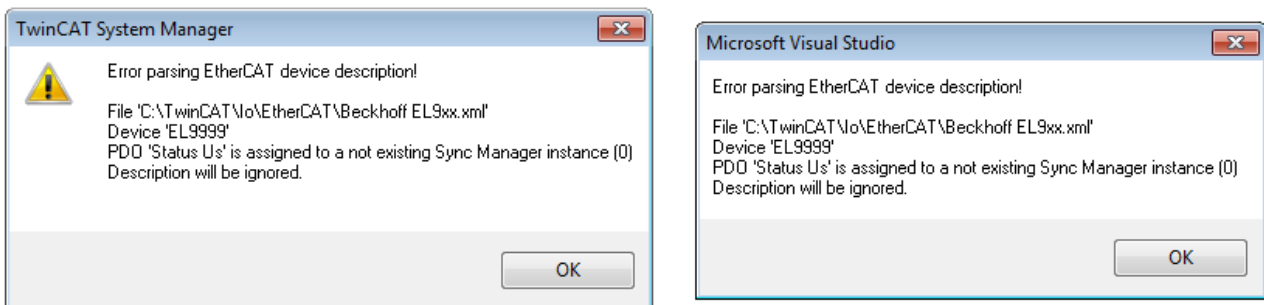


Fig. 59: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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

5.1.3 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

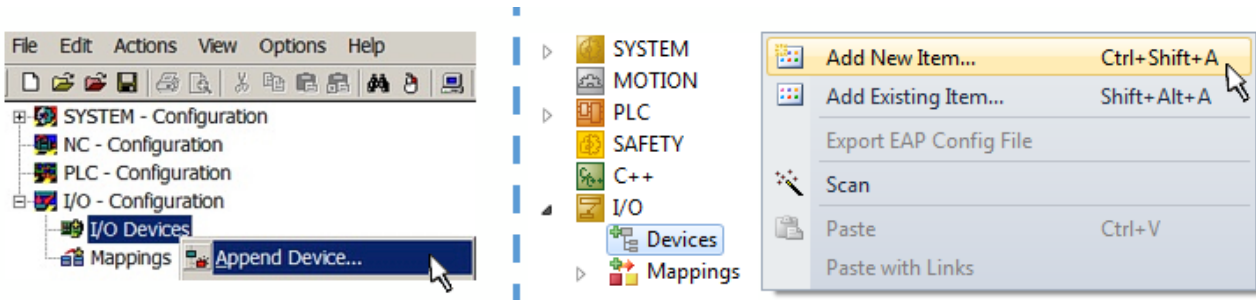


Fig. 60: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

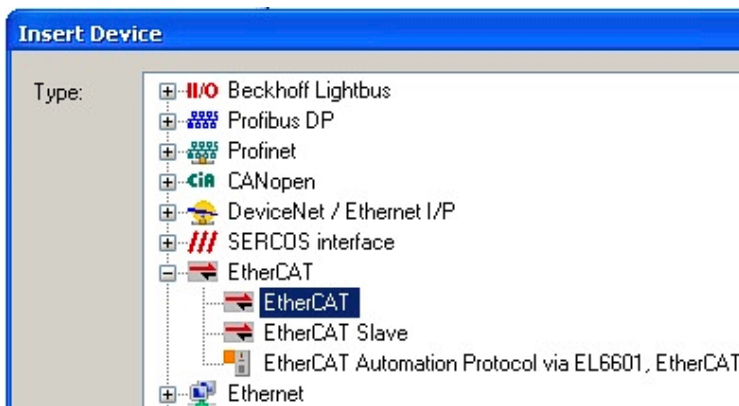


Fig. 61: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

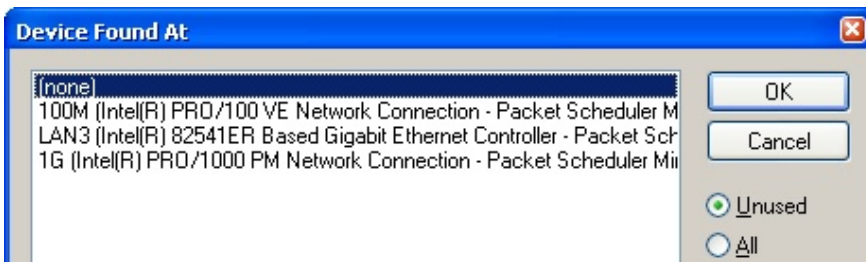


Fig. 62: Selecting the Ethernet port

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)”.

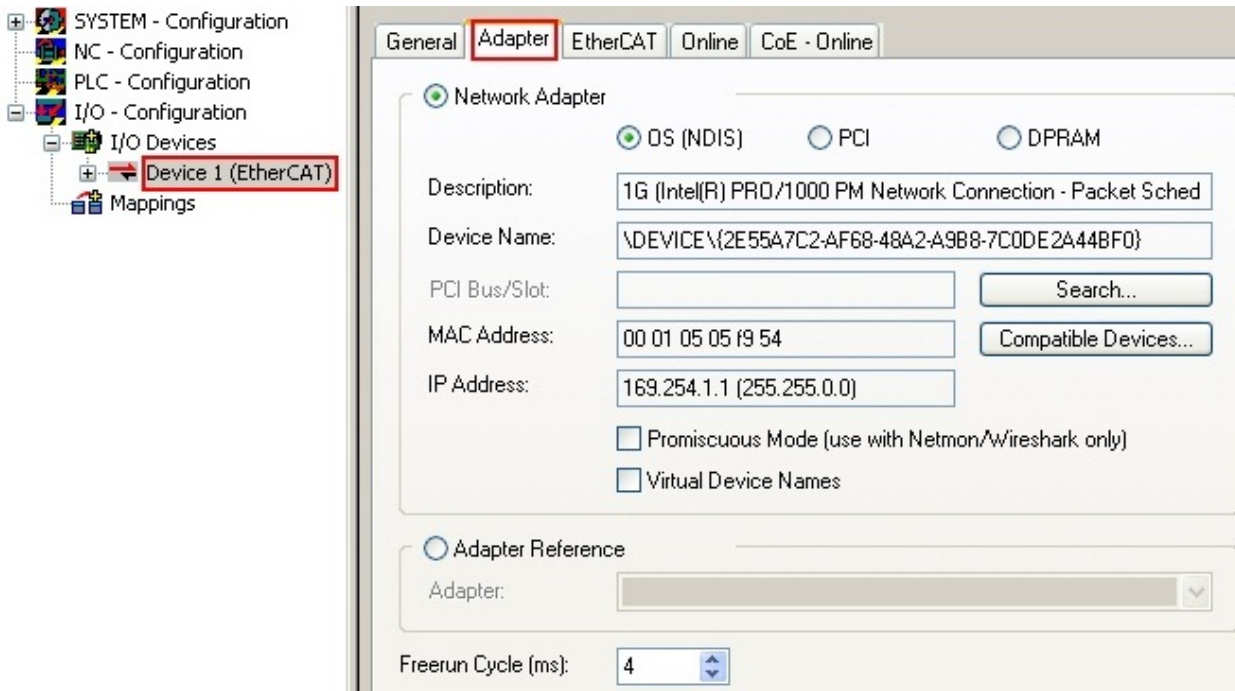
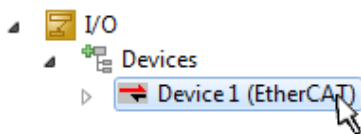


Fig. 63: EtherCAT device properties (TwinCAT 2)

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](#) [▶ 67].

Defining EtherCAT slaves

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

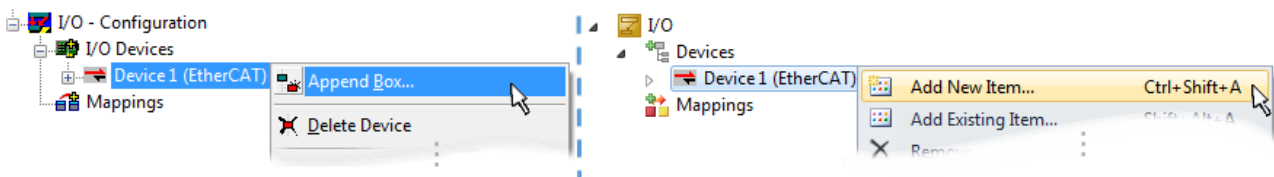


Fig. 64: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

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

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: couplers, box modules, devices with RJ45/M8/M12 connector
- “E-Bus”: LVDS “terminal bus”, EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

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

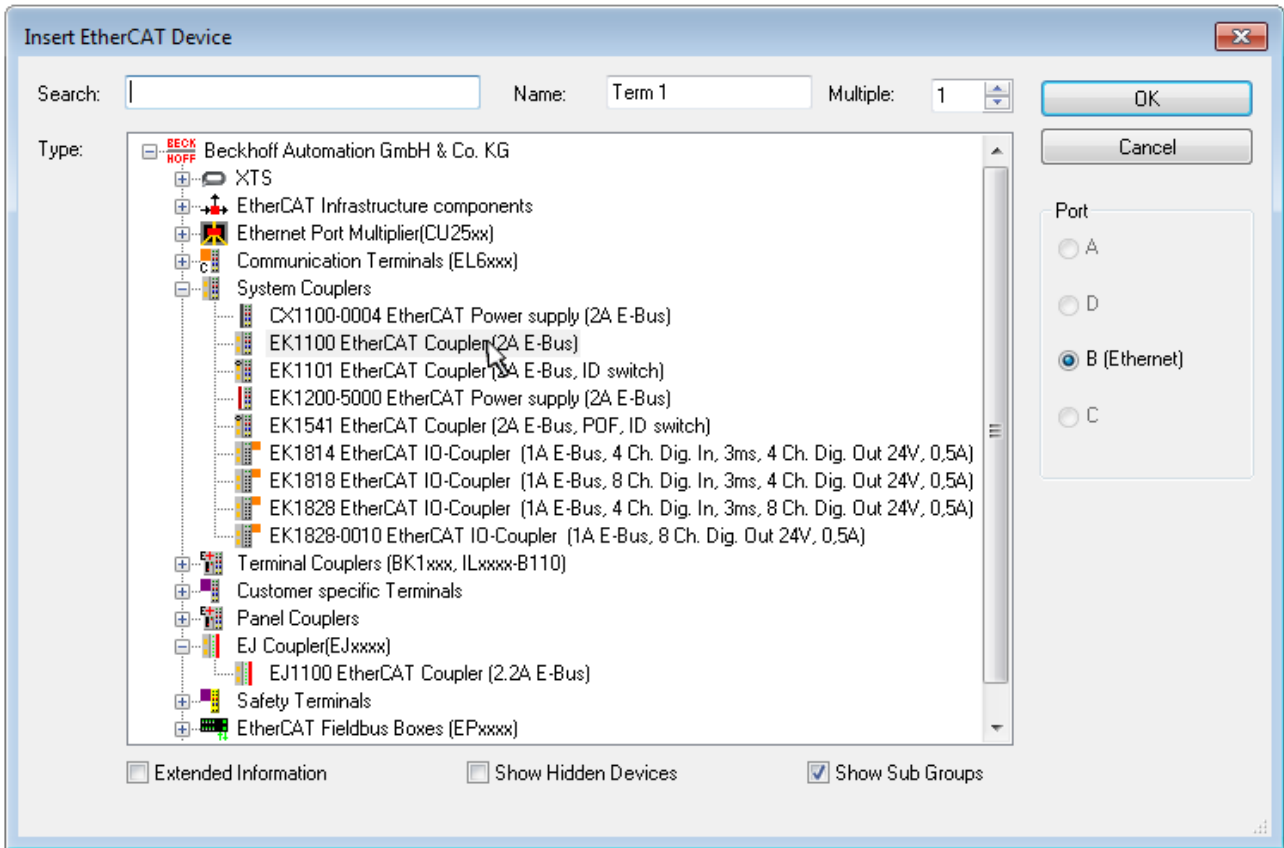


Fig. 65: Selection dialog for new EtherCAT device

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”.

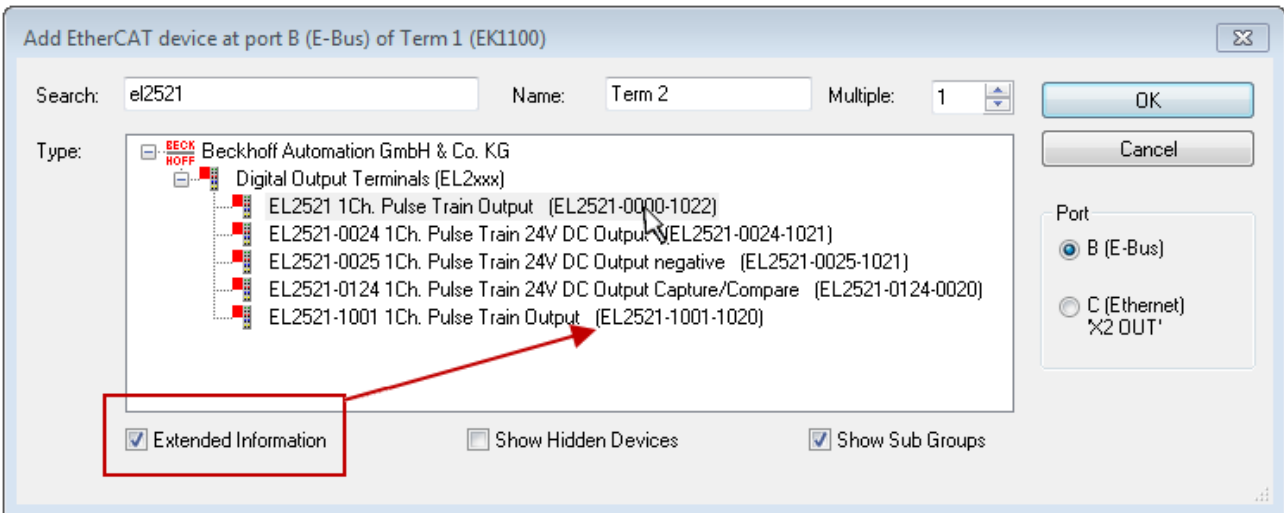


Fig. 66: Display of device revision

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”.

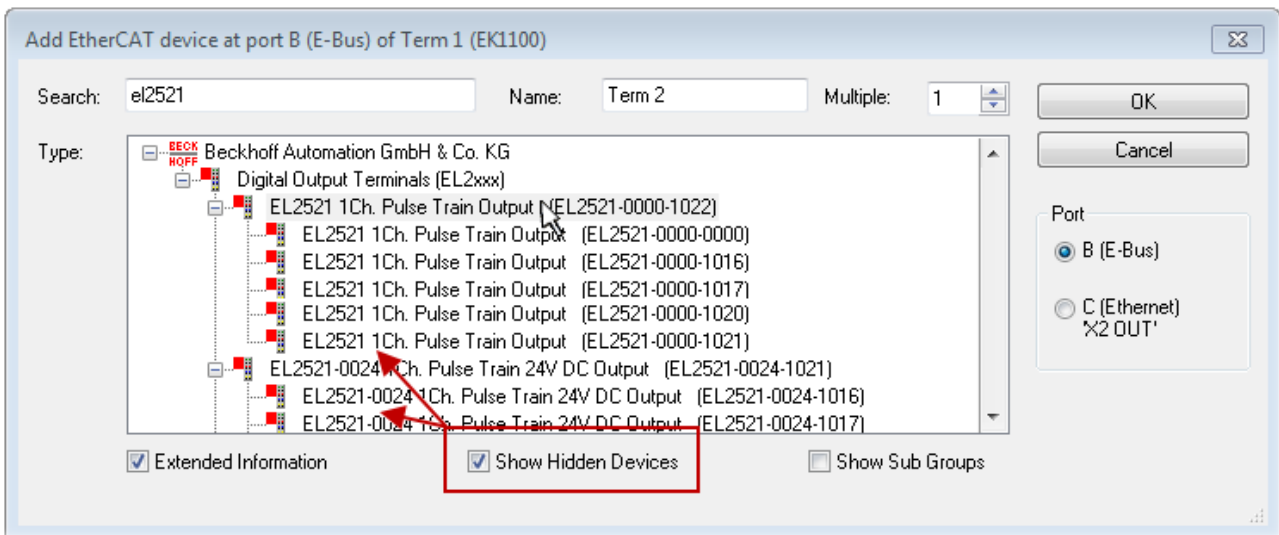


Fig. 67: Display of previous revisions

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.

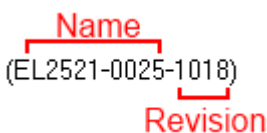


Fig. 68: Name/revision of the terminal

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

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

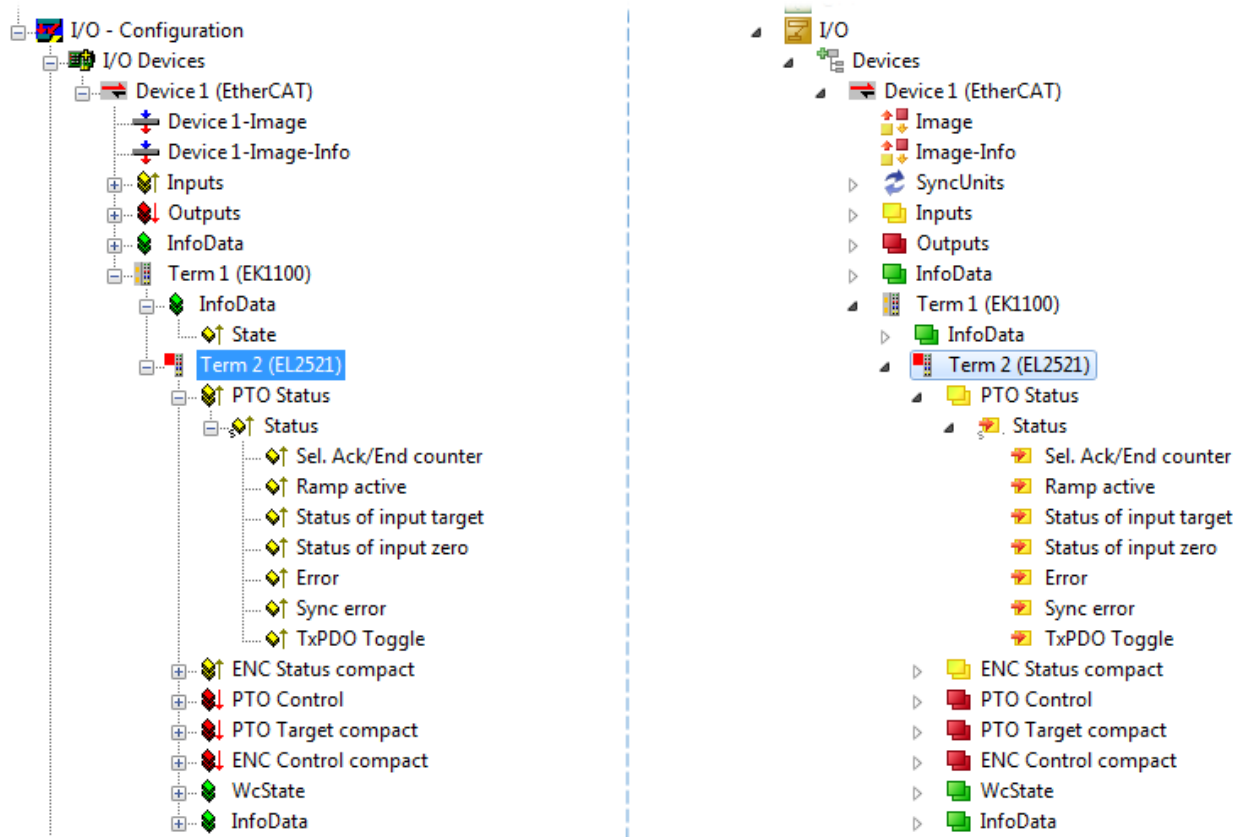




Fig. 69: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



5.1.4 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.

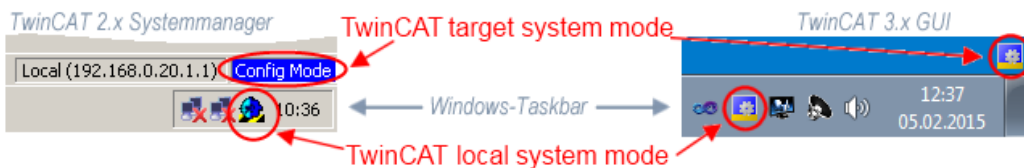


Fig. 70: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

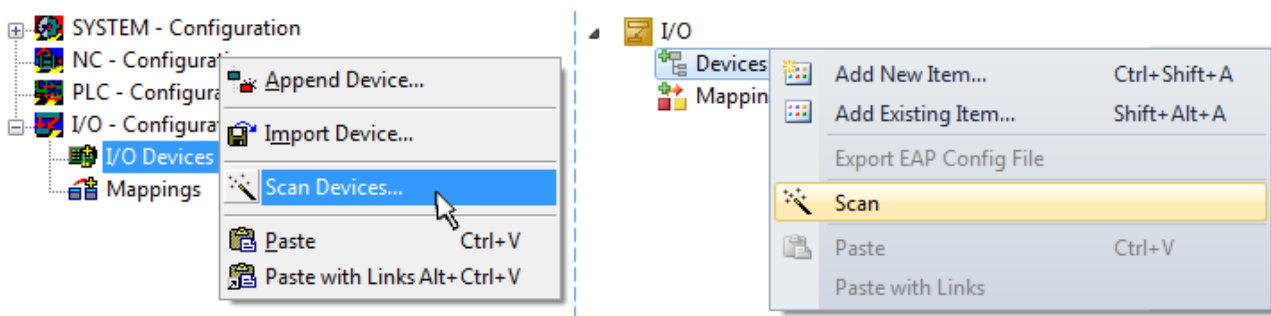


Fig. 71: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

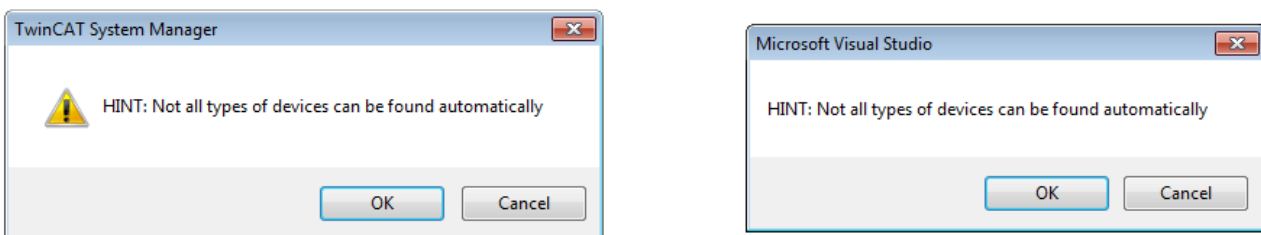


Fig. 72: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

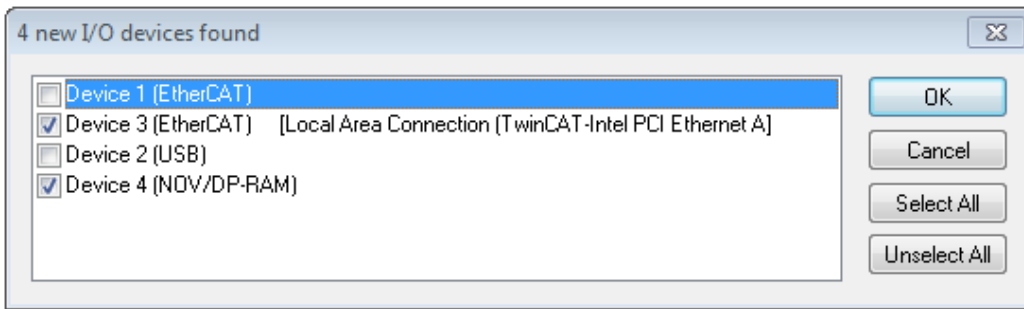


Fig. 73: Detected Ethernet devices

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](#) [▶ 67].

Detecting/Scanning the EtherCAT devices

● Online scan functionality



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

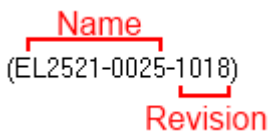


Fig. 74: Example default state

NOTICE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for [comparison](#) [▶ 87] 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:

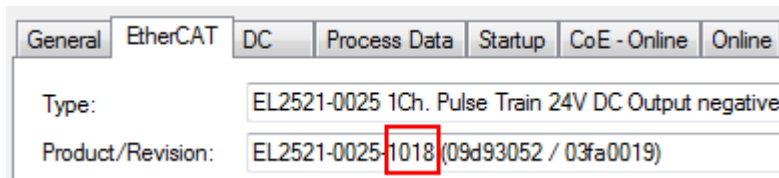


Fig. 75: Installing EthetCAT terminal with revision -1018

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

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

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and a **new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of “B.tsm” or even “B.pro” is therefore unnecessary. The series-produced machines can continue to be built with “B.tsm” and “B.pro”; it makes sense to perform a comparative scan [[▶ 87](#)] 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:

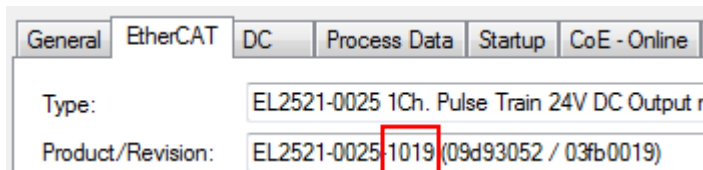


Fig. 76: Detection of EtherCAT terminal with revision -1019

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

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

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



Fig. 77: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

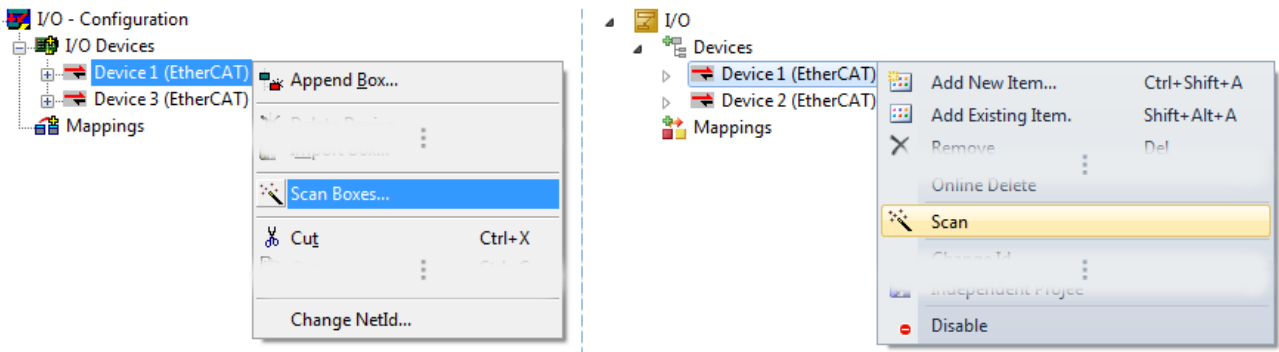


Fig. 78: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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



Fig. 79: Scan progress exemplary by TwinCAT 2

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



Fig. 80: 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. 81: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 82: 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*.

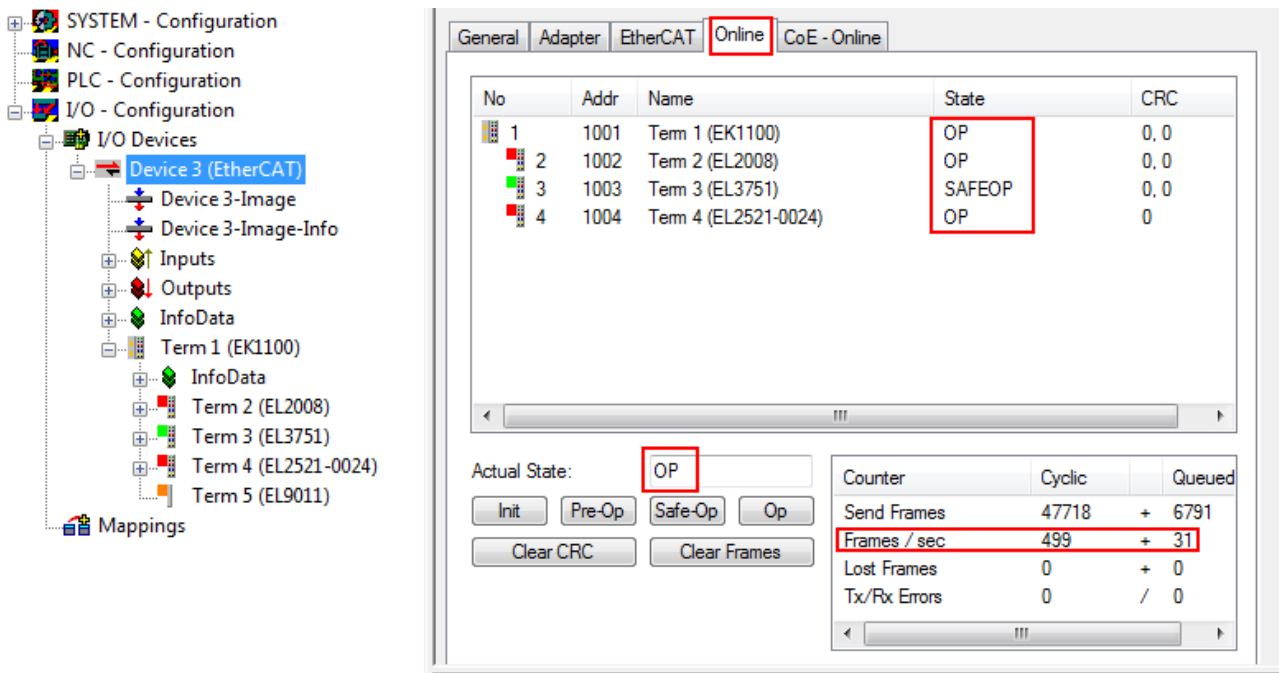


Fig. 83: Online display example

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 \[► 77\]](#).

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.

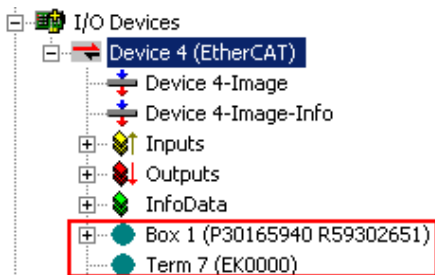


Fig. 84: Faulty identification

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

Scan over existing Configuration

NOTICE

Change of the configuration after comparison

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

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



Fig. 85: 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.

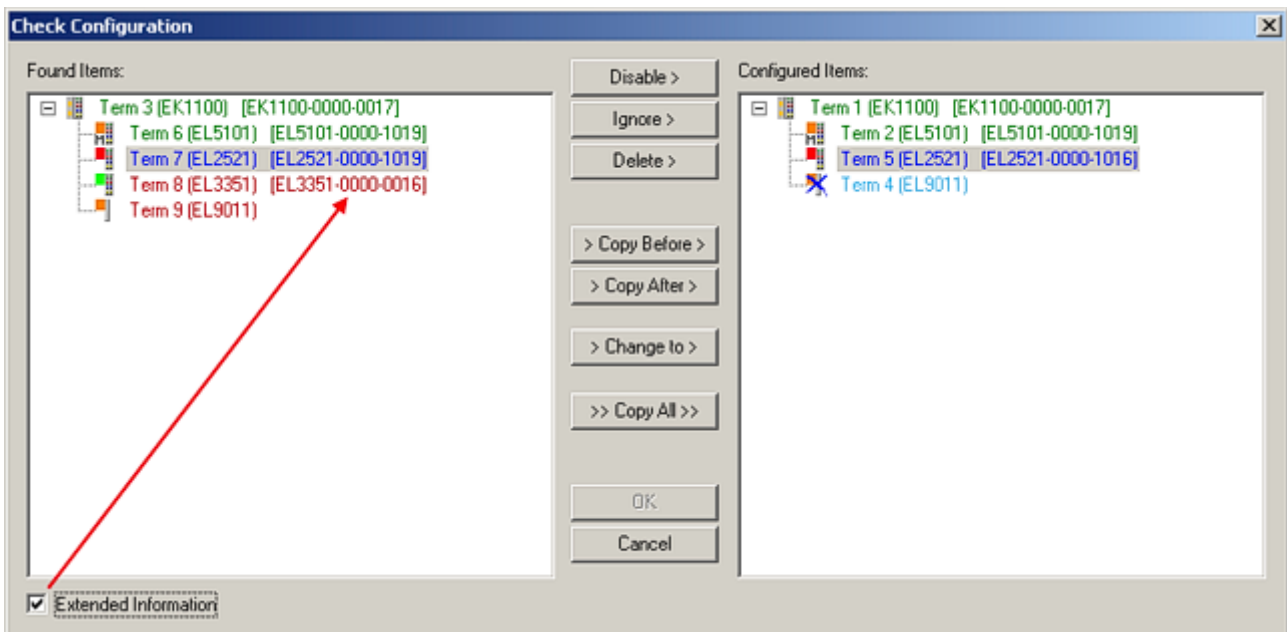


Fig. 86: Correction dialog

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

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored ("Ignore" button)
red	<ul style="list-style-type: none"> This EtherCAT slave is not present on the other side. It is present, but in a different revision, which also differs in its properties from the one specified. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

i Device selection based on revision, compatibility

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

device revision in the system \geq device revision in the configuration

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

Example

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

Name
(EL2521-0025-1018)
Revision

Fig. 87: Name/revision of the terminal

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

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

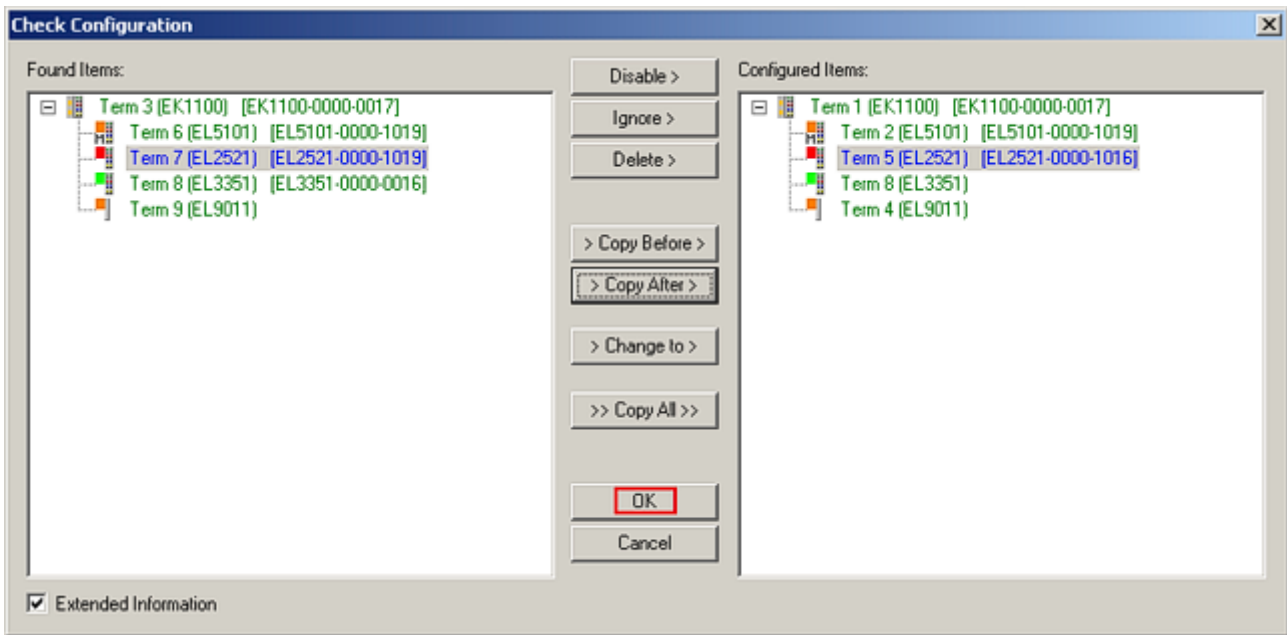


Fig. 88: Correction dialog with modifications

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.

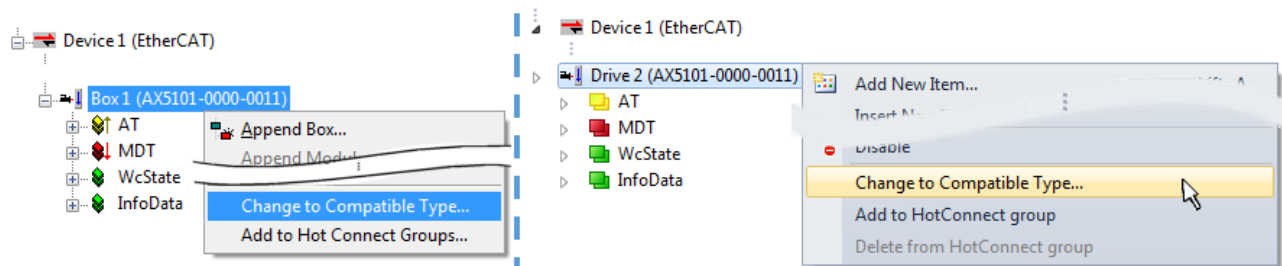


Fig. 89: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

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

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

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

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

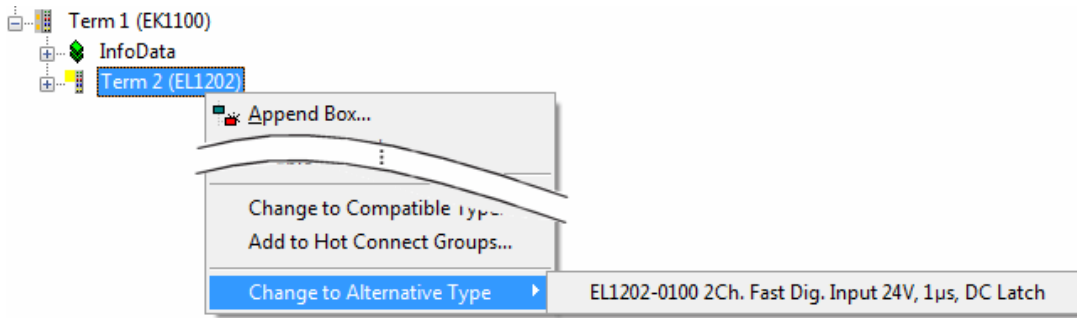


Fig. 90: TwinCAT 2 Dialog Change to Alternative Type

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).

5.1.5 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).

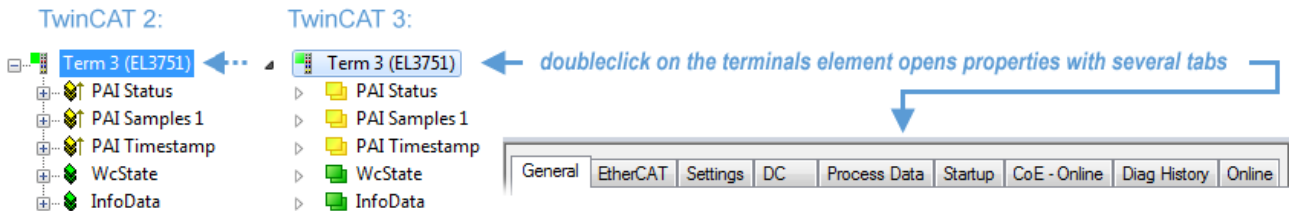


Fig. 91: Branch element as terminal EL3751

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

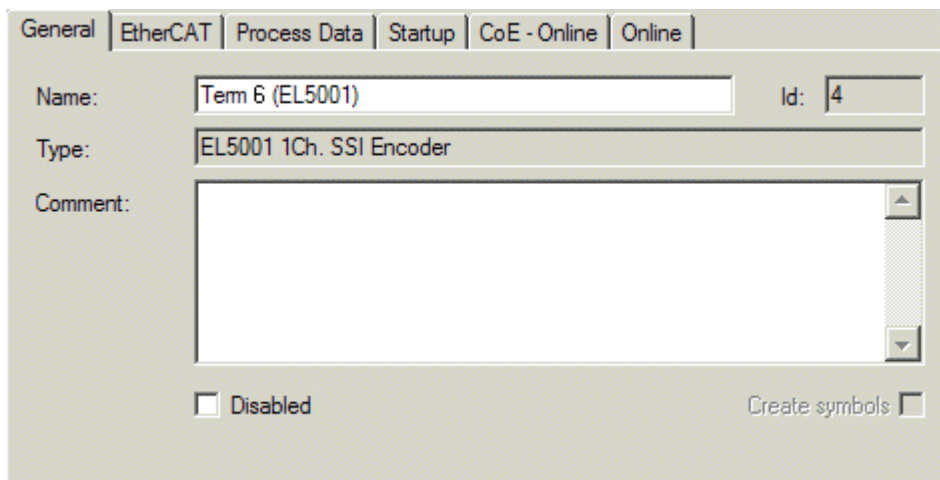


Fig. 92: “General” tab

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

“EtherCAT” tab

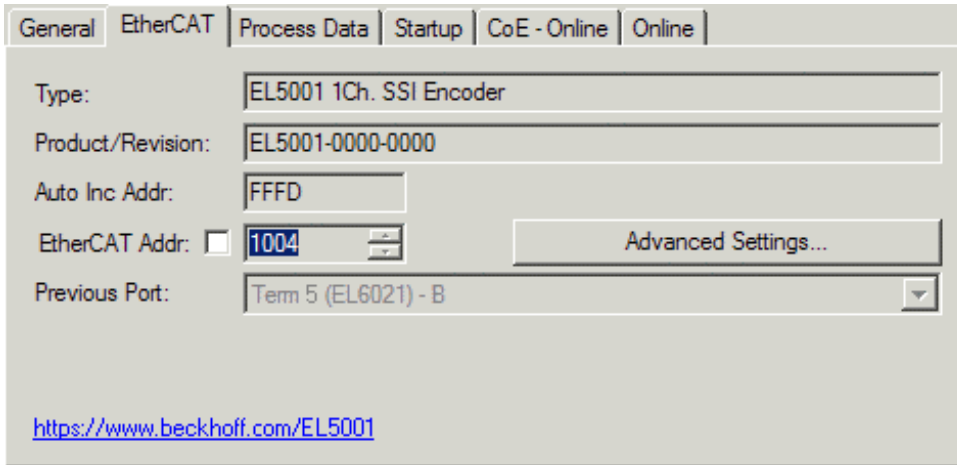


Fig. 93: “EtherCAT” tab

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

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

“Process Data” tab

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

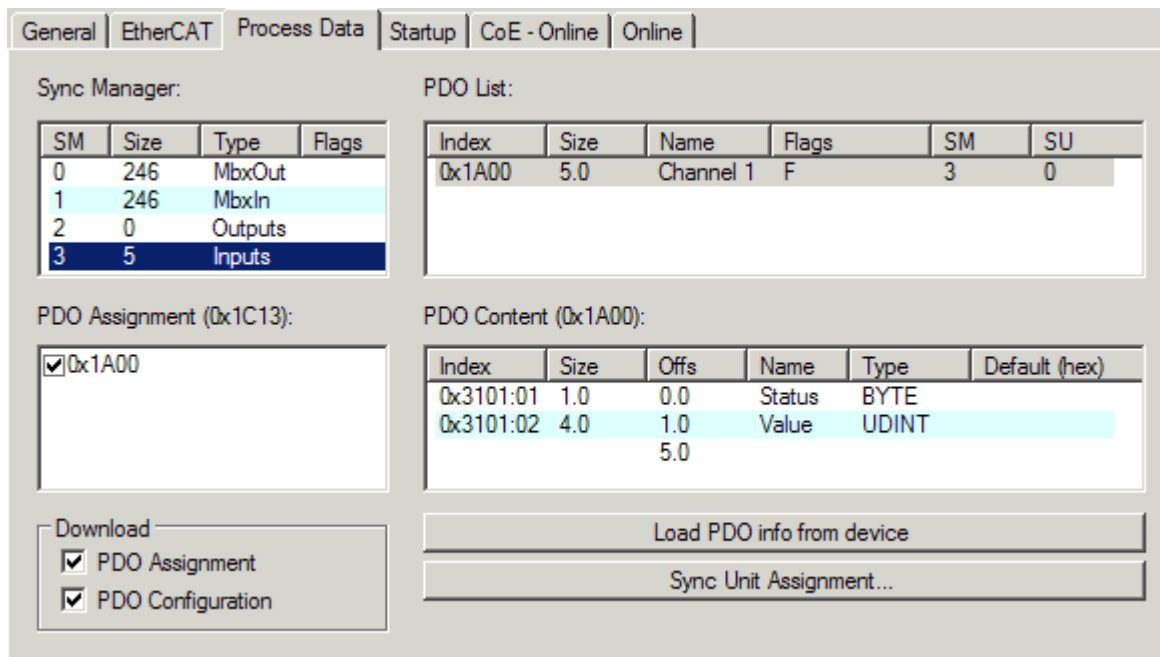


Fig. 94: "Process Data" tab

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").

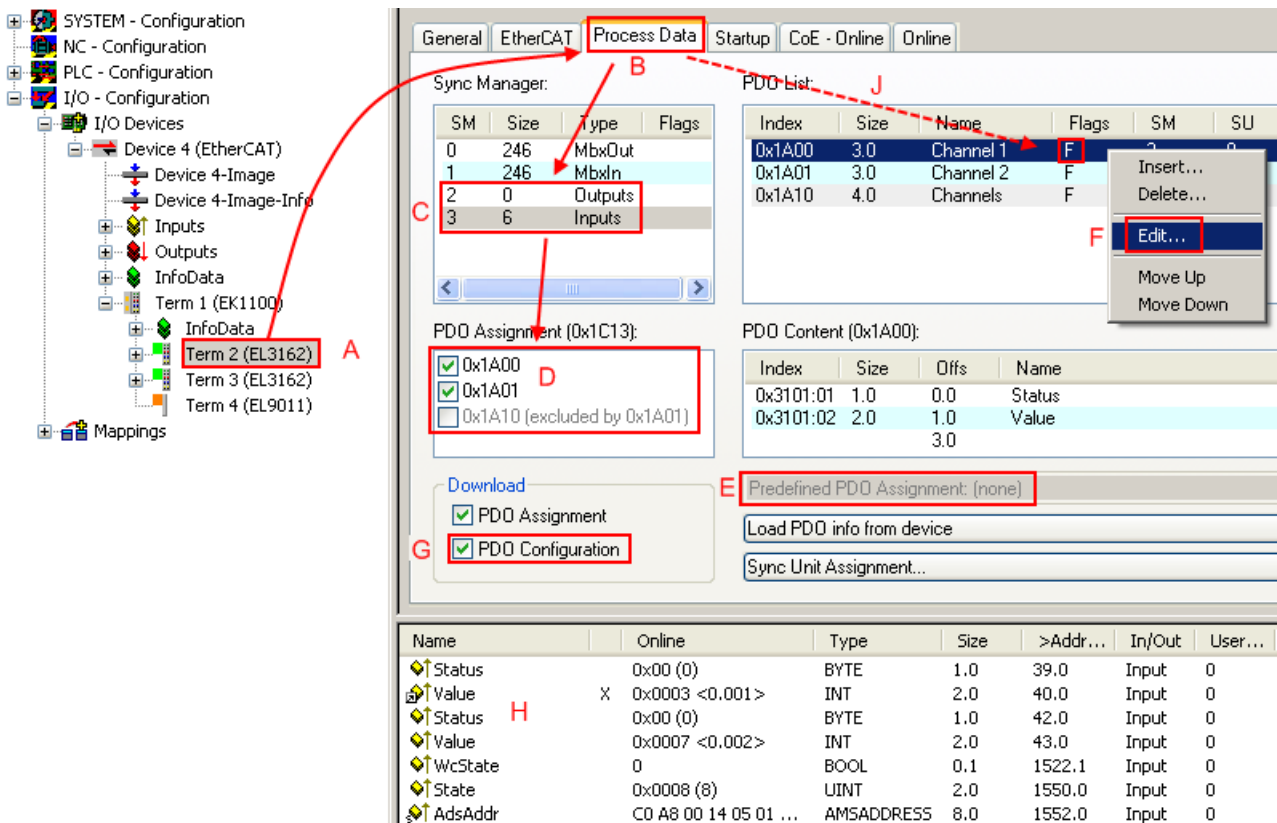


Fig. 95: Configuring the process data

i Manual modification of the process data

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

A [detailed description](#) [► 98] 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.

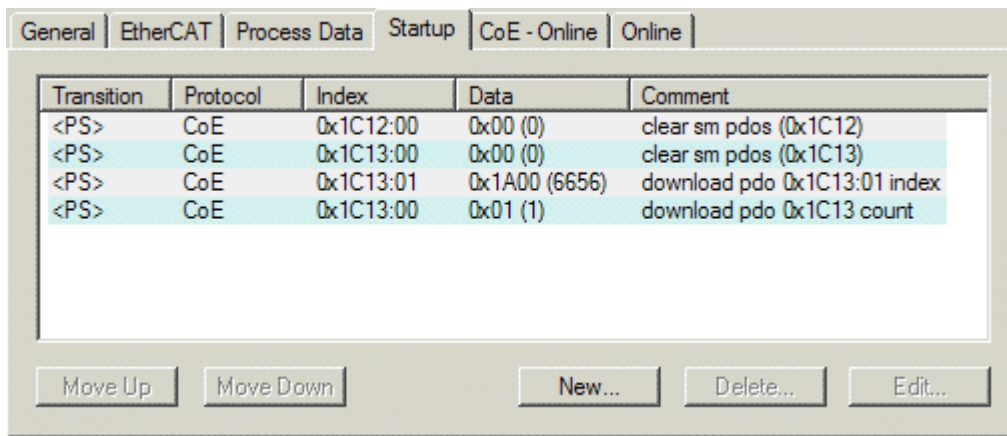


Fig. 96: "Startup" tab

Column	Description
Transition	Transition to which the request is sent. This can either be <ul style="list-style-type: none"> the transition from pre-operational to safe-operational (PS), or the transition from safe-operational to operational (SO). If the transition is enclosed in "<>" (e.g. <PS>), the mailbox request is fixed and cannot be modified or deleted by the user.
Protocol	Type of mailbox protocol
Index	Index of the object
Data	Date on which this object is to be downloaded.
Comment	Description of the request to be sent to the mailbox

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

"CoE - Online" tab

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

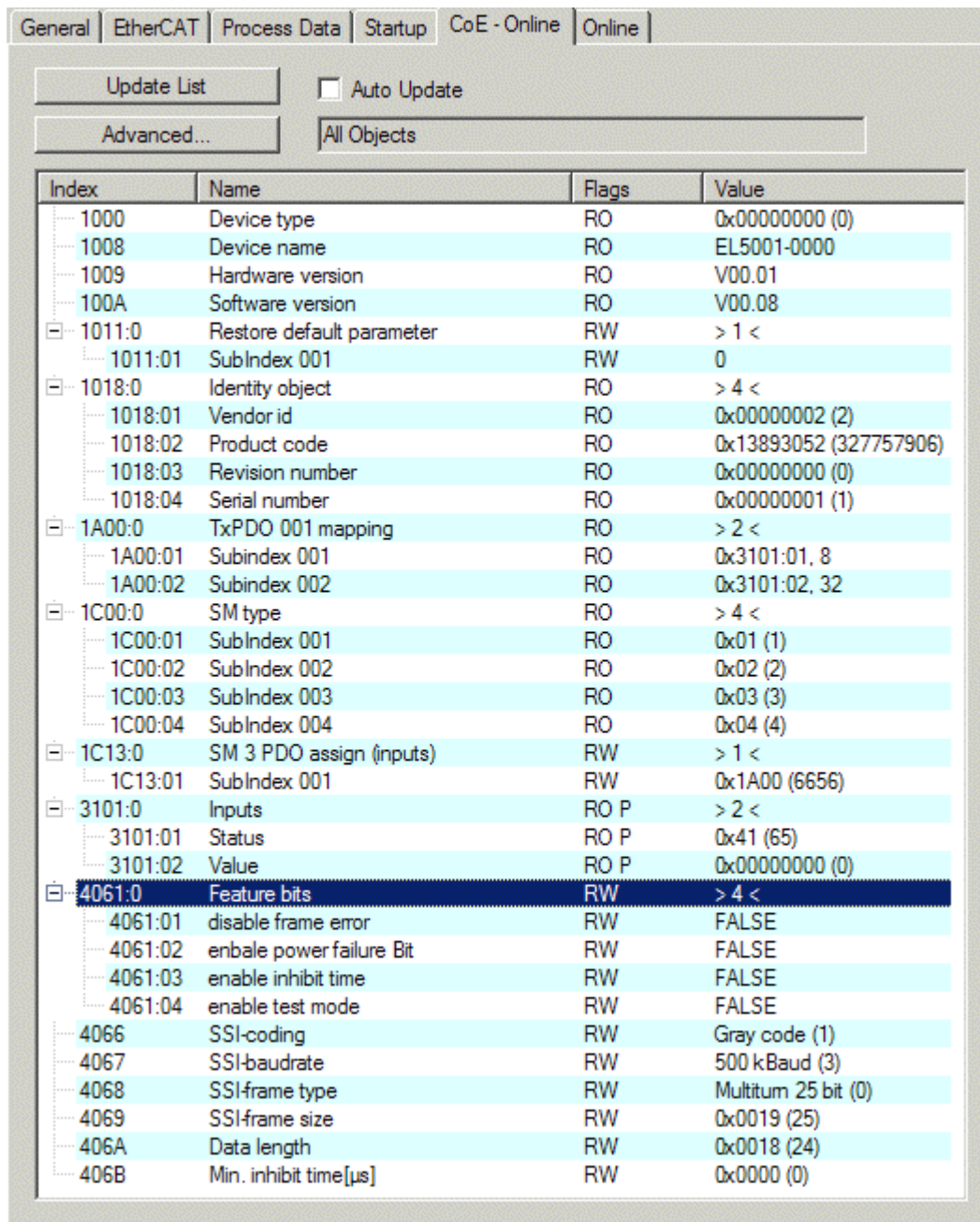


Fig. 97: "CoE - Online" tab

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.

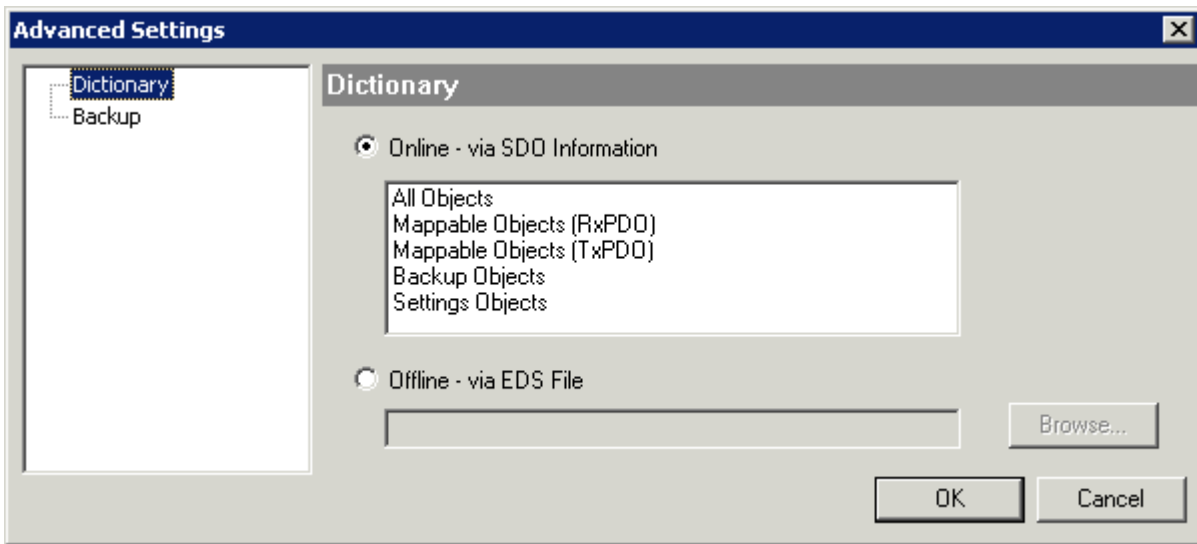


Fig. 98: Dialog “Advanced settings”

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

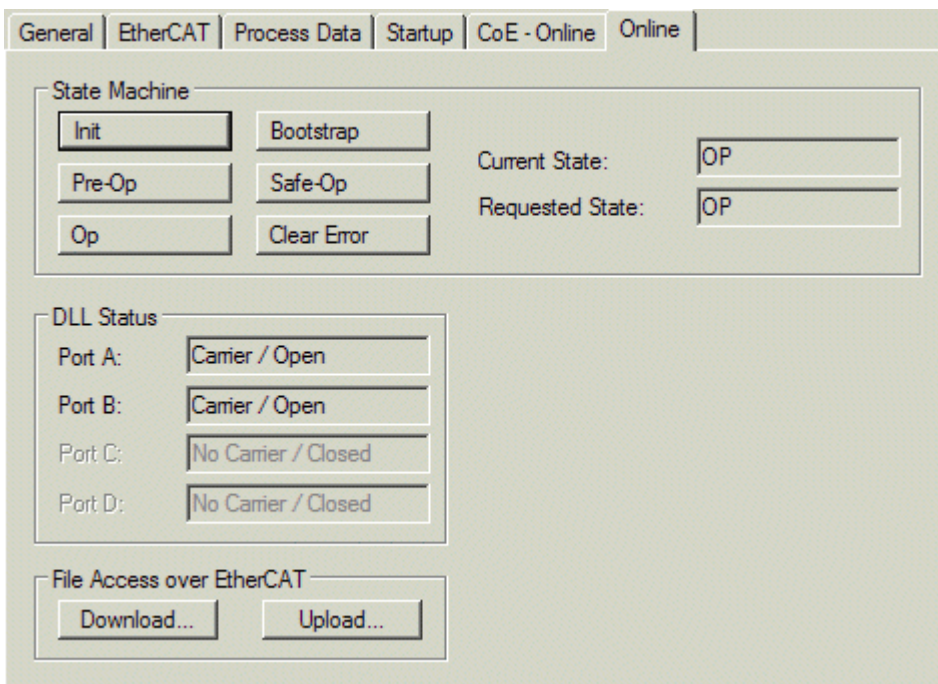


Fig. 99: “Online” tab

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)

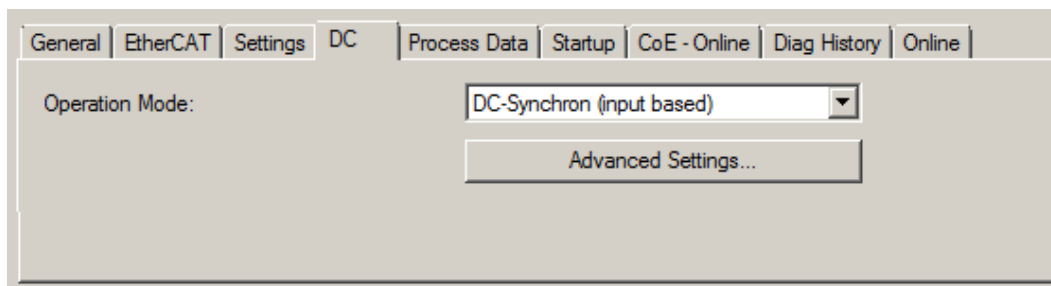


Fig. 100: “DC” tab (Distributed Clocks)

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

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

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

5.1.5.1 Detailed description of Process Data tab

Sync Manager

Lists the configuration of the Sync Manager (SM).

If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn).

SM2 is used for the output process data (outputs) and SM3 (inputs) for the input process data.

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

PDO Assignment

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

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

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

● **Activation of PDO assignment**



- ✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,
 - a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see [Online tab \[▶ 96\]](#)),
 - 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 [▶ 93] tab.

PDO Configuration

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

5.2 General Commissioning Instructions for an EtherCAT Slave

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

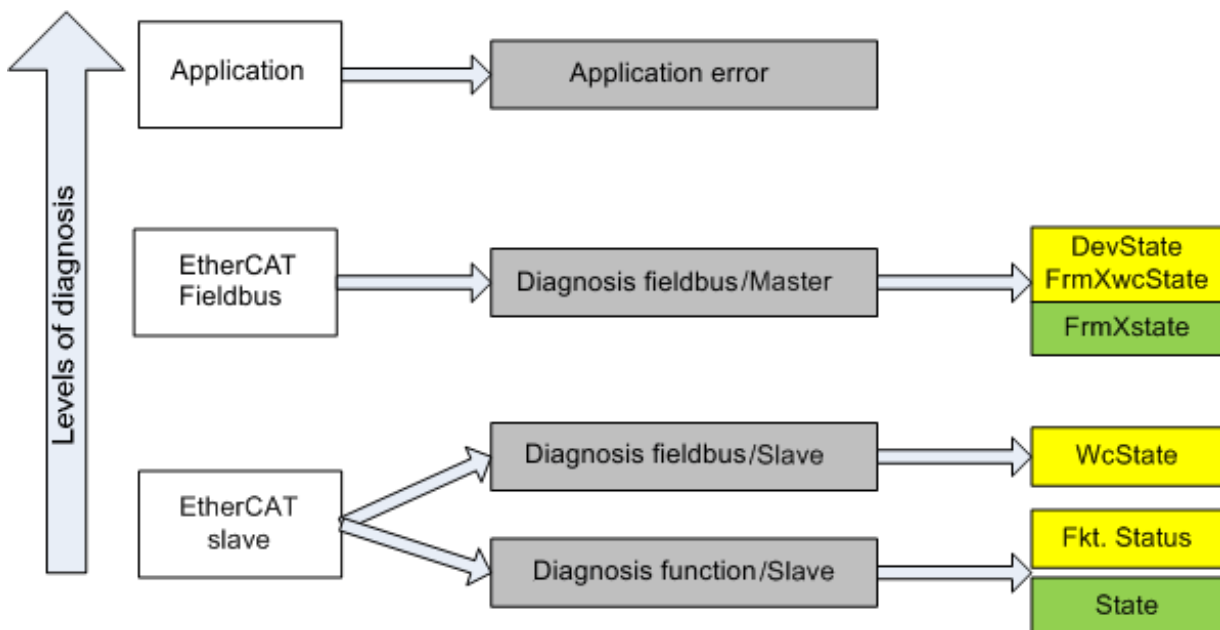


Fig. 101: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

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

as well as

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

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

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

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

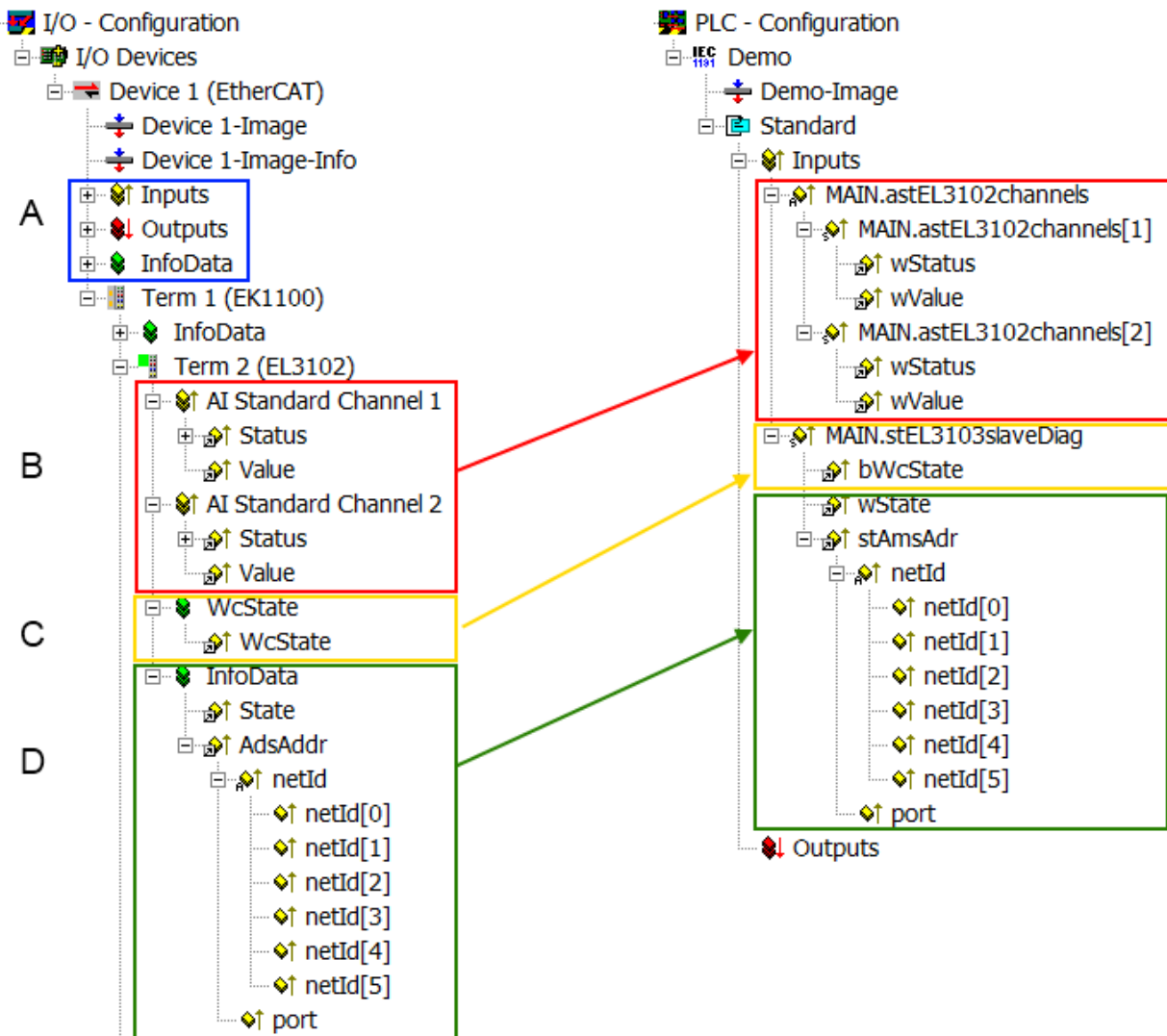


Fig. 102: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

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

NOTICE

Diagnostic information

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

CoE Parameter Directory

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

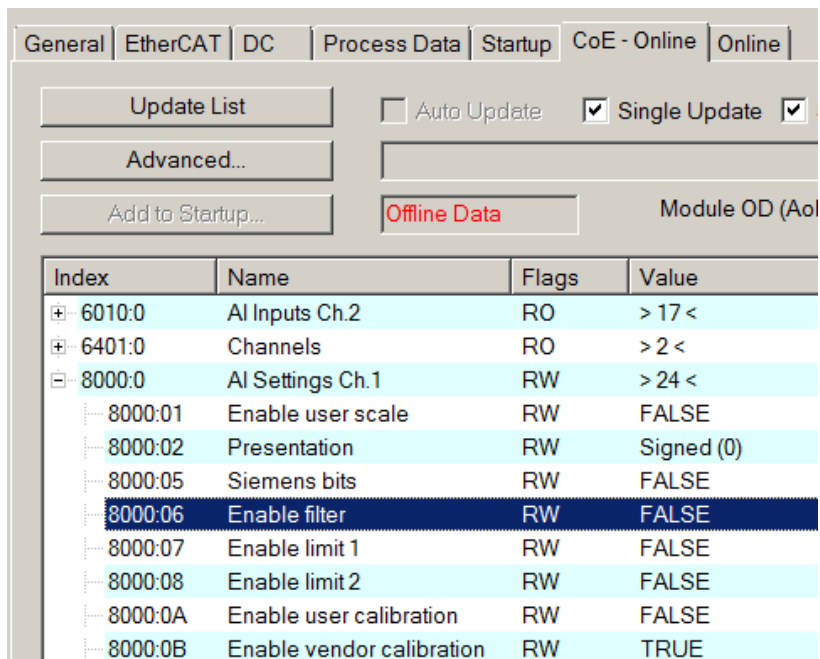


Fig. 103: EL3102, CoE directory

i EtherCAT System Documentation

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

A few brief extracts:

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

Commissioning aid in the TwinCAT System Manager

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

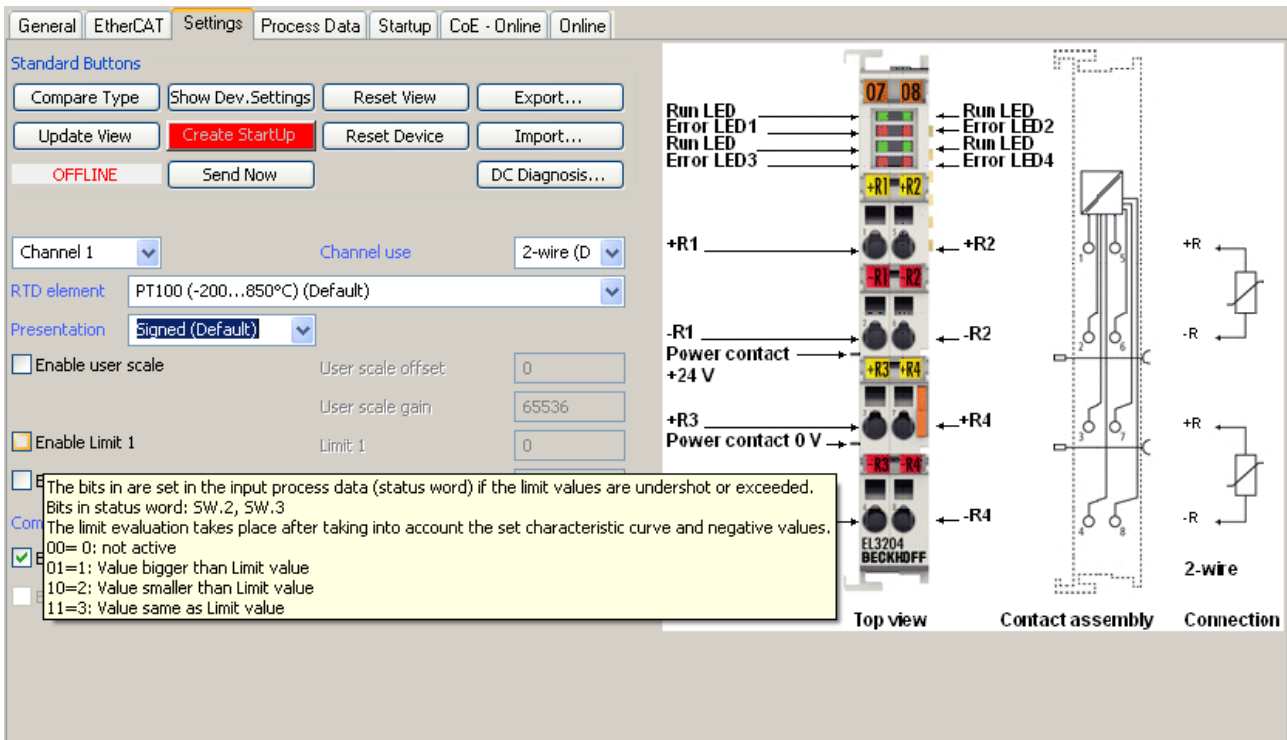


Fig. 104: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

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

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

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

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

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

- INIT
- PREOP
- SAFEOP
- OP

to ensure sound operation. The EtherCAT Master directs these statuses in accordance with the initialization routines that are defined for commissioning the device by the ES/XML and user settings (Distributed Clocks (DC), PDO, CoE). See also the section on "Principles of [Communication, EtherCAT State Machine \[▶ 44\]](#)" in this connection. Depending how much configuration has to be done, and on the overall communication, booting can take up to a few seconds.

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

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

Standard setting

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

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

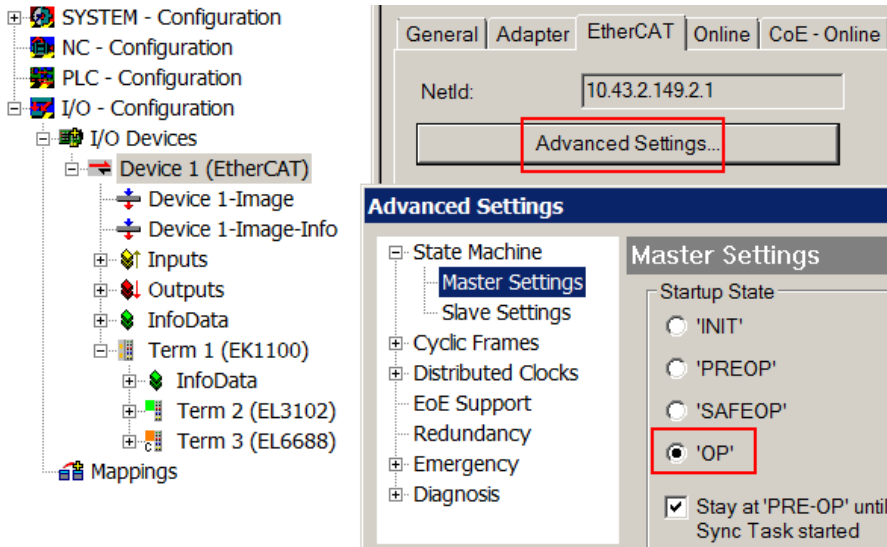


Fig. 105: Default behaviour of the System Manager

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

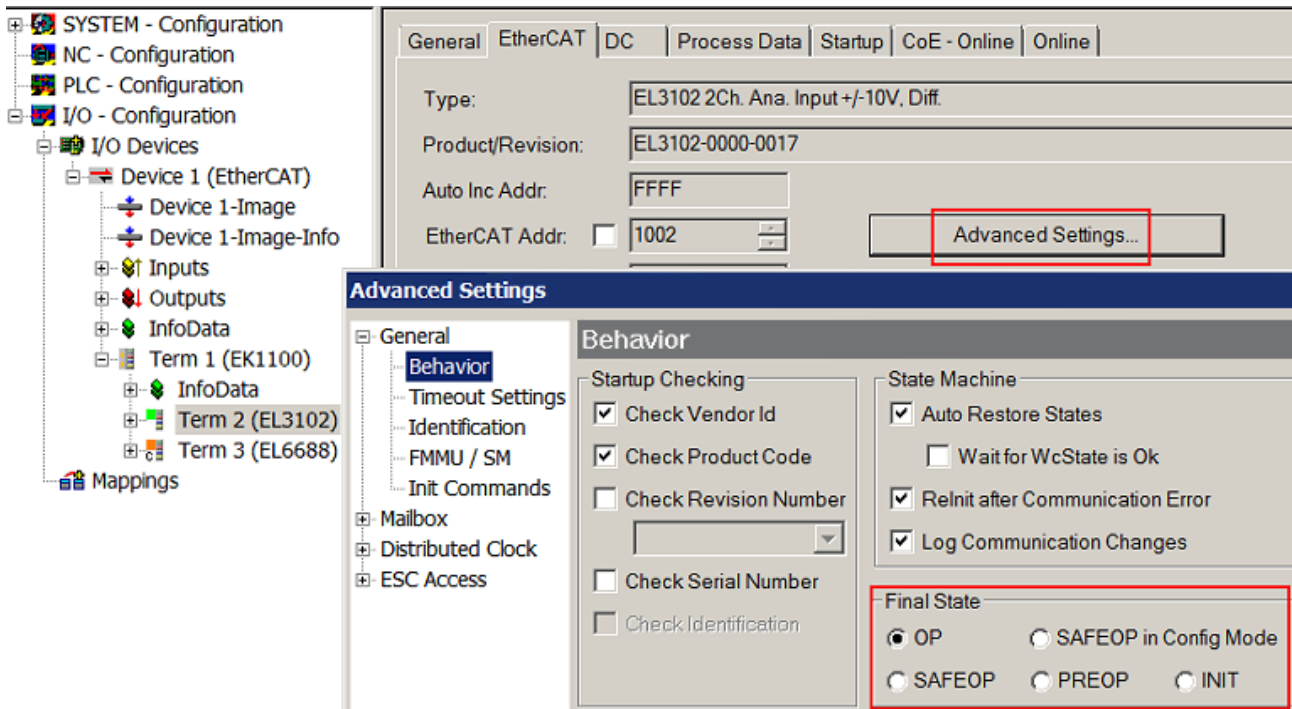


Fig. 106: Default target state in the Slave

Manual Control

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

- for diagnostic reasons
- to induce a controlled restart of axes

- because a change in the times involved in starting is desirable

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

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

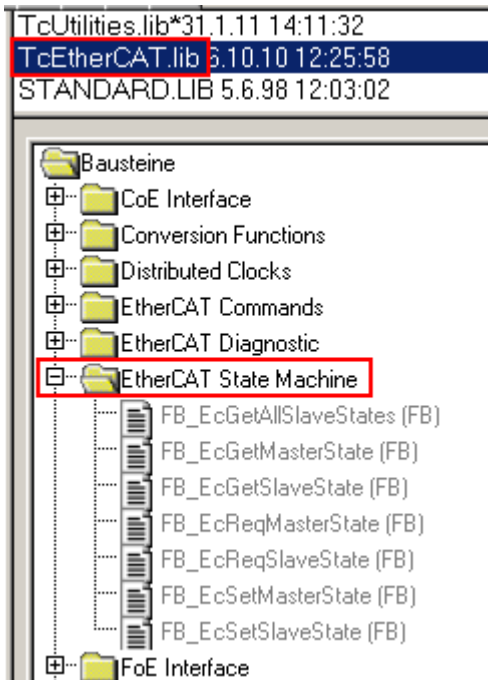


Fig. 107: PLC function blocks

Note regarding E-Bus current

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

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

General Adapter EtherCAT Online CoE - Online						
NetId:		10.43.2.149.2.1		Advanced Settings...		
Number	Box Name	Address	Type	In Size	Out S...	E-Bus (..
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL3102)	1002	EL3102	8.0		1830
3	Term 4 (EL2004)	1003	EL2004		0.4	1730
4	Term 5 (EL2004)	1004	EL2004		0.4	1630
5	Term 6 (EL7031)	1005	EL7031	8.0	8.0	1510
6	Term 7 (EL2808)	1006	EL2808		1.0	1400
7	Term 8 (EL3602)	1007	EL3602	12.0		1210
8	Term 9 (EL3602)	1008	EL3602	12.0		1020
9	Term 10 (EL3602)	1009	EL3602	12.0		830
10	Term 11 (EL3602)	1010	EL3602	12.0		640
11	Term 12 (EL3602)	1011	EL3602	12.0		450
12	Term 13 (EL3602)	1012	EL3602	12.0		260
13	Term 14 (EL3602)	1013	EL3602	12.0		70
14	Term 3 (EL6688)	1014	EL6688	22.0		-240 !

Fig. 108: Illegally exceeding the E-Bus current

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

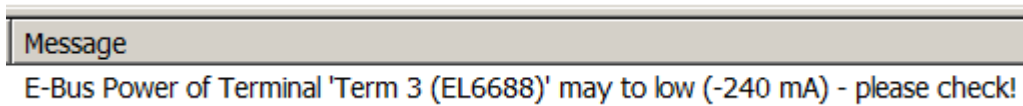


Fig. 109: Warning message for exceeding E-Bus current

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

5.3 Object description and parameterization

i EtherCAT XML Device Description

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

i Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE-Online tab [▶ 94] (double-click on the respective object) or via the Process Data tab [▶ 91](allocation of PDOs). Please note the following general CoE notes [▶ 46] when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary, existence of current XML description
- use “CoE reload [▶ 176]” for resetting changes

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization [▶ 107] during commissioning
- Objects intended for regular operation [▶ 108], e.g. through ADS access.
- Objects for indicating internal settings [▶ 108] (may be fixed)

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

5.3.1 Objects for commissioning

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4 _{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2 _{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x19C93052 (432615506 _{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the low word (bit 0-15) indicates the special terminal number, the high word (bit 16-31) refers to the device description	UINT32	RO	0x00100000 (1048576 _{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the low byte (bit 0-7) of the low word contains the year of production, the high byte (bit 8-15) of the low word contains the week of production, the high word (bit 16-31) is 0	UINT32	RO	0x00000000 (0 _{dec})

Index F800 EL6601 Para

Index (hex)	Name	Meaning	Data type	Flags	Default
F800:0	EL6601 Para	Max. subindex	UINT8	RW	0x02 (2 _{dec})
F800:01	General	0x0000: Standard operation 0x0001: VLAN TAGS are removed before filtering. 0x4000: EoE frames are blocked.	UINT16	RW	0x0000 (0 _{dec})
F800:02	NetVars	This switch determines whether received subscriber data from frames with 0x88A4 in the header, that have not passed the subscriber filter, will be transported further via EoE/Mailbox to the EtherCAT Master. 0x0000: default, subscriber data are forwarded via EoE 0x0100: Subscriber data are being discarded	UINT16	RW	0x0000 (0 _{dec})

5.3.2 Objects for regular operation

Index F100 Master Info

Index (hex)	Name	Meaning	Data type	Flags	Default
F100:0	Master Info	Max. subindex	UINT8	RO	0x02 (2 _{dec})
F100:01	Status	Link Status of the Ethernet port 0: Link 1: No link	UINT16	RO	0x0000 (0 _{dec})
F100:02	Control	reserved	UINT16	RO	0x0000 (0 _{dec})

Index FA01 MAC Info

Index (hex)	Name	Meaning	Data type	Flags	Default
FA01:0	MAC Info	Max. subindex	UINT8	RW	0x03 (3 _{dec})
FA01:01	RxPackets	Received Ethernet telegrams	UINT16	RW	0x0000 (0 _{dec})
FA01:02	TxPackets	Sent Ethernet telegrams	UINT16	RW	0x0000 (0 _{dec})
FA01:03	Reserved	Reserved	UINT16	RW	0x0000 (0 _{dec})

5.3.3 Standard objects (0x1000-0x1FFF)

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

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5002).	UINT32	RO	0x0000138A (5002 _{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	e.g. EL6601-0000-0017

Index 1009 Hardware version

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

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	00

Index 1600-16FE RxPDO-Map

Index (hex)	Name	Meaning	Data type	Flags	Default
1600+n:0	RxPDO-Map	PDO Mapping RxPDO (each module gets its own entry (Index 0x1600+n), 0 ≤ n < maximum number of modules)	UINT8	RW	0x02 (2 _{dec})
(1600+n):01	Output Mapping Area 001	1. PDO Mapping Entry (Object 7000+n*8:07)	UINT32	RW	0x7000+n*8:07, 16
(1600+n):02	Output Mapping Area 002	2. PDO Mapping Entry (Object 7000+n*8:0B)	UINT32	RW	0x7000+n*8:0B, 16

Index 1680 PDO control

Index (hex)	Name	Meaning	Data type	Flags	Default
1680:0	PDO control	Max. subindex	UINT8	RW	0x01 (1 _{dec})
1680:01	PDO control	Entry Master Control	UINT32	RW	0xF100:02, 16

Index 1A00-1AFE TxPDO-Map

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00+n:0	TxPDO Map Ch.2	PDO Mapping TxPDO (each module gets its own entry (Index 0x1A00+n), 0 ≤ n < maximum number of modules)	UINT8	RW	0x03 (3 _{dec})
(1A00+n):01	Input Mapping Area 001	1. PDO Mapping Entry (Object 0x6000+n*8:03)	UINT32	RW	0x6000+n*8:03, 16
(1A00+n):02	Input Mapping Area 002	2. PDO Mapping Entry (Object 0x6000+n*8:04)	UINT32	RW	0x6000+n*8:04, 16
(1A00+n):03	Input Mapping Area 003	3. PDO Mapping Entry (Object 0x6000+n*8:05)	UINT32	RW	0x6000+n*8:05, 16

Index 1A80 PDO status

Index (hex)	Name	Meaning	Data type	Flags	Default
1680:0	PDO status	Max. subindex	UINT8	RW	0x01 (1 _{dec})
1680:01	PDO status	Entry Master Status	UINT32	RW	0xF100:01, 16

Index 1C00 Sync manager type

Index (hex)	Name	Meaning	Data type	Flags	Default
1C00:0	Sync manager type	Using the sync managers	UINT8	RO	0x02 (2 _{dec})
1C00:01	SubIndex 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1 _{dec})
1C00:02	SubIndex 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2 _{dec})
1C00:03	SubIndex 003	Sync-Manager Type Channel 3: Outputs	UINT8	RO	0x02 (2 _{dec})
1C00:04	SubIndex 004	Sync-Manager Type Channel 4: Inputs	UINT8	RO	0x02 (2 _{dec})

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	-
1C32:01	SubIndex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT32	RW	0x1600 (5632 _{dec})
...					
1C12:80	SubIndex 128	128. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT32	RW	0x167F (5759 _{dec})
1C12:81	SubIndex 129	PDO Control	UINT32	RW	0x1680 (5760 _{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	-
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT32	RW	0x1A00 (6656 _{dec})
...					
1C13:80	SubIndex 128	128. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT32	RW	0x1A7F (6783 _{dec})
1C13:81	SubIndex 129	PDO status	UINT32	RW	0x1A80 (6784 _{dec})

Index 1C32 SM output parameter (only with network variables)

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RW	0x0E (14 _{dec})
1C32:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> 1: Synchronous with SM 2 event 	UINT16	RW	0x0001 (1 _{dec})
1C32:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> Free Run: Cycle time of the local timer Synchronous with SM 2 event: Master cycle time DC mode: SYNC0/SYNC1 Cycle Time 	UINT32	RW	0x00000000 (0 _{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RW	0x00000000 (0 _{dec})
1C32:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> Bit 1 = 1: Synchron with SM 2 event is supported 	UINT16	RW	0x0002 (2 _{dec})

Index 1C33 SM input parameter (only with network variables)

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RW	0x0E (14 _{dec})
1C33:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> 34: Synchron with SM 2 Event (outputs available) 	UINT16	RW	0x0022 (34 _{dec})
1C33:02	Cycle time	as 0x1C32:02 [▶ 110]	UINT32	RW	0x00000000 (0 _{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RW	0x00000000 (0 _{dec})
1C33:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> Bit 1: Synchronous with SM 2 Event is supported (outputs available) 	UINT16	RW	0x0002 (2 _{dec})

5.3.4 Profile-specific objects (0x6000-0xFFFF)

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

Index 6000-67F8 Receiving Frame Data (Net Var Subscriber)

Index (hex)	Name	Meaning	Data type	Flags	Default
6000+n*8:0	Receiving Frame Data (Net Var Subscriber)	Max. subindex (each module gets its own entry (Index 0x6000+n*8), 0 ≤ n < maximum number of modules)	UINT8	RW P	0x05 (5 _{dec})
(6000+n*8):01	Net Id	Source AMS Net Id	UINT48	RW P	-
(6000+n*8):02	Var Id	Identification of network variables	UINT16	RW P	0x0001 (1 _{dec})
(6000+n*8):03	Quality	Period over which the variable was not updated (resolution 100 μs)	UINT16	RO P	0x0000 (0 _{dec})
(6000+n*8):04	Cycle Index	Entry is incremented with each Publisher cycle	UINT16	RO P	0x0000 (0 _{dec})
(6000+n*8):05	Data area 001	Data range	OCTED_STRING	RO P	00 00

Index 6001-67F9 Sending frame State (Frame status)

Index (hex)	Name	Meaning	Data type	Flags	Default
6001+n*8:0	Sending frame State (Frame status)	Max. subindex (each module gets its own entry (Index 0x6001+n*8), 0 ≤ n < maximum number of modules)	UINT8	RW P	0x01 (1 _{dec})
(6001+n*8):01	Frame status	Status <ul style="list-style-type: none"> • Bit 0: Frame bypassed • Bit 1: Frame too large 	UINT16	RW P	0x0000 (0 _{dec})

Index 6002-67FA Receiving Frame Identification (Ignore Item Net Var Subscriber)

Index (hex)	Name	Meaning	Data type	Flags	Default
6002+n*8:0	Receiving Frame Identification (Ignore Item Net Var Subscriber)	Max. subindex (each module gets its own entry (Index 0x6002+n*8), 0 ≤ n < maximum number of modules)	UINT8	RW P	0x05 (5 _{dec})
(6002+n*8):01	Net Id	"Net Id"	UINT8	RW P	0x01 (1 _{dec})
(6002+n*8):02	Var Id	"Var Id"	UINT8	RW P	0x00 (0 _{dec})
(6000+n*8):03	Quality	"Quality"	UINT8	RW P	0x01 (1 _{dec})
(6002+n*8):04	Cycle Index	"Cycle Index"	UINT8	RW P	0x01 (1 _{dec})
(6002+n*8):05	Data area 001	"Data area 001"	UINT8	RW P	0x01 (1 _{dec})

Index 6003-67FB Receiving Frame Length (Area Length Nat Var Subscriber)

Index (hex)	Name	Meaning	Data type	Flags	Default
6003+n*8:0	Receiving Frame Length (Area Length Nat Var Subscriber)	Max. subindex (each module gets its own entry (Index 0x6003+n*8), 0 ≤ n < maximum number of modules)	UINT8	RW P	0x05 (5 _{dec})
(6003+n*8):01	Net Id	Length of the "Net Id" field	UINT16	RW P	0x0006 (6 _{dec})
(6003+n*8):02	Var Id	Length of the "Var Id" field	UINT16	RW P	0x0002 (2 _{dec})
(6003+n*8):03	Quality	Length of the "Quality" field	UINT16	RW P	0x0002 (2 _{dec})
(6003+n*8):04	Cycle Index	Length of the "Cycle Index" field	UINT16	RW P	0x0002 (2 _{dec})
(6003+n*8):05	Data area 001	Length of the "Data area" field	UINT16	RW P	0x0002 (2 _{dec})

Index 7000-77F8 Sending Frame Data (Net Var Publisher)

Index (hex)	Name	Meaning	Data type	Flags	Default
7000+n*8:0	Sending Frame Data (Net Var Publisher)	Max. subindex (each module gets its own entry (Index 0x7000+n*8), $0 \leq n < \text{maximum number of modules}$)	UINT8	RW P	-
(7000+n*8):01	Destination MAC address	MAC-Destination address Ethernet telegram	UINT48	RW P	-
(7000+n*8):02	Source MAC address	MAC-Source address Ethernet telegram	UINT48	RW P	-
(7000+n*8):03	Ethernet type	Beckhoff Ethertype	UINT16	RW P	0x88A4 (42120 _{dec})
(7000+n*8):04	Header	Bit 0-10: Length of the following entries Bit 11: 0, Bit 12-15: 4, Network variable -type	UINT16	RW P	-
(7000+n*8):05	Net Id	Source AMS Net Id	UINT48	RW P	-
(7000+n*8):06	# of Vars	Number of variables	UINT16	RW P	-
(7000+n*8):07	Cycle Index	Entry is incremented with each Publisher cycle	UINT16	RO P	-
(7000+n*8):08	reserved	reserved	UINT16	RW P	-
(7000+n*8):09	Net Var 001 Id	Identification of network variables	UINT16	RW P	-
(7000+n*8):0A	Net Var 001 Header	Byte 0,1: Hash value Byte 2,3: Length of the data Byte 4,5: Quality	UINT48	RW P	-
(7000+n*8):0B	Net Var 001 Data	Data range	STRING	RO P	-
-					
(7000+n*8):(3*y+6)	Net Var y Id	Identification of network variables	UINT16	RW P	-
(7000+n*8):(3*y+7)	Net Var y Header	Byte 0,1: Hash value Byte 2,3: Length of the data Byte 4,5: Quality	UINT48	RW P	-
(7000+n*8):(3*y+8)	Net Var y Data	Data range	OCTED_STRING	RO P	-

Index 7001-77F9 Sending Frame Control (Frame control)

Index (hex)	Name	Meaning	Data type	Flags	Default
7001+n*8:0	Sending Frame Control (Frame control)	Max. subindex (each module gets its own entry (Index 0x7001+n*8), $0 \leq n < \text{maximum number of modules}$)	UINT8	RW P	0x01 (1 _{dec})
(7001+n*8):01	Frame Control	Frame Control • Bit 0 = 0: Send Frame • Bit 0 = 1: Skip Frame	UINT8	RW P	0x00 (0 _{dec})

Index 7003-77FB Sending Frame Length (Area length Net Var Publisher)

Index (hex)	Name	Meaning	Data type	Flags	Default
7003+n*8:0	Sending Frame Length (Area length Net Var Publisher)	Max. subindex (each module gets its own entry (Index 0x7003+n*8), 0 ≤ n < maximum number of modules)	UINT8	RW P	-
(7003+n*8):01	Destination MAC address	MAC-Destination address Ethernet telegram	UINT48	RW P	-
(7003+n*8):02	Source MAC address	MAC-Source address Ethernet telegram	UINT48	RW P	-
(7003+n*8):03	Ethernet type	Beckhoff Ethertype	UINT16	RW P	0x88A4 (42120 _{dec})
(7003+n*8):04	Header	Bit 0-10: Length of the following entries Bit 11: 0, Bit 12-15: 4, Network variable -type	UINT16	RW P	-
(7003+n*8):05	Net Id	Source AMS Net Id	UINT48	RW P	-
(7003+n*8):06	# of Vars	Number of variables	UINT16	RW P	-
(7003+n*8):07	Cycle Index	Entry is incremented with each Publisher cycle	UINT16	RW P	-
(7003+n*8):08	reserved	reserved	UINT16	RW P	-
(7003+n*8):09	Net Var 001 Id	Identification of network variables	UINT16	RW P	-
(7003+n*8):0A	Net Var 001 Header	Byte 0,1: Hash value Byte 2,3: Length of the data Byte 4,5: Quality	UINT48	RW P	-
(7003+n*8):0B	Net Var 001 Data	Data range	STRING	RW P	-
-					
(7003+n*8):(3*y+6)	Net Var y Id	Identification of network variables	UINT16	RW P	-
(7003+n*8):(3*y+7)	Net Var y Header	Byte 0,1: Hash value Byte 2,3: Length of the data Byte 4,5: Quality	UINT48	RW P	-
(7003+n*8):(3*y+8)	Net Var y Data	Data range	OCTED_STRING	RW P	-

Index 8000-87F8 Frame Config

Index (hex)	Name	Meaning	Data type	Flags	Default
8000+n*8:0	Frame Config	Max. SubIndex	UINT8	RW	0x04 (4 _{dec})
(8000+n*8):04	Device type	3: Subscriber network variable, corresponding indexes 600x are being created. 4: Publisher network variable, corresponding indexes 0x700x are being created.	UINT16	RW	-

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x04 (4 _{dec})
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0008 (8 _{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x00FF (255 _{dec})
F000:03	Standard Entries in Object 0x8yy0	Standard Entries in the Objects 0x8yy0	UINT32	RO	0x00000000 (0 _{dec})
F000:04	Standard Entries in Object 0x9yy0	Standard Entries in the Objects 0x8yy0	UINT32	RO	0x00000000 (0 _{dec})

5.4 Beckhoff network variables

5.4.1 Introduction

Network variables are any variables that are cyclically exchanged between PC/CX1000 via TwinCAT. Variables with any data types, including complex types, can be exchanged. The Publisher/Subscriber model is used. For highly deterministic communication, the real-time Ethernet driver for TwinCAT must be installed.

Publisher/Subscriber model

In the Publisher/Subscriber model, the Publisher makes variables available. Subscribers can subscribe to a variable. The Publisher can make the variable available to a Subscriber, several Subscribers or all Subscribers. In Broadcast mode the variable is made available to all PCs, in Multicast mode to selected PCs and in Unicast mode only to one selected PC. A Subscriber can also be Publisher at the same time. In this way, a bidirectional data link can be provided.

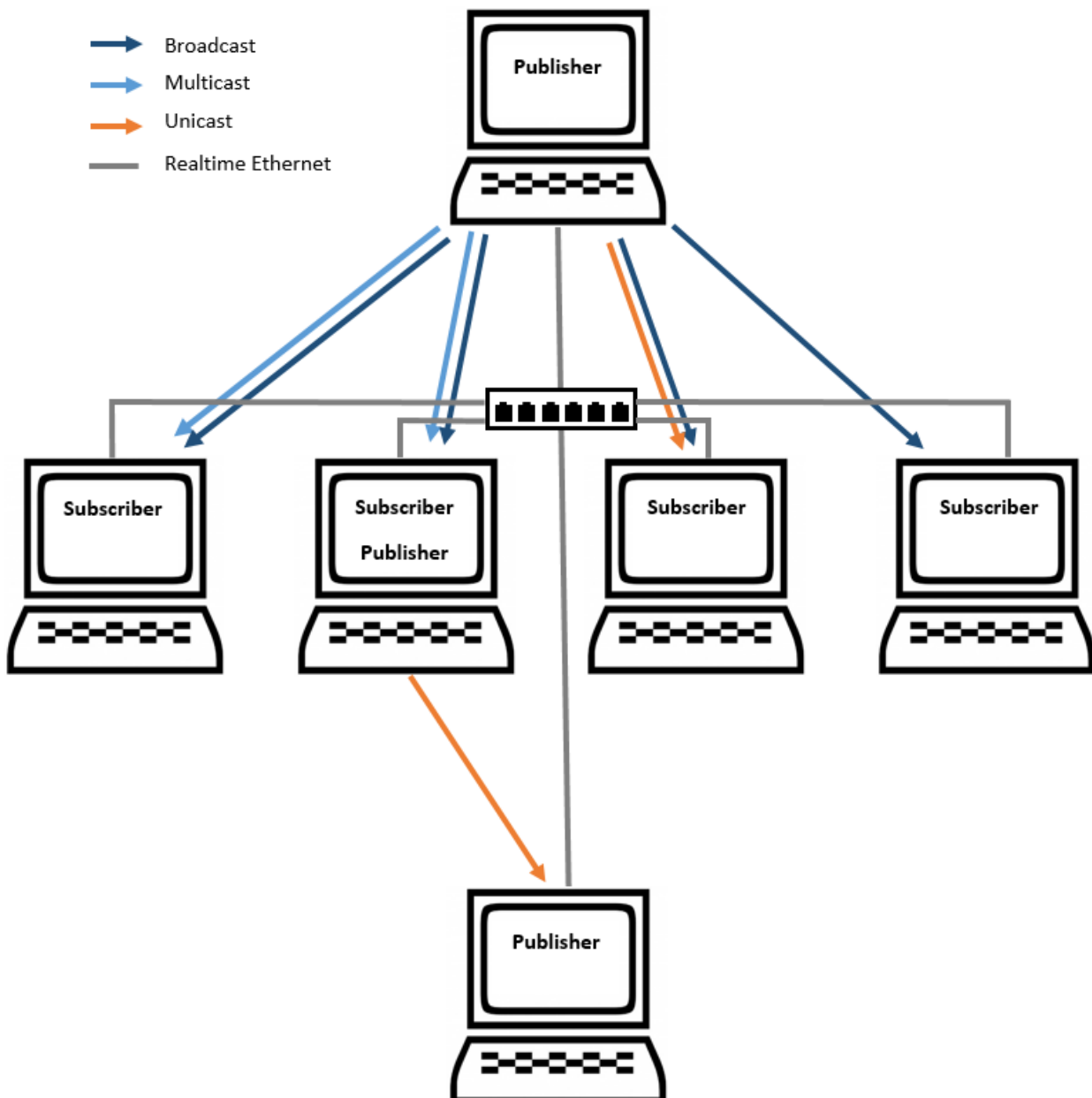


Fig. 110: Publisher/Subscriber model

Unicast

The Publisher makes the network variable available to a single selected PC.

Multicast

The Publisher makes the network variable available to selected PCs.

Broadcast

The Publisher makes the network variable available to all PCs.

5.4.2 Configuration of the Publisher

In the TwinCAT System Manager, a new box is added for the Publisher under the RT Ethernet device.

Insert a Publisher Box

A Publisher box must be added under the RT Ethernet device.

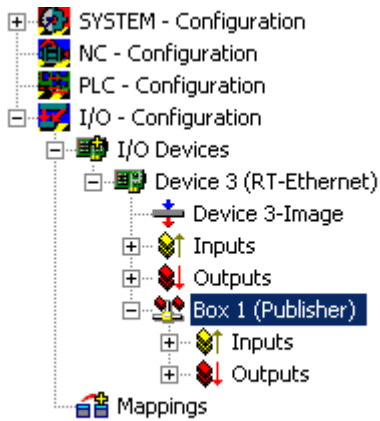


Fig. 111: Insertion of a Publisher Box in the TwinCAT configuration

Insert a Network Variable

The network variables are inserted underneath the box. Enter a name (nCounterPub in the sample) and a data type (UINT32 in the sample, corresponding to UDINT).

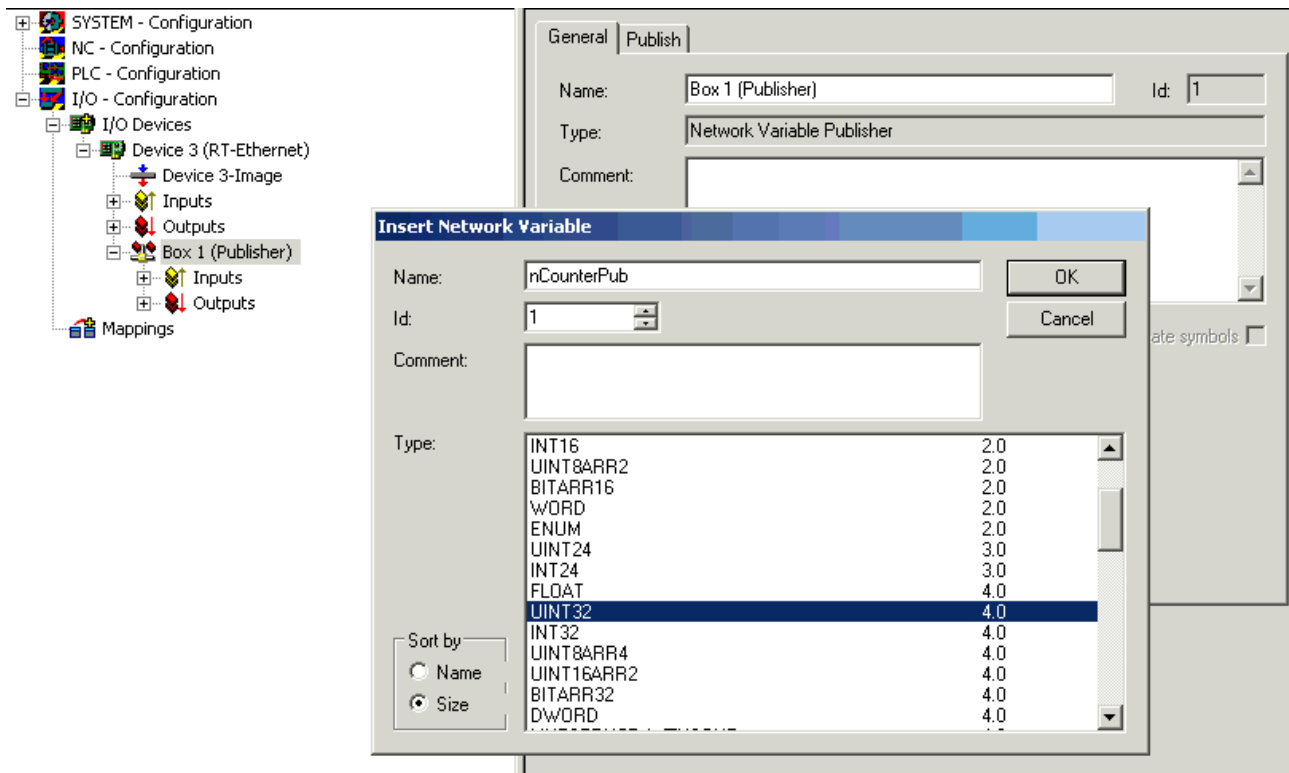


Fig. 112: Adding a network variable

Inputs and outputs are created underneath the added variables.

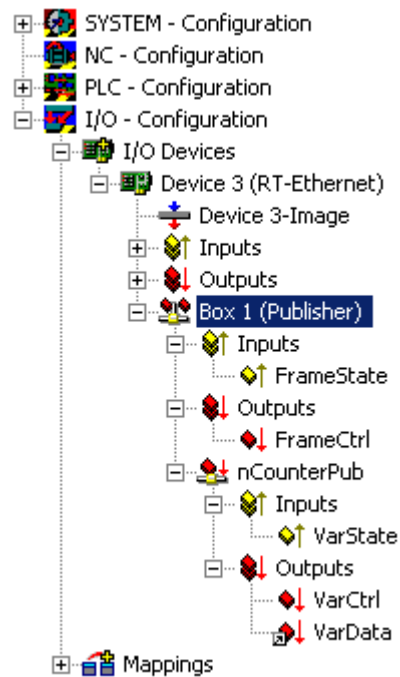


Fig. 113: Inputs/Outputs of the inserted variables

The FrameState input under the box indicates the current status of the sent Ethernet frames.

The following values are possible for the FrameState:

Short description	Value	Description
Not sent (frame skipped)	0x0001	
Error (frame oversized)	0x0001	The maximum size of an Ethernet frames was exceeded. The linked variable should be smaller.

A Control Word can be written in the FrameCtrl output under the box.

The following values are possible for FrameCtrl:

Short description	Value	Description
Disable sending	0x0001	Sending of a frame is interrupted. Sending of the frame does not restart until the value is 0 again.

The VarState input under the network variable indicates the current status of the network variable.

The following values are possible for VarState:

Short description	Value	Description
Not sent (variable skipped)	0x0001	

A Control Word can be written in the VarCtrl output under the network variable.

The following values are possible for FrameCtrl:

Short description	Value	Description
Disable publishing	0x0001	Sending of the network variable is interrupted. Sending of the network variable does not restart until the value is 0 again.

Mappings

The network variable of the Publisher can be mapped to any output variable with a suitable data type. In the sample, the network variable is linked to the output variable of a PLC.

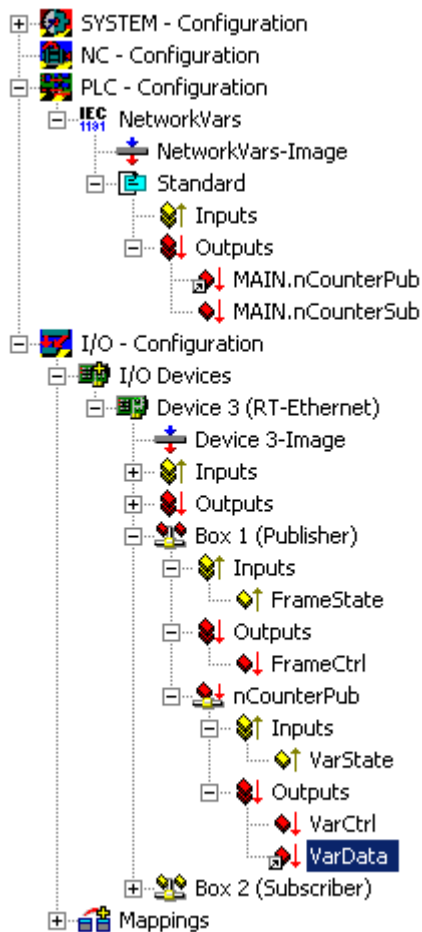


Fig. 114: Mapping of the network variable with an output variable of the PLC

5.4.3 Configuration of the Subscriber

In the TwinCAT System Manager, a new box is added for the Subscriber under the RT Ethernet device.

Adding a Subscriber box and linking the network variables

A Subscriber box must be added under the RT Ethernet device.

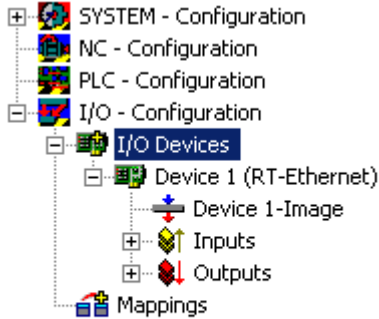


Fig. 115: Creation of an RT-Ethernet device

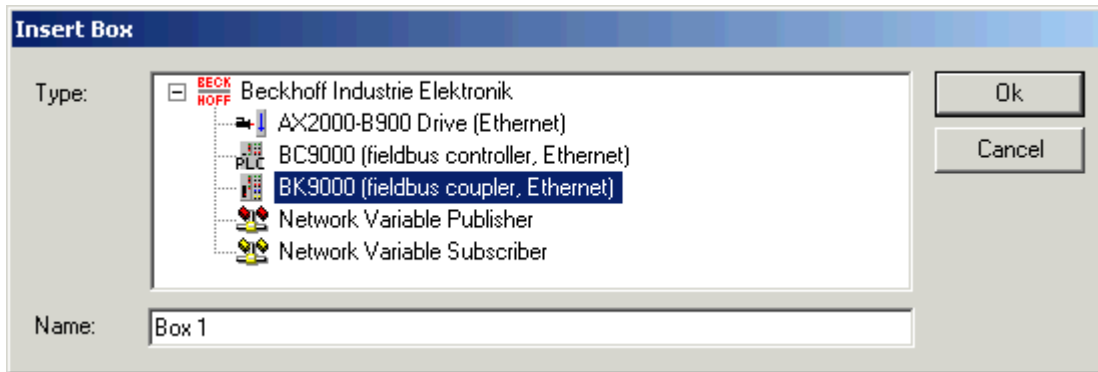


Fig. 116: Creation of a Subscriber Box (“Box 1”)

A network variable is then created under the Subscriber box.

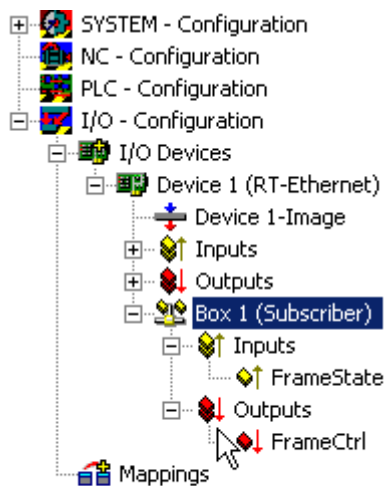


Fig. 117: TwinCAT tree with “Box 1” Subscriber Box

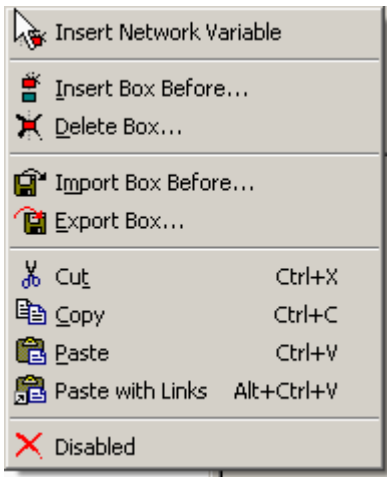


Fig. 118: Create network variable with right click

The link to a Publisher variable can be created automatically. First, you need to find the Publisher computer. All variables of this Publisher are then shown in a list.

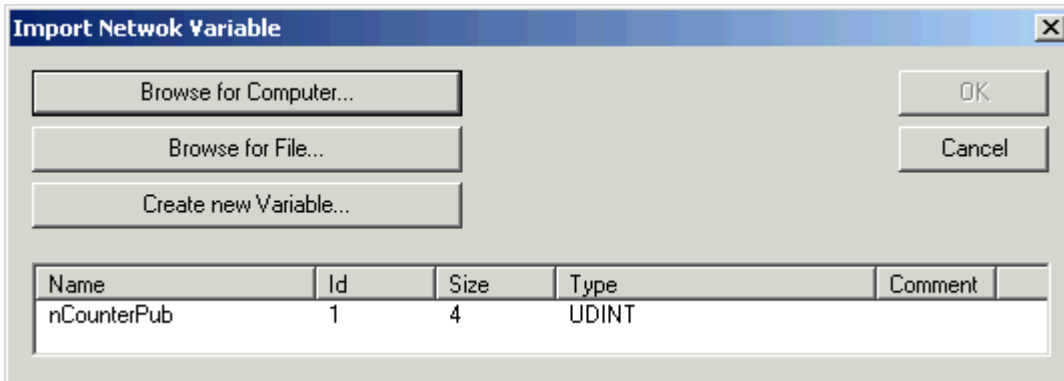


Fig. 119: Searching for Publisher computer

Inputs and outputs were created under the added variable.

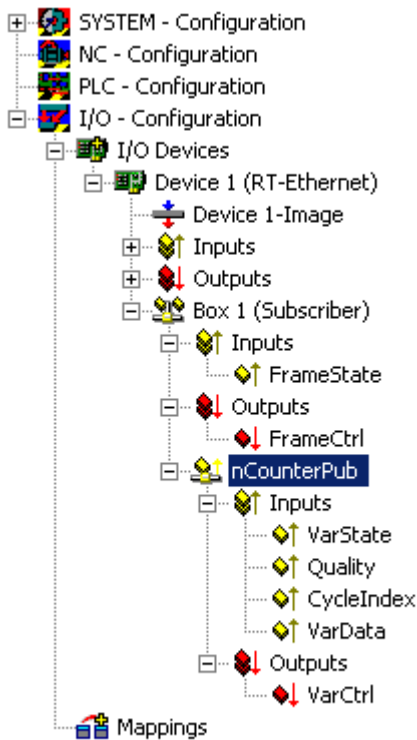


Fig. 120: Created in/outputs of the inserted variables

The FrameState input and the FrameCtrl output underneath the box give information about the current status or control of the received Ethernet Frames. FrameState and FrameCtrl are reserved and are currently not supported.

The VarState input and the VarCtrl output under the network variable indicate the current status (or control) of the received network variable. VarState and VarCtrl are reserved and are currently not supported.

Quality of the network variables

The quality of a network variable is assessed on the Subscriber side. Two input variables are available for this purpose under the network variable. The Quality variable provides a counter with a resolution of 100 μ s. The counter value indicates the variable delay. The sample below shows the online value of Quality when the network connector is unplugged (counter increases) and reconnected (counter value 0).

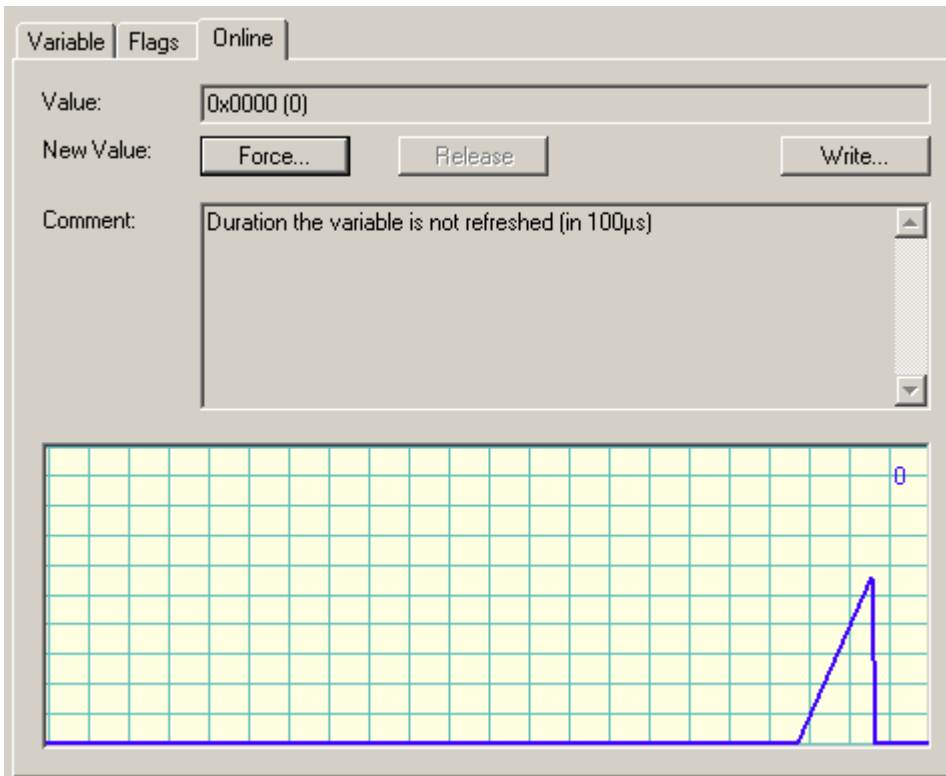


Fig. 121: Time diagram (Online value) of the Quality variables

In addition to the Quality variables, the CycleIndex variable is incremented in each Publisher cycle.

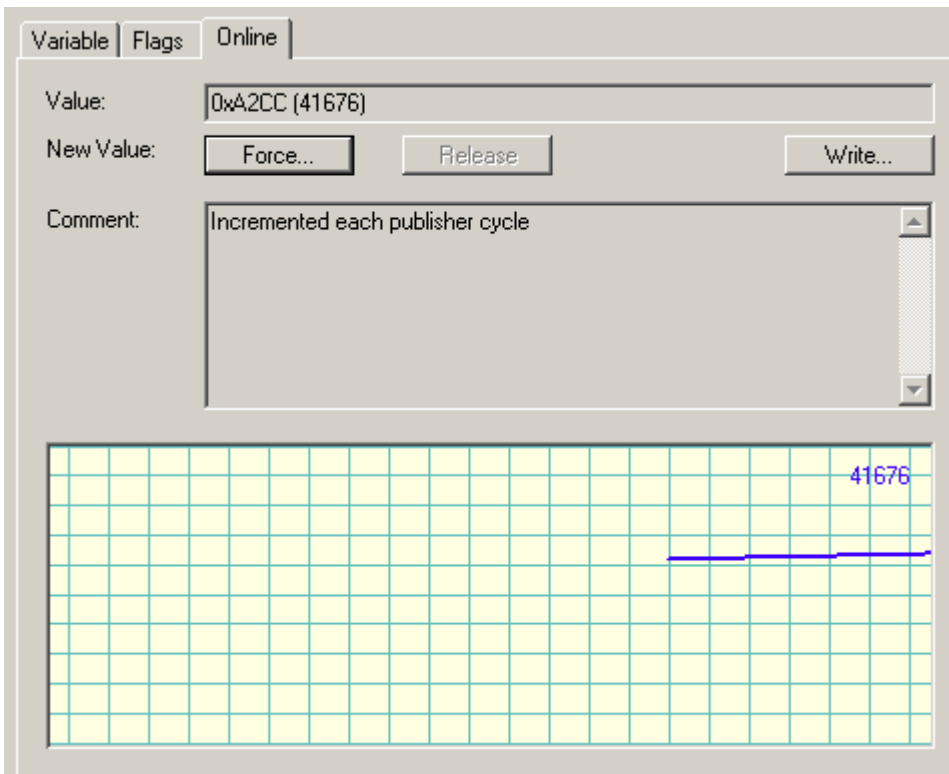


Fig. 122: Time diagram (Online value) of the CycleIndex variables

5.4.4 Beckhoff network variables - Settings

Beckhoff network variables (NWV) can be used for cyclic or acyclic sending of data between Windows-based PCs. In a device declared as a publisher (sender), such a network variable is received on the other side by a subscriber declared as the same type. As the name suggests, this data traffic is network-based, and the configuration is directly based on the protocols used.

A choice of two protocols is available:

- **MAC:** An ISO Layer 2 frame is sent with a sender and receiver MAC address, Ethertype 0x0806. An IP part with the destination IP address (e.g. 192.168.0.1) is not included. The telegram can therefore be further processed via a switch, but usually not via a router.
MAC stands for media access control and in this case refers to the (unique) hardware address assigned to each Ethernet device during production. For sample, the Ethernet port of a Beckhoff PC might have the MAC ID 00:01:05:34:05.84, with "00:01:05" representing the Beckhoff ID and the rest assigned during production. The route of each Ethernet telegram between two Ethernet cable ends is determined by the source MAC and the destination MAC.
The Ethernet telegram is identified as Beckhoff real-time Ethernet by the Ethertype 0x88A4. As a real-time Ethernet telegram (RT Ethernet) it bypasses the regular Windows TCP stack and is sent with higher priority, i.e. "immediately", via the specified Ethernet port of the PC.
An option is available for configuring whether the sent telegram is received by all (broadcast), many (multicast) or a single subscriber (unicast).
- **UDP/IP:** The recipient is identified via an additional IP header in the Ethernet telegram. The UDP Ethernet frame can thus be further processed via a router.
Once again, broadcast, multicast and unicast are available as options. The Ethernet telegram is identified as Beckhoff real-time Ethernet through the Ethertype 0x88A4 and treated as an RT protocol in the TwinCAT PC.
In contrast to TCP, as a connection-less protocol UDP requires no acknowledgement of receipt for the message, i.e. the publisher does not know whether the subscriber has received the message. The [ARP protocol](#) [▶ 126] is therefore used for remote terminal monitoring in TwinCAT.

The telegram with the process data arrives at the recipient device (network port) via these addressing modes. In the Ethernet device/TwinCAT several transported process data are allocated via a variable ID

All network variables must be declared in the System Manager before they can be used.

The following intervention options are then available during operation:

- Sending of a configured network variable can be blocked dynamically
- The destination IP or destination MAC can be changed dynamically
- The variable ID "variable ID" can be changed dynamically
- The NWV content can be changed, but not the size (bit size)

Diagnostic variables on the publisher and subscriber side provide information about the connection quality.

If network variables are used, the temporal boundary conditions for the network topology used must be taken into account: in the case of IP addressing (routed) on a case-by-case basis several 100 ms communications cycle can be achieved, in the case of MAC addressing (switched) approximately 10 ms and less.

● Diagnostic variable "quality"

I If the processing tasks operate with different cycle times or the user changes the DataExchangeDivider, this must be taken into account in the analysis of the diagnostic variables. In conjunction with a fast Subscriber (e.g. 10 ms), a slow Publisher (e.g. 100 ms) leads to poor connection quality (as reported by the diagnostic variable "Quality").
Dynamic temporary blocking of sending a Publisher must also be taken into account. In this case the Subscriber registers poor quality.

● Diagnostic variable "CycleIndex"

I Please note the following information in order to decide whether you have to serve the variable CycleIndex.

Basic principles of Beckhoff network variables

- **Quality:**
 Time in [100 μs] by which arrival of the NWV at the Publisher was delayed.
 Relative arrival location:
 Input process image of the TwinCAT system
 Relative arrival time:
 Time at which the next cycle is loaded into the input image

Note:

The reason for determining the delay so precisely is that the NWVs are managed directly by the IO driver, independent of the cycle. Nevertheless, the data of an NWV that is delayed by a few percent of the cycle time will not be taken into account until the input process image is read during the next task cycle.

Note EL6601/EL6614:

Even with the EL66xx the NWV arrival time is defined as the time when the data are available in the input process image of the RT device, not the time of arrival at the EL66xx or in the input image of the EtherCAT device.

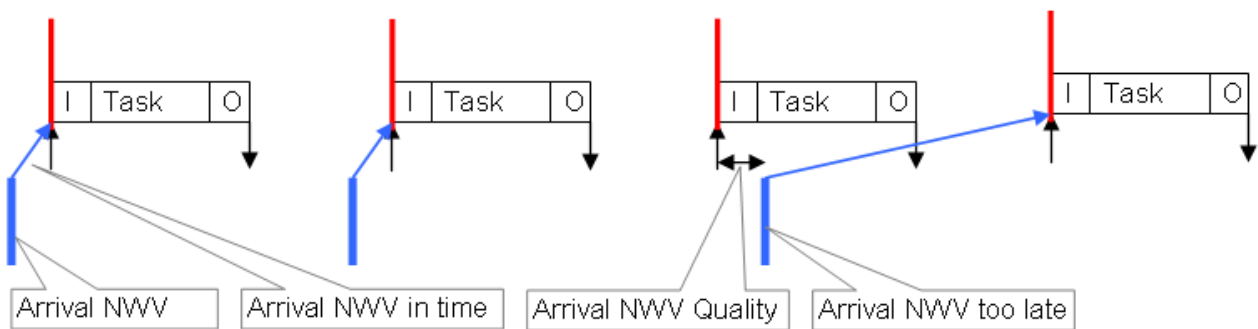


Fig. 123: Interrelationship between quality and delayed network variable

- **Variable ID**
 The variable ID (16 bit) is used for global identification of the individual process data. Therefore, an ID in the Publisher or Subscriber group may only be used once within a TwinCAT device, see Fig. *Sample for communication via network variables*: Publishers 1 and 2 on PC1 must have different IDs (10 and 8), although the same ID (8) may be used in Publisher 2 and Subscriber 1.

Selecting the variable ID

i In order to achieve unambiguous allocation we recommend using different IDs for each data communication between connected PCs. Reason: In Fig. *Sample for communication via network variables*, PC2/Subscriber2 not only receives the designed ID=8-variable from PC1/Publisher2, but, since it is sent as a broadcast (!), it also receives the NWV from PC3/Publisher1. Differentiation is then no longer possible in PC2.

- **Cycle Index**
 The 16-bit cycle index is a counter sent by the Publisher together with the data. It is generally incremented with each transmission and can therefore be used as an indicator for flawless transfer. It can be read on the subscriber side as *CycleIndex*. Its appearance depends on the Publisher platform:
 - Publisher on a PC: The variable *CycleIndex* is not visible and is automatically and cyclically incremented by the System Manager
 - Publisher on an EL66xx: The variable *CycleIndex* is visible and must be incremented/served by the user such that it is not equal 0 on the subscriber side.

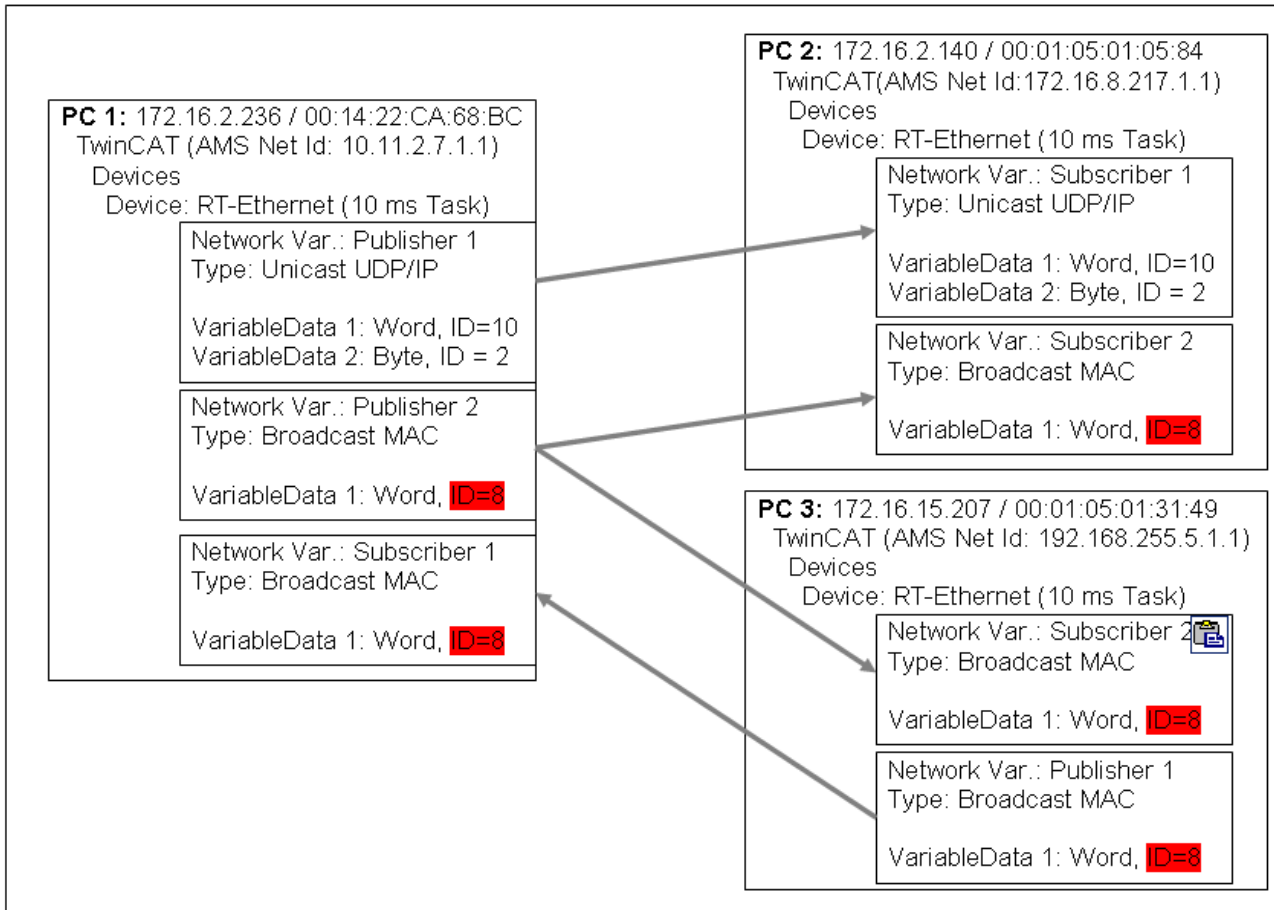


Fig. 124: Sample for communication via network variables

i Data representation on different platforms

Please note that simple and complex data (WORD, ARRAYs, REAL, STRING, user-defined structures) are represented internally in a different manner on different platforms!

x86 platforms use byte-alignment, others (ARM) 2-byte or 4-byte alignment.

This means that if a complex structure is created in an x86/PC PLC project and an ARM PLC project, they can each have a different effective size and a different internal structure. (see Fig. "Data representation e.g. x86 Systems vs. ARM Systems")

In the sample, the structure in the CX (and hence the network variable to be created there) is larger than in the PC; also the word and real variables do not match each other because a variable can begin at any byte position in the PC, but only at every even-numbered one in the CX.

Consequences

- Recommendation for structures that are identical on both end devices
- firstly, all 4-byte variables (must lie at an address that is divisible by 4)
 - then all 2-byte variables (must lie at an address that is divisible by 2)
 - then all 1-byte variables

Further recommendations

- if STRING(x) is used, the "EndOfString" zero is also interpreted as a character, otherwise x+1 must be divisible by 4
- the above rules also apply to sub-structures.

Please refer to the notes in the Structure section of the Infosys.

Consequences

Use of Bus Terminal Controllers (BICxxxx, BXxxxx)

Since the representation of floating point numbers (REAL) on Bus Terminal Controllers (BCxxxx, BXxxxx) differs from that in the x86, these cannot be transmitted. "SINT", for sample, can be used for signed values.

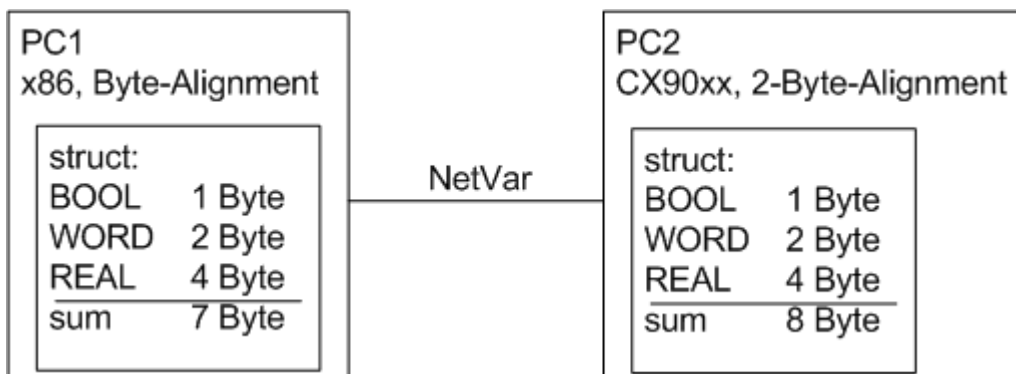


Fig. 125: Data representation e.g. x86 Systems vs. ARM Systems

Settings in the System Manager

i Appearance of the variables

Depending on the platform used (PC or EL66xx), the publisher/subscriber will appear differently. A publisher/subscriber can be created

- On a PC network interface, see Fig. *Publisher settings - RT Ethernet*
- on an EL66xx, see [Beckhoff network variables - Settings](#) [▶ 122]

The following settings options are available in the Beckhoff System Manager TwinCAT 2.10 build 1328:

Publisher, Box

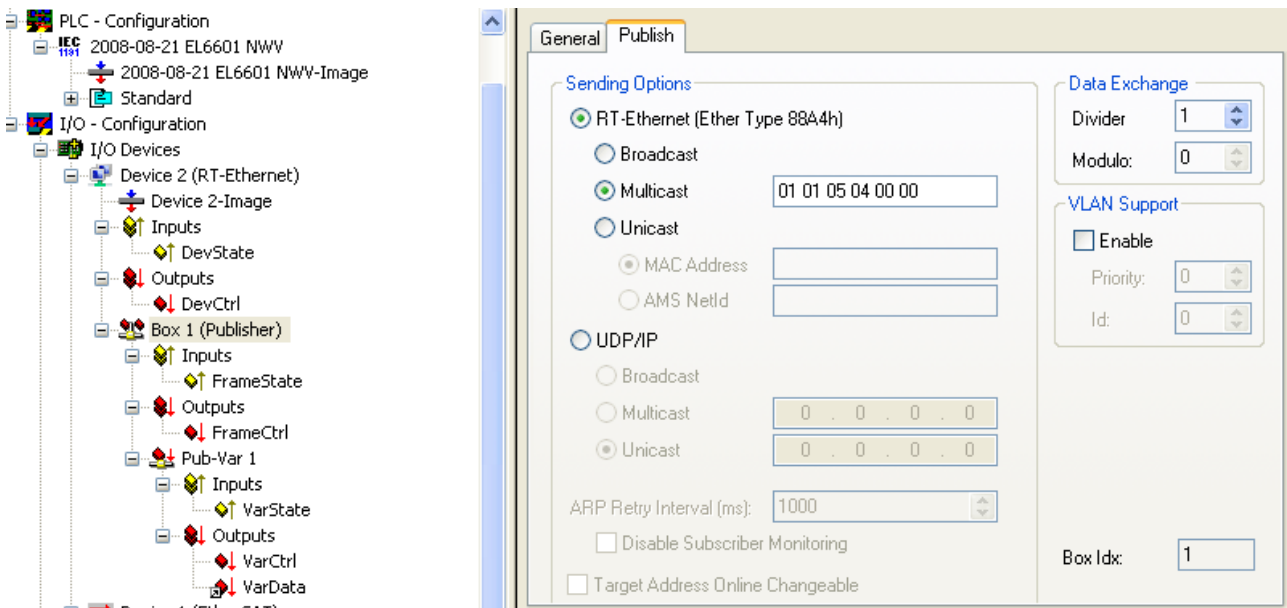


Fig. 126: Publisher RT Ethernet settings

RT Ethernet settings:

- **MAC-Broadcast:** Sent to all network devices, destination MAC FF:FF:FF:FF:FF:FF.
- **Multicast:** A destination MAC address becomes a multicast address if the first bit in the first byte of the MAC (the so-called group bit) is set. With the Beckhoff ID "00 01 05" the default target address "01 01 05 04 00 00" is formed, as shown in Fig. *Publisher RT Ethernet settings*.
The MAC range 01:00:5E:00:00:00 to 01:00:5E:FF:FF:FF is intended for general multicast application, with the first 3 bytes specified by the IEEE and the last 3 bytes derived from the lower part of the IP address of the destination PC. The resulting destination MAC therefore never physically exists in the network. Instead, the destination network card detects Ethernet frames formed in this way as multicast frames sent to it, although the Ethernet port itself can have another, unique MAC address. Please refer to the relevant literature for further rules relating to the formation of multicast MAC/IP addresses.
- **Unicast:** Either direct entry of the destination MAC or via the AMS Net ID of the destination device, e.g. 123.456.123.456.1.1, in which case this route must be entered in the local AMS router (right-click on the TwinCAT icon in the taskbar --> Properties --> AMS router)

● Use of broadcast and multicast

i Network variables sent as broadcast or multicast at MAC or IP level can generate high network load (depending on the cycle time), since they are multiplied into the whole connected network. This may cause simple network devices such as printers to crash. With short cycle times all network traffic may become blocked. We strongly recommend using unicast addressing, taking into account variable identification, as described above.

Advanced Settings:

- **Data exchange:** The task cycle time * divider is the rhythm at which this network variable is sent. (not for EL66xx).
- **VLAN support:** In conjunction with manageable switches the Ethernet frame parameterized here can be assigned a fixed route via VLAN tagging (Virtual Local Area Network).

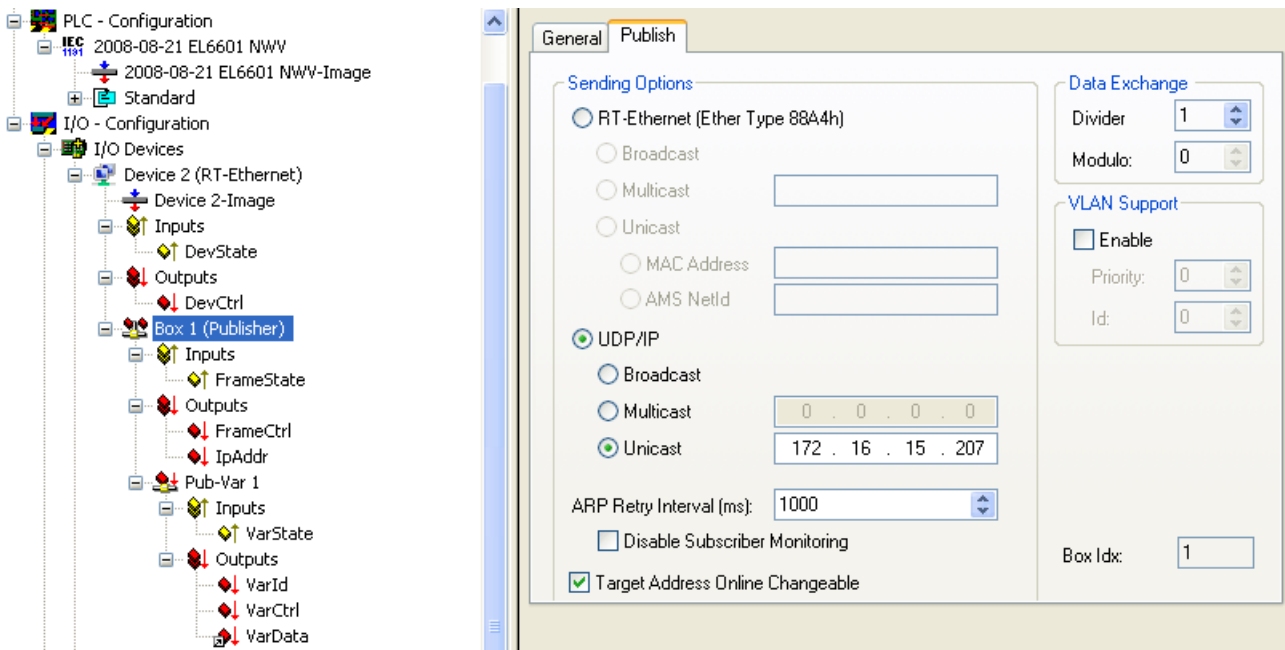


Fig. 127: Publisher settings - UDP/IP

UDP/IP settings - the addressing technique of the IP network layer with IP addresses is used. UDP is a connection-less protocol without feedback.

- Broadcast: Sent to all device with destination IP (v4) 255.255.255.255
- Multicast: The destination IP must be specified, see notes on MAC multicast
- Unicast: Specify the target device (e.g.: 192.168.0.1), making sure that it can be reached through the subnet mask

● Use of broadcast and multicast

i Network variables sent as broadcast or multicast at MAC or IP level can generate high network load (depending on the cycle time), since they are multiplied into the whole connected network. This may cause simple network devices such as printers to crash. With short cycle times all network traffic may become blocked. We strongly recommend using unicast addressing, taking into account variable identification, as described above.

Advanced Settings:

- "ARP Retry Interval": In order to ascertain the presence of the recipient, the publisher sends an ARP request to the target device at these intervals. If the network administration of the recipient is operational, it sends an ARP reply. This is only meaningful with unicast. In the event of an error bit 3 is set (0x0004) in the diagnostic FrameState variable.
Note: ARP handling (ARP = Address Resolution Protocol: allocation of hardware/MAC addresses to network addresses [IP]) is managed by the operating system (Windows).
- "Disable Subscriber Monitoring": deactivates the procedure described above.
- "Target Address changeable": In this case the destination IP can be changed dynamically.

Publisher, Variable

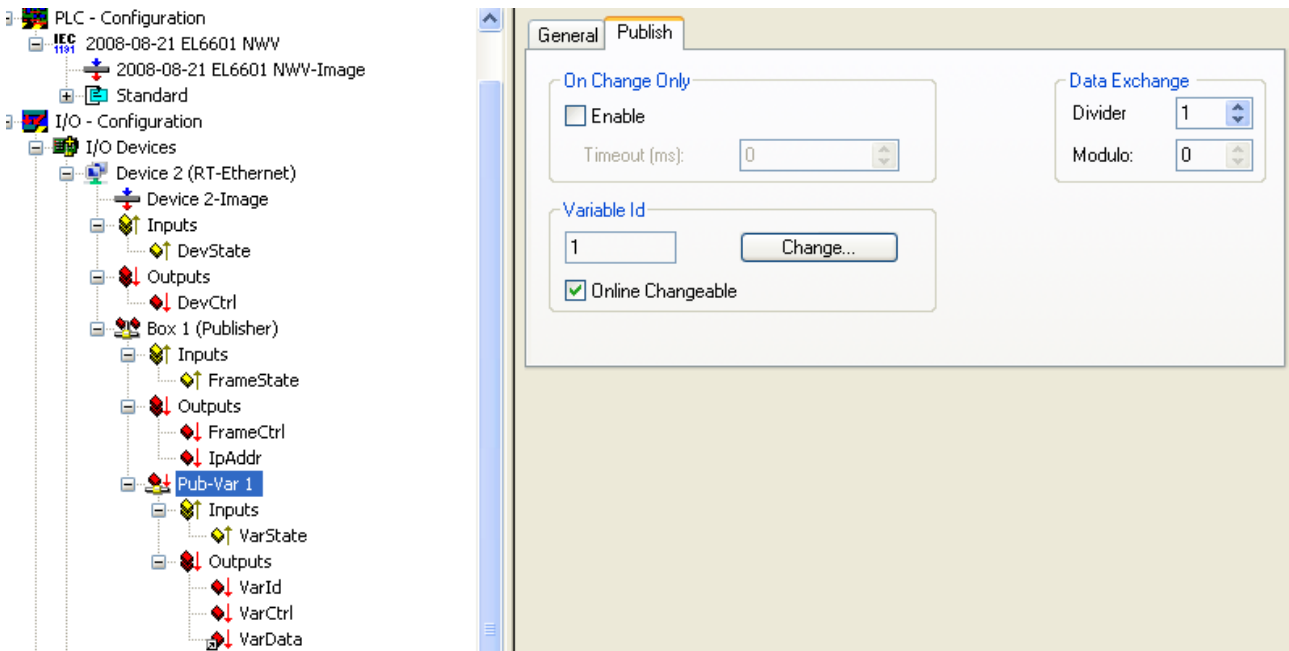


Fig. 128: Publisher variable settings

Settings:

- "Variable ID": Identification number with which the variable is sent. Can be changed online via PLC where appropriate.
- "Data exchange": see above (not for EL66xx).
- "On change only": NWV is only sent if the value changes (not for EL66xx).

Subscriber, Box

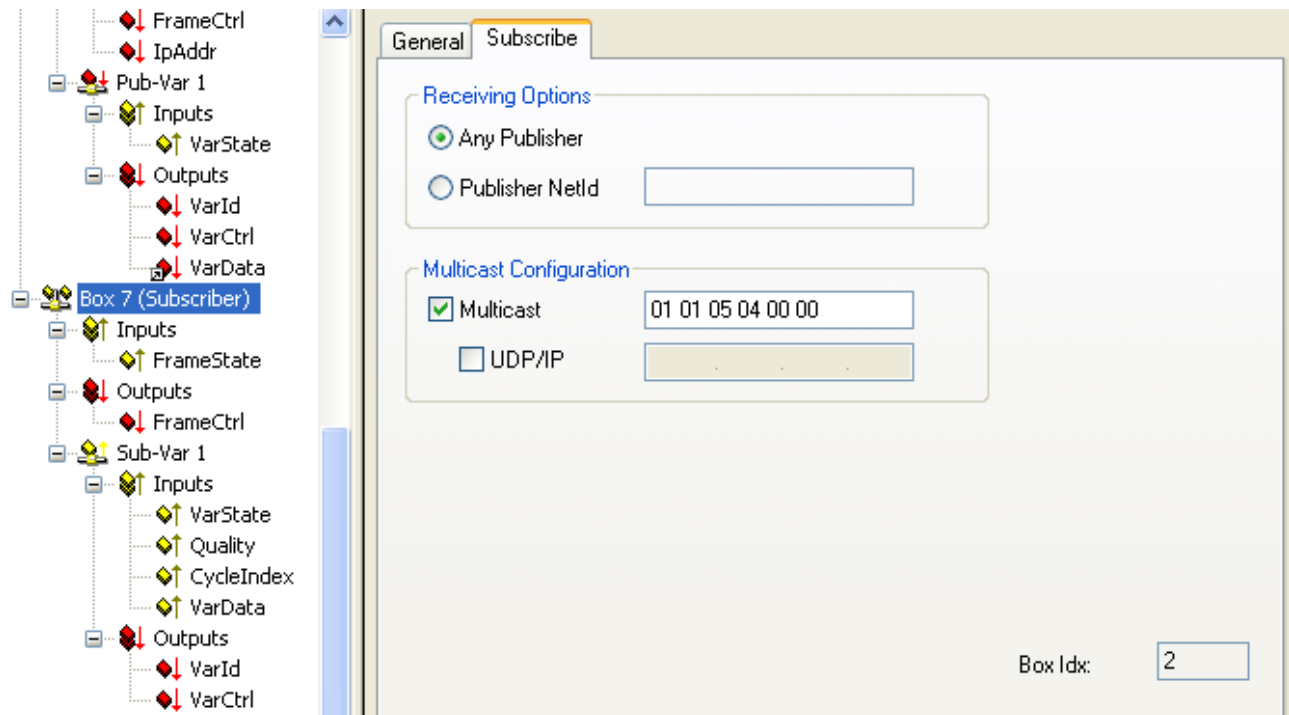


Fig. 129: Subscriber settings

Settings:

- "Receiving Options": Only permits NWVs from a certain publisher for this subscriber

- "Multicast Configuration": ditto.

Process data:

- "VarId": If activated, the variable ID can be modified online

Subscriber, variable

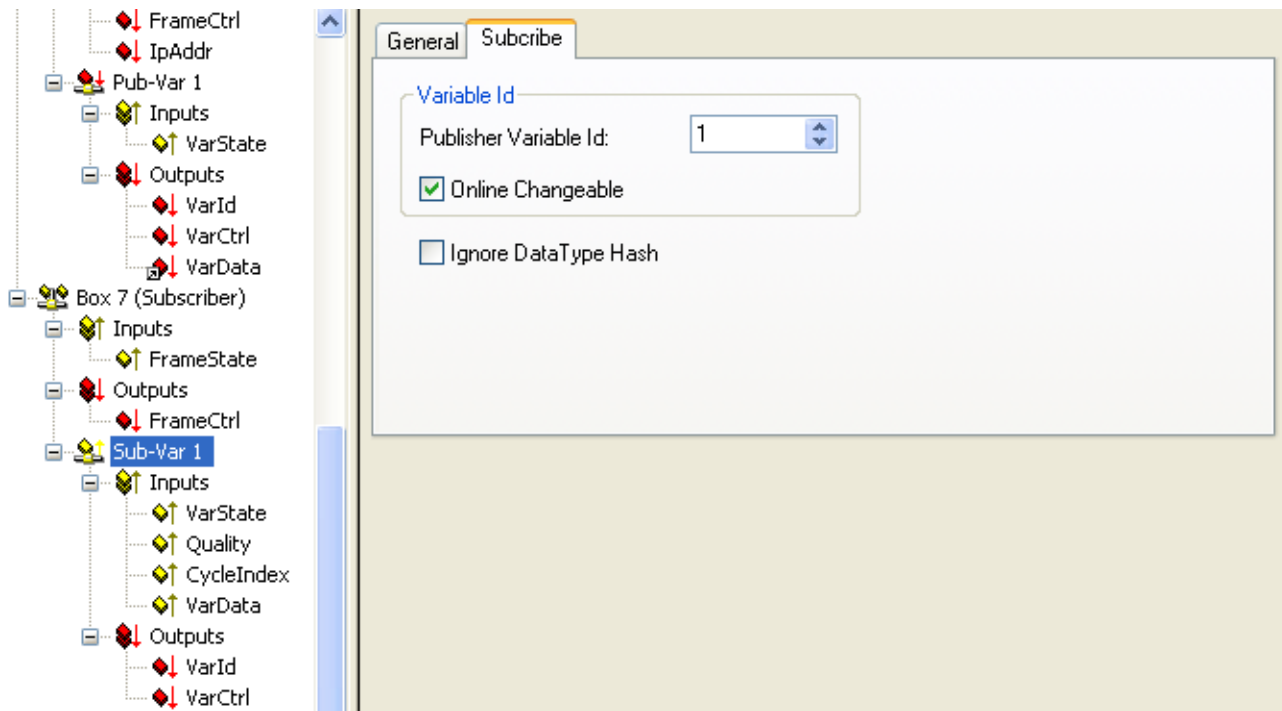


Fig. 130: Subscriber variable settings

Settings:

- "Variable ID": Only permits NWVs with a certain ID for this subscriber. Can be changed dynamically via PLC where appropriate.
- "Ignore Data Type Hash": Hash calculation is currently not supported

Process data:

- "Quality": See explanatory notes above.
- "CycleIndex": This index is incremented with each successful transfer, IF this is done by the opposite side, i.e. the publisher. If the publisher is an EL66x, the user must increment *CycleIdx* there.
- "VarData": Transferred data.

6 Application samples

6.1 Sample programs

● Using the sample programs

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

Sample 1: Determine Publisher/Subscriber data throughput

 Sample program (Download)

With appropriate EtherCAT cycle time and depending on the scale and number of the publishers/subscribers configured in the EL66xx, real-time cycle times down to 500 µs or below are possible.

Typical throughput values for EL6601, FW08, Rev. EL6601-0000-0018 are

- 1 publisher with 1000 bytes, 1 subscriber with 1000 bytes, simultaneous bidirectional operation: 2 ms
- 1 publisher with 100 bytes, 1 subscriber with 100 bytes, simultaneous bidirectional operation: 300 µs

Both characteristic values were determined with this sample program. TwinCAT from version 2.11 is required for the *.tsm System Manager file.

The EL6601 is used as a sample to explain configuration as publisher or subscriber for network variables. The dialogs under TwinCAT 2.10 and TwinCAT 2.11 here are slightly different.

Sample 2: Online diagnostics

 Sample program (https://infosys.beckhoff.com/content/1033/el6601_el6614/Resources/2349552907.zip)

The following objects are available for initial diagnostic in the CoE directory:

- 0xFA01, subindex 01: Frame Counter Rx (incoming to RJ45 socket)
- 0xFA01, subindex 02: Frame Counter Tx (outgoing from RJ45 socket)

The values can be read from the controller using PLC function blocks (FB_EcCoeSdoRead in TcEtherCAT.lib).

This and further diagnostic information from the CoE of the EL66xx are accessible via https://infosys.beckhoff.com/content/1033/el6601_el6614/Resources/2349552907.zip.

Documents about this

 https://infosys.beckhoff.com/content/1033/el6601_el6614/Resources/17350688139.zip

6.2 Application sample - network printer

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

The following hardware was used for the test configuration:

- TwinCAT master PC with Windows XP Professional SP 2, TwinCAT version 2.10 (Build 1305) and INTEL PRO/100 VE Ethernet adapter
- Beckhoff Ethernet coupler terminal EK1100, terminals EL1012, EL6601 and EL9010
- Printer HP LaserJet 4200tn

A network printer is connected to the EL6601 within the terminal network via the Control Panel of the control system IPC:

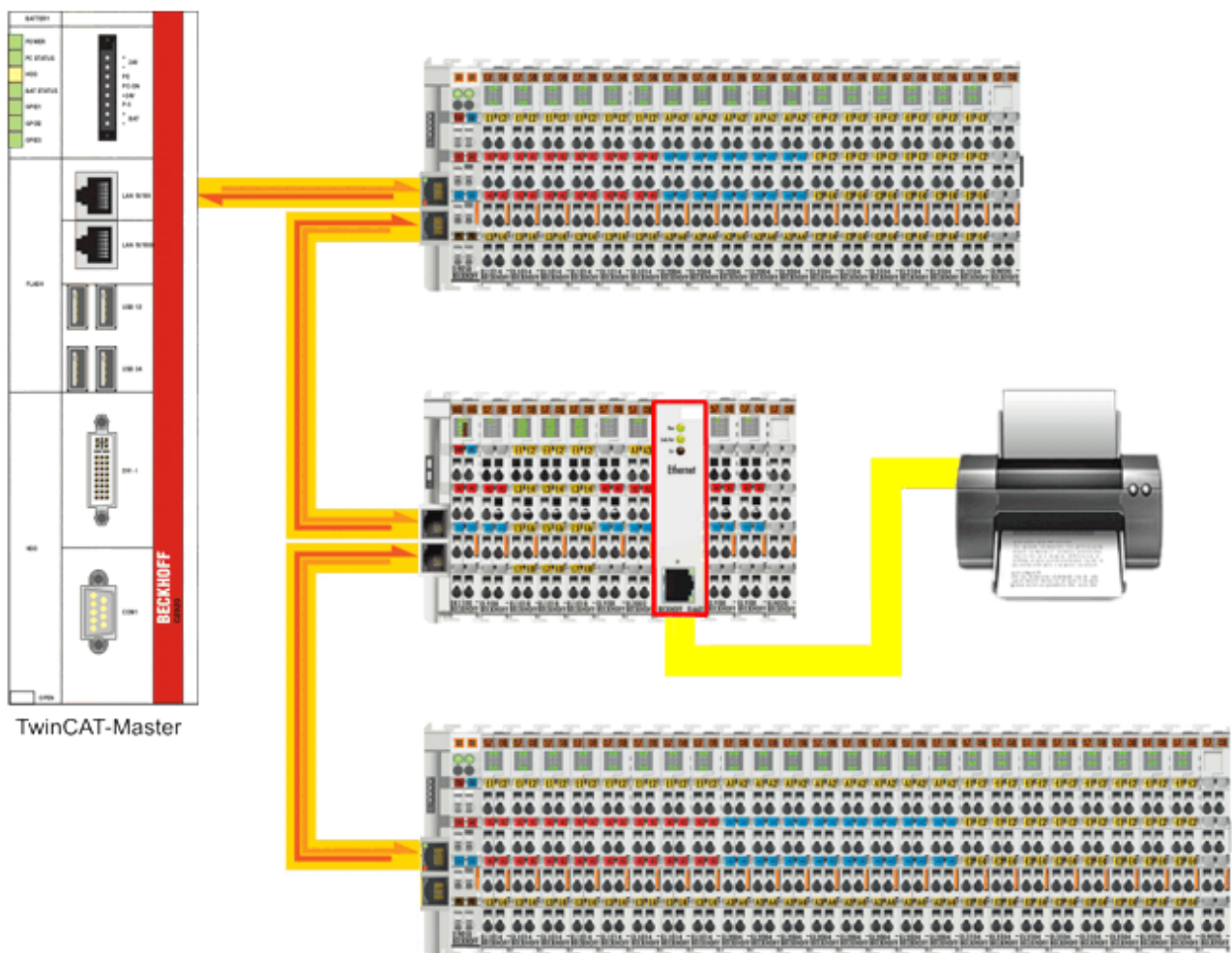


Fig. 131: Connection of a network printer

Legend:

-  EtherCAT connection
-  Ethernet connection

Checking the network address of the TwinCAT master PC

- Start the TwinCAT System Manager in Config mode, read the Bus Terminal configuration, and activate free-run mode

- The EL6601 should appear in the system configuration being in OP mode (see [Online](#) [▶ 90] tab EtherCAT Status Machine)

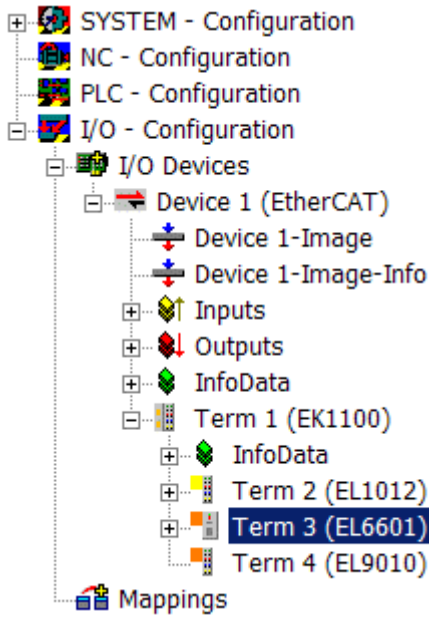


Fig. 132: EL6601 in the configuration

- Connect the network printer with the EL6601 via a CAT-5 (1:1) cable
- In the Control Panel of the controller IPC, check the network adapter via which the EtherCAT system is operated:
Network Connections [Properties] -> *Local Area Connection (TwinCAT LAN adapter)*, [Properties]-> *Internet Protocol (TCP/IP)* [Properties]

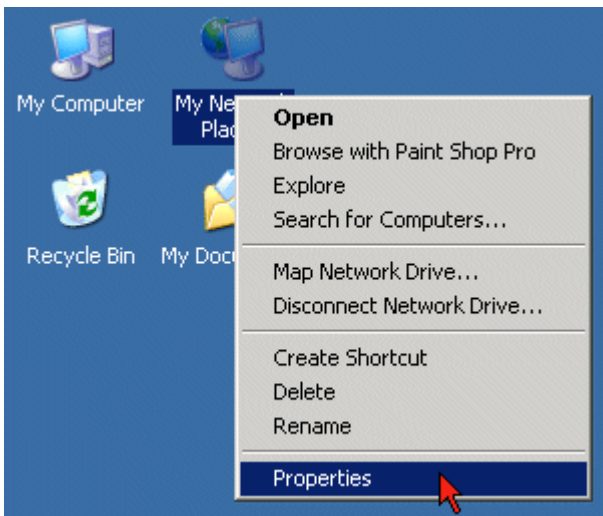


Fig. 133: Context Menu Network Environment -> "Properties"

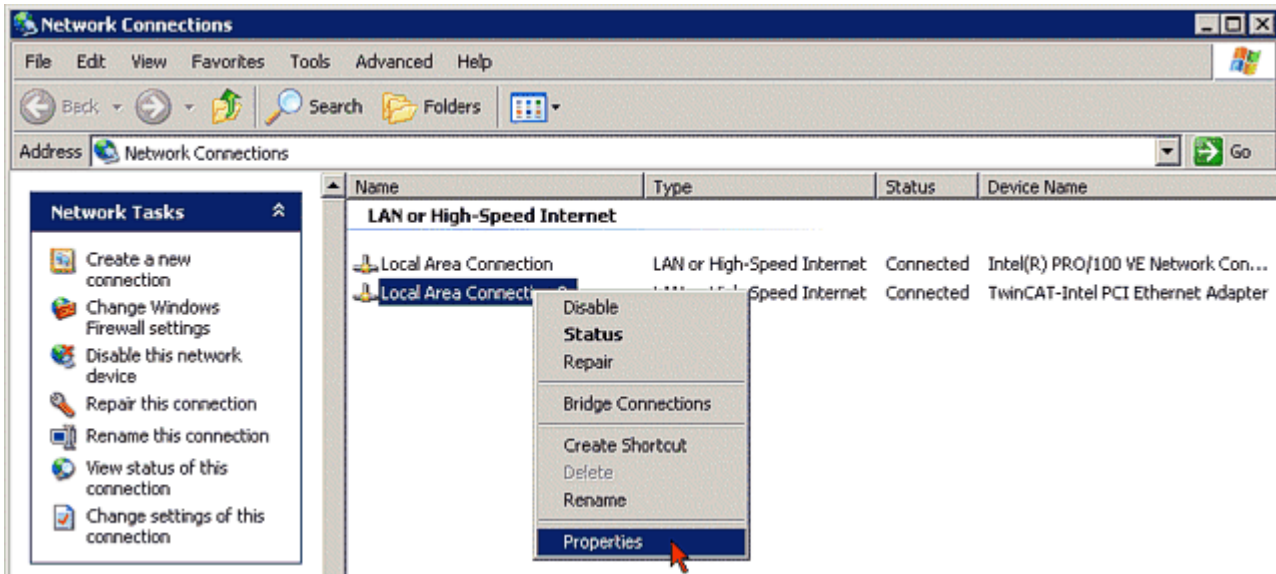


Fig. 134: Context Menu “LAN Connection (TwinCAT-LAN-Adapter),” -> “Properties”

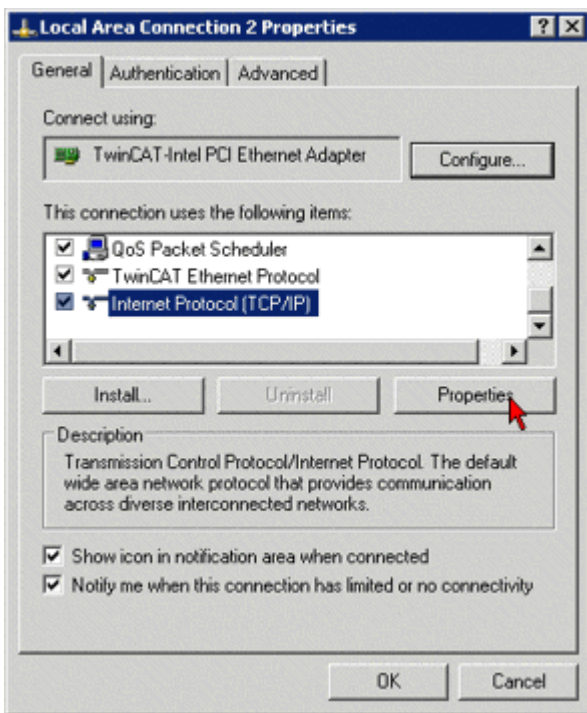


Fig. 135: Properties Internet Protocol (TCP/IP)

- Enter an IP address for the subnet of the network printer (e.g. 192.168.0.1/255.255.255.0) and confirm with OK

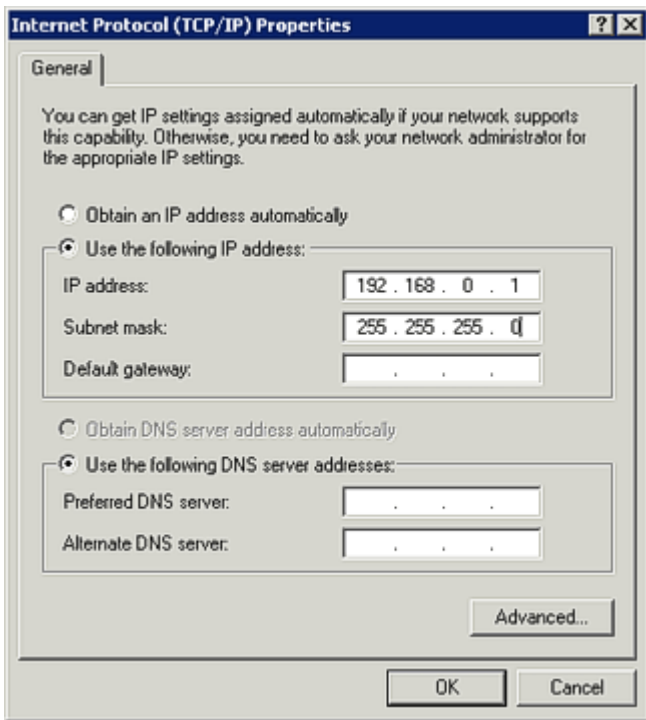


Fig. 136: Entry of the IP address of the subnetwork of the network printer

● Network address

i The network address in the sample is only used for the purpose of illustrating the configuration. Please note that in your application the IP address of network printer must match the IP number range/subnet mask of the master PC.

- Configure the network printer (in the configuration menu of the printer) and allocate the printer an IP address from the IP number range of the subnet defined above (Follow the user guide of your printer), e.g.: 192.168.0.37
- Configure the network printer via the Control Panel: *Printers & Faxes* -> *Add a printer*, in the Add Printer Wizard select *Local printer*, and in the following pull-down menu *Create a new port...* select *Standard TCP/IP Port*

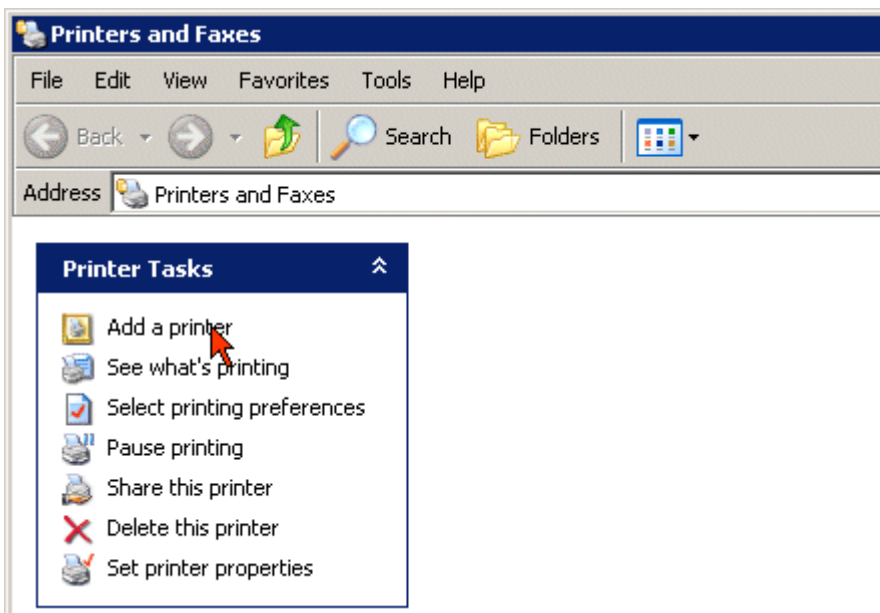


Fig. 137: Control Panel “Add a Printer” under “Printers and Faxes”

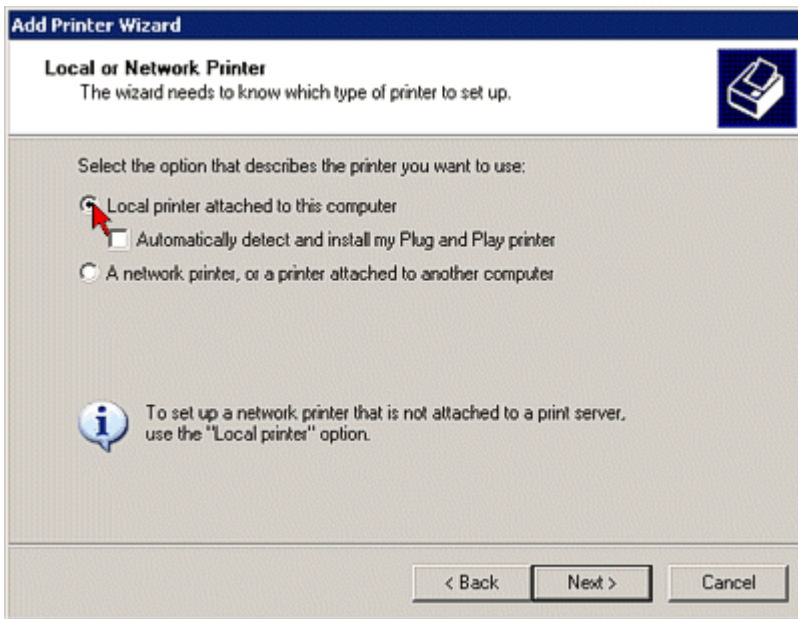


Fig. 138: Select "Local Printer" radio button

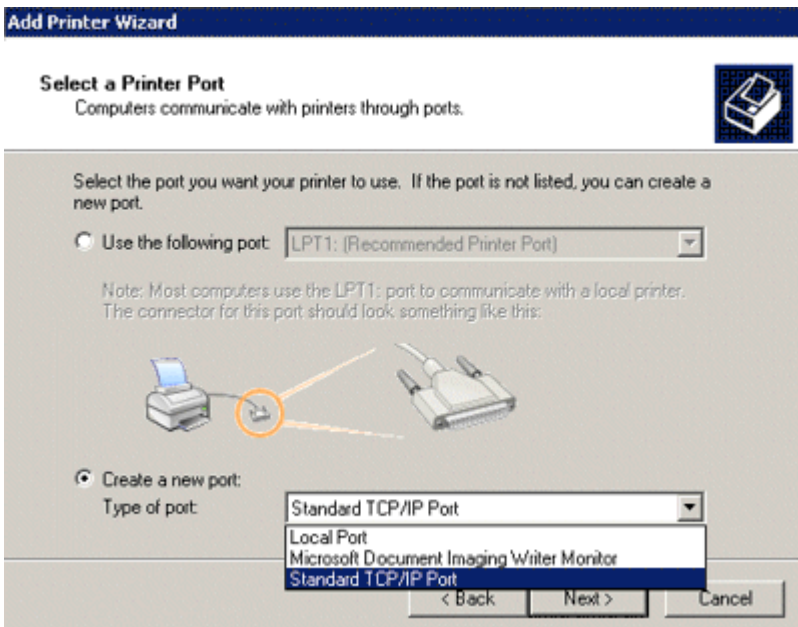


Fig. 139: Select Connection Type

- In the following menu under Printer Name or IP Address enter the IP address of the printer defined in the configuration menu (in the sample 192.168.0.37)

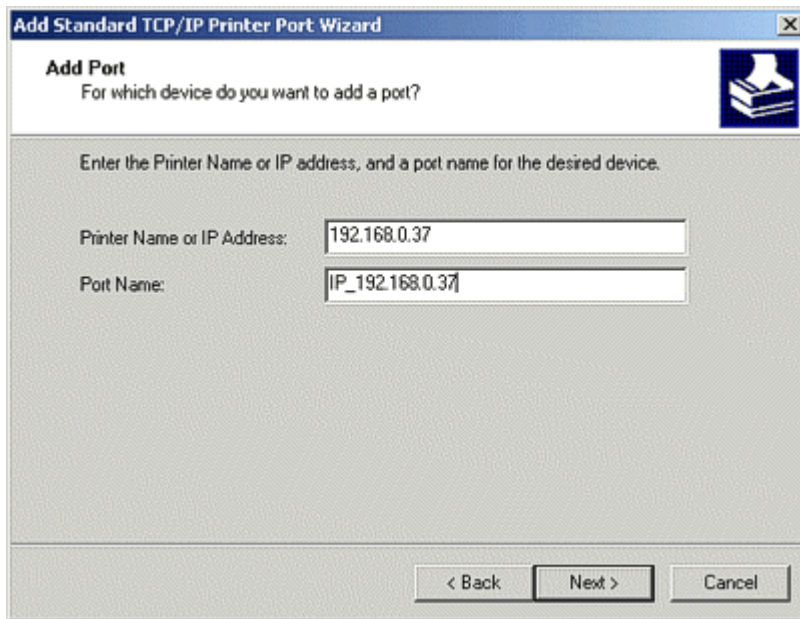


Fig. 140: Entry of the IP address and printer name

- Confirm the port properties and exit the wizard
- In the Add Printer Wizard install the drivers from your printer manufacturer

6.3 Application sample - Service interface with remote desktop

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

The following hardware was used for the test configuration:

- TwinCAT master PC with Windows XP Professional SP 2, TwinCAT version 2.10 (Build 1305) and INTEL PRO/100 VE Ethernet adapter
- Beckhoff Ethernet coupler terminal EK1100, terminals EL1012, EL6601 and EL9010
- Notebook (service PC) for remote desktop connection with Windows XP Professional SP 2 and standard Ethernet interface

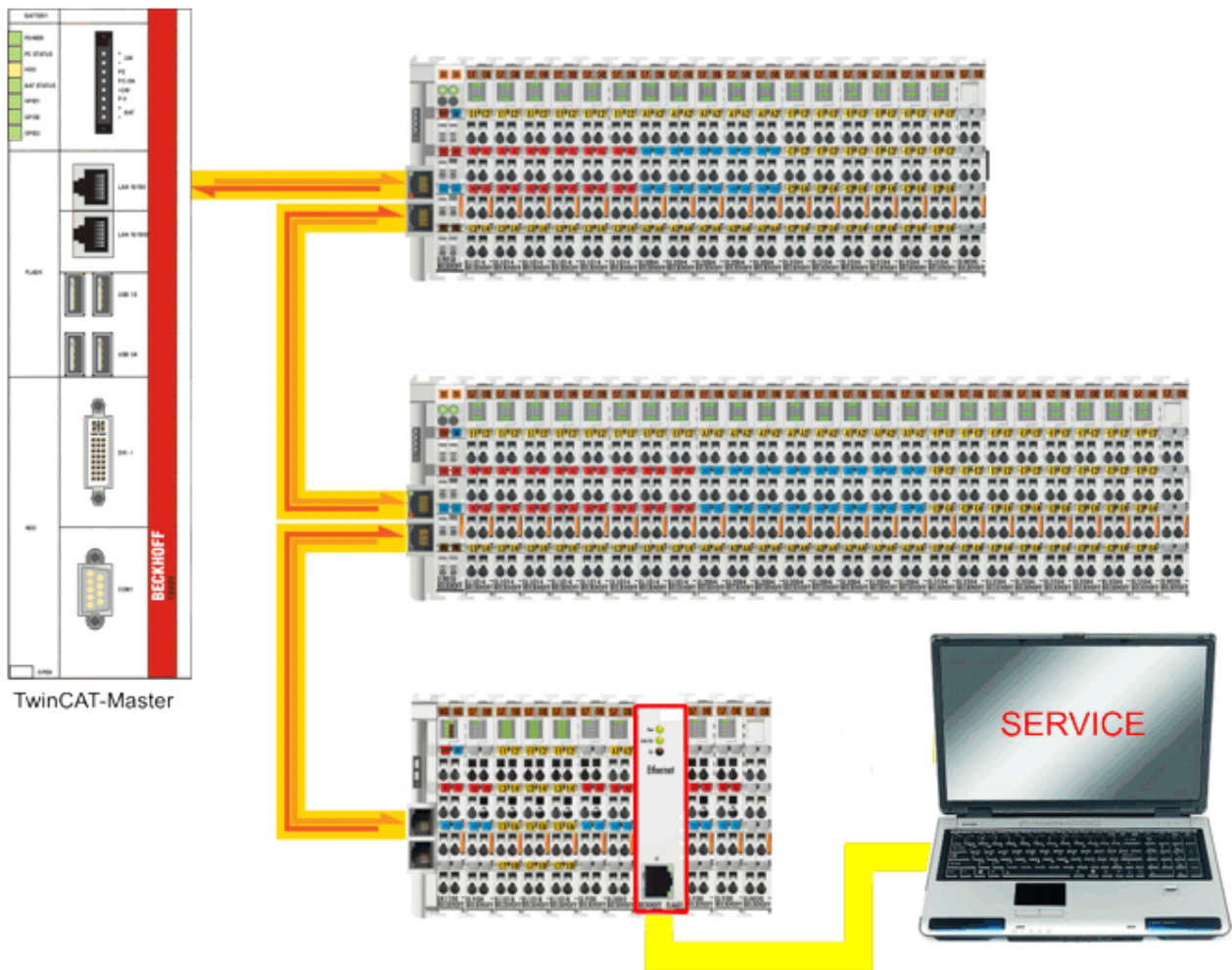


Fig. 141: Integration of a Service interface with remote desktop-PC

Configuration at the TwinCAT master PC

- Start the TwinCAT System Manager in Config mode, read the Bus Terminal configuration, and activate free-run mode
- The EL6601 should appear in the system configuration being in OP mode (see [Online](#) [► 90] tab EtherCAT Status Machine)

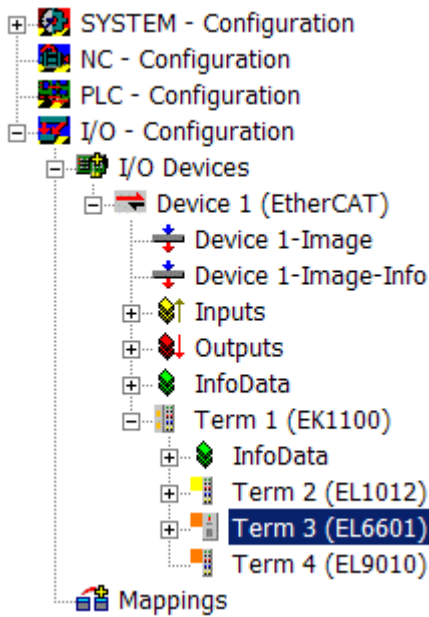


Fig. 142: EL6601 in the configuration

- In analogy to the configuration of a network printer the IP number of the EtherCAT network port of the TwinCAT Master-PC in the subnet is e.g. 192.168.0.1
- Open the Control Panel of the TwinCAT master PC and click on *Security Center -> Windows Firewall*

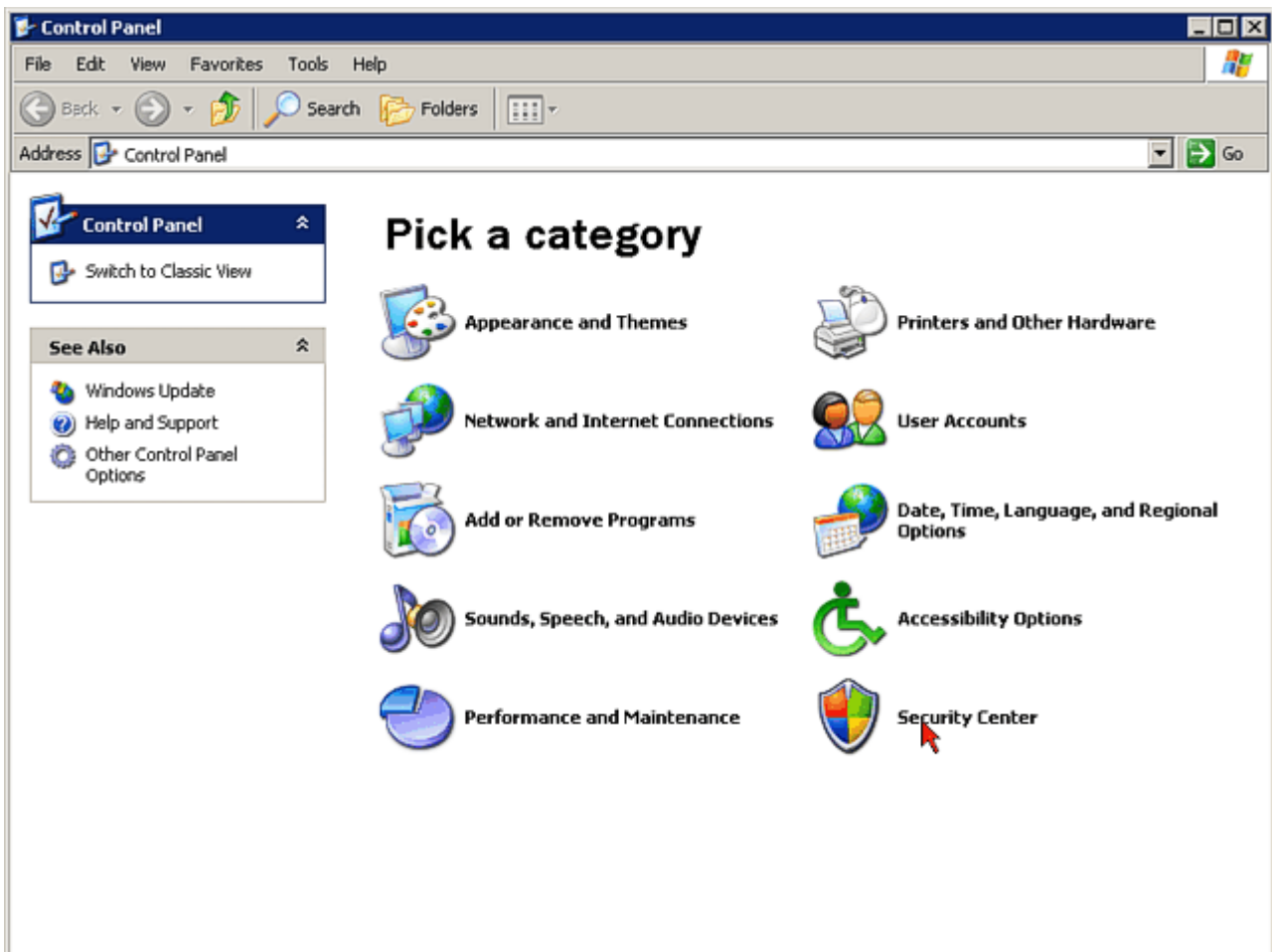


Fig. 143: Selection of "Security Center" in the View Control Panel Category



Fig. 144: Choose security settings for Windows Firewall

- If the Windows firewall is activated click on tab *Exceptions*
- Select *Remote Desktop* and confirm with *OK*

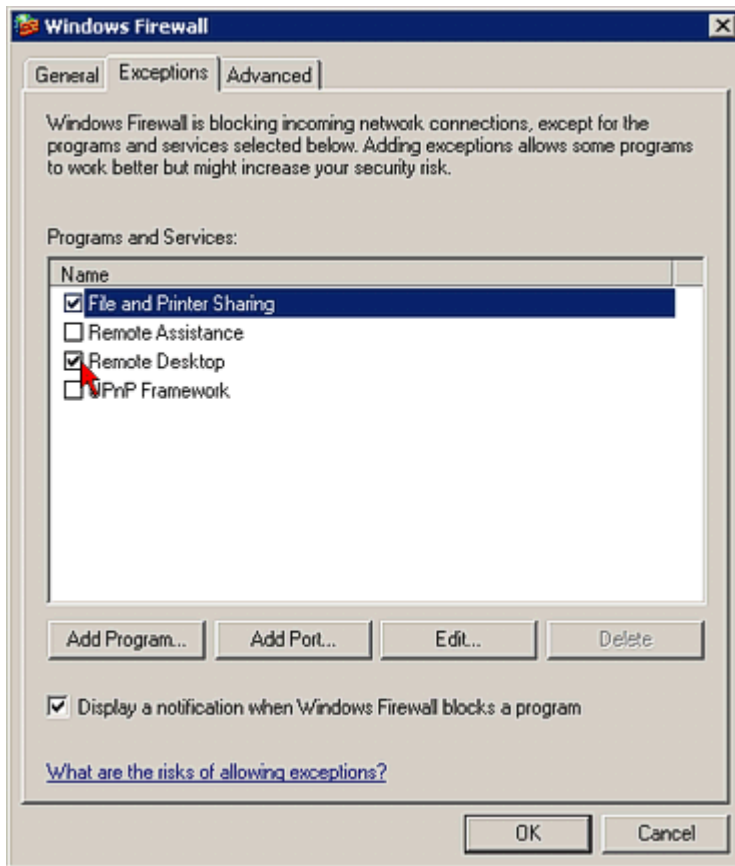


Fig. 145: At the “Exceptions” Tab activate the “Remote desktop” service

- Return to the Control Panel of TwinCAT master PC and select *Performance and Maintenance* -> *System*

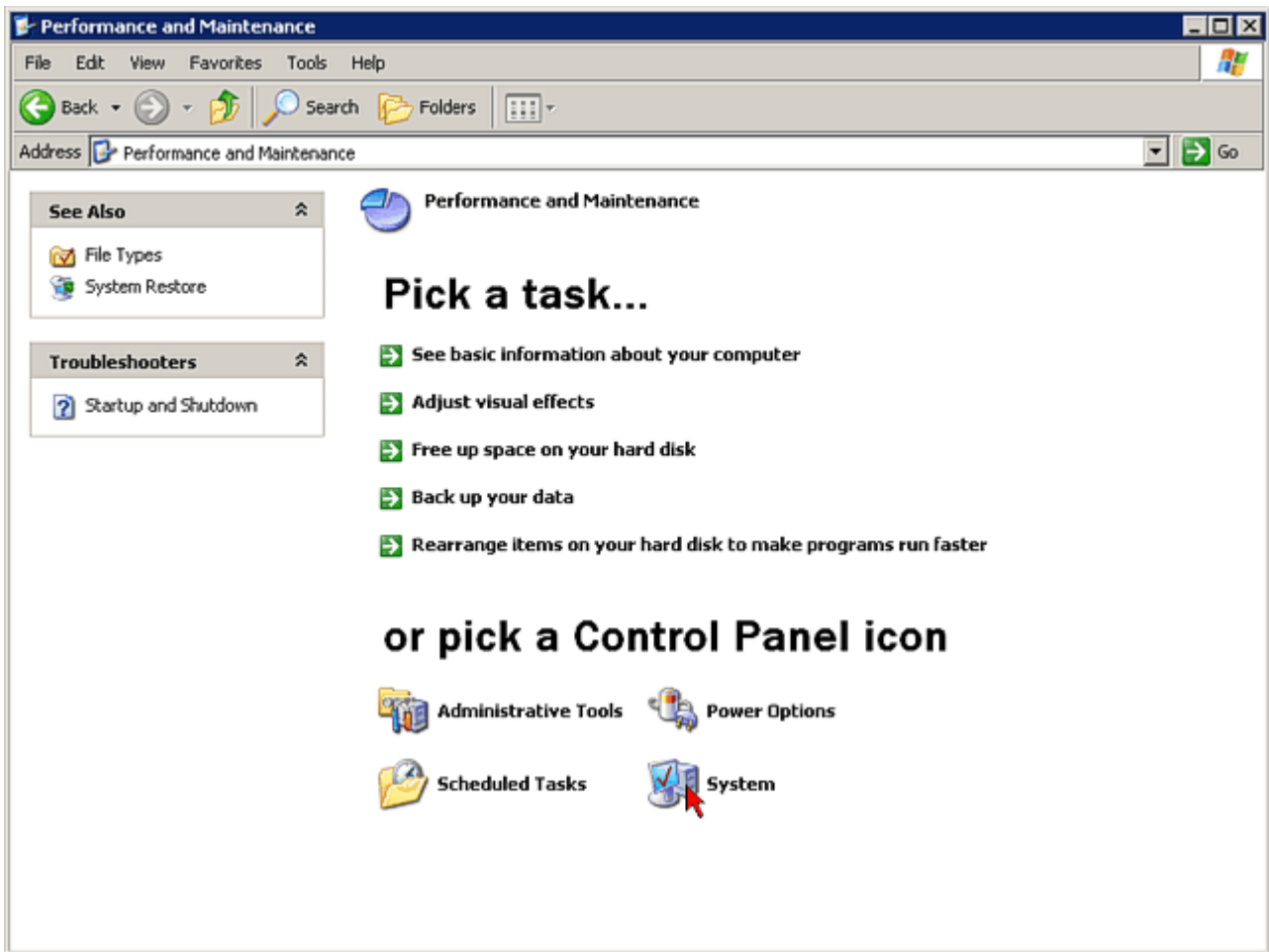


Fig. 146: Select “Service and Maintenance” from “System” in the View Control Panel

- Click on the *Remote* tab, and in the Remote Desktop category select the option *Allow users to establish a remote desktop connection*

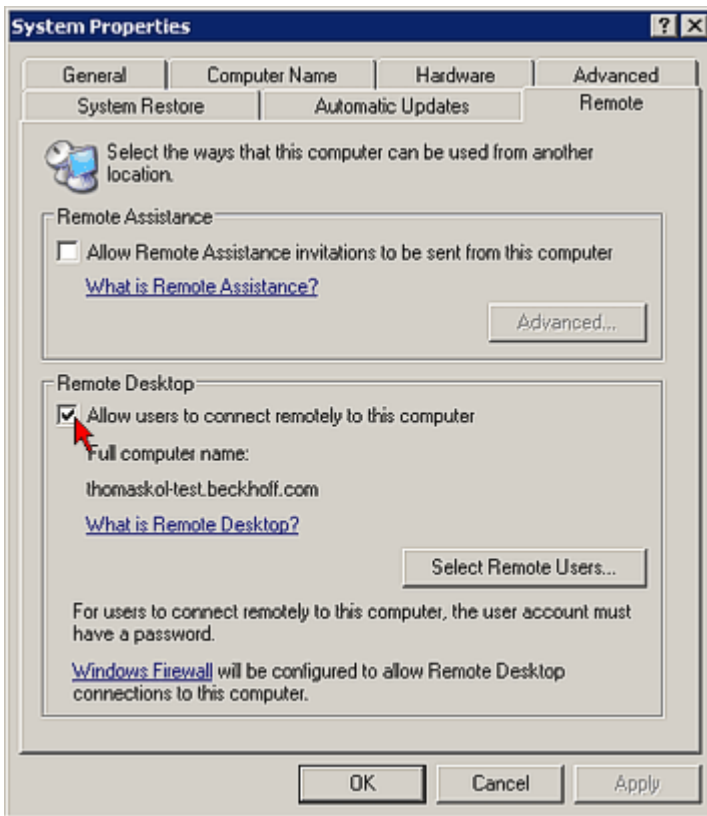


Fig. 147: Check "Allow users to establish a remote desktop connection"

- Open the list of remote desktop users via the button *Select remote users...* and add additional users if required
- Confirm with *OK*.

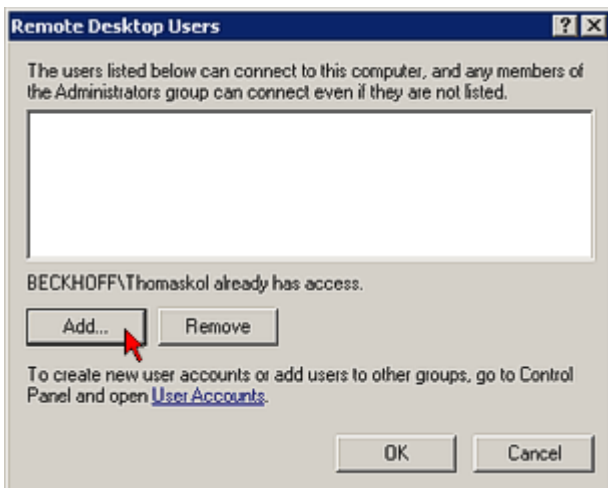


Fig. 148: Adding remote desktop users

Configuration at the service PC

For a remote desktop connection with the TwinCAT master PC via the EL6601 the network address of the service PC has to be set according to the IP number range of the TwinCAT master PC subnet.

- Connect the Ethernet port of the service PC with the EL6601 via a CAT-5 (1:1) cable
- In the Control Panel of the Service-PC check the Ethernet network adapter, via which the EtherCAT system is operated:
Network Connections [Properties] -> Local Area Connection (Ethernet LAN adapter), [Properties]-> Internet Protocol (TCP/IP) [Properties]

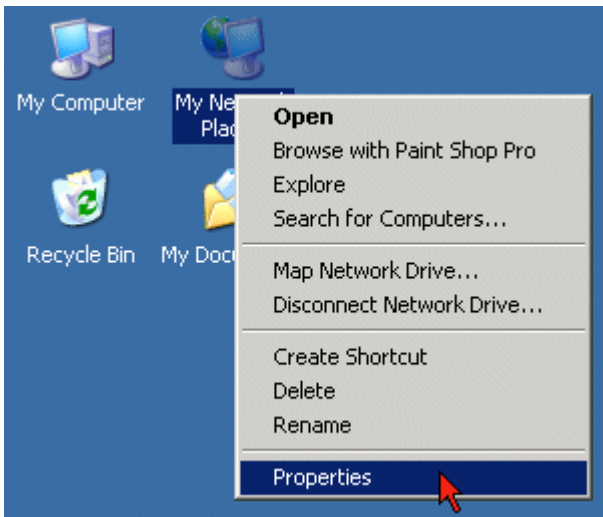


Fig. 149: Context Menu Network Environment -> "Properties"

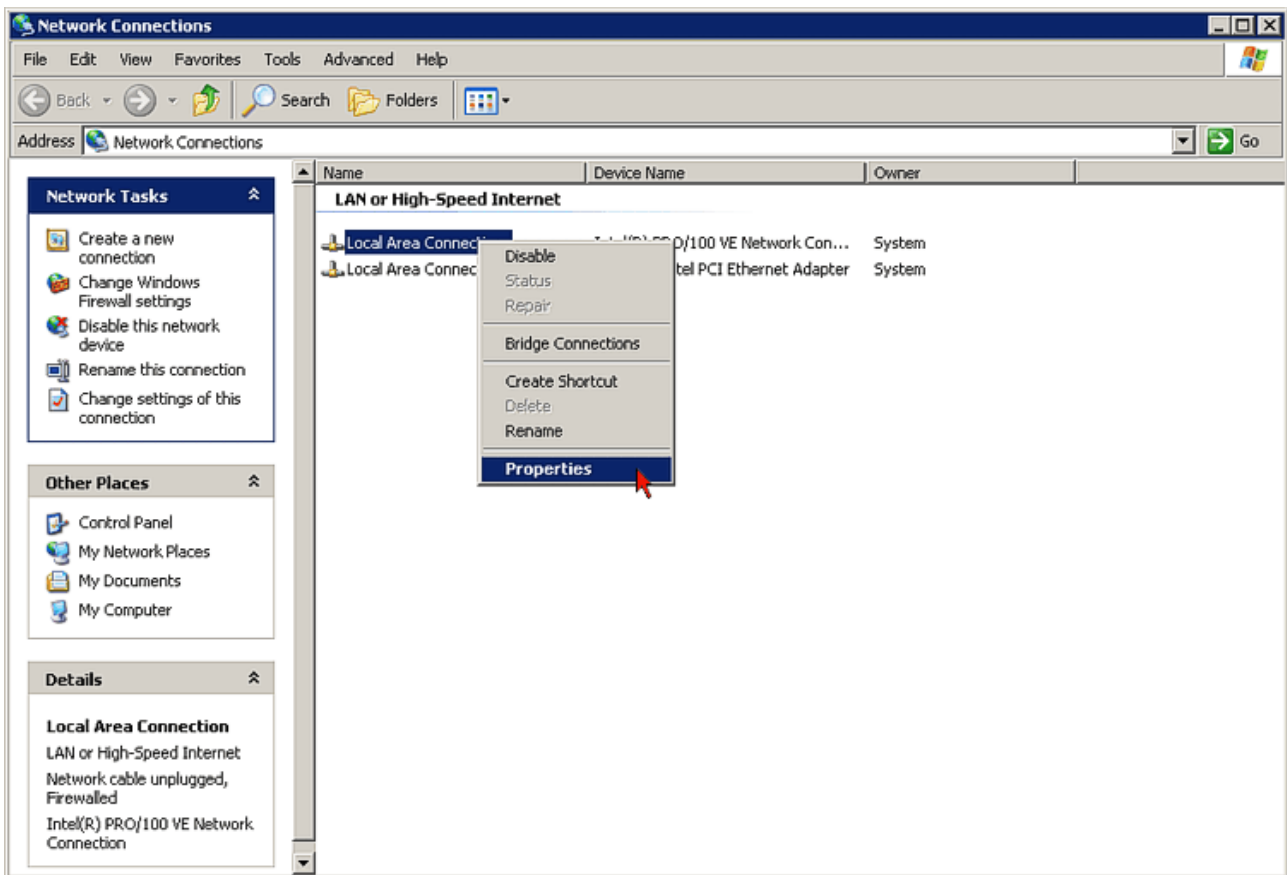


Fig. 150: Context Menu "LAN Connection (Ethernet-LAN-Adapter)," -> "Properties"

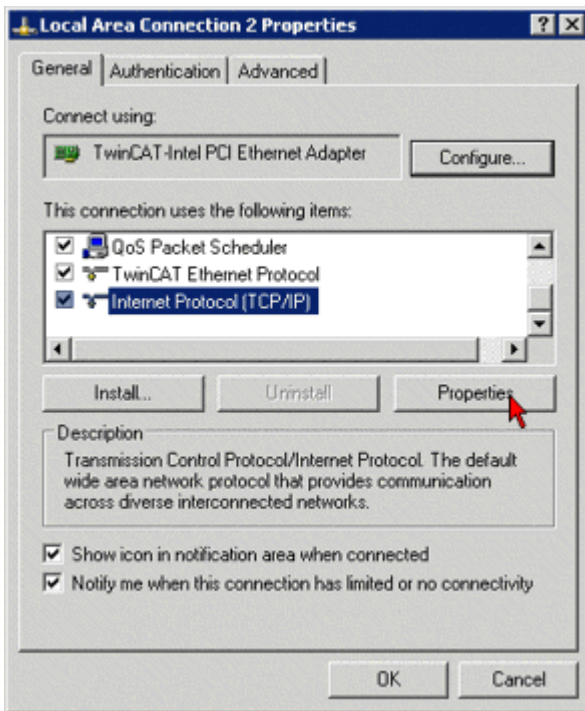


Fig. 151: Properties Internet Protocol (TCP/IP)

- Enter an IP address for the subnet of the network printer (e.g. 192.168.0.38/255.255.255.0) and confirm with *OK*

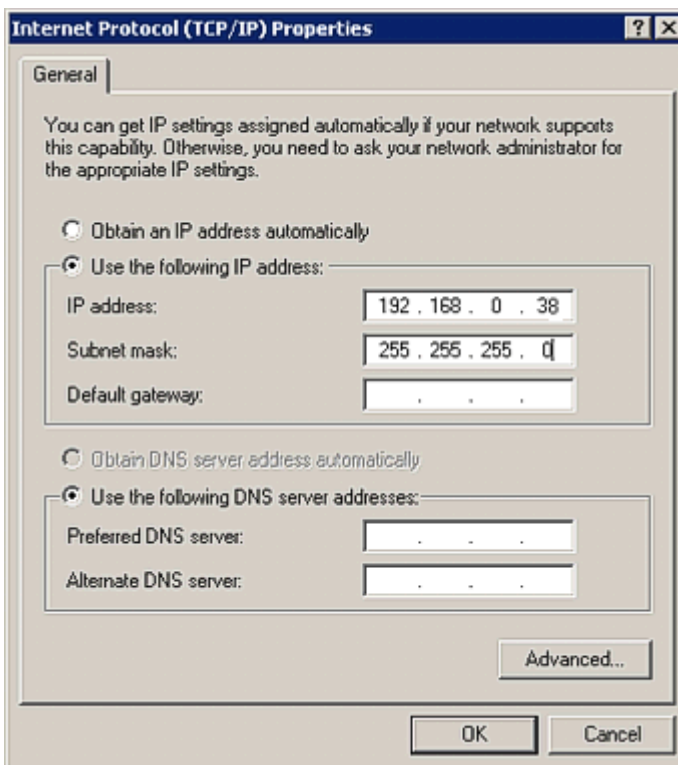


Fig. 152: Entry of the IP address of the subnetwork of the Service-PC

- The Ethernet connection to the TwinCAT master PC is now tunneled via the EL6601 through the EtherCAT terminal network and can be established via the remote desktop.

Establishing the remote desktop connection

- On the service PC start the remote desktop connection via *Start -> Programs -> Accessories -> Communication -> Remote desktop connection*

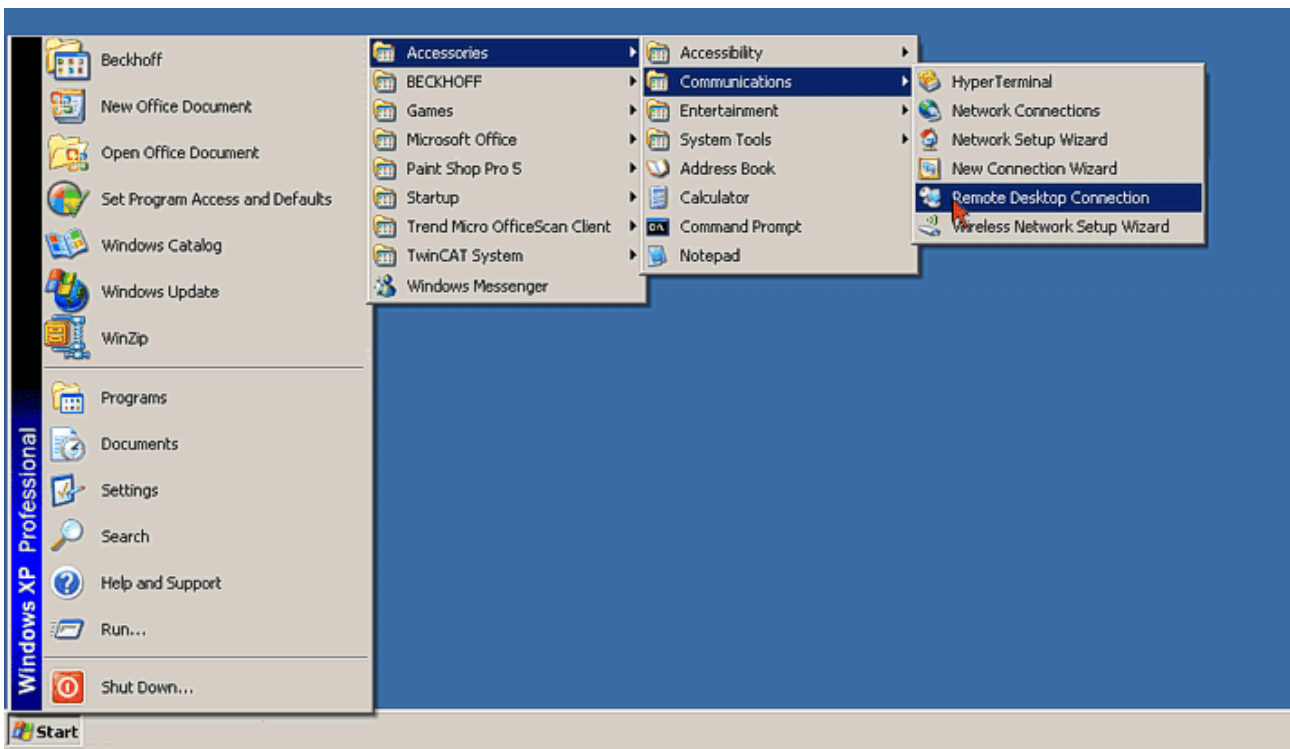


Fig. 153: Start remote desktop connection

- Enter the IP address of the TwinCAT master PC (e.g. 192.168.0.1) and click *Connect*

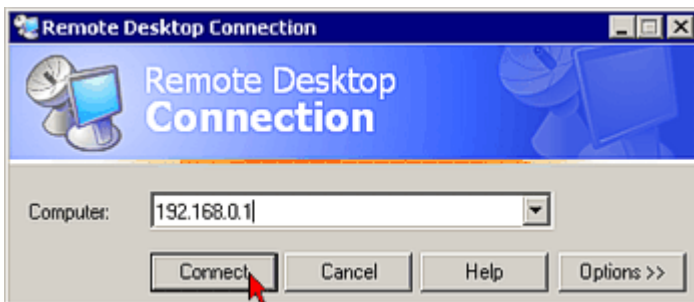


Fig. 154: Entry of the IP address of the Remote-PC

- The login window of the TwinCAT master PC appears on the desktop of the service PC
- Enter your user name and password for the TwinCAT master PC and confirm with *OK*
The IP address of the TwinCAT master PC is shown above the remote desktop

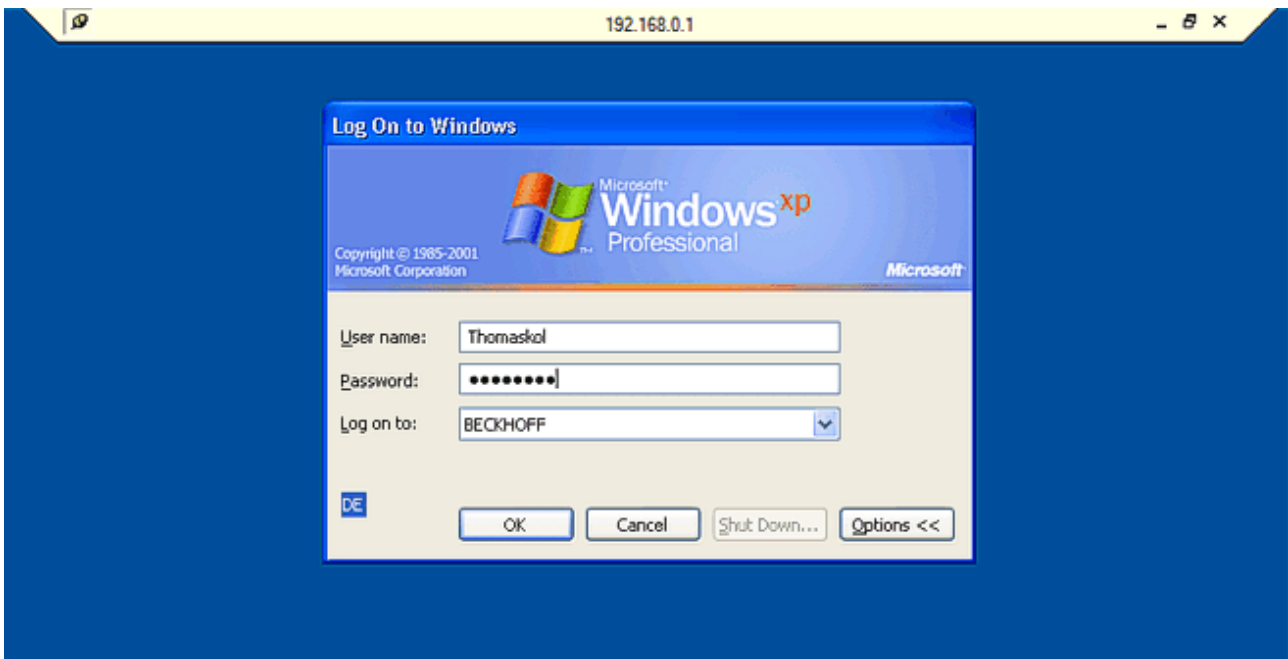


Fig. 155: Login window of the remote PC via the remote desktop connection

● Login via remote desktop

i Users can only log in via a remote desktop if a user name and password have been set up in the TwinCAT master PC!

- The TwinCAT system can now be remotely controlled via the service PC.
To log out close the remote desktop window.

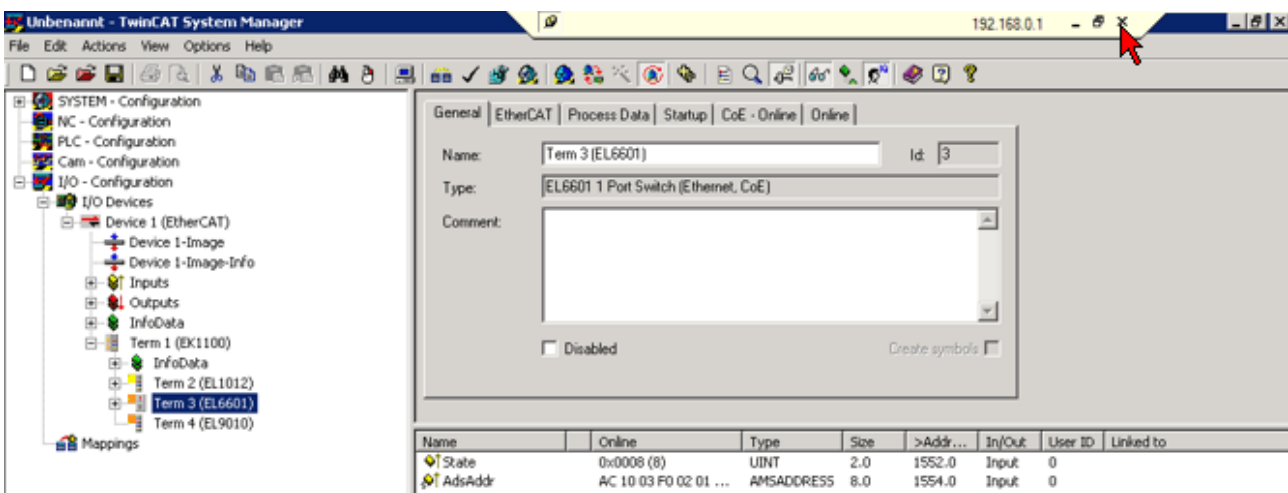


Fig. 156: Closing the remote desktop connection

● Access to the remote system

i When controlling the TwinCAT master PC via a remote desktop TwinCAT must be executed in Run mode or Config mode (free run). If the task is interrupted the service PC can no longer access the remote system!

6.4 Application sample - Lower-level control system

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

The following hardware was used for the test configuration:

- TwinCAT master PC with Windows XP Professional SP 2, TwinCAT version 2.10 (Build 1305) and INTEL PRO/100 VE Ethernet adapter
- Beckhoff Ethernet coupler terminal EK1100, terminals EL1012, EL6601 and EL9010
- Beckhoff Embedded PC CX9000 (subordinate control unit) with Windows CE v5.00, HW 1.4, terminals EL2032, EL9010

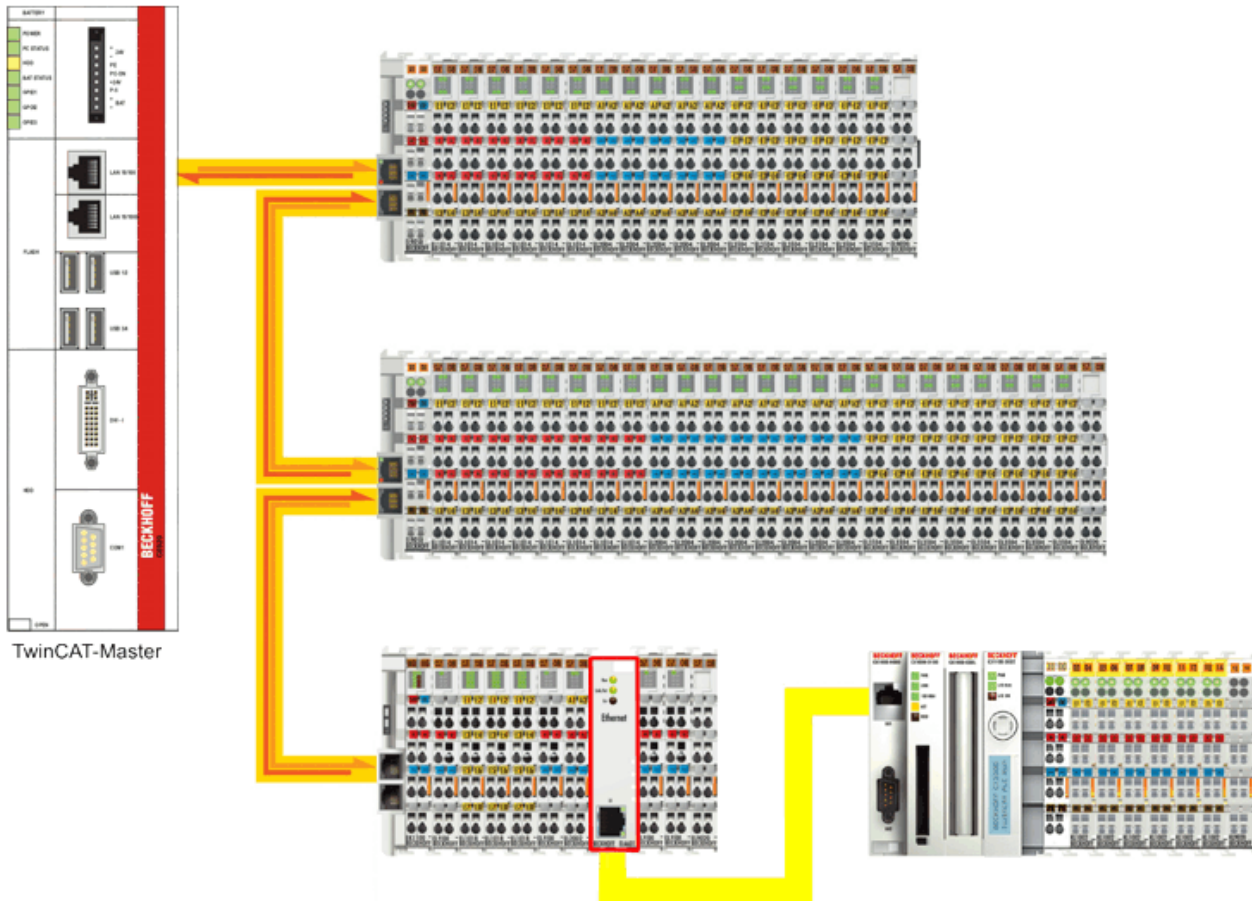


Fig. 157: Integration of a lower-level control system

Configuration of the subordinate control system

For a connection with the TwinCAT master PC via the EL6601 the network address of the subordinate control system has to be set according to the IP number range of the TwinCAT master PC subnet.

- From the desktop of the lower-level control system (Windows CE) choose *Start -> Settings -> Network and Dial-up Connections* to start the configuration of the network settings (Notes on setting the network address of a lower-level controller can also be found in the [EtherCAT system documentation](#) and the [TwinCAT documentation](#))

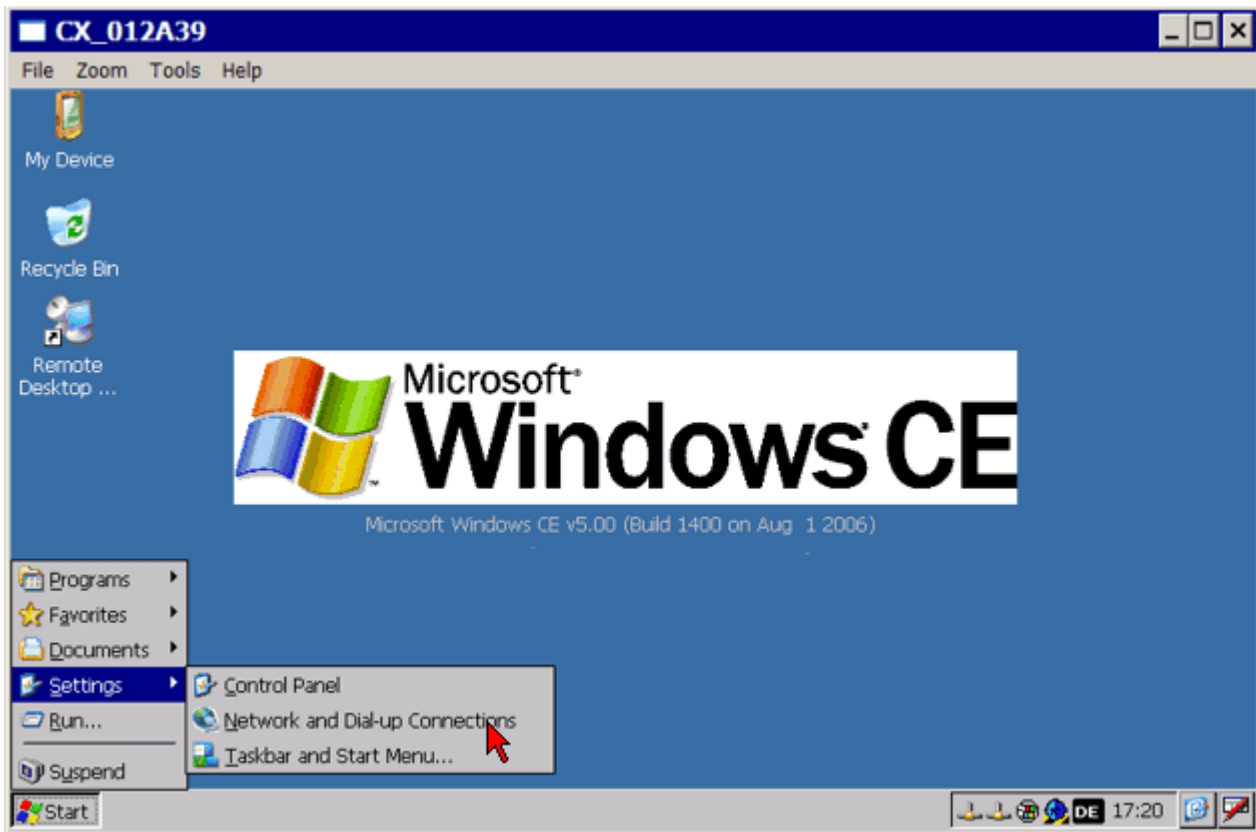


Fig. 158: Select Configuration of network connections

- Open the network settings for this connection by double-clicking on the *TCIXPNPE1* connection (Ethernet port)

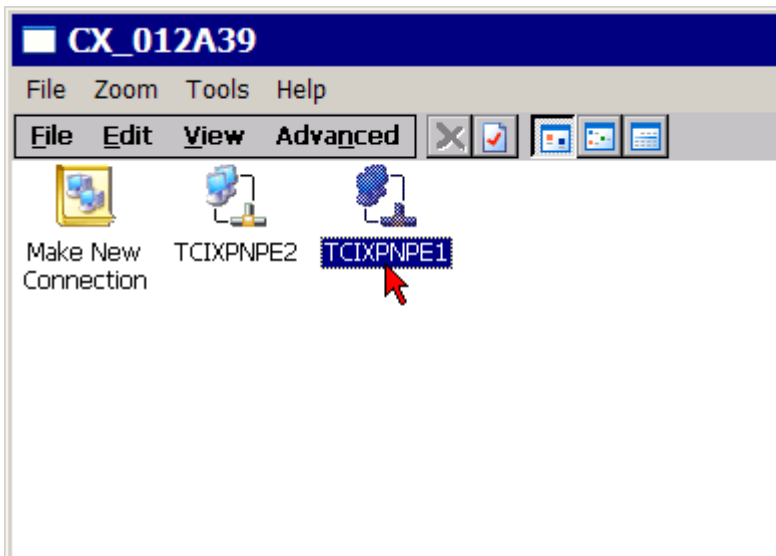


Fig. 159: Select network settings for Ethernet port

- Enter an IP number from the IP number range of the subnet (e.g. 192.168.0.39/255.255.255.0) for the TwinCAT master PC and confirm with *OK*.

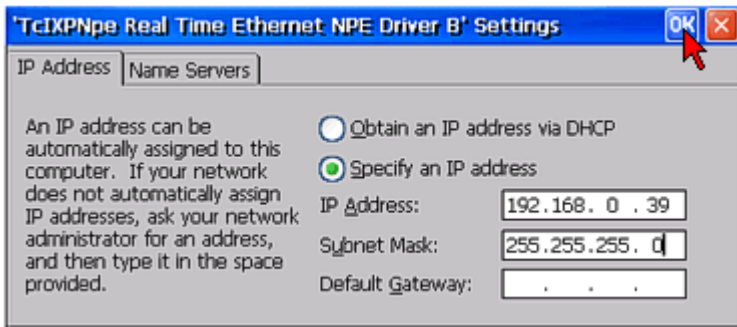


Fig. 160: Enter IP address

Configuration at the TwinCAT master PC and establishing a connection

- Start the TwinCAT System Manager in Config mode, read the Bus Terminal configuration, and activate free-run mode
- The EL6601 should appear in the system configuration being in OP mode (see [Online \[▶ 90\]](#) tab EtherCAT Status Machine)

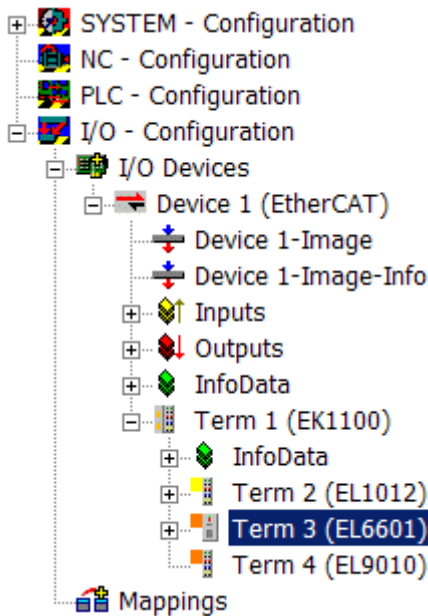


Fig. 161: EL6601 in the configuration

- In analogy to the configuration of a network printer, the IP number of the TwinCAT master PC in the subnet might be 192.168.0.1, for sample
- Connect the subordinate control unit (target system) with the EL6601 via a CAT-5 (1:1) cable
- Start another system manager session in Config mode and click on *Select Target System...-> Search (Ethernet)*

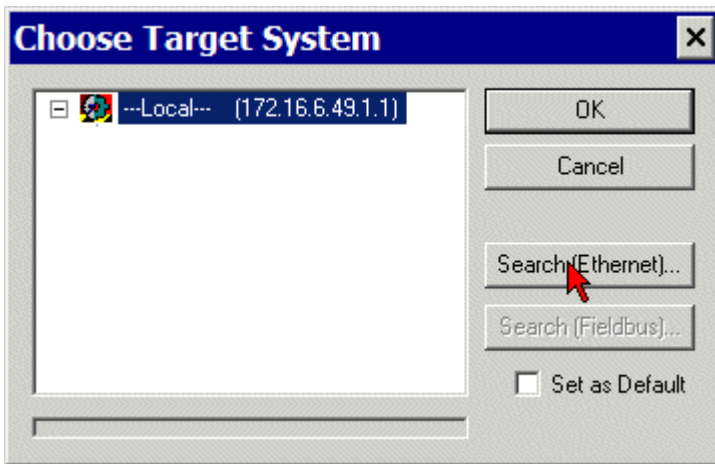


Fig. 162: Select target system

- Enter the IP address of the subordinate control system in the field next to *Enter Host Name / IP:*
- After confirming the button *Enter Host Name / IP:* the name of the target system appears in the selection box

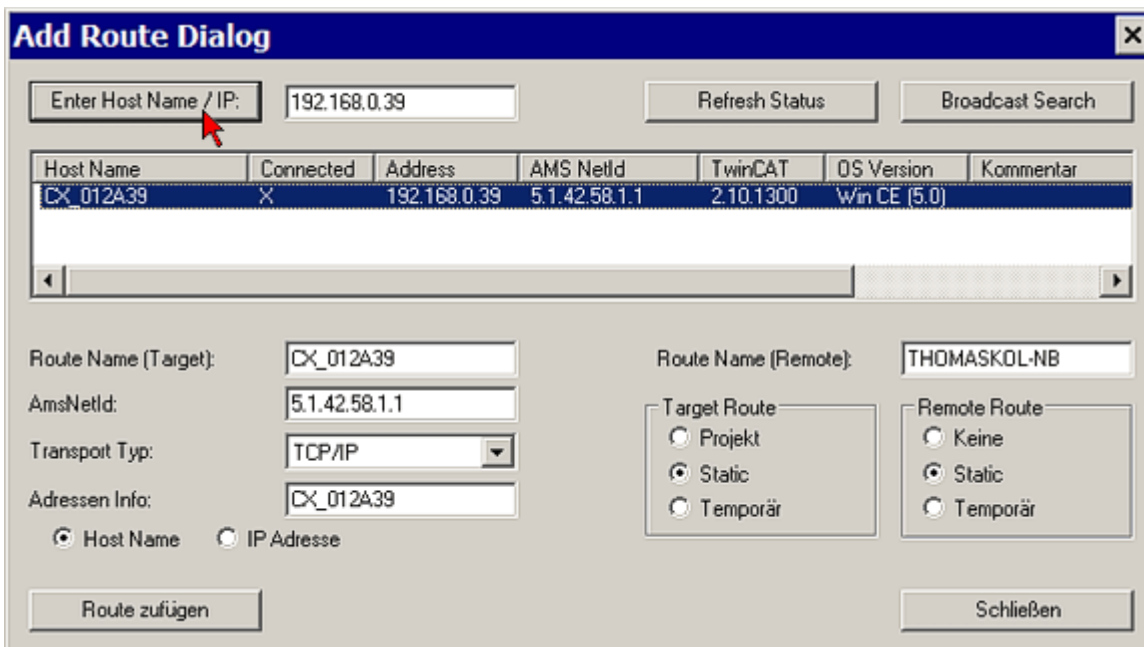


Fig. 163: In the “Route Dialog” add the route of the target system

- Click on *Add Route* and enter the access data for the target system as required
- After closing the dialog box the name of the target system appears in the selection menu
- Make your selection and confirm with *OK*

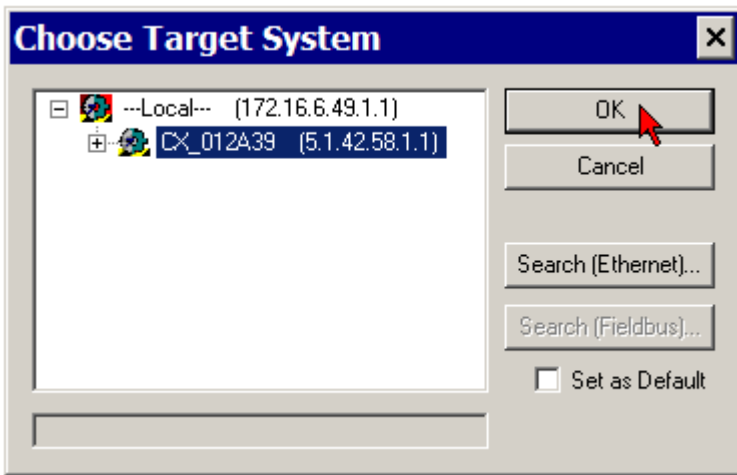


Fig. 164: Confirm selection

- Activate free-run mode for reading the Bus Terminal configuration of the subordinate control system

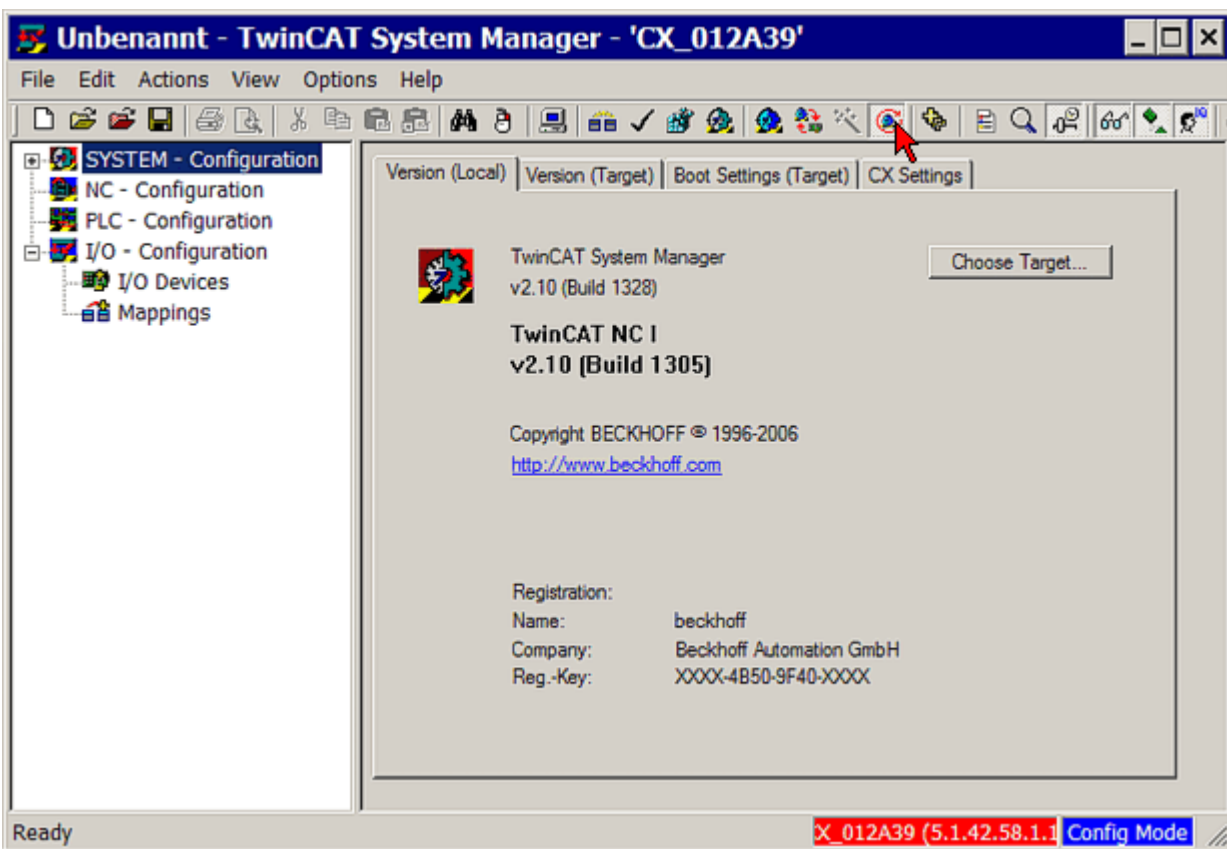


Fig. 165: Enable FreeRun

6.5 Application sample – setting up an EtherCAT Master PC as a network bridge

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

In this sample, transmission is intended to take place from a subordinate PC via the EL6601 and EtherCAT and a Master PC into a superordinate network.

The following hardware was used:

- Computers A and B: Windows XP SP2
- Computer C: Windows XP Embedded

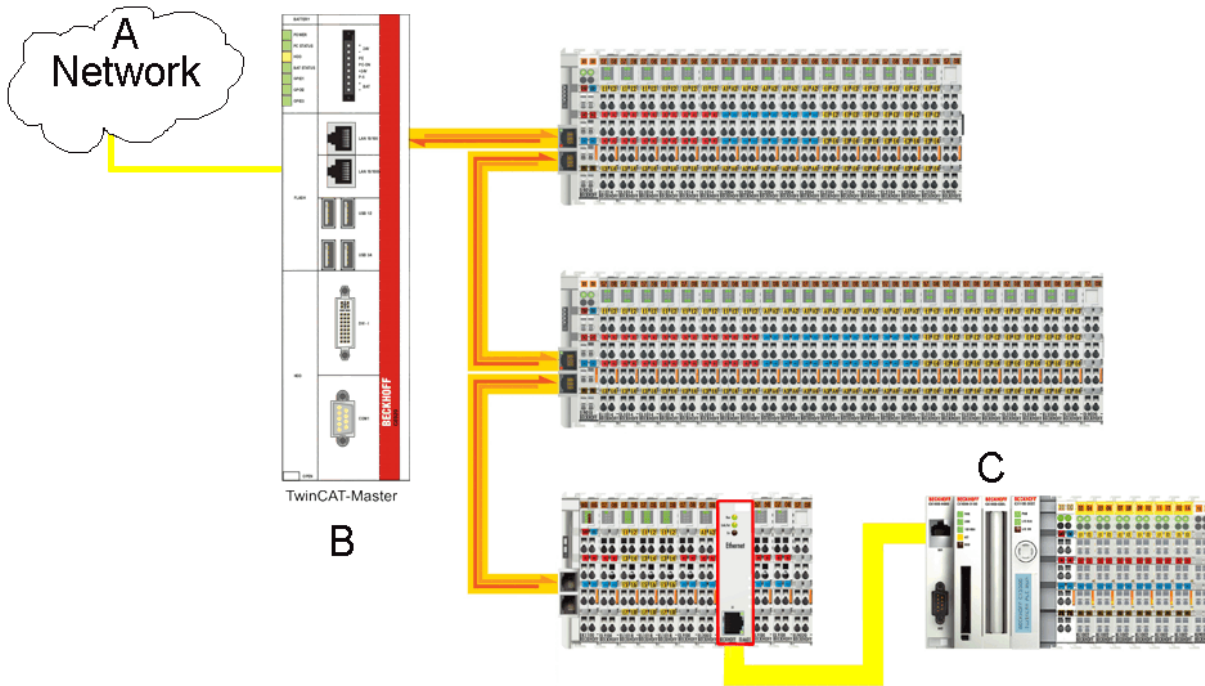


Fig. 166: Setting up an EtherCAT-Master-PC as Network Bridge

To do this, the Master PC "TwinCAT Master" (B) must be setup as a TCP/IP bridge in order to route TCP/IP telegrams through to the subordinate CX1020 (C) or to relay them in the opposite direction. The partner device is a PC (A).

Distinction must be made between two cases here: static addresses [▶ 152] and DHCP address assignment [▶ 156] (network).

The following explanations assume the preceding samples as basic knowledge.

Computers (A), (B) and (C) have a static IP address

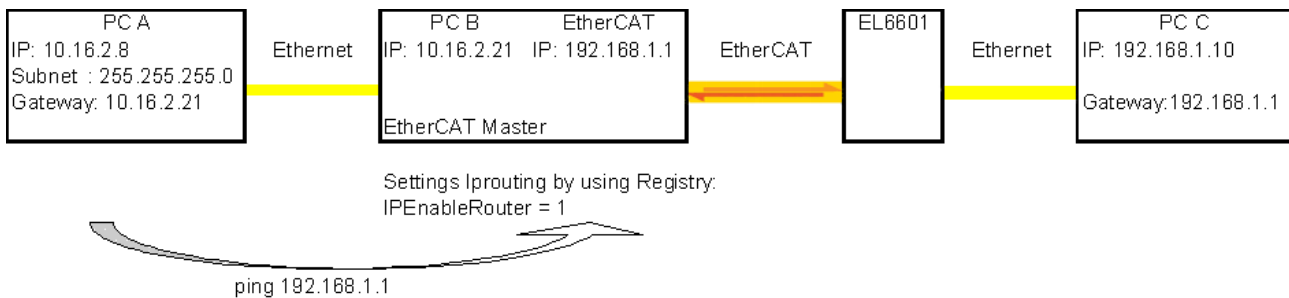


Fig. 167: Configuration of the network PC

- Configure the three PCs as shown in the diagram. The dialogs for the properties of the Internet protocol were explained in the preceding samples.

- Using the registry editor (Start --> Run --> regedit), set the entry "IPEnableRouter" in the "bridge" PC B to the value 1. The entry is usually located under HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\TCPIP\Parameters or also under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TCPIP\Parameters.
- Following a restart, routing for IP telegrams is active for this PC. The connection can then be checked by means of a ping (Start --> Run --> cmd --> ping 192.168.1.1) to the EtherCAT port 192.168.1.1.
- If the EtherCAT master is in a normal and error-free state (TwinCAT in "Config"/"FreeRun" or "Run" mode), the connected Ethernet device with the IP address 192.168.1.10 can be reached via the EL6601.
- Explanation of the entry *Gateway*, using PC A as an sample: PC A has the IP address 10.16.2.8 with a subnet mask 255.255.255.0. This means that it can reach all IP telegrams within a range from 10.16.2.0 to 10.16.2.255. If it is required to send an IP telegram to a different address range (e.g. to 192.168.1.1), this must run via the defined gateway, in this case 10.16.2.21, which in turn can relay the telegram to 192.168.1.10 via the configured IP routing.
- Windows offers the useful command line command *ipconfig* for checking the configuration. If PC A is configured correctly, the command Start Run cmd ipconfig/all displays the following screen:

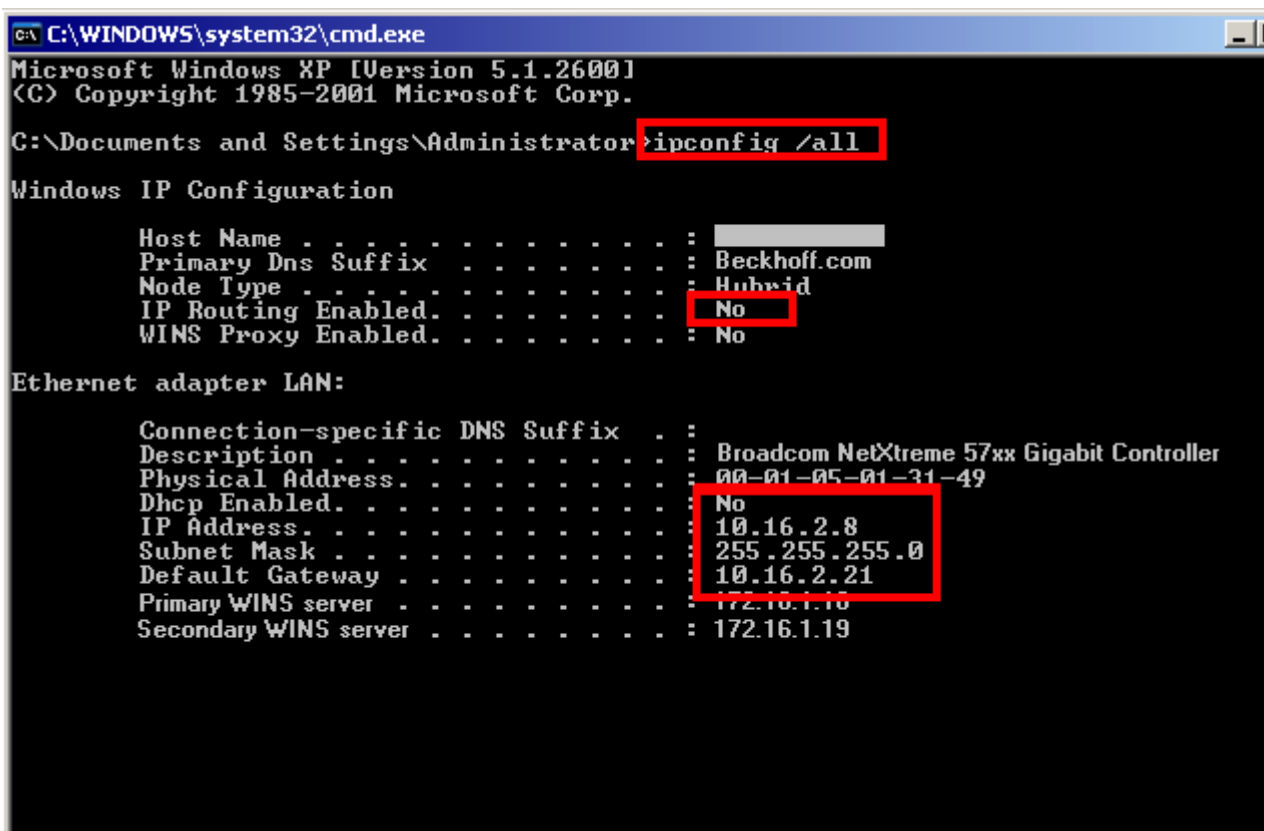


Fig. 168: Checking the configuration with command line command "ipconfig /all"

- IP routing is disabled (default) and the IP address of the selected adaptor (here called LAN) is assigned fixedly. Since DHCP was set beforehand, WINS entries are still present.
- The same command on PC C displays:

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\MartinPodrouschek>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : C6140_MARTINP
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Hybrid
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No

Ethernet adapter FC9004_3:

    Connection-specific DNS Suffix . . :
    Description . . . . . : TwinCAT-Intel PCI Ethernet Adapter #14
    Physical Address. . . . . : 00-01-05-01-05-82
    Dhcp Enabled. . . . . : No
    IP Address. . . . . : 192.168.1.10
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
  
```

Fig. 169: Checking the configuration PC C

- This was entered under XP in the dialog

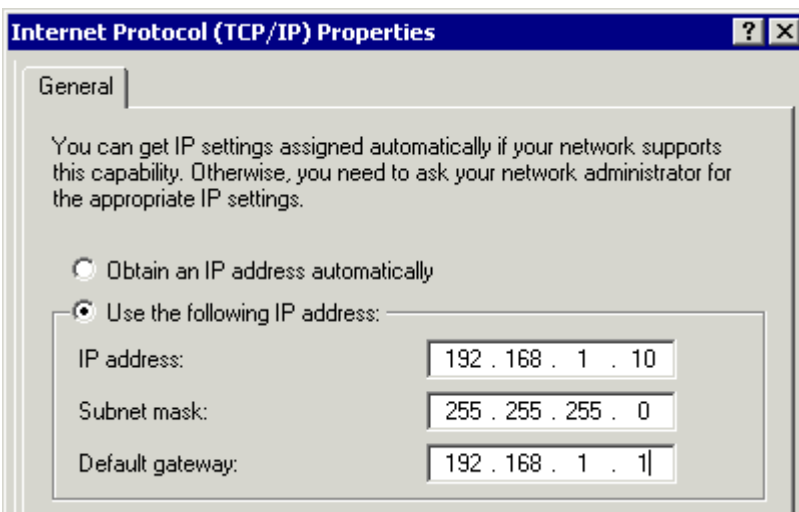


Fig. 170: Entry of network parameters

- Accordingly, the command on the bridge PC B displays:

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : CX_013149
    Primary Dns Suffix . . . . . :
    Mode Type . . . . . : Hubwid
    IP Routing Enabled. . . . . : Yes
    WINS Proxy Enabled. . . . . : No

Ethernet adapter LAN:

    Connection-specific DNS Suffix . . :
    Description . . . . . : TwinCAT-Intel PCI Ethernet Adapter
    Physical Address. . . . . : 00-01-05-01-31-49
    Dhcp Enabled. . . . . : No
    IP Address. . . . . : 10.16.1.21
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Ethernet adapter EtherCAT:

    Connection-specific DNS Suffix . . :
    Description . . . . . : TwinCAT-Intel PCI Ethernet Adapter
    Physical Address. . . . . : 00-01-05-01-31-4A
    Dhcp Enabled. . . . . : No
    IP Address. . . . . : 192.168.1.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

```

Fig. 171: Checking the Configuration Bridge PC B

● "ipconfig" switches

I Using the switches /release and /renew, the command ipconfig can be used to re-establish IP connections.

Checklist for connection settings:

- Bridge PC configured for "IP routing" by means of registry entry
- Fixed IP addresses in correlating ranges for all 4 Ethernet ports involved
- Firewalls disabled if need be (unnecessary in the case of the default setting)
- EtherCAT/TwinCAT in "Freerun" or "Run" mode
- Link display on all ports involved
- "Ping" works

● Observation of network traffic

I Experienced users can observe and evaluate the network traffic using a network sniffer, such as Wireshark.

● Notice on Operating system Windows CE/embedded

I Under Windows CE IP Routing must be set in the CX Configuration Dialog (see Fig. "Setting IP Routing in Windows CE")

The entry "TCP/IP Settings" -> "IP Routing" sets the Registry-Key "IPEnableRouter"= 1 under [HKEY_LOCAL_MACHINE\Comm\Tcpiip\Parms]
Restart required.

See also Microsoft Developer Network: [Routing for IPv4 - Explanation on Routing.](#)

Ordinarily the Routing table is automatically maintained, if necessary routes will still have to be added with "Route Add" command, see also Windows XP/7 etc.

Windows CE does not support some Routing protocols such as RIP or OSPF.

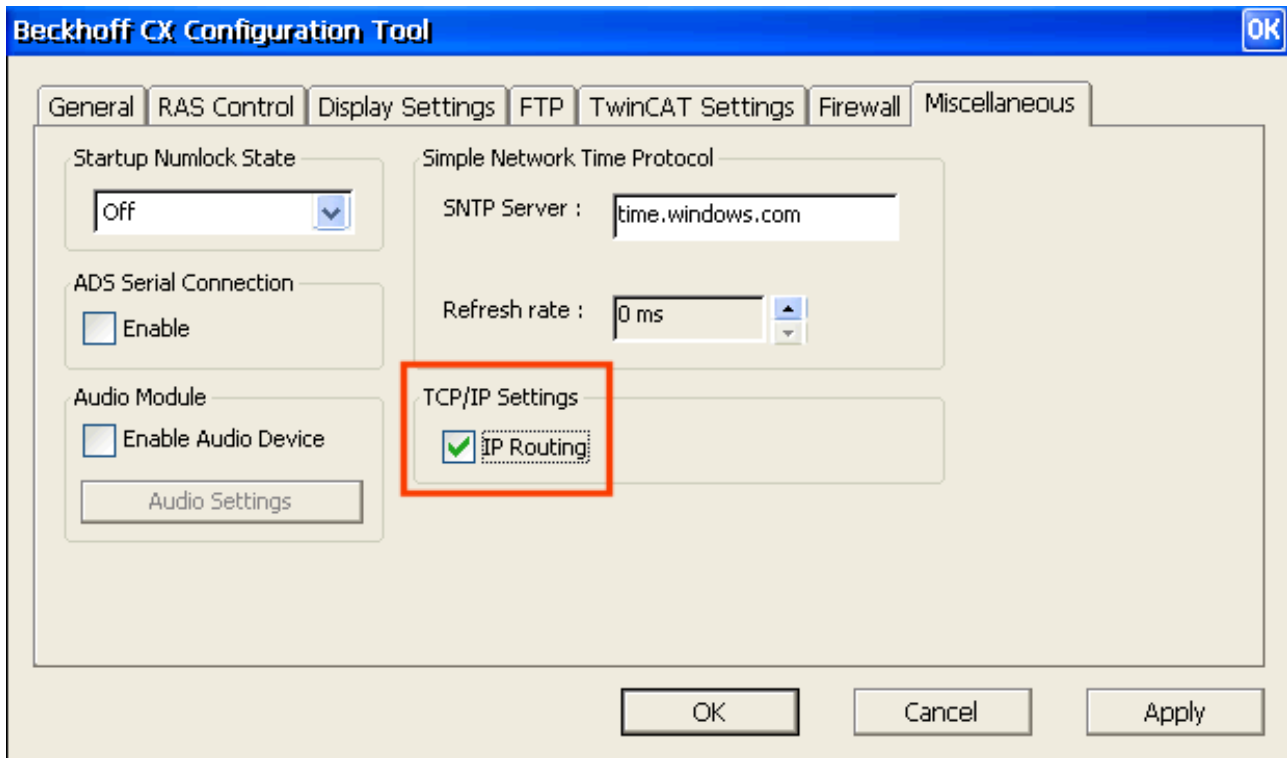


Fig. 172: Setting "IP Routing" in Windows CE

Network with DHCP address assignment

A connection between the PCs C and A (or the network) is not possible.

Explanation: Computers (A) and (B) receive their IP addresses from the DHCP server if the Ethernet ports are configured accordingly. One possible sequence is as follows (see RFC1541 and RFC2131):

- A newly connected PC sends a DHCPDISCOVER message with its MAC address via MAC Broadcast to all other network devices
- The DHCP server offers the requesting device one or more IP addresses in the DHCPOFFER message
- The requesting PC selects an IP address and answers via MAC Broadcast with a DHCPREQUEST message to the DHCP server
- The DHCP server confirms the selection with a DHCPACK message.

This works for computers (A) and (B). The DHCPDISCOVER messages from computer (C) are accepted by the EL6601 and relayed in computer (B) to Windows via the EtherCAT port, but Windows (B) does not route these messages (and all other DHCP telegrams) through to the relaying Ethernet port – this function, known as DHCP Relay Agent, is only available in the server version of Windows.

This is the normal operating case when Windows PCs configured as DHCP devices are linked with each other.

Remedy:

- Install a local DHCP server on computer (B), taking care that no conflict arises with the superordinate network DHCP server
- It is possible that two Windows PCs, which are configured as DHCP devices and connected to each other, will only establish their addresses after a short time (several minutes)

6.6 Application sample - Flexible Ethernet Port

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

This sample illustrates the principle of remote access from a subordinate PC to the central EtherCAT controller via a flexibly connected EL6601 and EtherCAT.

The following elements were used:

- IPC with Windows XP SP2
- TwinCAT 2.11 b1534
- EL6601, SN xxxx0605

The approach

The task is to commission an extensive system with EtherCAT topology. To this end frequent access to the TwinCAT System Manager on the central PC is required. In practice the technician will primarily be at the local terminals/EtherCAT slaves during commissioning. The central System Manager as the TwinCAT target system should therefore be accessed from the respective coupler location via a remote PC and Ethernet.

Two mechanisms are used for this purpose:

- EtherCAT tunnels standard IP telegrams through dedicated slaves, e.g. EL6601, via the acyclic mailbox procedure
- Via the HotConnect functionality a coupler station (coupler + terminals) can be classified as freely pluggable in the configuration (.tsm). Special couplers must be used for this purpose.

A freely pluggable combination of EK1101 and EL6601 (HotConnect group) can now be used as a flexible Ethernet interface.

Possible configuration

A possible configuration is described below:

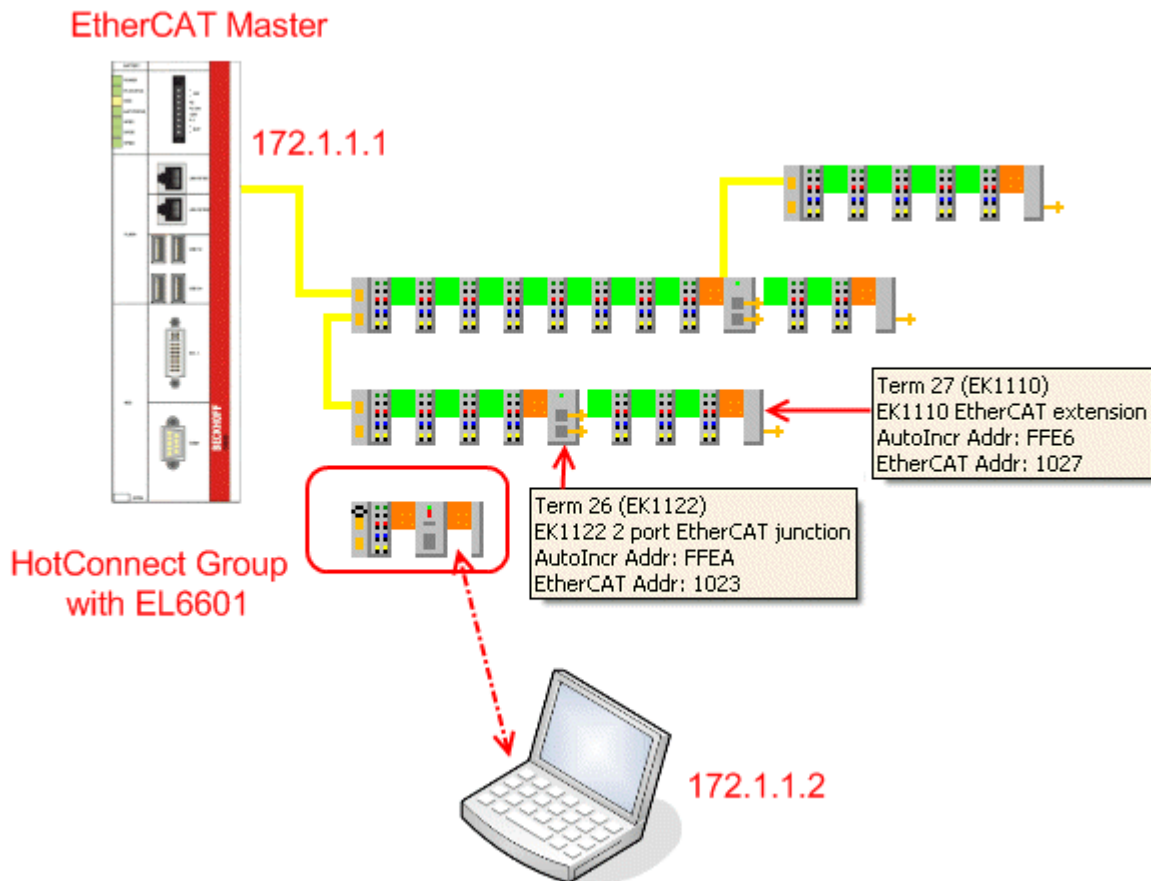


Fig. 173: Structure HotConnect Group

The system offers EK1110 (end couplers), EK1122 (junction terminals) and free EK1100 ports for connecting the HotConnect group. The IP addresses should be regarded as samples, the notes from the previous samples must be taken into account.

No further settings are required at the TwinCAT/EtherCAT master. EoE is automatically forwarded in TwinCAT 2.11.

● Remote access operability and EtherCAT state

i The process of Ethernet pass-through described here only works as long as the EtherCAT master and the EL66xx are at least in state PREOP. Otherwise no mailbox traffic and therefore no Ethernet transport takes place. However, if the network structure is already functional, this does not represent an insurmountable obstacle for general commissioning.

● Change of location and EtherCAT state

i Once the HotConnect group is reconnected with the EtherCAT network it may take several seconds for the group to participate in the data traffic again, the EL66xx to be in OP state again and the link (see link LED) to be restored.

Sample configuration

A small EtherCAT system is commissioned, the offline topology view in the System Manager shows two coupler stations and one non-localized station with an EK1101.

- In offline state the System Manager does not know where the EK1101 station with its ID was connected.

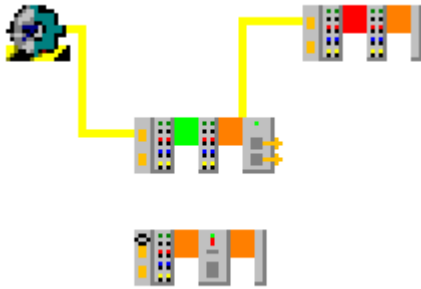


Fig. 174: EK1101 localized with EL6601

- The current assignment is visible online, the EK1101 with the EL6601 is connected to the second port of the EK1100.

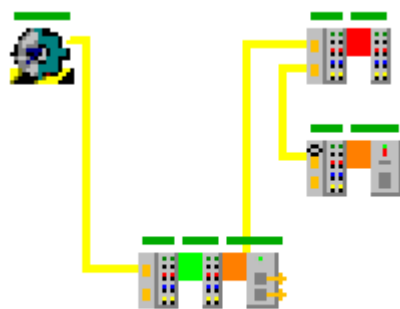


Fig. 175: EK1101 localized with EL6601

- All slaves are in OP state, so that this system can now be accessed as target system from the remote PC.
- In the remote system an empty System Manager is opened and the EtherCAT computer with the IP 172.1.1.1 (in this sample) is located as the target system via IP (Tab Actions -> Choose Target System -> "Search (Ethernet)" -> [enter IP 172.1.1.1] -> "Enter Host Name / IP")

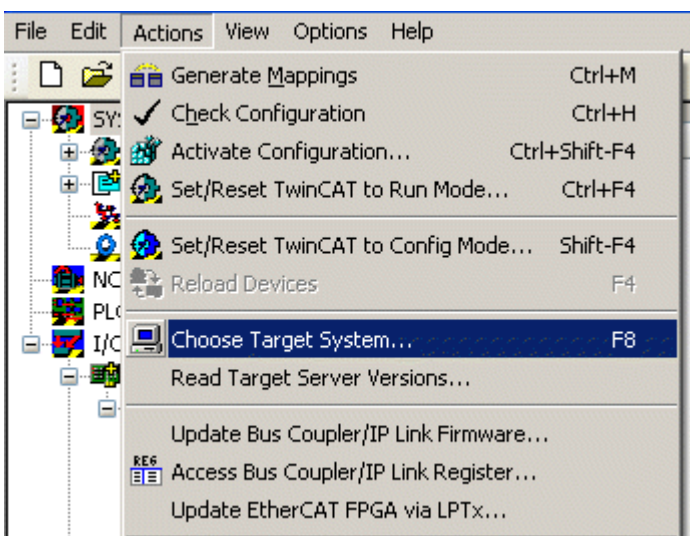


Fig. 176: Selection of the target system

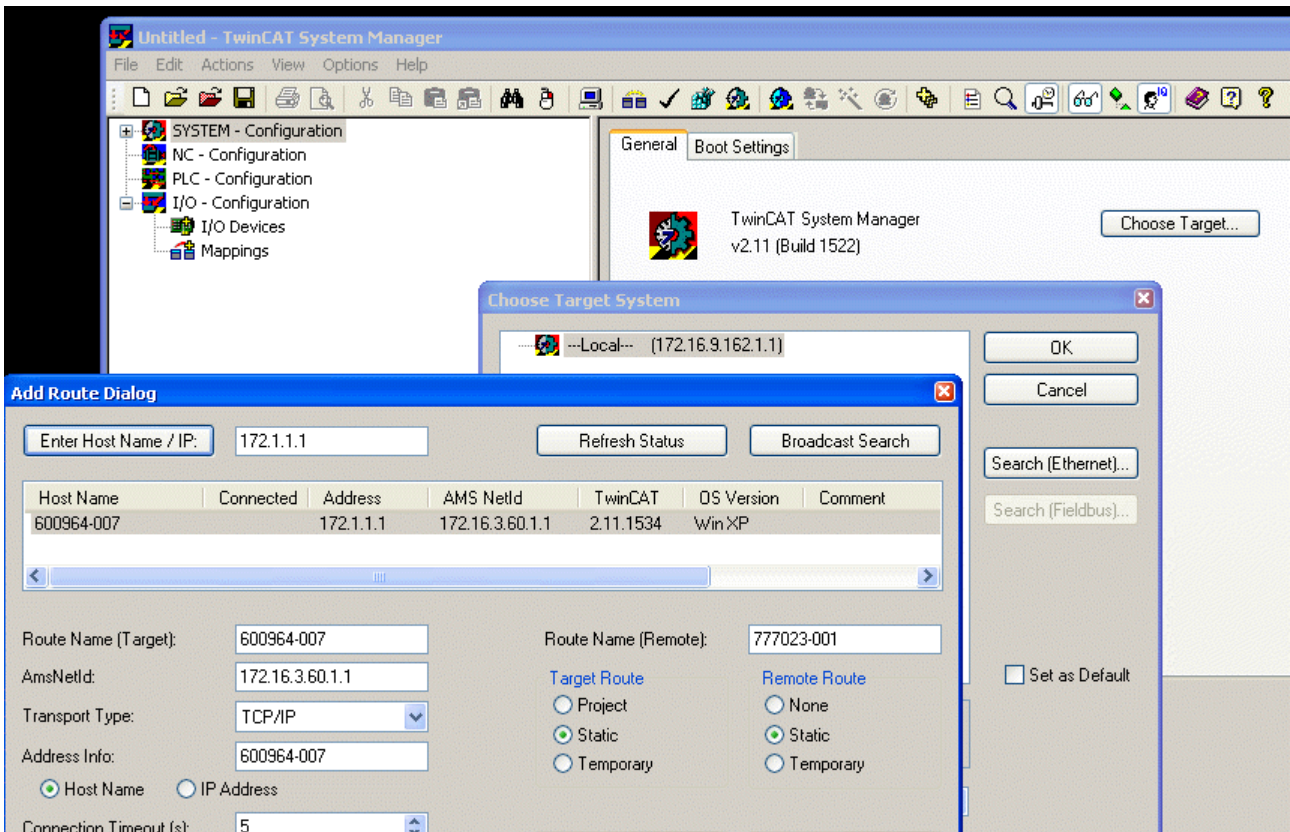


Fig. 177: Enter target system 172.1.1.1

- After double-clicking on the line showing the found target system the user can log into the target system:

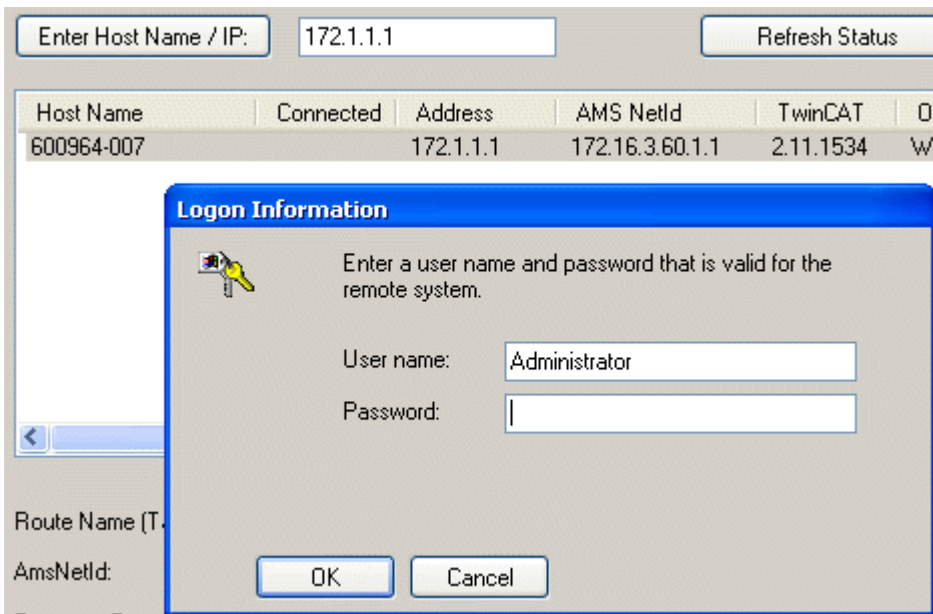


Fig. 178: Entry of logon information on target system

- After successful connection the configuration can be loaded by the target system through "Open from target system".

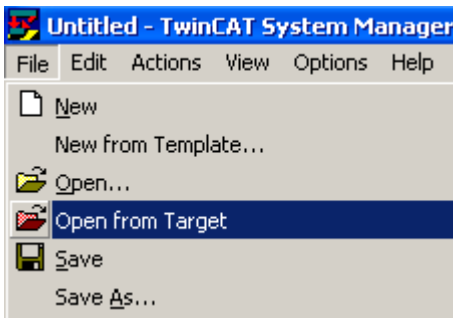


Fig. 179: Opening of target system and loading of configuration

Checklist

The following checklist can be used for a successful configuration up to remote access:

1. Successful EtherCAT configuration
 - no LostFrames
 - EtherCAT master and all slaves in OP state
 - all WorkingCounters = 0; if necessary, set up special SyncUnits for deliberately inactive slaves
2. Set a different IP address for the EtherCAT port and the port used at the laptop, but within the same subnet section.
(here: 172.1.1.1 and 172.1.1.2)
3. Check the connection from both sides through ping commands
4. Establish connection as target system

● Ethernet connection

i It is advisable to use credible addresses in the Properties dialog of the network interface, IP protocol (Windows XP SP2) Default gateway and DNS server, although they are not actually required. Settings of the remote PC from the sample:

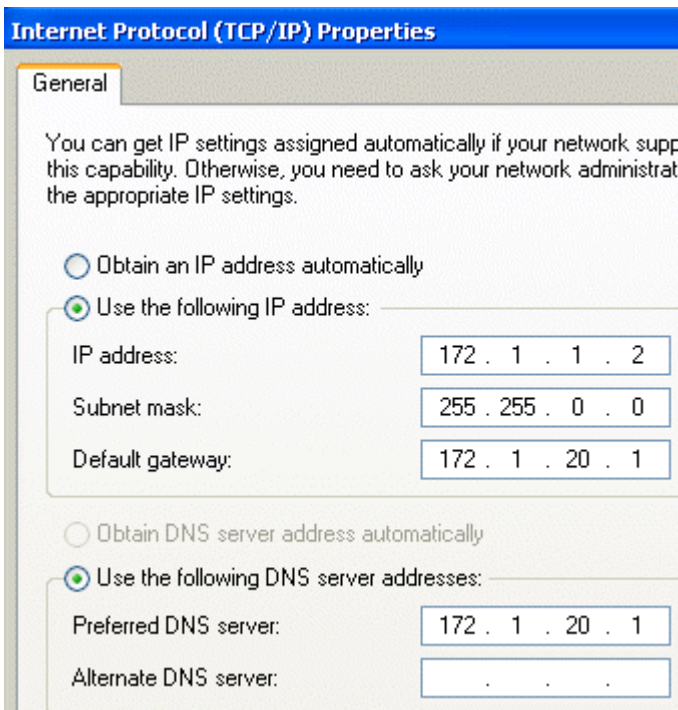


Fig. 180: Entry of network addresses in the remote system

7 Appendix

7.1 Firmware compatibility

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

Note

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

NOTICE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the [separate page \[▶ 164\]](#).

If a device is placed in BOOTSTRAP mode for a firmware update, it does not check when downloading whether the new firmware is suitable.

This can result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

EL6601			
Hardware (HW)	Firmware (FW)	Revision no.	Date of release
05 - 14	06		2009/07
	07	EL6601-0000-0017	2009/09
	08		2009/11
	09		2010/07
		EL6601-0000-0018	2010/09
	10		2011/09
		EL6601-0000-0019	2012/10
	11		2013/04
	12	EL6601-0000-0020	2014/07
15 – 21*	13	EL6601-0000-0021	2014/12
	14		2015/07
	15		2017/02
	16		2023/06
	17*		2023/09

EL6614			
Hardware (HW)	Firmware (FW)	Revision no.	Date of release
00 - 09*	01	EL6614-0000-0017	2008/05
	02		2008/12
	03		2009/08
	04		2009/11
	05		2010/07
		EL6614-0000-0018	2010/09
	06		2011/09
		EL6614-0000-0019	2012/10
	07		2013/04
10 – 17*	08	EL6614-0000-0020	2014/07
	09	EL6614-0000-0021	2014/12
	10		2015/07
	11		2017/02
	12		2023/06
	13*		2023/09

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

7.2 Firmware Update EL/ES/EM/ELM/EP/EPP/ERPxxxx

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

NOTICE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the [Beckhoff website](#).

To update the firmware, TwinCAT can be operated in the so-called FreeRun mode, a paid license is not required.

The device to be updated can usually remain in the installation location, but TwinCAT has to be operated in the FreeRun. Please make sure that EtherCAT communication is trouble-free (no LostFrames etc.).

Other EtherCAT master software, such as the EtherCAT Configurator, should not be used, as they may not support the complexities of updating firmware, EEPROM and other device components.

Storage locations

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

- Each EtherCAT slave has a device description, consisting of identity (name, product code), timing specifications, communication settings, etc.
This device description (ESI; EtherCAT Slave Information) can be downloaded from the Beckhoff website in the download area as a [zip file](#) and used in EtherCAT masters for offline configuration, e.g. in TwinCAT.
Above all, each EtherCAT slave carries its device description (ESI) electronically readable in a local memory chip, the so-called **ESI EEPROM**. When the slave is switched on, this description is loaded locally in the slave and informs it of its communication configuration; on the other hand, the EtherCAT master can identify the slave in this way and, among other things, set up the EtherCAT communication accordingly.

NOTICE

Application-specific writing of the ESI-EEPROM

The ESI is developed by the device manufacturer according to ETG standard and released for the corresponding product.

- Meaning for the ESI file: Modification on the application side (i.e. by the user) is not permitted.
- Meaning for the ESI EEPROM: Even if a writeability is technically given, the ESI parts in the EEPROM and possibly still existing free memory areas must not be changed beyond the normal update process. Especially for cyclic memory processes (operating hours counter etc.), dedicated memory products such as EL6080 or IPC's own NOVDRAM must be used.

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in *.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.

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

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

Simplified update by bundle firmware

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

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxxx-xxxx_REV0016_SW01.efw

- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

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

NOTICE

Risk of damage to the device!

- ✓ Note the following when downloading new device files

- 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.

7.2.1 Device description ESI file/XML

NOTICE

Attention regarding update of the ESI description/EEPROM

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

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

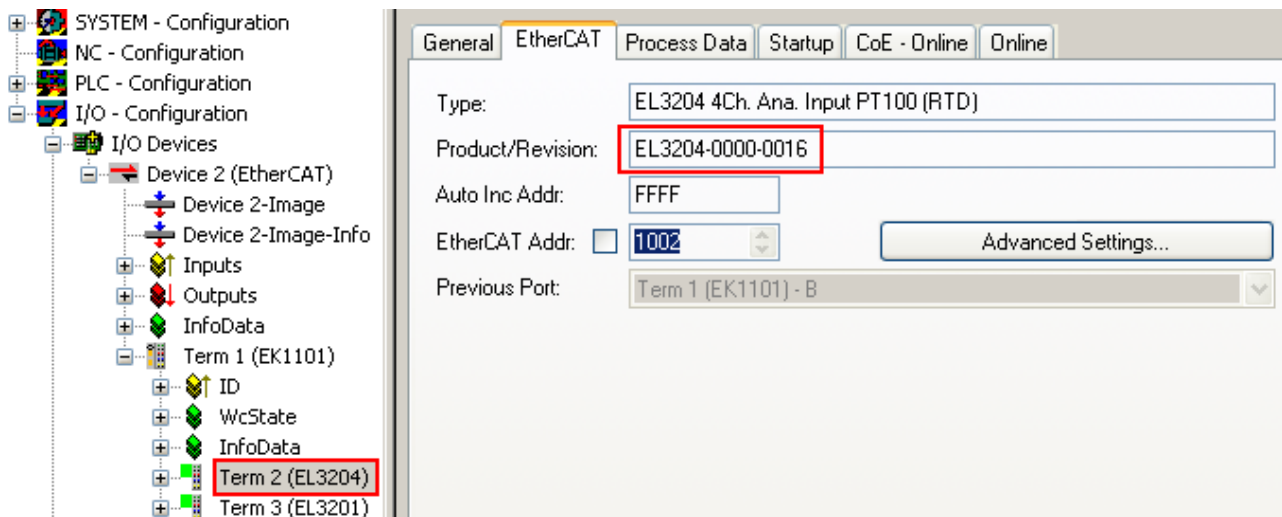


Fig. 181: Device identifier consisting of name EL3204-0000 and revision -0016

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](#).

i Update of XML/ESI description

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

Display of ESI slave identifier

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

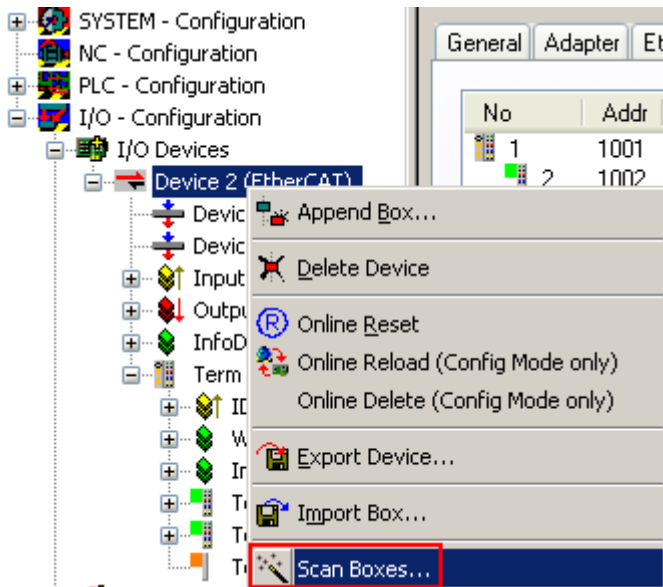


Fig. 182: Scan the subordinate field by right-clicking on the EtherCAT device

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



Fig. 183: Configuration is identical

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

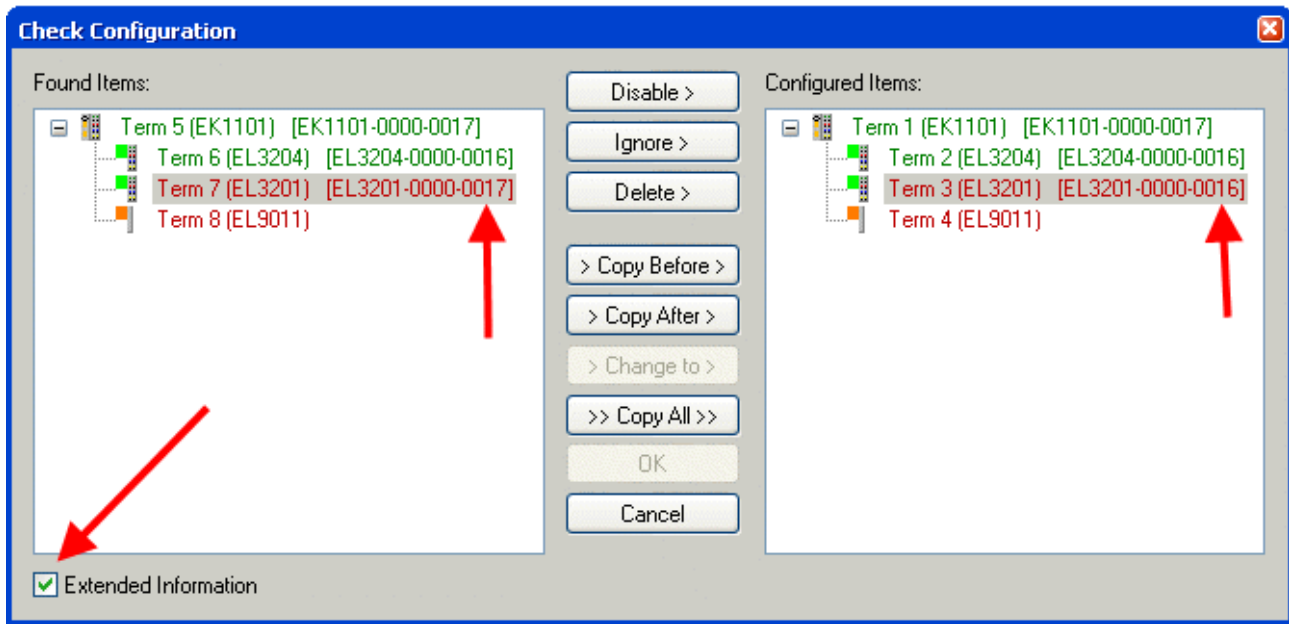


Fig. 184: Change dialog

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*

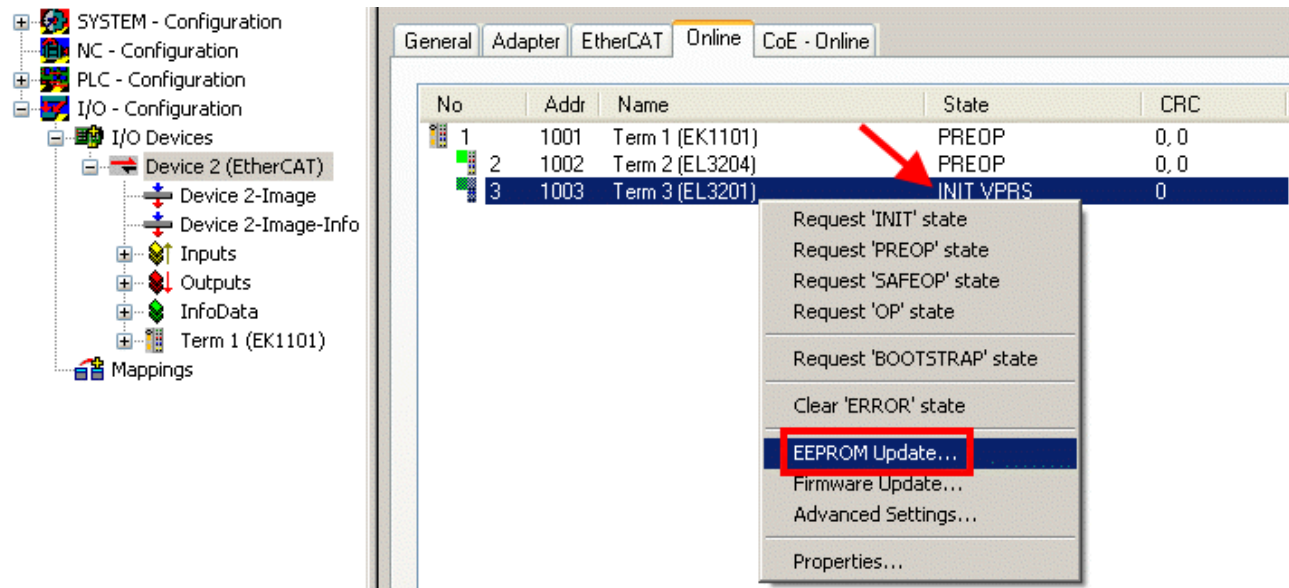


Fig. 185: EEPROM Update

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.

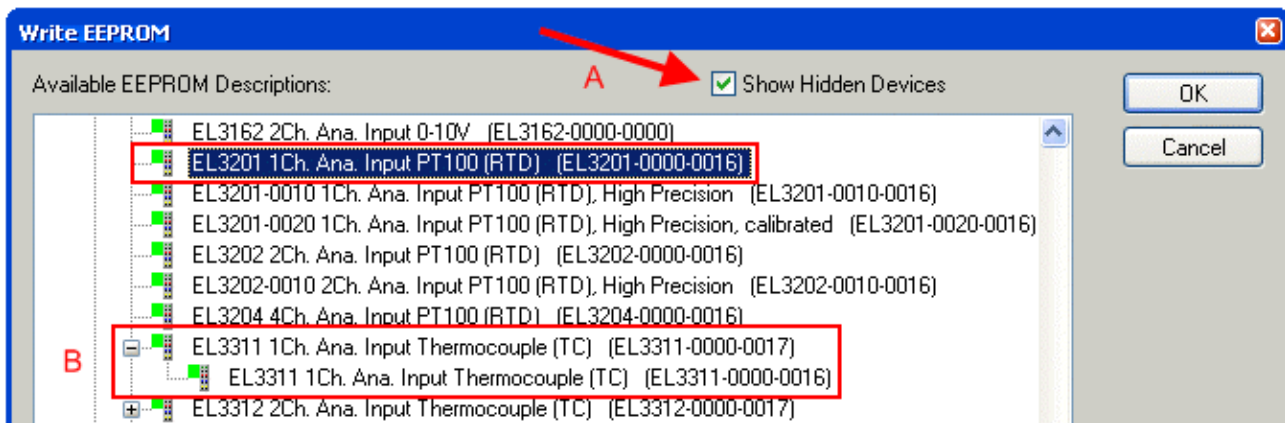


Fig. 186: Selecting the new ESI

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.

7.2.2 Firmware explanation

Determining the firmware version

Determining the version via the TwinCAT System Manager

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

i **CoE Online and Offline CoE**

Two CoE directories are available:

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

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

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

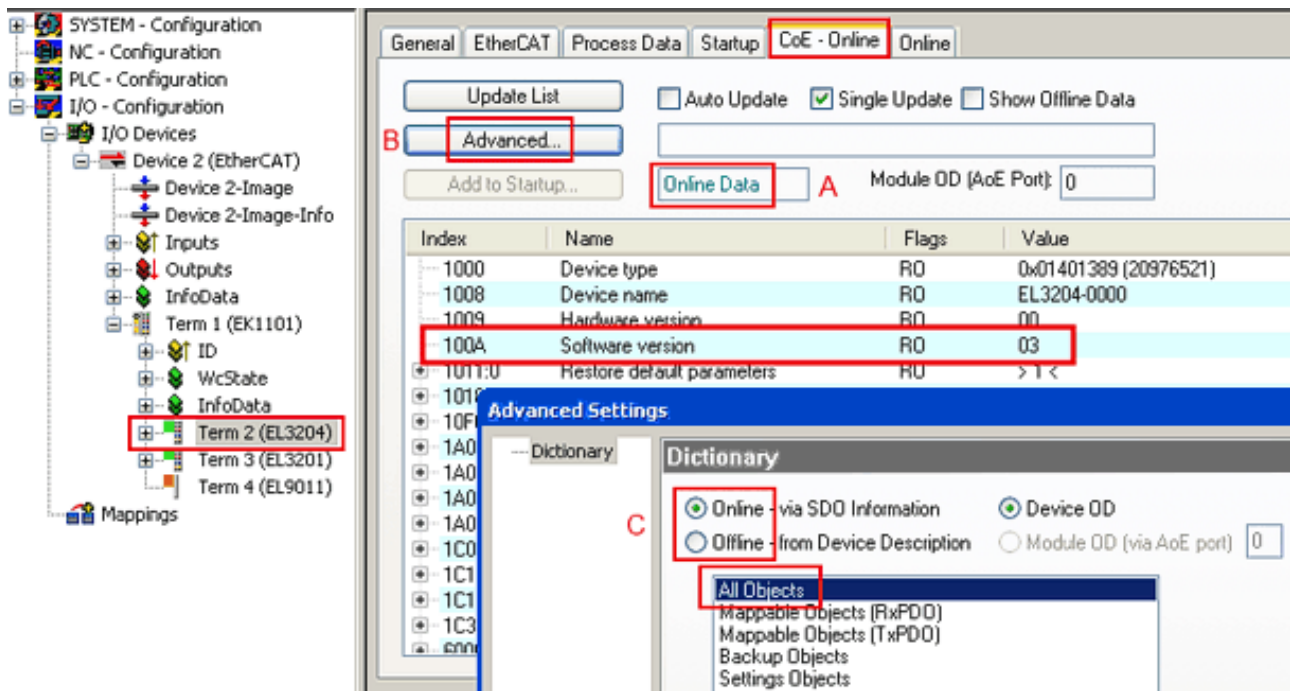


Fig. 187: Display of EL3204 firmware version

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

7.2.3 Updating controller firmware *.efw

● CoE directory

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

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

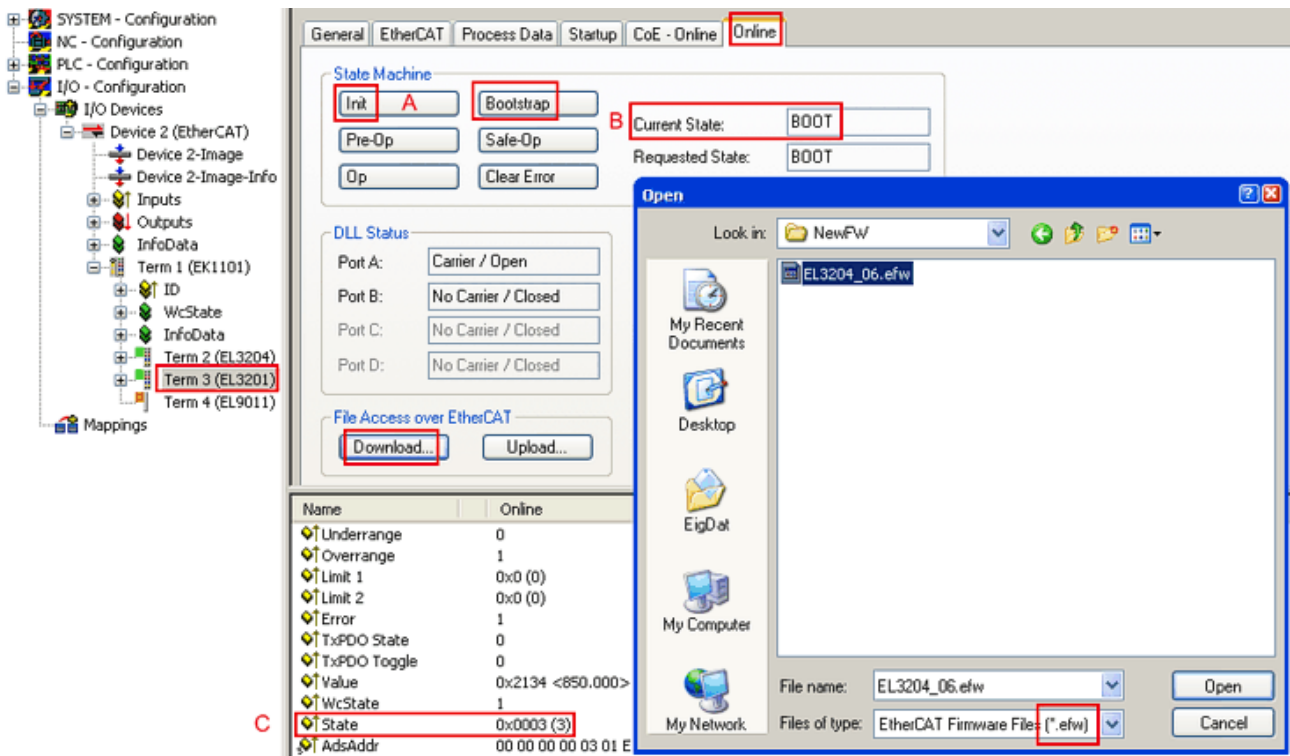
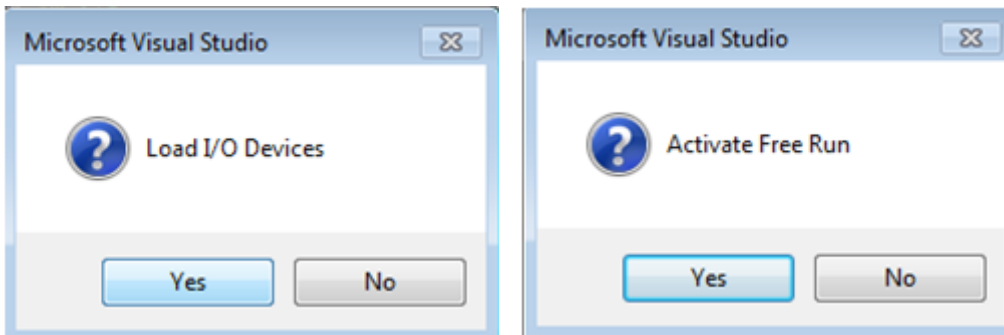


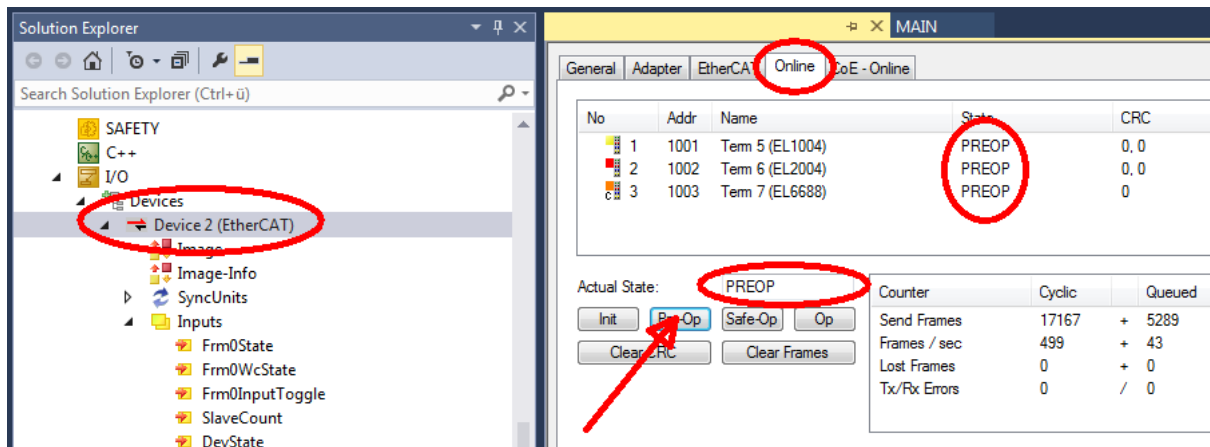
Fig. 188: Firmware Update

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

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

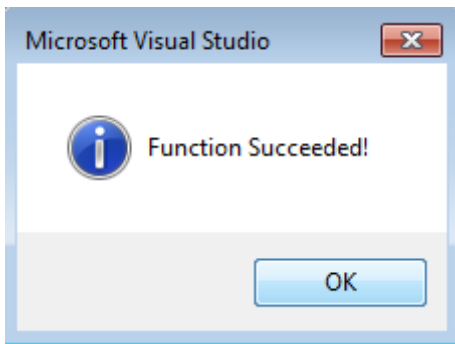


- Switch EtherCAT Master to PreOP



- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP

- Check the current status (B, C)
- Download the new *efw file (wait until it ends). A password will not be necessary usually.



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

7.2.4 FPGA firmware *.rbf

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

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

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

Determining the version via the TwinCAT System Manager

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

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

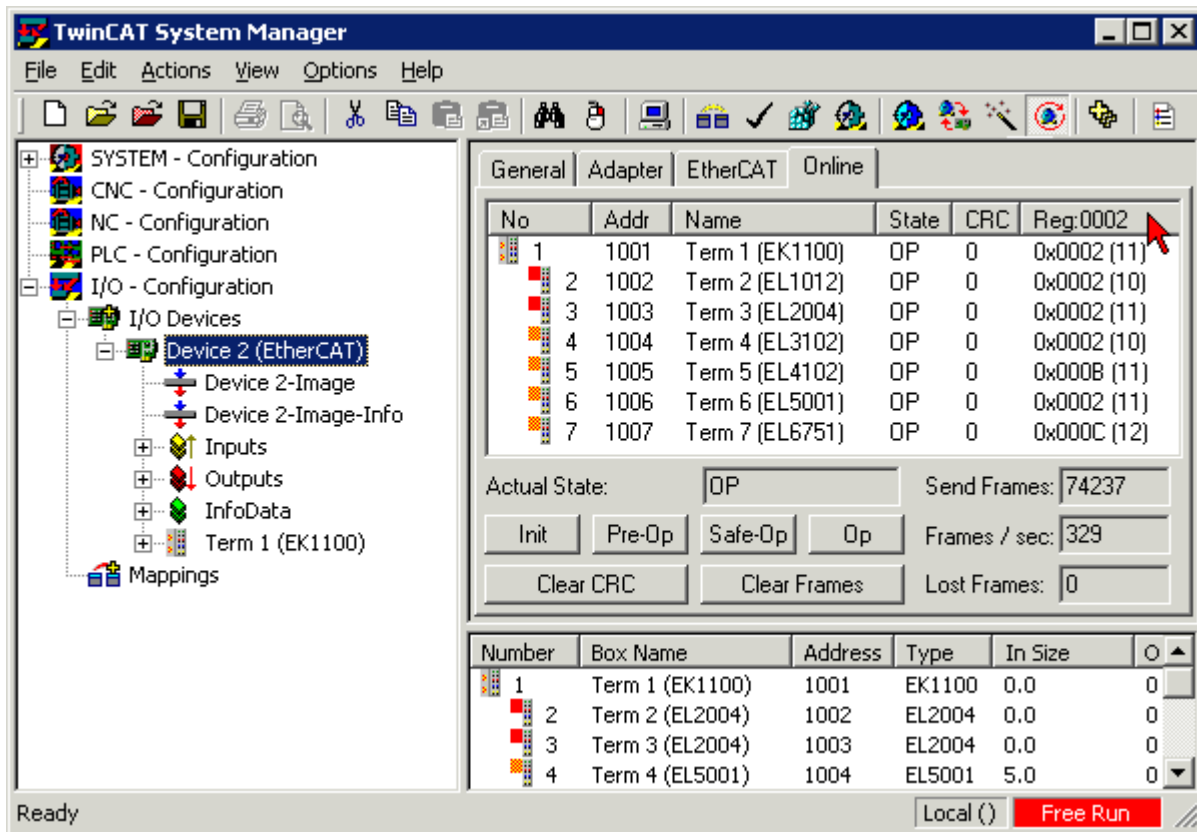
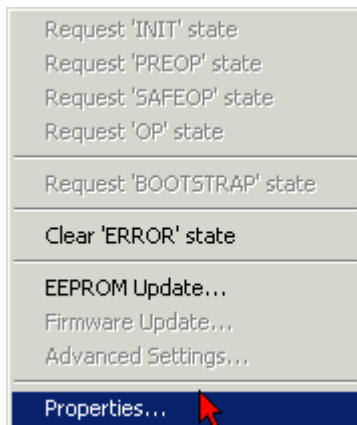


Fig. 189: FPGA firmware version definition

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

Fig. 190: 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.

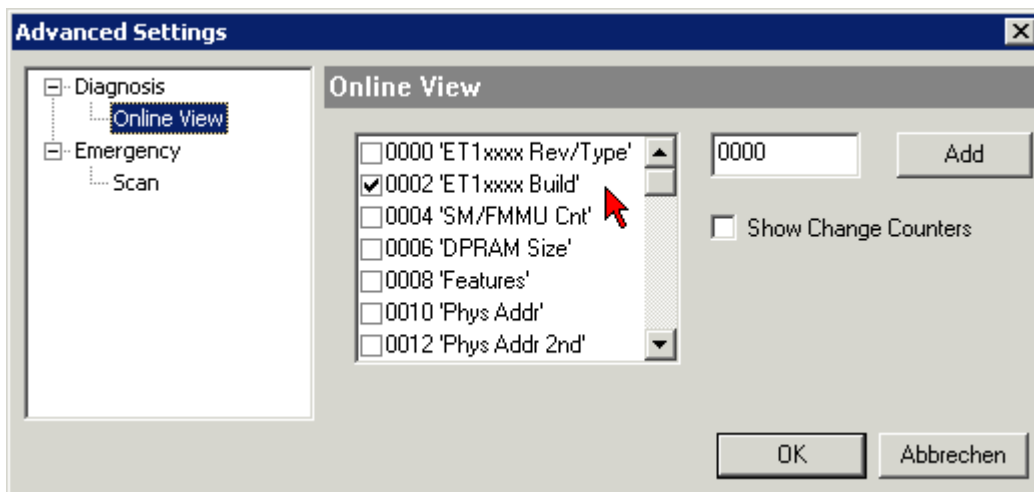


Fig. 191: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

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

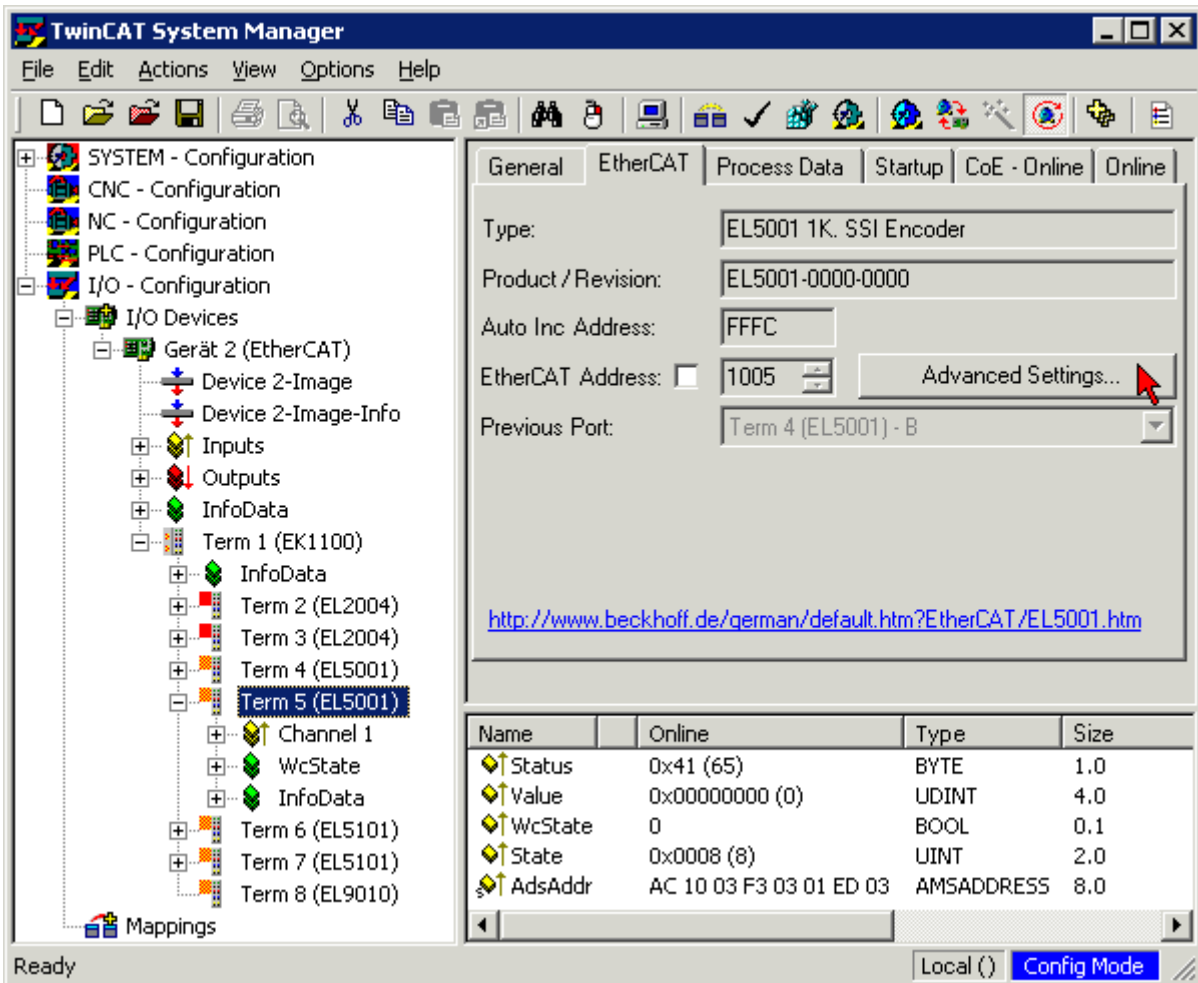
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

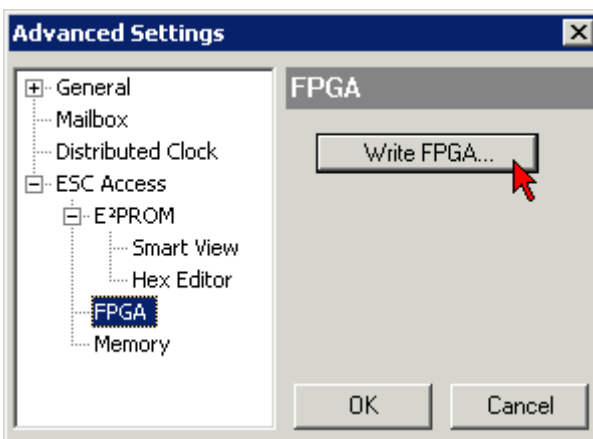
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

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

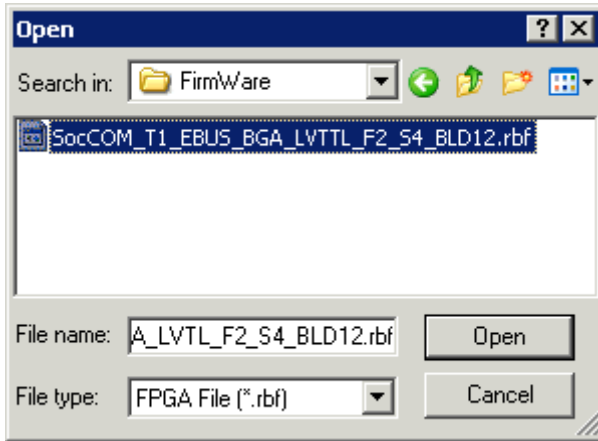
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



- The *Advanced Settings* dialog appears. Under *ESC Access/E²PROM/FPGA* click on *Write FPGA* button:



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



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

NOTICE

Risk of damage to the device!

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

7.2.5 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.

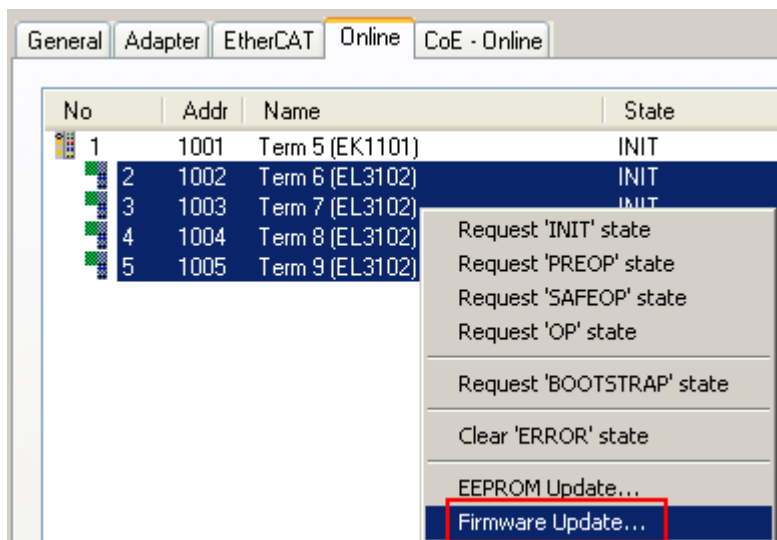


Fig. 192: Multiple selection and firmware update

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

7.3 Restoring the delivery state

To restore the delivery state (factory settings) of CoE objects for EtherCAT devices (“slaves”), the CoE object *Restore default parameters*, SubIndex 001 can be used via EtherCAT master (e.g. TwinCAT) (see Fig. *Selecting the Restore default parameters PDO*).

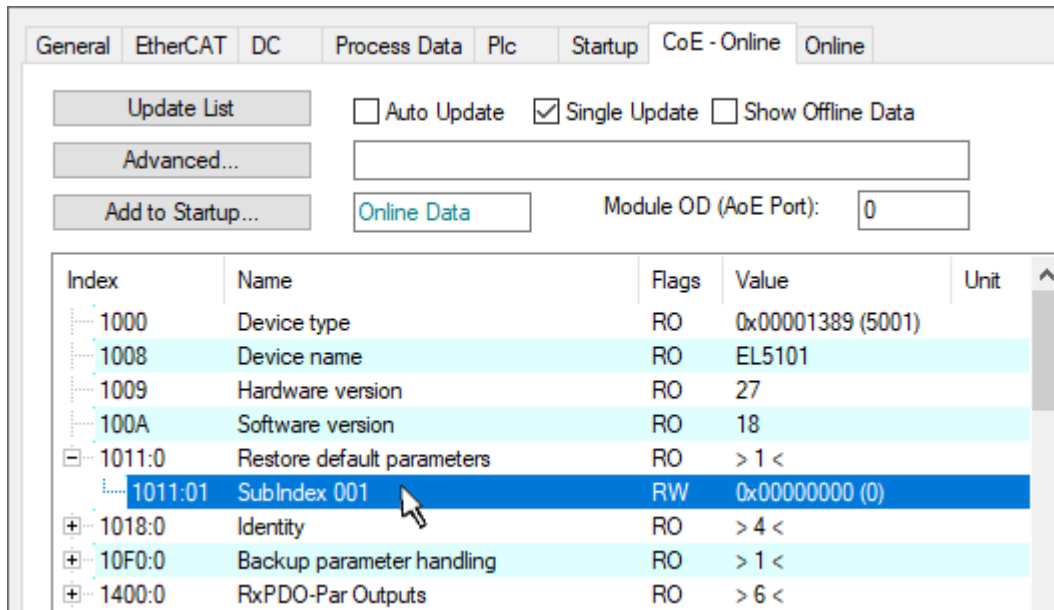


Fig. 193: Selecting the *Restore default parameters* PDO

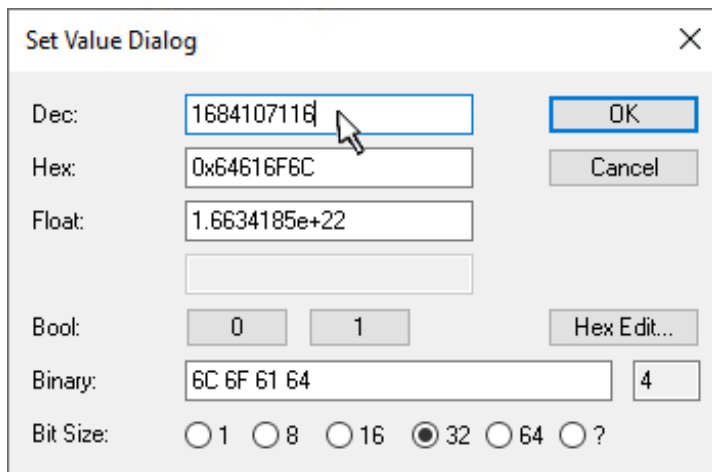


Fig. 194: Entering a restore value in the Set Value dialog

Double-click on *SubIndex 001* to enter the Set Value dialog. Enter the reset value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* (ASCII: “load”) and confirm with *OK* (Fig. *Entering a restore value in the Set Value dialog*).

- All changeable entries in the slave are reset to the default values.
- The values can only be successfully restored if the reset is directly applied to the online CoE, i.e. to the slave. No values can be changed in the offline CoE.
- TwinCAT must be in the RUN or CONFIG/Freerun state for this; that means EtherCAT data exchange takes place. Ensure error-free EtherCAT transmission.
- No separate confirmation takes place due to the reset. A changeable object can be manipulated beforehand for the purposes of checking.
- This reset procedure can also be adopted as the first entry in the startup list of the slave, e.g. in the state transition PREOP->SAFEOP or, as in Fig. *CoE reset as a startup entry*, in SAFEOP->OP.

All backup objects are reset to the delivery state.

i Alternative restore value

In some older terminals (FW creation approx. before 2007) the backup objects can be switched with an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164.

An incorrect entry for the restore value has no effect.

7.4 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Support

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

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

Hotline: +49 5246 963 157
e-mail: support@beckhoff.com
web: www.beckhoff.com/support

Service

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

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

Hotline: +49 5246 963 460
e-mail: service@beckhoff.com
web: www.beckhoff.com/service

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963 0
e-mail: info@beckhoff.com
web: www.beckhoff.com

More Information:

www.beckhoff.com/EL6601

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

