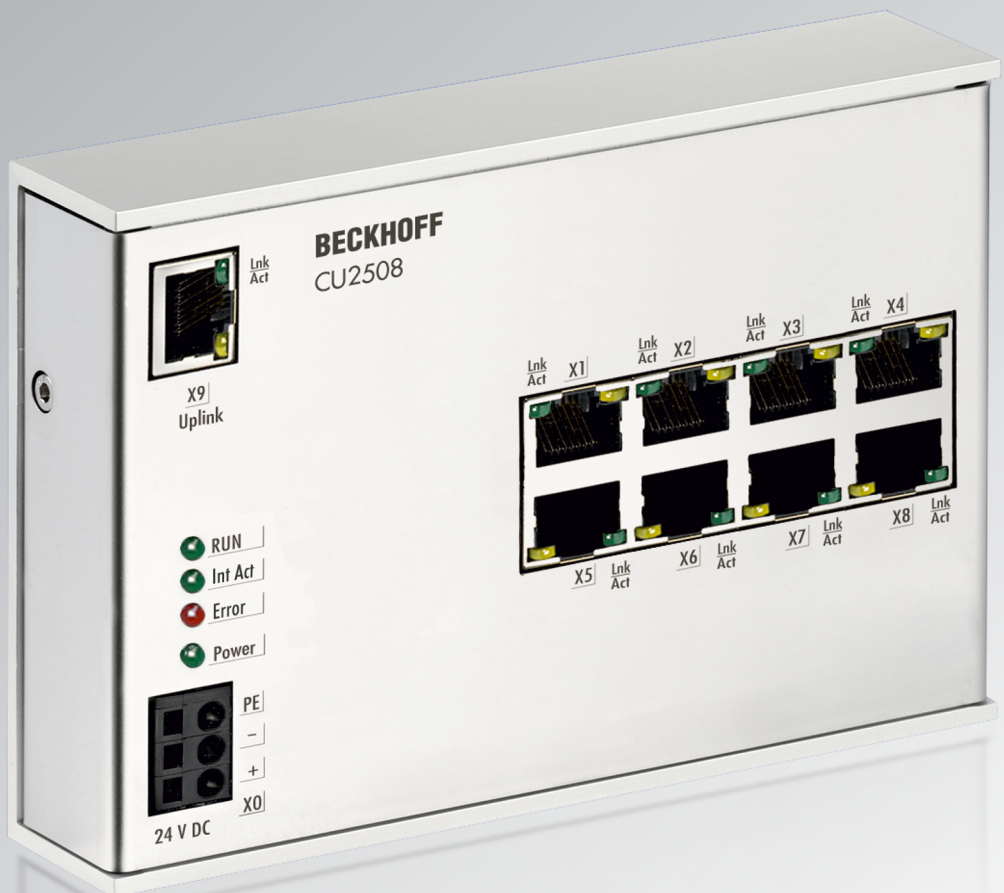
BECKHOFF New Automation Technology

Documentation | EN

CU2508

Real-time Ethernet port multiplier



2025-09-02 | Version: 3.4.0

Table of contents

1 Foreword 5
1.1 Notes on the documentation 5
1.2 Guide through documentation 6
1.3 Safety instructions 7
1.4 Documentation issue status 8
1.5 Version identification of EtherCAT devices 9
1.5.1 General notes on marking 9
1.5.2 Version identification of CU Infrastructure Components 10
1.5.3 Beckhoff Identification Code (BIC) 10
1.5.4 Electronic access to the BIC (eBIC) 12

2 Product overview 14
2.1 Introduction 14
2.2 Technical data 15
2.3 Technology 15
2.4 Start 22

3 Mounting and wiring 23
3.1 Dimensions 23
3.2 Installation positions 24
3.3 Mounting and demounting 26
3.4 Cabling 28
3.5 LEDs and connection 29
3.6 UL notice 30
3.7 Disposal 31

4 Commissioning 32
4.1 Setup in TwinCAT 32
4.2 Port definition TwinCAT 2.11 35
4.2.1 Setup: Device EtherCAT 35
4.2.2 Setting up: Device EtherCAT with cable redundancy 40
4.2.3 Setting up: Device Ethernet and TCP/IP notes 43
4.2.4 Setting up: Real-time Ethernet (Publisher/Subscriber, EAP push) 46
4.2.5 Setting up: Real-time Ethernet (BK90xx, ADS/RAW, EAP polling) 47
4.2.6 Setting up: PROFINET Controller/Device 49
4.2.7 Configuration: Ethernet/IP 50
4.3 Beckhoff EAP network variables 50
4.3.1 Introduction 50
4.3.2 Configuration of the Publisher 52
4.3.3 Configuration of the Subscriber 54
4.3.4 Beckhoff network variables - Settings 59

5 Appendix 67
5.1 ESL Protocol 67
5.2 Firmware Update 67
5.3 Firmware compatibility 76
5.4 Support and Service 77

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
The qualified personnel is obliged to always use the currently valid documentation.

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

Disclaimer

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

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

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

Trademarks

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

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



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

Third-party brands

Trademarks of third parties may be used in this documentation. You can find the trademark notices here:
https://www.beckhoff.com/trademarks

1.2 Guide through documentation

NOTICE



Further components of documentation

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

Title	Description
EtherCAT System Documentation (PDF)	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • EtherCAT devices configuration
Infrastructure for EtherCAT/Ethernet (PDF)	Technical recommendations and notes for design, implementation and testing
Software Declarations I/O (PDF)	Open source software declarations for Beckhoff I/O components

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

- the “Documentation and Download” area of the respective product page,
- the Download finder,
- the Beckhoff Information System.

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

1.3 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

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

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product



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

1.4 Documentation issue status

Version	Comment
3.4.0	- Chapter „Technology“ updated - Update structure
3.3.0	- Update chapter “Commissioning” - Update structure
3.2.0	- Update chapter “Setup in TwinCAT” - Update structure
3.1.0	- Update chapter “Technical data” - Update structure
3.0.0	- Update chapter “Commissioning” - Update chapter “Mounting and wiring” - Update chapter “Technical data” - Update structure
2.9.0	- Update chapter “Mounting and wiring” - Update structure
2.8	- Update chapter “Mounting and wiring” - Update revision status - Update structure
2.7	- New front page - Update chapter “Version identification of EtherCAT devices” - Chapter „Technical data“ updated - Chapter “Installation positions” added - Chapter “Disposal” added - Update revision status - Update structure
2.6	- Chapter „Firmware Update“ updated - Update revision status - Update structure
2.5	- Chapter „Technical data“ updated - Chapter „Technology“ updated - Update revision status - Update structure
2.4	- Chapter „Technology“ updated - Update revision status - Update structure
2.3	- Chapter „Technology“ updated - Update revision status - Update structure
2.2	- Chapter „Firmware Update“ updated - Update revision status - Update structure
2.1	- Chapter „Portdefinition TwinCAT 2.11“ updated - Chapter „Technical data“ updated - Update structure
2.0	- Migration
1.8	- Update chapter "LEDs and connection" - Update structure
1.7	- Addenda chapter "Setting up under TwinCAT 2.11"
1.6	- Addenda chapter "Setting up: PROFINET Controller/Device"
1.5	- Addenda chapter "Technology"
1.4	- Addenda note chapter Ethernet/IP slave
1.3	- Addenda RT Ethernet
1.2	- Addenda current firmware status
1.1	- Addenda Ethernet/IP slave
1.0	- Corrections, 1st public issue
0.1	- Preliminary documentation for CU2508

1.5 Version identification of EtherCAT devices

1.5.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

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

1.5.2 Version identification of CU Infrastructure Components

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

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



Fig. 1: CU1521 switch with serial number 4820/ and the unique Beckhoff Traceability Number (BTN) 00007su0

1.5.3 Beckhoff Identification Code (BIC)

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

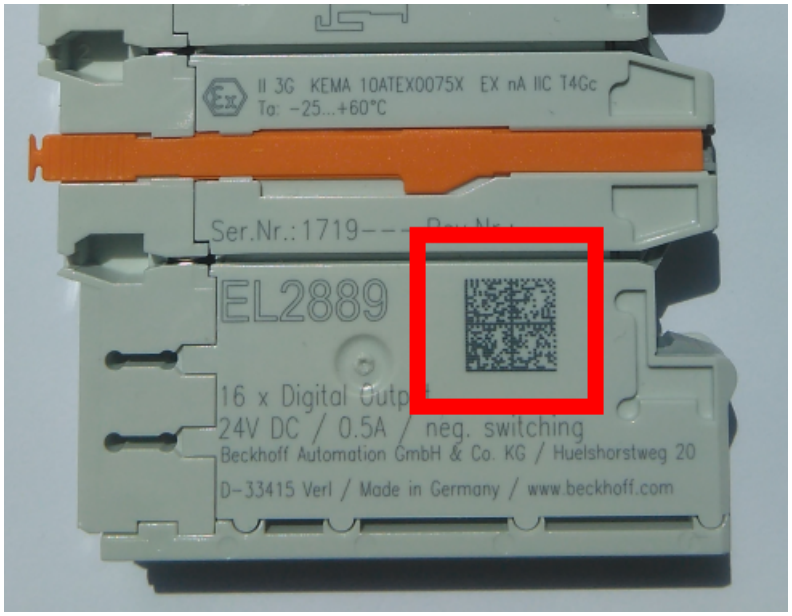


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

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

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

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

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

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

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

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

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

Structure of the BIC

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

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

Accordingly as DMC:



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

BTN

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

NOTICE

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

1.5.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

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

K-bus devices (IP20, IP67)

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

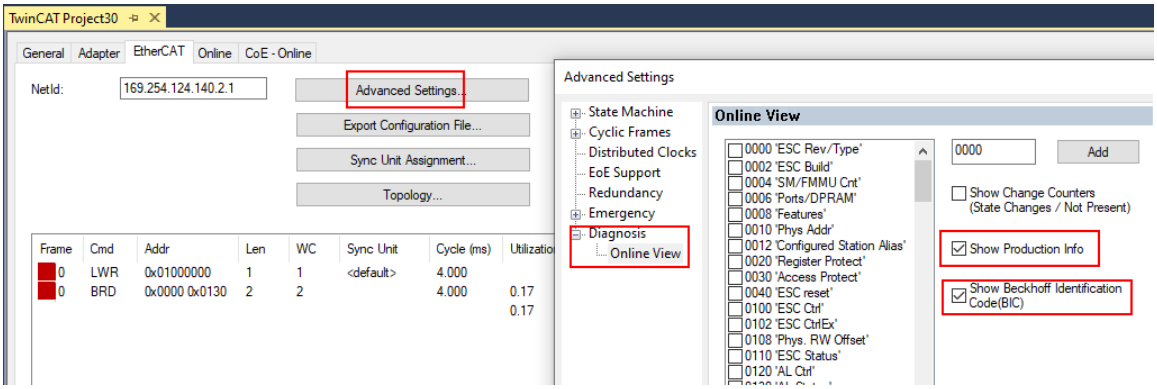
EtherCAT devices (IP20, IP67)

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

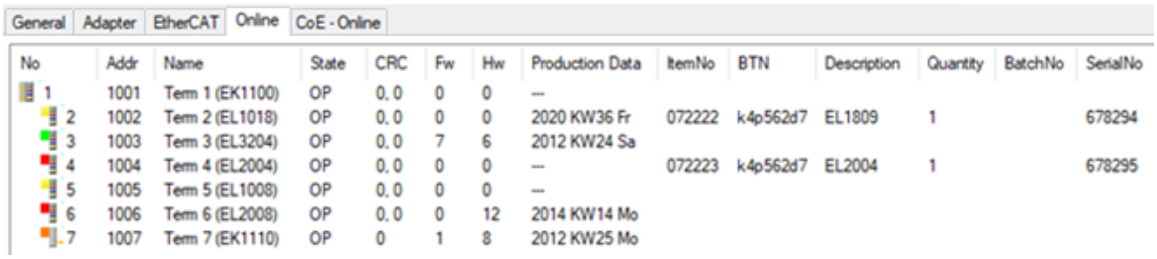
Beckhoff also stores the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

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

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

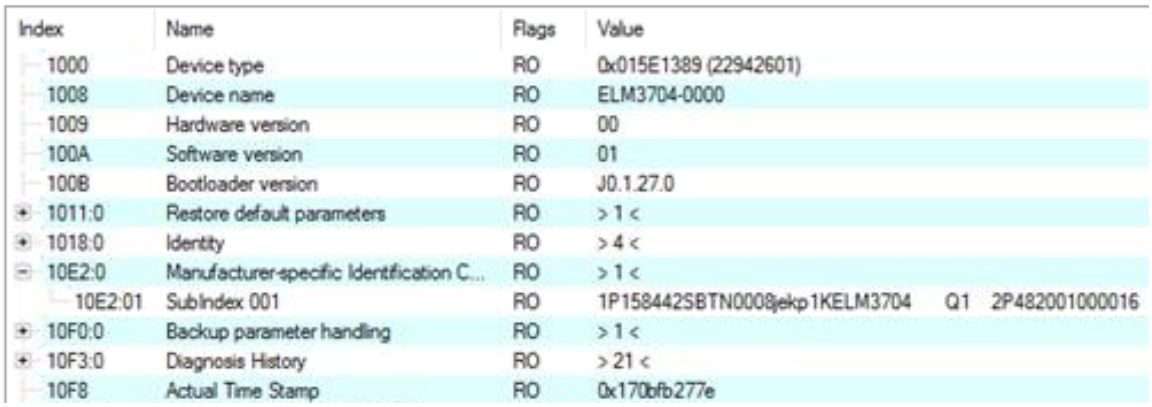


  - The BTN and its contents are then displayed:



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

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



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

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

PROFIBUS, PROFINET, and DeviceNet® devices

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

2 Product overview

2.1 Introduction



Fig. 4: CU2508

Real-time Ethernet port multiplier

The real-time Ethernet port multiplier allows the connection of eight independent FastEthernet networks (100 Mbit/s). The CU2508 is connected to the PC via a gigabit uplink.
The PC offers high-performance data transfer to the multiplier, which allocates the data to the relevant 100BASE-TX port based on an analysis of a frame prefix and sends them time-controlled with µs precision. Received frames are also allocated a prefix including a time stamp and sent to the PC.

With the aid of a driver, users have eight independent ports with full real-time characteristics available. The CU2508 is used in applications in which several Ethernet ports are required that need to be realized externally to the PC.

TwinCAT 2.11 supports the CU2508 and avoids the need for further network ports at the PC. For extremely high demands, an EtherCAT installation can, for example, be distributed or expanded to up to eight strands in order to multiply the performance. The distributed clocks of the EtherCAT strands are synchronized. An EtherCAT cable redundancy in conjunction with distributed clocks can also be realized with two ports of the CU2508.

Quick links

- Technology basics [▶ 15]
- Setup CU2508 [▶ 32]
- Setup EtherCAT [▶ 35]
- Setup EtherCAT cable redundancy [▶ 40]
- ESL protocol [▶ 67]

2.2 Technical data

Technical data	CU2508
Operating voltage	24 (18...30) V_{DC}, 250 mA, 3-pin connector (+, -, PE)
Number of Ethernet ports	8 x 100 Mbit/s (100BASE-TX) and 1 x 1 Gbit/s (1000BASE-T, Uplink)
Ethernet interface	RJ 45
Cable length	up to 100 m, min. CAT5 twisted-pair, shielded
Diagnosis	via LED, software
Protocols	depends on used CU2508 driver TwinCAT: Ethernet TCP/IP; real-time protocols: EtherCAT, PROFINET, RT Ethernet etc.
TwinCAT support	TwinCAT from version 2.11R2 (build 2028) TwinCAT3
Electrical isolation	1,500 V
Weight	approx. 500 g
permissible ambient temperature range during operation	-25°C ... + 55°C
permissible ambient temperature range during storage	-40°C ... + 85°C
permissible relative humidity	95 %, no condensation
Dimensions [▶ 23] (W x H x D)	approx. 146 mm x 100 mm x 30 mm
Mounting [▶ 26]	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
Approval	CE, EAC, UKCA cULus [▶ 30]

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

2.3 Technology

Introduction

As a port multiplier that is as transparent as possible, the CU2508 extends *one* Gbit Ethernet port on the controller to *8* FastEthernet ports in the field. It transports IEEE802.3 conformant Ethernet frames with arbitrary contents.

Each port of the CU2508 sends and receives FastEthernet frames (100 Mbit, 100BASE-TX) over up to 100 m of copper cable/RJ45. The CU2508 does not generate or process the contents of any frames itself; instead, it exclusively forwards frames sent to it by a special software driver to the field selectively via its 8 ports or forwards frames received from the field to the driver. The highly precise time information regarding when the frames are sent or received is thereby optional.

To this end, the CU2508 has at its disposal

- the Uplink Gbit Master Port (X9) to the driver in the controller, which needs a Gbit connection at the opposite end
- 8 equivalent 10/100 Mbit downlink ports (X1-X8) for the real-time traffic to the connected field devices

A CU2508 system thus consists of the CU2508 device and the CU2508 driver, e.g. integrated in TwinCAT 2.11R2 or TwinCAT 3.

The CU2508 system does not replace master implementations of Ethernet based field buses; instead, it tunnels specified data telegrams via the Gbit connection and then sends the frames at the specified time. It behaves transparently for the protocols fed across it, with exception of the EtherCAT protocol – in this case a CU2508 device is visible as the first slave in the configuration. Each materially existent I/O system on the field side must therefore match a logical master component in the controller.

Several CU2508s can be used in each TwinCAT system.

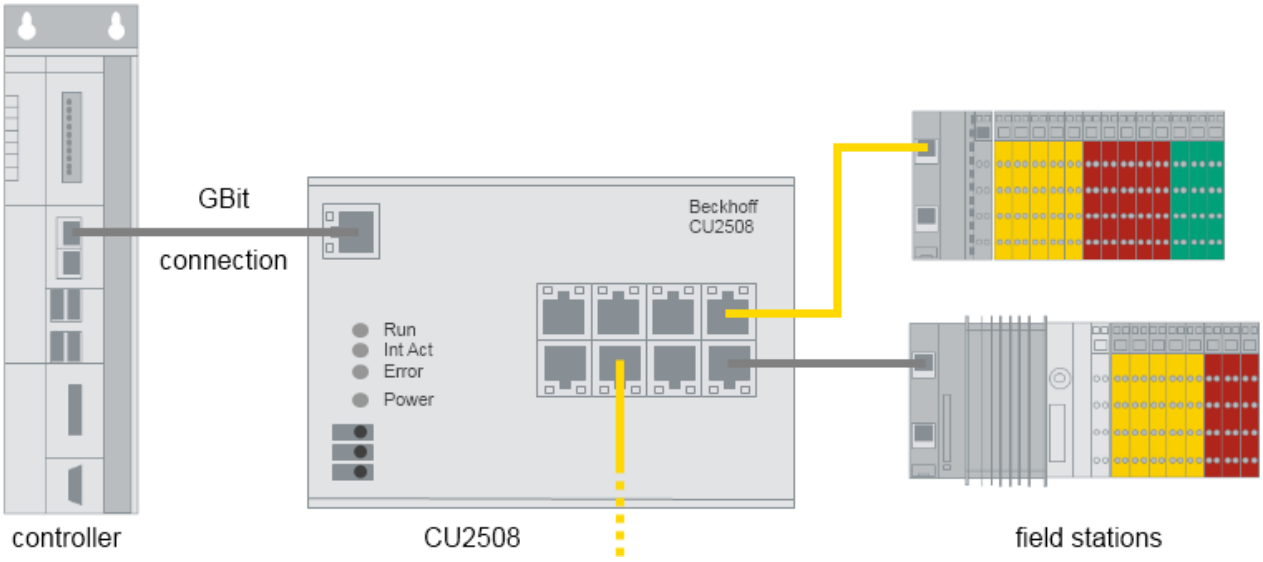


Fig. 5: Principle CU2508 topology

Downlink Port characteristics

The default setting of the CU2508 is optimized for use with EtherCAT downlinks, especially for EtherCAT IO redundancy operation. Therefore the 8 downlinks up to and including FW11 in the factory setting had the function to mirror incoming 100 Mbit frames back to the transmitter if the 1 GBit uplink is missing, called "Auto Link Close".

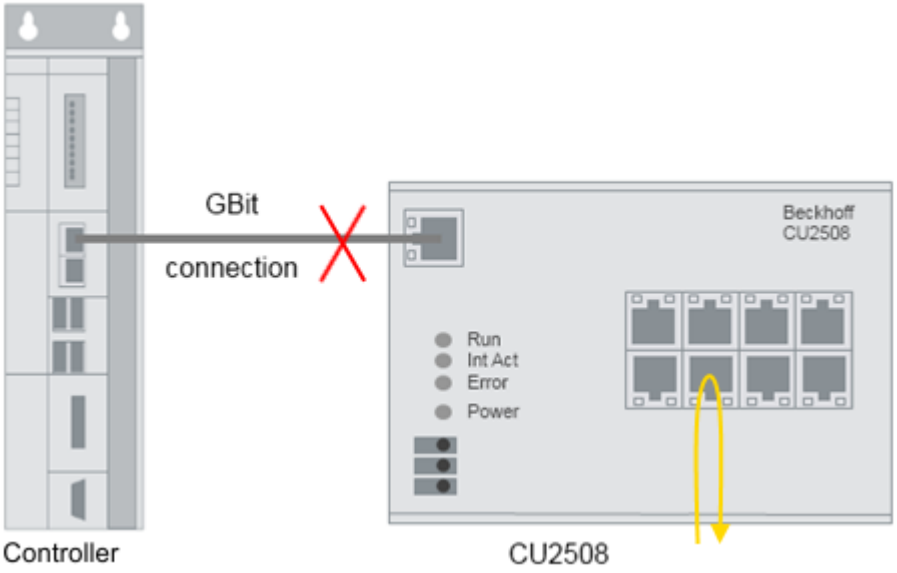


Fig. 6: Display "Auto Link Close"

In non-EtherCAT operation this feature can be a problem and might disrupt the underlying network. Therefore it is deactivated globally for all downlink ports starting from FW12, so that in case of an uplink loss no return of incoming 100 Mbit frames takes place but the frames in the CU2508 "percolate". The electrical link is not changed.

ESL protocol

The software driver in the controller/control device/IPC forms the counterpart to the CU2508. It works on a Gbit port in the controller and "packs" the user data into the EtherCAT Switch Link Protocol (ESL) or unpacks the ESL protocol from the CU2508 and forwards the user data to the application. Therefore, no extra telegram containing control data is sent to/from the CU2508 for the handling of the user data; instead, the user data generated by user programs are supplemented by several bytes of control and information data for the connection between controller and CU2508.

The CU2508 driver is integrated in TwinCAT from version 2.11R2 onwards; pay attention to the specifications in the Technical data. The ESL protocol is disclosed, see the Description page. In addition, it has been included in the Wireshark-Installation since version 1.4.2.

EtherCAT systems and CU2508

The CU2508 can be used to operate several FastEthernet EtherCAT systems on one port of the IPC, i.e. quasi as port-multiplier. Hence the term "port multiplier".

When operating several EtherCAT systems on the ports of a CU2508, temporal effects may occur that may be relevant for the application. Some explanations are provided below.

The CU2508 essentially supports the following three operation modes. To understand this, basic knowledge of the EtherCAT operation modes and synchronization methods is helpful.

1. Standard mode: no frame influence, no DistributedClocks
   - The CU2508 forwards incoming frames via Gbit-ESL to the desired FastEthernet port, also in the opposite direction. There is no time control for the Ethernet frames.
   - The EtherCAT slaves of the lower-level systems therefore operate on a frame-triggered basis (also referred to as FreeRun mode), and the output times are essentially dependent on frame delays/ jitter, for example.
2. Time-controlled sending/time-stamped receiving: with frame influence, no DistributedClocks
   - The CU2508 forwards incoming frames via Gbit-ESL to the desired FastEthernet port at the requested time. Time stamping takes place in the opposite direction. In other words, the Ethernet frames are time-controlled.
   - Frame-triggered EtherCAT slaves thus operate with "low jitter", and "synchronized" between the EtherCAT systems.
   - In order for the frames to be forwarded on a time-controlled basis, buffering in the CU2508 is required, which may cause considerable delays. For short cycle times the feasibility should be verified!
   - This operation mode is not yet supported (as of 2019).
3. DistributedClocks mode, no frame influence
   - The forwarded EtherCAT frames are subject to temporal influence by the transmitting IPC, the CU2508 and the EtherCAT slaves.
   - The ports X1..8 are parameterized as DistributedClocks - ReferenceClocks
   - Thus, the EtherCAT slaves of the lower-level systems that support DistributedClocks also operate DC-synchronously. This means that the input/output operations in these slaves can be synchronized, even at the "same" time between the EtherCAT systems on ports X1..8.
     In this case the overall system is essentially independent of frame delays/jitter, as long as these are not significant enough to impair the DistributedClocks control.
   - With regard to EtherCAT operation, this method is essentially the most sensible, because
     • it offers the best temporal definition for the input/output operations of the EtherCAT devices
     • no time buffers are necessary in the CU2508

The following aspects must be taken into account in order to be able to estimate temporal effects in operation modes 1 and 3:

- Depending on the data content, Ethernet frames have a time length of
  - X1..8 FastEthernet: 7..128 µs, InterFrameGap (IFG) 0.96 µs
  - X9 Gbit: 0.7..12 µs, IFG 0.096 µs
- The CU2508 has an internal delaying data buffer for each port, due to the different transport speed.
- TwinCAT sends the Gbit frames for X9 serially (one after another) via the GBIT/ESL connection. The GBit frame lengths can be of a magnitude that is significant in the context of short TwinCAT cycle times!
- If several tasks are to be processed in TwinCAT, by default TwinCAT processes them serially (one after the other). As a result, the ESL frames are sent with a corresponding delay.
  The "isolated core" setting in the RealTime tasks provides a remedy so that the tasks can be processed in parallel.

- The EtherCAT frame length must also be taken into account.
  Example: An EtherCAT system is installed at ports X1 and X2, each with an EL2202 as output terminal. The edges are to be measured with an oscilloscope for demonstration purposes. In system X1 the bit of the respective output terminal used is in a short 7 µs frame, whereas in system X2 it is in a long 128 µs frame. This alone causes the signal to be output 121 µs later in system X2.
  A remedy is provided by DistributedClocks, see above.
  (The position of the output data in the EtherCAT frame is usually irrelevant since output data is only output after the checksum procedure, once the frame has fully passed the output device.)
- Typical delays caused by the management of the CU2508 are as follows
  - In the downlink
    Gbit X9 to FastEthernet X1..4: tFE = 1 µs
    Gbit X9 to FastEthernet X5..8: tFE = 1.6 µs

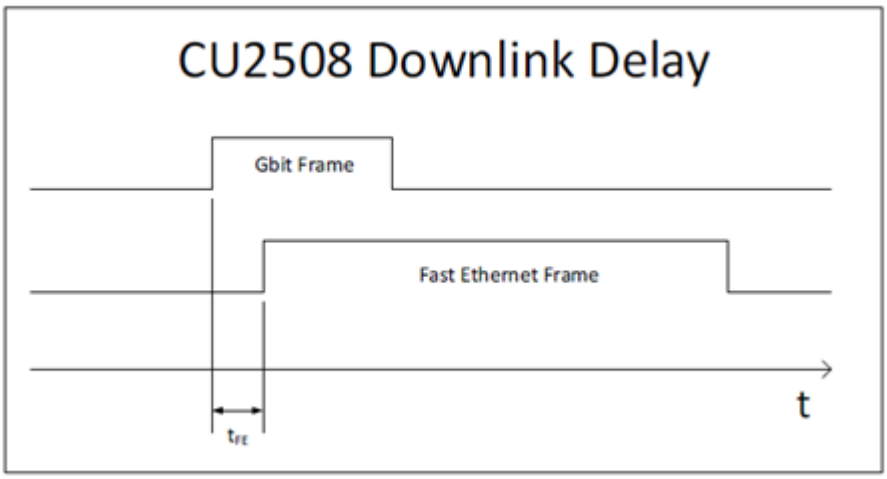


Fig. 7: CU2508 Downlink Delay

  - In the uplink
    FastEthernet X1..X4 to Gbit X9: tGE = 0.7 µs
    FastEthernet X5..X8 to Gbit X9: tGE = 1.1 µs

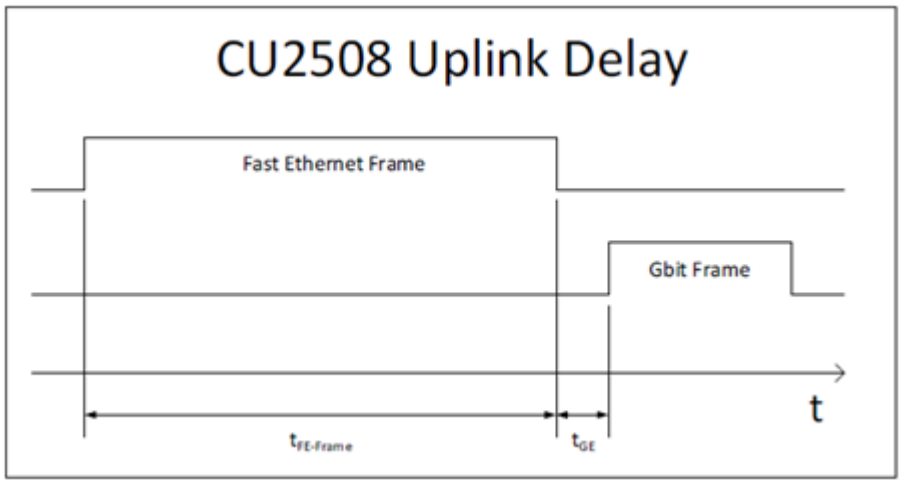


Fig. 8: CU2508 Uplink Delay

  - These delays are therefore relatively insignificant compared to the other factors mentioned above. What immediately catches the eye, however, is the importance of the frame length and the required buffering in the uplink.

CU2508 as EtherCAT slave

Each downlink port of the CU2508 can be configured as a separate "EtherCAT device", see chapter EtherCAT device setup [▶ 35]. In this case the CU2508 port represents the first EtherCAT device in the system. It is capable of Distributed Clocks and can therefore operate as a reference clock in the strand.

The combination of EtherCAT cable redundancy and Distributed Clock function is possible by combining two such EtherCAT ports using the "TwinCAT cable redundancy" supplement (subject to charge).

Time-controlled send/receive (in preparation)

The frame forwarding in the CU2508 can be subjected to precise time control by the local clock:

- the driver or the user application specifies the downlink port via which the frame is to be sent by the CU2508 and when.
  These data are added by the driver to each frame as additional information.
- each frame received by the CU2508 at a Downlink port is supplemented by receipt information (receiving port, time) and forwarded to the controller via the uplink.

The local hardware-based clock in the CU2508 then controls the sending of the frames with a high temporal quality. In this way, the CU2508 permits the construction of a real-time Ethernet network (TwinCAT Publisher/Subscriber, Profinet etc.), even if the control device cannot guarantee hard real-time when sending the protocol data. However, the control device must be able to deliver or accept the data with sufficient speed.

The time controller uses the 64-bit time format familiar from the EtherCAT Distributed Clocks system: resolution 1 ns, starting from 1.1.2000 00:00 and thus adequate for ~584 years

The time stamp information (sending and receiving) is for the time being only evaluated by the CU2508 driver and is not available to the user application.

As the start of an Ethernet frame, the SFD (Start of Frame Delimiter) is evaluated according to the IEEE802.3 standard.

EoE and TCP/IP

The CU2508 is connected to the IPC via the Gbit interface. This Ethernet interface appears in the operating system/Windows of the IPC with its properties (IP address, IP mask etc.). From the point of view of the operating system, therefore, there is “only” this network connection via which telegrams can be sent or received. The CU2508 driver can now either convey data traffic from the operating system level to a dedicated CU2508 port or feed it into the virtual EtherCAT EoE “switch”. Also compare the documentation for the EL6601/EL6614 here. The selection is made in the settings in the System Manager. Either the specific CU2508 port or generally *EoE* can be selected via "TCP/IP Port".

Refer in particular to the TCP/IP notes [▶ 43] here.

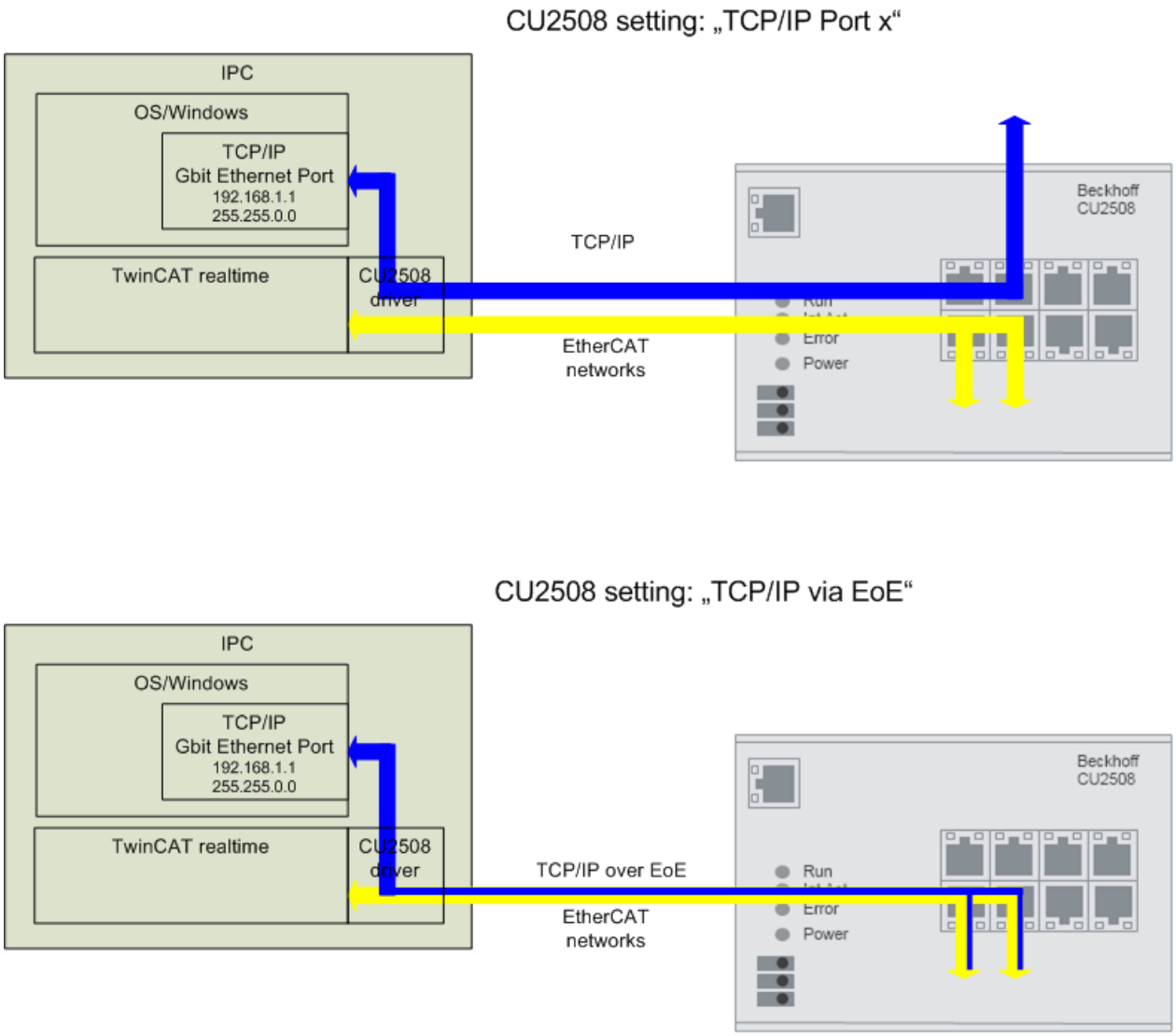


Fig. 9: Different IP data traffic depending on the setting

Applications

The above-described functions permit the use of the CU2508 for the following applications, among others:

- Multi EtherCAT adapter [▶ 35]
  Up to 8 independent EtherCAT systems can be created.
- Synchronized EtherCAT systems [▶ 35]
  If the CU2508 is selected as the ReferenceClock, the EtherCAT systems connected to the CU2508 are operated with the same synchronized time base.
- EtherCAT cable redundancy [▶ 40]
  2 downlink ports of each CU2508 can be combined into a cable-redundant EtherCAT system. Hence, fewer Ethernet ports are occupied on the controller; only *one* Gbit Ethernet port is required for the uplink. Up to 4 cable-redundant EtherCAT systems are possible. The CU2508 is placed next to the controller in the control cabinet; I/O cable connections that are in danger of being disconnected are then fed redundantly out of the control cabinet in the ring closure.

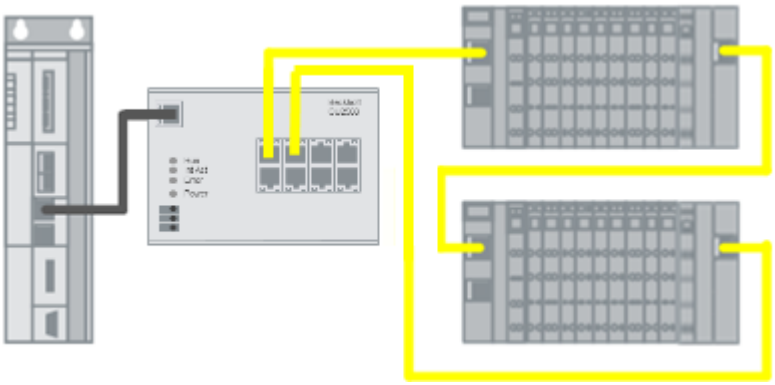


Fig. 10: Use of the CU2508 as an EtherCAT cable redundancy star point

- EtherCAT cable redundancy with Distributed Clocks [▶ 40]
  The common time base of the CU2508 ensures that EtherCAT slaves that require Distributed Clocks are still subject to synchronization in the case of redundancy.
  The CU2508 is currently the only way to operate Distributed Clocks compatible slaves in a cable-redundant ring.
- Use of TCP/IP without real-time [▶ 43]
  A downlink port on the CU2508 can be configured as a non-real-time Ethernet port, or the CU2508 works in the Ethernet over EtherCAT (EoE) network and forwards TCP/IP frames from the connected EtherCAT systems.

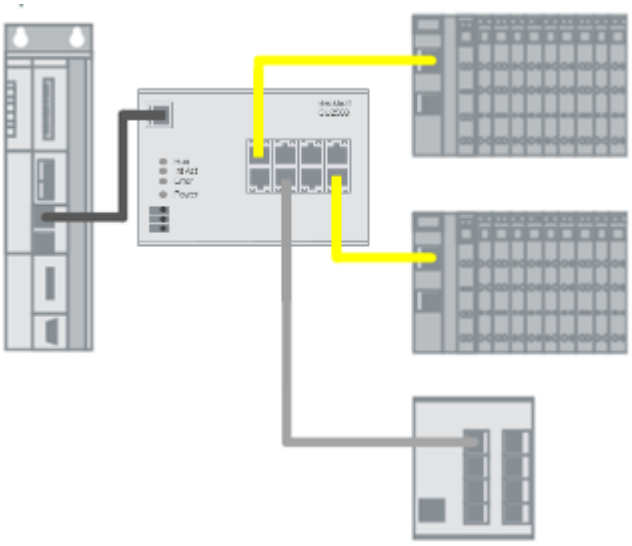


Fig. 11: Use of the CU2508 as an Ethernet multi-protocol handler with EtherCAT and TCP/IP

- Real time fieldbus on non-real-time controller
  If an Ethernet-based fieldbus requires a reliable constancy with respect to the sending of communication telegrams, low jitter is demanded in the cyclic operations of the controller. If a high-performance controller is able to process the cyclic operations with sufficient frequency (= demanded short cycle time), but the jitter, i.e. the regular interval between the cycles is inadmissibly high, then the CU2508 system, as a real-time frame handler, can provide the constant interval in frame sending, provided the new data are available early enough in the CU2508.

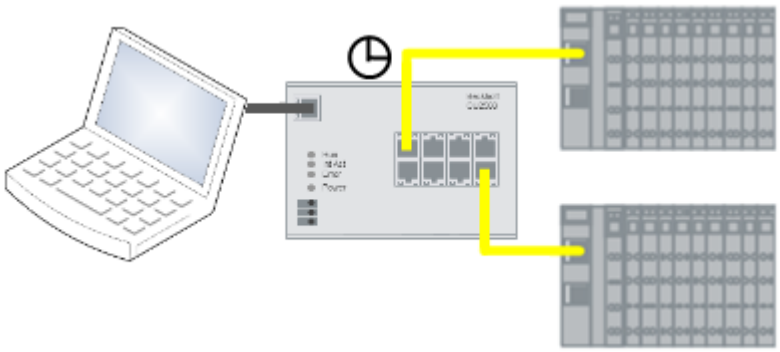


Fig. 12: Use of the CU2508 as an EtherCAT real-time environment

There are currently still restrictions (TwinCAT 2.11/3.1, FW10) with regard to

- external synchronization of the CU2508
  The influence on the internal CU2508 clock is still in preparation.
- Profinet [▶ 49]
- BACNet
- time stamp-controlled sending/receiving of frames

The implementation of these and further protocols is in preparation.

Data traffic in the lower-level EtherCAT segments

From FW07 and ESI revision -0018, Ports 1 and 5 have a larger data buffer of 16 kbyte (instead of 8 kbyte) for EtherCAT segments with particularly high data transfer rates.

"High data traffic" is generated by IO systems with many cyclical data, e.g. when many subscribers (over 100) and/or subscribers with large data requirements (e.g. analog oversampling terminals) are used.

If a "large" IO system is operated in EtherCAT redundancy mode, it is advisable to use ports 1 and 5.

The memory situation found is occasionally reported by TwinCAT with "Cu2508 fifo sizes...":

Server (Port)	Timestamp	Message
TCOM Server (10)	8.03.2019 15:47:40 96 ms	Cu2508 fifo sizes: 1:16 2:8 3:8 4:8 5:16 6:8 7:8 8:8

Fig. 13: Message relating to CU2508 fifo sizes

2.4 Start

For commissioning

- install the CU2508 as described in the Mounting and wiring [▶ 23] chapter
- configure the CU2508 in TwinCAT as described in the Commissioning [▶ 32] chapter

Furthermore, the field devices typical for the application are required on the downlink ports.

3 Mounting and wiring

3.1 Dimensions

Dimensions without plugs (W x H x D)

- CU2508: approx. 146.5 mm x 100 mm x 38 m

Space requirement in the control cabinet

- The RJ45 connector increase the depth depending on their design and the Ethernet cable used.
- Above the mounting rail an additional height of approx. 10 mm is required to enable latching [▶ 26] of the switch onto the rail.
- Take into account the recommended distances of 35 mm above and below the device (see chapter Installation positions [▶ 24]).

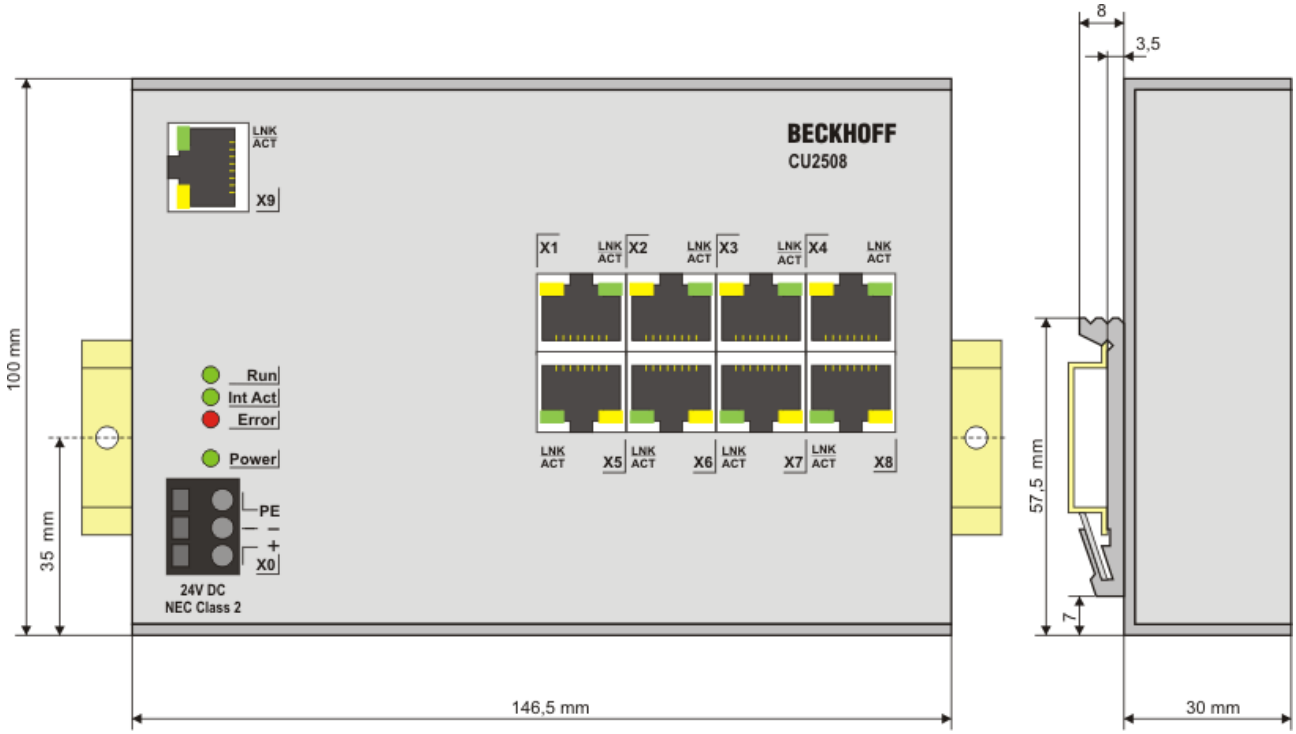


Fig. 14: CU2508 dimensions

3.2 Installation positions

NOTICE

Constraints regarding installation position and operating temperature range

Please refer to the technical data for a component to ascertain whether any restrictions regarding the installation position and/or the operating temperature range have been specified. When installing high power dissipation components 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 components to face forward (see Fig. *Recommended distances for standard installation position*). “From below” is relative to the acceleration of gravity.

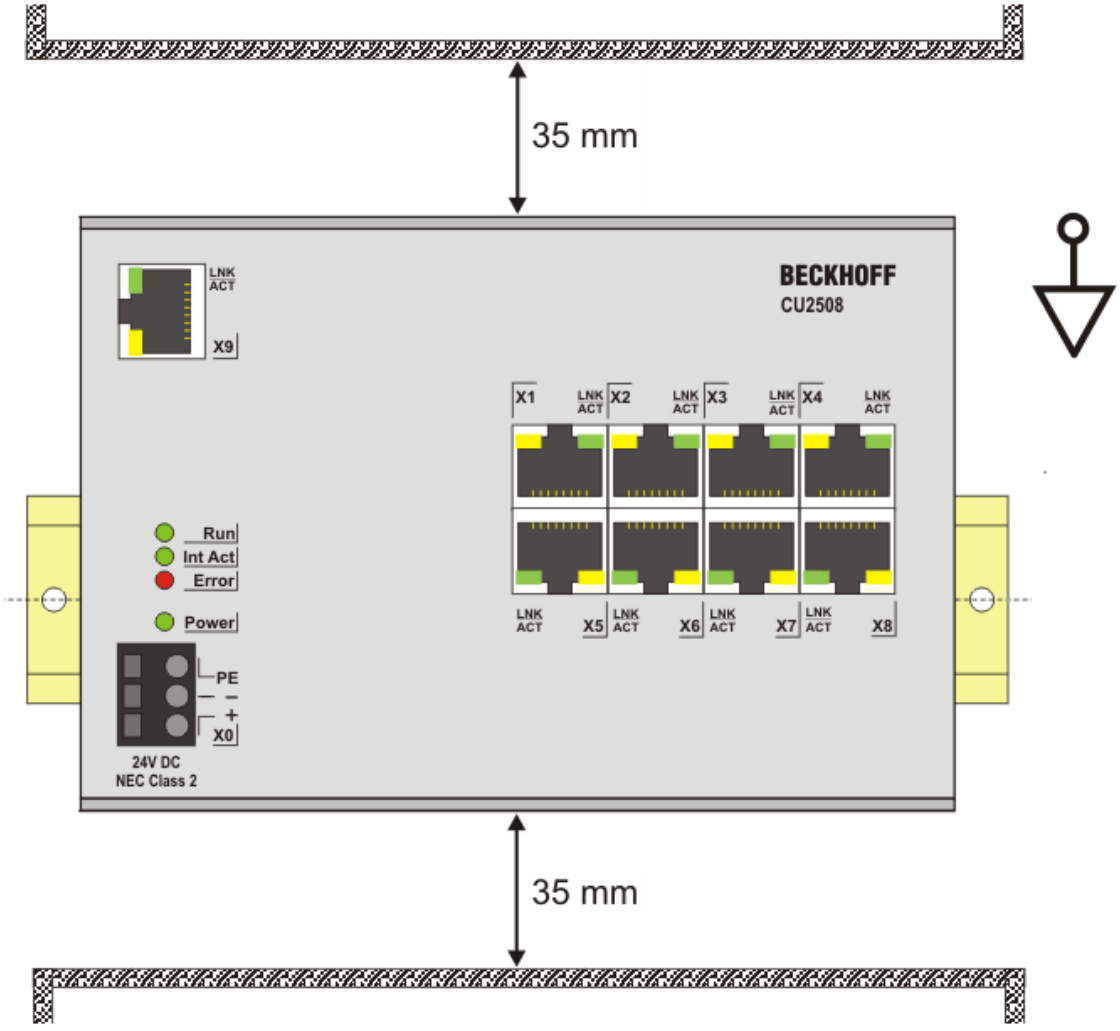


Fig. 15: Recommended distances for standard installation position

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.

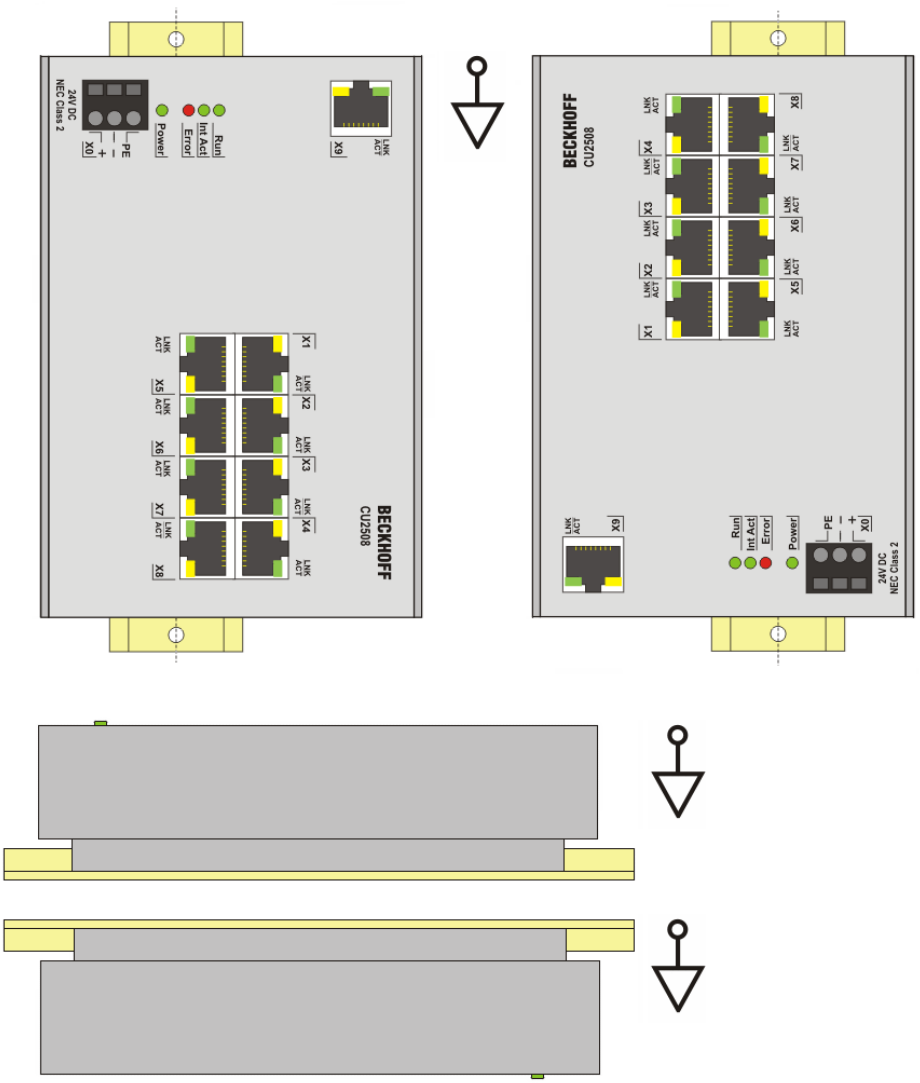


Fig. 16: Other installation positions

3.3 Mounting and demounting

The CU2xx switches are mounted on the mounting surface with the aid of a 35 mm DIN rail (mounting rail according to EN 60715).

Mounting

- Fit the mounting rail to the planned assembly location.
- Suspend the switch on the mounting rail with the spring on the lower side of its latching flange.
- Press the switch upwards (1).
- Press the upper side of the switch (2) against the assembly surface until it latches in the mounting rail.
- Attach the cable.

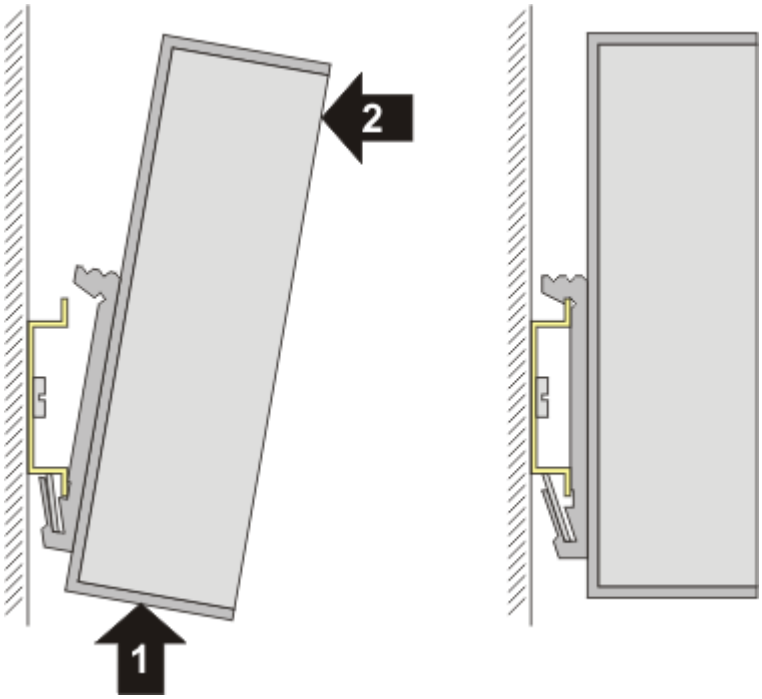


Fig. 17: Mounting

Removal

- Remove all the cables.
- Press the switch upwards (3).
- Pull the other side of the switch (4) away from the assembly surface.

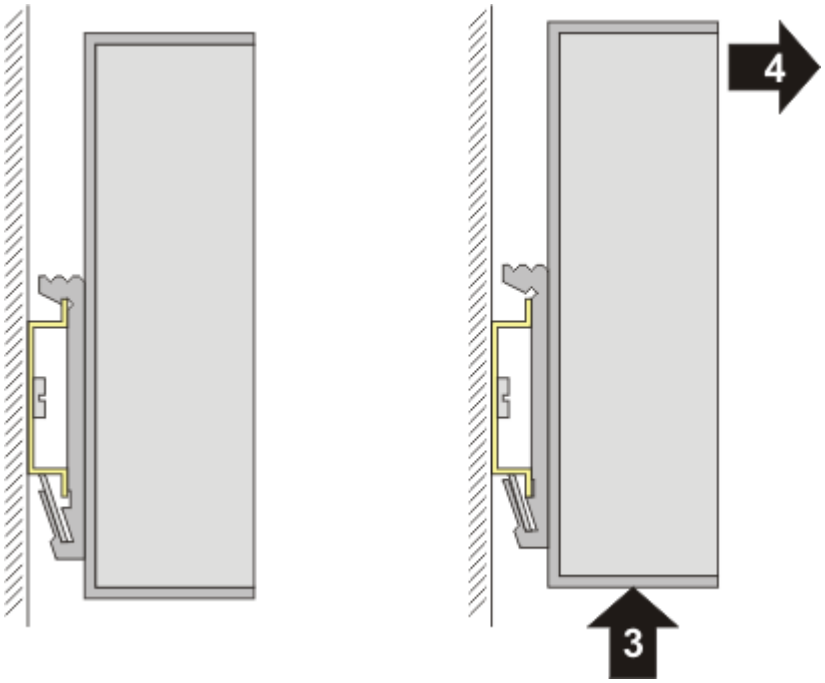


Fig. 18: Removal

3.4 Cabling

Power supply

Contact assignment of the spring loaded supply connector

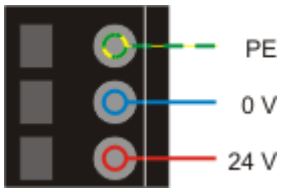


Fig. 19: CU20xx – Supply connector

UL requirements

If terminals certified with restrictions are used, then the current consumption at 24 VDC must be limited accordingly by means of supply
• from an isolated source protected by a fuse of max. 4 A (according to UL248) or
• from a voltage supply complying with NEC class 2.

An NEC class 2 voltage source must not be connected in series or parallel with another NEC class 2 voltage source! These requirements apply to the supply of all Bus Couplers, power supply terminals, Bus Terminals and their power contacts.

UL requirements

In order to comply with UL requirements, the EtherCAT components must not be connected to unlimited voltage sources!

Permissible cable cross-sections of the supply connector

The 3-pin supply connector is connected using push-in wiring.

The following cable cross-sections are permitted:

Device	CU15xx
Solid conductor	0.14 ... 1.5 mm² (AWG26-14)
Fine stranded conductor	0.14 ... 1.5 mm² (AWG26-14)
Fine stranded conductor; with ferrule with plastic collar	0.14 ... 1.5 mm² (AWG26-14)
Stripping length	8 ... 9 mm

Ethernet

Pin configuration of the RJ45 sockets

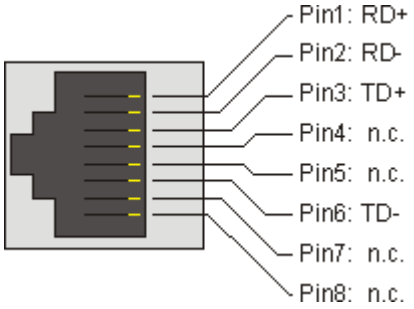


Fig. 20: Pin configuration of the RJ45 sockets

3.5 LEDs and connection

The CU2508 has four diagnosis LEDs.

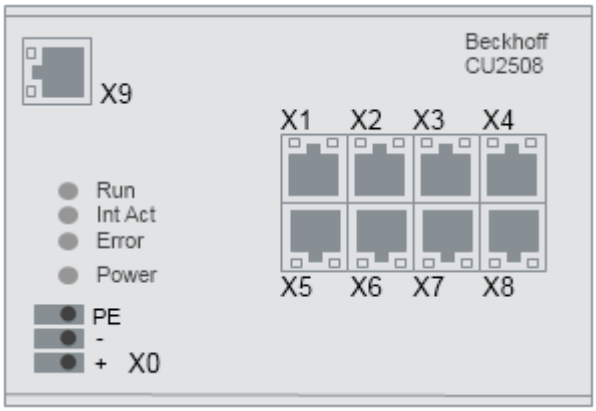


Fig. 21: CU2508 front view

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the operating state of the CU2508 running as an EtherCAT device	
		off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal or **BOOTSTRAP**
		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 sync manager 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
Int Act	green	flashes when data traffic occurs	
Power	green	on, when the device is powered correctly	
Error	yellow	an error has occurred	
X1 - X8	green	Link/Activity	
		On	Ethernet link established
		flashing	receiving frames
	yellow	Port diagnosis	
		Off	no error
		flashing	internal buffer overflow, alignment error detected or 10 Mbit connection (not supported)

Pin assignment

Terminal point	Comment
X0	24 V_{DC} power supply: +, GND and shield connection (functional earth)
X1..X8	DownLink Port FastEthernet 10/100 Mbit
X9	UpLink Port Gbit Ethernet

Shielding concept

The PE/shield connection is connected to the metal housing. It should be used as functional earth to protect from interference.

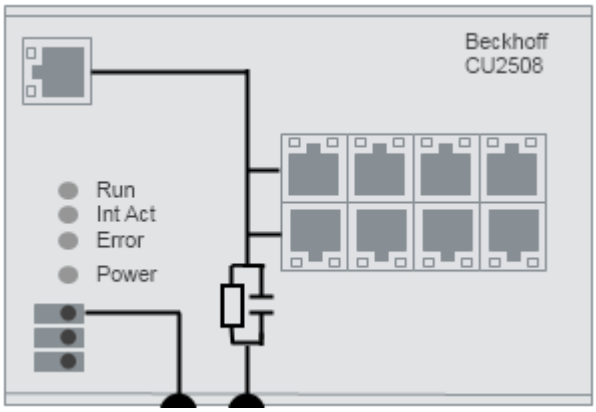


Fig. 22: CU2508 shield concept

3.6 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:

3.7 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.

4 Commissioning

4.1 Setup in TwinCAT

The CU2508 must be created as a MultipleProtocolHandler in the configuration before functions can be assigned to its downlink ports.

Online scanning of the CU2508 is for the time being only possible with EtherCAT systems connected to the downlink ports. The notes in the EtherCAT port definition [▶ 35] regarding the ESI XML files must then be observed.

Setup

Append a new device in the TwinCAT System Manager with a right click.

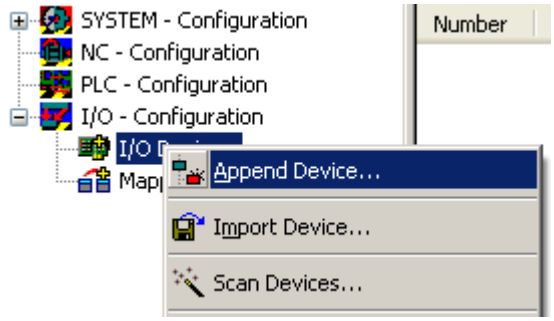


Fig. 23: Appending a new device in TwinCAT 2.11

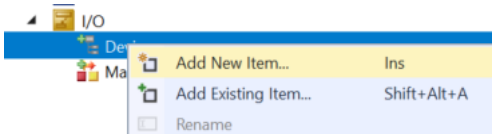


Fig. 24: Appending a new device in TwinCAT 3

Select the "Real-Time Ethernet Adapter (Multiple Protocol Handler)".

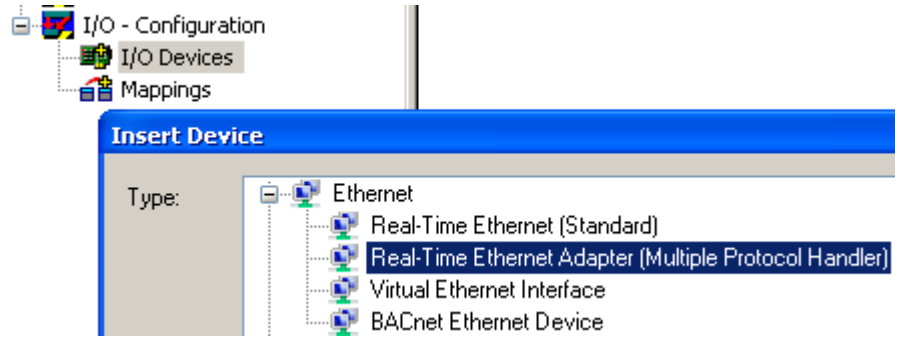


Fig. 25: Selection of the MultipleProtocolHandler in TwinCAT 2.11

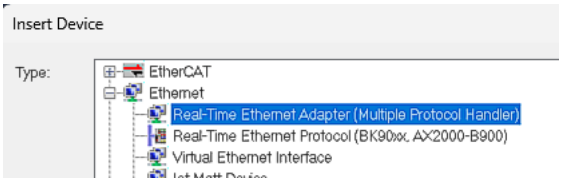


Fig. 26: Selection of the MultipleProtocolHandler in TwinCAT 3

Now select the Gbit uplink port used for the CU2508. You can only select ports on which the Beckhoff real-time driver is installed/activated. <none> can also be selected if the port is to be selected later.



Fig. 27: Selection of the Gbit Ethernet port used on the controller

If the real-time driver is not yet installed, it has to be installed first, e.g. via the System Manager options.

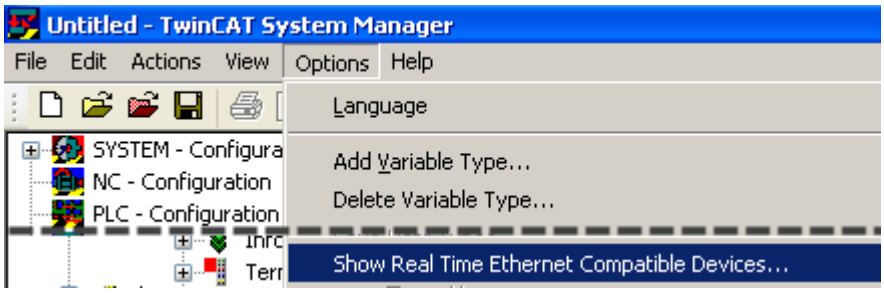


Fig. 28: Installation of the Beckhoff real-time driver

Characteristics that are important for the further operation, e.g. the MAC and the IP addresses of the Gbit network port are now listed in the adapter characteristics of the MultipleProtocolHandler.

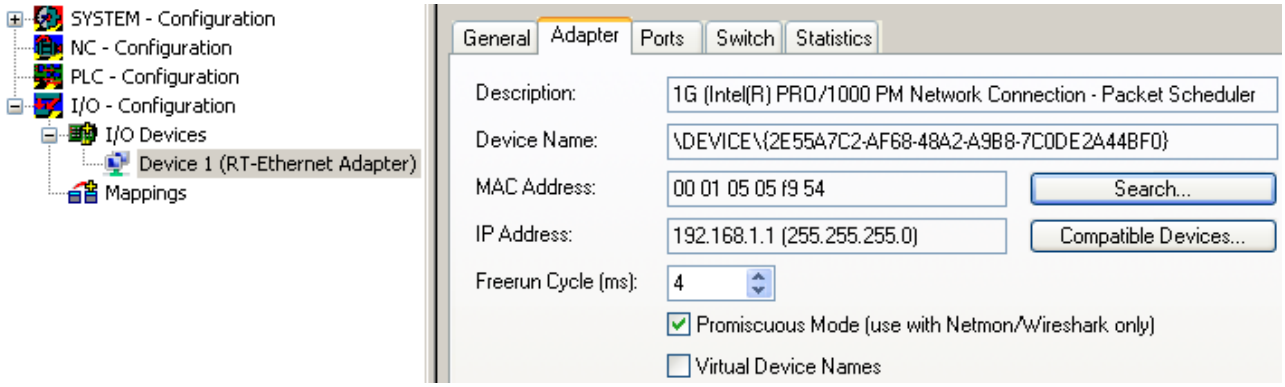


Fig. 29: Adapter data for the Gbit Ethernet port used

Name	Explanation
Description	Name of the adapter in the operating system (the same on several identical PC systems)
Device Name	Unique adapter name (different on several identical PC systems)
MAC Address	MAC address of the hardware adapter; not changeable
IP Address	IP address of the hardware adapter; changeable via the Windows operating system "Network Connection" settings
Freerun Cycle	In ConfigMode, the adapter is operated with this cycle time, i.e. usually 250 cycles/sec.
Promiscuous Mode	If activated, the Ethernet Frames are copied from the real-time context into the Windows NDIS layer and can thus be read by Netmon/Wireshark®
Virtual Device Names	deactivated: "Device Name" is used by TwinCAT for the identification of the network adapter. Thus a TwinCAT configuration is not transferable to another PC system without being changed. *activated*: "Description" is used by TwinCAT for the identification of the network adapter. Thus a TwinCAT configuration is transferable to another PC system without being changed if an adapter with this name is present there.

The number of necessary ports must be set in the port mask (A). The TCP/IP operation mode must also be set (B); “via EoE” must be set for pure EtherCAT operation. See the Notes on TCP/IP settings [▶ 43]. In each case the ESL protocol must be activated for the CU2508 on this Gbit port (C).

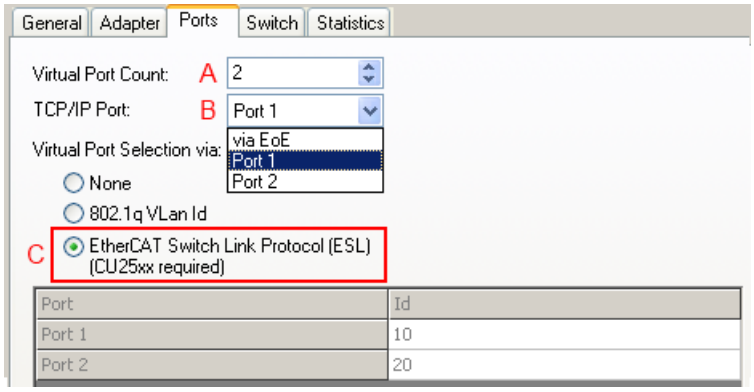


Fig. 30: Number of adapters

The CU2508 has no separate diagnostic variables, since it is designed to be as transparent an infrastructure device as possible.

In the following

- individual downlink ports can now be assigned to protocols
- the TCP/IP behavior of the CU2508 can be set

Diagnostics

The TwinCAT System Manager provide several information about the online operation.

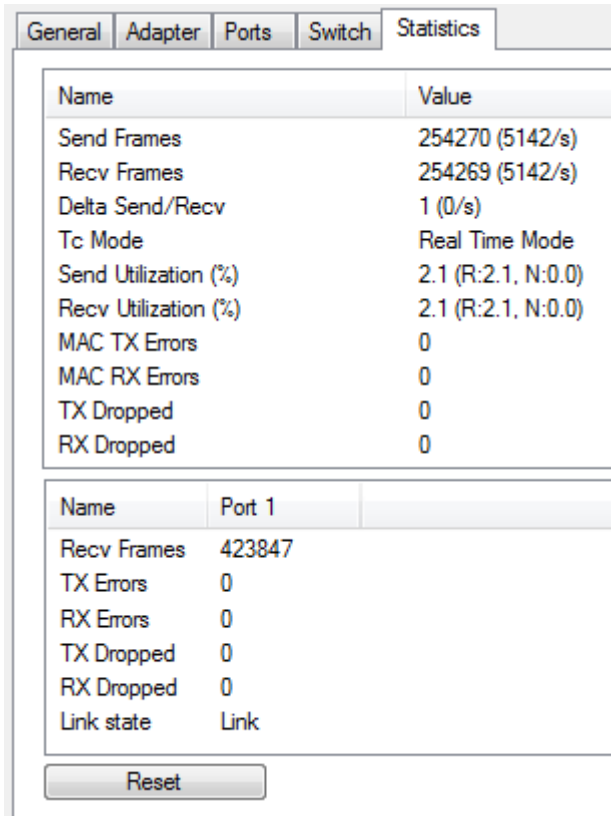


Fig. 31: Statistics

Name	Explanation
Send Utilization (%)	Utilization of the Gbit ESL transmission in send direction R: real-time data, N: non-real-time data
Recv Utilization (%)	Utilization of the Gbit ESL transmission in receive direction R: real-time data, N: non-real-time data

The diagnostic data can also be retrieved via ADS.

Read Rx/Tx error counter	
CMD	Ads Read
Port	10
Index Group	Object ID of the RT Ethernet device + port number (1..8) (for port 1 of the CU from Fig. "Statistics": 0x03010011)
Index Offset	0x03010038
Read data	
UINT32	Tx Error Count
UINT32	Rx Error Count

Reading the port status	
CMD	Ads Read
Port	10
Index Group	Object ID of the RT Ethernet device + port number (1..8)
Index Offset	0x03010042
Read data	
UINT64	Timestamp in distributed clocks (DC) time (ns)
UINT32	Tx Frame Count
UINT32	Rx Frame Count
UINT32	Rx Dropped Frame Count
UINT32	Tx Dropped Frame Count
UINT32	Flags [LinkError = 0x00000001]
UINT32	reserved

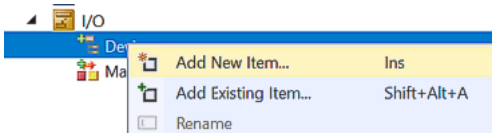


Fig. 32: Object ID of the RT Ethernet device, here: 0x03010010

4.2 Port definition TwinCAT 2.11

4.2.1 Setup: Device EtherCAT

Table of contents
• Notes on the EtherCAT device [▶ 35] • Setting up an EtherCAT device [▶ 36] • Configuration settings under TwinCAT 2.11R2 [▶ 36] • Distributed Clocks behavior [▶ 38] • EoE settings [▶ 39]

A downlink port of the CU2508 is set up as an EtherCAT device. The correct setup of the CU2508 [▶ 32] device is assumed.

Notes on the EtherCAT device

- The CU2508 downlink port appears as the first regular EtherCAT slave. It can optionally represent the Distributed Clocks reference clock in this EtherCAT system. The port has no controllable I/O.
- All CU2508 downlink ports are synchronized with each other.
- A second CU2508 port can be used as a ring closure for EtherCAT cable redundancy. The primary EtherCAT connection must then be connected to an odd CU2508 port (X1, X3 etc.) and the secondary redundancy adapter to the following even CU2508 port (refer to the corresponding chapter Device EtherCAT with cable redundancy [▶ 40]).
- The CU2508 sends EtherCAT frames in pass-through mode without time stamps.

Setting up an EtherCAT device

In EtherCAT operation the CU2508 downlink port used is not transparent, but appears as a Distributed Clocks-capable EtherCAT slave in the configuration. As a result it can appear as a reference clock. However, the EtherCAT slave is otherwise not parameterizable or addressable.

As applies to every EtherCAT device, the current ESI xml files should be present in the TwinCAT system for the CU2508 also. This applies both to the offline creation of the configuration and to scanning in online operation. The files (in particular “Beckhoff CUxxxx.xml”) can be procured from the download area on the Beckhoff web page. They are to be saved under "… \ TwinCAT\Io\EtherCAT\".

Updating the ESI XML files

If the following information window should appear in online mode when scanning the devices on the CU2508: (see Fig. *Information window "Online description"*) then an update for above mentioned file is necessary:
Click on "No", close the System Manager window, install the current ESI xml files and create the configuration in a newly opened System Manager window.

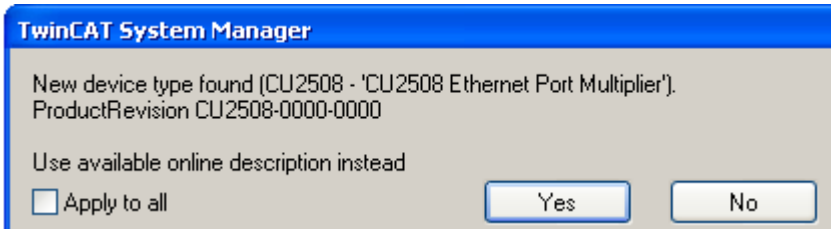


Fig. 33: Information window "Online description"

Creation of the configuration

The simple option to create the configuration by scanning the real hardware is only possible if an EtherCAT device is manually connected to a CU2508 port in the configuration.

Configuration settings under TwinCAT 2.11R2

Creating an EtherCAT device

An EtherCAT device can be added by right-clicking on *I/O Device - > Append Device*.

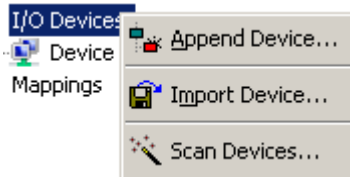


Fig. 34: Creating an EtherCAT device

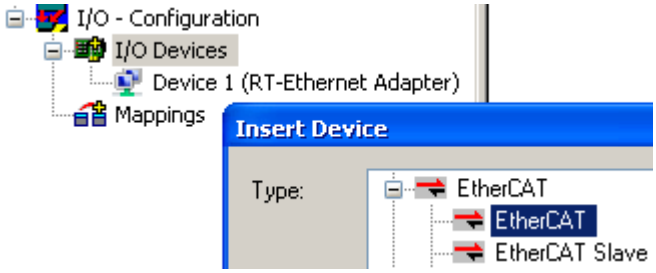


Fig. 35: Selection of the "EtherCAT" device

The desired downlink port of the CU2508 can now be selected.

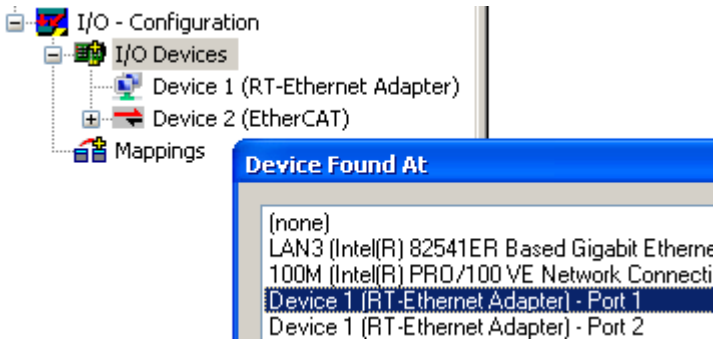


Fig. 36: Selection of the CU2508 port

The device "Device 2 (EtherCAT)" is now connected, according to the adapter reference, not to a physical Ethernet port of the controller, but to a logical adapter reference of the CU2508. This assignment can also be changed later. The freerun cycle time in the TwinCAT ConfigMode is shown here for the adapter.

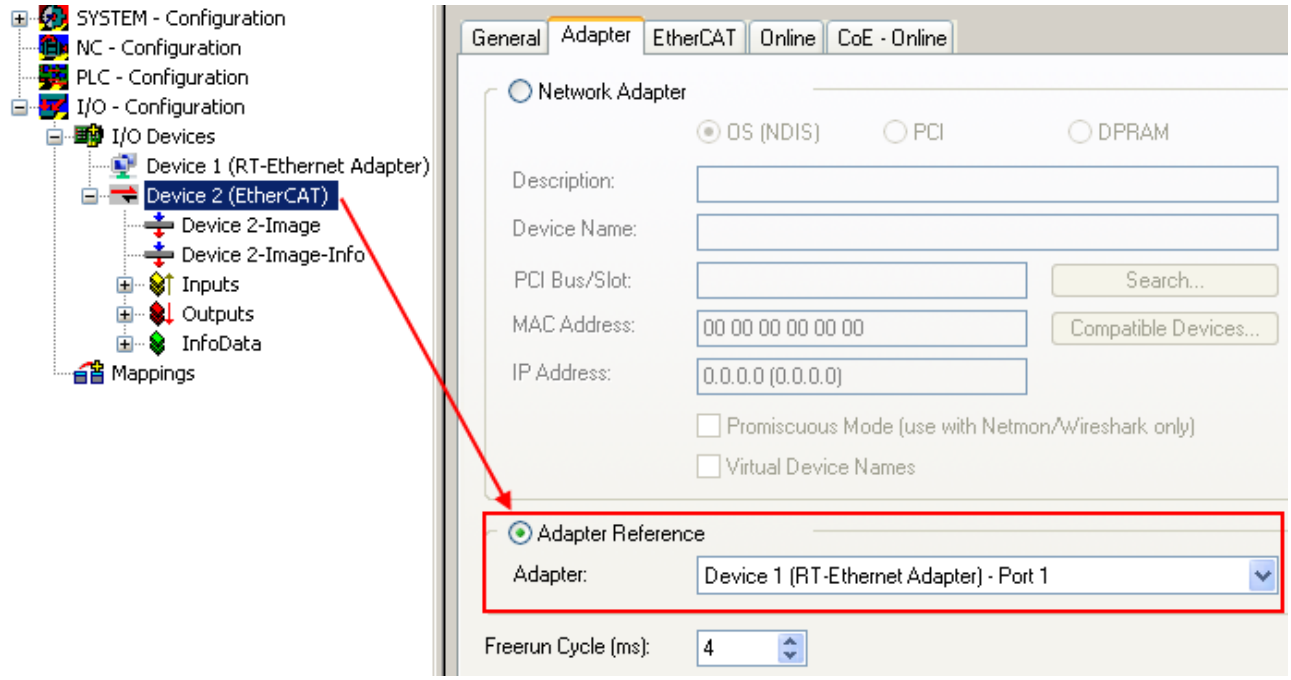


Fig. 37: Adapter setting for the EtherCAT device

Creation of the configuration

Scanning of the EtherCAT configuration is also possible from this state onwards.

Creating a CU2508 box

The CU2508 downlink port appears as an EtherCAT slave and can be scanned or created manually.

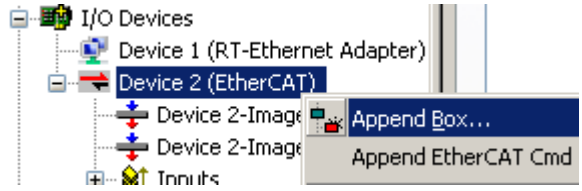


Fig. 38: Appending a CU2508 EtherCAT slave

If creating manually, the “CU2508 Ethernet port” CU2508-0000 must be selected. The check box "ExtendedInformation" displays the revision numbers, in this case -0000 (fig. *Selection of “CU2508 Ethernet Port”*).

If "CU2508" is entered in the *Search* mask, the overview is reduces to the sought device.

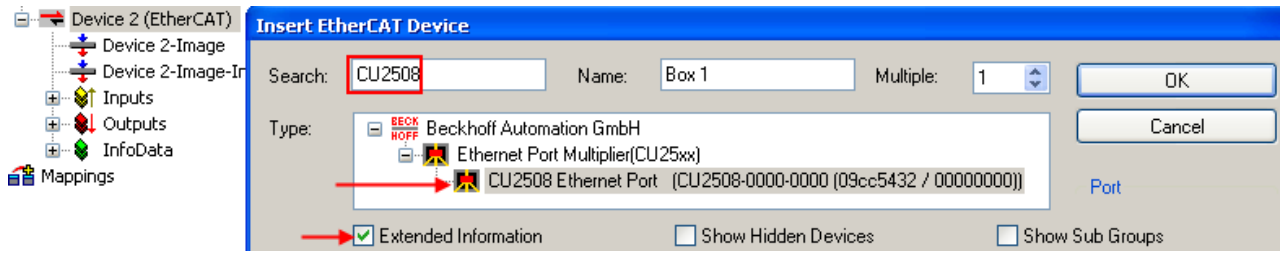


Fig. 7: Selection of "CU2508 Ethernet Port"

Further EtherCAT slaves can now be appended to the CU2508 Box.

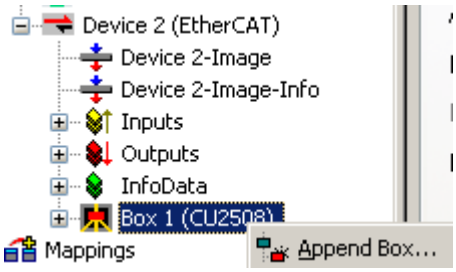


Fig. 39: Appending further EtherCAT slaves

Distributed Clocks settings

All CU2508 downlink ports are synchronized with each other and can appear as ReferenceClock in the respective EtherCAT strand. In the default setting, however, the CU2508 port is not marked as Distributed-Clocks-capable, hence under TwinCAT the next subsequent Distributed-Clocks-capable slave is selected as the ReferenceClock. In each case the CU2508 port is the 1st slave in the respective strand.

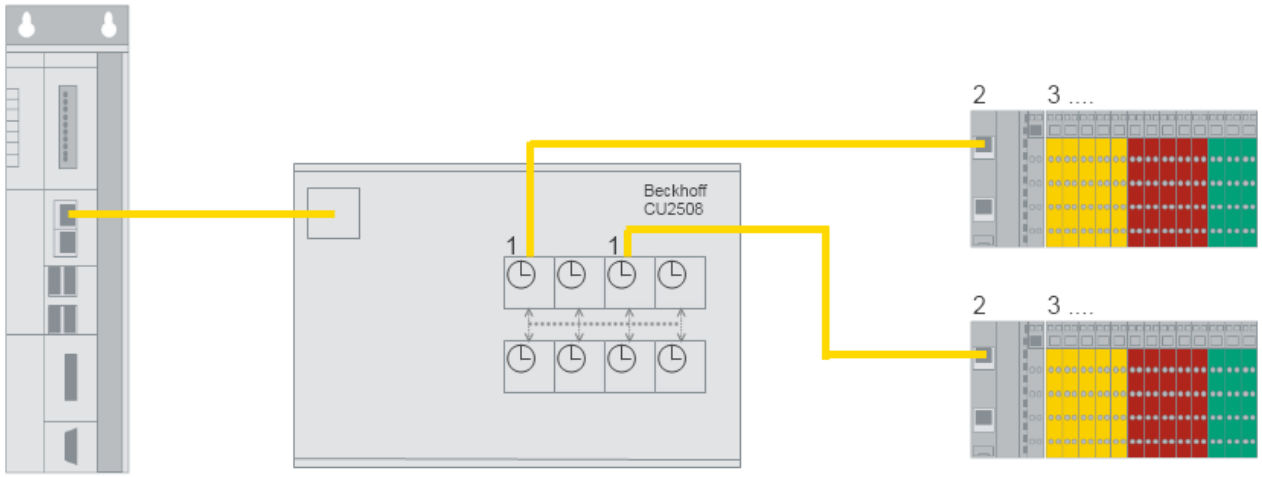


Fig. 40: EtherCAT Distributed Clocks reference

If the CU2508 port is to be used as the ReferenceClock in a system, the setting "Use as potential ReferenceClock" must be changed; see fig. *CU2508 slave - Distributed Clocks setting*.

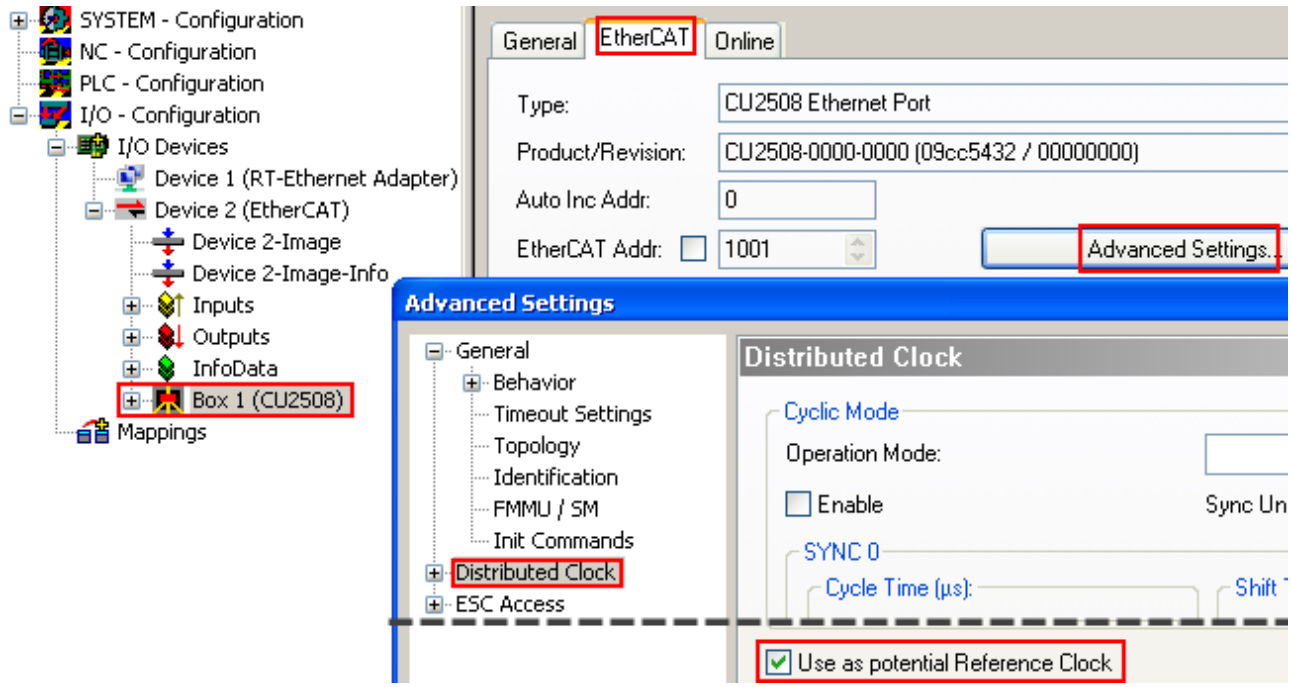


Fig. 41: CU2508 slave - Distributed Clocks setting

Synchronized EtherCAT systems

If several EtherCAT systems are to be operated on CU2508 ports synchronized with the same time base, the flag "Use as potential Reference Clock" must be set in the appropriate "CU2508 box".

External synchronization to another time base is then not possible.

The internal clock in the CU2508 works independently and is only managed by the CU2508 driver.

EoE settings (Ethernet over EtherCAT)

If devices with EoE support are used in a lower-level EtherCAT system, this affects the settings of the EoE switch in the CU2508 if this is configured for EoE transport. Refer to the Notes pages on TCP/IP traffic [▶ 43].

If such devices are present in the EtherCAT system,

- the *Virtual Ethernet Switch* must be enabled (“Enable”)
- “Connect to TCP/IP Stack” can be activated
- the numbers "Max Frames" and "Max MAC IDs" can be increased if throughput problems are determined

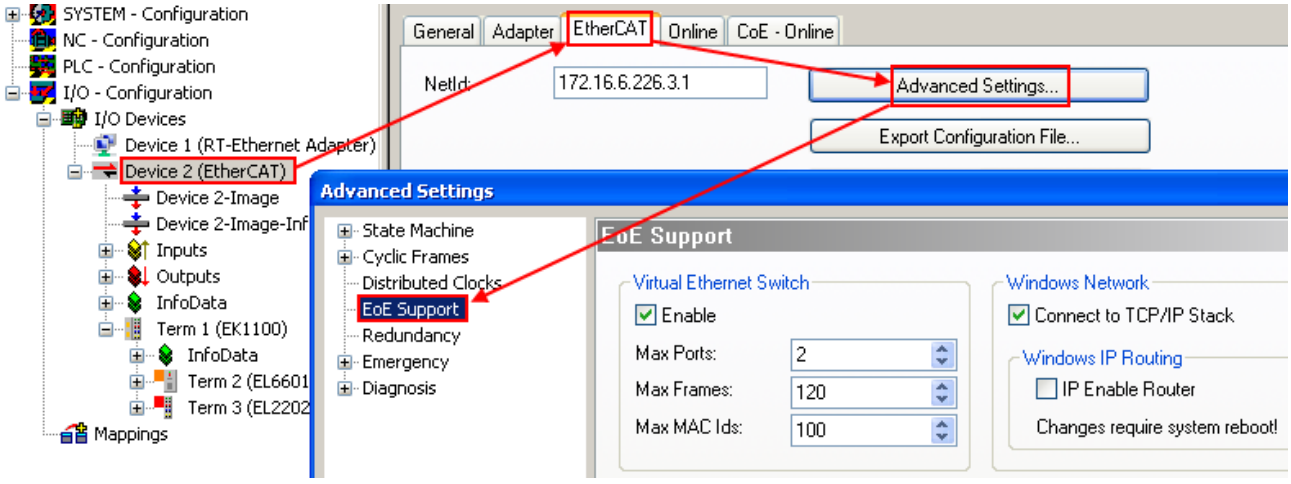


Fig. 42: EtherCat device EoE settings

If several EtherCAT system EoE devices are used, this increases the settings of the CU2508 EoE switch.

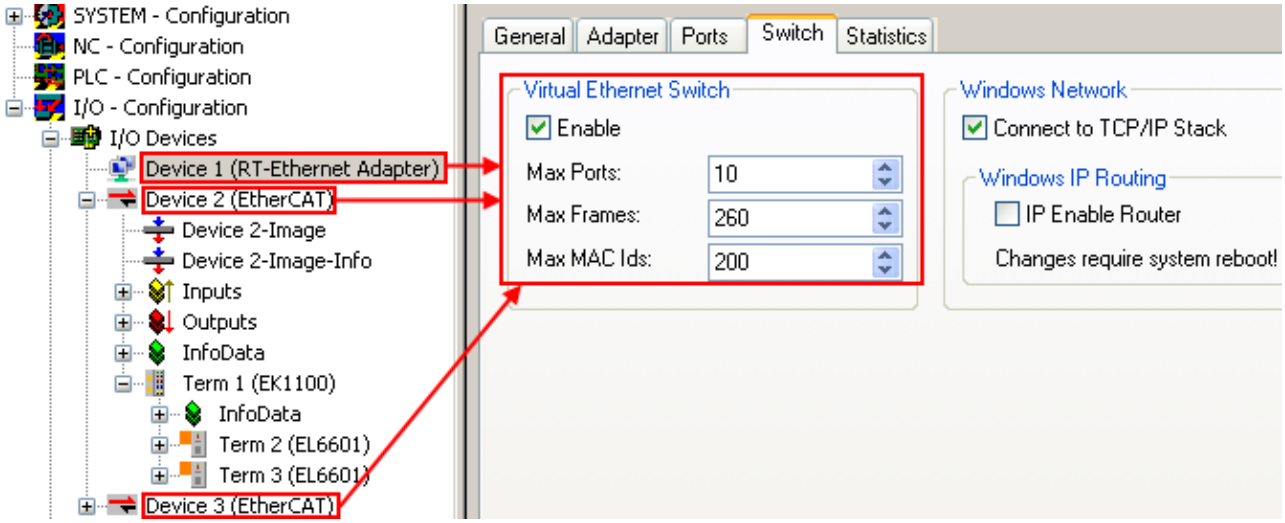


Fig. 43: CU2508 EoE settings

4.2.2 Setting up: Device EtherCAT with cable redundancy

If a downlink port of the CU2508 has been set up as an EtherCAT device, a further port can be used as a 2nd EtherCAT port for the purpose of cable redundancy.

The "TwinCAT cable redundancy" supplement serves to keep the EtherCAT system in operation in the case of an unforeseen interruption in communication due to cable breakage. The system is single-error tolerant. This achieved by the ring closure of the EtherCAT system to the controller. If more than 1 error occurs, slaves drop out of the communication. They are only readmitted after all connections have been restored.

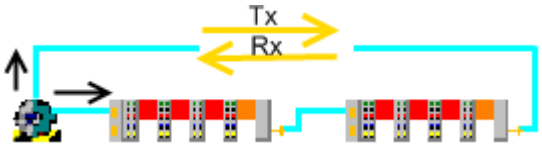


Fig. 44: TwinCAT EtherCAT cable redundancy

Regarding this, refer to the principle explanations in the EtherCAT system documentation.

The combination of the *EtherCAT cable redundancy* and *Distributed Clocks* technologies is possible using the CU2508.

Notes on EtherCAT cable redundancy

- The downlink port combination X1/2, X3/4, X5/6 and X7/8 should preferably be used for EtherCAT cable redundancy. However, if ports from groups X1..4 and X5..8 are mixed, the total internal processing time increases from approx. 2 µs to approx. 3 µs. This may need to be taken into account in applications with very short EtherCAT cycle times.
- If EtherCAT cable redundancy is activated, the CU2508 ports used is automatically set as the Distributed Clock ReferenceClock. The synchronization of the devices can thus be maintained even in the case of media interruption.
- During EtherCAT startup with Distributed Clock devices, the cable-redundant ring must be closed, and continuous communication must be possible, otherwise no synchronization is possible.
- An external Distributed Clocks synchronization of such an EtherCAT system is not possible for the time being.

Setting up EtherCAT cable redundancy

The EtherCAT configuration must be created on a CU2508 downlink port as described in the chapter "Setup: Device EtherCAT [▶ 35]". The resulting topology can be displayed in the Device EtherCAT using the *Topology* button; see fig. *Display of a simple EtherCAT topology*.

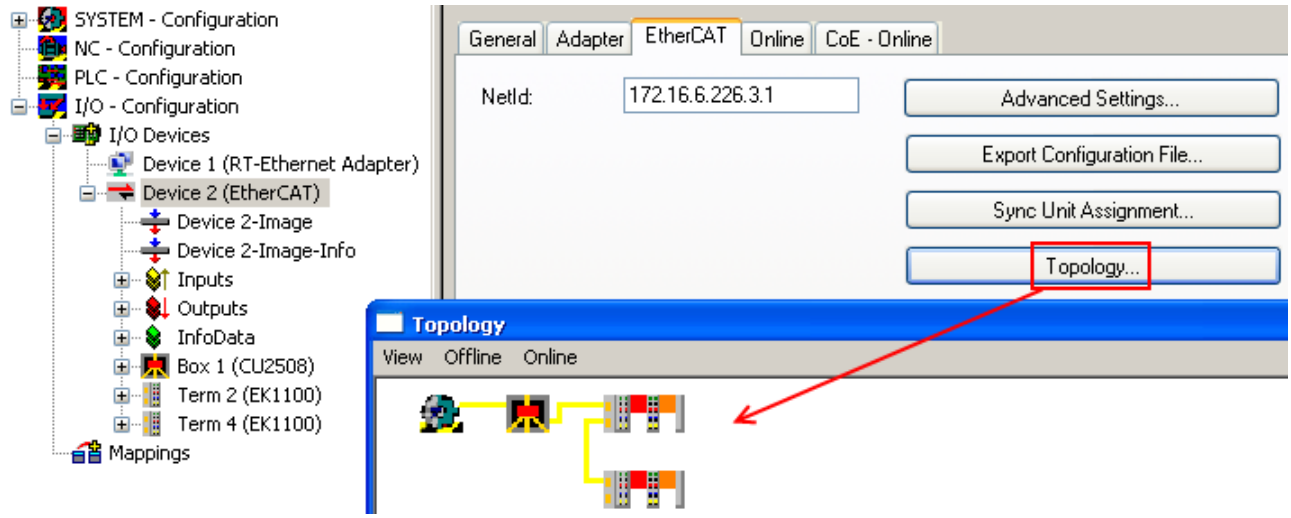


Fig. 45: Display of a simple EtherCAT topology

The items below are to be specified in the following steps

- the control-side redundancy adapter, in this case on the CU2508
- the field-side redundancy adapter on the last EtherCAT slave

The EtherCAT redundancy port is specified in Device -> EtherCAT -> Advanced Settings.

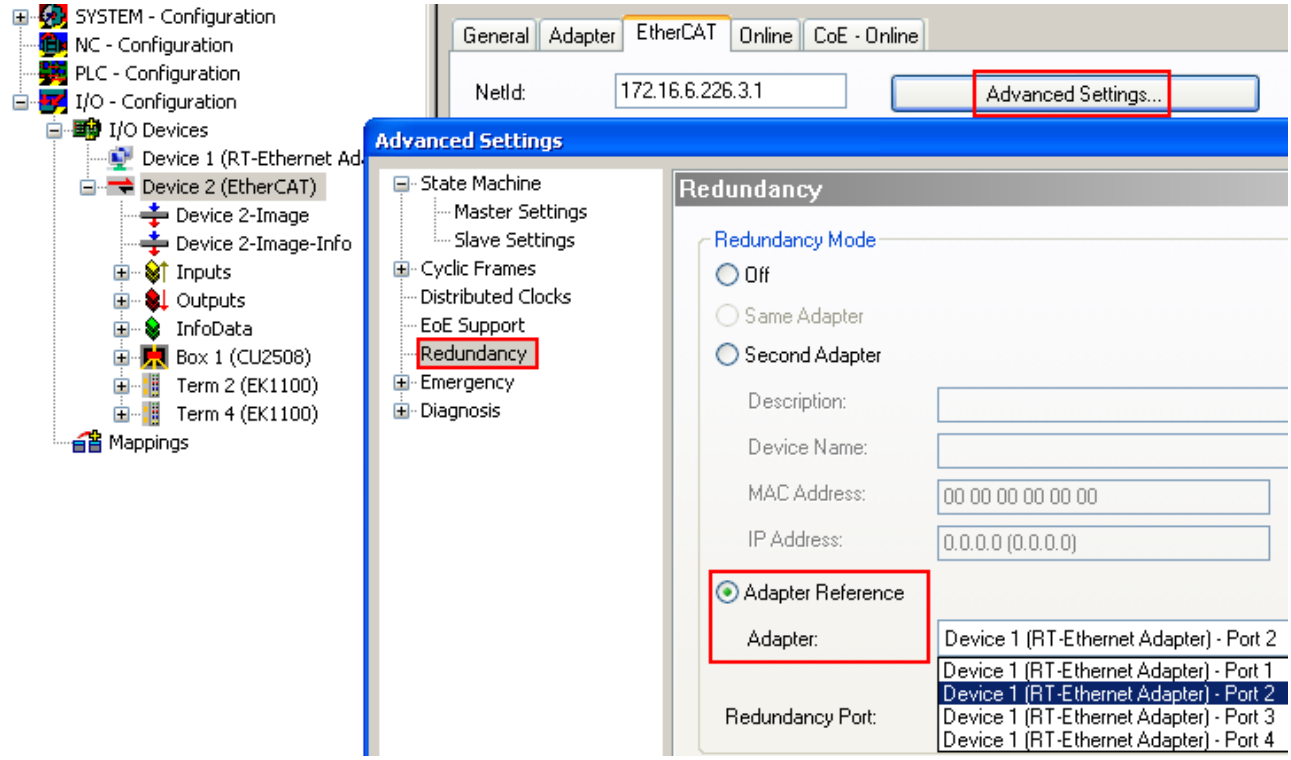


Fig. 46: Specification of the redundancy port

Specification	Meaning
Off	no EtherCAT cable redundancy
Second Adapter	physical adapter/Ethernet port of the controller
Adapter Reference	logical adapter such as in the CU2508

Number of ports

A corresponding number of ports must be set to available in the settings for the CU2508 adapter.

After specifying the redundancy adapter, it must be checked whether the System Manager has selected the correct EtherCAT slave port as the field-side connection point, see fig. *Specification of the field-side redundancy port*.

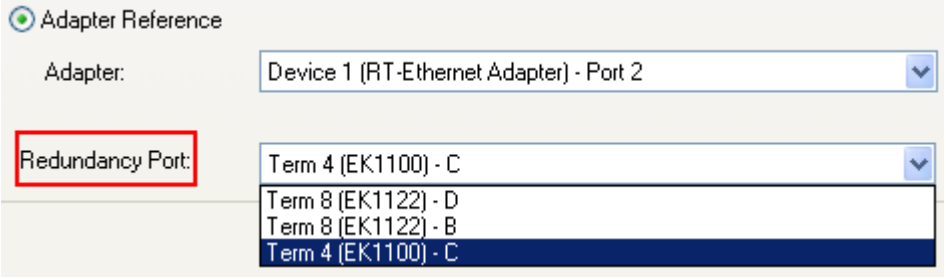


Fig. 47: Specification of the field-side redundancy port

The correct selection can be checked in the Topology window.

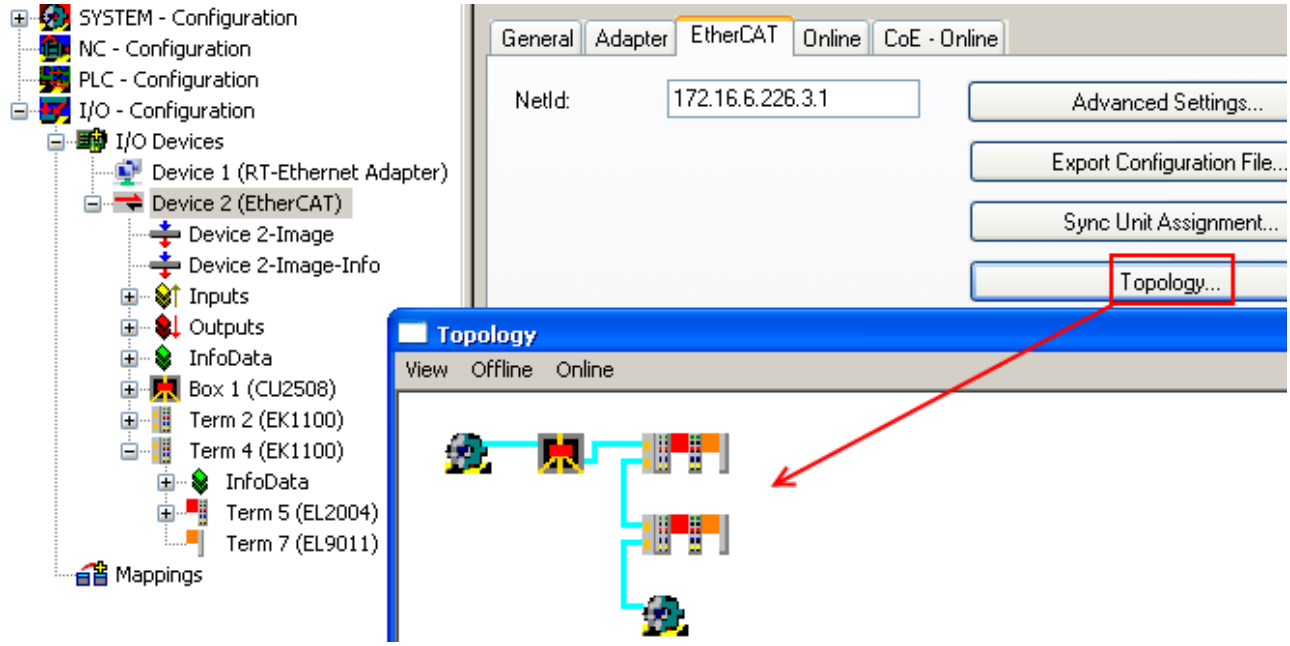


Fig. 48: Checking topology with EtherCAT cable redundancy

4.2.3 Setting up: Device Ethernet and TCP/IP notes

Table of contents

• Explanation of TCP/IP v4 [▶ 43]
• Application CU2508 [▶ 43]
• Settings for the dedicated TCP/IP port [▶ 44]
• EoE settings [▶ 44]

Explanation TCP/IP v4

TransmissionControlProtocol/InternetProtocol (TCP/IP) is the name given to a bundle of procedures from the 1970s, which defines a packet-mediated forwarding of data. This mediation procedure is used in contemporary everyday communication, in particular on the Ethernet physics (e.g. Fast Ethernet, Gbit) and the hardware-orientated MAC addressing normally used there. The basic principles of this set are

- packets of user data of 64 to 1486 bytes (special forms are not uncommon) are sent from a transmitter to one (unicast) or more (multicast, broadcast) receivers.
- In IPv4 the source and destination addresses each consist of 4 bytes separated by a point, e.g. "192.168.1.1" or "123.001.002.001".
  Note: IPv6 extends the address space by a further 2 bytes and will replace IPv4 in the mid-term.
- There does not have to be a direct physical connection (cable) between transmitter and receiver; instead, intelligent packet mediators analyze the source and destination addresses and forward the packets (key words: routing and switching). This results in delays and possible loss of data.
- The IP is a bi-directional reconfirmed communication, because the receiver acknowledges the receipt of data to the transmitter. This causes a further communication load and increased transport time. Lost data can thus be repeated automatically.
- UDP (User Datagram Protocol) can also be used instead of IP. It works without reconfirmation, lost data must then be requested again at a higher communication level. In return, less time is required for data communication.

The Windows operating system manages TCP/IP communication with its Ethernet adapters in the so-called NDIS network mediation layer. The adapter settings can be accessed via the Windows Control Panel->Network Settings or the Windows CE CXconfig tool from Beckhoff.

These circumstances make TCP/IP-supported communication unsuitable for an application with higher real-time requirements, but very suitable for flexible (variable) connections via many hubs (intermediate stations) with an arbitrarily high data traffic.

Application CU2508

Two methods are available if TCP/IP frames are to be transmitted via the CU2508 (with no real-time requirement):

- one port of the CU2508 is defined as a dedicated TCP/IP port [▶ 44].
  Hence, it works for the operating system as an "extension" of the port to which the CU2508 is connected, with the MAC and IP data of the Gbit port used according to the operating system data. Under TwinCAT with its integrated CU2508 driver, this port is then connected to the Windows NDIS mediation layer in the Windows operating system. Real-time traffic is then no longer possible via this port; the CU2508 works as a transparent frame transporter.
- the CU2508 is generally set to EoE operation [▶ 44] (Ethernet over EtherCAT), “via EoE”.
  Then the integrated virtual EoE switch is activated in the CU2508 system and the connection is established to the EtherCAT systems connected to the CU2508. In the case of EoE, Ethernet frames are transported within acyclic EtherCAT telegrams by mailbox procedures, fragmented if necessary in the case of overlength. The EL6601 or EL6614 can be used as an EtherCAT/Ethernet interface.
  The MAC and IP data of the Gbit port used according to operating system data are used for addressing.
  This procedure is usable only for connected EtherCAT circuits.

The setting is made in the *Ports* tab for the CU2508 and does not change the mode of operation of the CU2508, but only the handling of TCP/IP Frames in the CU2508 driver.

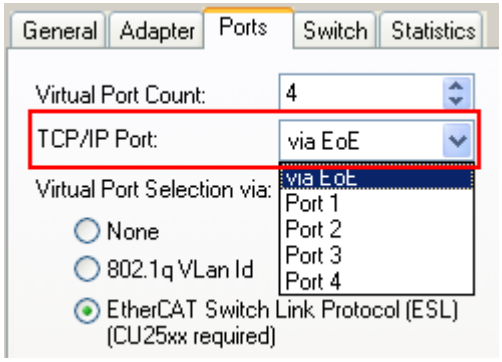


Fig. 49: Setting the TCP/IP operating mode

It is not possible to operate the CU2508 as an independent switch with forwarding rules according to MAC rules (switch) or IP rules (router).

Settings for the dedicated TCP/IP port

The specified CU2508 port (100Mbit/FastEthernet) is then connected directly to the Windows network mediation layer; real-time operation of this port is no longer possible. Free TCP/IP network traffic is then no longer possible on the other CU2508 ports.

Notes

- The IP and MAC address of the used Gbit port are used for TCP/IP communication. Address area/ subnet mask must therefore correspond to the addressed devices.
  In the example in fig. *Declaration of a CU2508 downlink port as a TCP/IP port*, the address area is under the IP mask 255.255.0.0; therefore, all devices must have an IP address from the area 192.168.x.x.
- These settings are changed via the Windows Control Panel -> Network Settings for the adapter. The CXconfig tool must be used under Windows CE.
- The CU2508 works transparently; all procedures for the address assignment (DHCP, BootP,…) can be used.
- The throughput depends on the extent of utilization of the downlink port and the Windows network mediation layer.

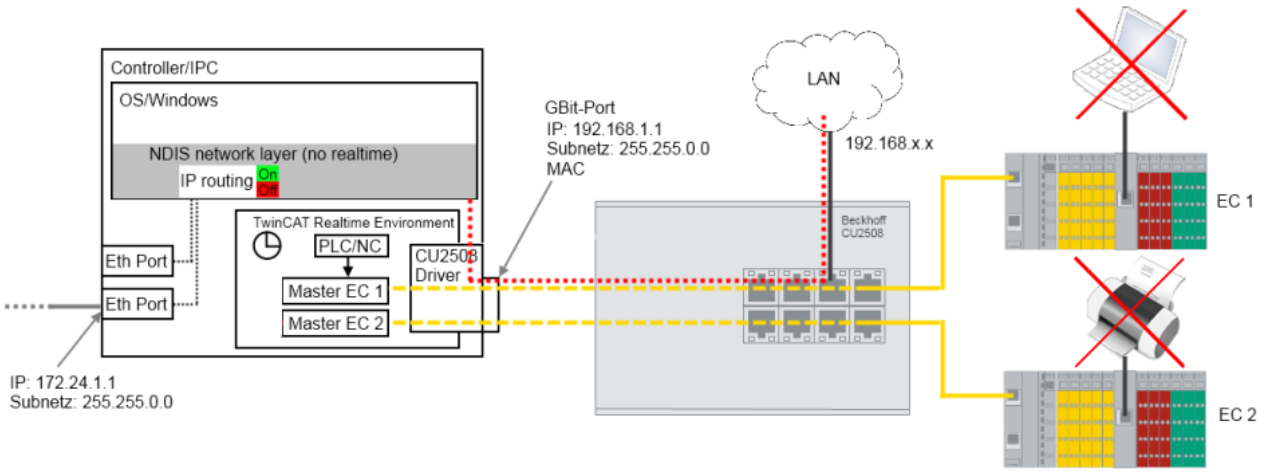


Fig. 50: Declaration of a CU2508 downlink port as a TCP/IP port

EoE settings

The virtual EoE switch of the CU2508 driver is tied into the CU2508 Ethernet adapter and accessible in the *Switch* tab of the RT Ethernet adapter. The VirtualEthernetSwitch is activated automatically if devices that use Ethernet-over-EtherCAT are used in the lower-level EtherCAT systems - the numbers of max. frames and ports are adapted accordingly.

Notes

- The IP address of the used Gbit adapter and the subnet mask is used for communication. The EoE devices (printer, scanner, remote PC etc.) must therefore have their address within this area.
- The throughput depends among other things on the EtherCAT cycle time, the number EoE devices and the extent of utilization of the individual connections. Attention must be paid to the notes regarding this in the documentation for the EoE devices.
- The EoE system can be connected to the NDIS layer via the switch "Connect to TCP/IP Stack" (fig. *Flow of data when using the CU2508 as an EoE switch, A*).
- IP routing is a function inherent to the operating system for the mediation of IP telegrams even between networks that do not lie in the same subnet. Thus, for example, between the devices 192.168.1.1 and 172.168.1.1, which each lie in the mask 255.255.0.0.
  This function is activated with "IP Enable Router" (fig. *Flow of data when using the CU2508 as an EoE switch*, B). It is also available under Windows CE (CXconfig tools).

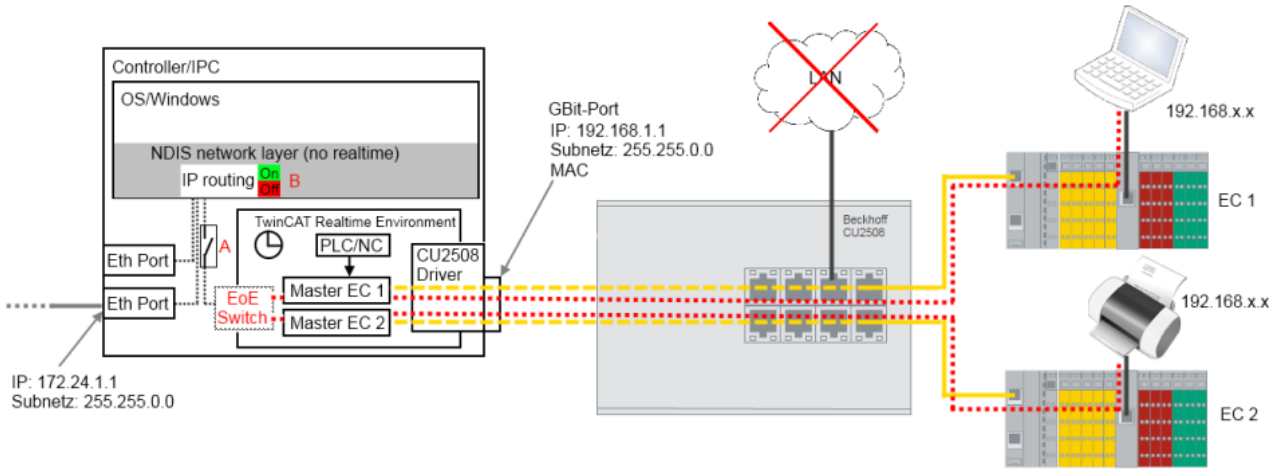


Fig. 51: Flow of data when using the CU2508 as an EoE switch

The switch settings for the EoE mediation are accessible in the System Manager, see fig. *EoE switch settings*. The limit values are adapted automatically to the types and numbers of the devices during the creation of the configuration.

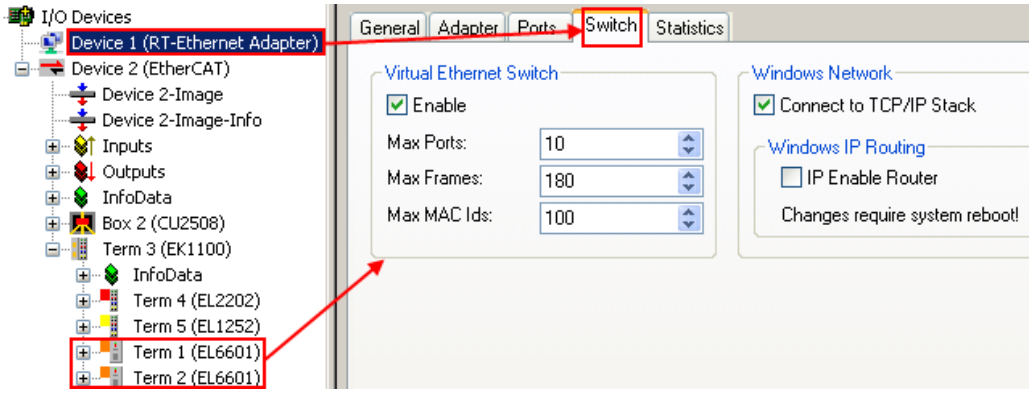


Fig. 52: EoE switch settings

Name	Explanation
Max. Ports	Number of ports of the virtual switch. Each EoE device (e.g. EL6601, AX5000) in the lower-level EtherCAT systems occupies one port.
Max Frames	Max. number of temporarily stored Ethernet frames. Can be increased if there are noticeable throughput bottlenecks.
Max MAC Ids	MAC addresses of the connected devices stored in the internal Switch-Look-Up table. Must be larger than the maximum number of Ethernet devices ever connected to all systems.
Connect to TCP/IP Stack	Connects the EoE switch to the internal Windows NDIS network layer IP range must thereby correspond
IP Enable Router	This function is provided by the Windows operating system and is accessible via the registry or, in the case of embedded CX systems under Windows CE/WEC, via the CX-config tool. If activated, the NDIS network layer also mediates IP packets whose IP addresses do not correspond to the subnet mask .

4.2.4 Setting up: Real-time Ethernet (Publisher/Subscriber, EAP push)

Table of contents

- Notes on the device [▶ 46]
- Setting up the device [▶ 46]

Notes on the device

- An EAP adapter must be created in the configuration in order to operate a CU2508 downlink port as a publisher/subscriber device of EAP real-time Ethernet telegrams ("network variables").
- Attention is to be paid to the general notes on the EAP/Publisher/Subscriber procedure [▶ 50].
- The EAP (EtherCAT Automation Protocol) is operated here in the so-called push mode as opposed to the so-called polling mode for flexible data transfer.
- At least TwinCAT 2.11R3 is required for operation on the master side.

Setting up the real-time Ethernet device

For the setup a real-time Ethernet adapter is to be created as usual.



Fig. 53: Creating a new device

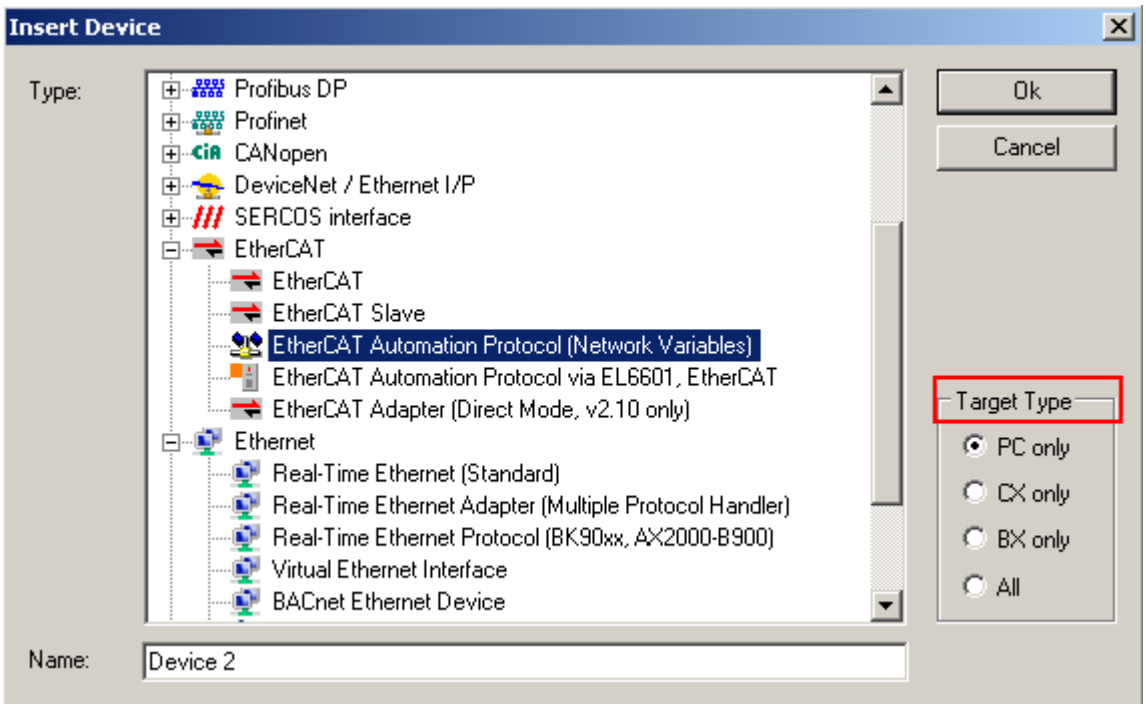


Fig. 54: Selection of the adapter for EAP

If not all devices are offered, the target platform can be selected via *TargetType*.

Not an actual port of the PC, but rather a virtual adapter of the CU2508 is then merely selected as the Ethernet adapter for the CU2508.

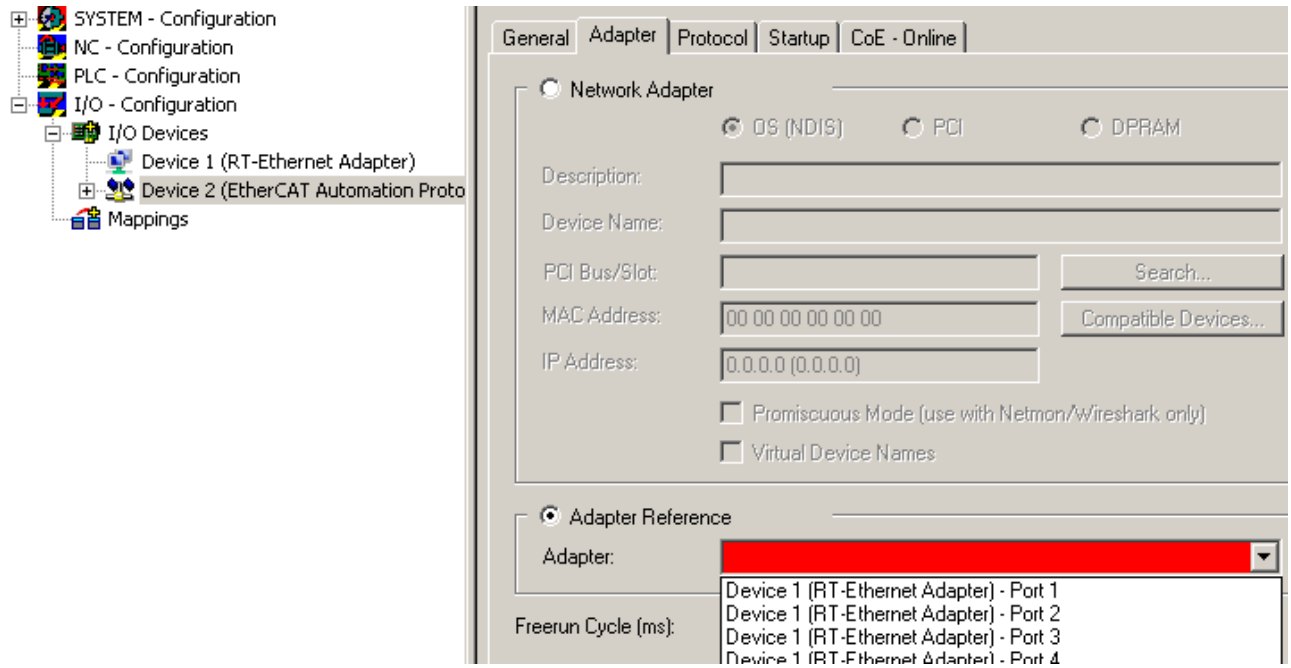


Fig. 55: Selection of the Ethernet adapter

The number of available ports (in this case 4) depends on the settings of the CU2508 device (see there).

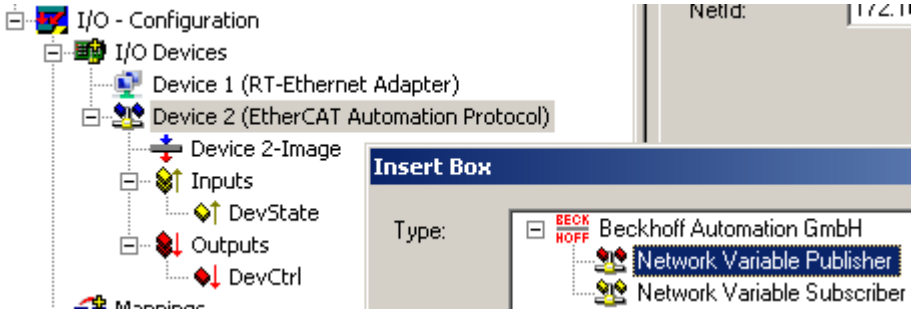


Fig. 56: Adding the Ethernet devices

Subsequently the required Ethernet couplers/boxes/drives can be manually added or scanned. Before a scan TwinCAT should be restarted once in the Config/FreeRun mode.

4.2.5 Setting up: Real-time Ethernet (BK90xx, ADS/RAW, EAP polling)

Table of contents
• Notes on the device [▶ 47] • Setting up the device [▶ 47]

Notes on the device

- An EAP (poll) adapter must be created in the configuration in order to operate a CU2508 downlink port as an EAP real-time Ethernet interface for IO devices (BK90xx, EK90x).
- The EAP (EtherCAT Automation Protocol) is operated here in the so-called polling mode
- At least TwinCAT 2.11R3 is required for operation on the master side.

Setting up the real-time Ethernet device

For the setup a real-time Ethernet adapter is to be created as usual.

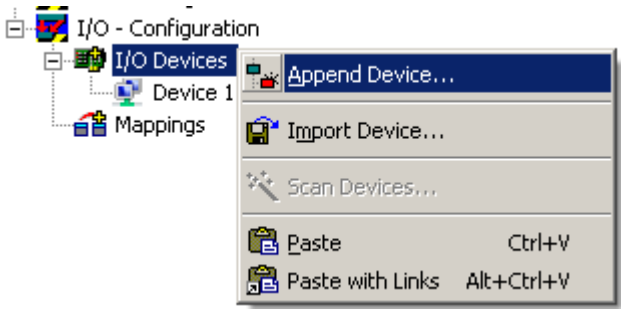


Fig. 57: Creating a new device

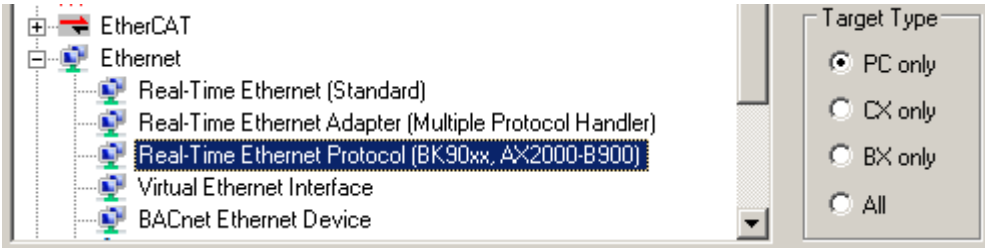


Fig. 58: Selection of the device for BK90xx and others

Not an actual port of the PC, but rather a virtual adapter of the CU2508 is then merely selected as the Ethernet adapter for the CU2508.

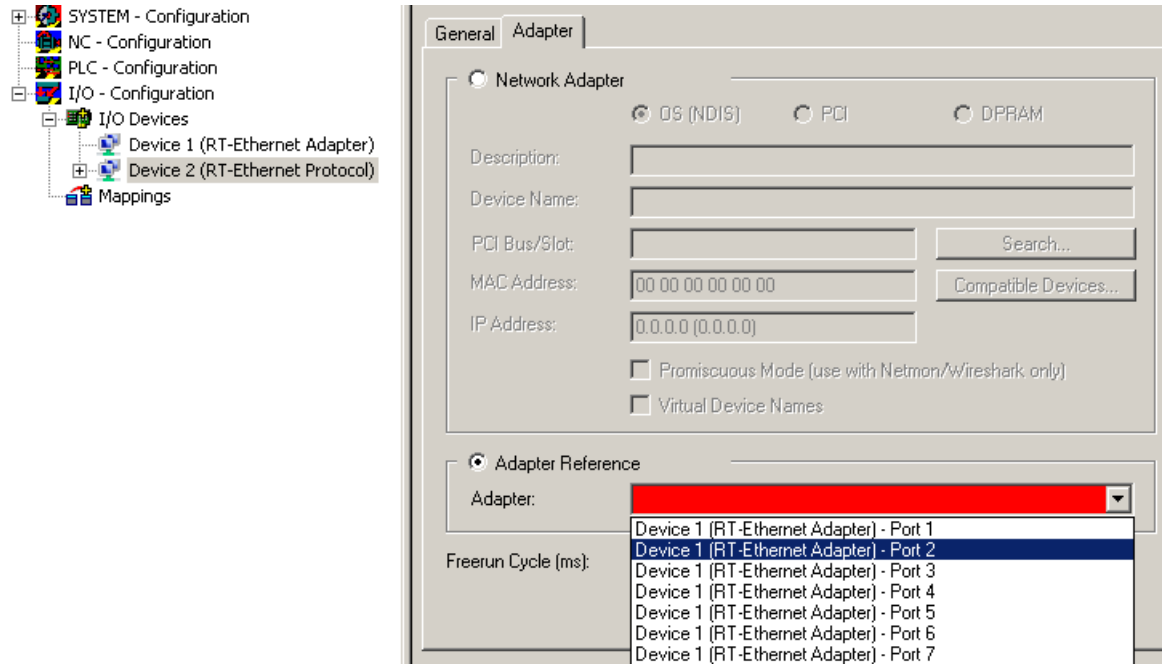


Fig. 59: Selection of the CU2508 port to be used

The number of available ports (in this case 7) depends on the settings of the CU2508 device (see there).

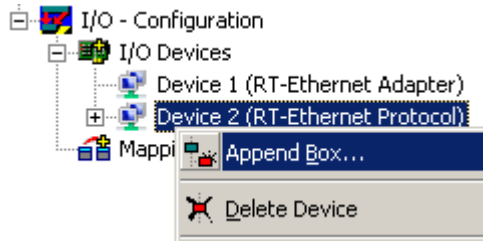


Fig. 60: Adding the Ethernet devices

Subsequently the required Ethernet couplers/boxes/drives can be manually added or scanned. Before a scan TwinCAT should be restarted once in the Config/FreeRun mode.

4.2.6 Setting up: PROFINET Controller/Device

The usage of the PROFINET real-time protocol is possible for the device (slave) or controller (master) supplement with the CU2508. However, you can use only one controller or device, if they are in a unique network.

It is possible to use more than one controller or device supplement with one CU2508. The networks need to be physically separated, due to the routing of one MAC address in total.

The function of several PROFINET segments can only be secured, if the MAC address in a network segment is unique and there is no connection to other networks.

Availability

This function can be used from TwinCAT 2.11R3 build 2234.

RT Ethernet adapter settings

Enable the lower setting under the Ports tab (radio button "EtherCAT Switch Link Protocol"); all other settings should be left at default.

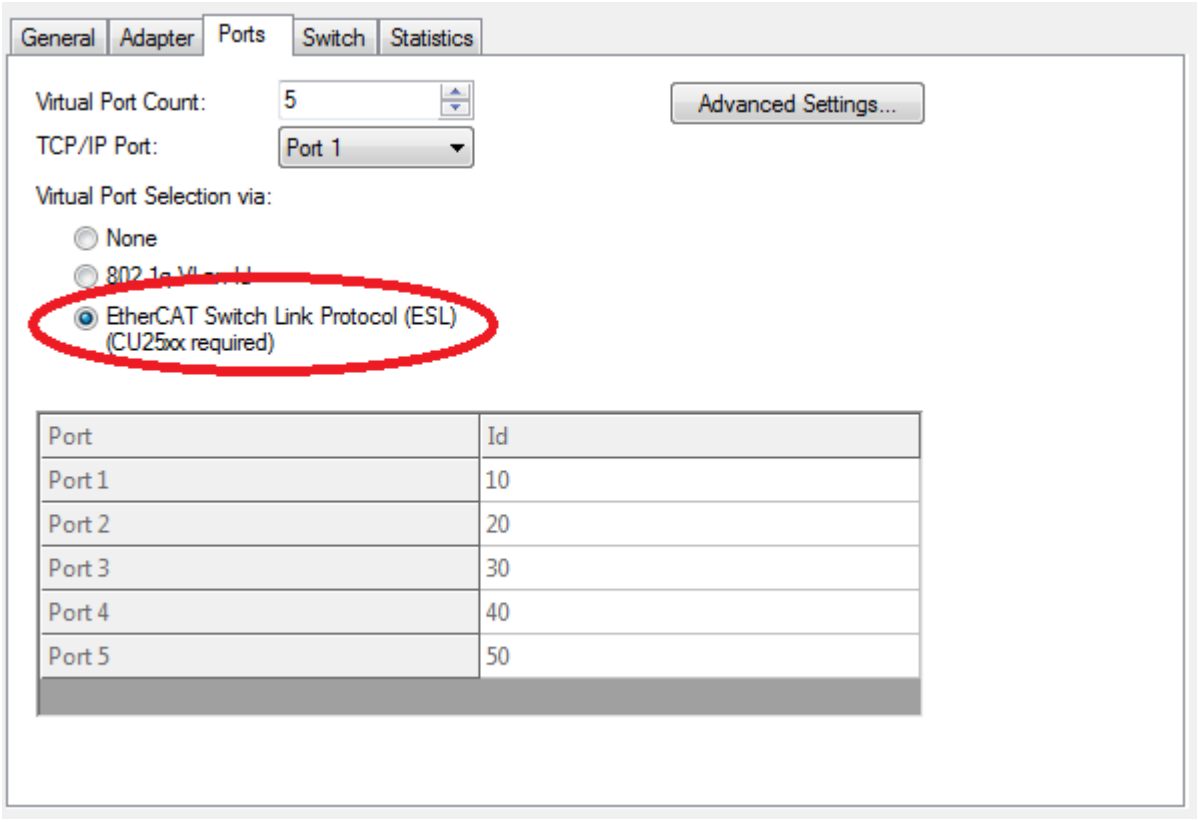


Fig. 61: Selection of the "Virtual Port" setting

Setting of the PROFINET Device or Controller adapter

Once the CU2508 has been activated as Multiple Protocol Handler in the system and the relevant ports have been enabled, you can select an "Adapter Referent" under the "Adapter" tab for the Controller or the Device. Enter the port through which you want to operate PROFINET.

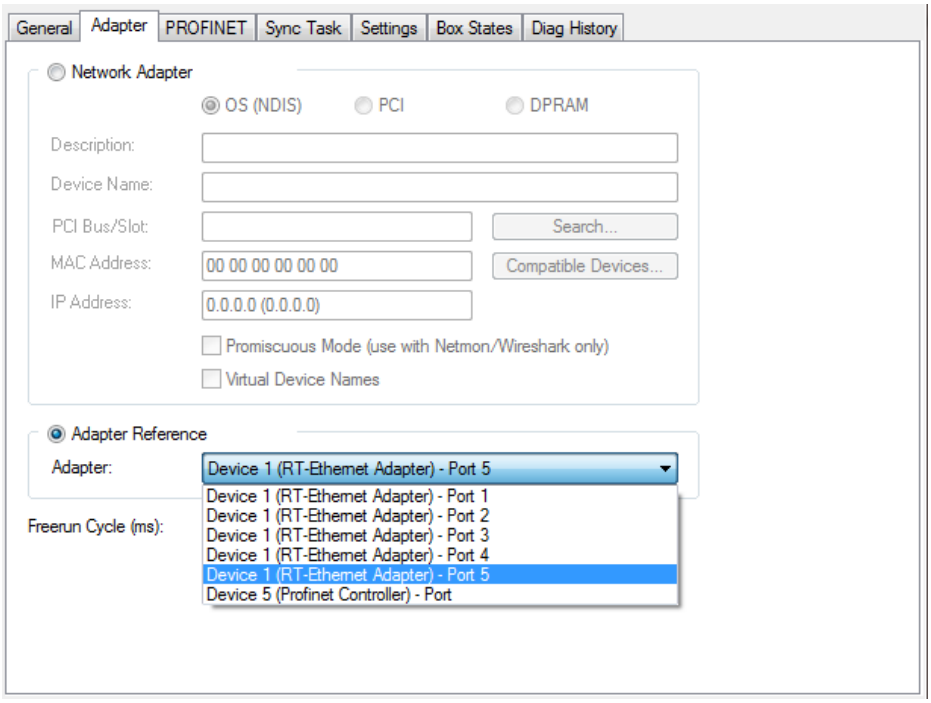


Fig. 62: Selecting the "Adapter Referent"

No other special considerations are required for the CU2508; you can proceed as usual.

4.2.7 Configuration: Ethernet/IP

Ethernet/IP is currently (2020) not supported.

4.3 Beckhoff EAP network variables

4.3.1 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.

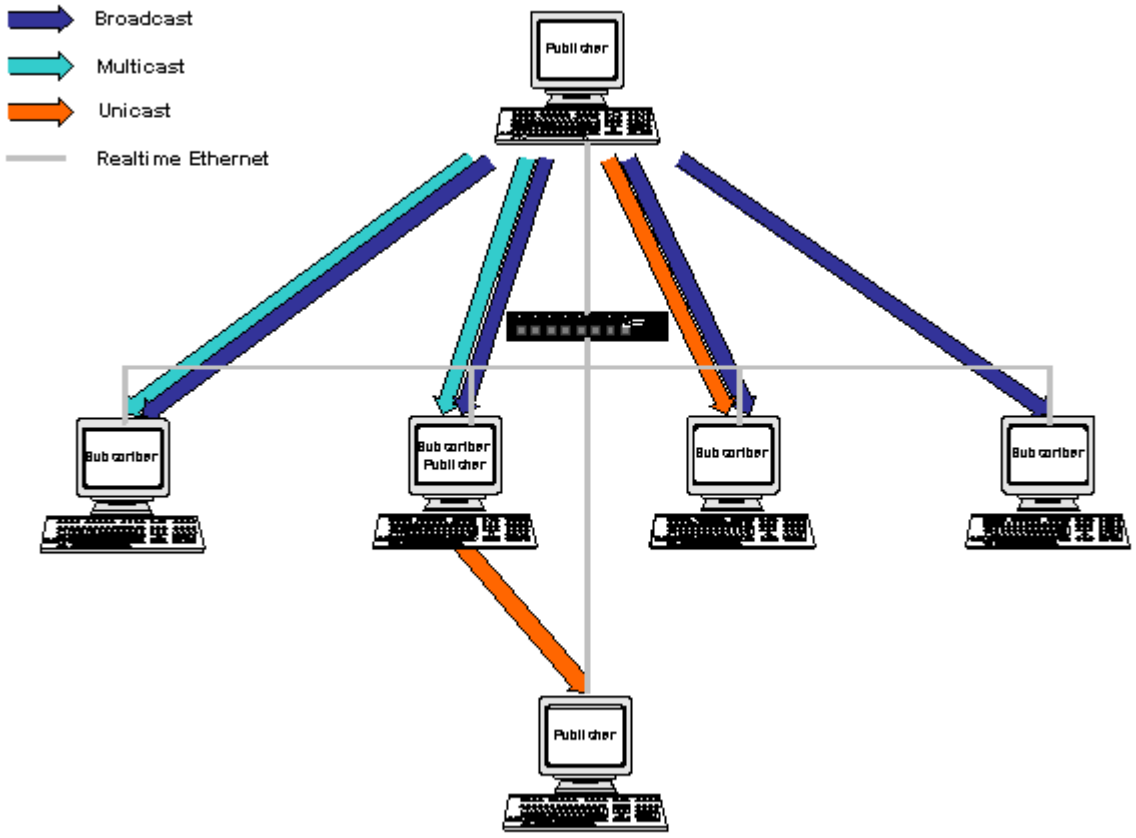


Fig. 63: Publisher/Subscriber model

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.

4.3.2 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.

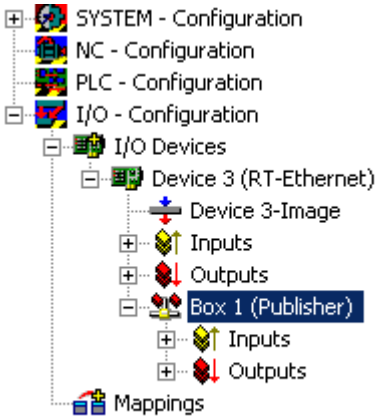


Fig. 64: Insert a Publisher Box

Insert a Network Variable

Network variables can then be added under the box. Enter a name (nCounterPub in the example) and a data type (UINT32 in the example, corresponding to UDINT).

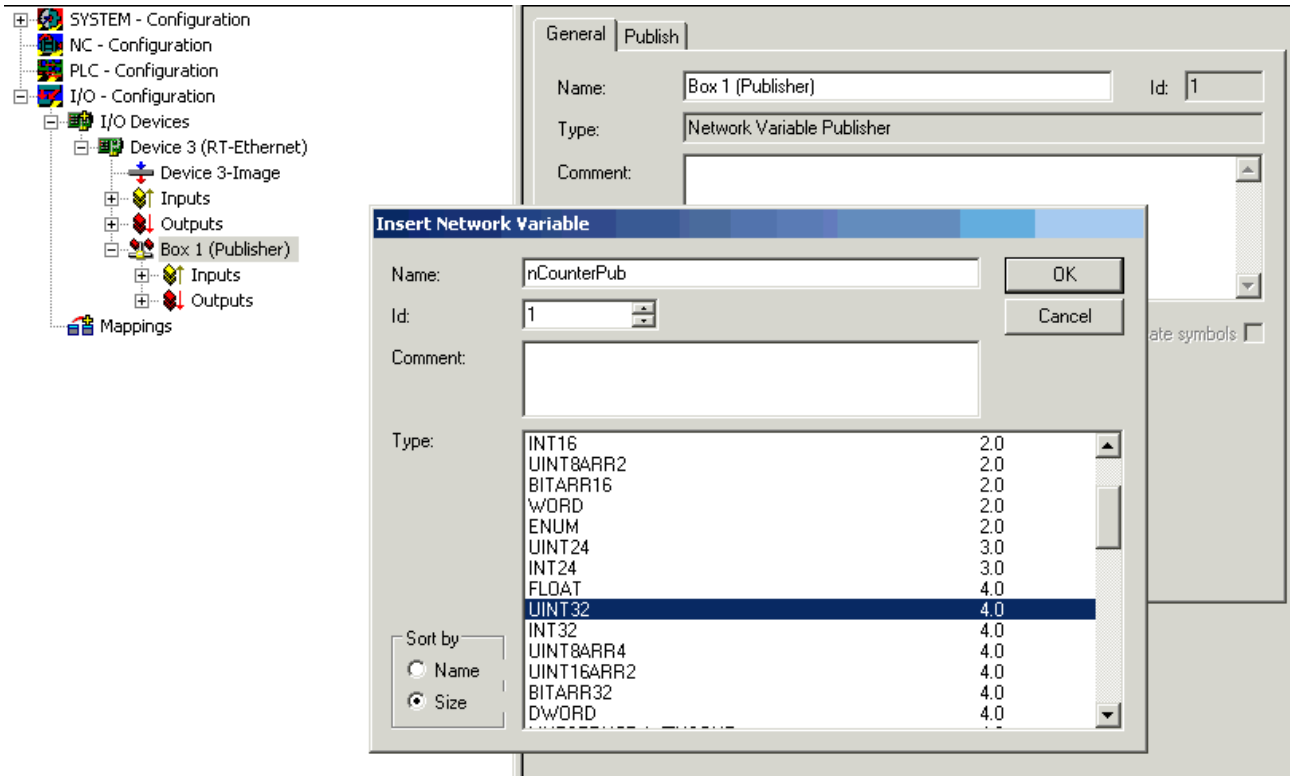


Fig. 65: Insert a Network Variable

Inputs and outputs were created under the added variable.

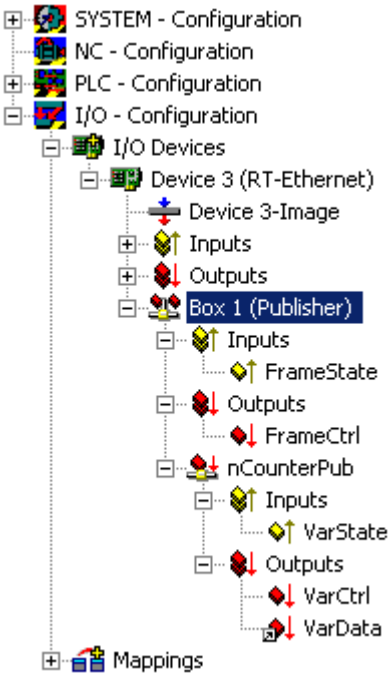


Fig. 66: Displaying the inputs and outputs of the added Publisher box

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 example, the network variable is linked to the output variable of a PLC.

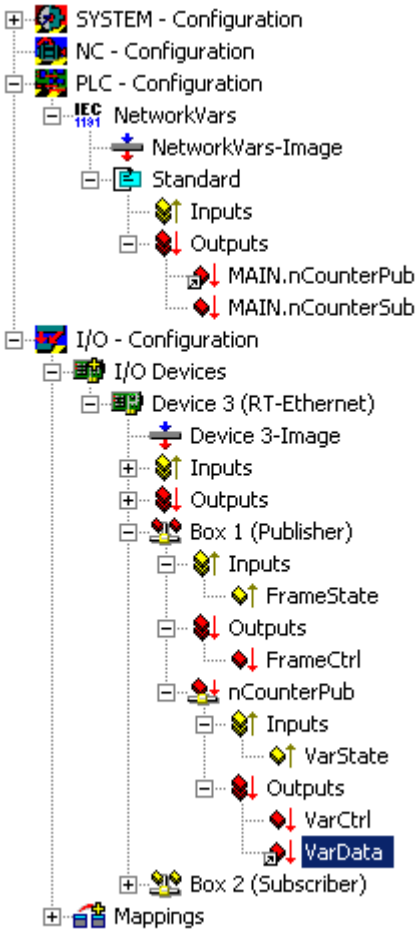


Fig. 67: Mapping of the network variable with an output variable of the PLC

4.3.3 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.

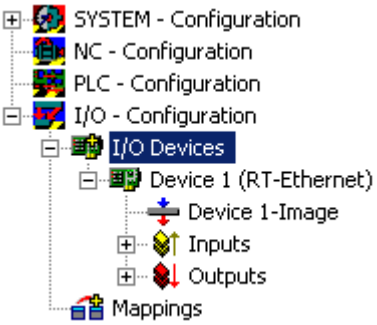


Fig. 68: Insert a Subscriber box under the RT Ethernet device

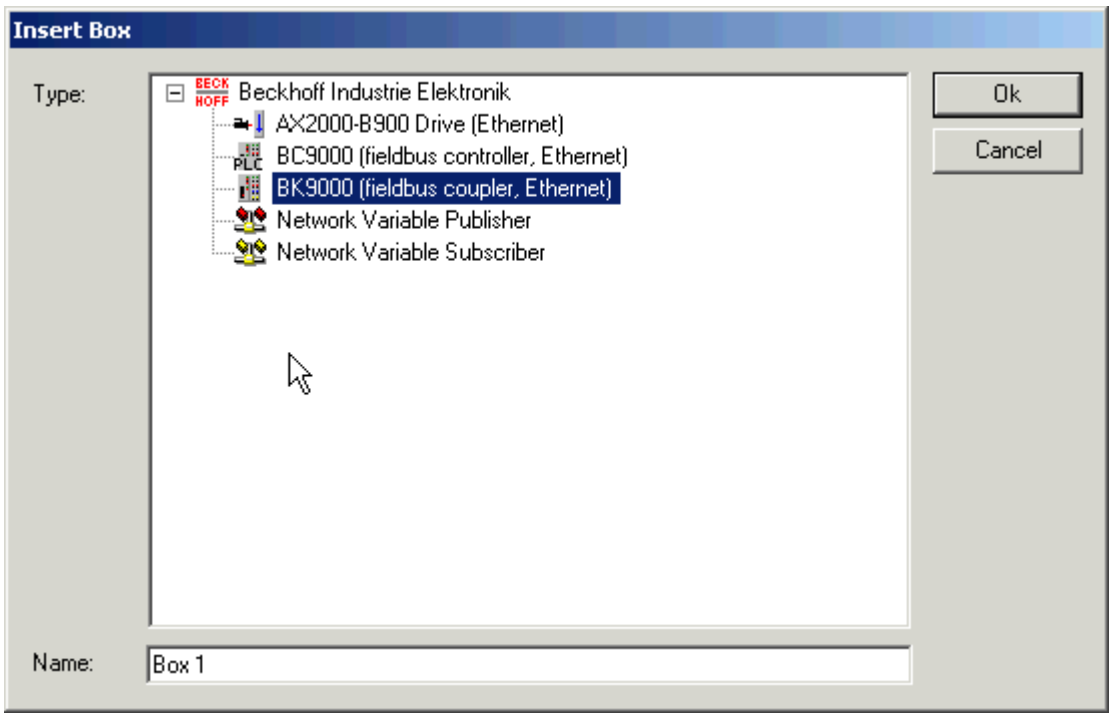


Fig. 69: "Insert Box" dialog - selecting and naming the Subscriber box (Box1)

A network variable is then created under the Subscriber box.

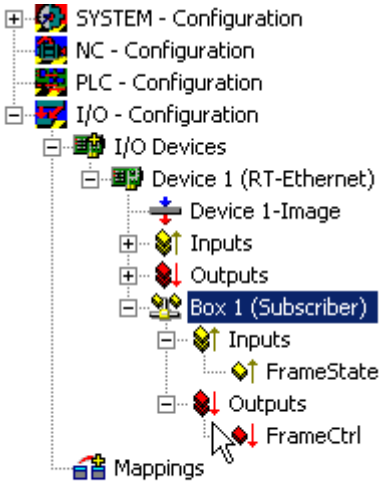


Fig. 70: Creating the network variable under the Subscriber box

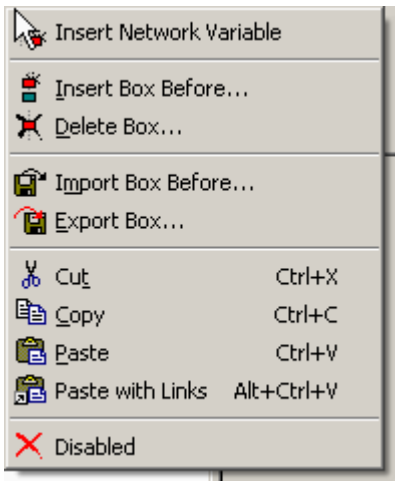


Fig. 71: Adding a network variable

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.

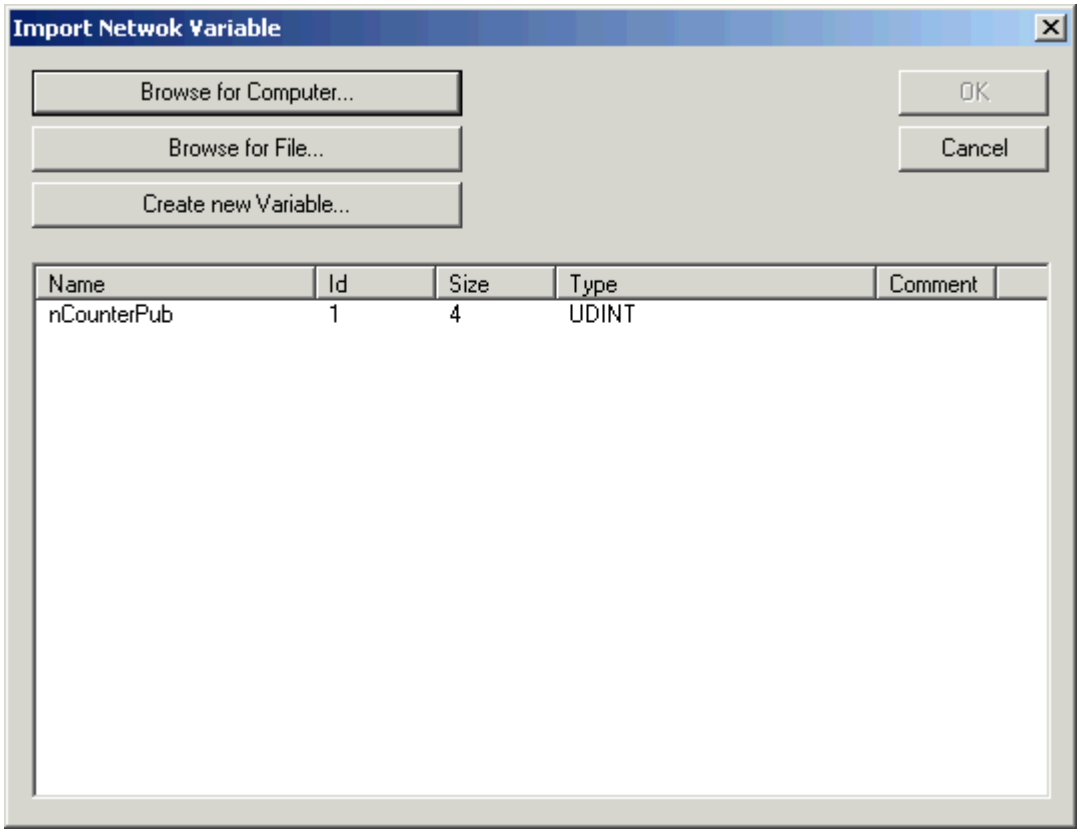


Fig. 72: "Import Network Variable" dialog - display of Publisher variables

Inputs and outputs were created under the added variable.

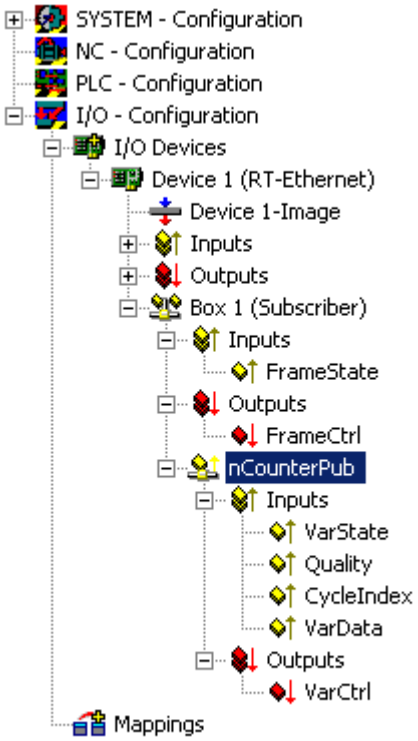


Fig. 73: Display of created inputs and outputs

The FrameState input and the FrameCtrl output under the box indicate 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 µs. The counter value indicates the variable delay. The example below shows the online value of Quality when the network connector is unplugged (counter increases) and reconnected (counter value 0).

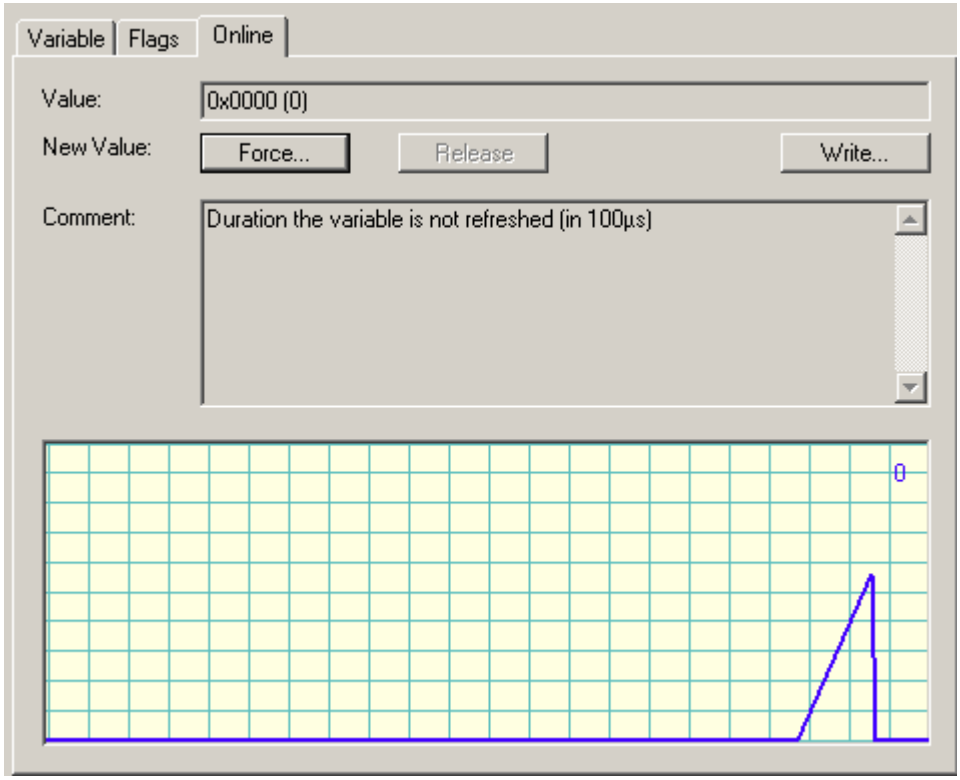


Fig. 74: Online value of Quality after unplugging and reconnecting of the network connector.

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

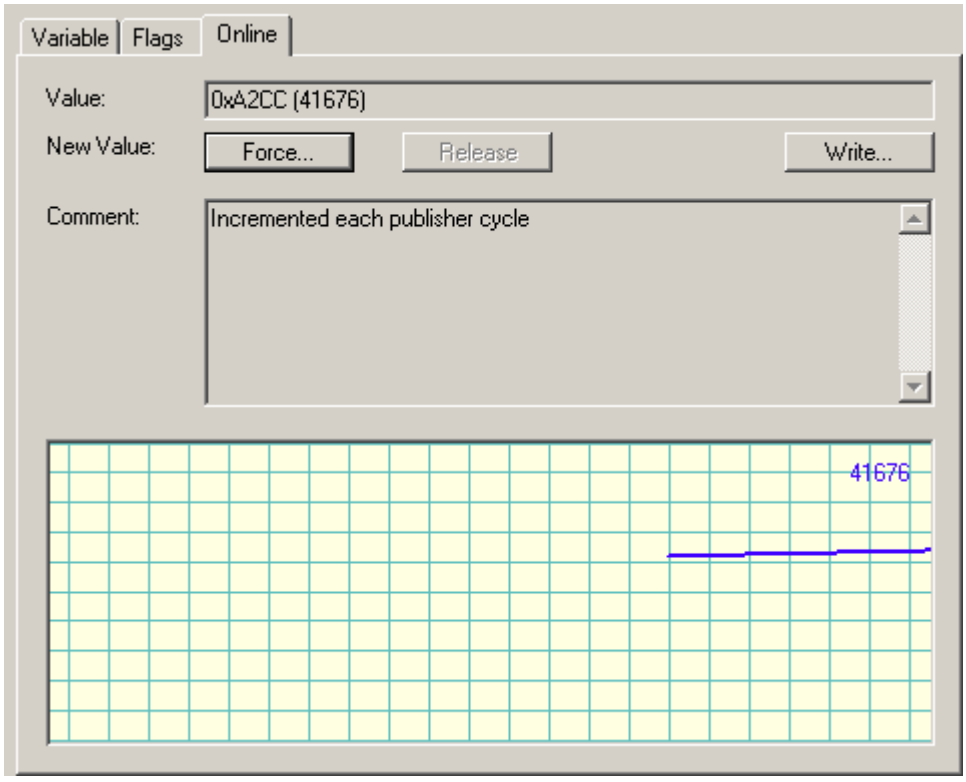


Fig. 75: Incrementing of the CycleIndex variables

4.3.4 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 example, 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 [▶ 62] 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 switched mode (MAC addressing) communication cycles of approx. 10 ms and below can be achieved, in routed mode (IP addressing) a few 100 ms may be achievable in some cases.

Diagnostic variable "quality"

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"

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 regarding 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.

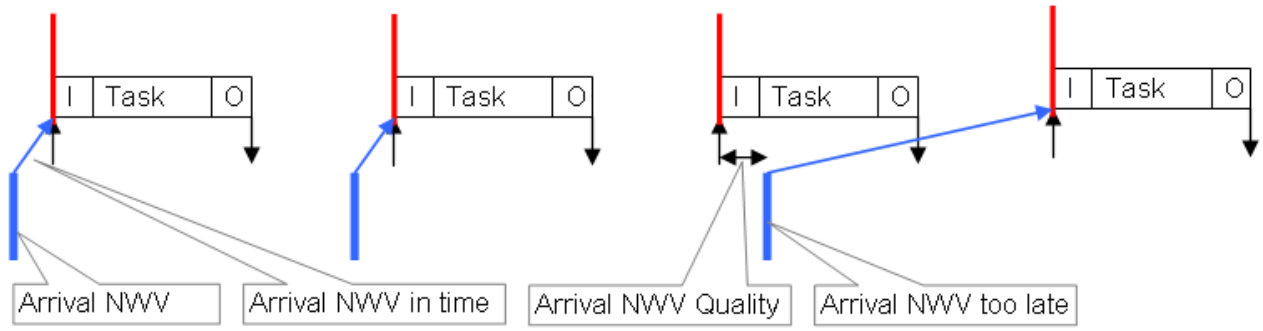


Fig. 76: Interrelationship between quality and delayed network variable

- **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. *Example 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

In order to achieve unambiguous allocation we recommend using different IDs for each data communication between connected PCs.

Reason: In Fig. *Example 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.

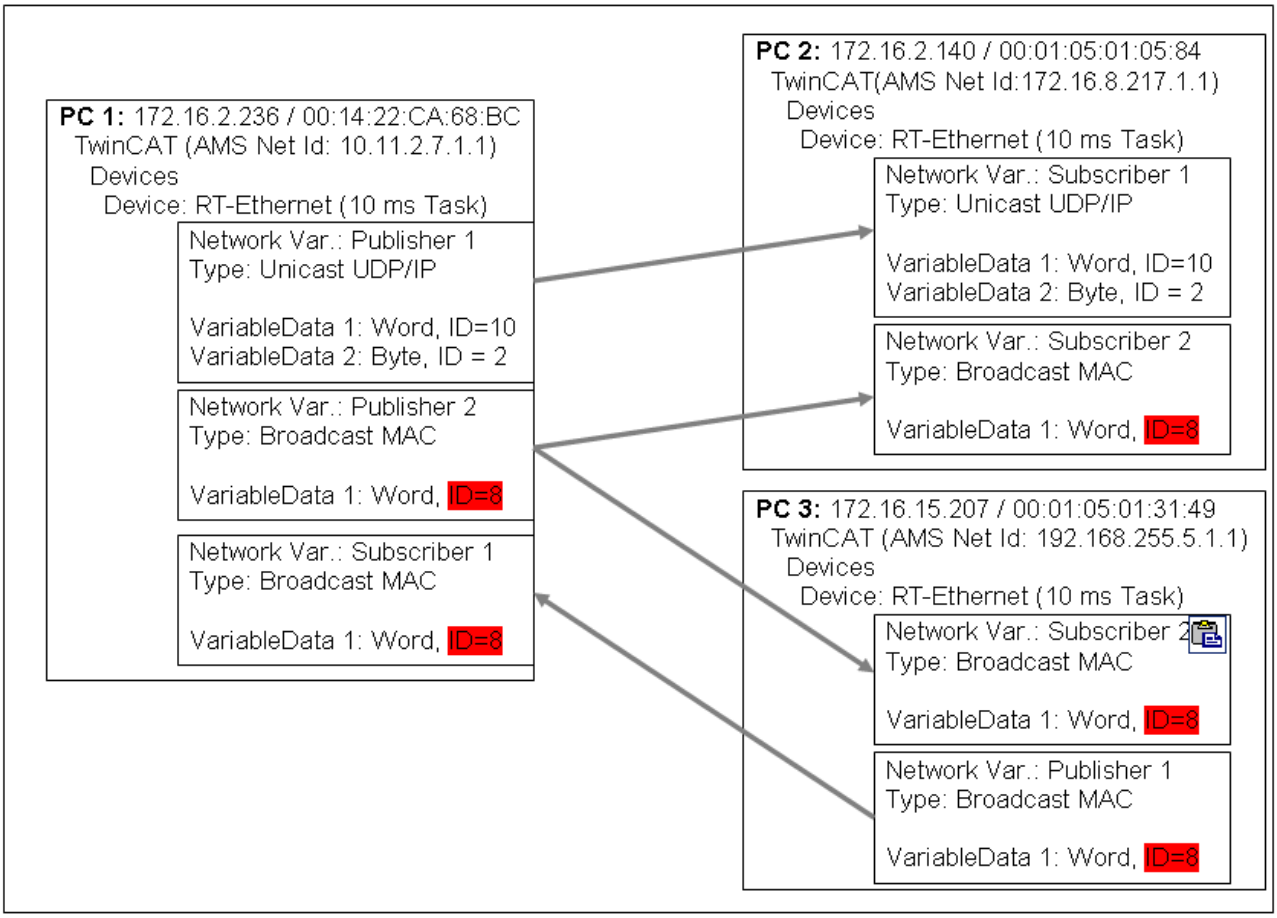


Fig. 77: Example for communication via network variables

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.
In Fig. *Example for different size and structure in PC and CX*, 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.

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.

Use of Bus Terminal Controllers (BCxxxx, 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 example, can be used for signed values.

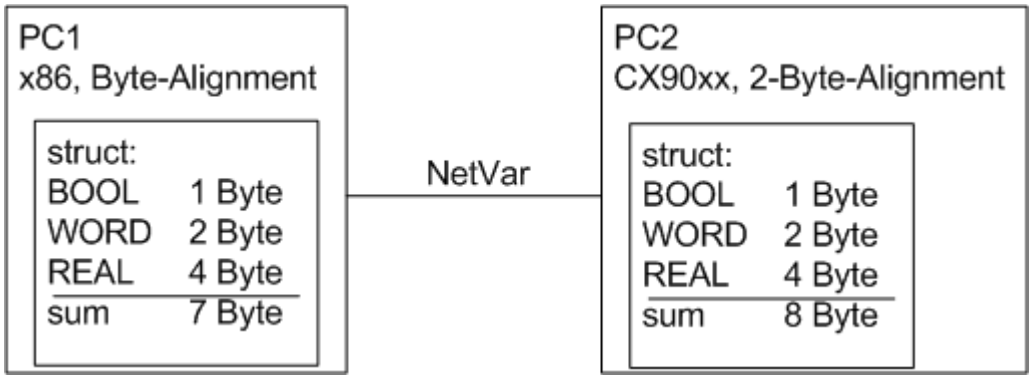


Fig. 78: Example for different size and structure in PC and CX

Settings in the System Manager

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 [▶ 59]

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

Publisher, Box

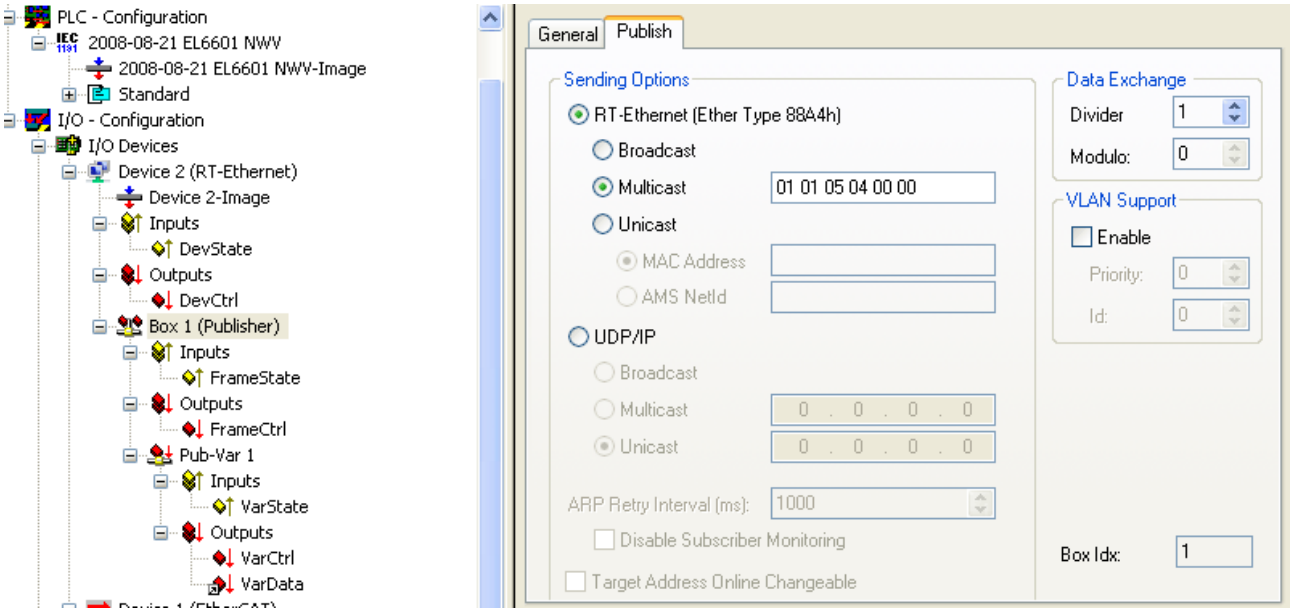


Fig. 79: Publisher RT Ethernet settings

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

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 [▶ 60], 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).

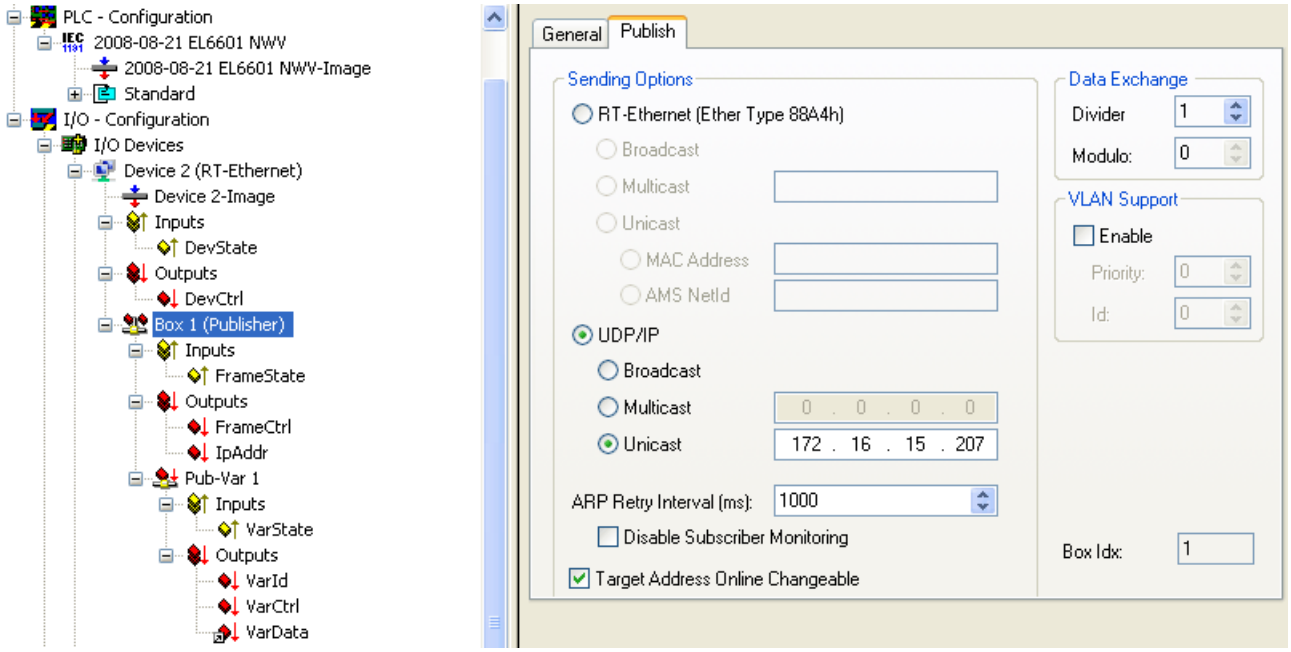


Fig. 80: Publisher settings - UDP/IP

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

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

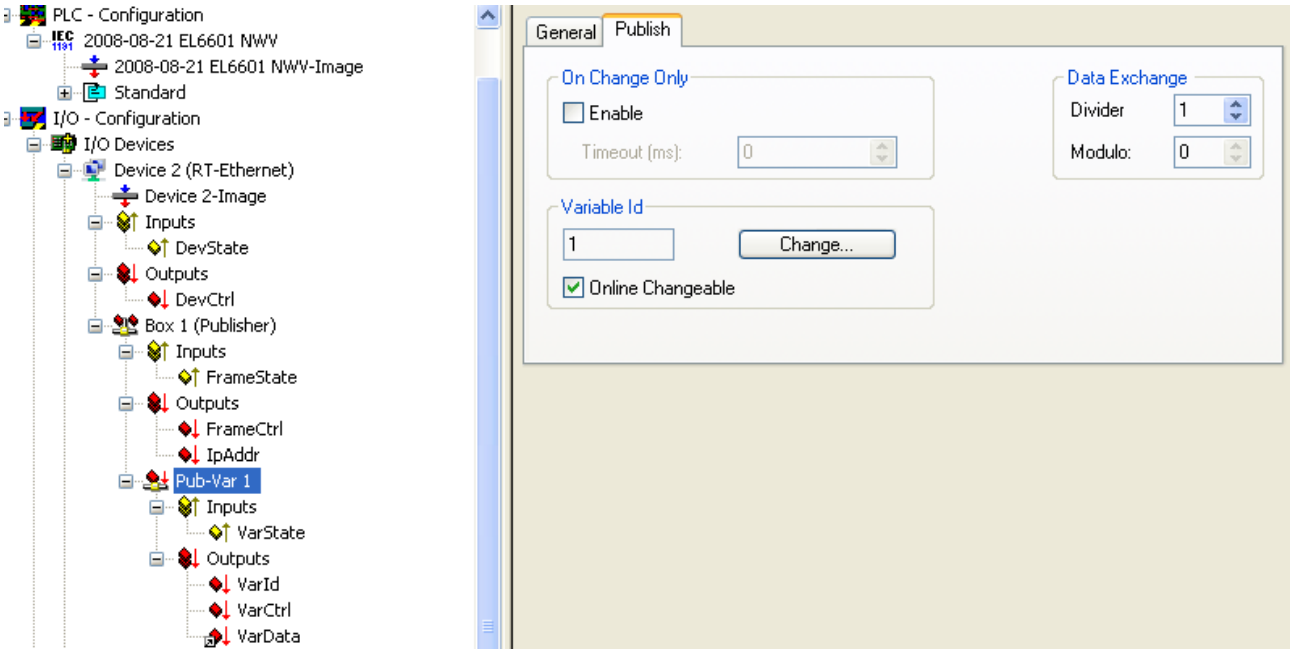


Fig. 81: Publisher variable settings

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

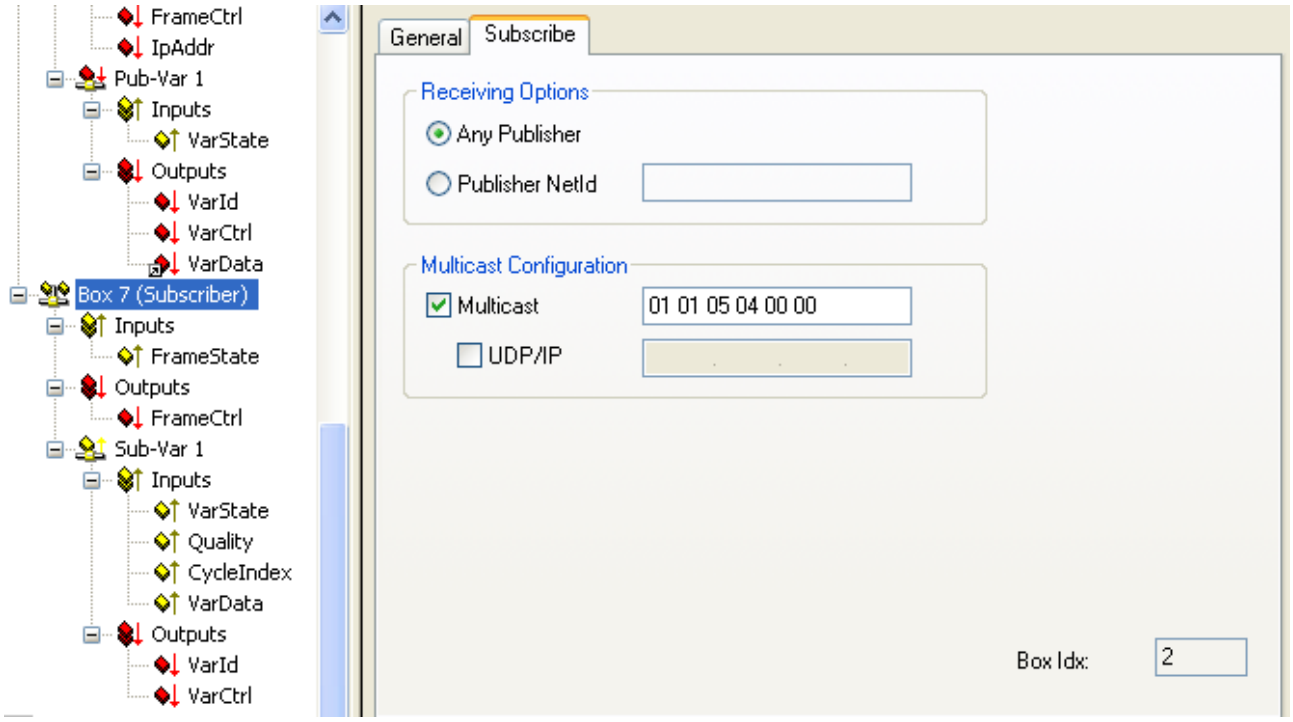


Fig. 82: Subscriber settings

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

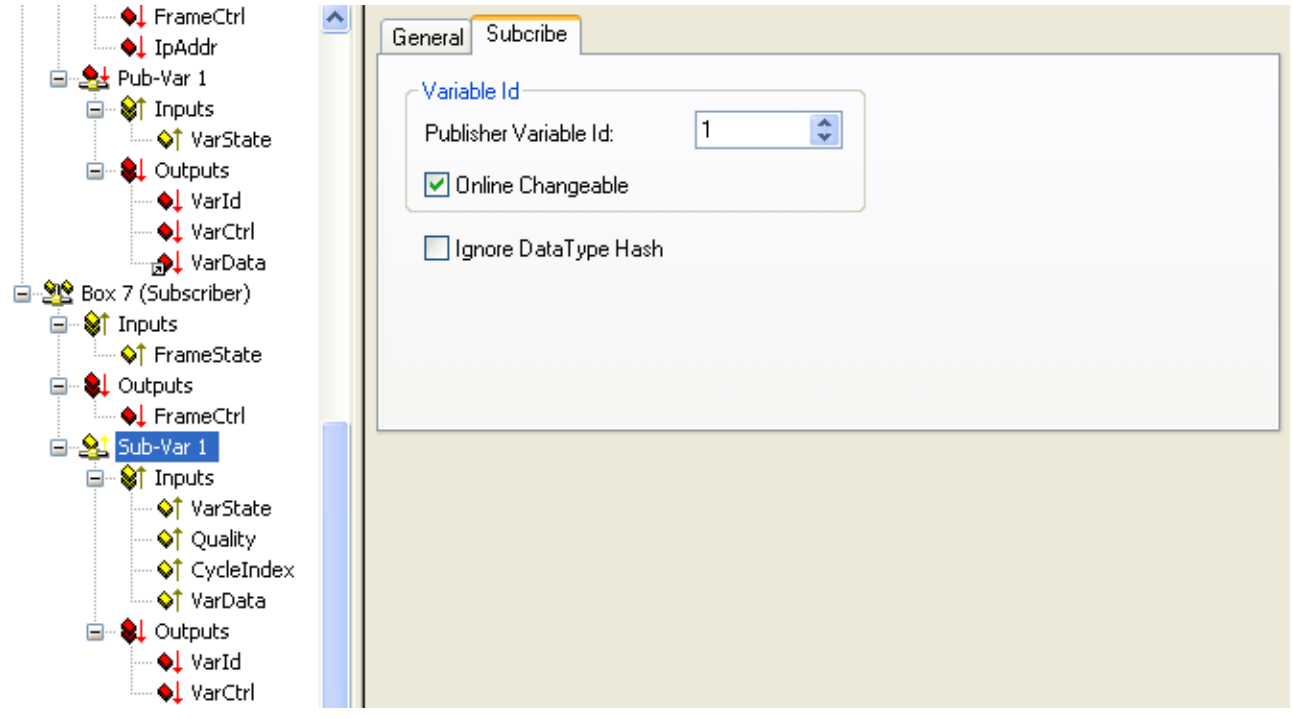


Fig. 83: Subscriber variable settings

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

5 Appendix

5.1 ESL Protocol

The EtherCAT Switch Link protocol (ESL) from Beckhoff is a simple extension for the control of EtherCAT infrastructure components. It is currently used (as of 2010) for the operation of the ET2000 Industrial Ethernet Multichannel Probe and the CU2508.

The ESL data consist of 16 bytes of control data and information, which are added as a prefix (in front) or as a postfix (to the end) of an existing Ethernet data frame. The check sum (CRC) is adapted accordingly.

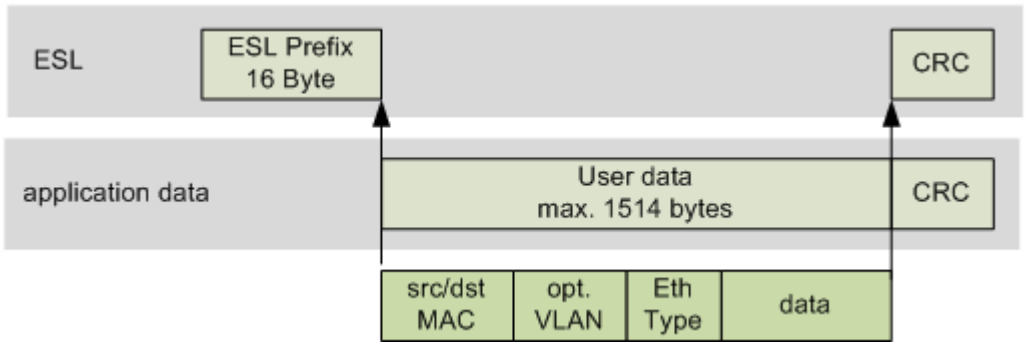


Fig. 84: CU2508 ESL prefix

The ESL data are structured as follows:

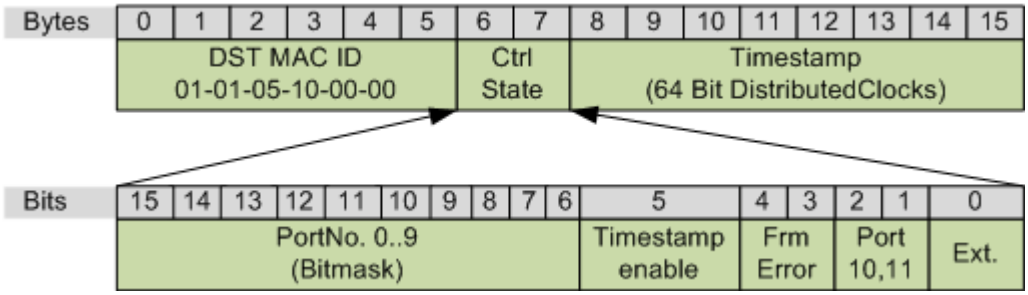


Fig. 85: ESL structure

Ethernet network monitors

The freely available network monitor Wireshark has contained an interpreter for the ESL protocol since the version 1.4.2.

5.2 Firmware Update

The CU2508 has 2 different firmware versions, one for the FPGA and one for the controller.

The CU2508 must be operated in a special configuration for the determination of the current firmware version and for updating. It must therefore be removed from the application and is to be operated as an individual EtherCAT device on a TwinCAT system.

The following steps must be carried out in order to update:

1. TwinCAT

On the PC system used for programming, TwinCAT must have at least version 2.11R2 build 2021 or TwinCAT 3.1.

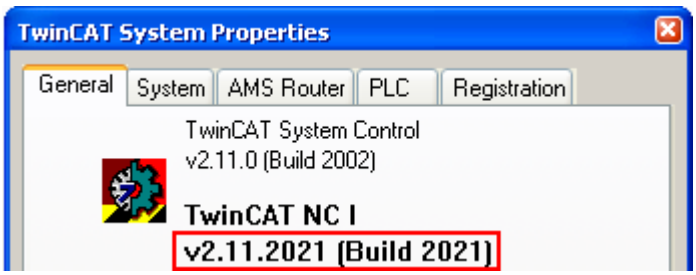


Fig. 86: TwinCAT Version

2. ESI/XML description

The current CU2508 ESI/XML data must be present on the system; they can be downloaded from the download area on the Beckhoff web page and are to be saved under …"\ TwinCAT\Io\EtherCAT\".

3. Installation of the CU2508

Connect the CU2508 with its uplink to an Ethernet port on the master PC and connect the power supply. The port must be set up as a real-time port with the Beckhoff TwinCAT driver. The downports X1..8 must not be connected.



Fig. 87: CU2508 structure

4. Creating the TwinCAT configuration

The CU2508 must now be added manually as an EtherCAT device. To do this, set up an EtherCAT device and specify the adapter to be used.

Instruction TwinCAT 2:

Open a new System Manager window. This causes the ESI data updated above to be read in again.

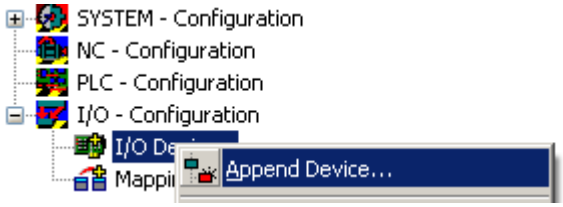


Fig. 88: Right mouse click, Append device

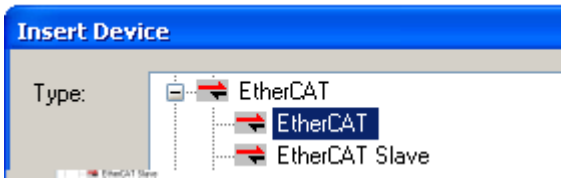


Fig. 89: Selection of the device "EtherCAT"

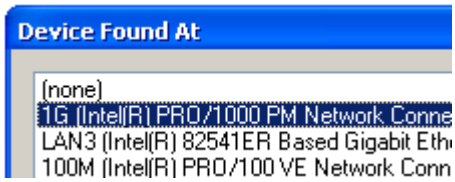


Fig. 90: Selection of the Ethernet adapter

If the adapter you want is not visible, then the Beckhoff real-time driver is not yet installed on this adapter.

Now append the CU2508.

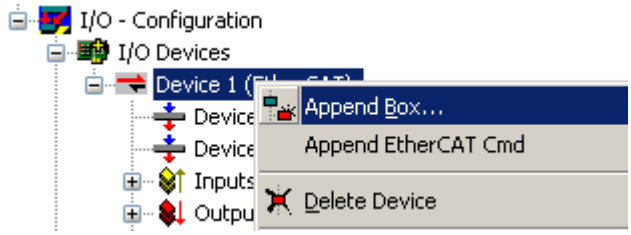


Fig. 91: Right mouse click, Append device

If the device name is entered in the search mask (A) the overview is reduced in size. Activate the check boxes "ExtendedInformation" and "ShowHiddenDevices". As a result of this, a previously hidden element, the CU2508 configuration port, becomes visible (in this case: CU2508-0000-0016).

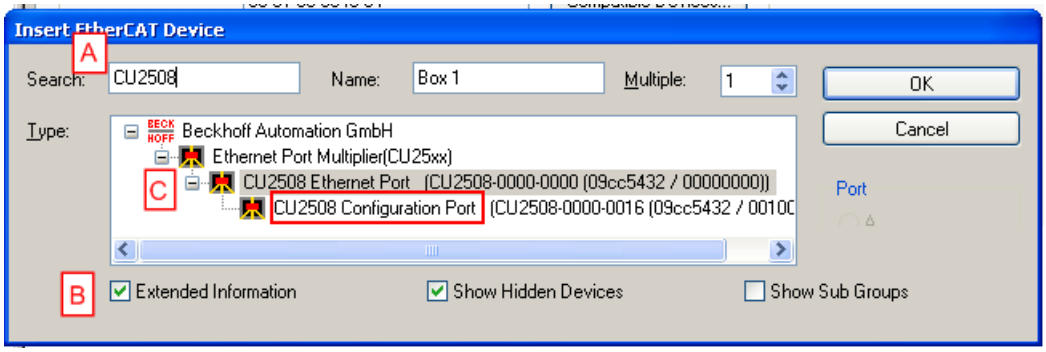


Fig. 92: Selection of the CU2508-0000 configuration port

Select this device. The configuration now consists of the “EtherCAT Device” and the single slave CU2508.

Now ensure that the function is correct by restarting TwinCAT (B) in ConfigMode and carrying out ReloadIoDevices (press key F4). If the EtherCAT device is selected (A), the current State OP (D) and Traffic utilization (E) are displayed in the *Online* tab (C). Approx. 250 CyclicFrames per second are usual in the ConfigMode with a 4 ms cycle time; no lost frames or TXRXErrors may occur.

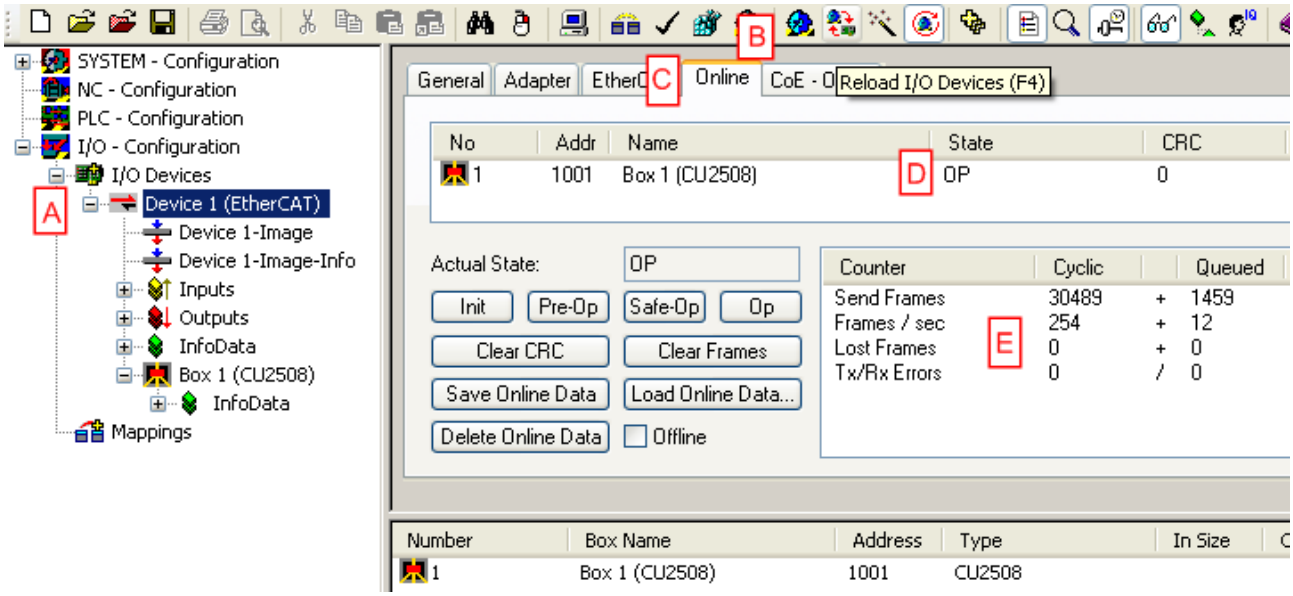


Fig. 93: Online configuration

Instruction TwinCAT 3:

The procedure is the same as for TwinCAT 2: add the "CU2508 Configuration Port" to the selected EtherCAT adapter.

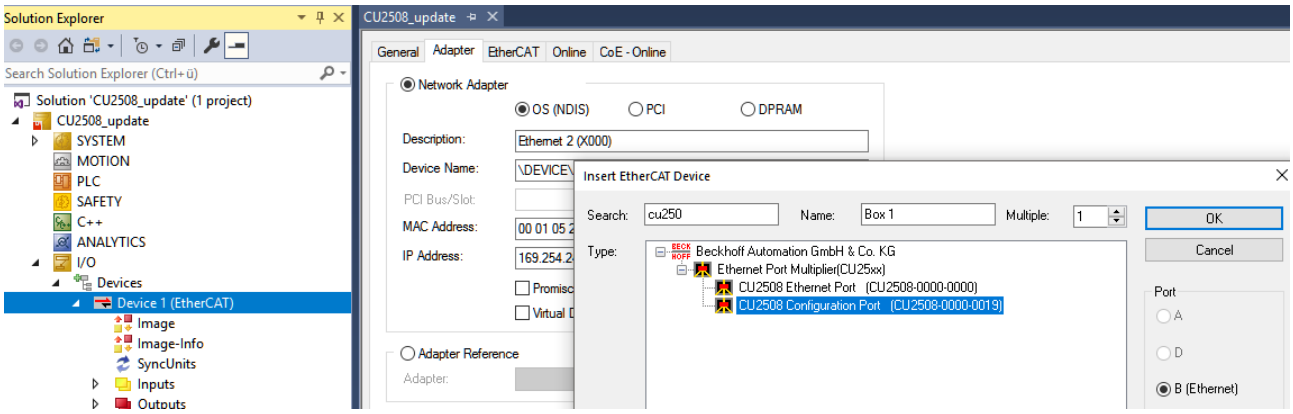


Fig. 94: Append Configuration Port

Then check that it is operating correctly:

- No Lost frames
- State:OP

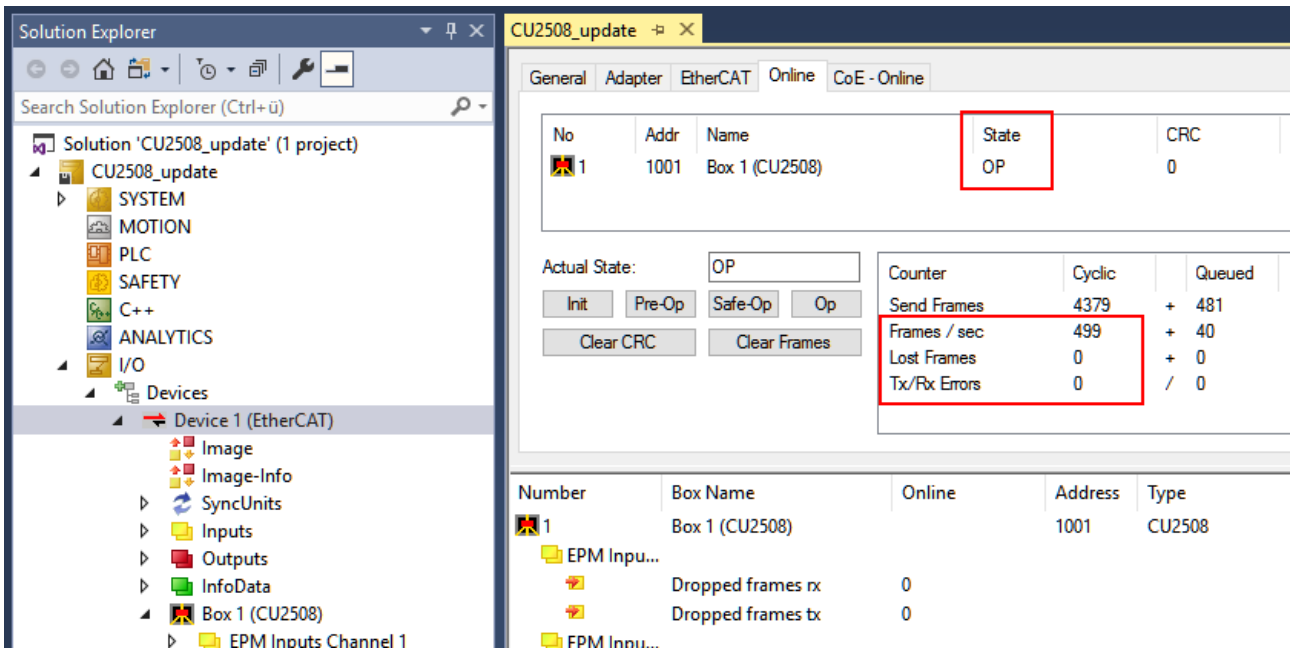


Fig. 95: Check for lost frames and State: OP

5. Reading out the current ESI/FW version

On the basis of fig. *Online configuration*, EtherCAT ESI revision, controller and FPGA-FW can now be checked.

ESI Revision

When the system is running in ConfigMode, the revision found can be displayed by scan:

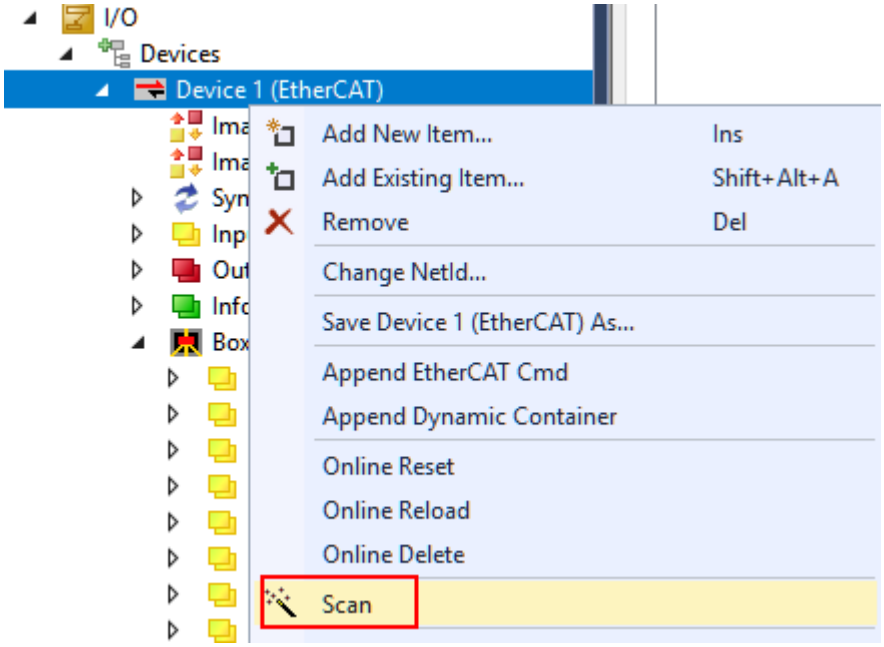


Fig. 96: Display revision after scan

If the connected CU2508 carries a different revision, the following is displayed:

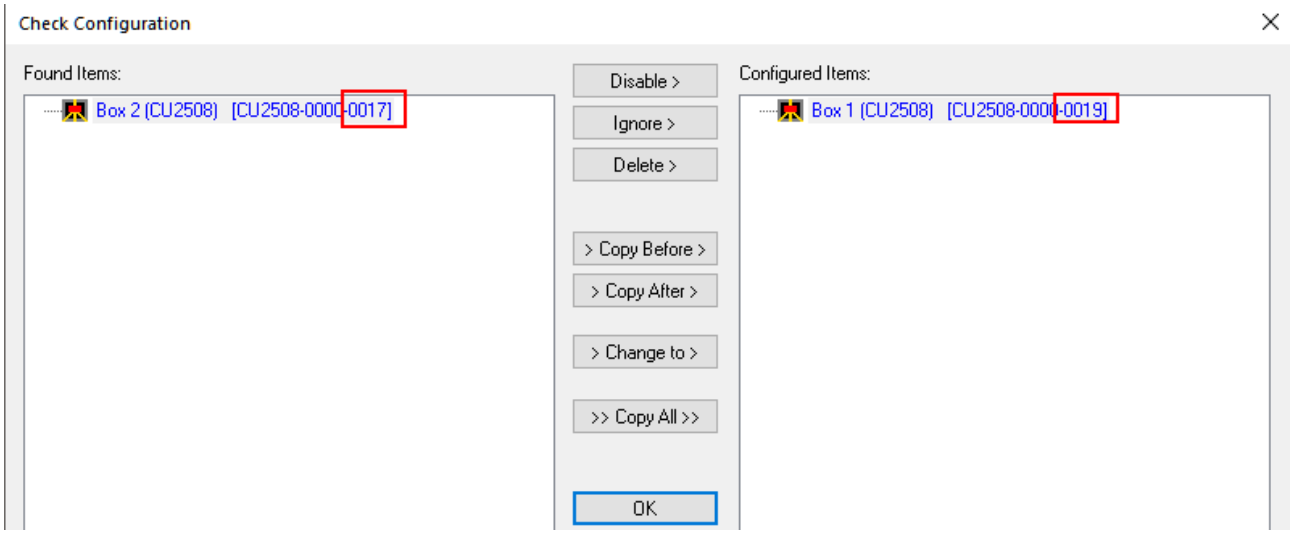


Fig. 97: Display different revision

Controller firmware (*.efw)

Select device CU2508 (A); the online CoE directory of the device is displayed in the *CoE-Online* tab (B); the firmware version (in this case 01) is displayed in the index 0x100A (D). Note that in (C) *"OnlineData" (green)* is displayed.

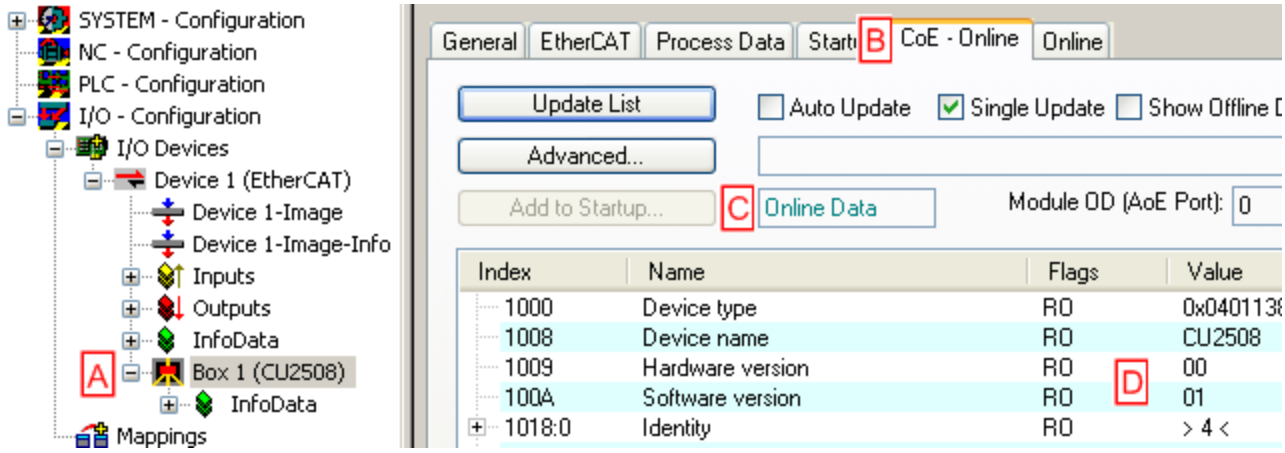


Fig. 98: Reading out the controller firmware version

FPGA firmware (*.rbf)

Select device CU2508 (A); press *Advanced Settings* ... (C) in the *EtherCAT* tab (B). There, select *Memory* (D) and read out the data starting from *StartOffset* 0E00 (E).
If the function is correct, the firmware version (here: 12) can be read at position x0E00 (F) and 2508dec at position x0E06.

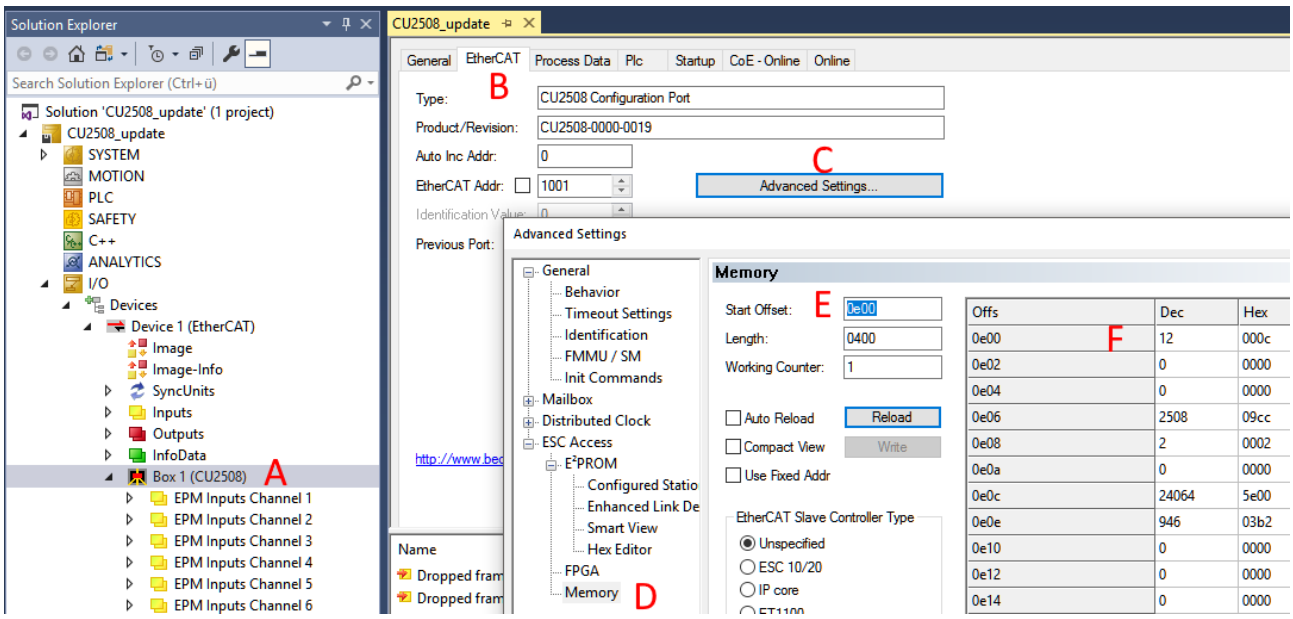


Fig. 99: Reading FPGA firmware

6. Firmware and ESI Update

On the basis of fig. *Online configuration*, if communication is working perfectly, an update of ESI, controller and FPGA firmware can be performed by right-clicking on the CU2508 device.

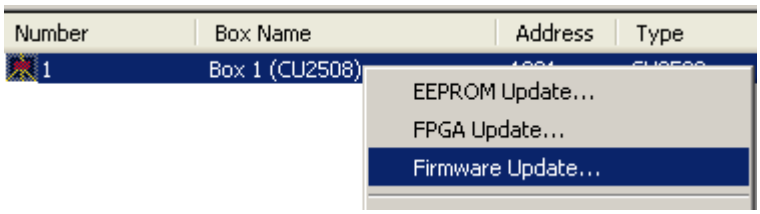


Fig. 100: Update by right-clicking

ESI/XML- update

The ESI *.xml can also be overwritten by "EEPROM Update". Then select and program the corresponding XML revision of the "CU2508 Configuration Port".

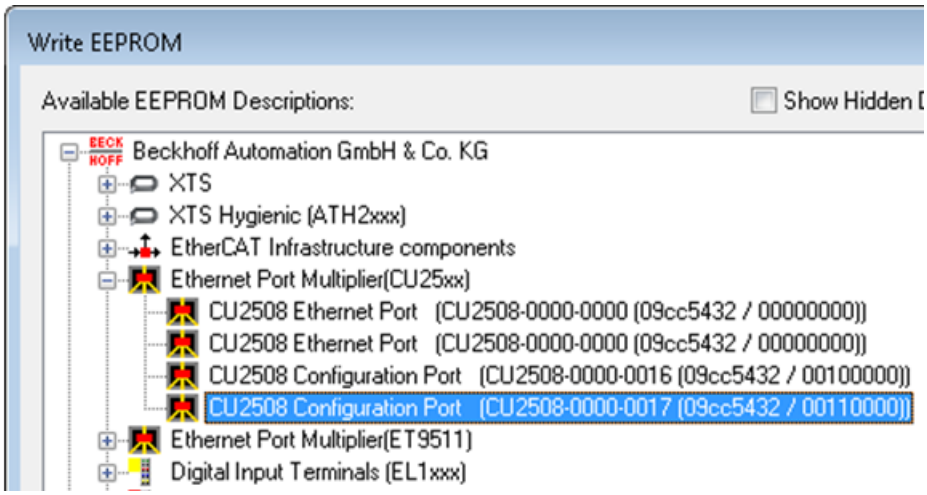


Fig. 101: Select XML revision

Display of the revision CU2508 slaves

Only the ConfigurationPort of the CU2508 is described with an updated ESI, if necessary. This is used to set some of the CU2508's properties. The CU2508 slaves running in regular EtherCAT operation always report with revision 0000 and cannot be programmed separately.

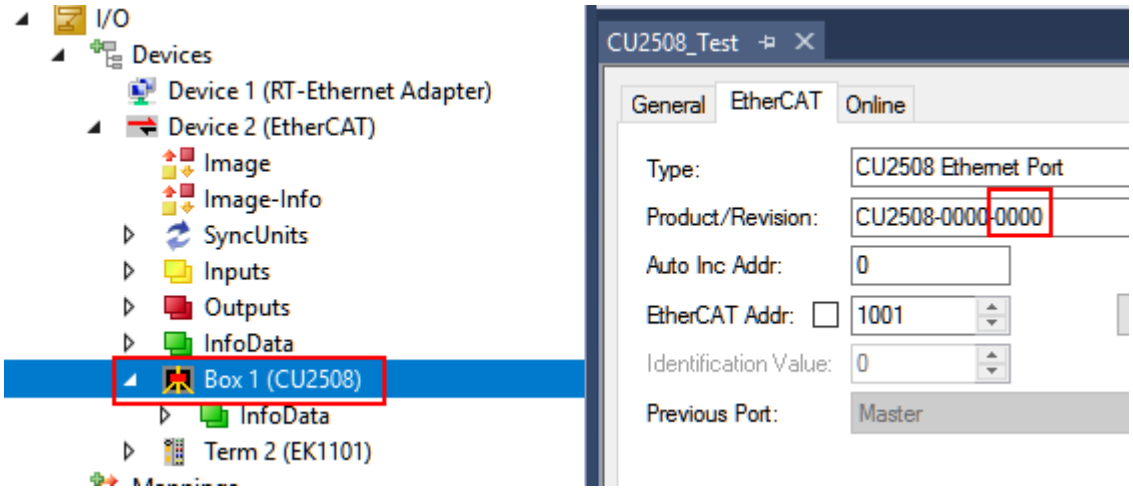


Fig. 102: Display of the revision -0000

FW Update

The firmware is ideally downloaded to the CU2508 in bootstrap mode.

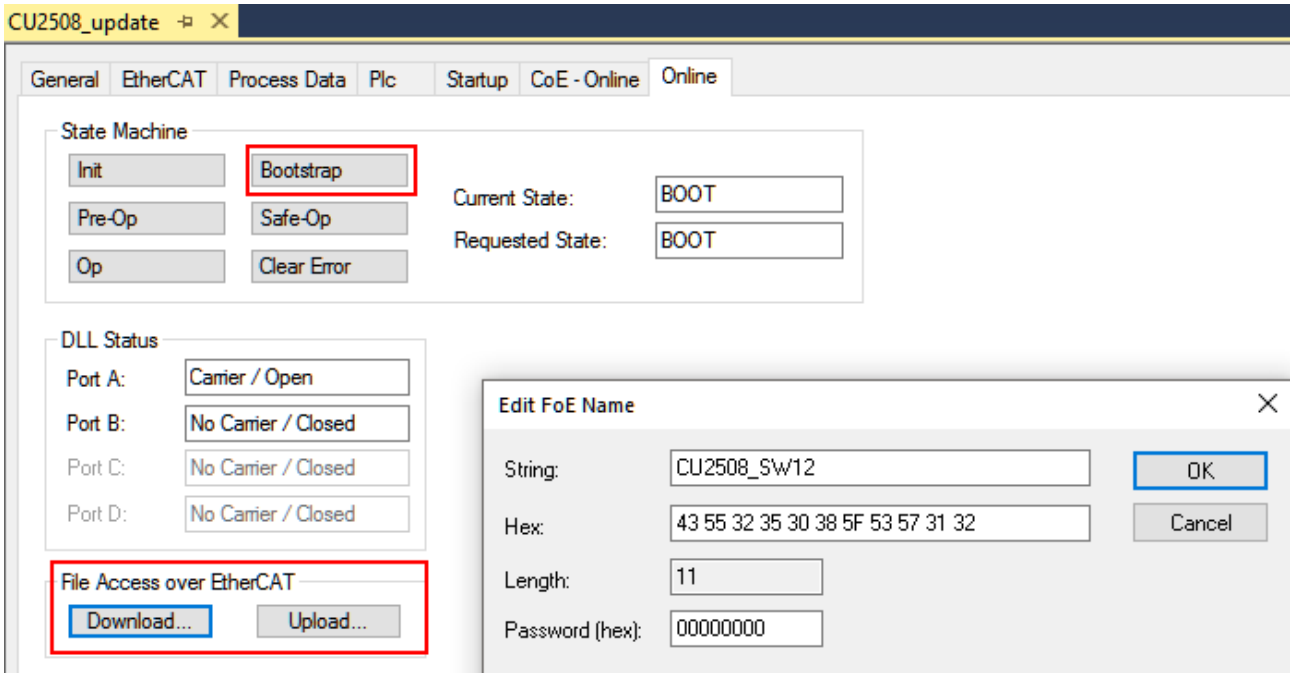


Fig. 103: Update in Bootstrap Mode

FPGA update

The FPGA firmware is loaded by right clicking, this can take a few seconds.

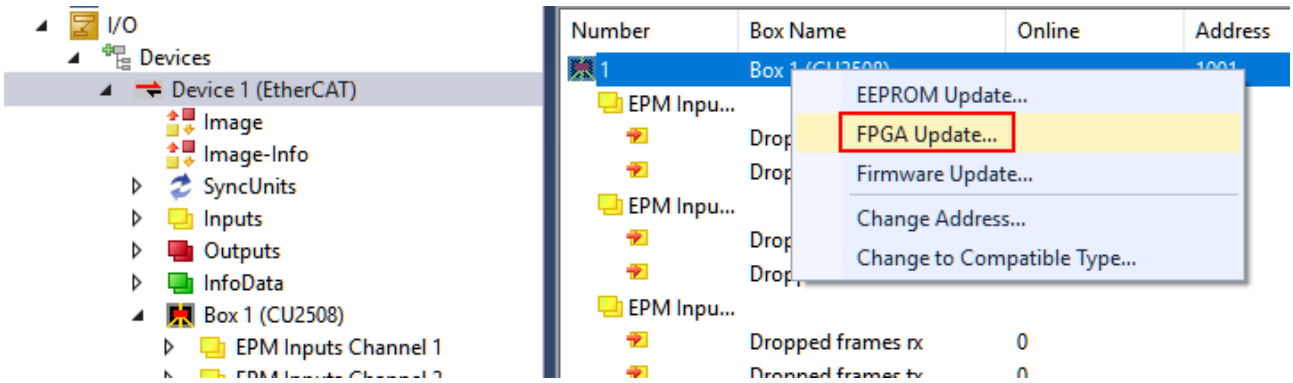


Fig. 104: Update FPGA firmware

A TwinCAT3 message window then appears:

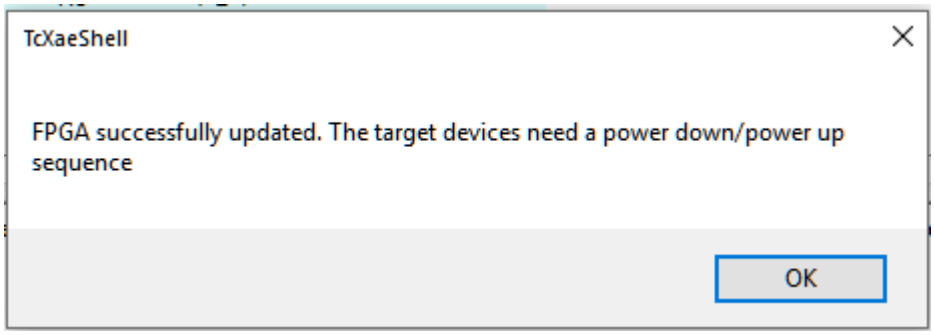


Fig. 105: Message window FPGA Update

The CU2508 must then be briefly switched to the de-energized state.

5.3 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 [▶ 67].
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!

CU2508			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date**
00 - 11*	01	CU2508-0000-0016	2011/01
	02	CU2508-0000-0017	2011/03
	03		2012/05
	04		2013/01
	05		2013/01
	06		2014/10
	07	CU2508-0000-0018	2015/08
	08		2016/02
	09		2016/06
	10		2017/05
	11		2019/08
	12	CU2508-0000-0019	2019/10
	13		2021/07
	14		2022/10
	15		2022/10
	16*		2024/05

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

5.4 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

Trademark statements

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

Third-party trademark statements

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

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

Wireshark is a registered trademark of Sysdig, Inc.

More Information:
www.beckhoff.com/CU2508

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