YASKAWA

VIPA System SLIO

CPU | 015-CEFNR00 | Manual

HB300 | CPU | 015-CEFNR00 | en | 17-24

SPEED7 CPU 015N

www.vipa.com/en/service-support/manuals

VIPA CONTROLS

VIPA GmbH
Ohmstr. 4
91074 Herzogenaurach
Telephone: 09132-744-0
Fax: 09132-744-1864
Email: info@vipa.com
Internet: www.vipa.com

015-CEFNR00_000_CPU 015N,4,EN - © 2017

Table of contents

1 General

1.1 Copyright © VIPA GmbH

All Rights Reserved

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

EMail: info@vipa.de

http://www.vipa.com

Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. Nevertheless, the authors retain the right to modify the information.

This customer document describes all the hardware units and functions known at the present time. Descriptions may be included for units which are not present at the customer site. The exact scope of delivery is described in the respective purchase contract.

CE Conformity Declaration

Hereby, VIPA GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions. Conformity is indicated by the CE marking affixed to the product.

Conformity Information

For more information regarding CE marking and Declaration of Conformity (DoC), please contact your local VIPA customer service organization.

Trademarks

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S and Commander Compact are registered trademarks of VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 is a registered trademark of profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300 and S7-400 are registered trademarks of Siemens AG.

Microsoft and Windows are registered trademarks of Microsoft Inc., USA.

Portable Document Format (PDF) and Postscript are registered trademarks of Adobe Systems, Inc.

All other trademarks, logos and service or product marks specified herein are owned by their respective companies.

Information product support

Contact your local VIPA Customer Service Organization representative if you wish to report errors or questions regarding the contents of this document. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany
Telefax: +49 9132 744-1204
EMail: documentation@vipa.de

Technical support

Contact your local VIPA Customer Service Organization representative if you encounter problems with the product or have questions regarding the product. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany
Tel.: +49 9132 744-1150 (Hotline)
EMail: support@vipa.de

1.2 About this manual

Objective and contents

This manual describes the CPU 015-CEFNR00 of the System SLIO from VIPA. It contains a description of the construction, project implementation and usage.

Product	Order no.	as of state:		
		CPU HW	CPU FW	CP FW
CPU 015N	015-CEFNR00	02	V1.5.9	V3.1.3

Target audience

The manual is targeted at users who have a background in automation technology.

Structure of the manual

The manual consists of chapters. Every chapter provides a self-contained description of a specific topic.

Guide to the document

The following guides are available in the manual:

- An overall table of contents at the beginning of the manual
- References with page numbers

Availability

The manual is available in:

- printed form, on paper
- in electronic form as PDF-file (Adobe Acrobat Reader)

Icons Headings

Important passages in the text are highlighted by following icons and headings:

DANGER!
Immediate or likely danger. Personal injury is possible.

CAUTION!
Damages to property is likely if these warnings are not heeded.

Supplementary information and useful tips.

1.3 Safety information

Applications conforming with specifications

The system is constructed and produced for:

- communication and process control
- general control and automation tasks
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle

DANGER!
This device is not certified for applications in

– in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

CAUTION!
The following conditions must be met before using or commissioning the components described in this manual:

– Hardware modifications to the process control system should only be carried out when the system has been disconnected from power!
– Installation and hardware modifications only by properly trained personnel.
– The national rules and regulations of the respective country must be satisfied (installation, safety, EMC ...)

Disposal

National rules and regulations apply to the disposal of the unit!

2 Basics and mounting

2.1 Safety information for users

Handling of electrostatic sensitive modules

VIPA modules make use of highly integrated components in MOS-Technology. These components are extremely sensitive to over-voltages that can occur during electrostatic discharges. The following symbol is attached to modules that can be destroyed by electrostatic discharges.

The Symbol is located on the module, the module rack or on packing material and it indicates the presence of electrostatic sensitive equipment. It is possible that electrostatic sensitive equipment is destroyed by energies and voltages that are far less than the human threshold of perception. These voltages can occur where persons do not discharge themselves before handling electrostatic sensitive modules and they can damage components thereby, causing the module to become inoperable or unusable. Modules that have been damaged by electrostatic discharges can fail after a temperature change, mechanical shock or changes in the electrical load. Only the consequent implementation of protection devices and meticulous attention to the applicable rules and regulations for handling the respective equipment can prevent failures of electrostatic sensitive modules.

Shipping of modules

Modules must be shipped in the original packing material.

Measurements and alterations on electrostatic sensitive modules

When you are conducting measurements on electrostatic sensitive modules you should take the following precautions:

- Floating instruments must be discharged before use.
- Instruments must be grounded.

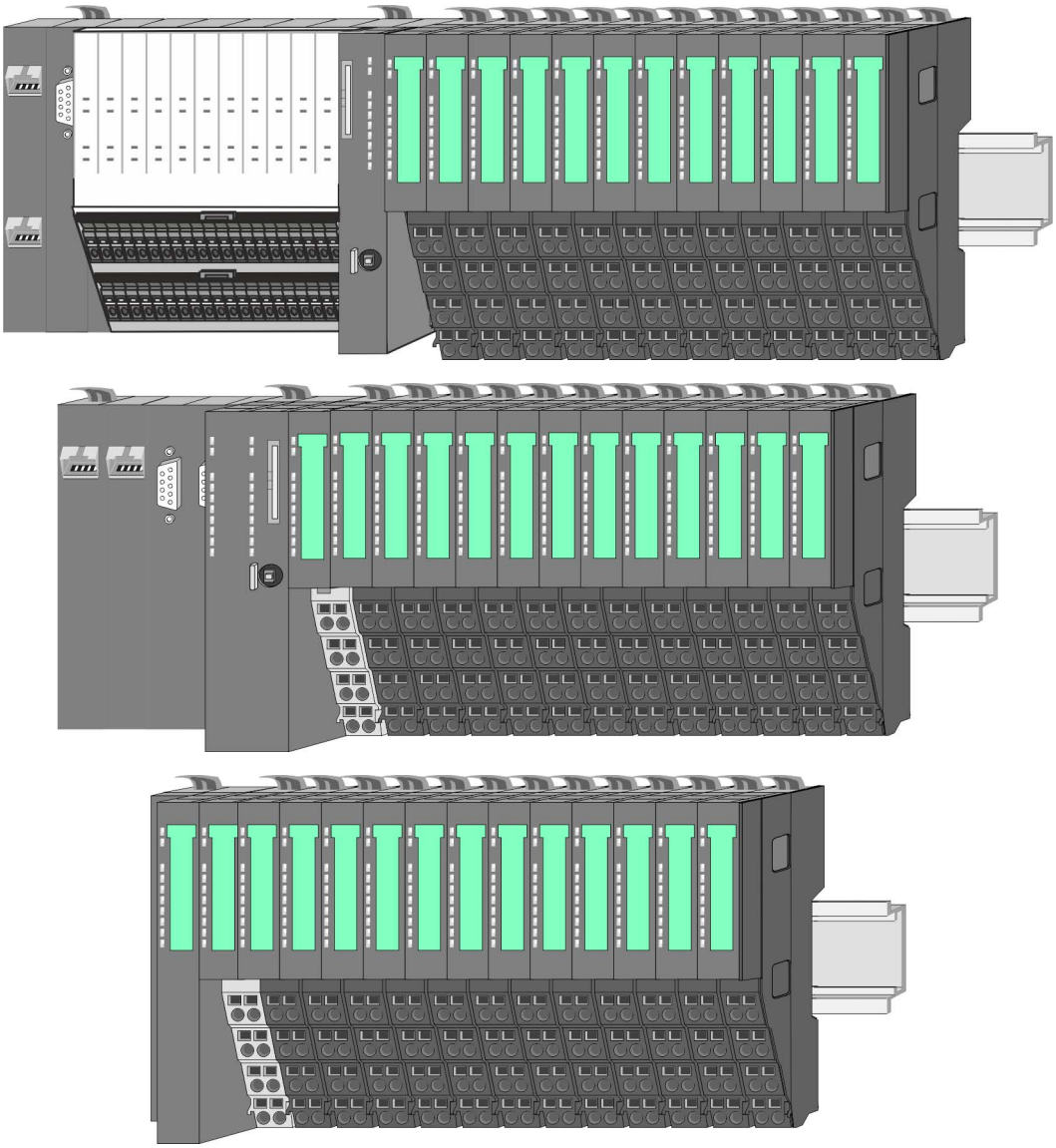
Modifying electrostatic sensitive modules you should only use soldering irons with grounded tips.

CAUTION!
Personnel and instruments should be grounded when working on electrostatic sensitive modules.

2.2 System conception

2.2.1 Overview

System SLIO is a modular automation system for assembly on a 35mm mounting rail. By means of the peripheral modules with 2, 4 or 8 channels this system may properly be adapted matching to your automation tasks. The wiring complexity is low, because the supply of the DC 24V power section is integrated to the backplane bus and defective modules may be replaced with standing wiring. By deployment of the power modules in contrasting colors within the system, further isolated areas may be defined for the DC 24V power section supply, respectively the electronic power supply may be extended with 2A.

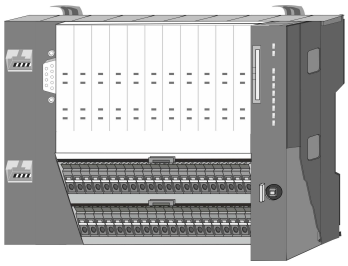


2.2.2 Components

- CPU (head module)
- Bus coupler (head module)
- Line extension
- Periphery modules
- Accessories

CAUTION!
Only modules of VIPA may be combined. A mixed operation with third-party modules is not allowed!

CPU 01xC



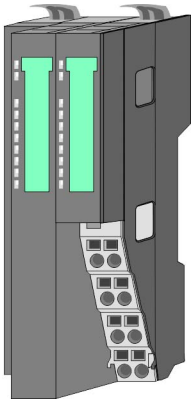
With this CPU 01xC, the CPU electronic, input/output components and power supply are integrated to one casing. In addition, up to 64 periphery modules of the System SLIO can be connected to the backplane bus. As head module via the integrated power supply CPU electronic and the I/O components are power supplied as well as the electronic of the connected periphery modules. To connect the power supply of the I/O components and for DC 24V power supply of via backplane bus connected peripheral modules, the CPU has removable connectors. By installing of up to 64 periphery modules at the backplane bus, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

CPU 01x

With this CPU 01x, the CPU electronic and power supply are integrated to one casing. As head module, via the integrated power module for power supply, CPU electronic and the electronic of the connected periphery modules are supplied. The DC 24 power section supply for the linked periphery modules is established via a further connection of the power module. By installing of up to 64 periphery modules at the backplane bus, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

CAUTION!
CPU part and power module may not be separated!
Here you may only exchange the electronic module!

Bus coupler

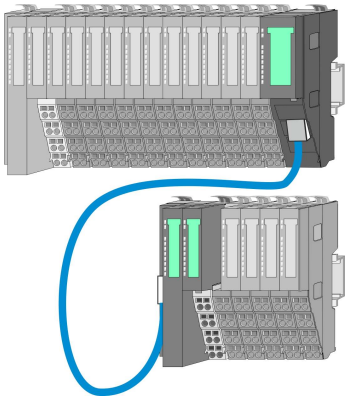


With a bus coupler bus interface and power module is integrated to one casing. With the bus interface you get access to a subordinated bus system. As head module, via the integrated power module for power supply, bus interface and the electronic of the connected periphery modules are supplied. The DC 24 power section supply for the linked periphery modules is established via a further connection of the power module. By installing of up to 64 periphery modules at the bus coupler, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

CAUTION!
Bus interface and power module may not be separated!

Here you may only exchange the electronic module!

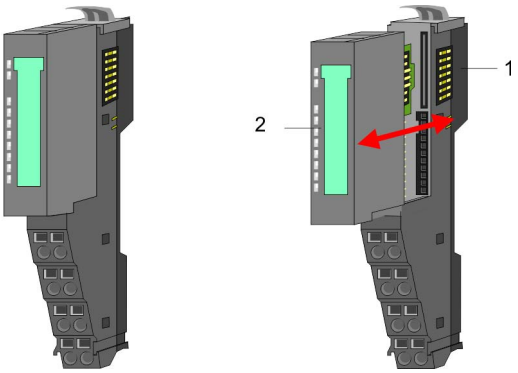
Line extension



In the System SLIO there is the possibility to place up to 64 modules in on line. By means of the line extension you can divide this line into several lines. Here you have to place a line extension master at each end of a line and the subsequent line has to start with a line extension slave. Master and slave are to be connected via a special connecting cable. In this way, you can divide a line on up to 5 lines. For each line extension the maximum number of pluggable modules at the System SLIO bus is decreased by 1. To use the line extension no special configuration is required.

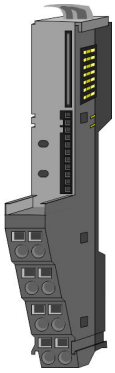
Periphery modules

Each periphery module consists of a *terminal* and an *electronic module*.



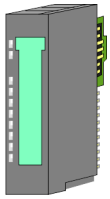
1 Terminal module
2 Electronic module

Terminal module



The *terminal* module serves to carry the electronic module, contains the backplane bus with power supply for the electronic, the DC 24V power section supply and the staircase-shaped terminal for wiring. Additionally the terminal module has a locking system for fixing at a mounting rail. By means of this locking system your SLIO system may be assembled outside of your switchgear cabinet to be later mounted there as whole system.

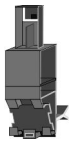
Electronic module



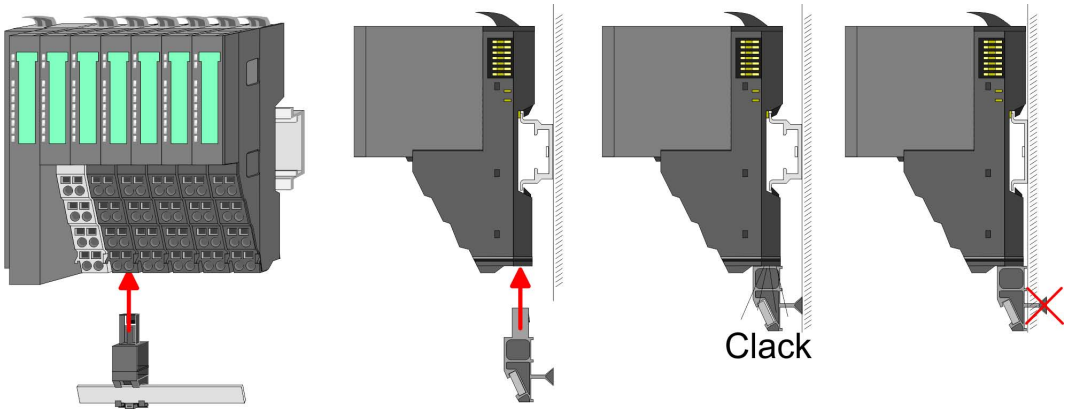
The functionality of a SLIO periphery module is defined by the *electronic* module, which is mounted to the terminal module by a sliding mechanism. With an error the defective module may be exchanged for a functional module with standing installation. At the front side there are LEDs for status indication. For simple wiring each module shows a corresponding connection diagram at the front and at the side.

2.2.3 Accessories

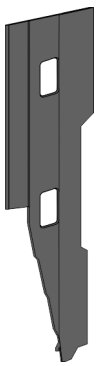
Shield bus carrier



The shield bus carrier (order no.: 000-0AB00) serves to carry the shield bus (10mm x 3mm) to connect cable shields. Shield bus carriers, shield bus and shield fixings are not in the scope of delivery. They are only available as accessories. The shield bus carrier is mounted underneath the terminal of the terminal module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.



Bus cover



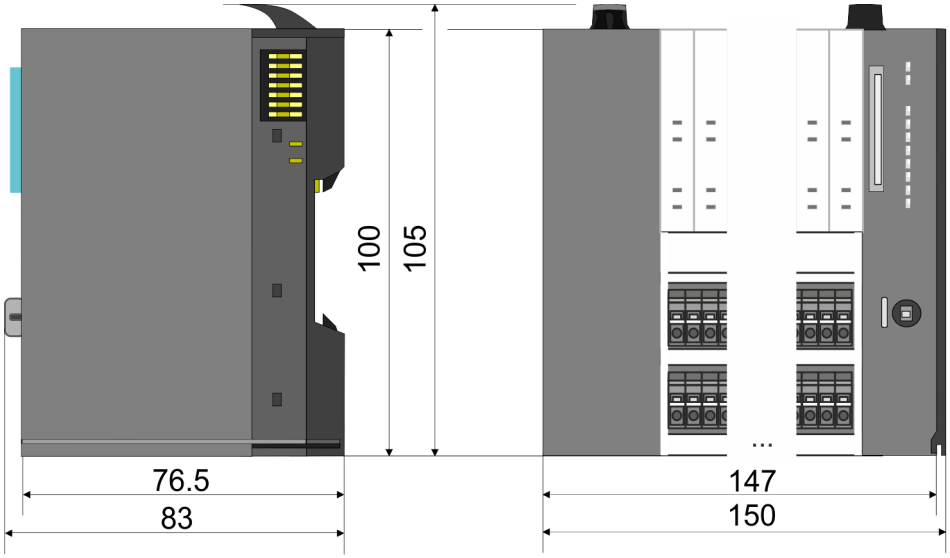
With each head module, to protect the backplane bus connectors, there is a mounted bus cover in the scope of delivery. You have to remove the bus cover of the head module before mounting a System SLIO module. For the protection of the backplane bus connector you always have to mount the bus cover at the last module of your system again. The bus cover has the order no. 000-0AA00.

Coding pins

There is the possibility to fix the assignment of electronic and terminal module. Here coding pins (order number 000-0AC00) from VIPA can be used. The coding pin consists of a coding jack and a coding plug. By combining electronic and terminal module with coding pin, the coding jack remains in the electronic module and the coding plug in the terminal module. This ensures that after replacing the electronics module just another electronic module can be plugged with the same encoding.

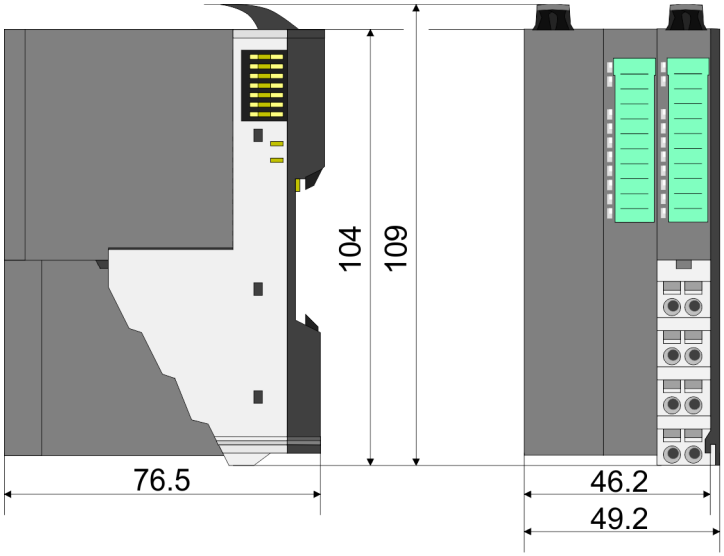
2.3 Dimensions

Dimensions CPU 01xC

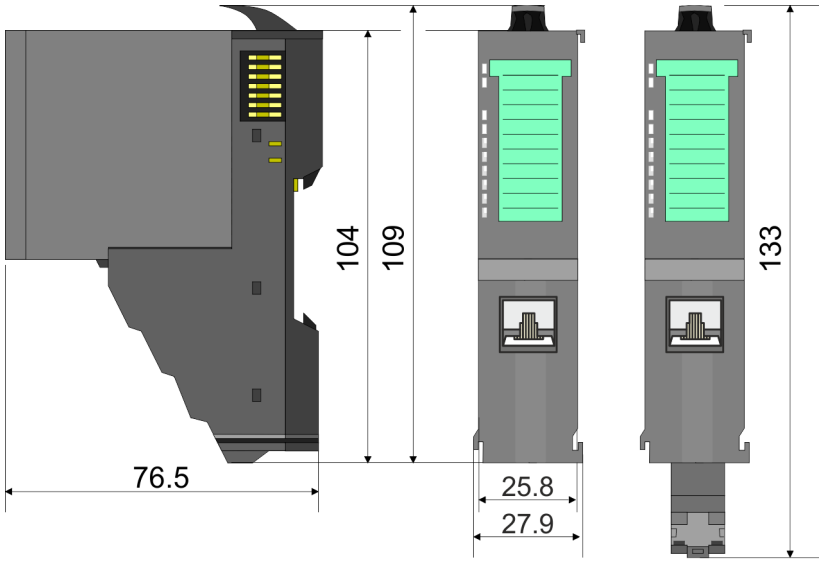


Dimensions CPU 01x

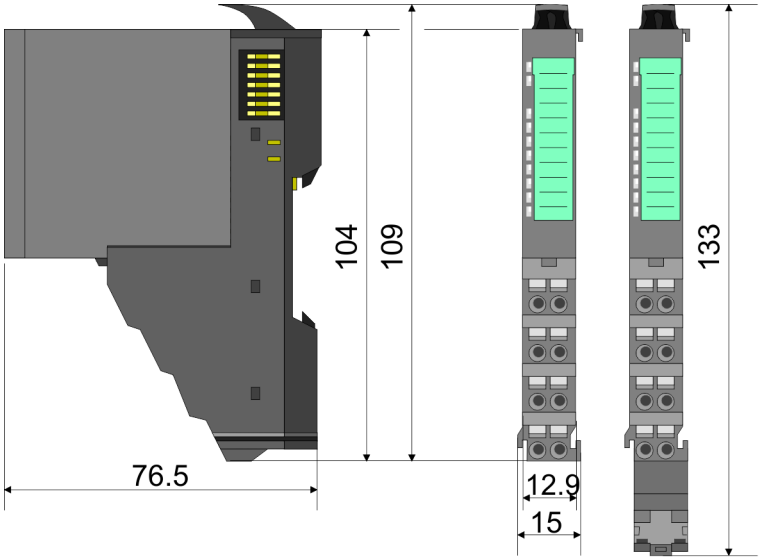
Dimensions bus coupler and line extension slave



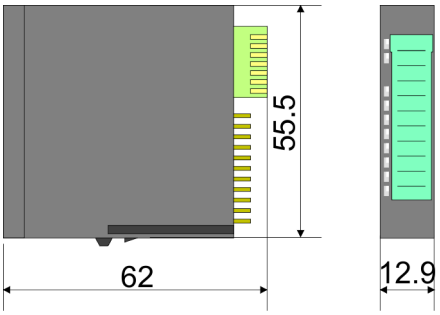
Dimensions line extension master



Dimension periphery module



Dimensions electronic module



Dimensions in mm

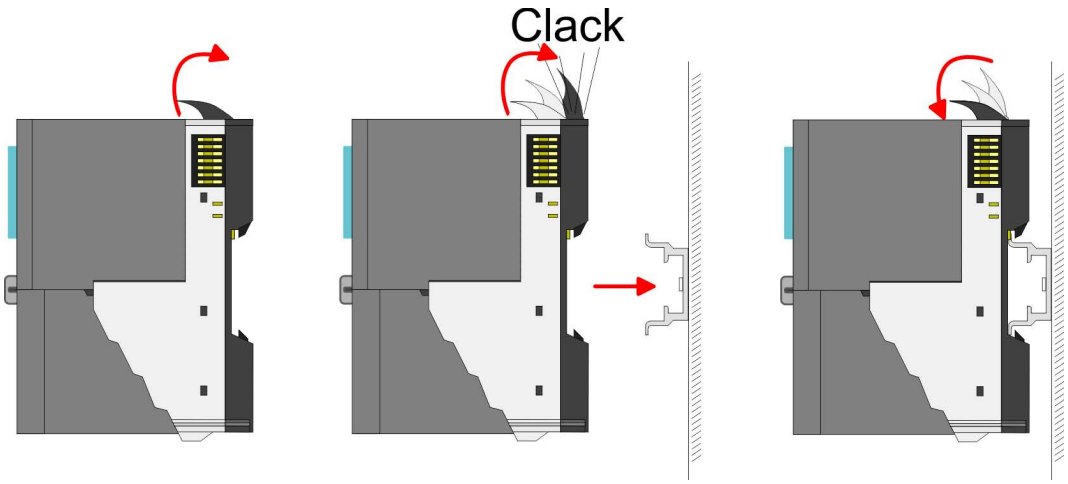
2.4 Mounting

Requirements for UL compliance use

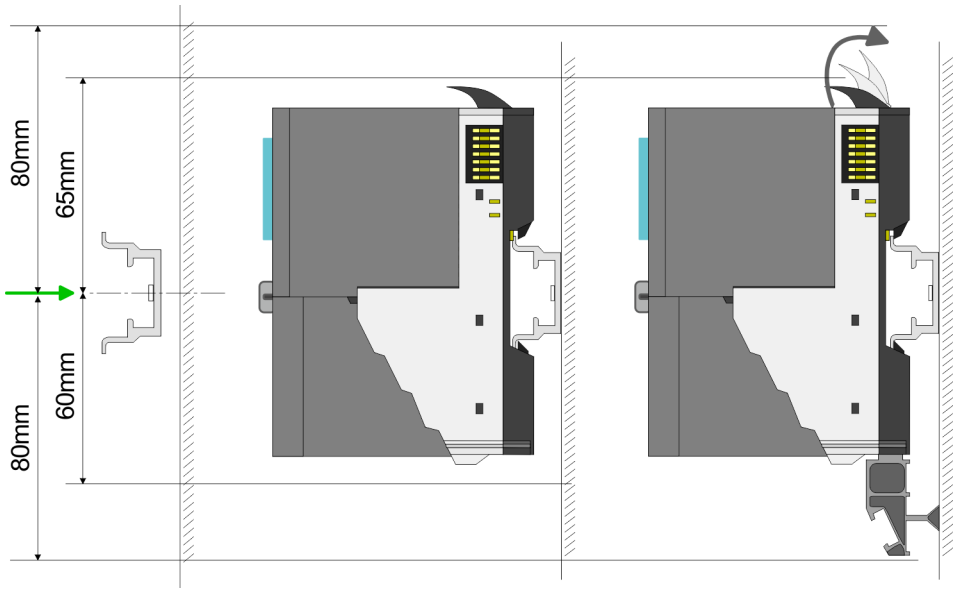
- *Use for power supply exclusively SELV/PELV power supplies.*
- *The System SLIO must be installed and operated in a housing according to IEC 61010-1 9.3.2 c).*

2.4.1 Mounting CPU 01x

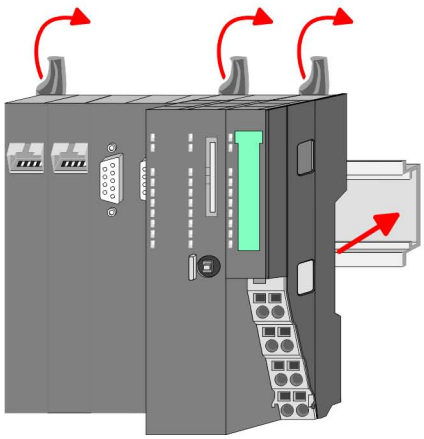
There are locking lever at the top side of the CPU. For mounting and demounting these locking lever are to be turned upwards until these engage. Place the CPU at the mounting rail. The CPU is fixed to the mounting rail by pushing downward the locking levers. The CPU is directly mounted at a mounting rail. Up to 64 modules may be mounted. The electronic and power section supply are connected via the backplane bus. Please consider here that the sum current of the electronic power supply does not exceed the maximum value of 3A. By means of the power module 007-1AB10 the current of the electronic power supply may be expanded accordingly.

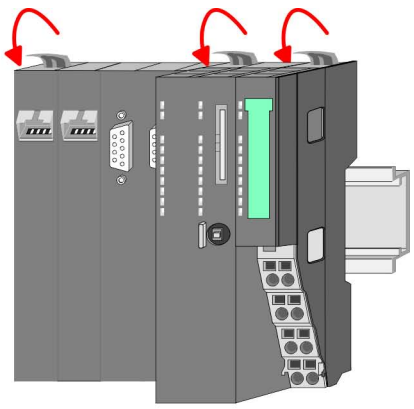


Proceeding



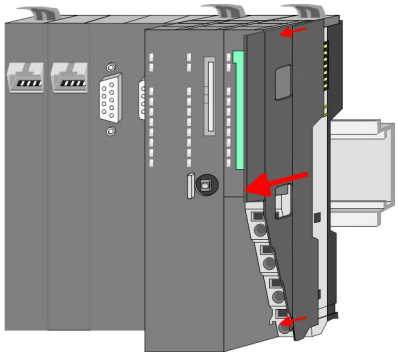
1. Mount the mounting rail! Please consider that a clearance from the middle of the mounting rail of at least 80mm above and 60mm below, respectively 80mm by deployment of shield bus carriers, exist.



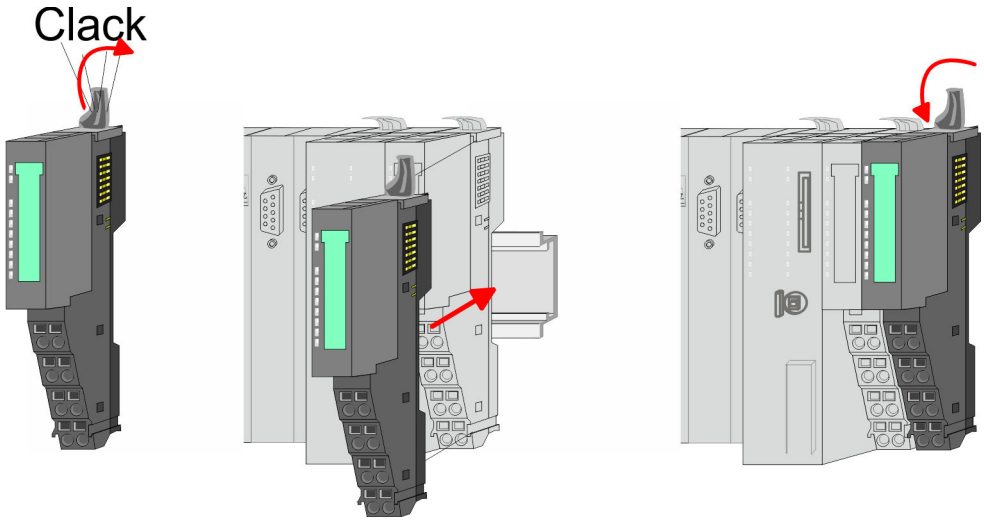


2. Turn the locking lever upwards, place the CPU at the mounting rail and turn the lever downward.

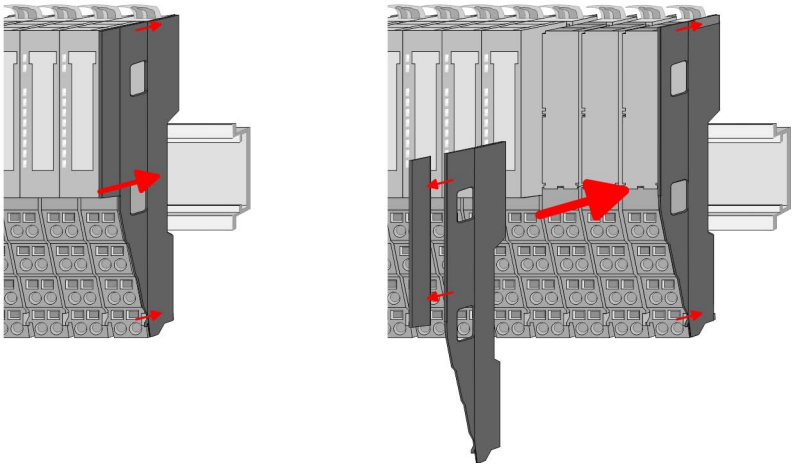
Mounting periphery modules



1. Before mounting the periphery modules you have to remove the bus cover at the right side of the CPU by pulling it forward. Keep the cover for later mounting.



2. Mount the periphery modules you want.



3. After mounting the whole system, to protect the backplane bus connectors at the last module you have to mount the bus cover, now. If the last module is a clamp module, for adaptation the upper part of the bus cover is to be removed.

2.5 Wiring

CAUTION!
Consider temperature for external cables!
Cables may experience temperature increase due to system heat dissipation. Thus the cabling specification must be chosen 5°C above ambient temperature!

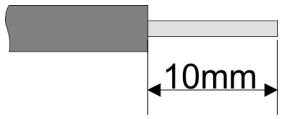
CAUTION!
Separate insulation areas!
The system is specified for SELV/PELV environment. Devices, which are attached to the system must meet theses specifications. Installation and cable routing other than SELV/PELV specification must be separated from the system's equipment!

2.5.1 Wiring CPU 01x

Terminal module terminals

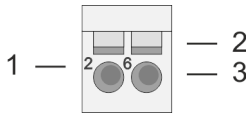
The System SLIO CPUs have a power module integrated. Terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines.

Data

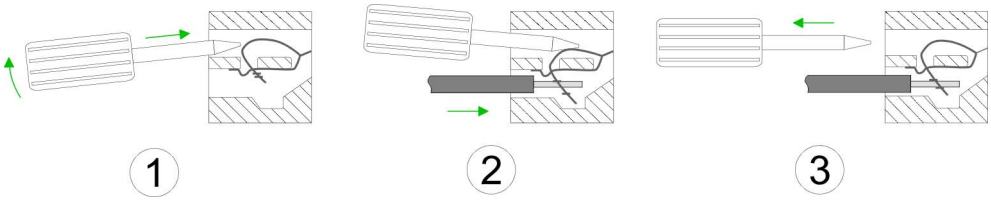


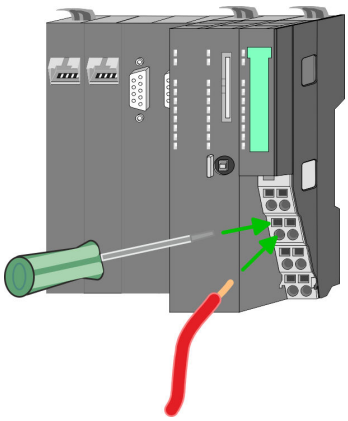
U_{max}	240V AC / 30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm^2 (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



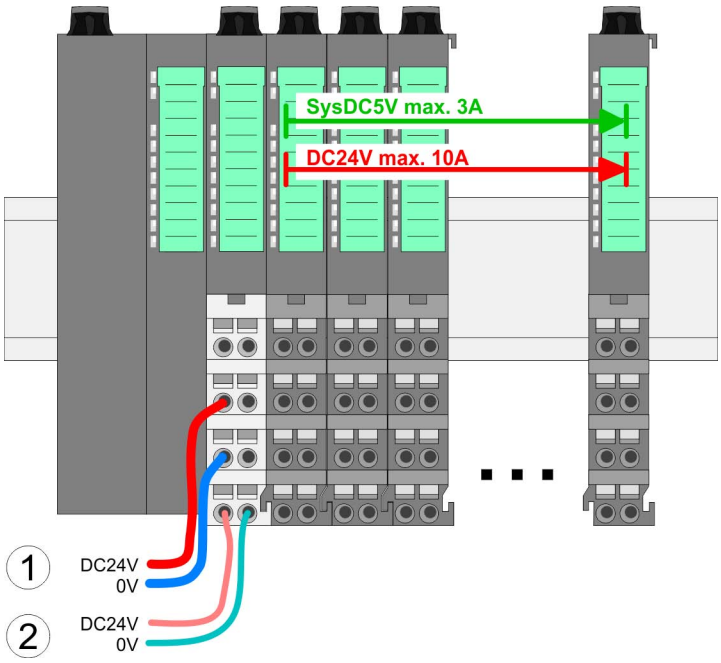
1 Pin number at the terminal module
2 Opening for screwdriver
3 Connection hole for wire





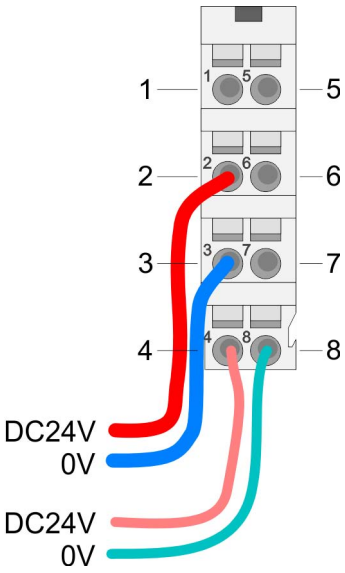
1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can connect wires with a cross section of 0.08mm^2 up to 1.5mm^2.
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

Standard wiring



(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area

PM - Power module



For wires with a core cross-section of 0.08mm² up to 1.5mm².

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

CAUTION!
Since the power section supply is not internally protected, it is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected by a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!

The electronic power section supply is internally protected against higher voltage by fuse. The fuse is within the power module. If the fuse releases, its electronic module must be exchanged!

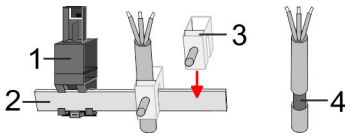
Fusing

- The power section supply is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected with a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!
- It is recommended to externally protect the electronic power supply for bus coupler and I/O area with a 2A fuse (fast) respectively by a line circuit breaker 2A characteristics Z.
- The electronic power supply for the I/O area of the power module 007-1AB10 should also be externally protected with a 1A fuse (fast) respectively by a line circuit breaker 1A characteristics Z.

State of the electronic power supply via LEDs

After PowerON of the System SLIO the LEDs RUN respectively MF get on so far as the sum current does not exceed 3A. With a sum current greater than 3A the LEDs may not be activated. Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

Shield attachment

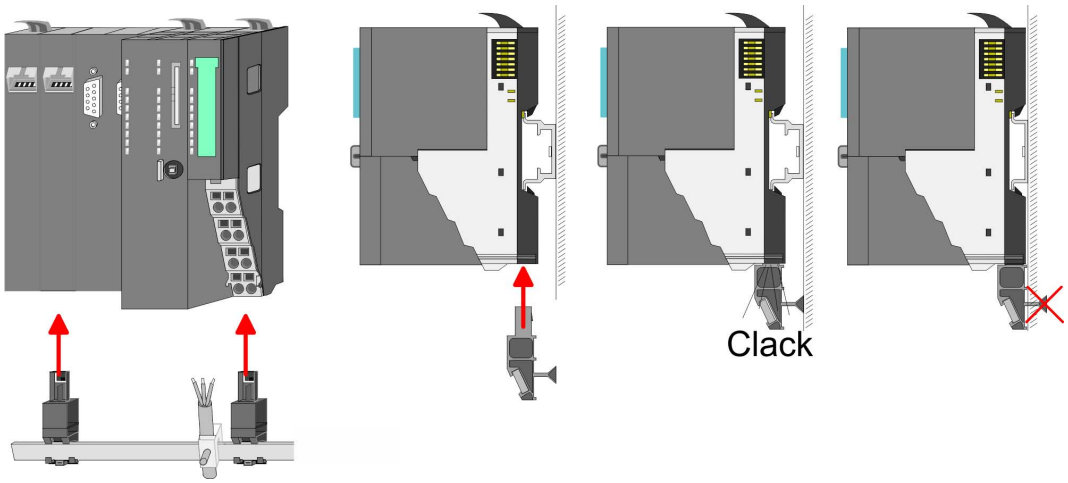


1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.

2. Put your shield bus into the shield bus carrier.



3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

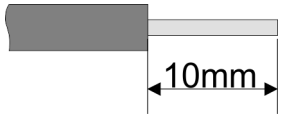
2.5.2 Wiring periphery modules

Terminal module terminals

CAUTION!
Do not connect hazardous voltages!
If this is not explicitly stated in the corresponding module description, hazardous voltages are not allowed to be connected to the corresponding terminal module!

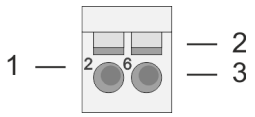
With wiring the terminal modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

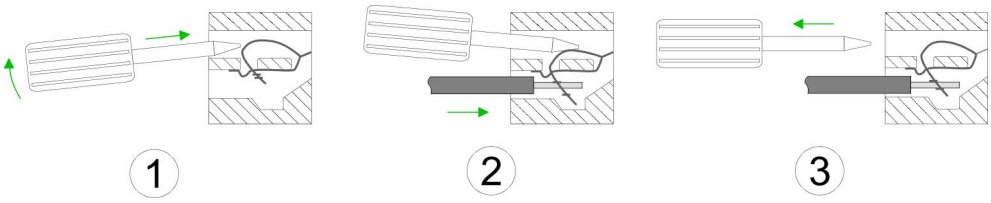


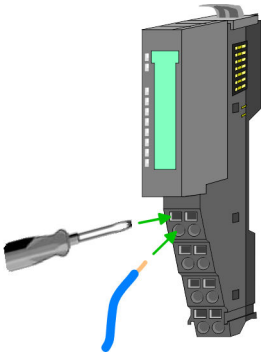
U_{max}	240V AC / 30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm^2 (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



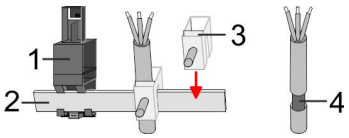
1 Pin number at the connector
2 Opening for screwdriver
3 Connection hole for wire





1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm² up to 1.5mm²
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

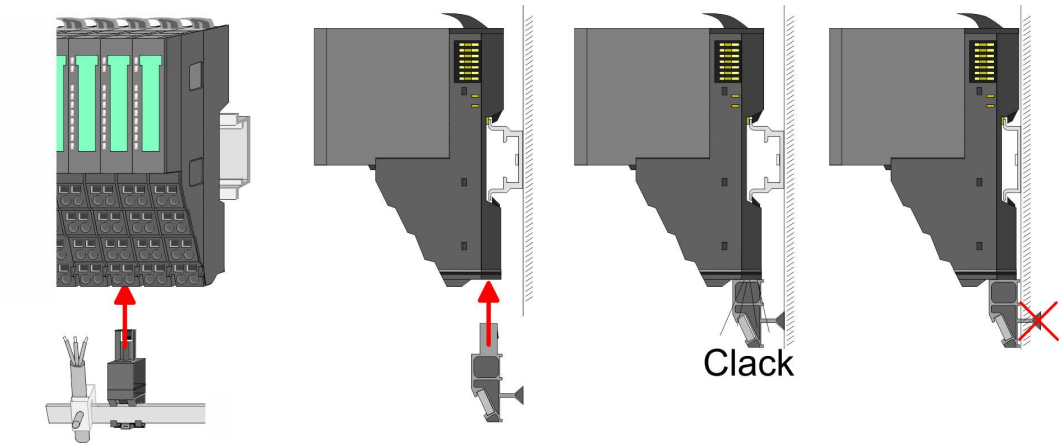
Shield attachment



1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.
2. Put your shield bus into the shield bus carrier.



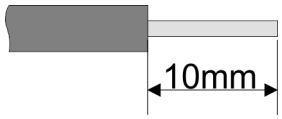
3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

2.5.3 Wiring power modules

Terminal module terminals

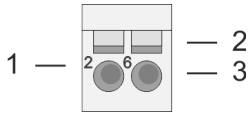
Power modules are either integrated to the head module or may be installed between the periphery modules. With power modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

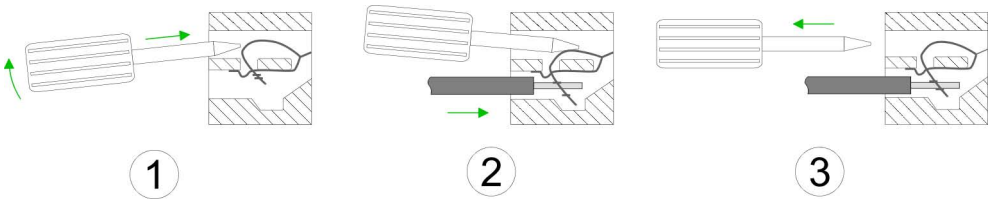


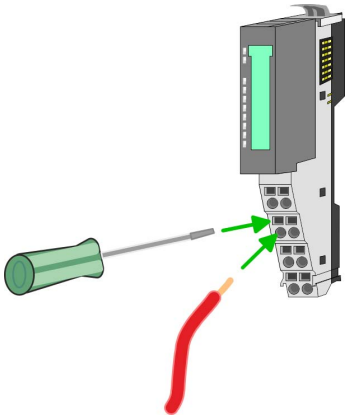
U_{max}	240V AC / 30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm^2 (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



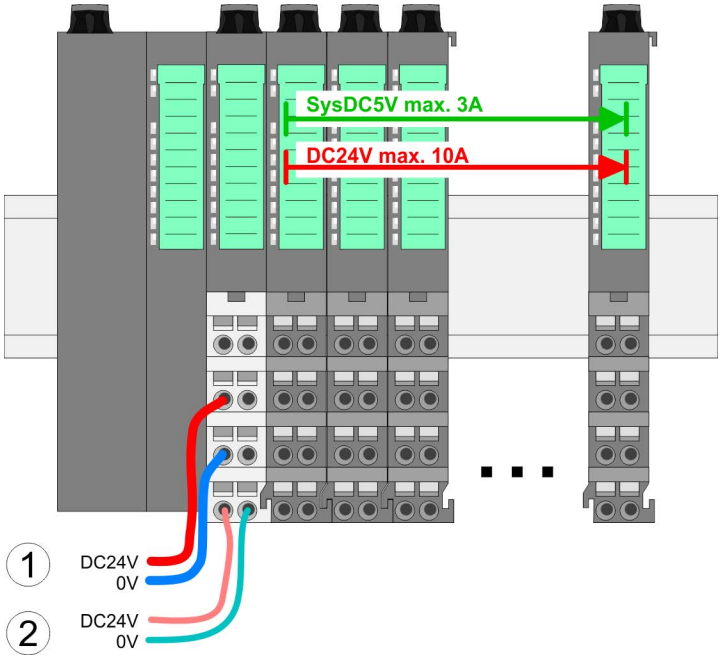
1 Pin number at the connector
2 Opening for screwdriver
3 Connection hole for wire





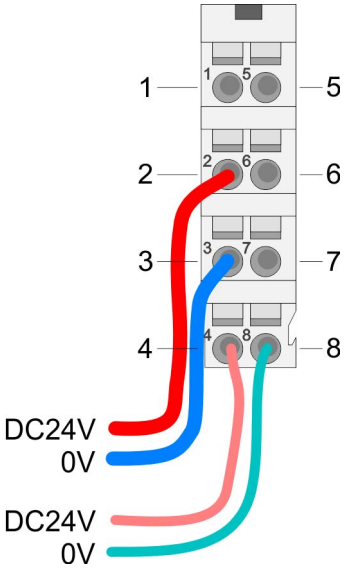
1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm^2 up to 1.5mm^2
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

Standard wiring



(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area

PM - Power module



For wires with a core cross-section of 0.08mm² up to 1.5mm².

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

CAUTION!
Since the power section supply is not internally protected, it is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected by a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!

The electronic power section supply is internally protected against higher voltage by fuse. The fuse is within the power module. If the fuse releases, its electronic module must be exchanged!

Fusing

- The power section supply is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected with a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!
- It is recommended to externally protect the electronic power supply for head modules and I/O area with a 2A fuse (fast) respectively by a line circuit breaker 2A characteristics Z.
- The electronic power supply for the I/O area of the power module 007-1AB10 should also be externally protected with a 1A fuse (fast) respectively by a line circuit breaker 1A characteristics Z.

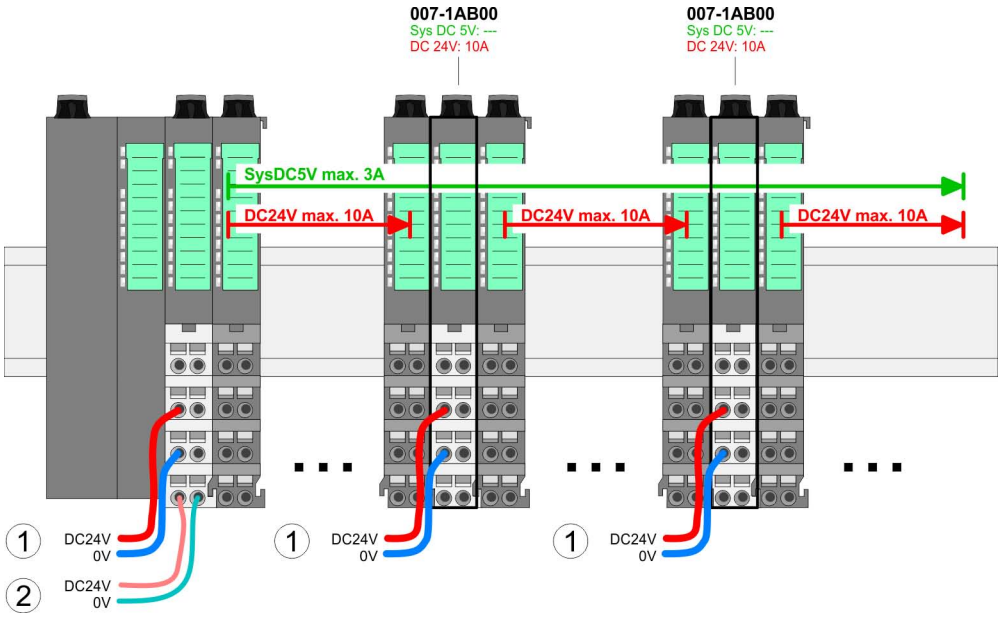
State of the electronic power supply via LEDs

After PowerON of the System SLIO the LEDs RUN respectively MF get on so far as the sum current does not exceed 3A. With a sum current greater than 3A the LEDs may not be activated. Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

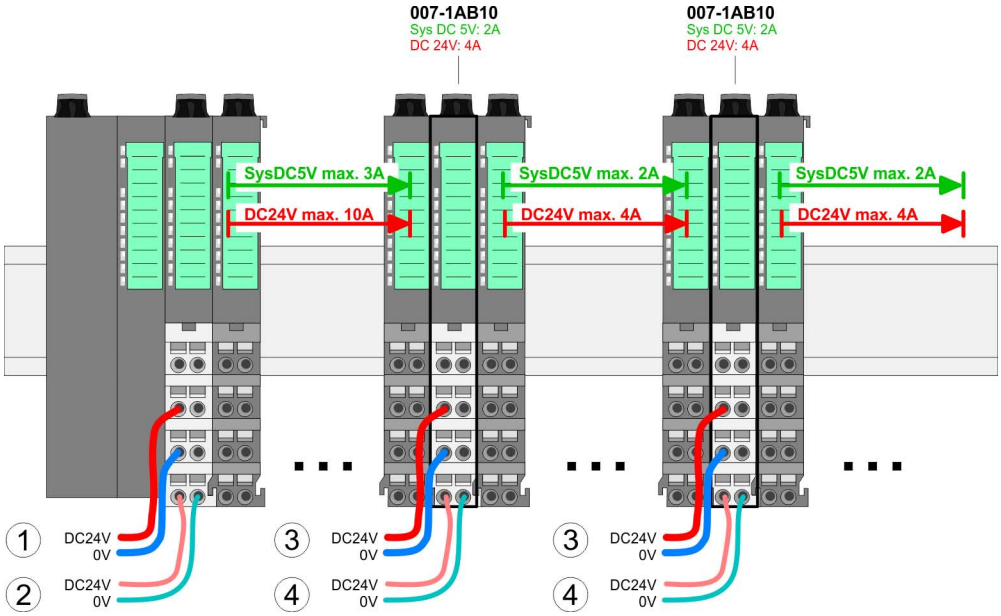
Deployment of the power modules

- If the 10A for the power section supply is no longer sufficient, you may use the power module from VIPA with the order number 007-1AB00. So you have also the possibility to define isolated groups.
- The power module with the order number 007-1AB10 is to be used if the 3A for the electronic power supply at the backplane bus is no longer sufficient. Additionally you get an isolated group for the DC 24V power section supply with max. 4A.
- By placing the power module 007-1AB10 at the following backplane bus modules may be placed with a sum current of max. 2A. Afterwards a power module is to be placed again. To secure the power supply, the power modules may be mixed used.

Power module 007-1AB00

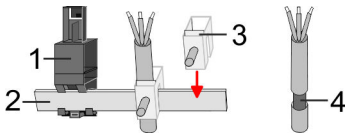


Power module 007-1AB10



(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area
(3) DC 24V for power section supply I/O area (max. 4A)
(4) DC 24V for electronic power supply I/O area

Shield attachment

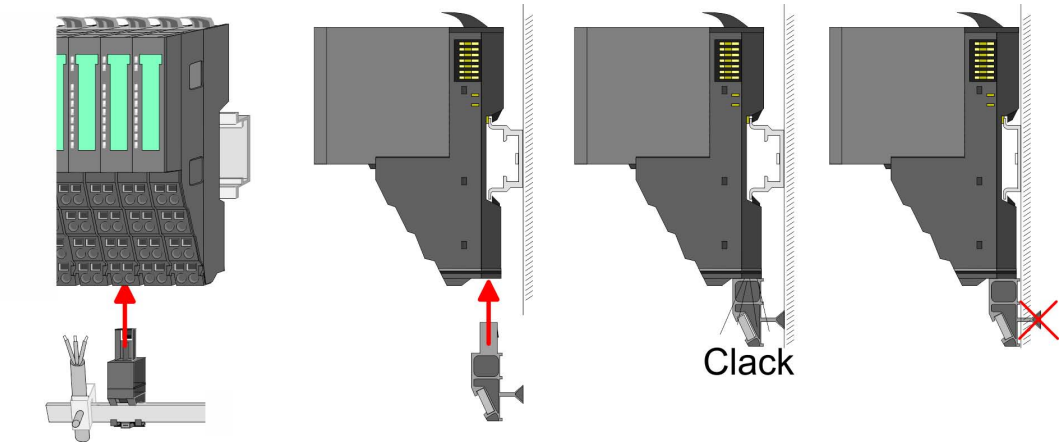


1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.

2. Put your shield bus into the shield bus carrier.



3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

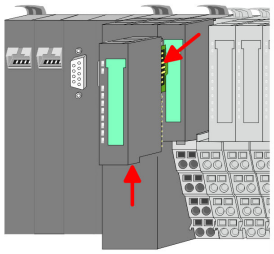
2.6 Demounting

2.6.1 Demounting CPU 01x

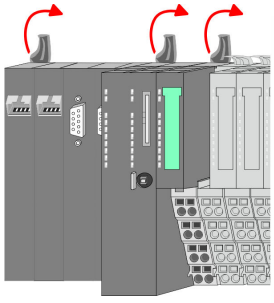
Proceeding

CAUTION!
CPU part and power module may not be separated! Here you may only exchange the electronic module!

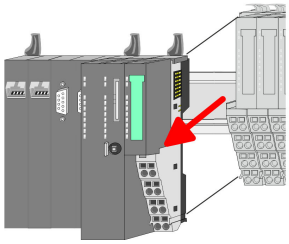
1. Power-off your system.
2. Remove if exists the wiring of the CPU.
3. *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*



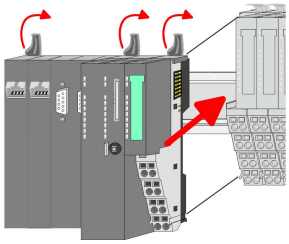
Press the unlocking lever at the lower side of the just mounted right module near the CPU and pull it forward.



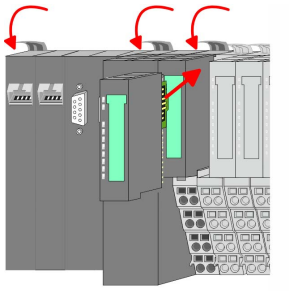
4. Turn all the locking lever of the CPU to be exchanged upwards.



5. Pull the CPU forward.
6. For mounting turn all the locking lever of the CPU to be mounted upwards.



7. To mount the CPU put it to the left periphery module and push it, guided by the stripes, to the mounting rail.
8. Turn all the locking lever downward, again.



9. Plug again the electronic module, which you have removed before. For installation plug the electronic module guided by the strips at the lower side until this engages to the terminal module.
10. Wire your CPU.

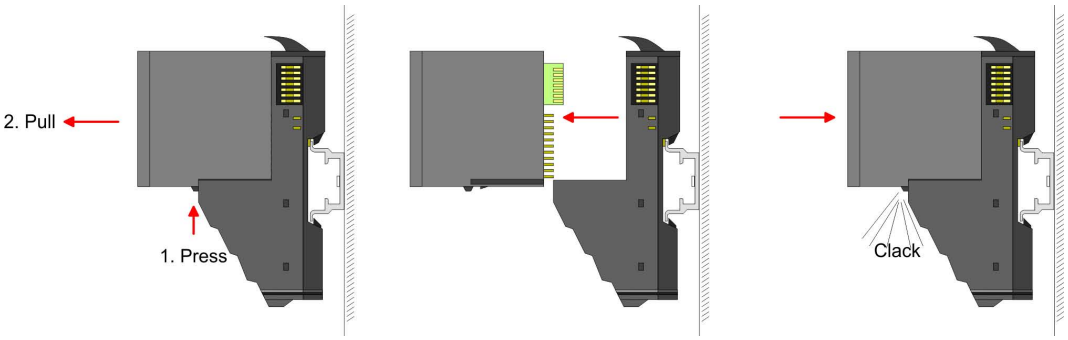
    ⇨ Now you can bring your system back into operation.

2.6.2 Demounting periphery modules

Proceeding

Exchange of an electronic module

1. Power-off your system.



2. For the exchange of a electronic module, the electronic module may be pulled forward after pressing the unlocking lever at the lower side of the module.
3. For installation plug the new electronic module guided by the strips at the lower side until this engages to the terminal module.

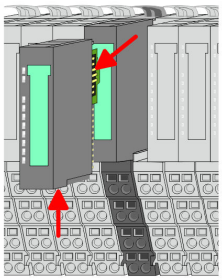
   ⇨ Now you can bring your system back into operation.

Exchange of a periphery module

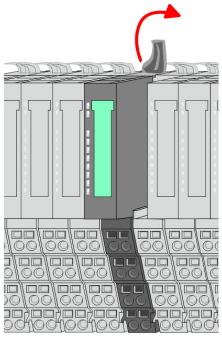
1. Power-off your system.
2. Remove if exists the wiring of the module.
3.

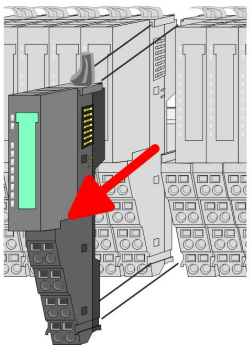
> *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

Press the unlocking lever at the lower side of the just mounted right module and pull it forward.

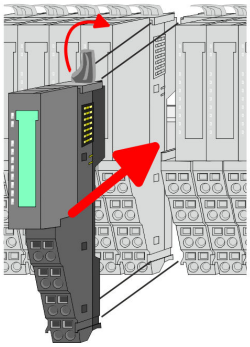


4. Turn the locking lever of the module to be exchanged upwards.

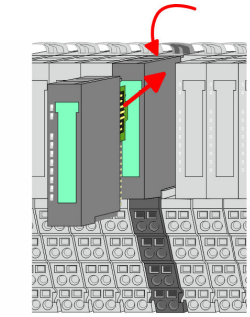




5. Pull the module.
6. For mounting turn the locking lever of the module to be mounted upwards.



7. To mount the module put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.
8. Turn the locking lever downward, again.

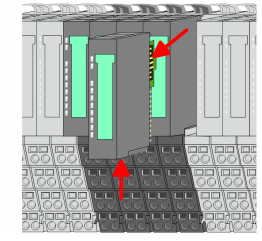


9. Plug again the electronic module, which you have removed before.
10. Wire your module.

⇨ Now you can bring your system back into operation.

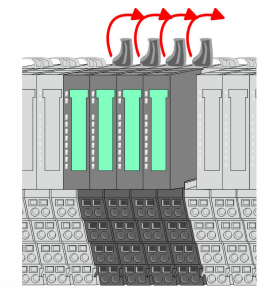
Exchange of a module group

1. Power-off your system.
2. Remove if exists the wiring of the module group.

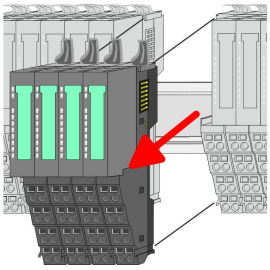


3. *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

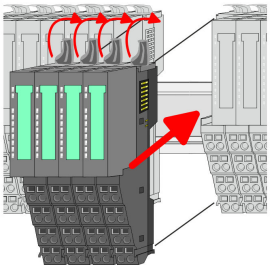
Press the unlocking lever at the lower side of the just mounted right module near the module group and pull it forward.



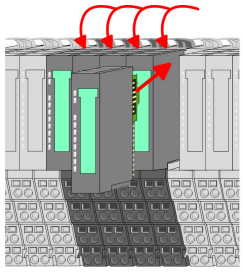
4. Turn all the locking lever of the module group to be exchanged upwards.



5. Pull the module group forward.
6. For mounting turn all the locking lever of the module group to be mounted upwards.



7. To mount the module group put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.
8. Turn all the locking lever downward, again.



9. Plug again the electronic module, which you have removed before.
10. Wire your module group.

    ⇨ Now you can bring your system back into operation.

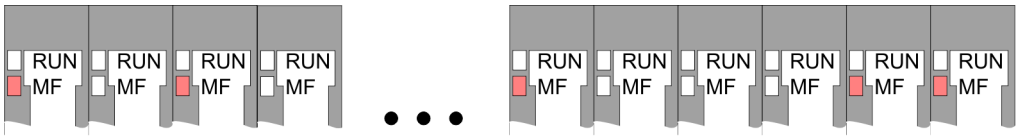
2.7 Trouble shooting - LEDs

General

Each module has the LEDs RUN and MF on its front side. Errors or incorrect modules may be located by means of these LEDs.

In the following illustrations flashing LEDs are marked by ☼.

Sum current of the electronic power supply exceeded

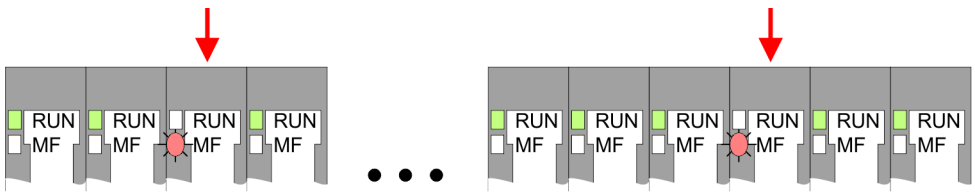


Behaviour: After PowerON the RUN LED of each module is off and the MF LED of each module is sporadically on.

Reason: The maximum current for the electronic power supply is exceeded.

Remedy: As soon as the sum current of the electronic power supply is exceeded, always place the power module 007-1AB10. ↳ *Chapter 2.5.3 'Wiring power modules' on page 25*

Error in configuration

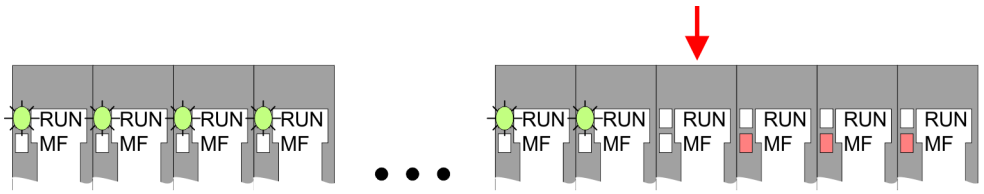


Behaviour: After PowerON the MF LED of one module respectively more modules flashes. The RUN LED remains off.

Reason: At this position a module is placed, which does not correspond to the configured module.

Remedy: Match configuration and hardware structure.

Module failure



Behaviour: After PowerON all of the RUN LEDs up to the defective module are flashing. With all following modules the MF LED is on and the RUN LED is off.

Reason: The module on the right of the flashing modules is defective.

Remedy: Replace the defective module.

2.8 Installation guidelines

General

The installation guidelines contain information about the interference free deployment of a PLC system. There is the description of the ways, interference may occur in your PLC, how you can make sure the electromagnetic compatibility (EMC), and how you manage the isolation.

What does EMC mean?

Electromagnetic compatibility (EMC) means the ability of an electrical device, to function error free in an electromagnetic environment without being interfered respectively without interfering the environment.

The components of VIPA are developed for the deployment in industrial environments and meets high demands on the EMC. Nevertheless you should project an EMC planning before installing the components and take conceivable interference causes into account.

Possible interference causes

Electromagnetic interferences may interfere your control via different ways:

- Electromagnetic fields (RF coupling)
- Magnetic fields with power frequency
- Bus system
- Power supply
- Protected earth conductor

Depending on the spreading medium (lead bound or lead free) and the distance to the interference cause, interferences to your control occur by means of different coupling mechanisms.

There are:

- galvanic coupling
- capacitive coupling
- inductive coupling
- radiant coupling

Basic rules for EMC

In the most times it is enough to take care of some elementary rules to guarantee the EMC. Please regard the following basic rules when installing your PLC.

- Take care of a correct area-wide grounding of the inactive metal parts when installing your components.
  - Install a central connection between the ground and the protected earth conductor system.
  - Connect all inactive metal extensive and impedance-low.
  - Please try not to use aluminium parts. Aluminium is easily oxidizing and is therefore less suitable for grounding.
- When cabling, take care of the correct line routing.
  - Organize your cabling in line groups (high voltage, current supply, signal and data lines).
  - Always lay your high voltage lines and signal respectively data lines in separate channels or bundles.
  - Route the signal and data lines as near as possible beside ground areas (e.g. suspension bars, metal rails, tin cabinet).

- Proof the correct fixing of the lead isolation.
  - Data lines must be laid isolated.
  - Analog lines must be laid isolated. When transmitting signals with small amplitudes the one sided laying of the isolation may be favourable.
  - Lay the line isolation extensively on an isolation/protected earth conductor rail directly after the cabinet entry and fix the isolation with cable clamps.
  - Make sure that the isolation/protected earth conductor rail is connected impedance-low with the cabinet.
  - Use metallic or metallised plug cases for isolated data lines.
- In special use cases you should appoint special EMC actions.
  - Consider to wire all inductivities with erase links.
  - Please consider luminescent lamps can influence signal lines.
- Create a homogeneous reference potential and ground all electrical operating supplies when possible.
  - Please take care for the targeted employment of the grounding actions. The grounding of the PLC serves for protection and functionality activity.
  - Connect installation parts and cabinets with your PLC in star topology with the isolation/protected earth conductor system. So you avoid ground loops.
  - If there are potential differences between installation parts and cabinets, lay sufficiently dimensioned potential compensation lines.

Isolation of conductors

Electrical, magnetically and electromagnetic interference fields are weakened by means of an isolation, one talks of absorption. Via the isolation rail, that is connected conductive with the rack, interference currents are shunt via cable isolation to the ground. Here you have to make sure, that the connection to the protected earth conductor is impedance-low, because otherwise the interference currents may appear as interference cause.

When isolating cables you have to regard the following:

- If possible, use only cables with isolation tangle.
- The hiding power of the isolation should be higher than 80%.
- Normally you should always lay the isolation of cables on both sides. Only by means of the both-sided connection of the isolation you achieve high quality interference suppression in the higher frequency area. Only as exception you may also lay the isolation one-sided. Then you only achieve the absorption of the lower frequencies. A one-sided isolation connection may be convenient, if:
  - the conduction of a potential compensating line is not possible.
  - analog signals (some mV respectively µA) are transferred.
  - foil isolations (static isolations) are used.
- With data lines always use metallic or metallised plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!
- At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.
- To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.
- Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to your PLC and don't lay it on there again!

CAUTION!

Please regard at installation!

At potential differences between the grounding points, there may be a compensation current via the isolation connected at both sides.

Remedy: Potential compensation line

2.9 General data

Conformity and approval		
Conformity		
CE	2014/35/EU	Low-voltage directive
	2014/30/EU	EMC directive
Approval		
UL	-	Refer to Technical data
others		
RoHS	2011/65/EU	Restriction of the use of certain hazardous substances in electrical and electronic equipment

Protection of persons and device protection		
Type of protection	-	IP20
Electrical isolation		
to the field bus	-	electrically isolated
to the process level	-	electrically isolated
Insulation resistance	-	-
Insulation voltage to reference earth		
Inputs / outputs	-	AC / DC 50V, test voltage AC 500V
Protective measures	-	against short circuit

Environmental conditions to EN 61131-2		
Climatic		
Storage / transport	EN 60068-2-14	-25…+70°C
Operation		
Horizontal installation hanging	EN 61131-2	0…+60°C
Horizontal installation lying	EN 61131-2	0…+55°C
Vertical installation	EN 61131-2	0…+50°C
Air humidity	EN 60068-2-30	RH1 (without condensation, rel. humidity 10…95%)
Pollution	EN 61131-2	Degree of pollution 2
Installation altitude max.	-	2000m
Mechanical		
Oscillation	EN 60068-2-6	1g, 9Hz ... 150Hz
Shock	EN 60068-2-27	15g, 11ms

General data

Mounting conditions		
Mounting place	-	In the control cabinet
Mounting position	-	Horizontal and vertical

EMC	Standard		Comment
Emitted interference	EN 61000-6-4		Class A (Industrial area)
Noise immunity zone B	EN 61000-6-2		Industrial area
		EN 61000-4-2	ESD 8kV at air discharge (degree of severity 3), 4kV at contact discharge (degree of severity 2)
		EN 61000-4-3	HF field immunity (casing) 80MHz … 1000MHz, 10V/m, 80% AM (1kHz) 1.4GHz ... 2.0GHz, 3V/m, 80% AM (1kHz) 2GHz ... 2.7GHz, 1V/m, 80% AM (1kHz)
		EN 61000-4-6	HF conducted 150kHz … 80MHz, 10V, 80% AM (1kHz)
		EN 61000-4-4	Burst, degree of severity 3
		EN 61000-4-5	Surge, degree of severity 3 *

*) Due to the high-energetic single pulses with Surge an appropriate external protective circuit with lightning protection elements like conductors for lightning and overvoltage is necessary.

3 Hardware description

3.1 Properties

CPU 015N

- SPEED7 technology integrated
- Programmable via VIPA *SPEED7 Studio*, Siemens SIMATIC Manager, Siemens TIA Portal
- 256kbyte work memory integrated (128kbyte code, 128kbyte data)
- Work memory expandable up to 512kbyte (256kbyte code, 256kbyte data)
- 512kbyte load memory integrated
- Slot for external storage media (lockable)
- Status LEDs for operating state and diagnostics
- X1/X5: Ethernet PG/OP channel integrated
- X2: PtP(MPI) interface Serial integrated interface for PtP communication with the protocols: ASCII, STX/ETX , USS, 3964(R), MODBUS RTU, master/slave switch able to MPI communication
- X3: MPI(PB) interface: MPI interface with via VSC unlock able field bus functions
- X4: Ethernet interface Ethernet interface with EtherCAT master functionality
- X6: NET CP: Ethernet-interface for TCP/IP communication
- Up to 64 SLIO modules placeable
- I/O address area digital/analog 2048byte
- 512 timer/counter, 8192 flag byte

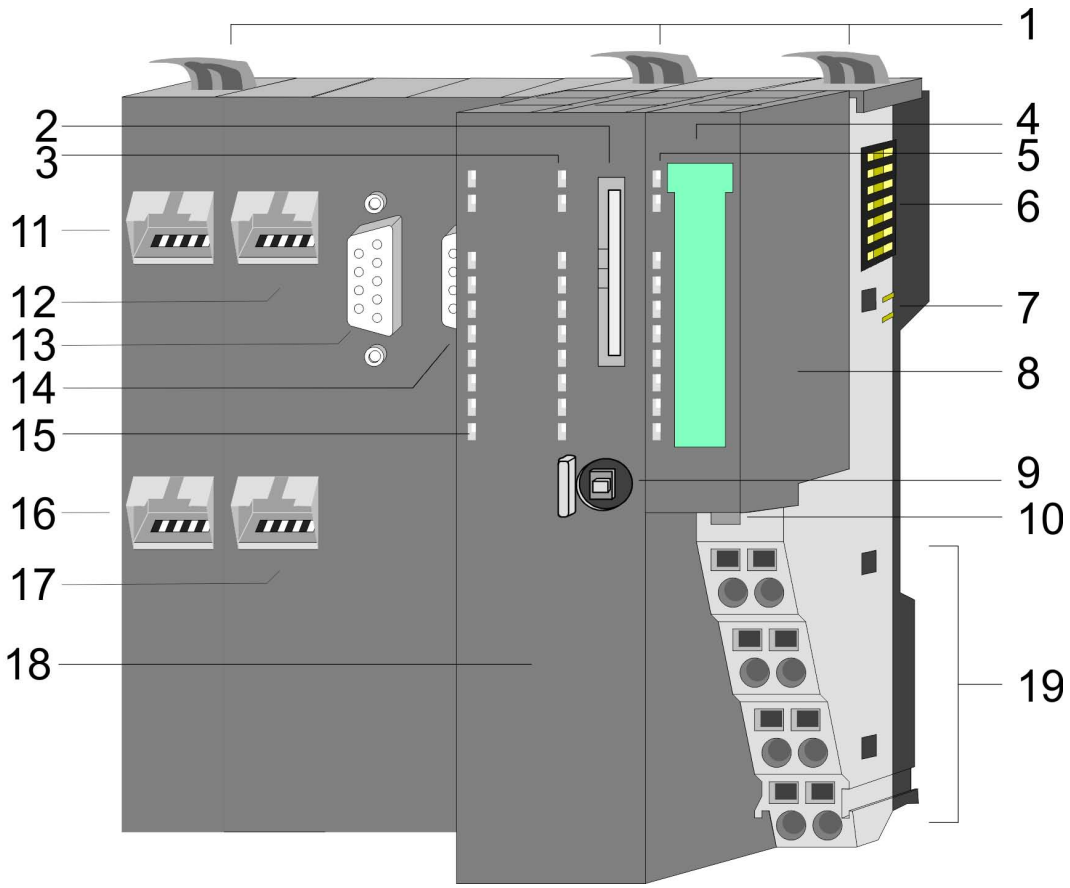
Ordering data

Type	Order number	Description
CPU 015N	015-CEFNR00	Basic CPU 015N with NET CP communication processor and EtherCAT master and options to extend work memory and bus interface.

3.2 Structure

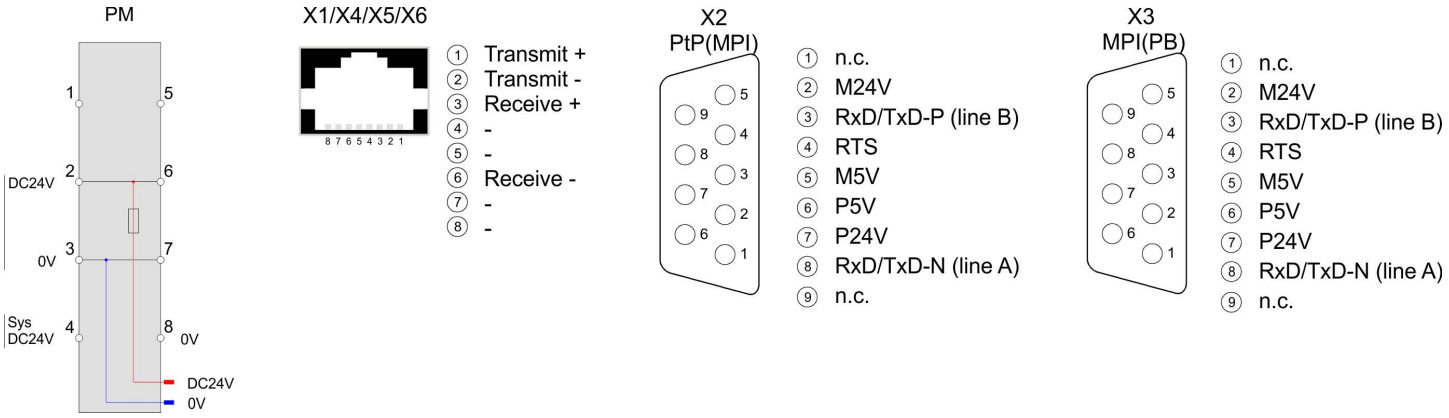
3.2.1 Basic CPU

CPU 015N



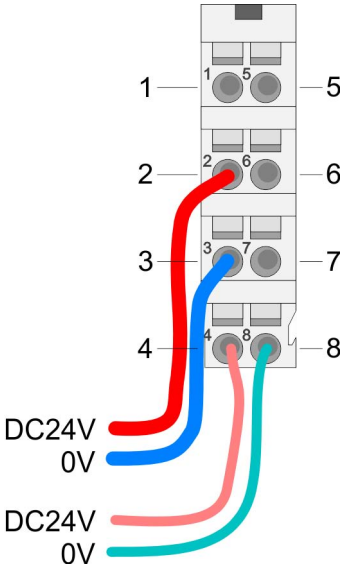
1 Locking lever
2 Slot for external storage media (lockable)
3 LEDs CPU part
4 Labelling strip power module
5 LED status indication power module
6 Backplane bus
7 DC 24V power section supply
8 Power module
9 Operating mode switch CPU
10 Unlocking lever power module
11 X4: EtherCAT master
12 X1: Ethernet PG/OP channel (switch)
13 X2: PtP(MPI) interface
14 X3: MPI(PB) interface
15 LED status indication EtherCAT master
16 X6: NET CP
17 X5: Ethernet PG/OP channel (switch)
18 CPU part
19 Terminal power module

3.2.2 Interfaces



CAUTION!
CPU part and power module may not be separated! Here you may only exchange the electronic module!

PM - Power module



For wires with a core cross-section of 0.08mm² up to 1.5mm².

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

X1/X5: Ethernet PG/OP channel

8pin RJ45 jack:

- The RJ45 jack serves as interface to the Ethernet PG/OP channel.
- This interface allows you to program respectively remote control your CPU and to access the internal web server.
- The connection happens via an integrated 2-port switch
- DHCP respectivle the assignment of the network configuration by specifying a DHCP server is supported.
- Configurable connections are not possible.
- Default diagnostic addresses: 2025 ... 2040
- For online access to the CPU via Ethernet PG/OP channel, you have to assign IP address parameters to this.

Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61

X2: PtP(MPI) interface

9pin SubD jack: (isolated):

The interface supports the following functions, which are switch able via the *VIPA specific CPU parameters* *63*:

- PtP (default / after overall reset)
  Per default, the RS485 interface is set to PtP functionality. Using the *PtP* functionality the RS485 interface is allowed to connect via serial point-to-point connection to different source res. target systems.
  The following protocols are supported:
  – ASCII
  – STX/ETX
  – 3964R
  – USS
  – Modbus master (ASCII, RTU)
- MPI
  The MPI interface serves for the connection between programming unit and CPU. By means of this the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU. Standard setting is MPI address 2.

X3: MPI(PB) interface

9pin SubD jack: (isolated):

The interface supports the following functions, which are switch able via *'MPI interface'* in the hardware configuration:

- MPI (default / after reset to factory setting *Chapter 4.13 'Reset to factory settings' on page 84*)
  Per default, the RS485 interface is set to MPI functionality. The MPI interface serves for the connection between programming unit and CPU. By means of this the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU. Standard setting is MPI address 2.
- PB
  The PROFIBUS master/slave functionality of this interface can be activated by configuring the *'MPI interface'* of the CPU in the hardware configuration.

Enable bus functionality via VSC

To switch the MPI(PB) interface X3 to PROFIBUS functionality, you have to enable the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is enabled.

'Overview' on page 85

X4: EtherCAT master

8pin RJ45 jack:

- Connect this interface with the RJ45 jack "IN" of your slave station.
- EtherCAT uses Ethernet as transfer medium. Standard CAT5 cables are used. Here distances of about 100m between 2 stations are possible.
- Only EtherCAT components may be used in an EtherCAT network. For topologies, which depart from the line structure, the corresponding EtherCAT components are necessary. Hubs may not be used.

- An EtherCAT network always consists of a master and an various number of EtherCAT salves (coupler).
- Each EtherCAT slave has an "IN" and "OUT" RJ45 jack. The arriving EtherCAT cable from the direction of the master is to be connected to the "IN" jack. The "OUT" jack is to be connected to the next station. With the respective last station the "OUT" jack remains free.

CAUTION!
Using a switch
When using an EoE terminal (Ethernet over EtherCAT) X4 and X6 must not be connected to the same switch! Due to the internal connection, this leads to a ring closure on Ethernet.

X6: NET CP

8pin RJ45 jack:

- NET CP Ethernet interface for TCP/IP communication
- Productive connections via configuration
- Productive connections via user program
- PG/OP connections

3.2.3 Memory management

General

The CPU has an integrated memory. Information about the capacity of the memory may be found at the front of the CPU. The memory is divided into the following parts:

- Load memory 512kbyte
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 256kbyte
  - There is the possibility to extend the work memory to its maximum capacity 512kbyte by means of a VSC.

3.2.4 Slot for storage media

Overview

At this slot the following storage media can be plugged:

- VSD - **V**IPA **SD**-Card
  - External memory card for programs and firmware.
- VSC - **V**IPA**S**et**C**ard
  - External memory card (VSD) for programs and firmware with the possibility to unlock optional functions like work memory and field bus interfaces.
  - These functions can be purchased separately. *Chapter 4.14 'Deployment storage media - VSD, VSC' on page 85*
  - To activate the corresponding card is to be installed and a *Overall reset* is to be established. *Chapter 4.11 'Overall reset' on page 81*

A list of the currently available VSD respectively VSC can be found at www.vipa.com.

3.2.5 Buffering mechanisms

The SLIO CPU has a capacitor-based mechanism to buffer the internal clock in case of power failure for max. 30 days. With PowerOFF the content of the RAM is automatically stored in the Flash (NVRAM).

CAUTION!
Please connect the CPU for approximately 1 hour to the power supply, so that the internal buffering mechanism is loaded accordingly.

In case of failure of the buffer mechanism Date and Time 01.09.2009 00:00:00 set. Additionally, you receive a diagnostics message. *Chapter 4.18 'Diagnostic entries' on page 92*

3.2.6 Operating mode switch

General

- With the operating mode switch you may switch the CPU between STOP and RUN.
- During the transition from STOP to RUN the operating mode START-UP is driven by the CPU.
- Placing the switch to MR (**M**emory **R**eset), you request an overall reset with following load from memory card, if a project there exists.

3.2.7 LEDs

CPU part

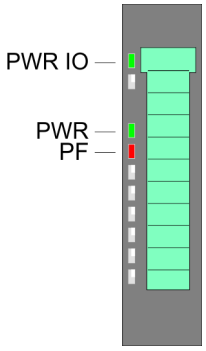
RN green	ST yellow	SF red	FC yellow	SD yellow	Description
Boot-up after PowerON - as soon as the CPU is supplied with 5V, the green PW-LED (Power) is on.					
	X	flickers			Firmware is loaded.
					Initialization: Phase 1
					Initialization: Phase 2
					Initialization: Phase 3
					Initialization: Phase 4
Operation					
		X	X	X	CPU is in STOP state.
2Hz		X	X	X	CPU is in start-up state. During the start-up (OB 100) the RUN LED blinks for at least 3s.
	10Hz	X	X	X	Activation of a new hardware configuration
			X	X	CPU is in state RUN without error.
X	X		X	X	There is a system fault. More information can be found in the diagnostics buffer of the CPU.
X	X	X		X	Variables are forced.

RN green	ST yellow	SF red	FC yellow	SD yellow	Description
X	X	X	X		Accessing the memory card.
X	10Hz	X	X	X	Configuration is loaded.
Overall reset					
	1Hz	X	X	X	Overall reset is requested
	2Hz	X	X	X	Overall reset is executed.
	10Hz	X	X	X	Overall reset with none hardware configuration respectively hardware configuration from memory card.
Factory reset					
					Reset to factory setting is executed.
					Reset to factory setting finished without error Then power OFF/ON is mandatory.
Firmware update					
		2Hz	2Hz		The alternate blinking indicates that there is new firmware on the memory card.
		2Hz	2Hz		The alternate blinking indicates that a firmware update is executed.
					Firmware update finished without error.
	10Hz	10Hz	10Hz	10Hz	Error during Firmware update.
not relevant: X					

Ethernet PG/OP channel

L/A (Link/Activity) green	S (Speed) green	Description
	X	The Ethernet PG/OP channel is physically connected to Ethernet.
	X	There is no physical connection.
flickers	X	Shows Ethernet activity.
		The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 100Mbit.
		The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 10Mbit.
not relevant: X		

LEDs power module



PWR IO green	PWR green	PF red	Description
☐	☐	☐	Both power supplies are missing
■	X	☐	Power section supply OK
X	■	☐	Electronic section OK
X	X	■	Fuse electronic section supply defective
not relevant: X			

CAUTION!
CPU part and power module may not be separated! Here you may only exchange the electronic module!

LEDs PROFIBUS

Dependent on the mode of operation the LEDs show information about the state of operation of the PROFIBUS part according to the following pattern:

Master operation

DE (Data Exchange) green	BF (bus error) red	Description
☐	☐	Master has no project, this means the interface is deactivated respectively the master configured without slaves with no errors.
◪ 2Hz	☐	CPU is in STOP state, the master is in "clear" state. All the slaves are in DE and the outputs are of the slaves are disabled.
■	☐	CPU is in RUN state, the master is in "operate" state. All the slaves are in DE. The outputs are enabled.
■	◪ 2Hz	CPU is in RUN state, at least 1 slave is missing and at least 1 slave is in DE.
◪ 2Hz	◪ 2Hz	CPU is in STOP state, the master is in "clear" state. At least 1 slave is missing and 1 slave is in DE.
☐	■	PROFIBUS is interrupted (no communication possible)
☐	◪ 2Hz	At least 1 slave is missing and no slave is in DE.
X	◪ 2Hz	At least 1 slave is not in DE.
not relevant: X		

Slave operation

DE (Data Exchange) green	BF (bus error) red	Description
☐	☐	Slave has no configuration.
☐	■	There is a bus error.
◪ 2Hz	☐	Slave exchanges data with the master. Slave CPU is in state STOP.
■	☐	Slave exchanges data with the master. Slave CPU is in state RUN.

LEDs EtherCAT interface X4

BF2 red	BS green	MT yellow	Meaning
☐	☐	☐	Master is in INIT state
☐	◪ 2Hz	☐	Master is in Pre-Op state
☐	◪ pulses	☐	Master is in Safe-Op state
☐	■	☐	Master is in OP state
X	X	☐	There is no maintenance event pending.
X	X	■	There is a maintenance event pending. More may be found in the diagnostics data
☐	X	X	There is no error on the EtherCAT bus pending
■	X	X	■ EtherCAT bus error, no connection to sub net ■ wrong transfer rate ■ Full-duplex-transmission is not activated
◪ 1Hz	X	X	■ Failure of a connected IO device ■ At least one IO device cannot be reached (topology mis-match) ■ Faulty configuration
◪ 4s on, 1s off	☐	◪ 4s on, 1s off	Error in configuration: ■ 0xEA64 was added to the diagnostics buffer ■ Additionally the SF LED of the CPU is on
◪ 4Hz	☐	◪ 4Hz	The alternate blinking indicates that a firmware update of the EtherCAT master is executed.
■	■	■	Firmware update of the EtherCAT master is finished without error.
not relevant: X			

L/A (Link/Activity) green	S (Speed) green	Meaning
■	X	The EtherCAT master is physically connected to the Ethernet.
□	X	There is no physical connection.
◪ flickers	X	Shows Ethernet activity of the EtherCAT master.
■	■	The Ethernet interface of the EtherCAT master has a transfer rate of 100Mbit.
■	□	The Ethernet interface of the EtherCAT master has a transfer rate of 10Mbit.
not relevant: X		

3.3 Technical data

Order no.	015-CEFNR00
Type	CPU 015N
Module ID	-
Technical data power supply	
Power supply (rated value)	DC 24 V
Power supply (permitted range)	DC 20.4...28.8 V
Reverse polarity protection	✓
Current consumption (no-load operation)	175 mA
Current consumption (rated value)	1.1 A
Inrush current	3 A
I^2t	0.1 A^2s
Max. current drain at backplane bus	3 A
Max. current drain load supply	10 A
Power loss	8 W
Load and working memory	
Load memory, integrated	512 KB
Load memory, maximum	512 KB
Work memory, integrated	256 KB
Work memory, maximal	512 KB
Memory divided in 50% program / 50% data	✓
Memory card slot	SD/MMC-Card with max. 2 GB
Hardware configuration	
Racks, max.	1
Modules per rack, max.	64
Number of integrated DP master	-

Order no.	015-CEFNR00
Number of DP master via CP	-
Operable function modules	64
Operable communication modules PtP	64
Operable communication modules LAN	-
Command processing times	
Bit instructions, min.	0.01 µs
Word instruction, min.	0.01 µs
Double integer arithmetic, min.	0.01 µs
Floating-point arithmetic, min.	0.06 µs
Timers/Counters and their retentive characteristics	
Number of S7 counters	512
S7 counter remanence	adjustable 0 up to 512
S7 counter remanence adjustable	C0 .. C7
Number of S7 times	512
S7 times remanence	adjustable 0 up to 512
S7 times remanence adjustable	not retentive
Data range and retentive characteristic	
Number of flags	8192 Byte
Bit memories retentive characteristic adjustable	adjustable 0 up to 8192
Bit memories retentive characteristic preset	MB0 .. MB15
Number of data blocks	4096
Max. data blocks size	64 KB
Number range DBs	1 ... 8191
Max. local data size per execution level	4096 Byte
Max. local data size per block	4096 Byte
Blocks	
Number of OBs	24
Maximum OB size	64 KB
Total number DBs, FBs, FCs	4096
Number of FBs	4096
Maximum FB size	64 KB
Number range FBs	0 ... 8191
Number of FCs	4096
Maximum FC size	64 KB
Number range FCs	0 ... 8191
Maximum nesting depth per priority class	16

Order no.	015-CEFNR00
Maximum nesting depth additional within an error OB	4
Time	
Real-time clock buffered	✓
Clock buffered period (min.)	30 d
Type of buffering	Goldcap
Load time for 50% buffering period	15 min
Load time for 100% buffering period	1 h
Accuracy (max. deviation per day)	10 s
Number of operating hours counter	8
Clock synchronization	✓
Synchronization via MPI	Master/Slave
Synchronization via Ethernet (NTP)	Slave
Address areas (I/O)	
Input I/O address area	2048 Byte
Output I/O address area	2048 Byte
Process image adjustable	✓
Input process image preset	128 Byte
Output process image preset	128 Byte
Input process image maximal	2048 Byte
Output process image maximal	2048 Byte
Digital inputs	16384
Digital outputs	16384
Digital inputs central	512
Digital outputs central	512
Integrated digital inputs	-
Integrated digital outputs	-
Analog inputs	1024
Analog outputs	1024
Analog inputs, central	256
Analog outputs, central	256
Integrated analog inputs	-
Integrated analog outputs	-
Communication functions	
PG/OP channel	✓
Global data communication	✓
Number of GD circuits, max.	8

Order no.	015-CEFNR00
Size of GD packets, max.	22 Byte
S7 basic communication	✓
S7 basic communication, user data per job	76 Byte
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
S7 communication, user data per job	160 Byte
Number of connections, max.	32
Functionality Sub-D interfaces	
Type	X2
Type of interface	RS485
Connector	Sub-D, 9-pin, female
Electrically isolated	✓
MPI	✓
MP²I (MPI/RS232)	-
DP master	-
DP slave	-
Point-to-point interface	✓
5V DC Power supply	max. 90mA, isolated
24V DC Power supply	max. 100mA, non-isolated
Type	X3
Type of interface	RS485
Connector	Sub-D, 9-pin, female
Electrically isolated	✓
MPI	✓
MP²I (MPI/RS232)	-
DP master	optional
DP slave	optional
Point-to-point interface	-
5V DC Power supply	max. 90mA, isolated
24V DC Power supply	max. 100mA, non-isolated
Functionality MPI	
Number of connections, max.	32
PG/OP channel	✓
Routing	✓

Order no.	015-CEFNR00
Global data communication	✓
S7 basic communication	✓
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
Transmission speed, min.	19.2 kbit/s
Transmission speed, max.	12 Mbit/s
Functionality PROFIBUS master	
PG/OP channel	✓
Routing	✓
S7 basic communication	✓
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
Activation/deactivation of DP slaves	-
Direct data exchange (slave-to-slave communication)	-
DPV1	✓
Transmission speed, min.	9.6 kbit/s
Transmission speed, max.	12 Mbit/s
Number of DP slaves, max.	124
Address range inputs, max.	2 KB
Address range outputs, max.	2 KB
User data inputs per slave, max.	244 Byte
User data outputs per slave, max.	244 Byte
Functionality PROFIBUS slave	
PG/OP channel	✓
Routing	✓
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
Direct data exchange (slave-to-slave communication)	-
DPV1	✓
Transmission speed, min.	9.6 kbit/s
Transmission speed, max.	12 Mbit/s
Automatic detection of transmission speed	-
Transfer memory inputs, max.	244 Byte

Order no.	015-CEFNR00
Transfer memory outputs, max.	244 Byte
Address areas, max.	32
User data per address area, max.	32 Byte
Point-to-point communication	
PtP communication	✓
Interface isolated	✓
RS232 interface	-
RS422 interface	-
RS485 interface	✓
Connector	Sub-D, 9-pin, female
Transmission speed, min.	150 bit/s
Transmission speed, max.	115.5 kbit/s
Cable length, max.	500 m
Point-to-point protocol	
ASCII protocol	✓
STX/ETX protocol	✓
3964(R) protocol	✓
RK512 protocol	-
USS master protocol	✓
Modbus master protocol	✓
Modbus slave protocol	✓
Special protocols	-
Functionality RJ45 interfaces	
Type	X1
Type of interface	Ethernet 10/100 MBit Switch
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓
Number of connections, max.	4
Productive connections	-
Fieldbus	-
Type	X5
Type of interface	Ethernet 10/100 MBit Switch
Connector	RJ45
Electrically isolated	✓

Order no.	015-CEFNR00
PG/OP channel	✓
Number of connections, max.	4
Productive connections	-
Fieldbus	-
Type	X4
Type of interface	Ethernet 100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	-
Number of connections, max.	-
Productive connections	-
Type	X6
Type of interface	Ethernet 10/100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓
Number of connections, max.	8
Productive connections	✓
Ethernet communication CP	
Number of configurable connections, max.	8
Number of productive connections by Siemens NetPro, max.	8
S7 connections	BSEND, BRCV, GET, PUT, Connection of active and passive data handling
User data per S7 connection, max.	32 KB
TCP-connections	FETCH PASSIV, WRITE PASSIV, Connection of passive data handling
User data per TCP connection, max.	64 KB
ISO-connections	-
User data per ISO connection, max.	-
ISO on TCP connections (RFC 1006)	FETCH PASSIV, WRITE PASSIV, Connection of passive data handling
User data per ISO on TCP connection, max.	32 KB
UDP-connections	-
User data per UDP connection, max.	-
UDP-multicast-connections	-

Order no.	015-CEFNR00
UDP-broadcast-connections	-
Ethernet open communication	
Number of connections, max.	8
ISO on TCP connections (RFC 1006)	TSEND, TRCV, TCON, TDISCON
User data per ISO on TCP connection, max.	8 KB
TCP-Connections native	TSEND, TRCV, TCON, TDISCON
User data per native TCP connection, max.	8 KB
User data per ad hoc TCP connection, max.	1460 Byte
UDP-connections	TUSEND, TURCV
User data per UDP connection, max.	1472 Byte
Ethernet communication via PG/OP	
Number of productive connections via PG/OP, max.	-
Number of productive connections by Siemens NetPro, max.	-
S7 connections	-
User data per S7 connection, max.	-
TCP-connections	-
User data per TCP connection, max.	-
ISO on TCP connections (RFC 1006)	-
User data per ISO connection, max.	-
Ethernet open communication via PG/OP	
Number of configurable connections, max.	-
ISO on TCP connections (RFC 1006)	-
User data per ISO on TCP connection, max.	-
TCP-Connections native	-
User data per native TCP connection, max.	-
User data per ad hoc TCP connection, max.	-
UDP-connections	-
User data per UDP connection, max.	-
EtherCAT Master	
Number of EtherCAT-slaves	128
Update time	1 ms .. 512 ms
Address range inputs, max.	2 KB
Address range outputs, max.	2 KB
EoE support	✓
FoE support	✓

Order no.	015-CEFNR00
Distributed Clock support	✓
Hotconnect Slaves	✓
Isochronous mode	✓
Management & diagnosis	
Protocols	ICMP DCP
Web based diagnosis	-
NCM diagnosis	-
Housing	
Material	PPE / PPE GF10
Mounting	Profile rail 35 mm
Mechanical data	
Dimensions (WxHxD)	131.5 mm x 109 mm x 83 mm
Net weight	325 g
Weight including accessories	-
Gross weight	-
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL certification	in preparation
KC certification	in preparation

4 Deployment CPU 015-CEFNR00

4.1 Assembly

Information about assembly and cabling ↳ Chapter 2 'Basics and mounting' on page 10

4.2 Start-up behavior

Turn on power supply

- The CPU checks whether a project AUTOLOAD.WLD exists on the memory card. If so, an overall reset is executed and the project is automatically loaded from the memory card.
- The CPU checks whether a command file with the name VIPA_CMD.MMC exists on the memory card. If so the command file is loaded from the memory card and the commands are executed.
- After PowerON and CPU STOP the CPU checks if there is a *.pkg file (firmware file) on the memory card. If so, this is shown by the CPU by blinking LEDs and the firmware may be installed by an update request. ↳ *further information on page 84*
- The CPU checks if a previously activated VSC is inserted. If not, the SD LED gets on and a diagnostics entry is released. The CPU switches to STOP after 72 hours. With a just installed VSC activated functions remain activated. ↳ *Chapter 4.18 'Diagnostic entries' on page 92*

After this the CPU switches to the operating mode, which is set on the operating mode switch.

Delivery state

In the delivery state the CPU is overall reset. After a STOP→RUN transition the CPU switches to RUN without program.

4.3 Addressing

4.3.1 Overview

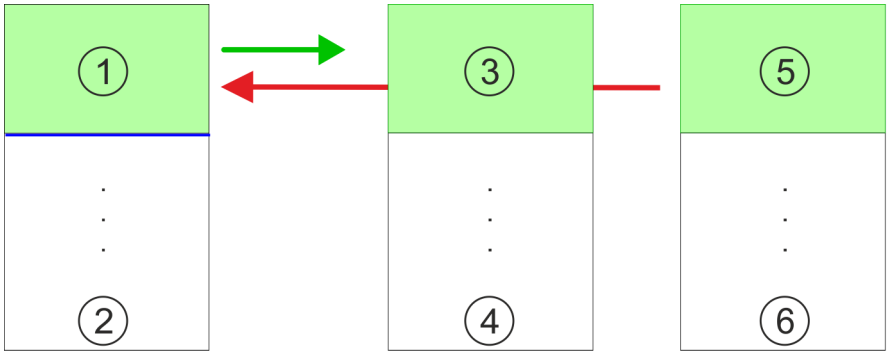
To provide specific addressing of the installed periphery modules, certain addresses must be allocated in the CPU. This address mapping is in the CPU as hardware configuration. If there is no hardware configuration, depending on the slot, the CPU assigns automatically periphery addresses for digital in-/output modules starting with 0 and analog modules are assigned to even addresses starting with 256.

4.3.2 Addressing backplane bus periphery

The CPU 015-CEFNR00 provides an periphery area (address 0 ... max. periphery address) and a process image of the in- and outputs (each address default 0 ... 127). The size of the process image can be preset via the parametrization. ↳ *'Cycle/Clock memory' on page 65*

The process image is updated automatically when a cycle has been completed. The process image is divided into two parts:

- Process image to the inputs (PII)
- Process image to the outputs (PIQ)



1 Periphery area: 0 ... 127 (default)
2 max periphery area
3 Process image of the inputs (PII): 0 ... 127
4 max. process image of the inputs (PII)
5 Process image of the outputs (PIQ): 0 ... 127
6 max. process image of the outputs (PIQ)

Max. number of pluggable modules

Up to 64 SLIO modules can be connected to a SLIO CPU. This sum includes power and clamp modules.

Define addresses by hardware configuration

You may access the modules with read res. write accesses to the periphery bytes or the process image. To define addresses a hardware configuration may be used. For this, click on the properties of the according module and set the wanted address.

Automatic addressing

If you do not like to use a hardware configuration, an automatic addressing is established. Here the address assignment follows the following specifications:

- Starting with slot 1, the central plugged modules are assigned with ascending logical addresses.
- The length of the memory area corresponds to the size of the process data of the according module. Information about the sizes of the process data can be found in the according manual of the module.
- The memory areas of the modules are assigned without gaps separately for input and output area.
- Digital modules are mapped starting at address 0 and all other modules are mapped starting from address 256. ETS modules are mapped starting from address 256.
- As soon as the mapping of digital modules exceeds the address 256, by regarding the order, these are mapped starting from address 256.

Example for automatic address allocation

Slot	Type	Description	Length	I address	O address
1	021-1BF00	DI 8x	1 Byte	0	
2	021-1BF00	DI 8x	1 Byte	1	
3	022-1BF00	DO 8x	1 Byte		0
4	031-1BB30	AI 2x	4 Byte	256...259	
5	032-1BB30	AO 2x	4 Byte		256...259
6	031-1BD40	AI 4x	8 Byte	260...267	
7	032-1BD40	AO 4x	8 Byte		260...267
8	022-1BF00	DO 8x	1 Byte		1
9	021-1BF00	DI 8x	1 Byte	2	

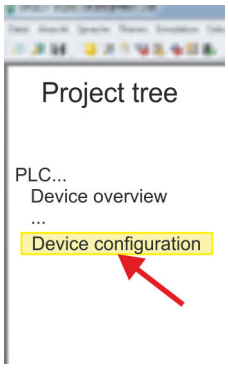
4.4 Hardware configuration - CPU

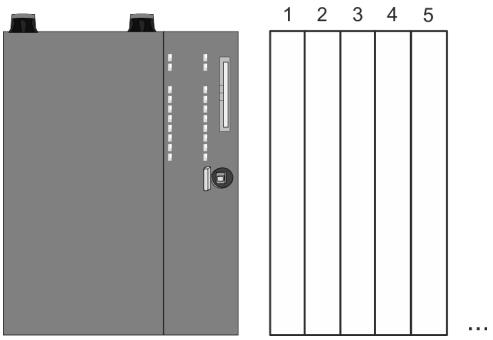
Precondition

For project engineering a thorough knowledge of the SPEED7 Studio is required!

Proceeding

1. Start the *SPEED7 Studio*.
2. Create a new project in the *Work area* with *'New project'*.
   ⇨ A new project is created and the view *'Devices and networking'* is shown.
3. Click in the *Project tree* at *'Add new device ...'*.
   ⇨ A dialog for device selection opens.
4. Select from the *'Device templates'* your CPU and click at [OK].
   ⇨ The CPU is inserted in *'Devices and networking'* and the *'Device configuration'* is opened.





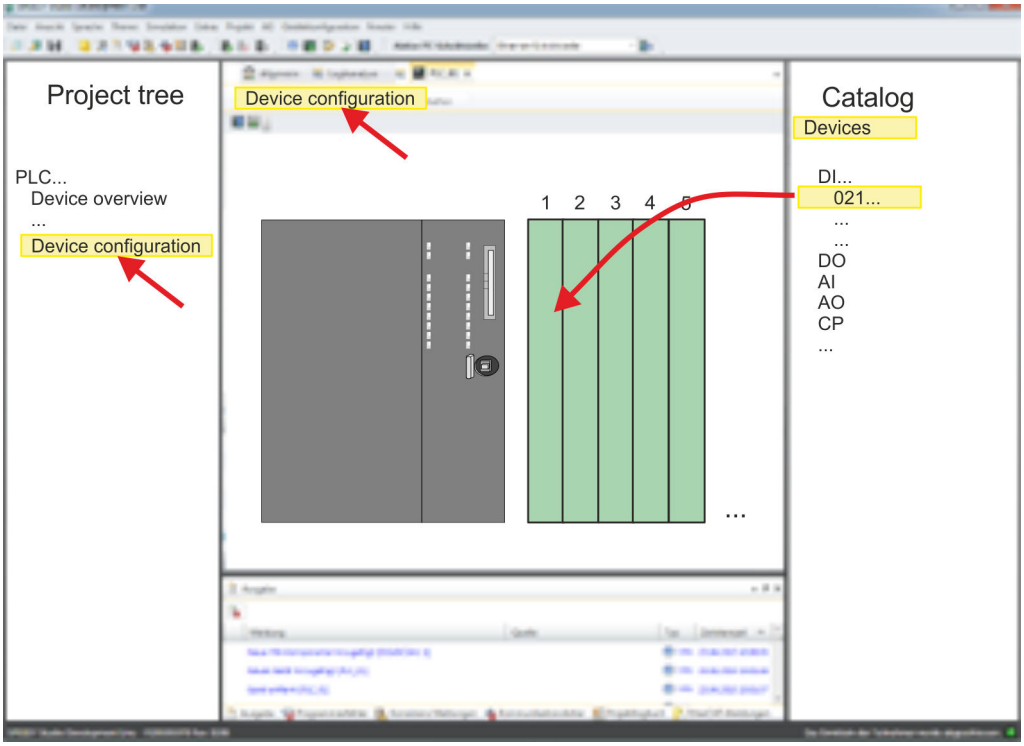
Device configuration

Slot	Module	...	...	...	...
0	CPU 015-CEFNR00				
-X1	PG_OP_Ethernet				
-X3	MPI interface				
...	...			...	

4.5 Hardware Configuration - I/O modules

Hardware configuration of the modules

1. Click in the *'Project tree'* at *'PLC... > Device configuration'*.
2. Starting with slot 1 place in the *'Device configuration'* your System SLIO modules in the plugged sequence. For this drag from the hardware catalog the corresponding module to the corresponding position in the *Device configuration*.



Parametrization

For parametrization double-click in the *'Device configuration'* on the module you want to parameterize. Then the parameters of the module are shown in a dialog. Here you can make your parameter settings.

Parametrization during runtime

By using the SFCs 55, 56 and 57 you may alter and transfer parameters for wanted modules during runtime. For this you have to store the module specific parameters in so called "record sets". More detailed information about the structure of the record sets is to find in the according module description.

4.6 Hardware configuration - Ethernet PG/OP channel

Overview

The CPU has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU.

- The Ethernet PG/OP channel (X1/X5) is designed as switch. This enables PG/OP communication via the connections X1 and X5.
- The Ethernet PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc.
- At the first commissioning respectively after a factory reset the Ethernet PG/OP channel has no IP address.
- For online access to the CPU via the Ethernet PG/OP channel, valid IP address parameters have to be assigned to this. This is called "initialization".
- This can be done with the *SPEED7 Studio*.

Assembly and commissioning

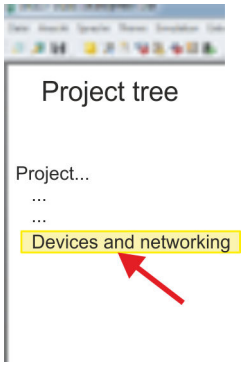
1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the one of the Ethernet jacks (X1, X5) of the Ethernet PG/OP channel to Ethernet.
4. Switch on the power supply.
   ⇨ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization"

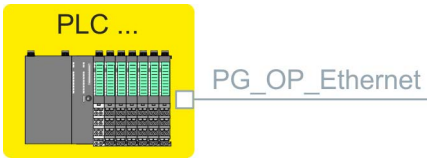
You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the *SPEED7 Studio*with the following proceeding:

MAC PG/OP: 00-20-D5-77-05-10

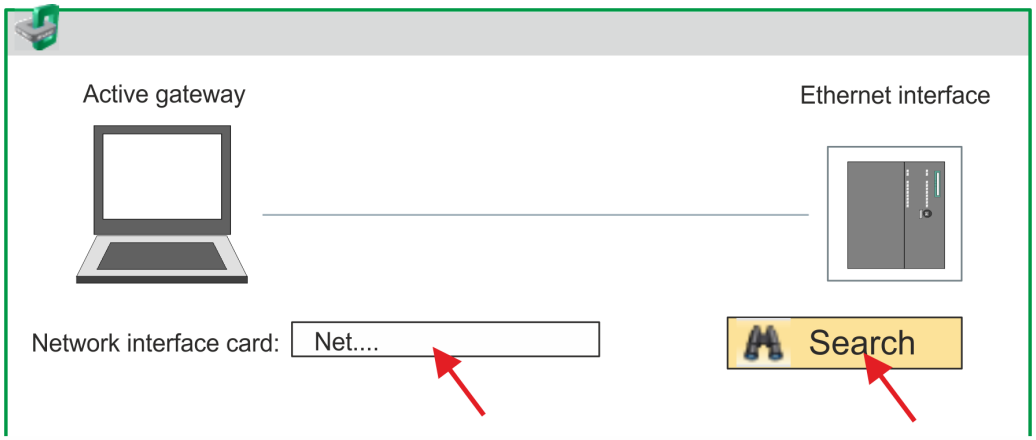
1. Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".
2. Start the *SPEED7 Studio* with your project.



3. Click in the *Project tree* at *'Devices and networking'*.
   ⇨ You will get a graphical object view of your CPU.



4. Click at the network *'PG_OP_Ethernet'*.
5. Select *'Context menu ➔ Determine accessible partner'*.
   ⇨ A dialog window opens.



6. Select the according network interface card, which is connected to the Ethernet PG/OP channel and click at *'Search'* to determine the via MAC address reachable device.
   ⇨ The network search is started and the found stations are listed in a table.

7.

	Devices...	IP...	MAC...	Device...	...	...
1		172.20. ..	00:20: ...	VIPA ...		
2		...	...	...		

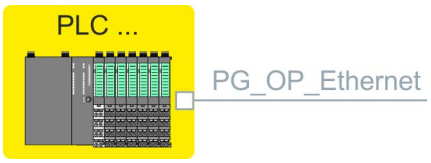
   Click in the list at the module with the known MAC address. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".

8. Click at *'Set IP address'*. Now set the IP configuration by entering *'IP address'*, *'Subnet mask'* and *'Gateway'*.
9. Click at *'Set IP address'*.
   ⇨ The IP address is transferred to the module and the list is refreshed. Directly after the assignment the Ethernet PG/OP channel is online reachable using the set IP address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or a factory reset is executed.
10. With clicking at *'Apply settings'* the IP address data a stored in the project.

Take IP address parameters in project

If you are not online, you can assign IP address data to your Ethernet PG/OP channel with following proceeding:

1. Start the *SPEED7 Studio* with your project.
2. Click in the *Project tree* at *'Devices and networking'*.
   ⇨ You will get a graphical object view of your CPU.



3. Click at the network *'PG_OP_Ethernet'*.
4. Select *'Context menu ➔ Interface properties'*.
   ⇨ A dialog window opens. Here you can enter the IP address data for your Ethernet PG/OP channel.
5. Confirm with [OK].
   ⇨ The IP address data are stored in your project listed in *'Devices and networking'* at *'Local components'*.

   After transferring your project your CPU can be accessed via Ethernet PG/OP channel with the set IP address data.

Local components

Slot	Module	...	...	...IP address	...
0	CPU ...			...	
-X1	PG_OP_Ethernet			172.20.120.40	
-X3	MPI interface			...	
...	...			...	

4.7 Setting CPU parameters

Proceeding

1. Click in the *'Project tree'* at *'PLC... > Device configuration'*.
2. Click at the CPU and select *'Context menu ➔ Components properties'*.
   ⇨ The properties dialog is opened. Here you can adjust all the CPU parameters.

Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.

4.7.1 Parameter CPU

General

Here you can make general settings for the current CPU.

- Name
  - Name of the PLC. This name is shown in the project tree.
- Plant designation
  - Here is the possibility to specify a plant designation for the CPU.
  - This plant designation identifies parts of the plant according to their function.
  - Its structure is hierarchic according to IEC 1346-1.
- Location designation
  - The location designation is part of the resource designation.
  - Here the exact location of your module within a plant may be specified.
- MPI data
  - Here you can adjust the setting of the MPI subnet (Multi Point Interface) for serial connection between the MPI participants.
  - Address: Here you can enter the MPI address. Standard setting of the VIPA CPUs is MPI address 2. The address 0 is reserved for programming devices.
  - Highest address: By specifying the highest address number you can limit the range of addresses.
  - Secondary transmission speed MPI The transmission rate (Bit/s) of the MPI subnet must not be higher than the transmission rate of the slowest accessible MPI partner.

Feature set

Clicking at *'EtherCAT master functionality ...'* the following additional functions can be activated:

- Isochronous mode with enabling OB 60 and OB 61
- EtherCAT master functionality
- Memory expansion to 512kB work respectively load memory

With *'inactive'* the additional functions can be disabled.

Please consider the additional functions in the SPEED7 Studio can only be activated, if you have valid license for these functions!

Start-up

Here you can make general configurations for the start-up behaviour of the current CPU.

- Start-up if present configuration does not match actual configuration
  - The expected configuration is the configuration of the components which is defined in the project and uploaded to the CPU.
  - The actual configuration is the implemented configuration of the components.
  - If this option is deselected, the CPU remains in the STOP mode for the following cases:

    One or more components are not located in the configured slot.

    A component of another type is located in the configured slot.
  - If this option is selected, the CPU switches to the RUN mode even if the components are not located in the configured slots or if components of another type are located there.
- Deleting PAA after warm restart
  - If this option is selected, the process image of the outputs (PAA) is deleted after the warm restart of the CPU.

- Disable hot restart by operator
  - The types of start-up are restricted when triggered by the operation or communication job.
  - If this option is selected, only restart or cold start are possible. Warm restart is not possible.
  - If this option is deselected, all types of start-up are possible.
- Start-up after Power On
  - Here you can select whether a restart, warm restart or cold start should be made after having activated the power supply (powerON).
  - Cold start: All variables and memory blocks are initialised.
  - Restart (warm start) The non-retentive memory areas are initialised, the retentive memory areas are restored.
  - Warm restart: The user program is continued where it has been interrupted.
- Monitoring time for ...
  - The time base of the following parameters is 100 milliseconds. Multiply the entered value with the time base. Example: Entered value 650 * 100 ms = 65.000 ms of monitoring time
  - Finished message from modules (100 ms): Maximum duration of the Ready signal of all configured components after having switched on the power supply (powerON).
  - Transfer of parameters to modules (100 ms) Maximum duration of the parameter transfer to the parameterizable components.
  - Hot restart (100 ms) Maximum duration of the warm restart. If the time between powerOFF and PowerON or between STOP mode and RUN mode is longer than the time entered here, there is no warm restart. The CPU remains in STOP mode.

Cycle/Clock memory

Here you can make general configurations for the start-up behaviour of the current CPU.

- Refresh process image cyclically
  - If this option is selected, the process image of the organisation block OB 1 is cyclically updated. This expands the cycle time.
- Scan cycle monitoring time (ms)
  - Here you can enter the scan cycle monitoring time in ms.
  - If the run time of the user program exceeds the scan cycle monitoring time, the CPU switches to STOP mode.
  - Reasons for time-out:
    - Communication principle
    - Accumulation of alarm events
    - Error in the CPU program
- Minimum scan cycle time (ms)
  - Guaranteed compliance with a minimum scan cycle time: The start of a new cycle is delayed until the minimum cycle time has been reached.
- Scan cycle load from communication (%)
  - Percentage of communication processes compared to the complete cycle time.
  - With this parameter you can control the duration of communication processes within certain limits, which always extend the cycle time.
  - If e.g. set to 50%, the cycle time might double. In addition, the OB 1 cycle will be extended by asynchronous events (e.g. process interrupts).
- OB 85 calling at periphery access errors
  - Reaction of the CPU after periphery access errors during the update of the process image.
  - The VIPA CPU is preset such that on a periphery access error the OB 85 is not called and there is no entry made to the diagnostic buffer.

- Size of the process-image input/output area
  - Here you can define the size of the process image max. 2048 for input and output periphery (default: 128).
- Clock memory
  - Clock memory Enable this option, if the CPU should provide clock memories. Clock memories periodically change their value in pre-set intervals.
  - Memory byte: Number of the memory byte for the clock memory. The memory byte is used only if you select the *'Clock memory'* option.

The selected memory byte cannot be used for temporary data storage.

Retentive memory

In order to maintain data in case of power failure, certain data areas can be marked as retentive. A restart (warm start) will restore the values of the retentive memory areas from the last program cycle.

- Number of memory bytes starting with MB0
  - Here you can enter the number of retentive memory bytes starting from memory byte 0. Example: Example: Input value 16 = memory bytes 0 to 15 are retentive.
- Number of timers starting with T0
  - Here you can enter the number of retentive timers starting from T0.
- Number of counters starting with Z0
  - Here you can enter the number of retentive counter starting from Z0.
- Areas
  - You can define up to 8 retentive memory areas in the data blocks:
  - DB no.: Number of the retentive data block
  - Byte address: Starting address within the retentive data block
  - Number of bytes: Number of retentive bytes from the starting address within the data block.

Interrupts

Here you can define the order for processing the individual interrupt organisation blocks. OBs with the smallest number have lowest priority. OBs with priority 0 are not processed.

- Priority: The following interrupt OBs are listed:
  - OB 40 - OB 47: Hardware interrupts
  - OB 20 - OB 23: Time delay interrupts
  - OB 50, OB 51, OB 55 - OB 57: Communication interrupts
  - OB 81 - OB 87: Async. error interrupts

Time of day interrupts

The time of day interrupt organisation blocks OB10 to OB17 can interrupt the processing of OB1 once or at a certain interval. Depending on the CPU used, you can parameterize up to 8 time of day interrupts.

- Priority
  - Order in which a time of day interrupt organisation block is processed.
  - OBs with the smallest number have lowest priority.
  - OBs with priority 0 are not processed.
- Active
  - By selecting *'Active'* the time of day interrupt is activated.

- Execution
  - Here you can select the execution of interrupts once or at certain intervals.
  - The intervals from every minute to yearly are related to the settings at *Start date* and *Time*.
- Start date/time
  - Here you can define the time of the initial execution of the time of day interrupt.

Cyclic interrupts

The cyclic interrupt organisation blocks OB 30 to OB 38 can interrupt the processing of OB1 once or at a certain interval. Depending on the CPU used, you can parameterize up to 9 cyclic interrupts.

- Priority
  - Order in which a cyclic interrupt organisation block is processed.
  - OBs with the smallest number have lowest priority.
  - OBs with priority 0 are not processed.
- Execution
  - Specify the time intervals in ms, in which the watchdog interrupt OBs should be processed.
  - The starting time is the switching from STOP mode to RUN mode.
- Phase shift
  - Time in milliseconds by which the execution time of the cyclic interrupt is to be delayed.
  - By selecting several cyclic interrupts, you can use the phase offset to make sure that the cyclic interrupts do not start at the same time.

Diagnostics/Clock

Here you can define, which clock is to be synchronised with which clock.

- Extended functions
  - The expanded range of functions for diagnostics is not supported.
- Report cause of STOP
  - Activate this parameter, if on transition to STOP the CPU has to report a cause for the STOP to PG respectively OP.
- Report to process control active
  - Currently this function is not supported.
- Clock
  - Here you can define, which clock is to be synchronised with another clock.
- Synchronization type
  - Here you specify whether the clock synchronizes other clocks or not.
  - As slave: The clock is synchronized by another clock.
  - As master: The clock synchronizes other clocks as master.
  - none: There is no synchronization.
- Time interval
  - Time intervals within which the synchronization should take place.
- Correction factor
  - By specifying a correction factor in ms, you can compensate the deviation of the clock within 24 hours.
  - If the clock runs slow by 1 second within 24 hours, you can adjust this deviation with the correction factor "+1000".

Protection

- Protection Level
  - Here you can configure a protection level to protect the CPU against unauthorised access.
  - *No protection* (default):
    No password adjustable; no restrictions
  - *Write-protection* with password:
    Known password: Read and write access
    Unknown password: Read access only.
  - *Read/write protection* with password:
    Known password: Read and write access
    Unknown password: No read and write access
- Password
  - Here you can specify a password for the write and read protection.
  - Depending on the setting of the protection level for read respectively write access a password is requested.

Extended configurations

Here you can adjust the functionality of the interface and specify the number of flags, timers and counters:

- Function X2
  - Function PtP(MPI) interface X2
  - PtP (default): With this operating mode the RS485 interface acts as an interface for serial point-to-point communication. Here data may be exchanged between two stations by means of protocols.
  - MPI: With this operating mode the interface serves for the connection between programming unit and CPU via MPI. By means of this e.g. the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU.
- MPI address X2
  - With *MPI* you can specify the MPI address here. With *PTP* this parameter is ignored by the CPU.
  - Range of values: 2 (default) ... 31
- MPI Baud rate X2:
  - With *MPI* you can specify the MPI transmission rate here. With *PTP* this parameter is ignored by the CPU.
  - Range of values: 19.2kB/s ... 12MB/s, default: 187.5kB/s
- Additional retentive memory
  - Here enter the number of retentive memory bytes. With 0 the preset value of *'Number of memory bytes starting with MB0'* of *'Retentive memory'* is taken.
  - Range of values: 0 (default) ... 8192
- Additional retentive timer
  - Here enter the number of timer. With 0 the preset value of *'Number of timer starting with T0'* of *'Retentive memory'* is taken.
  - Range of values: 0 (default) ... 512
- Additional retentive counter
  - Enter here the number of counter. With 0 the preset value of *'Number of counter starting with Z0'* of *'Retentive memory'* is taken.
  - Range of values: 0 (default) ... 512

4.7.2 Parameter MPI interface

Via double-click at *'MPI interface'* of *'Device configuration'* the properties dialog of the MPI interface is opened.

- MPI data
  - Here you can adjust the setting of the MPI subnet (Multi Point Interface) for serial connection between the MPI participants.
  - Address: Here you can enter the MPI address. Standard setting of the VIPA CPUs is MPI address 2. The address 0 is reserved for programming devices.
  - Highest address: By specifying the highest address number you can limit the range of addresses.
  - Secondary transmission speed MPI: The transmission rate (Bit/s) of the MPI subnet must not be higher than the transmission rate of the slowest accessible MPI partner.

4.7.3 Parameter Ethernet

Via double-click at *'Ethernet'* of *'Device configuration'* the properties dialog of the NET CP interface (X6) is opened.

- General
  - General: Here you can enter a name for your station.
  - Subnet ID The subnet ID is used to uniquely identify your network.
- IP configuration
  - IP address: Here you can assign an IP address to your NET CP.
  - Subnet: Here you can enter the subnet mask of your network.

4.8 Project transfer

Overview

There are the following possibilities for project transfer into the CPU:

- Transfer via MPI
- Transfer via Ethernet
- Transfer via memory card

4.8.1 Transfer via MPI

General

For transfer via MPI the CPU has the following interface:

'X3: MPI(PB) interface' on page 42

'X2: PtP(MPI) interface' on page 42

With an overall reset CPU the configuration via X2 PtP(MPI) is not possible!

Net structure

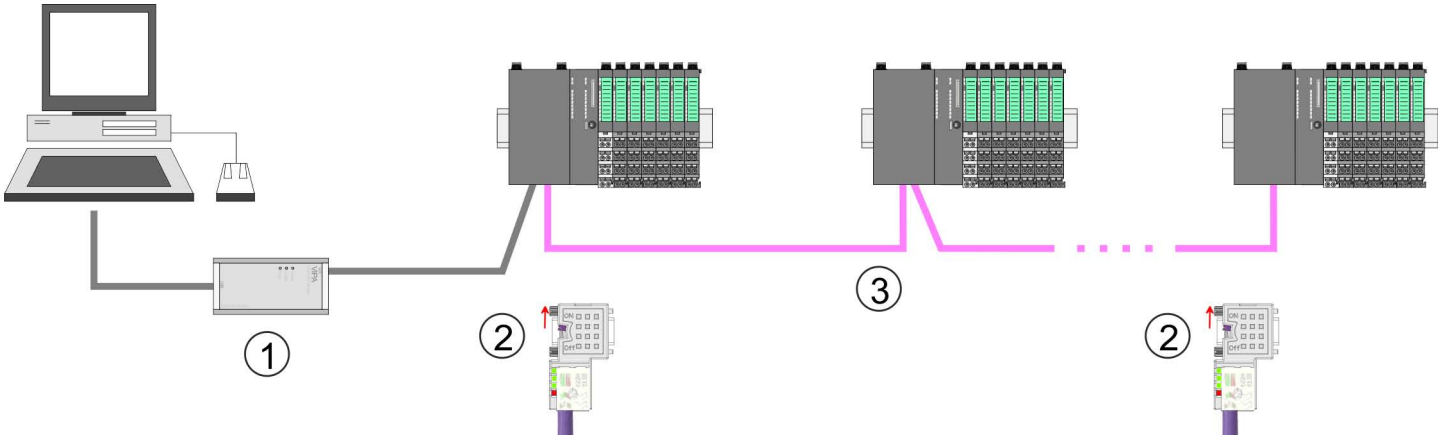
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs with 187.5kbaud. VIPA CPUs are delivered with MPI address 2.

MPI programming cable

The MPI programming cables are available at VIPA in different variants. The cables provide a RS232 res. USB plug for the PC and a bus enabled RS485 plug for the CPU. Due to the RS485 connection you may plug the MPI programming cables directly to an already plugged plug on the RS485 jack. Every bus participant identifies itself at the bus with an unique address, in the course of the address 0 is reserved for programming devices.

Terminating resistor

A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment. Please make sure that the participants with the activated terminating resistors are always power supplied. Otherwise it may cause interferences on the bus.

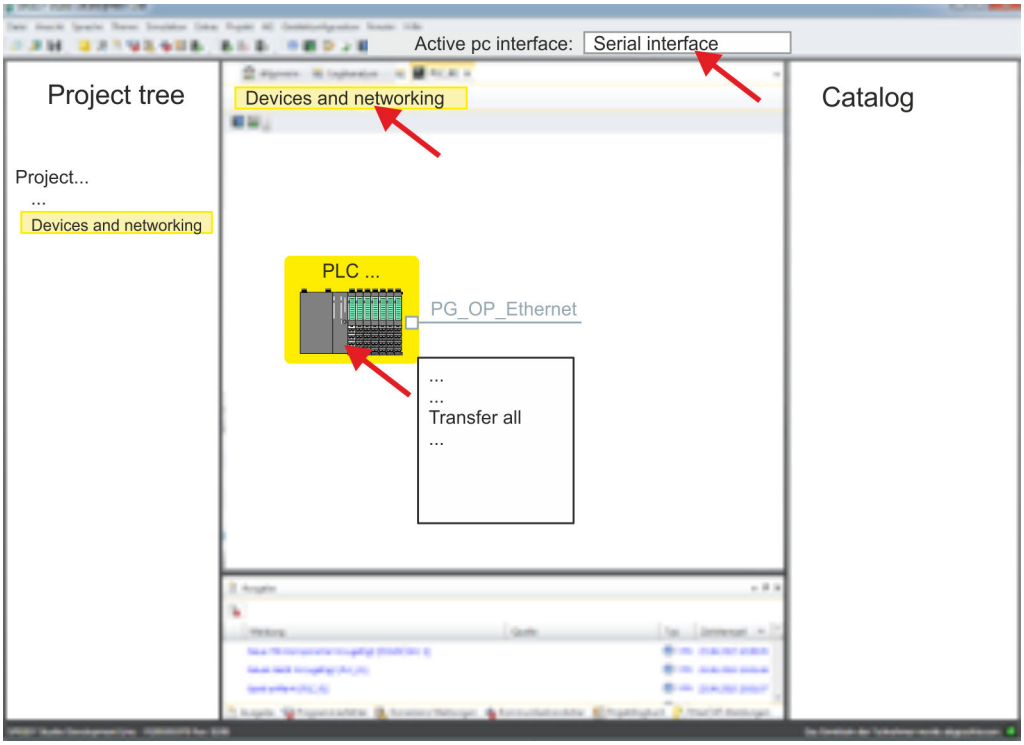


1 MPI programming cable
2 Activate the terminating resistor via switch
3 MPI network

Proceeding transfer via MPI

1. Connect your PC to the MPI jack of your CPU via a MPI programming cable.
2. Switch-ON the power supply of your CPU and start the *SPEED7 Studio* with your project.
3. Set at *'Active PC interface'* the "Serial interface".
4. Click in the *'Project tree'* to your project and select *'Context menu ➔ Recompile'*.

   ⇨ Your project will be translated and prepared for transmission.



5. To transfer the user program and hardware configuration click in the *Project tree* at your CPU and select *'Context menu ➔ Transfer all'*.

   ⇨ A dialog window for project transfer opens

6. Select the *'Port type'* "Serial interface" and start the transfer with *'Transfer'*.
7. Confirm the request that the CPU is to be brought into the state STOP.

   ⇨ The user program and the hardware configuration are transferred via MPI to the CPU.

8. Close after transmission the dialog.
9. With *'Context menu ➔ Copy RAM to ROM'* you can save your project on a memory card, if one is plugged.

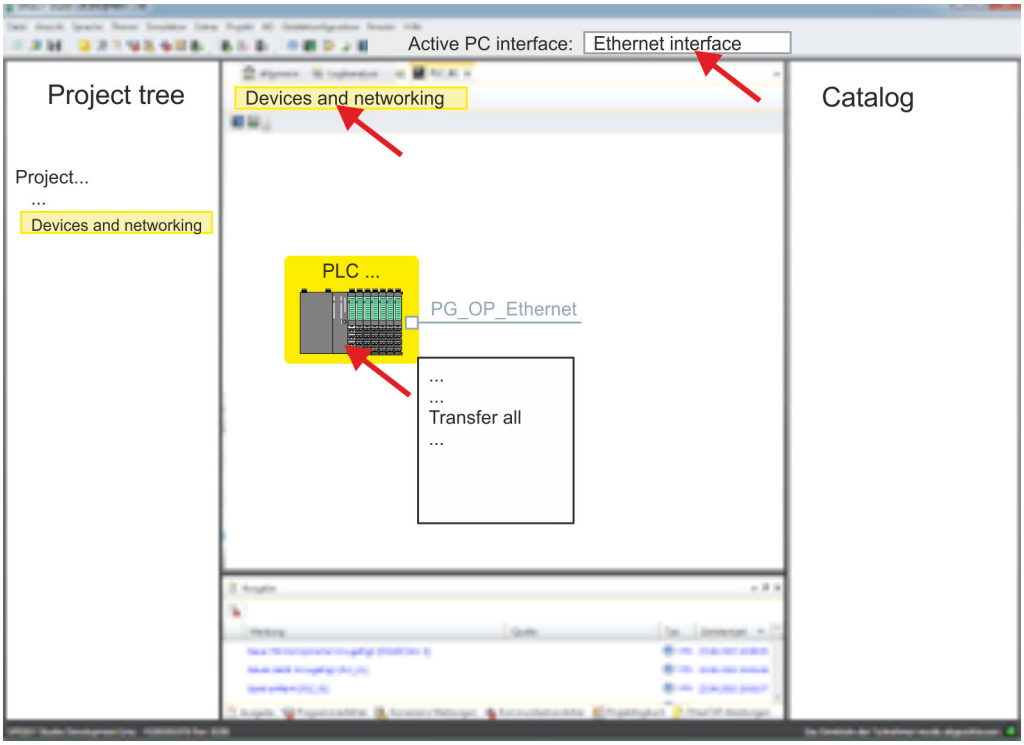
4.8.2 Transfer via Ethernet

Proceeding transfer via Ethernet

For transfer via Ethernet the CPU has an Ethernet PG/OP channel. For online access to this, you have to assign IP address parameters to this by means of "initialization" and transfer them into your project. For the transfer, connect, if not already done, the Ethernet PG/OP channel jack to your Ethernet. The connection happens via an integrated 2-port switch (X1, X5).

1. Switch-ON the power supply of your CPU and start the *SPEED7 Studio* with your project.
2. Set at *'Active PC interface'* the "Ethernet interface".
3. Click in the *'Project tree'* to your project and select *'Context menu ➔ Recompile'*.

   ⇨ Your project will be translated and prepared for transmission.



4. To transfer the user program and hardware configuration click in the *Project tree* at your CPU and select *'Context menu ➔ Transfer all'*.
   ⇨ A dialog window for project transfer opens
5. Select the *'Port type'* "Ethernet interface" and start the transfer with *'Transfer'*.
6. Confirm the request that the CPU is to be brought into the state STOP.
   ⇨ The user program and the hardware configuration are transferred via Ethernet to the CPU.
7. Close after transmission the dialog.
8. With *'Context menu ➔ Copy RAM to ROM'* you can save your project on a memory card, if one is plugged.

4.8.3 Transfer via memory card

Proceeding transfer via memory card

The memory card serves as external storage medium. There may be stored several projects and sub-directories on a memory card. Please regard that your current project is stored in the root directory and has one of the following file names:

- S7PROG.WLD
- AUTOLOAD.WLD

1. Start the *SPEED7 Studio* with your project.
2. Click in the *'Project tree'* at the CPU.
3. Create in the *SPEED7 Studio* with *'Context menu ➔ Export device configuration (WLD)'* a wld file.
   ⇨ The wld file is created. This contains the user program and the hardware configuration

4. Copy the wld file at a suited memory card. Plug this into your CPU and start it again.

   ⇨ The transfer of the application program from the memory card into the CPU takes place depending on the file name after an overall reset or PowerON.

   S7PROG.WLD is read from the memory card after overall reset.

   AUTOLOAD.WLD is read from the memory card after PowerON.

   The blinking of the SD LED of the CPU marks the active transfer. Please regard that your user memory serves for enough space for your user program, otherwise your user program is not completely loaded and the SF LED gets on.

4.9 Accessing the web server

4.9.1 Access via the Ethernet PG/OP channel

There is a web server, which can be accessed via the IP address of the Ethernet PG/OP channel with an Internet browser. At the web page information about the CPU and its connected modules can be found. *Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61*

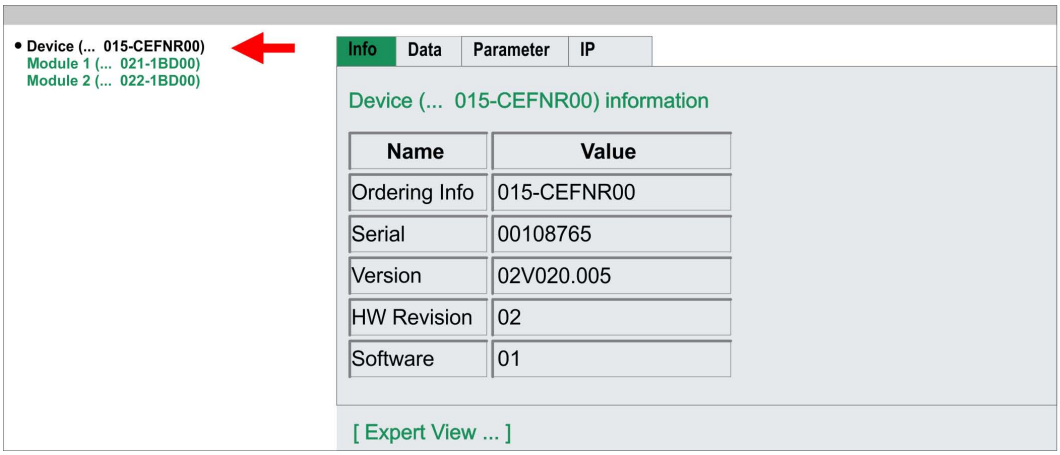
It is assumed that there is a connection between PC and CPU with Internet browser via the Ethernet PG/OP channel. This may be tested by Ping to the IP address of the Ethernet PG/OP channel.

4.9.2 Structure of the web page

The web page is built dynamically and depends on the number of modules, which are connected to the CPU. The web page only shows information. The shown values cannot be changed

Please consider the System SLIO power and clamp modules do not have any module ID. These may not be recognized by the CPU and so are not listed and considered during slot allocation.

4.9.3 Web page with selected CPU



Info - Overview

Here order number, serial number and the version of firmware and hardware of the CPU are listed. [Expert View] takes you to the advanced "Expert View".

Expert View

Runtime Information		CPU
Operation Mode	STOP_INTERNAL	Operating mode
Mode Switch	STOP	
System Time	28.03.17 16:09:31:262	Date, time
Up Time	0 days 02 hrs 07 min 08 sec	Time of changing the operating modes
Last Change to RUN	n/a	
Last Change to STOP	28.03.17 16:09:03:494	
OB1-Cycle Time	cur = 0us, min = 0us, max = 0us, avg = 0us	Cyclic time: min = minimum cur = current max = maximum avg = average

Interface Information			Interfaces
X1	PG/OP Ethernet Port 1	Address 2025..2040	Ethernet PG/OP channel port 1
X2	PTP		PtP: point-to-point operation or DPM: DP master operation
X3	MPI	Address 2047	Operating mode RS485 MPI: MPI operation
X4	EtherCAT Port	Address 2045	EtherCAT interface
X5	PG/OP Ethernet Port 2	Address 2025..2040	Ethernet PG/OP channel port 2
X6	Ethernet Port	Address 2044	CP interface

Card Information		Memory card
TYPE	SD	Memory card information
Manufacturer ID	9	
Application ID	16720	
Card Name	AF SD	
Card Revision	16	
Card S/N	79C74010	
Manufacture Month	8	
Manufacture Year	2013	
Size	470.75 MByte	
Free	468.98 MByte	

VSC information		VSC
VSC Product Number	955-C0ME040	VSC information
VSC Product S/N	00007807	
Memory Extension	256 kByte	
Profibus	not activated	
Motion	4 Axes	

Active Feature Set Information		VSC
Status	Media present	Unlocked features information
VSC Product Number	955-C000M40	
VSC Product S/N	00002650	
Memory Extension	256 kByte	
Profibus	not activated	
Motion	4 Axes	

Memory Usage				CPU
	free	used	max	Memory configuration information
LoadMem	512.0 kByte	0 byte	512.0 kByte	Load memory, working memory (code/data)
WorkMemCode	256.0 kByte	0 byte	256.0 kByte	
WorkMemData	256.0 kByte	0 byte	256.0 kByte	

PG/OP Network Information		Ethernet PG/OP channel
Device Name	Onboard PG/OP	Name
IP Address	172.20.139.76	Address information
Subnet Mask	255.255.255.0	
Gateway Address	172.20.139.76	
MAC Address	00:20:D5:02:6C:27	
Link Mode X1	100 Mbps - Full Duplex	Link status and speed
Link Mode X5	Not Available	

CP Network Information (According To Project Settings)		
Device Name	n/a	Adressangaben
IP Address	0.0.0.0	Name, Firmware-Version, Package
Subnet Mask	0.0.0.0	Angaben für den Support
Gateway Address	0.0.0.0	Diagnose-Adresse

CP Firmware Information		CP:
Bx000689	V3.0.0.32	Support information
PRODUCT	VIPA EtherCAT-CP V3.1.3 Px000249.pkg	Name, firmware version, package

CP Firmware Information		CP:
ExtSvnRev.txt	V128.0.0.0	Support information
MX000337	V0.0.1.0	
Diagnosis Address	2046	Diagnostic address

CPU firmware information		CPU
File System	V1.0.2	Name, firmware version, package
PRODUCT	VIPA 015-CEFNR00 V1.5.9 Px000310.pkg	
HARDWARE	V0.2.0.0 5841L-V10 MX000336.004	Support information
BOOTLOADER	Bx000645 V125	
Bx000501	V2.3.0.255	
Ax000136	V1.0.6.0	
Ax000150	V1.1.4.0	
fx000018.wld	V1.0.2.0	
syslibex.wld	n/a	
Protect.wld	n/a	

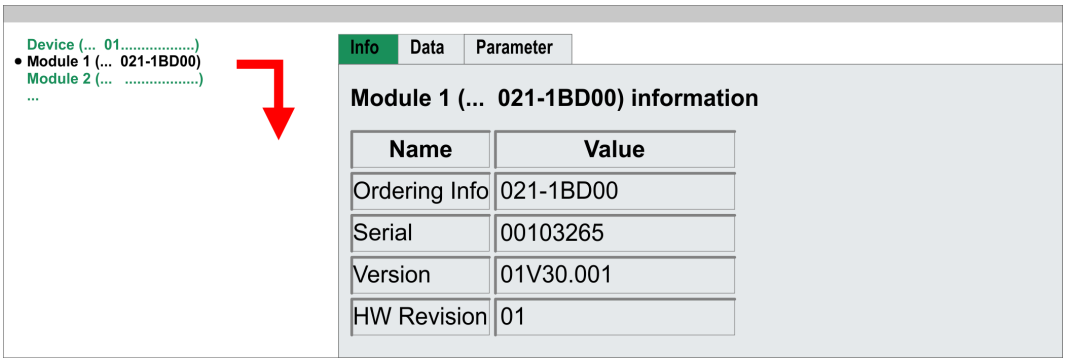
ARM Processor Load		CPU
Measurement Cycle Time	100 ms	Support information
Last Value	5%	
Average Of Last 10 Values	5%	
Minimum Load	5%	
Maximum Load	16%	

Data Currently nothing is displayed here.

Parameter Currently nothing is displayed here.

IP Here the IP address data of your Ethernet PG/OP channel are shown

4.9.4 Web page with selected module



Info

Here product name, order number, serial number, firmware version and hardware state number of the according module are listed.

Data

Here the address and the state of the inputs respectively outputs are listed. Please note with the outputs that here exclusively the states of outputs can be shown, which are within the OB 1 process image.

Parameter

With parameterizable modules e.g. analog modules the parameter setting is shown here. These come from the hardware configuration.

4.10 Operating modes

4.10.1 Overview

The CPU can be in one of 4 operating modes:

- Operating mode STOP
- Operating mode START-UP
  (OB 100 - restart / OB 102 - cold start *)
- Operating mode RUN
- Operating mode HOLD

Certain conditions in the operating modes START-UP and RUN require a specific reaction from the system program. In this case the application interface is often provided by a call to an organization block that was included specifically for this event.

Operating mode STOP

- The application program is not processed.
- If there has been a processing before, the values of counters, timers, flags and the process image are retained during the transition to the STOP mode.
- Command output disable (BASP) is activated this means the all digital outputs are disabled.
- RUN-LED off
- STOP-LED on

Operating mode START-UP

- During the transition from STOP to RUN a call is issued to the start-up organization block OB 100.
  - The processing time for this OB is not monitored.
  - The START-UP OB may issue calls to other blocks.
- All digital outputs are disabled during the START-UP, this means BASP is activated.
- RUN LED
  - The RUN LED blinks as soon as the OB 100 is operated and for at least 3s, even if the start-up time is shorter or the CPU gets to STOP due to an error.
  - This indicates the start-up.
- STOP LED
  - During the processing of the OB 100 the STOP LED is on and then turns off.
- When the CPU has completed the START-UP OB, it assumes the operating mode RUN.

**** OB 102 (Cold start)***

If there is a "Watchdog" error the CPU still remains in STOP state. With such an error the CPU must be manually started again. For this the OB 102 (cold start) must exist. The CPU will not go to RUN without the OB 102. Alternatively you can bring your CPU in RUN state again by an overall reset respectively by reloading your project.

Please consider that the OB 102 (cold start) may exclusively be used for treatment of a watchdog error.

Operating mode RUN

- The application program in OB 1 is processed in a cycle. Under the control of alarms other program sections can be included in the cycle.
- All timers and counters being started by the program are active and the process image is updated with every cycle.
- BASP is deactivated, i.e. all outputs are enabled.
- RUN-LED on
- STOP-LED off

Operating mode HOLD

The CPU offers up to 3 breakpoints to be defined for program diagnosis. Setting and deletion of breakpoints happens in your programming environment. As soon as a breakpoint is reached, you may process your program step by step.

Precondition

For the usage of breakpoints, the following preconditions have to be fulfilled:

- Testing in single step mode is possible with STL. If necessary switch the view via *'View ➔ STL'* to STL.
- The block must be opened online and must not be protected.

Approach for working with breakpoints

1. Activate *'View ➔ Breakpoint Bar'*.
2. Set the cursor to the command line where you want to insert a breakpoint.
3. Set the breakpoint with *'Debug ➔ Set Breakpoint'*.
   ⇨ The according command line is marked with a circle.
4. To activate the breakpoint click on *'Debug ➔ Breakpoints Active'*.
   ⇨ The circle is changed to a filled circle.
5. Bring your CPU into RUN.
   ⇨ When the program reaches the breakpoint, your CPU switches to the state HOLD, the breakpoint is marked with an arrow and the register contents are monitored.
6. Now you may execute the program code step by step via *'Debug ➔ Execute Next Statement'* or run the program until the next breakpoint via *'Debug ➔ Resume'*.
7. Delete (all) breakpoints with the option *'Debug ➔ Delete All Breakpoints'*.

Behavior in operating state HOLD

- The RUN-LED blinks and the STOP-LED is on.
- The execution of the code is stopped. No level is further executed.
- All times are frozen.
- The real-time clock runs is just running.
- The outputs were disabled (BASP is activated).
- Configured CP connections remain exist.

The usage of breakpoints is always possible. Switching to the operating mode test operation is not necessary.

With more than 2 breakpoints, a single step execution is not possible.

4.10.2 Function security

The CPUs include security mechanisms like a Watchdog (100ms) and a parameterizable cycle time surveillance (parameterizable min. 1ms) that stop res. execute a RESET at the CPU in case of an error and set it into a defined STOP state. The VIPA CPUs are developed function secure and have the following system properties:

Event	concerns	Effect
RUN → STOP	general	BASP (**B**efehls-**A**usgabe-**Sp**erre, i.e. command output lock) is set.
	central digital outputs	The outputs are disabled.
	central analog outputs	The outputs are disabled. ■ Voltage outputs issue 0V ■ Current outputs 0...20mA issue 0mA ■ Current outputs 4...20mA issue 4mA If configured also substitute values may be issued.
	decentral outputs	Same behaviour as the central digital/analog outputs.
	decentral inputs	The inputs are cyclically be read by the decentralized station and the recent values are put at disposal.
STOP → RUN res. PowerON	general	First the PII is deleted, then OB 100 is called. After the execution of the OB, the BASP is reset and the cycle starts with: Delete PIO → Read PII → OB 1.
	decentral inputs	The inputs are be read by the decentralized station and the recent values are put at disposal.
RUN	general	The program is cyclically executed: Read PII → OB 1 → Write PIO.

PII = Process image inputs

PIO = Process image outputs

4.11 Overall reset

Overview

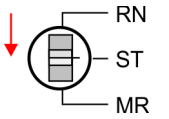
During the overall reset the entire user memory is erased. Data located in the memory card is not affected. You have 2 options to initiate an overall reset:

- Overall reset by means of the operating mode switch
- Overall reset by means of the *SPEED7 Studio*

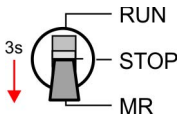
You should always establish an overall reset to your CPU before loading an application program into your CPU to ensure that all blocks have been cleared from the CPU.

4.11.1 Overall reset by means of the operating mode switch

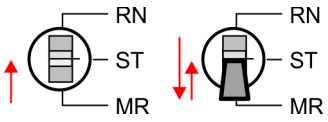
Proceeding



1. Your CPU must be in STOP mode. For this switch the operating mode switch of the CPU to STOP.
   ⇨ The ST LED is on.



2. Switch the operating mode switch to MR position for about 3 seconds.
   ⇨ The ST LED changes from blinking to permanently on.



3. Place the operating mode switch in the position STOP and switch it to MR and quickly back to STOP within a period of less than 3 seconds.
   ⇨ The overall reset is carried out. Here the ST LED flashes.
4. The overall reset has been completed when the ST LED is permanently on.

4.11.2 Overall reset via *SPEED7 Studio*

Proceeding

For the following proceeding you must be online connected to your CPU.

1. For an overall reset the CPU must be switched to STOP state. For this enable with *'View ➔ CPU control centre'* the *CPU control centre* and there switch your CPU to STOP state.
2. Request an overall reset via the *CPU control centre* or with *'Device ➔ Clear memory'*.
   ⇨ A dialog window opens. Here you can bring your CPU in STOP state, if not already done, and start the overall reset. During the overall reset procedure the STOP-LED flashes. When the STOP-LED is on permanently the overall reset procedure has been completed.

4.11.3 Actions after a memory reset

Activating functionality by means of a VSC

If there is a VSC from VIPA plugged, after an overall reset the according functionality is automatically activated. ↬ *'VSD' on page 86*

Automatic reload

If there is a project S7PROG.WLD on the memory card, after an overall reset the CPU attempts to reload this project from the memory card. → The SD LED is on. When the reload has been completed the LED expires. The operating mode of the CPU will be STOP respectively RUN, depending on the position of the operating mode switch.

Reset to factory setting

The *Reset to factory setting* deletes completely the internal RAM of the CPU and resets this to delivery state. Please regard that the MPI address is also set back to default 2!
↳ *Chapter 4.13 'Reset to factory settings' on page 84*

4.12 Firmware update

Overview

There is the opportunity to execute a firmware update for the CPU and its components via memory card. For this an accordingly prepared memory card must be in the CPU during the start-up. So a firmware files can be recognized and assigned with start-up, a pkg file name is reserved for each update-able component and hardware release, which begins with "px" and differs in a number with 6 digits. The pkg file name of every update-able component can be found at a label on the module. The SLIO CPU has no label. Here the pkg file name can be shown via the web page. After PowerON and operating mode switch in STOP position, the CPU checks if there is a *.pkg file at the memory card. If this firmware version is different to the existing firmware version, this is indicated by blinking of the LEDs and the firmware may be installed by an update request.

Current firmware at www.vipa.com

The latest firmware versions can be found in the service area at www.vipa.com. For example the following files are necessary for the firmware update of the CPU 015-CEFNR00 and its components with hardware release 02:

- CPU 015N, Hardware release 02: Px000310.pkg
- CP: Px000249.pkg

CAUTION!
When installing a new firmware you have to be extremely careful. Under certain circumstances you may destroy the CPU, for example if the voltage supply is interrupted during transfer or if the firmware file is defective. In this case, please call the VIPA hotline!

Please regard that the version of the update firmware has to be different from the existing firmware otherwise no update is executed.

Display the firmware version via web page

The CPU has an integrated web page that monitors information about the firmware version of the SPEED7 components. The Ethernet PG/OP channel provides the access to this web page. To activate the PG/OP channel you have to enter according IP parameters. This happens in the *SPEED7 Studio* via the "Initialization". After that you may access the PG/OP channel with a web browser via the set IP address. *Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61*

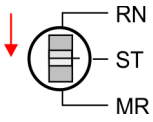
Load firmware and transfer it to memory card

1. Go to www.vipa.com
2. Click *'Service Support ➔ Downloads ➔ Firmware'*.
3. Via *'System SLIO ➔ CPU'* navigate to your CPU and download the zip file to your PC.
4. Unzip the zip file and copy the pkg file to the root directory of your memory card.

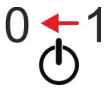
CAUTION!
With a firmware update an overall reset is automatically executed. If your program is only available in the load memory of the CPU it is deleted! Save your program before executing a firmware update! After a firmware update you should execute a "Reset to factory setting". *Chapter 4.13 'Reset to factory settings' on page 84*

Transfer firmware from memory card into CPU

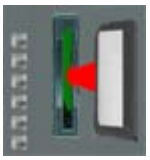
Please note that with some firmware versions an additional firmware update via alternate blinking of the LEDs SF and FC can be indicated even when the operating mode switch is in RUN position. In this state the CPU can only restart, if you establish a further firmware update process. For this tap the operating mode switch shortly downwards to MR and follow the procedures described below.



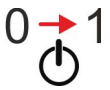
1. Switch the operating mode switch of your CPU in position STOP.



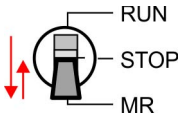
2. Turn off the power supply.



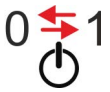
3. Plug the memory card with the firmware files into the CPU. Please take care of the correct plug-in direction of the memory card.



4. Turn on the power supply.
   ⇨ After a short boot-up time, the alternate blinking of the LEDs SF and FC shows that at least a more current firmware file was found at the memory card.



5. You start the transfer of the firmware as soon as you tip the operating mode switch downwards to MR within 10s and then leave the switch in STOP position.
   ⇨ During the update process, the LEDs SF and FC are alternately blinking and SD LED is on. This may last several minutes.
6. The update is successful finished when the LEDs PW, ST, SF, FC and SD are on. If they are blinking fast, an error occurred.

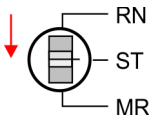


7. Turn power OFF and ON.
   ⇨ Now it is checked by the CPU, whether further firmware updates are to be executed. If so, again the LEDs SF and FC flash after a short start-up period. Continue with step 5. If the LEDs do not flash, the firmware update is finished.
8. Now execute a *Reset to factory setting*. After that the CPU is ready for duty.
   Chapter 4.13 'Reset to factory settings' on page 84

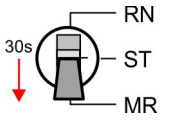
4.13 Reset to factory settings

Proceeding

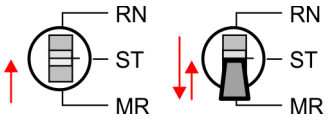
- With the following proceeding the internal RAM of the CPU is completely deleted and the CPU is reset to delivery state.
- Please regard that the MPI address is also reset to default 2 and the IP address of the Ethernet PG/OP channel is reset to 0.0.0.0!
- A factory reset may also be executed by the command *FACTORY_RESET*.
  Chapter 4.16 'CMD - auto commands' on page 88



1. Switch the CPU to STOP.



2. Push the operating mode switch down to position MR for 30 seconds. Here the ST LED blinks. After a few seconds the ST LED changes to static light. Now the ST LED changes between static light and blinking. Start here to count the static light of the ST LED.

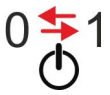


3. After the 6. Static light release the operating mode switch and tip it downwards to MR.

⇨ For the confirmation of the resetting procedure the green RN LED lights up once. This means that the RAM was deleted completely.

If the ST LED is on, only an overall reset has been performed and the reset to factory setting has been failed. In this case you can repeat the procedure. A factory reset can only be executed if the ST LED has static light for exact 6 times.

4. The update is successful finished when the LEDs PW, ST, SF, FC and MC are on.



5. Turn power OFF and ON.

After a firmware update of the CPU you always should execute a factory reset.

4.14 Deployment storage media - VSD, VSC

Overview

At the front of the CPU there is a slot for storage media. Here the following storage media can be plugged:

- VSD - **V**IPA **SD**-Card
  - External memory card for programs and firmware.
- VSC - **V**IPA**S**et**C**ard
  - External memory card (VSD) for programs and firmware with the possibility to unlock optional functions like work memory and field bus interfaces.
  - These functions can be purchased separately.
  - To activate the corresponding card is to be installed and a *Overall reset* is to be established. *Chapter 4.11 'Overall reset' on page 81*

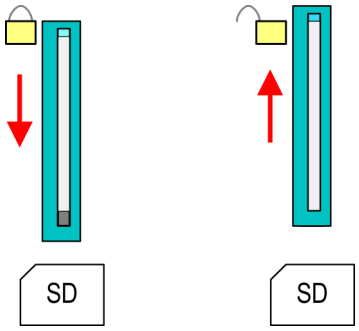
A list of the currently available VSD respectively VSC can be found at www.vipa.com

You can cause the CPU to load a project automatically respectively to execute a command file by means of pre-defined file names.

VSD

VSDs are external storage media based on SD memory cards. VSDs are pre-formatted with the PC format FAT 16 (max. 2GB) and can be accessed via a card reader. After PowerON respectively an overall reset the CPU checks, if there is a VSD with data valid for the CPU.

Push the VSD into the slot until it snaps in leaded by a spring mechanism. This ensures contacting. By sliding down the sliding mechanism, a just installed VSD card can be protected against drop out.



To remove, slide the sliding mechanism up again and push the storage media against the spring pressure until it is unlocked with a click.

CAUTION!
If the media was already unlocked by the spring mechanism, with shifting the sliding mechanism, a just installed memory card can jump out of the slot!

VSC

The VSC is a VSD with the possibility to enable optional functions. Here you have the opportunity to accordingly expand your work memory respectively enable field bus functions. Information about the enabled functions can be shown via the web page.
↳ *Chapter 4.9 'Accessing the web server' on page 73*

CAUTION!
Please regard that the VSC must remain plugged when you've enabled optional functions at your CPU. Otherwise the SF LED is on and the CPU switches to STOP after 72 hours. As soon as an activated VSC is not plugged, the SF LED is on and the "TrialTimer" counts downwards from 72 hours to 0. After 72 hours the CPU switches to STOP state. By plugging the VSC, the SF LED expires and the CPU is running again without any restrictions.

The VSC cannot be replaced by a VSC of the same optional functions. The activation code is fixed to the VSD by means of an unique serial number. Here the function as an external memory card is not affected.

Accessing the storage medium

To the following times an access takes place on a storage medium:

After overall reset

- The CPU checks if a VSC is inserted. If so, the corresponding optional functions are enabled.
- The CPU checks whether a project S7PROG.WLD exists. If so, it is automatically loaded.

After PowerON

- The CPU checks whether a project AUTOLOAD.WLD exists. If so, an overall reset is executed and the project is automatically loaded.
- The CPU checks whether a command file with the name VIPA_CMD.MMC exists. If so the command file is loaded and the commands are executed.
- After PowerON and CPU STOP the CPU checks if there is a *.pkg file (firmware file). If so, this is shown by the CPU by blinking LEDs and the firmware may be installed by an update request. *Chapter 4.12 'Firmware update' on page 83*

Once in STOP state

- If a memory card is plugged, which contains a command file VIPA_CMD.MMC, the command file is loaded and the containing instructions are executed.

The FC/SFC 208 ... FC/SFC 215 and FC/SFC 195 allow you to include the memory card access into your user application. More can be found in the manual "Operation list".

4.15 Extended know-how protection

Overview

Besides the "standard" Know-how protection the CPUs from VIPA provide an "extended" know-how protection that serves a secure block protection for accesses of 3. persons.

- Standard protection
  - The standard protection from Siemens transfers also protected blocks to the PG but their content is not displayed.
  - But with according manipulation the know-how protection is not guaranteed.
- Extended protection
  - The "extended" know-how protection developed by VIPA offers the opportunity to store blocks permanently in the CPU.
  - With the "extended" protection you transfer the protected blocks to a memory card into a WLD-file named protect.wld.
  - By plugging the memory card and then an overall reset the blocks in the protect.wld are permanently stored in the CPU.
  - You may protect OBs, FBs and FCs.
  - When back-reading the protected blocks into the PG, exclusively the block header are loaded. The block code that is to be protected remains in the CPU and cannot be read.

Protect blocks with protect.wld

1. Create a new wld file in your project engineering tool with *'File ➔ Memory Card file ➔ New'*.
2. Rename the wld file to "protect.wld".
3. Transfer the according blocks into the file by dragging them with the mouse from the project to the file window of protect.wld.
4. Transfer the file protect.wld to a memory card.

5. Plug the memory card into the CPU and execute an *overall reset*. ↳ *Chapter 4.11 'Overall reset' on page 81*

   ⇨ The overall reset stores the blocks in protect.wld permanently in the CPU protected from accesses of 3. persons.

Protection behaviour

Protected blocks are overwritten by a new protect.wld. Using a PG 3. persons may access protected blocks but only the block header is transferred to the PG. The block code that is to be protected remains in the CPU and cannot be read.

Change respectively delete protected blocks

Protected blocks in the RAM of the CPU may be substituted at any time by blocks with the same name. This change remains up to next overall reset. Protected blocks may permanently be overwritten only if these are deleted at the protect.wld before. By transferring an empty protect.wld from the memory card with an overall reset, you may delete all protected blocks in the CPU.

Usage of protected blocks

Due to the fact that reading of a "protected" block from the CPU monitors no symbol labels it is convenient to provide the "block covers" for the end user. For this, create a project of all protected blocks. Delete all networks in the blocks so that these only contain the variable definitions in the according symbolism.

4.16 CMD - auto commands

Overview

A *Command* file at a memory card is automatically executed under the following conditions:

- CPU is in STOP and memory card is plugged
- After each PowerON

Command file

The *Command* file is a text file, which consists of a command sequence to be stored as **vipa_cmd.mmc** in the root directory of the memory card. The file has to be started by CMD_START as 1. command, followed by the desired commands (no other text) and must be finished by CMD_END as last command.

Text after the last command *CMD_END* e.g. comments is permissible, because this is ignored. As soon as the command file is recognized and executed each action is stored at the memory card in the log file logfile.txt. In addition for each executed command a diagnostics entry may be found in the diagnostics buffer.

Commands

Please regard the command sequence is to be started with *CMD_START* and ended with *CMD_END*.

Command	Description	Diagnostics entry
CMD_START	In the first line CMD_START is to be located.	0xE801
	There is a diagnostics entry if CMD_START is missing.	0xE8FE
WAIT1SECOND	Waits about 1 second.	0xE803
LOAD_PROJECT	The function "Overall reset and reload from memory card" is executed. The wld file located after the command is loaded else "s7prog.wld" is loaded.	0xE805

Command	Description	Diagnostics entry
SAVE_PROJECT	The recent project (blocks and hardware configuration) is stored as "s7prog.wld" at the memory card. If the file just exists it is renamed to "s7prog.old". If your CPU is password protected so you have to add this as parameter. Otherwise there is no project written. Example: SAVE_PROJECT password	0xE806
FACTORY_RESET	Executes "factory reset".	0xE807
DIAGBUF	The current diagnostics buffer of the CPU is stored as "diagbuff.txt" at the memory card.	0xE80B
SET_NETWORK	IP parameters for Ethernet PG/OP channel may be set by means of this command. The IP parameters are to be given in the order IP address, subnet mask and gateway in the format x.x.x.x each separated by a comma. Enter the IP address if there is no gateway used.	0xE80E
CMD_END	In the last line CMD_END is to be located.	0xE802

Examples

The structure of a command file is shown in the following. The corresponding diagnostics entry is put in parenthesizes.

Example 1

`CMD_START`	Marks the start of the command sequence (0xE801)
`LOAD_PROJECT proj.wld`	Execute an overall reset and load "proj.wld" (0xE805)
`WAIT1SECOND`	Wait ca. 1s (0xE803)
`DIAGBUF`	Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)
`CMD_END`	Marks the end of the command sequence (0xE802)
`... arbitrary text ...`	Text after the command CMD_END is not evaluated.

Example 2

`CMD_START`	Marks the start of the command sequence (0xE801)
`LOAD_PROJECT proj2.wld`	Execute an overall reset and load "proj2.wld" (0xE805)
`WAIT1SECOND`	Wait ca. 1s (0xE803)
`WAIT1SECOND`	Wait ca. 1s (0xE803)
	IP parameter (0xE80E)
`SET_NETWORK 172.16.129.210,255.255.224.0,172.16.129.210`	
`WAIT1SECOND`	Wait ca. 1s (0xE803)
`WAIT1SECOND`	Wait ca. 1s (0xE803)
`DIAGBUF`	Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)
`CMD_END`	Marks the end of the command sequence (0xE802)
`... arbitrary text ...`	Text after the command CMD_END is not evaluated.

The parameters IP address, subnet mask and gateway may be received from the system administrator. Enter the IP address if there is no gateway used.

4.17 Control and monitoring of variables with test functions

Overview

For troubleshooting purposes and to display the status of certain variables you can access certain test functions in the *SPEED7 Studio*

- Test of the user program in the PLC simulator
- Monitoring blocks in the editor
- Viewing and changing variables in watch tables
- Recording of signals by logic analysis

4.17.1 Test of the user program in the PLC simulator

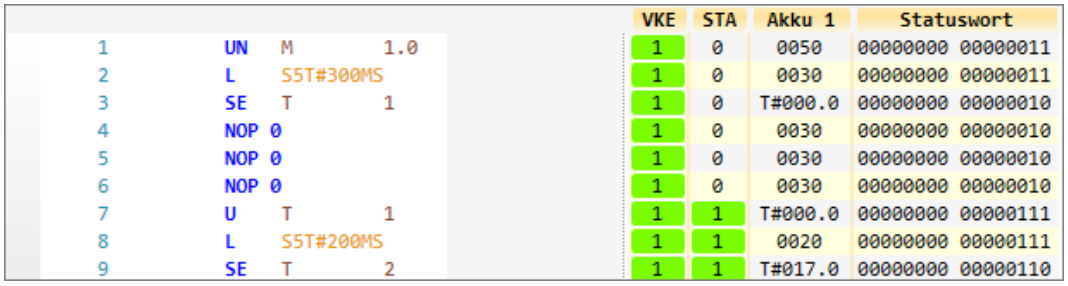
With the PLC simulator, you can test your application program in a virtual CPU before loading it into your PLC. The happens with the following proceeding:

1. Load your user program.
2. Compile your user program.
3. Set at *'Active PC interface'* the virtual interface *'Simulation'*.
4. Open the dialog *'PLC simulation settings'* and make your simulation adjustments if necessary.
5. Start the simulation with *'Simulation ➔ Start PLC simulation'*
   ⇨ The simulation is started.
6. Here you can test your application program, e.g. monitor values of variables or signal states or overwrite variables with values.
7. With *'Simulation ➔ Stop PLC simulation'* the simulation is stopped.

4.17.2 Monitoring blocks in the editor

In the *SPEED7 Studio* you can monitor variables of a block in the block editor. For this the block to be monitored must be loaded in the CPU and you must be online connected to the CPU.

1. Open the block of type OB, FB, FC or DB in the block editor.
2. Click at .
   ⇨ The variable values are cyclically read from the PLC and shown. Please note that in this case you can not make any changes at the block.



   Depending on the editor the result of logic operation (RLO), status bit (STA) and values of the accu and status word register are shown.
3. To stop the monitoring click again at .

4.17.3 Viewing and changing variables in watch tables

In the watch table you can monitor (read) and control (write) variables. You can specify , which variables of the CPU are to be read and controlled. You can create more watch tables if you want. This information is obtained from the corresponding area of the selected operands. During the controlling of variables respectively in operating mode STOP the input area is directly read. Otherwise only the process image of the selected operands is displayed.

Inputs can be monitored, but not be controlled. Outputs can be controlled, but not be observed.

Add a watch table

1. Click in the project tree within a PLC beneath *'PLC program'* at ' ➔ *Add watch table'*.
   ⇨ The dialog *'Add watch table'* is opened.
2. *'Name'*: Enter a name if needed.
3. *'Comment'*: Enter a comment if needed, e.g. add or description
4. Click at *'OK'*.
   ⇨ The watch table is added and listed in the project tree.
5. Open the watch table
6. Enter via the first line of the table the variables, which you want to monitor respectively control.
7. Marc with in the column *'Watch'* all the variables, you want to monitor.
8. Click at , to cyclically read data from the PLC.
9. Marc with in the column *'Control'* all the variables, you want to control.

10. Click at , to write all control values to the PLC with each PLC cycle.

CAUTION!
Please consider that controlling of output values represents a potentially dangerous condition.

These functions should only be used for test purposes respectively for troubleshooting.

4.17.4 Recording of signals by logic analysis

With the logic analysis you can cyclically record PLC signals. Please consider for this an appropriate license for the *SPEED7 Studio* is necessary. To start the logic analysis choose *'View ➔ Logic analysis'*. More information about the usage may be found in the online help of the *SPEED7 Studio*.

4.18 Diagnostic entries

Accessing diagnostic data

↳ *Appendix 'System specific event IDs' on page 242*

- You may read the diagnostic buffer of the CPU via the *SPEED7 Studio*. To show the diagnostic entries you choose in the *SPEED7 Studio 'Device ➔ Status of component'*. Here via *'Diagnostic buffer'* you can access the diagnostic buffer.
- The current content of the diagnostic buffer is stored at the memory card by means of the CMD DIAGBUF. ↳ *Chapter 4.16 'CMD - auto commands' on page 88*
- The diagnostic is independent from the operating mode of the CPU. You may store a max. of 100 diagnostic entries in the CPU.

5 Deployment PtP communication

5.1 Fast introduction

General

The CPU has a RS485 interface, which is per default set to PtP communication (point to point). This allows to connect via serial process connection to different source or target systems.

Protocols

The protocols respectively procedures ASCII, STX/ETX, 3964R, USS and Modbus are supported.

Parametrization

The parametrization of the serial interface happens during runtime using the FC/SFC 216 (SER_CFG). For this you have to store the parameters in a DB for all protocols except ASCII.

Communication

The FCs/SFCs are controlling the communication. Send takes place via FC/SFC 217 (SER_SND) and receive via FC/SFC 218 (SER_RCV). The repeated call of the FC/SFC 217 SER_SND delivers a return value for 3964R, USS and Modbus via RetVal that contains, among other things, recent information about the acknowledgement of the partner station. The protocols USS and Modbus allow to evaluate the receipt telegram by calling the FC/SFC 218 SER_RCV after SER_SND. The FCs/SFCs are included in the consignment of the CPU.

Use FCs in the SPEED7 Studio

To ensure compatibility with other programming tools, these blocks are available as FC and SFC and therefore designated as "FC/SFC". In the SPEED7 Studio you should always use FCs. This increases the compatibility with other programming tools.

Overview FCs/SFCs for serial communication

The following FCs/SFCs are used for the serial communication:

FC/SFC		Description
FC/SFC 216	SER_CFG	RS485 parameterize
FC/SFC 217	SER_SND	RS485 send
FC/SFC 218	SER_RCV	RS485 receive

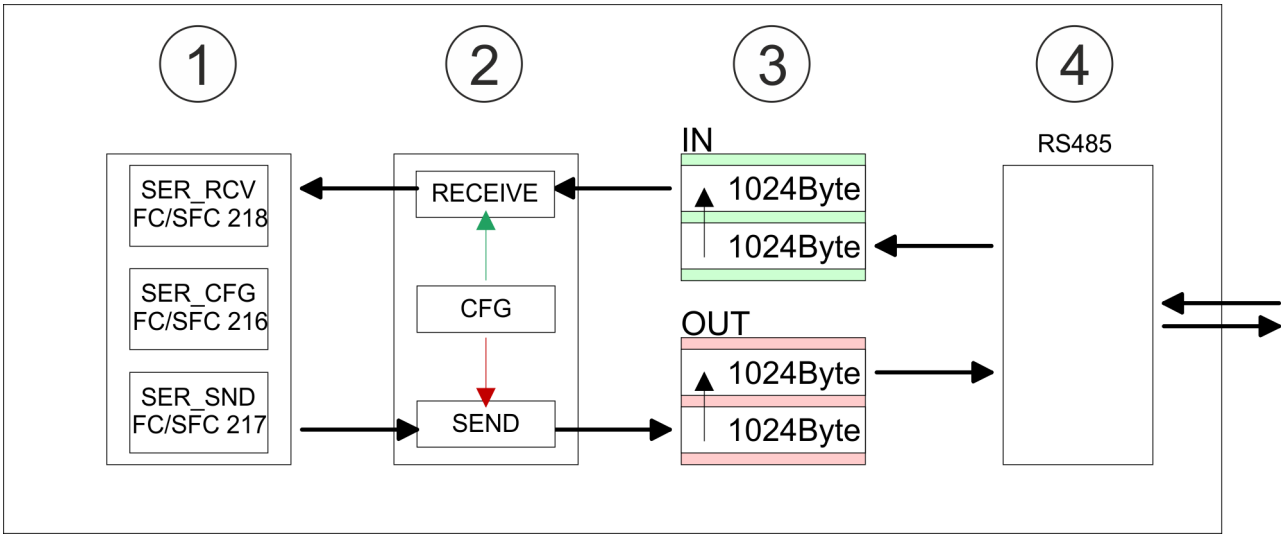
More information about the usage of these blocks may be found in the manual "SPEED7 Operation List" from VIPA.

5.2 Principle of the data transfer

RS485 PtP communication

The data transfer is handled during runtime by using FC/SFCs. The principle of data transfer is the same for all protocols and is shortly illustrated in the following.

- Data, which are written into the according data channel by the CPU, is stored in a FIFO send buffer (first in first out) with a size of 2x1024byte and then put out via the interface.
- When the interface receives data, this is stored in a FIFO receive buffer with a size of 2x1024byte and can there be read by the CPU.
- If the data is transferred via a protocol, the embedding of the data to the according protocol happens automatically.
- In opposite to ASCII and STX/ETX, the protocols 3964R, USS and Modbus require the acknowledgement of the partner.
- An additional call of the FC/SFC 217 SER_SND causes a return value in RetVal that includes among others recent information about the acknowledgement of the partner.
- Further on for USS and Modbus after a SER_SND the acknowledgement telegram must be evaluated by a call of the FC/SFC 218 SER_RCV.



1 Program
2 Protocol
3 FIFO buffer
4 Interface

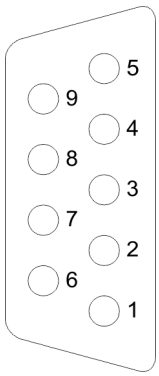
5.3 Deployment of RS485 interface for PtP

Properties RS485

- Logical states represented by voltage differences between the two cores of a twisted pair cable
- Serial bus connection in two-wire technology using half duplex mode
- Data communications up to a max. distance of 500m
- Data communication rate up to 115.2kbaud

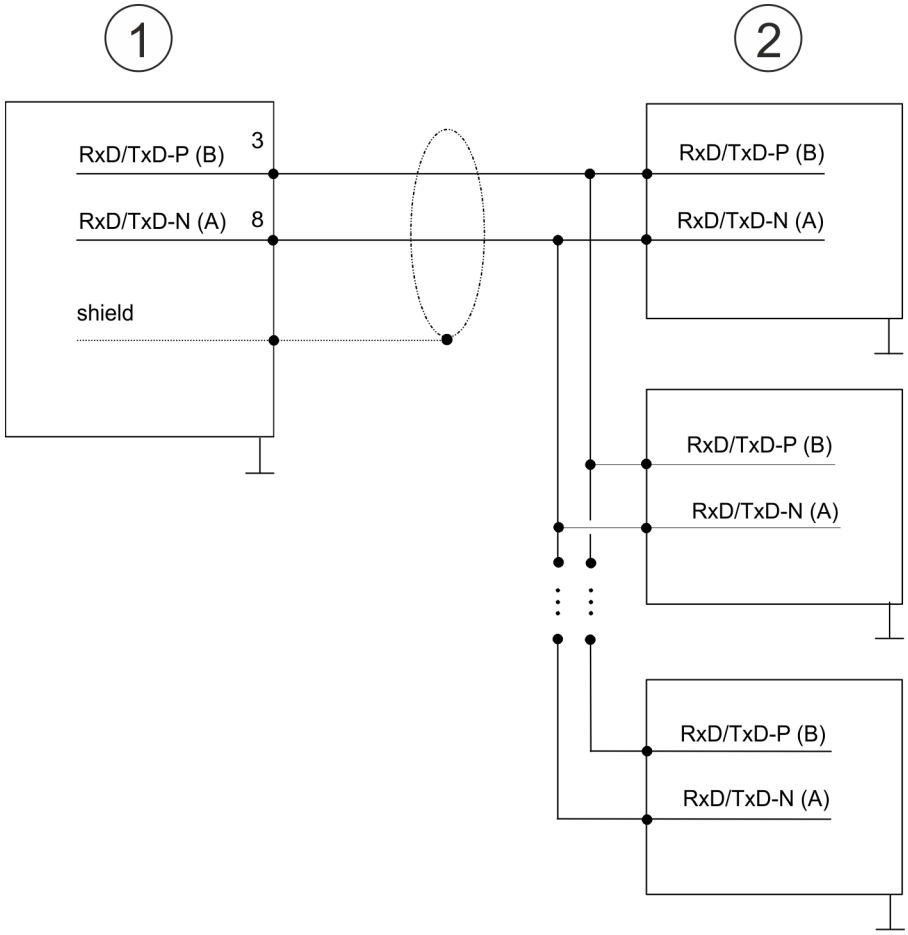
RS485

9pin SubD jack



Pin	RS485
1	n.c.
2	M24V
3	RxD/TxD-P (Line B)
4	RTS
5	M5V
6	P5V
7	P24V
8	RxD/TxD-N (Line A)
9	n.c.

Connection



1 RS485 interface
2 Periphery

5.4 Parametrization

5.4.1 FC/SFC 216 - SER_CFG - Parametrization PtP

The parametrization happens during runtime deploying the FC/SFC 216 (SER_CFG). You have to store the parameters for STX/ETX, 3964R, USS and Modbus in a DB.

5.5 Communication

5.5.1 FC/SFC 217 - SER_SND - Send to PtP

This block sends data via the serial interface. The repeated call of the FC/SFC 217 SER_SND delivers a return value for 3964R, USS and Modbus via RETVAL that contains, among other things, recent information about the acknowledgement of the partner station. The protocols USS and Modbus require to evaluate the receipt telegram by calling the FC/SFC 218 SER_RCV after SER_SND.

5.5.2 FC/SFC 218 - SER_RCV - Receive from PtP

This block receives data via the serial interface. Using the FC/SFC 218 SER_RCV after SER_SND with the protocols USS and Modbus the acknowledgement telegram can be read.

More information about the usage of these blocks may be found in the manual "SPEED7 Operation List" from VIPA.

5.6 Protocols and procedures

Overview

The CPU supports the following protocols and procedures:

- ASCII communication
- STX/ETX
- 3964R
- USS
- Modbus

ASCII

ASCII data communication is one of the simple forms of data exchange. Incoming characters are transferred 1 to 1. At ASCII, with every cycle the read FC/SFC is used to store the data that is in the buffer at request time in a parametrized receive data block. If a telegram is spread over various cycles, the data is overwritten. There is no reception acknowledgement. The communication procedure has to be controlled by the concerning user application. For this you can use the FB 1 - Receive_ASCII.

More information about the usage of this block may be found in the manual "SPEED7 Operation List" from VIPA.

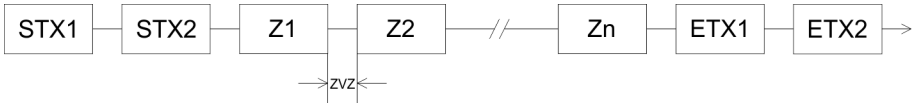
STX/ETX

STX/ETX is a simple protocol with start and end ID, where STX stands for **S**tart of **Tex**t and ETX for **E**nd of **Tex**t.

- Any data transferred from the periphery must be preceded by a Start followed by the data characters and the end character. Depending of the byte width the following ASCII characters can be transferred: 5bit: not allowed: 6bit: 20...3Fh, 7bit: 20...7Fh, 8bit: 20...FFh.
- The effective data, which includes all the characters between Start and End are transferred to the CPU when the End has been received.
- When data is send from the CPU to a peripheral device, any user data is handed to the FC/SFC 217 (SER_SND) and is transferred with added Start- and End-ID to the communication partner.
- You may work with 1, 2 or no Start- and with 1, 2 or no End-ID.
- If no End-ID is defined, all read characters are transferred to the CPU after a parameterizable character delay time (Timeout).

As Start-res. End-ID all Hex values from 01h to 1Fh are permissible. Characters above 1Fh are ignored. In the user data, characters below 20h are not allowed and may cause errors. The number of Start- and End-IDs may be different (1 Start, 2 End res. 2 Start, 1 End or other combinations). For not used start and end characters you have to enter FFh in the hardware configuration.

Message structure:



3964

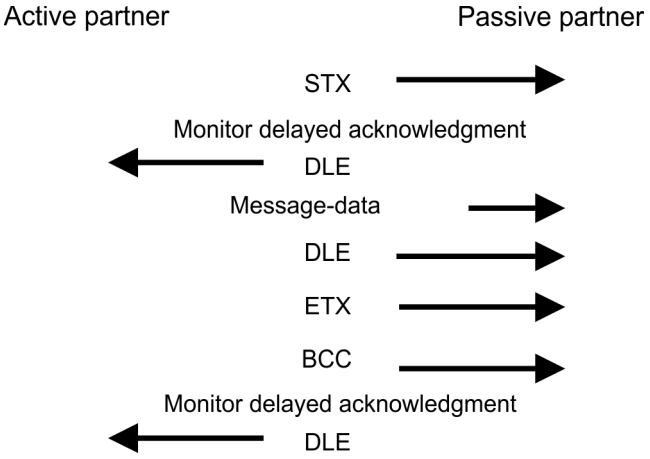
The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free receipt.

The procedure employs the following control characters:

- STX: **S**tart of **Te**x**t**
- DLE: **D**ata **L**ink **E**scape
- ETX: **E**nd of **Te**x**t**
- BCC: **B**lock **C**heck **C**haracter
- NAK: **N**egative **Ac**k**nowledge**

You may transfer a maximum of 255byte per message.

Procedure



When a DLE is transferred as part of the information it is repeated to distinguish between data characters and DLE control characters that are used to establish and to terminate the connection (DLE duplication). The DLE duplication is reversed in the receiving station.

The 3964R procedure requires that a lower priority is assigned to the communication partner. When communication partners issue simultaneous send commands, the station with the lower priority will delay its send command.

USS

The USS protocol (**U**niverselle **s**erielle **S**chnittstelle = universal serial interface) is a serial transfer protocol defined by Siemens for the drive and system components. This allows to build-up a serial bus connection between a superordinated master and several slave systems. The USS protocol enables a time cyclic telegram traffic by presetting a fix telegram length.

The following features characterize the USS protocol:

- Multi point connection
- Master slave access procedure
- Single master system
- Max. 32 participants
- Simple and secure telegram frame

It is essential:

- You may connect 1 master and max. 31 slaves at the bus
- The single slaves are addressed by the master via an address sign in the telegram.

- The communication happens exclusively in half-duplex operation.
- After a send command, the acknowledgement telegram must be read by a call of the FC/SFC 218 SER_RCV.

The telegrams for send and receive have the following structure:

Master slave telegram

STX	LGE	ADR	PKE		IND		PWE		STW		HSW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

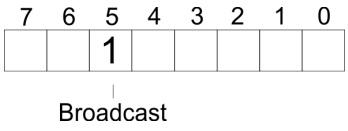
Slave master telegram

STX	LGE	ADR	PKE		IND		PWE		ZSW		HIW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

with

STX - Start sign
STW - Control word
LGE - Telegram length
ZSW - State word
ADR - Address
HSW - Main set value
PKE - Parameter ID
HIW - Main effective value
IND - Index
BCC - Block Check Character
PWE - Parameter value

Broadcast with set bit 5 in ADR byte



A request can be directed to a certain slave ore be send to all slaves as broadcast message. For the identification of a broadcast message you have to set bit 5 to 1 in the ADR byte. Here the slave addr. (bit 0 ... 4) is ignored. In opposite to a "normal" send command, the broadcast does not require a telegram evaluation via FC/SFC 218 SER_RCV. Only write commands may be sent as broadcast.

Modbus

- The Modbus protocol is a communication protocol that fixes a hierarchic structure with one master and several slaves.
- Physically, Modbus works with a serial half-duplex connection. There are no bus conflicts occurring, because the master can only communicate with one slave at a time.
- After a request from the master, this waits for a preset delay time for an answer of the slave. During the delay time, communication with other slaves is not possible.
- After a send command, the acknowledgement telegram must be read by a call of the FC/SFC 218 SER_RCV.
- The request telegrams send by the master and the respond telegrams of a slave have the following structure:

Telegram structure

Start sign	Slave address	Function Code	Data	Flow control	End sign

Broadcast with slave address = 0

- A request can be directed to a special slave or at all slaves as broadcast message.
- To mark a broadcast message, the slave address 0 is used.
- In opposite to a "normal" send command, the broadcast does not require a telegram evaluation via FC/SFC 218 SER_RCV.
- Only write commands may be sent as broadcast.

ASCII, RTU mode

Modbus offers 2 different transfer modes. The mode selection happens during runtime by using the FC/SFC 216 SER_CFG.

- ASCII mode: Every byte is transferred in the 2 sign ASCII code. The data are marked with a start and an end sign. This causes a transparent but slow transfer.
- RTU mode: Every byte is transferred as one character. This enables a higher data pass through as the ASCII mode. Instead of start and end sign, a time control is used.

Supported Modbus protocols

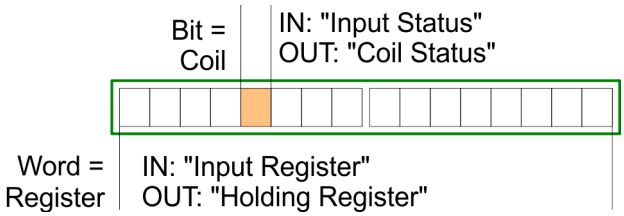
The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

5.7 Modbus - Function codes

Naming convention

Modbus has some naming conventions:



- Modbus differentiates between bit and word access; bits = "Coils" and words = "Register".
- Bit inputs are referred to as "Input-Status" and bit outputs as "Coil-Status".
- word inputs are referred to as "Input-Register" and word outputs as "Holding-Register".

Range definitions

Normally the access at Modbus happens by means of the ranges 0x, 1x, 3x and 4x.

0x and 1x gives you access to digital bit areas and 3x and 4x to analog word areas.

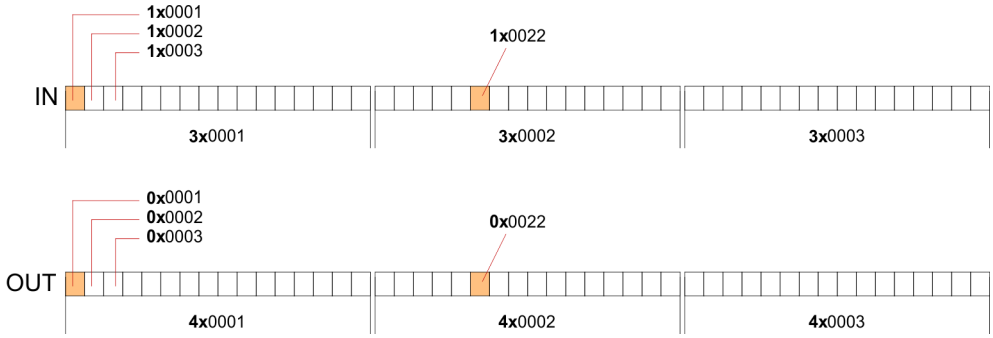
For the CPs from VIPA is not differentiating digital and analog data, the following assignment is valid:

0x - Bit area for master output data
Access via function code 01h, 05h, 0Fh

1x - Bit area for master input data
Access via function code 02h

3x - word area for master input data
Access via function code 04h

4x - word area for master output data
Access via function code 03h, 06h, 10h



A description of the function codes follows below.

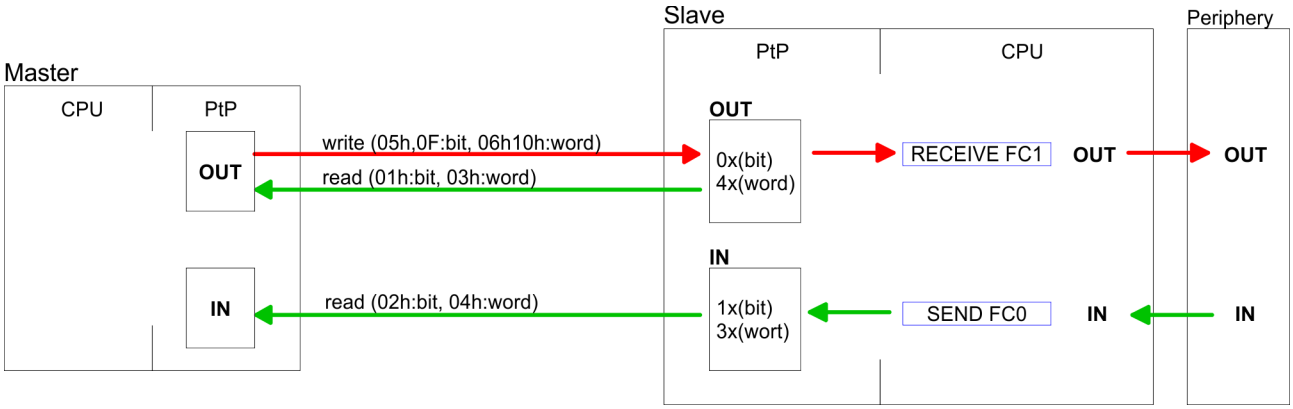
Overview

With the following Modbus function codes a Modbus master can access a Modbus slave:
With the following Modbus function codes a Modbus master can access a Modbus slave.
The description always takes place from the point of view of the master:

Code	Command	Description
01h	Read n bits	Read n bits of master output area 0x
02h	Read n bits	Read n bits of master input area 1x
03h	Read n words	Read n words of master output area 4x
04h	Read n words	Read n words master input area 3x
05h	Write 1 bit	Write 1 bit to master output area 0x
06h	Write 1 word	Write 1 word to master output area 4x
0Fh	Write n bits	Write n bits to master output area 0x
10h	Write n words	Write n words to master output area 4x

Point of View of "Input" and "Output" data

The description always takes place from the point of view of the master. Here data, which were sent from master to slave, up to their target are designated as "output" data (OUT) and contrary slave data received by the master were designated as "input" data (IN).



Respond of the slave

If the slave announces an error, the function code is send back with an "ORed" 80h.
Without an error, the function code is sent back.

Slave answer:	Function code OR 80h	→ Error
	Function code	→ OK

Byte sequence in a word

1 word	
High-byte	Low-byte

Check sum CRC, RTU, LRC

The shown check sums CRC at RTU and LRC at ASCII mode are automatically added to every telegram. They are not shown in the data block.

Read n bits 01h, 02h

Code 01h: Read n bits of master output area 0x
Code 02h: Read n bits of master input area 1x

Command telegram

Slave address	Function code	Address 1. bit	Number of bits	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Number of read bytes	Data 1. byte	Data 2. byte	...	Check sum CRC/LRC
1byte	1byte	1byte	1byte	1byte		1word
			max. 250byte			

Read n words 03h, 04h

03h: Read n words of master output area 4x
04h: Read n words master input area 3x

Command telegram

Slave address	Function code	Address 1. bit	Number of words	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Number of read bytes	Data 1. word	Data 2. word	...	Check sum CRC/LRC
1byte	1byte	1byte	1word	1word		1word
			max. 125words			

Write 1 bit 05h

Code 05h: Write 1 bit to master output area 0x

A status change is via "Status bit" with following values:

"Status bit" = 0000h → Bit = 0

"Status bit" = FF00h → Bit = 1

Command telegram

Slave address	Function code	Address bit	Status bit	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Address bit	Status bit	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Write 1 word 06h

Code 06h: Write 1 word to master output area 4x

Command telegram

Slave address	Function code	Address word	Value word	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Address word	Value word	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Write n bits 0Fh

Code 0Fh: Write n bits to master output area 0x

Please regard that the number of bits has additionally to be set in byte.

Command telegram

Slave address	Function code	Address 1. bit	Number of bits	Number of bytes	Data 1. byte	Data 2. byte	...	Check sum CRC/LRC
1byte	1byte	1word	1word	1byte	1byte	1byte	1byte	1word
					max. 250byte			

Respond telegram

Slave address	Function code	Address 1. bit	Number of bits	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Write n words 10h Code 10h: Write n words to master output area 4x

Command telegram

Slave address	Function code	Address 1. word	Number of words	Number of bytes	Data 1. word	Data 2. word	...	Check sum CRC/LRC
1byte	1byte	1word	1word	1byte	1word	1word	1word	1word
					max. 125words			

Respond telegram

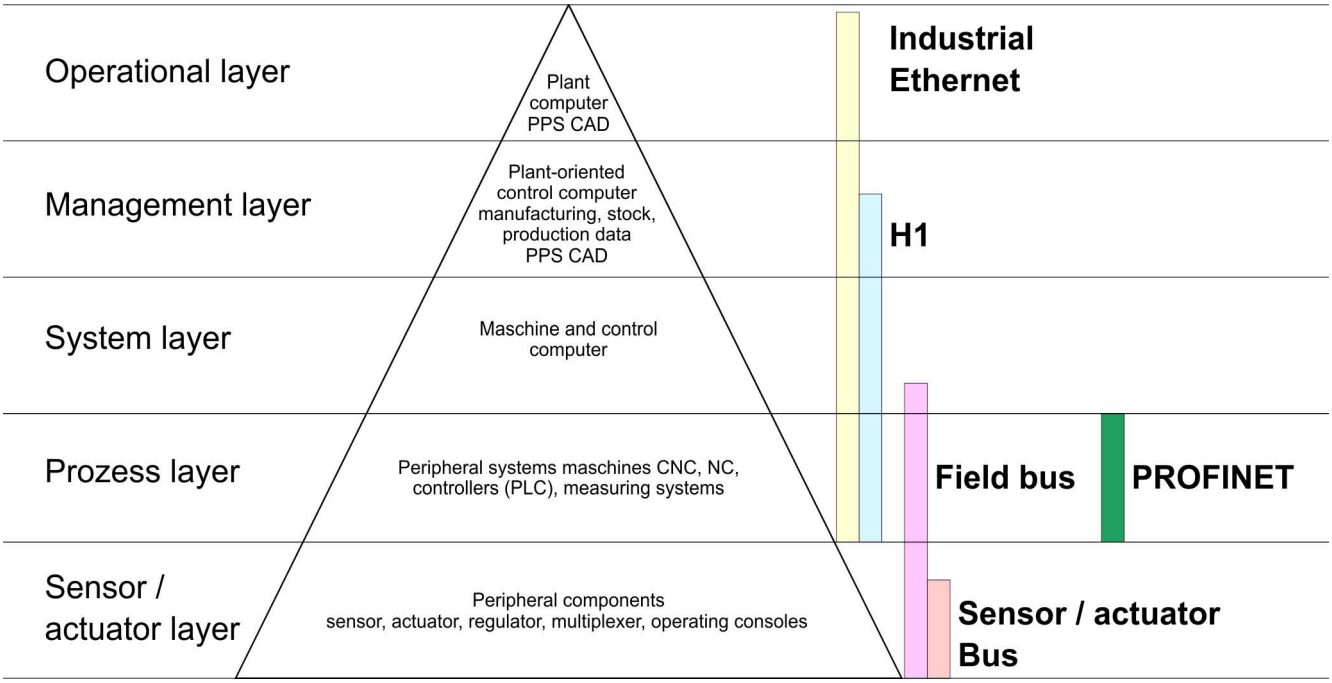
Slave address	Function code	Address 1. word	Number of words	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

6 Deployment Ethernet communication - productive

6.1 Basics - Industrial Ethernet in automation

Overview

The flow of information in a company presents a vast spectrum of requirements that must be met by the communication systems. Depending on the area of business the bus system or LAN must support a different number of users, different volumes of data must be transferred and the intervals between transfers may vary, etc. It is for this reason that different bus systems are employed depending on the respective task. These may be subdivided into different classes. The following model depicts the relationship between the different bus systems and the hierarchical structures of a company:



Industrial Ethernet

Industrial Ethernet is an electrical net based on shielded twisted pair cabling or optical net based on optical fibre. Industrial Ethernet is defined by the international standard IEEE 802.3

The net access of Industrial Ethernet corresponds to IEEE 802.3 - CSMA/CD (**C**arrier **S**ense **M**ultiple **A**ccess/**C**ollision **D**etection) scheme:

- Every station "listens" on the bus cable and receives communication messages that are addressed to it.
- Stations will only initiate a transmission when the line is unoccupied.
- In the event that two participants should start transmitting simultaneously, they will detect this and stop transmitting to restart after a random delay time has expired.
- Using switches there is the possibility for communication without collisions.

6.2 Basics - ISO/OSI reference model

Overview

The ISO/OSI reference model is based on a proposal that was developed by the International Standards Organization (ISO). This represents the first step towards an international standard for the different protocols. It is referred to as the ISO-OSI layer model. OSI is the abbreviation for **O**pen **S**ystem **I**nterconnection, the communication between open systems. The ISO/OSI reference model does not represent a network architecture as it does not define the services and protocols used by the different layers. The model simply specifies the tasks that the different layers must perform. All current communication systems are based on the ISO/OSI reference model, which is defined by the ISO 7498 standard. The reference model structures communication systems into 7 layers that cover different communication tasks. In this manner the complexity of the communication between different systems is divided amongst different layers to simplify the task.

The following layers have been defined:

- Layer 7 - Application Layer
- Layer 6 - Presentation Layer
- Layer 5 - Session Layer
- Layer 4 - Transport Layer
- Layer 3 - Network Layer
- Layer 2 - Data Link Layer
- Layer 1- Physical Layer

Depending on the complexity and the requirements of the communication mechanisms a communication system may use a subset of these layers.

Layer 1 - Bit communication layer (physical layer)

The bit communication layer (physical layer) is concerned with the transfer of data bits via the communication channel. This layer is therefore responsible for the mechanical, electrical and the procedural interfaces and the physical communication medium located below the bit communication layer:

- Which voltage represents a logical 0 or a 1?
- The minimum time the voltage is present to be recognized as a bit.
- The pin assignment of the respective interface.

Layer 2 - Security layer (data link layer)

This layer performs error-checking functions for bit strings transferred between two communicating partners. This includes the recognition and correction or flagging of communication errors and flow control functions. The security layer (data link layer) converts raw communication data into a sequence of frames. This is where frame limits are inserted on the transmitting side and where the receiving side detects them. These limits consist of special bit patterns that are inserted at the beginning and at the end of every frame. The security layer often also incorporates flow control and error detection functions. The data security layer is divided into two sub-levels, the LLC and the MAC level. The MAC (**M**edia **A**ccess **C**ontrol) is the lower level and controls how senders are sharing a single transmit channel. The LLC (**L**ogical **L**ink **C**ontrol) is the upper level that establishes the connection for transferring the data frames from one device into the other.

Layer 3 - Network layer

The network layer is an agency layer. Business of this layer is to control the exchange of binary data between stations that are not directly connected. It is responsible for the logical connections of layer 2 communications. Layer 3 supports the identification of the single network addresses and the establishing and disconnecting of logical communication channels. Additionally, layer 3 manages the prior transfer of data and the error processing of data packets. IP (Internet Protocol) is based on Layer 3.

Layer 4 - Transport layer

Layer 4 connects the network structures with the structures of the higher levels by dividing the messages of higher layers into segments and passes them on to the network layer. Hereby, the transport layer converts the transport addresses into network addresses. Common transport protocols are: TCP, SPX, NWLink and NetBEUI.

Layer 5 - Session layer

The session layer is also called the communication control layer. It relieves the communication between service deliverer and the requestor by establishing and holding the connection if the transport system has a short time fail out. At this layer, logical users may communicate via several connections at the same time. If the transport system fails, a new connection is established if needed. Additionally this layer provides methods for control and synchronization tasks.

Layer 6 - Presentation layer

This layer manages the presentation of the messages, when different network systems are using different representations of data. Layer 6 converts the data into a format that is acceptable for both communication partners. Here compression/decompression and encrypting/decrypting tasks are processed. This layer is also called interpreter. A typical use of this layer is the terminal emulation.

Layer 7 - Application layer

The application layer is the link between the user application and the network. The tasks of the application layer include the network services like file, print, message, data base and application services as well as the according rules. This layer is composed from a series of protocols that are permanently expanded following the increasing needs of the user.

6.3 Basics - Terms

Network (LAN)

A network res. LAN (Local Area Network) provides a link between different stations that enables them to communicate with each other. Network stations consist of PCs, IPCs, TCP/IP adapters, etc. Network stations are separated by a minimum distance and connected by means of a network cable. The combination of network stations and the network cable represent a complete segment. All the segments of a network form the Ethernet (physics of a network).

Twisted Pair

In the early days of networking the Triaxial- (yellow cable) or thin Ethernet cable (Cheapernet) was used as communication medium. This has been superseded by the twisted-pair network cable due to its immunity to interference. The CPU has a twisted-pair connector. The twisted-pair cable consists of 8 cores that are twisted together in pairs. Due to these twists this system is provides an increased level of immunity to electrical interference. For linking please use twisted pair cable which at least corresponds to the category 5. Where the coaxial Ethernet networks are based on a bus topology the twisted-pair network is based on a point-to-point scheme. The network that may be established by means of this cable has a star topology. Every station is connected to the star coupler (hub/switch) by means of a separate cable. The hub/switch provides the interface to the Ethernet.

Hub (repeater)

The hub is the central element that is required to implement a twisted-pair Ethernet network. It is the job of the hub to regenerate and to amplify the signals in both directions. At the same time it must have the facility to detect and process segment wide collisions and to relay this information. The hub is not accessible by means of a separate network address since it is not visible to the stations on the network. A hub has provisions to interface to Ethernet or to another hub res. switch.

Switch

A switch also is a central element for realizing Ethernet on Twisted Pair. Several stations res. hubs are connected via a switch. Afterwards they are able to communicate with each other via the switch without interfering the network. An intelligent hardware analyses the incoming telegrams of every port of the switch and passes them collision free on to the destination stations of the switch. A switch optimizes the bandwidth in every connected segment of a network. Switches enable exclusive connections between the segments of a network changing at request.

6.4 Basics - Protocols

Overview

Protocols define a set of instructions or standards that enable computer to establish communication connections and exchange information as error free as possible. A commonly established protocol for the standardization of the complete communication is the ISO/OSI layer model. *Chapter 6.2 'Basics - ISO/OSI reference model' on page 106*

The following protocols are used:

- Communication connections
  - Siemens S7 connections
- Open communication
  - UDP according to RFC 793
  - ISO on TCP according to RFC 1006
  - UDP according to RFC 768

Siemens S7 connections

With the Siemens S7 connection large data sets may be transferred between PLC systems based on Siemens STEP®7. Here the stations are connected via Ethernet. Precondition for the Siemens S7 communication is a configured connection table, which contains the defined connections for communication. This can be configured in *SPEED7 Studio*.

Properties:

- A communication connection is specified by a connection ID for each connection partner.
- The acknowledgement of the data transfer is established from the partner station at level 7 of the ISO/OSI reference model.
- At the PLC side FB/SFB VIPA handling blocks are necessary for data transfer for the Siemens S7 connections.

More information about the usage of these blocks may be found in the manual "SPEED7 Operation List" from VIPA.

Open communication

In the *Open communication* the communication takes place via the user program by means of handling blocks. These blocks are part of the *SPEED7 Studio*. These can be found in the *'Catalog'* at *'Blocks'*.

- *Connection-oriented protocols:*
  Connection-oriented protocols establish a (logical) connection to the communication partner before data transmission is started. And if necessary they terminate the connection after the data transfer was finished. Connection-oriented protocols are used for data transmission when reliable, guaranteed delivery is of particular importance. Also the correct order of the received packets is ensured. In general, many logical connections can exist on one physical line. The following connection-oriented protocols are supported with FBs for open communication via industrial Ethernet:
  – *TCP native accord. to RFC 793:*
    During data transmission, no information about the length or about the start and end of a message is transmitted. However, the receiver has no means of detecting where one message ends in the data stream and the next one begins. The transfer is stream-oriented. For this reason, it is recommended that the data length of the FBs is identical for the sending and receiving station. If the number of received data does not fit to the preset length you either will get not the whole data, or you will get data of the following job.
  – *ISO on TCP according to RFC 1006:*
    During data transmission, information on the length and the end of the message is also transmitted. The transfer is block-oriented If you have specified the length of the data to be received greater than the length of the data to be sent, the receive block will copy the received data completely into the receive range.
- *Connection-less protocol:*
  There is thus no establishment and termination of a connection with a remote partner. Connection-less protocols transmit data with no acknowledge and with no reliable guaranteed delivery to the remote partner.
  – *UDP according to RFC 768:*
    In this case, when calling the sending block you have to specify the address parameters of the receiver (IP address and port number). During data transmission, information on the length and the end of the message is also transmitted. Analog after finishing the receive block you get a reference to the address parameter of the sender (IP address and port no.) In order to be able to use the sending and receiving blocks first you have to configure the local communications access point at both sides. With each new call of the sending block, you reference the remote partner by specifying its IP address and its port number.

6.5 Basics - IP address and subnet

IP address structure

Exclusively IPv4 is supported. At IPv4 the IP address is a 32bit address that must be unique within the network and consists of 4 numbers that are separated by a dot. Every IP address is a combination of a *Net-ID* and a *Host-ID* and has the following

Structure: **xxx.xxx.xxx.xxx**

Range: 000.000.000.000 to 255.255.255.255

Net-ID, Host-ID

The **Net**work-ID identifies a network res. a network controller that administrates the network. The Host-ID marks the network connections of a participant (host) to this network.

Subnet mask

The Host-ID can be further divided into a *Subnet-ID* and a new *Host-ID* by using a bit for bit AND assignment with the Subnet mask.

The area of the original Host-ID that is overwritten by 1 of the Subnet mask becomes the Subnet-ID, the rest is the new Host-ID.

Subnet mask	binary all "1"		binary all "0"
IPv4 address	Net-ID	Host-ID	
Subnet mask and IPv4 address	Net-ID	Subnet-ID	new Host-ID

Address at first start-up

At the first commissioning the Ethernet PG/OP channel and the NET-CP have no IP address.

How IP address data are assigned to the Ethernet PG/OP channel ↯ *Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61*.

How IP address data are assigned to the NET-CP ↯ *Chapter 6.8 'Commissioning and Initialization' on page 112*.

Address classes

For IPv4 addresses there are five address formats (class A to class E) that are all of a length of 4byte = 32bit.

Class A	0	Network-ID (1+7bit)	Host-ID (24bit)
Class B	10	Network-ID (2+14bit)	Host-ID (16bit)
Class C	110	Network-ID (3+21bit)	Host-ID (8bit)
Class D	1110	Multicast group	
Class E	11110	Reserved	

The classes A, B and C are used for individual addresses, class D for multicast addresses and class E is reserved for special purposes. The address formats of the 3 classes A, B, C are only differing in the length of Network-ID and Host-ID.

Private IP networks

These addresses can be used as net-ID by several organizations without causing conflicts, for these IP addresses are neither assigned in the Internet nor are routed in the Internet. To build up private IP-Networks within the Internet, RFC1597/1918 reserves the following address areas:

Network class	from IP	to IP	Standard subnet mask
A	10.0.0.0	10.255.255.255	255.0.0.0
B	172.16.0.0	172.31.255.255	255.255.0.0
C	192.168.0.0	192.168.255.255	255.255.255.0
(The Host-ID is underlined.)			

Reserved Host-IDs

Some Host-IDs are reserved for special purposes.

Host-ID = "0"	Identifier of this network, reserved!
Host-ID = maximum (binary complete "1")	Broadcast address of this network

Never choose an IP address with Host-ID=0 or Host-ID=maximum! (e.g. for class B with subnet mask = 255.255.0.0, the "172.16.0.0" is reserved and the "172.16.255.255" is occupied as local broadcast address for this network.)

6.6 Basics - MAC address and TSAP

MAC address

There is a unique MAC address (**M**edia **A**ccess **C**ontrol) necessary for each CP. Usually a module is labelled with its MAC address by the manufacturer. This address should be used for project engineering of the CP. The MAC address has a length of 6bytes. On delivery the first three bytes specify the manufacturer. These bytes are assigned by the IEEE committee. The last 3 bytes may be assigned by the manufacturer. In a network several stations with the same MAC address may not exist. The MAC address may be changed at any time. You will get a valid MAC address from your network administrator.

- Broadcast address
  - The MAC address, with which all bits are set to 1, is:
    FF-FF-FF-FF-FF-FF
    This address is used as Broadcast address and addresses all participants in the net.
- Address at first start-up
  - Each CP of a VIPA CPU has an unique MAC address. This may be found on a label beneath the front flap.

TSAP

TSAP means **T**ransport **S**ervice **A**ccess **P**oint. ISO transport connections support TSAP length of 1...16byte. TSAPs may be entered in ASCII format or hexadecimal.

Address parameters

Station A				Station B
remote TSAP	→	ISO transport	→	local TSAP
local TSAP	←	connection	←	remote TSAP
MAC address A				MAC address B

An ISO transport connection is specified by a local and a remote connection endpoint. The TSAPs of an ISO transport connection must match as follows:

- Remote TSAP (in CP) = local TSAP (in destination station)
- Local TSAP (in CP) = remote TSAP (in destination station)

6.7 Fast introduction

Overview

At the first commissioning respectively after an overall reset with PowerON again of the CPU, the Ethernet PG/OP channel and the NET-CP have no IP address. These are only reachable by its MAC address. By means of the MAC address, which is printed at the front with the sequence address NET-CP and beneath address Ethernet PG/OP channel, you can assign IP address data to the according component. The assignment takes place directly via the device configuration in *SPEED7 Studio*.

Steps of configuration

For the configuration of the NET-CP for productive connections should be done by the follow approach:

- Assembly and commissioning
- Hardware configuration - CPU
- Configure connections
  - Siemens S7 connections
    (Configuration happens via *'Device and networking'* in the *SPEED7 Studio*, the communication via VIPA handling blocks)
  - Open communication
    (Configuration and communication happens by handling blocks)
- Transfer of the entire project to the CPU

6.8 Commissioning and Initialization

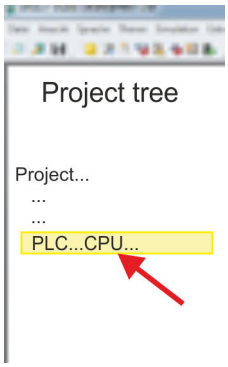
Assembly and commissioning

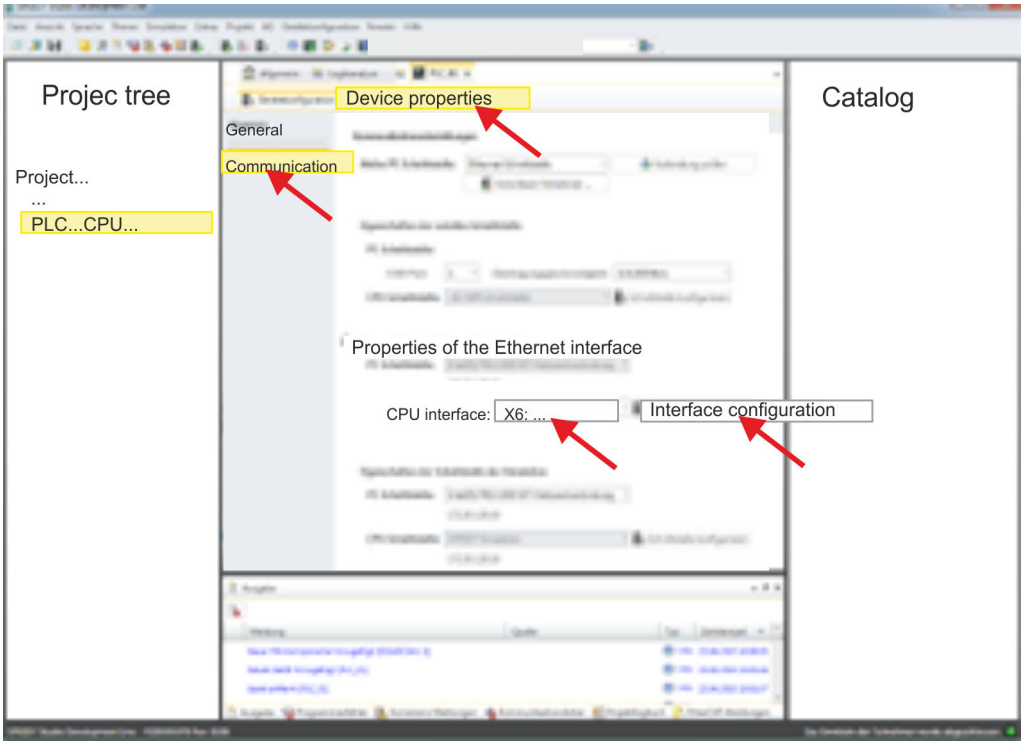
1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the Ethernet jack of the NET-CP (X6) to Ethernet.
4. Switch on the power supply.
   ⇨ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization"

You get valid IP address parameters from your system administrator. The assignment of the IP address data happens in the *SPEED7 Studio*with the following proceeding:

1. Start the *SPEED7 Studio* with your project.
2. Click in the *Project tree* at the CPU *'PLC ... CPU ...'*
3. Select *'Context menu ➔ Device properties'*.
   ⇨ The dialogue *'Device properties'* opens.





4. Here click at *'Communication'*
5. Select at *'Properties Ethernet interface'* as *'CPU-interface'* the interface *'X6:...'*.
6. Click at the button [Interface configuration].
7. Enter the wanted IP address data and confirm with [OK].
   ⇨ The IP address data are stored in your current project. After transferring your project, the NET-CP can be accessed via the set IP address data.

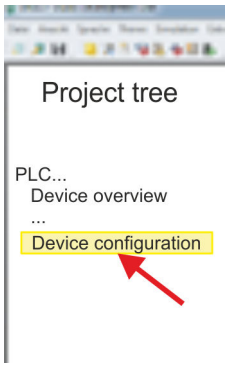
6.9 Hardware configuration - CPU

Precondition

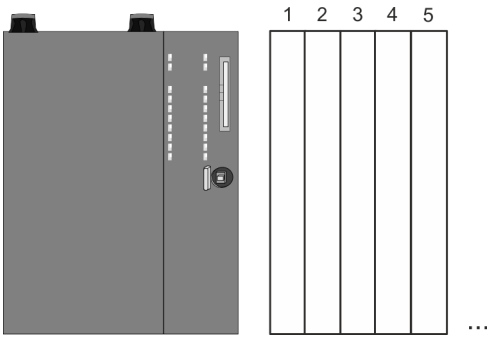
For project engineering a thorough knowledge of the SPEED7 Studio is required!

Proceeding

1. Start the *SPEED7 Studio*.
2. Create a new project in the *Work area* with *'New project'*.
   ⇨ A new project is created and the view *'Devices and networking'* is shown.
3. Click in the *Project tree* at *'Add new device ...'*.
   ⇨ A dialog for device selection opens.



4. Select from the *'Device templates'* your CPU and click at [OK].
   ⇨ The CPU is inserted in *'Devices and networking'* and the *'Device configuration'* is opened.



Device configuration

Slot	Module	...	...	...	...
0	CPU 015-CEFNR00				
-X1	PG_OP_Ethernet				
-X3	MPI interface				
...	...			...	

6.10 Configure Siemens S7 connections

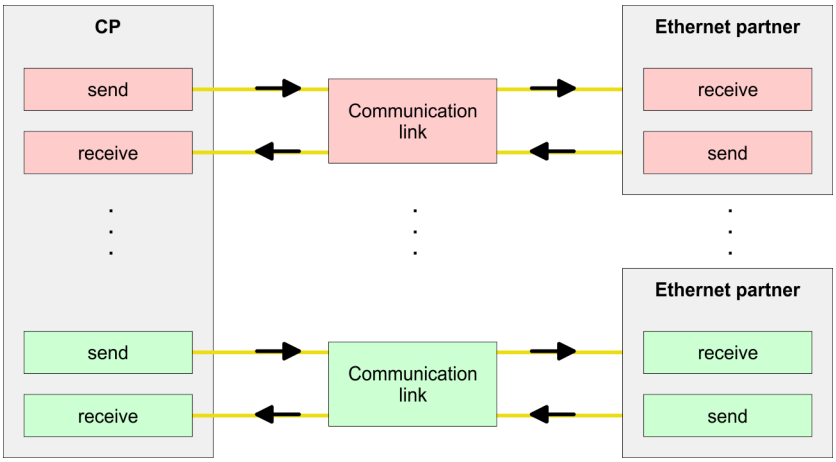
Overview

The configuration of S7 connections, i.e. the "link-up" between the stations happens in the *SPEED7 Studio* at *'Devices and networking'*. Here you can configure in tabular form communication connections In addition, the physical connections between the stations are graphically shown. A communication connection enables the program controlled communication between two participants at the Industrial Ethernet. The communication is controlled by the user program with VIPA handling blocks. To use this blocks, configured communication connections are always necessary in the active station.

Properties communication connection

The following properties are characterizing a communication connection:

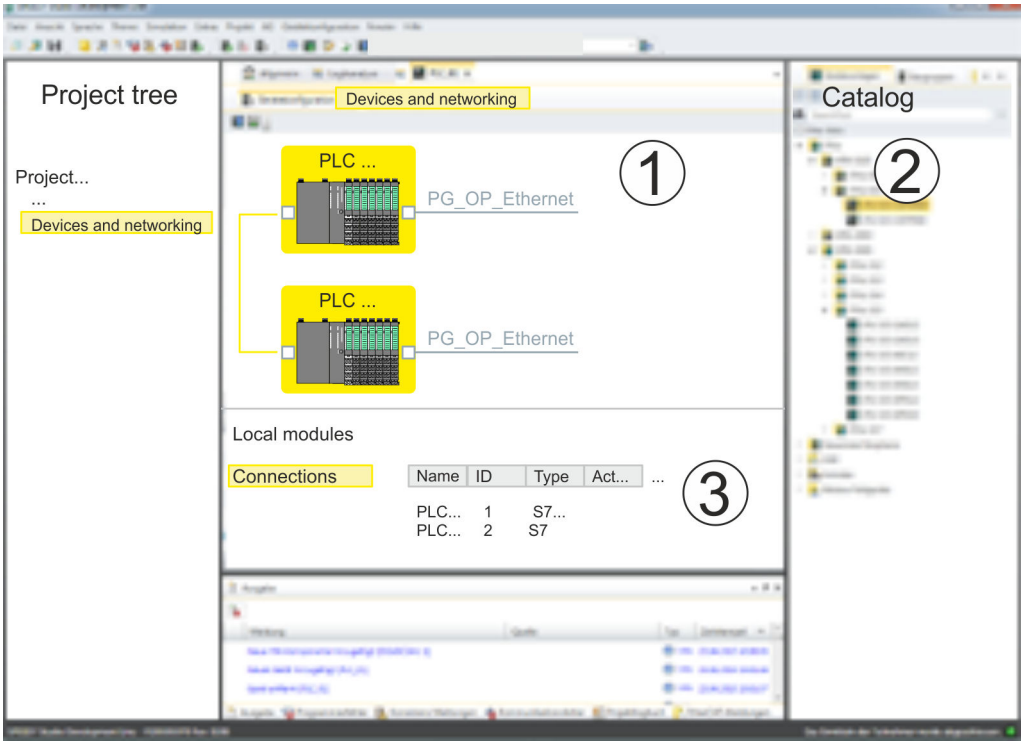
- One station always executes an active connection establishment.
- Bi-directional data transfer (Send and receive on one connection)
- Both participant have equal rights, i.e. every participant may initialize the send res. receive process event controlled.
- Except of the UDP connection, at a communication connection the address of the communication partner is set via the project engineering. Here the connection is active established by one station.



Working environment *'Devices and networking'*

For the configuration of connections a thorough knowledge of the *SPEED7 Studio* is required! The following passage only describes the basic usage *'Devices and networking'*. More information can be found in the online help respectively in documentation. After loading your project, *'Devices and networking'* can directly be called via the *Project tree*.

The working environment of *'Devices and networking'* has the following structure:

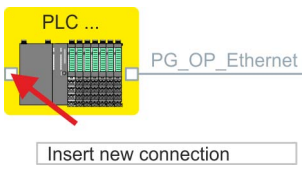


1 *Net view*: All stations and networks are displayed in a graphic view. By clicking on the according component you may access and alter the concerning properties.
2 *Catalog*: This area displays all available modules respectively net objects are shown in a directory view. By dragging a wanted object to the *net view* you may include further net objects.
3 *Connection table*: As soon as you select a module in the *net view*, the configured connections of this module are listed in the connection table. When you select a connection, via the context menu there is the option to edit, delete or to create a new connection.

Link up stations and configure connections

With *SPEED7 Studio* you can link up the communicating stations. The link up takes place in *Devices and networking* via the context menu of the NET-CP net marker with the following approach:

1. Select with the mouse the NET-CP net marker



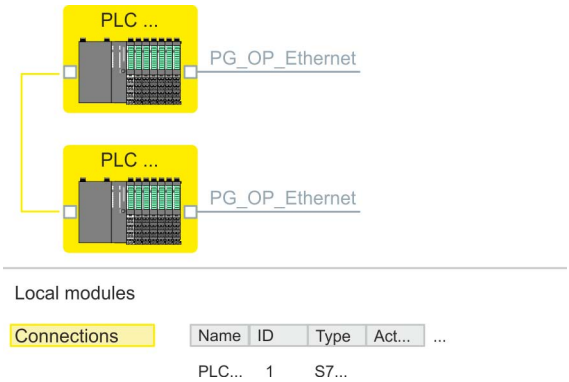
2. Open with *'Context menu ➔ Insert new connection'* the dialog to establish a new connection.

3. Choose partner station and connection type and confirm with [OK].

- *Connection partner - Station of your project*
  Each station configured in *SPEED7 Studio* is listed in the table of connection partner.
- *Connection partner - unspecified*
  Here the connection partner may exist in the current project or in an unknown project. Connection jobs to an unknown project must be defined by an unique IP address. Due to this allocation the connection remains *unspecified*.

⇨ A dialogue for setting the connection parameters opens.

4. Enter the according parameters and confirm with [OK].
   ⇨ The connection is established, listed in the *Connection table* and shown as graphical connection line between the stations at *Devices and networking*.



5. Configure in this way further connections. You also can edit you connections via the *Connection table* by selecting a connection and executing an according function via context menu like *'Insert new connection'*.

Connection types

With this CPU exclusively Siemens S7 connections may be configured with *SPEED7 Studio*.

Siemens S7 connection

- For data transfer with Siemens S7 connections the FB/SFB handling blocks are necessary; the deployment is described in the manual "Operation list" of your CPU.
- At Siemens S7 connections the communication connections are specified by a connection ID for each communication partner.
- A connection is specified by the local and partner connection end point.
- At Siemens S7 connections the TSAPs must be congruent crosswise.

The following parameters define a connection end point:

Station A				Station B
remote TSAP	→	Siemens S7-Verbindung	→	local TSAP
local TSAP	←		←	remote TSAP
ID A				ID B

Combination options with deployment of the FB/SFB handling blocks

Connection partner	Connection establishing	Connection
specified in the current project	active/passive	specified
unspecified	active	specified
	passive	unspecified

In the following every relevant parameter of a Siemens S7 connection is described:

- General
  - *End point:*
    Here you may define how the connection is to be established. Since the *SPEED7 Studio* can identify the communication options by means of the end points, some options are already preset and may not be changed.
  - *Name:*
    Here you can enter a name for your station
  - *Interface:*
    Here you can select the interfaces of your local and partner station.
  - *Address:*
    Here you can enter the MPI address of your local and partner station.
- Local ID
  - The ID is the link to your PLC program. The ID must be identical to the ID of the call interface of the FB/SFB handling block. Here you also can find the parameters *'ID'* and *'LADDR'*, which you have to specify in the handing blocks.
- Special features
  - *Active connection establishment:*
    An established connection is precondition for data transfer. By activating the option *'Establish an active connection'* the local station establishes the connection. Please regard not every station is able to active establish a connection. Here the job is to be made by the partner station.
- Address details
  - *Rack/slot:*
    Here you will find information about rack and slot of the local and the partner station.
  - Via this button a dialog window opens to show or enter address information of the local or partner station.
  - *Connection resources*
    The connection resource is part of the TSAP of the local station respectively of the partner. Not every connection resource may be used for every connection type. Depending on the connection partner and the connection type the range of values is limited respectively the connection resource is fix specified.
  - *TSAP:*
    With Siemens S7 connections a TSAP is automatically generated of the connection resource (one-way/two-way) and state of place (rack/slot).

Function blocks

FB/SFB	Designation	Description
FB/SFB 12	BSEND	Sending data in blocks FB/SFB 12 BSEND sends data to a remote partner FB/SFB of the type BRCV (FB/SFB 13). The data area to be transmitted is segmented. Each segment is sent individually to the partner. The last segment is acknowledged by the partner as it is received, independently of the calling up of the corresponding FB/SFB/FB BRCV. Due to the segmentation up to 65534byte data can be transferred with one send job.
FB/SFB 13	BRCV	Receiving data in blocks: The FB/SFB 13 BRCV can receive data from a remote partner FB/SFB of the type BSEND (FB/SFB 12). The parameter R_ID of both FB/SFBs must be identical. After each received data segment an acknowledgement is sent to the partner FB/SFB and the LEN parameter is updated.

FB/SFB	Designation	Description
FB/SFB 14	GET	Remote CPU read: The FB/SFB 14 GET can be used to read data from a remote CPU. The respective CPU must be in RUN mode or in STOP mode.
FB/SFB 15	PUT	Remote CPU write: The FB/SFB 15 PUT can be used to write data to a remote CPU. The respective CPU may be in RUN mode or in STOP mode.

6.11 Configure open communication

Connection-oriented protocols

- Connection-oriented protocols establish a (logical) connection to the communication partner before data transmission is started.
- And if necessary they terminate the connection after the data transfer was finished.
- Connection-oriented protocols are used for data transmission when reliable, guaranteed delivery is of particular importance.
- In general, many logical connections can exist on one physical line.

The following connection-oriented protocols are supported with FBs for open communication via Industrial Ethernet:

- *TCP/IP native according to RFC 793 (connection types 01h and 11h):*
  - During data transmission, no information about the length or about the start and end of a message is transmitted.
  - The receiver has no means of detecting where one message ends in the data stream and the next one begins.
  - The transfer is stream-oriented. For this reason, it is recommended that the data length of the FBs is identical for the sending and receiving station.
  - If the number of received data does not fit to the preset length you either will get not the whole data, or you will get data of the following job. The receive block copies as many bytes into the receive area as you have specified as length. After this, it will set NDR to TRUE and write RCVD_LEN with the value of LEN. With each additional call, you will thus receive another block of sent data.
- *ISO on TCP according to RFC 1006:*
  - During data transmission, information on the length and the end of the message is also transmitted.
  - The transfer is block-oriented
  - If you have specified the length of the data to be received greater than the length of the data to be sent, the receive block will copy the received data completely into the receive range. After this, it will set NDR to TRUE and write RCVD_LEN with the length of the sent data.
  - If you have specified the length of the data to be received less than the length of the sent data, the receive block will not copy any data into the receive range but instead will supply the following error information: ERROR = 1, STATUS = 8088h.

Connection-less protocol

- There is thus no establishment and termination of a connection with a remote partner.
- Connection-less protocols transmit data with no acknowledge and with no reliable guaranteed delivery to the remote partner.

The following connection-oriented protocol is supported with FBs for open communication via Industrial Ethernet:

- *UDP according to RFC 768 (with connection type 13h):*
  - In this case, when calling the sending block you have to specify the address parameters of the receiver (IP address and port number).
  - During data transmission, information on the length and the end of the message is also transmitted.
  - In order to be able to use the sending and receiving blocks first you have to configure the local communications access point at both sides.
  - With each new call of the sending block, you re-reference the remote partner by specifying its IP address and its port number.
  - If you have specified the length of the data to be received greater than the length of the data to be sent, the receive block will copy the received data completely into the receive range. After this, it will set NDR to TRUE and write RCVD_LEN with the length of the sent data.
  - If you have specified the length of the data to be received less than the length of the sent data, the receive block will not copy any data into the receive range but instead will supply the following error information: ERROR = 1, STATUS = 8088h.

Handling blocks

Those in the following listed UTDs and FBs serve for "open communication" with other Ethernet capable communication partners via your user program. These blocks are part of the *SPEED7 Studio*. You will find these in the *'Standard Library'*. Please consider when using the blocks for open communication that the partner station does not have to be configured with these blocks. This can be configured with AG_SEND/AG_RECEIVE or IP_CONFIG.

UDTs

FB	Designation	Connection-oriented protocols: TCP native as per RFC 793, ISO on TCP as per RFC 1006	Connectionless protocol: UDP as per RFC 768
UDT 65	TCON_PAR	Data structure for assigning connection parameters	Data structure for assigning parameters for the local communications access point
UDT 66	TCON_ADR		Data structure for assigning addressing parameters for the remote partner

FBs

FB	Designation	Connection-oriented protocols: TCP native as per RFC 793, ISO on TCP as per RFC 1006	Connectionless protocol: UDP as per RFC 768
FB 63	TSEND	Sending data	
FB 64	TRCV	Receiving data	
FB 65	TCON	Establishing a connection	Configuring the local communications access point
FB 66	TDISCON	Terminating a connection	Closing the local communications access point
FB 67*	TUSEND		Sending data
FB 68*	TURCV		Receiving data

)* in preparation

7 Ethernet communication - EtherCAT

7.1 Basics EtherCAT

7.1.1 General

Field buses were established for many years in the automation technology. Since higher speeds are required but the technical limits of this technology have already been reached, new solutions needed to be found.

At least in theory, the Ethernet, which is familiar to all of us from the office world, is fast with its 100Mbit/s speed, which is available everywhere today. However, these networks do not offer real-time capability due to the kind of cabling that they use and the rules governing access rights. This effect was corrected with EtherCAT®.

EtherCAT®

- For EtherCAT® is valid: EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- EtherCAT means Ethernet for Controller and Automation Technology. It was originally developed by Beckhoff Automation GmbH and is now supported and further developed by the EtherCAT Technology Group (ETG). ETG is the word biggest international user and producer connection for industrial Ethernet
- EtherCAT is an open Ethernet based field bus system, which is standardized at the IEC.
- As open field bus system EtherCAT matches the user profile for the part of industrial real-time systems.
- In opposition to the normal Ethernet communication at EtherCAT the data exchange of I/O data takes place during the frame passes the coupler with 100Mbit/s in full-duplex. Since in this way a frame to send and receive direction reaches the data of many stations, EtherCAT has a rate of user data of over 90%.
- The EtherCAT protocol, which is optimized for process data, is directly transported with the Ethernet frame. This again can consist of several sub-frames, which serve for a storage area of the process image.

Transfer medium

EtherCAT uses Ethernet as transfer medium Standard CAT5 cables are used. Here distances of about 100m between 2 stations are possible.

Only EtherCAT components may be used in an EtherCAT network. For topologies, which depart from the line structure, the corresponding EtherCAT components are necessary. Hubs may not be used.

Communication principle

At EtherCAT the master sends a telegram to the first station. The station takes its data from the current data stream, inserts its answer data and sends the frame to the succeeding station. Here the frame is handled with the same way.

When the frame has reached the last station this recognizes that no further is connected and sends the frame back to the master. Here the telegram is sent through every station via the other pair of leads (full-duplex). Due to the plug sequence and the use of the full-duplex technology EtherCAT represents a logical ring.

EtherCAT State Machine

Via the EtherCAT State Machine the state of the EtherCAT members is controlled.

Object dictionary (SDOs)

In the object directory the parameter, diagnostics, interrupt or other data are listed, which may be written or read via EtherCAT. The object directory may be accessed by the SDO information service. Additionally the object directory may be found in the device master file.

Process data (PDOs)

The EtherCAT data link layer is optimized for the fast transfer of process data. Here it is specified how the process data of the device are assigned to the EtherCAT process data and how the application of the device is synchronized to the EtherCAT cycle. The mapping of the process data happens by PDO mapping and by Sync-Manager-PDO-Assign objects. These describe, which objects of the object directory are transferred as object data via EtherCAT. The cycle time to transfer the process data via EtherCAT and how this is synchronized for the transfer is specified with the Sync-Manager-Communication objects.

Emergencies

Via Emergencies diagnostics, process events and errors at state change of the State Machine may be transferred.

Status messages, which show the current state of the device, should directly be transferred within the process data.

Distributed clocks (DC)

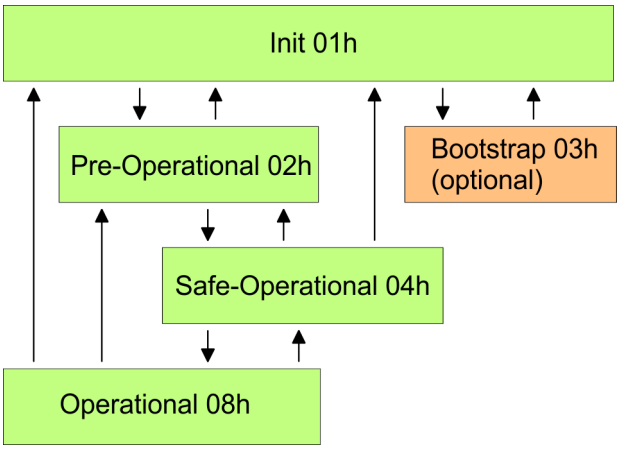
Due to the run time of an EtherCAT frame at the bus, the outputs of the EtherCAT slave stations are activated at different times and the inputs are read at different times. For an isochronous access to the process data EtherCAT provides the "distributed clock" functionality. In EtherCAT "**D**istributed **C**locks" means a logical combination of "clocks", which are located in the EtherCAT devices With this there is the possibility to locally provide a synchronized time in each bus device. With DC current output values are activated in equal time on the slave stations, and the input values read at precisely this time. This moment is called *Sync* signal. If an EtherCAT device supports the DC, it has its own clock. After PowerON this first locally works, based on an own pulse generator. By selecting an EtherCAT slave station, which has to provide the reference time, the distributed clocks can be synchronized. This *reference clock* so represents the system time. Among others there are the following DC setting parameters:

- Master/bus shift
  - Master shift: Based on DC the reference clock takes the "master" function, i.e. all DC slaves are adjusted based on the reference clock.
  - Bus Shift: Based on DC the reference clock takes the "slave" function, i.e. the EtherCAT master specifies how fast or how slow the reference clock has to run.
- Continuous Propagation Compensation
  - A command (datagram) will be inserted in the cyclic frame which allows the EtherCAT master to measure and compensate the propagation delay time by time.
- Sync Window Monitoring
  - A command (datagram) will be inserted in the cyclic frame which allows the EtherCAT master to determine the sync state ((*in-sync* or *out-of-sync*) of the system.

7.1.2 EtherCAT State Machine

States

In each EtherCAT communication device a *state machine* is implemented. For each state there is defined which communication service is active via EtherCAT. The state machine of the slave station is controlled by the state machine of the EtherCAT master.



Init - 01h

After power-on the EtherCAT members are in state *Init*. There is neither mailbox nor process data communication possible. The EtherCAT master initializes the SyncManager channels 0 and 1 for the mailbox communication.

Pre-Operational (Pre-Op) - 02h

The EtherCAT master initializes the SyncManager channels for process data (starting with SyncManager channel 2), the FMMU channels and the PDO mapping respectively the SyncManager PDO assignment. Further in this state the settings for process data transfer and the module-specific parameters, which deviate from the default values are transferred. During the transition from *Init* to *Pre-Op* the EtherCAT slave checks whether the mailbox was correctly initialized. In the state *Pre-Op* mailbox communication and Ethernet over EtherCAT (EoE) are possible but the process data communication is blocked.

Safe-Operational (Safe-Op) - 04h

In *Safe-Op* the input data are cyclically updated but the outputs are de-activated. With the transition from *Pre-Op* to *Safe-Op* the EtherCAT slave checks if the SyncManager channels for process data communication are correct. Before it acknowledges the state change, the EtherCAT slave copies current input data to the corresponding DP RAM areas of the EtherCAT slave controller. In the state *Safe-Op* mailbox and process data communication is possible.

Operational (Op) - 08h

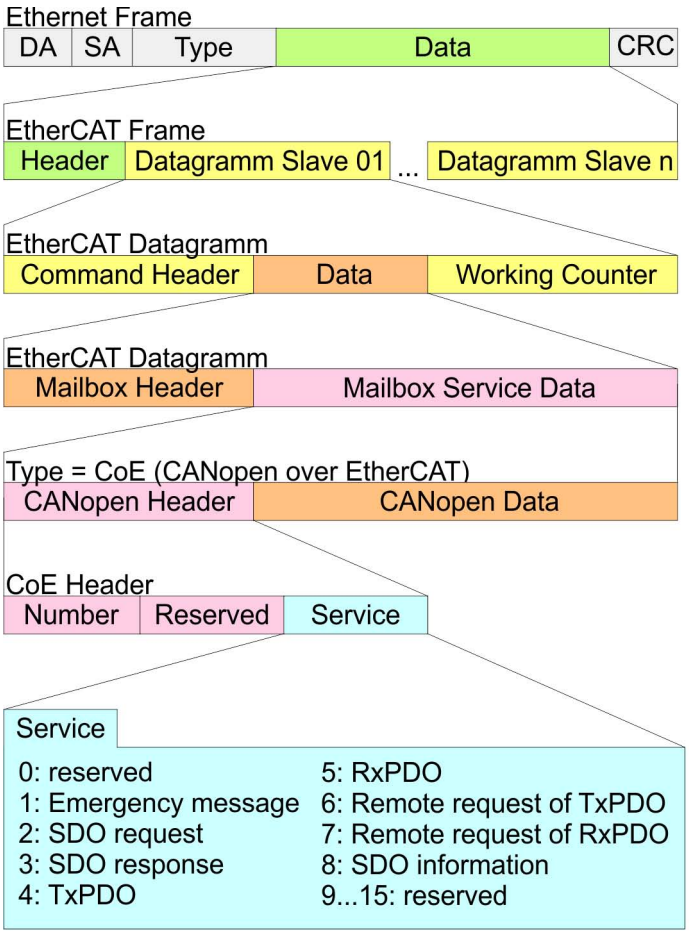
In the state *Op* the input data are cyclically updated and the EtherCAT master sends output data to the EtherCAT slave. The EtherCAT slave copies the output data of the master to its outputs and return input data to the EtherCAT master. In this state process data and mailbox communication is possible.

Bootstrap - option (Boot) - 03h

In state *Boot* the firmware of an EtherCAT slave may be updated via the EtherCAT master. This state may only be reached via Init. In the state *Boot* is mailbox communication via the protocol File-Access over EtherCAT (FoE) possible. Other mailbox and process data communications are de-activated.

7.1.3 CoE - CANopen over Ethernet

CoE means CANopen over EtherCAT. With CANopen you get a standard user interface, which makes a simplified system structure possible with most different devices. With CoE the device parameters may comfortably be accessed and data were may be read or written at the same time. Real-time data may be read by PDOs an the parametrization happens by SDOs. Further there are emergency objects available.



DA Destination address
SA Source address
CRC Checksum

7.2 Commissioning and start-up behavior

7.2.1 Assembly and commissioning

1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect your EtherCAT master to EtherCAT.
4. Switch on the power supply.

7.2.2 Start-up behavior

Preconditions for start-up

After PowerON and start-up (incl. OB100) the CPU is switched to RUN. This brings the EtherCAT master to *Op* state and he requests the *Op* state from its connected EtherCAT slave stations. Before the OB1 is called, the CPU waits for a defined time, that the EtherCAT slave stations have changed to*Op* state. You can specify the *Monitoring time* via the CPU parameter *'Transfer of parameters to modules'* in the property register *'start-up'*.

Using the EtherCAT master the following start-up behavior is distinguished. The terms and conditions can be found in the following table:

- **CPU switches to RUN, if topology is OK**
  The CPU waits for all the slaves, which mandatory have to exist, maximum until the *Monitoring time* expires and then switches to RUN. The topology mus be OK.
- **CPU switches to RUN mode regardless of topology or optional slaves**
  The CPU waits for all the slaves, which mandatory have to exist, maximum until the *Monitoring time* expires and then switches to RUN regardless of topology or optional slaves.

Truth table

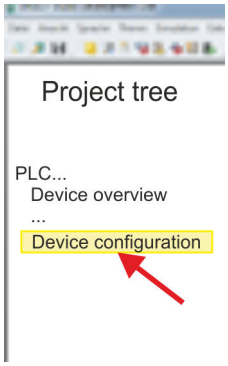
Is the CPU parameter: *'Start-up is preset configuration does not match actual configuration'* activated?	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Are all the mandatory slaves configured?	x	x	Y	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	Y	Y
Are there optional slaves configured (hot connect group)?	x	x	N	N	Y	Y	Y	Y	x	x	Y	Y	x	x	N	N	N	N	x	x
Do all the mandatory slaves exist?	x	x	Y	Y	N	N	Y	Y	x	x	Y	Y	x	x	x	x	x	x	N	N
Do optional slaves exist (not all must exist)?	x	x	N	N	Y	Y	Y	Y	x	x	Y	Y	x	x	x	x	x	x	x	x
Is there at least one mandatory slave with a wrong module?	x	x	N	N	N	N	N	N	Y	Y	x	x	x	x	x	x	x	x	x	x
Is there at least one optional slave with a wrong module?	x	x	N	N	N	N	N	N	x	x	Y	Y	x	x	x	x	x	x	x	x
Does at least on not configured slave exist?	x	x	N	N	N	N	N	N	x	x	x	x	Y	Y	Y	Y	N	N	x	x
Is DC (distributed clocks) activated?	N	Y	N	Y	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
Are all the slaves in-sync?	x	Y	x	Y	x	Y	x	Y	N	x	N	x	N	x	N	x	x	Y	N	x
Is the master in-sync to the first slave?	x	Y	x	Y	x	Y	x	Y	x	N	x	N	x	N	x	N	x	Y	x	N
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
CPU switches to RUN, if topology is OK.	Y	Y																		
CPU switches to RUN mode regardless of topology or optional slaves.			Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	N	N
Yes: Y | No: N | not relevant: X																				

7.3 Hardware configuration - CPU

The CPU is to be configured in the *SPEED7 Studio*. With the integrated *SPEED7 EtherCAT Manager* you can configure your EtherCAT network.

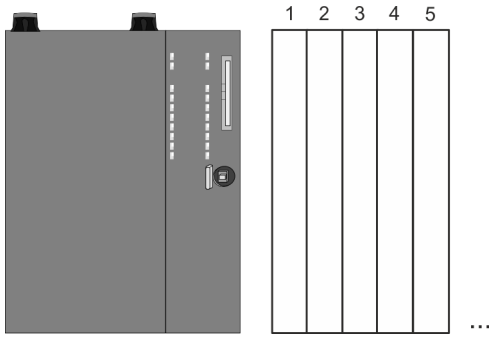
Proceeding

1. Start the *SPEED7 Studio*.
2. Create a new project in the *Work area* with *'New project'*.
   ⇨ A new project is created and the view *'Devices and networking'* is shown.
3. Click in the *Project tree* at *'Add new device ...'*.
   ⇨ A dialog for device selection opens.



4. Select from the *'Device templates'* your CPU and click at [OK].

⇨ The CPU is inserted in *'Devices and networking'* and the *'Device configuration'* is opened.



Device configuration

Slot	Module	...	...	...	...
0	CPU 015-CEFNR00				
-X1	PG_OP_Ethernet				
-X3	MPI interface				
...	...			...	

Please consider the additional functions in the SPEED7 Studio can only be activated, if you have valid license for these functions!

Proceeding

1. Click at the CPU in the *'Device configuration'* and select *'Context menu ➔ Components properties'*.

⇨ The properties dialog of the CPU is opened

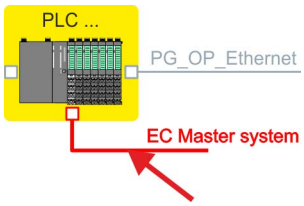
2. Click at *'Feature Sets'* and activate at *'Motion Control'* the parameter *'EtherCAT Master functionality+Motion Control+...'*.

3. Confirm your input with [OK].

⇨ The additional functions are now available in your project. More information about the usage may be found in the online help of the *SPEED7 Studio*.

CAUTION!
Please note that with any change in the feature set settings in the *SPEED7 Studio*, due to the system, the EtherCAT fieldbus system together with the motion control configuration are removed from your project!

Configuration EtherCAT master



1. Click in the Project tree at *'Add new device ...'*.
2. Click here at *'EC-Master system'* and select *'Context menu ➔ Bus system properties'*.
   ⇨ The *SPEED7 EtherCAT Manager* opens. Here you can configure the EtherCAT master system.
   More information about the usage of the *SPEED7 EtherCAT Manager* may be found in the online help of the *SPEED7 Studio*.
3. By closing the *SPEED7 EtherCAT Manager* the EtherCAT configuration is taken to the project and the *SPEED7 EtherCAT Manager* is closed. You can always edit your EtherCAT configuration in the *SPEED7 EtherCAT Manager*, since the configuration is stored in your project.

Since slave and module parameters are transmitted by means of SDO respectively SDO Init command, the configuration remains active, until a power cycle is performed or new parameters for the same SDO objects are transferred.

With an overall reset the slave and module parameters are not reset!

7.4 EtherCAT Diagnostics

Overview

There are the following ways to get diagnostics information from your system:

- Diagnostics via *SPEED7 EtherCAT Manager*
- Diagnostics during runtime in the user program (OB 1, SFB 52)
- Diagnostics via system status lists - SSL
- Diagnostics via OB start information
- Diagnostics via diagnostics CPU or CP
- Diagnostics via status LEDs

7.4.1 Diagnostics via *SPEED7 EtherCAT Manager*

Information

The *SPEED7 EtherCAT Manager* offers various opportunities for diagnostics:

- Diagnostics EtherCAT master
- Diagnostics EtherCAT slave station

More information about the usage of the SPEED7 EtherCAT Manager may be found in the online help.

Chapter 7.9.7 'Diagnostics - EC-Mastersystem' on page 181

Chapter 7.9.8 'Diagnostics - slave station' on page 184

7.4.2 Diagnostics during runtime in the user program (OB 1, SFB 52)

Handling block SFB 52 RDREC

With SFB 52 RDREC (read record) you can access diagnostics data from your user program e.g. in OB1. The SFB 52 RDREC operates asynchronously, that is, processing covers multiple SFB calls.

More information about the usage of the SFB 52 may be found in the online help of your programming tool or in the manual "Operation list".

The following data can be accessed with the SFB 52:

- CoE emergency messages (record set 0x4000 ... 0x4003)
- EtherCAT specific identification data (record set 0x1000)
- EtherCAT interface informations (record set 0x1037)
- EtherCAT register from slave station (record set 0x3000, 0x3001)
- EtherCAT register master (record set 0x3001)
- Analysis DC (record set 0x5000)

7.4.2.1 Accessing the CoE emergency messages

Record set 0x4000 ... 0x4003

With SFB 52 RDREC (read record) you can access CoE emergency messages from your user program e.g. in OB 1 by means of the record sets 0x4000 ... 0x4003. The SFB 52 RDREC operates asynchronously, that is, processing covers multiple SFB calls. An entry for the record sets 0x4000 ... 0x4003, which are described here, consists of the CoE emergency himself (8byte) and the station address of the CoE emergency comes from (2byte).

Record set structure

Index [byte]	Content	Description
0	NumberOfEntries	Number of following CoE emergency entries (0 ...n)
1		
2 + (n*12)	n * *CoE emergency entry*	CoE emergency entry according to the requested record set

CoE emergency entry

Index [byte]	Content	Description
0	Error Code	CoE emergency
1		
2	Error Register	
3	Error Data	
4		
5		
6		
7		
8	Station Address	Address of the station, which has sent the emergency.

Index [byte]	Content	Description
9		
10	Reserved	
11		

Record sets

Record set	Description
0x4000	The record set provides the last CoE emergency of each slave (on CoE emergency entry per slave, which has supplied a CoE emergency). There are no entries for slaves with no CoE emergency. Parameters: None NumberOfEntries: 0 ... 512
0x4001	The record set provides the last CoE emergency of a specific slave. If a slave ID is passed, which does not exist, an error is returned. If the slave ID is valid but no CoE emergency for this slave exists, the number of sent entries is equivalent to 0. Parameters: Slave ID (1 ... 512) NumberOfEntries: 0 ... 1
0x4002	The record set provides the last 20 CoE emergencies of the whole system (this means multiple entries for one slave can be reported). Is there a total of less than 20 entries, the number of messages is correspondingly smaller. Parameters: None NumberOfEntries: 0 ... 20
0x4003	The record set provides the last 10 CoE emergency of a specific slave. If a slave ID is passed, which does not exist, an error is returned. If the slave ID is valid but less than 10 CoE emergencies for this slave exist, the number of sent entries is correspondingly smaller. Parameters: Slave ID (1 ... 512) NumberOfEntries: 0 ... 10

Example OB 1

For cyclical access to a record set of the diagnostics data of an EtherCAT slave station, you can use the following example program in OB 1:

```
UN M10.3 'Read process finished (BUSY=0)
UN M10.1 'If there is no job activation
         'then (REQ=0)
S  M10.1 'start record set transfer (REQ:=1)
L  W#16#4000 'record set number(here record set 0x4000)
T  MW12
CALL SFB 52, DB52 'Call SFB 52 with instance DB
  REQ :=M10.1 'Start flag
  ID :=DW#16#0018 'Address of the EtherCAT slave
  INDEX :=MW12
  MLEN :=14 'Length record set 0x4000 with 1. entry
  VALID :=M10.2 'Validity of the record set
  BUSY :=M10.3 'Shows if job just running
  ERROR :=M10.4 'Error bit during read access
  STATUS :=MD14 'Error codes
  LEN :=MW16 'Length of the read record set
  RECORD :=P#M 100.0 Byte 40 'Target (MB100, 40byte)
U M10.1
R M10.1 'Reset of REQ
```

7.4.2.2 Accessing EtherCAT specific identification data

Record set 0x1000

The record set 0x1000 contains EtherCAT specific identification data, which can be read with the SFB 52. The values *Device Type*, *Serial Number*, *Hardware Version* and *Software Version* are directly retrieved via CoE from the slave station. If a slave station does not support CoE or one of these values in the object directory, the values are substituted with 0xFF. The record set has the following structure:

Index	Designation	Data type
1	Address	Unsigned32
2	Device Name	Array of char[32]
3	Vendor ID	Unsigned32
4	Product Code	Unsigned32
5	Device Type	Unsigned32
6	Serial Number	Unsigned32
7	Revision	Unsigned32
8	Hardware Version	Array of char[8]
9	Software Version	Array of char[8]

7.4.2.3 Accessing information of the EtherCAT interface

Record set 0x1037

The record set 0x1037 contains information about the Ethernet interface of the EtherCAT master, which can be read with the SFB 52. The record set has the following structure:

Index	Designation	Data type
1	Logical address	Unsigned16
2	IP address	Unsigned32
3	Subnet mask	Unsigned32
4	Default Router	Unsigned32
5	MAC address	Array of Unsigned8[6]
6	Source	Unsigned8
7	reserved	Unsigned8
8	DCP Mod Timestamp	Array of Unsigned8[8]
9	phys_mode_1	Unsigned8
10	phys_mode_2	Unsigned8
11	phys_mode_3	Unsigned8
12	phys_mode_4	Unsigned8
13	phys_mode_5	Unsigned8
14	phys_mode_6	Unsigned8
15	phys_mode_7	Unsigned8
16	phys_mode_8	Unsigned8
17	phys_mode_9	Unsigned8
18	phys_mode_10	Unsigned8
19	phys_mode_11	Unsigned8
20	phys_mode_12	Unsigned8
21	phys_mode_13	Unsigned8
22	phys_mode_14	Unsigned8
23	phys_mode_15	Unsigned8
24	phys_mode_16	Unsigned8
25	reserved	Unsigned8

7.4.2.4 Accessing the EtherCAT register from slave stations

Record set 0x3000

With the record set 0x3000 you can access the registers of an EtherCAT slave station, by calling it with the SFB 52. The record set has the following structure:

Byte	Content	Register
0	AL Status	0x0130, 0x0131
1		
2	AL Control	0x0120, 0x0121
3		
4	Al Status Code	0x0134, 0x0135

Byte	Content	Register
5		
6	ESC DL Status	0x0110, 0x0111
7		
8	Processing Unit Error Counter	0x030C
9	PDI Error Counter	0x030D
10	Link Lost Counter Port A	0x0310
11	Link Lost Counter Port B	0x0311
12	Link Lost Counter Port C	0x0312
13	Link Lost Counter Port D	0x0313
14	reserved	-
15	reserved	-

Record set 0x3001

The record set can be used to read the last reported *AL Status Code* of an EtherCAT slave station. The content of the record set remains until an overall reset is made or a new configuration is loaded.

Byte	Content	Register
0	AI Status Code	0x0134, 0x0135
1		

If you use an invalid slave address (slave ID) you will receive an error. If the slave ID is available but the EtherCAT slave station has not just reported an AL Status Code, so you also get an error.

7.4.2.5 Accessing the EtherCAT master register

Record set 0x3001

The record set can be used to read the last reported *AL Status Codes* of all the EtherCAT slave stations. If an EtherCAT slave station did not report a bug by the time of reading, so the returned *AI Status Code* is 0. The content of the record set remains until an overall reset is made or a new configuration is loaded.

Structure record set

Byte	Content
0	Data block for slave ID 1
4	Data block for slave ID 2
...	...
2043	Data block for slave ID 512

Structure data block

Byte	Content	Description
0	AI Status Code	*AL Status Code* of the corresponding EtherCAT slave station
1		
2	Validity	Validity: ■ 0: *AL Status Code* not valid (slave ID is not configured or EtherCAT slave station has not reported an *AL Status Code*, yet). ■ 1: *AL Status Code* is valid
3	reserved	-

Validity is only set to 1 if an AI Status Code is reported from the EtherCAT slave station. With an error-free EtherCAT slave station this byte is 0.

7.4.2.6 Analysis DC

Record set 0x5000

This record set informs about the current status of the DC system.

- These values are only updated with corresponding messages, which also generate a diagnostic buffer entry.
  - The parameters DC_InSync and DC_Deviation are updated with the message "EC_NOTIFY_DC_SLV_SYNC".
  - The parameters DCM_InSync, DCM_CtlErrorCur, DCM_CtlErrorAvg and DCM_CtlErrorMax are updated with the message "EC_NOTIFY_DCM_SYNC".
- Except the counter of "out of sync" the data come from the EtherCAT stack. For this reason, the nomenclature of the EtherCAT stack is taken for this data.

Structure of the data on reading

Index	Name	Type	Description	Default value
1	DC_InSync	DWORD	Indicates whether the DC slaves are synchronized with each other. 0: out of sync 1: in sync	0
2	DC_Deviation	DINT	Deviation in ns	0
3	DC_OutOfSyncCnt	DWORD	Counter, how often DC slaves were "out of sync". The counter is reset when an overall reset is performed or when a new configuration is loaded to the CPU.	0
4	DCM_InSync	DWORD	Indicates whether the DC master and Reference-Clock are synchronized with each other. 0: out of sync 1: in sync	0
5	DCM_CtlErrorCur	DINT	Current DC master deviation in ms	0
6	DCM_CtlErrorAvg	DINT	Average DC master deviation in ns.	0

Index	Name	Type	Description	Default value
7	DCM_CtlErrorMax	DINT	Maximum DC master deviation in ns	0
8	DCM_OutOfSyncCnt	DWORD	Counter, how often DC master was "out of sync". The counter is reset when an overall reset is performed or when a new configuration is loaded to the CPU.	0

7.4.3 Diagnostics via system status lists - SSL

SSL partial lists

In the following all the possible SSL partial lists with additional SSL-ID are listed, which are supported by the EtherCAT master system.

More information about the usage of the SSLs can be found in the manual "Operation list".

SSL partial lists	SSL-ID
SSL content	xy00h
Module identification	xy11h
Status of all LEDs	xy19h
Status of the LEDs	xy74h
Status information CPU	xy91h
Stations status information	xy94h
Module status information	xy96h
Diagnostic buffer of the CPU	xyA0h
Information EtherCAT Master/Slave	xyE0h
EtherCAT bus system	xyE1h
Type ID modules central bus system	xyF0h
Status of the VSC features from the System SLIO CPU	xyFCh

7.4.4 Diagnostics via OB start information

On an error the faulty system generates a diagnostics message for the CPU. Then the CPU calls the according diagnostics OB. Here the CPU operating system transfers start information to the local data of the OB. By evaluating the start information of the according OB you can get information about cause and location of the error. During runtime you can access the start information with the system function SFC 6 RD_SINFO. Please consider that you can even read the start information in the OB himself, because the data are temporary data.

Depending on the type of error, the following OBs are called in a diagnostics event:

- OB 82 on an error of an module at the EtherCAT slave station (Diagnostics alarm) *'Interrupt handling in the CPU' on page 139*
- OB 86 on failure respectively restart of an EtherCAT slave station *'Enter OB start information and call OB' on page 138*
- OB 57 Vendor specific interrupt

More information about OBs may be found in the online help of your programming tool or in the manual "Operation list".

7.4.5 Diagnostics via diagnostics buffer CPU respectively CP

Chapter 4.18 'Diagnostic entries' on page 92

7.4.6 Diagnostics via status LEDs

LEDs EtherCAT interface X4

BF2 red	BS green	MT yellow	Meaning
☐	☐	☐	Master is in INIT state
☐	◪ 2Hz	☐	Master is in Pre-Op state
☐	◪ pulses	☐	Master is in Safe-Op state
☐	■	☐	Master is in OP state
X	X	☐	There is no maintenance event pending.
X	X	■	There is a maintenance event pending. More may be found in the diagnostics data
☐	X	X	There is no error on the EtherCAT bus pending
■	X	X	■ EtherCAT bus error, no connection to sub net ■ wrong transfer rate ■ Full-duplex-transmission is not activated
◪ 1Hz	X	X	■ Failure of a connected IO device ■ At least one IO device cannot be reached (topology mismatch) ■ Faulty configuration
◪ 4s on, 1s off	☐	◪ 4s on, 1s off	Error in configuration: ■ 0xEA64 was added to the diagnostics buffer ■ Additionally the SF LED of the CPU is on
◪ 4Hz	☐	◪ 4Hz	The alternate blinking indicates that a firmware update of the EtherCAT master is executed.
■	■	■	Firmware update of the EtherCAT master is finished without error.
not relevant: X			

L/A (Link/Activity) green	S (Speed) green	Meaning
■	X	The EtherCAT master is physically connected to the Ethernet.
□	X	There is no physical connection.
◪ flickers	X	Shows Ethernet activity of the EtherCAT master.
■	■	The Ethernet interface of the EtherCAT master has a transfer rate of 100Mbit.
■	□	The Ethernet interface of the EtherCAT master has a transfer rate of 10Mbit.
not relevant: X		

7.5 Interrupt behaviour

7.5.1 Overview

Once a fault in the EtherCAT system occurs, it is made on OBs and SSLs available. Here, the SSLs are refreshed respectively OBs are called.

7.5.2 Interrupt types

Interrupt types

- MANUFACTURER_SPECIFIC_ALARM_MIN (0x0020 or 0x0021) - OB 57
- PROZESS_ALARM (0x0002) - OB 40 (process interrupt)
- BUS_STATUS_CHANGED (0x8001) - OB 86
- DIAGNOSE_ALARM_GEHEND (0x000C) - OB 82 (diagnostics interrupt going)
- DIAGNOSE_ALARM_KOMMEND (0x0001) - OB 82 (diagnostics interrupt coming)
- SLAVE_STATUS_CHANGED (0x8002) - OB 86
- TOPOLOGY_MISMATCH (0x8004) - OB 86
- TOPOLOGY_OK (0x8003) - OB 86
- DC_STATUS (0x8005)
- BUS_CYCLE_STATUS (0x8006) - OB82
- BUS_STATUS (0x8007) - OB86

7.5.2.1 MANUFACTURER_SPECIFIC_ALARM_MIN (0x0020 or 0x0021) - OB 57

Properties

Triggering event

- EtherCAT CoE emergency

Supplied data

- Slave address
- CoE emergency

Conditions

- The error code of the CoE emergency has to come from a VIPA slave station.
  - The error code of the CoE emergency must diver to 0x0000.
  - The error code of the CoE emergency must diver to 0xA000.
  - The error code of the CoE emergency must diver to 0xA001.
  - The error code of the CoE emergency must diver to 0xFF00.
  - If the error code is 0xFF00, then the 2. byte must be equal to 1 or 2.
- The error code of the CoE emergency has to come from another slave station.
  - Each emergency is reported as OB 57.
- A CoE emergency occurred during an topology change.
  - The error code of the CoE emergency must diver to 0x0000.
  - The error code of the CoE emergency must diver to 0xA000 and 0xA001.

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
Eventless	BYTE	0x11
FLT_ID	BYTE	0x5C
PrioLevel	BYTE	0x02
OBNo	BYTE	57
Reserved1	BYTE	0xCC
IoFlag	BYTE	0x54 or 0x55 (depending on the address type of the alarm-triggering module)
Info1	WORD	Diagnostics address of the slave
Info2	WORD	Error code of CoE emergency
Info3	WORD	Slave state of CoE emergency
User1	WORD	InterruptPrio, InterruptRef
User2	WORD	EtherCAT slave address

Update SSL data

Manufacturer specific interrupts do not change SSLs.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	OBNo.	PK:	Dat ID ½	Info1	Info2	Info3
0x115C	57	0x02	0x54CC	Diagnostics address of the slave	Interrupt type	Error code CoE emergency

7.5.2.2 PROZESS_ALARM (0x0002) - hardware interrupt - OB 40

Properties

Triggering event

- EtherCAT CoE emergency from a VIPA slave station

Supplied data

- Slave address
- CoE emergency

Conditions

- The error code of the CoE emergency must be equal to 0xFF00 and the CoE emergency has to come from a VIPA slave station.
- The 2. byte of *MEF* must be 1.

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0x11
FLT_ID	BYTE	0x41
PrioLevel	BYTE	Priority of the OB 40
OBNo	BYTE	40
Reserved1	BYTE	reserved
IoFlag	BYTE	0x54 or 0x55 (depending on the address type of the alarm-triggering module)
Info1	WORD	Diagnostics address of the slave
Info2	WORD	Error code of CoE emergency
Info3	WORD	Slave state of CoE emergency
User1	WORD	Alarmprio, AlarmRef
User2	WORD	EtherCAT slave address

Update SSL data

Hardware interrupts do not change SSLs.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

There is no diagnostics buffer entry.

7.5.2.3 BUS_STATUS_CHANGED (0x8001) - OB 86

Properties

Triggering event

- EtherCAT bus state was changed.

Supplied data

- Old and new state of the master and the number of slave modules, which are not in master state.

Conditions

- none

Interrupt handling in the CPU

In the event the master switches to "Operational" *Chapter 7.1.2 'EtherCAT State Machine' on page 122*, OB86 is released. Via its event class you can see, whether all configured slave stations have carried the state change. Should any or all slave stations are not be able to establish the state to "Operational", so you can check this via a SSL.

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0xEC on restoration or 0xED on failure or other VusStateChanged
FLT_ID	BYTE	0x10 failure or restoration with all slaves, 0x11 restoration with missing salve(s), 0x20 other BusStateChanged
PrioLevel	BYTE	Priority of the OB86
OBNo	BYTE	86
Reserved1	BYTE	1, if slave available, otherwise 0
IoFlag	BYTE	0x54 at input address in ZInfo1, 0x55 at output address
Info1	WORD	0xXXYY: XX=OldState, YY=NewState
Info2	WORD	Diagnostics address of the master
Info3	WORD	Number of missing salves
User1	WORD	0xXXYY: XX=InterruptPrio, YY=InterruptRef
User2	WORD	EtherCAT slave address

Chapter 7.4 'EtherCAT Diagnostics' on page 126

Update SSL data

In the SSL xy94 the corresponding bits are updated for each slave. Each to the CPU reported state change as interrupt event generates a diagnostics buffer entry and may be read in the SSL 0xE0.

Update I/O peripheral structure

I/O state of the slaves and its modules are set to EA_STATUS_BG_VORHANDEN (module available) on restoration and EA_STATUS_BG_NICHTVORHANDEN (module do not exist) on failure.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0xEC10, 0xEC11, 0xED10 or 0xED20 (depends on state change)	PrioLevel of OB86	86	see OB-Start-info Reserved1, IOFlag	old and new state of the slave	Diagnostics address master	Number of slaves, which differ from the status of the master

7.5.2.4 DIAGNOSE_ALARM_GEHEND (0x000C) - diagnostics interrupt going

Properties

Triggering event

- EtherCAT CoE emergency from a VIPA slave station

Supplied data

- Slave address
- CoE emergency

Conditions

- The error code of the CoE emergency must be equal to 0x0000 ("no error respectively "error resolved") and the CoE emergency has to come from a VIPA slave station.

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0x38
FLT_ID	BYTE	0x42
PrioLevel	BYTE	Priority of the OB 82
OBNo	BYTE	82
Reserved1	BYTE	0xC5
IoFlag	BYTE	0x54
Info1	WORD	Diagnostics address of the slave
Info2	WORD	Error code of CoE emergency
Info3	WORD	Slave state of CoE emergency
User1	WORD	InterruptPrio, InterruptRef
User2	WORD	EtherCAT slave address

Update SSL data

In SSL 0694 and 0692 the corresponding bit is updated for each slave.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0x3842	PrioLevel of OB 82	82	0xC554	Diagnostics address slave	EtherCAT error code	Slave state

7.5.2.5 DIAGNOSE_ALARM_Kommend (0x0001) - diagnostics interrupt coming

Properties

Triggering event

- EtherCAT CoE emergency from a VIPA slave station

Supplied data

- Slave address
- CoE emergency

Conditions

- The error code of the CoE emergency must diver to 0x0000
- The error code of the CoE emergency must diver to 0xA000 and 0xA001

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0x39
FLT_ID	BYTE	0x42
PrioLevel	BYTE	Priority of the OB 82
OBNo	BYTE	82
Reserved1	BYTE	0xC5
IoFlag	BYTE	0x54
Info1	WORD	Diagnostics address of the slave
Info2	WORD	Error code of CoE emergency
Info3	WORD	Slave state of CoE emergency
User1	WORD	InterruptPrio, InterruptRef
User2	WORD	EtherCAT slave address

Update SSL data

In SSL 0694 and 0692 the corresponding bit is updated for each slave.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0x3942	PrioLevel of OB 82	82	0xC554	Diagnostics address slave:	EtherCAT error code	Slave state

7.5.2.6 SLAVE_STATUS_CHANGED (0x8002) - OB 86

Properties

Triggering event

- Slave is not in the requested state.
- The application has successfully set a slave in a different state.

Supplied data

- current new state

*Especially when a master status change is performed , this message is **not** sent to the CPU, since the overall result for error slaves of the status change in the event BUS_STATE_CHANGED is transmitted.*

Interrupt handling in the CPU

For each slave the current state is stored inside the CPU.

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0xEC on restoration or 0xED on failure or other VusStateChanged
FLT_ID	BYTE	0x12 failure or restoration, 0x22 other BusState-Changed
PrioLevel	BYTE	Priority of the OB 86
OBNo	BYTE	86
Reserved1	BYTE	1, if slave available, otherwise 0
IoFlag	BYTE	0x54 at input address in ZInfo1, 0x55 at output address
Info1	WORD	0xXXYY: XX=OldState, YY=NewState
Info2	WORD	Diagnostics address of the slave

Structure element	Data type	Description
Info3	WORD	AI Status Code
User1	WORD	0xXXYY: XX=InterruptPrio, YY=InterruptRef
User2	WORD	EtherCAT slave address

Update SSL data

In the SSL xy94 the corresponding bits are updated for each slave. Each to the CPU reported state change as interrupt event generates a diagnostics buffer entry and may be read in the SSL 0xE0.

Update I/O peripheral structure

I/O state of the slaves and its modules are set to EA_STATUS_BG_VORHANDEN (module available) on restoration and EA_STATUS_BG_NICHTVORHANDEN (module do not exist) on failure.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0xEC10, 0xEC11, 0xED10 or 0xED20 (depends on state change)	PrioLevel of OB 86	86	see OB-Startinfo Reserved1, IOFlag	old and new state of the slave	Diagnostics address master	Number of slaves, which differ from the status of the master

7.5.2.7 TOPOLOGY_MISMATCH (0x8004) - OB 86

Properties

Triggering event

- Topology differs from the configured topology. The Interrupt is only triggered with an existing configuration.

Supplied data

- none

Conditions

- none

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0xED
FLT_ID	BYTE	0x30
PrioLevel	BYTE	Priority of the OB 86
OBNo	BYTE	86
Reserved1	BYTE	0
IoFlag	BYTE	0
Info1	WORD	0
Info2	WORD	Diagnostics address of the master
Info3	WORD	0
User1	WORD	0
User2	WORD	0

Update SSL data

In the SSL xy94 the difference of set point and effective value is entered.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0xED30	PrioLevel of OB 86	86	0x0000	0	Diagnostics address master	0

7.5.2.8 TOPOLOGY_OK (0x8003) - OB 86

Properties

Triggering event

- Interrupt is triggered, if the topology on the bus corresponds to the configured topoly.

Supplied data

- none

Conditions

- none

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0xED
FLT_ID	BYTE	0x30

Structure element	Data type	Description
PrioLevel	BYTE	Priority of the OB 86
OBNo	BYTE	86
Reserved1	BYTE	0
IoFlag	BYTE	0
Info1	WORD	0
Info2	WORD	Diagnostics address of the master
Info3	WORD	0
User1	WORD	0
User2	WORD	0

Update SSL data

In the SSL xy94 the difference of set point and effective value is entered.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0xED30	PrioLevel of OB 86	86	0x0000	0	Diagnostics address master	0

7.5.2.9 DC_STATUS (0x8005) - OB82

If a change in sync status of a DC slave or the master is recognized by the master, the EtherCAT master sends an appropriate interrupt to the CPU. This then triggers, if available, an OB 82, and writes an entry in the diagnostic buffer.

Interrupt handling in the CPU

Enter OB start information and call OB

Structural element	Data type	Description
EventClass	BYTE	0xEC: Incoming event 0xED Outgoing event
FLT_ID	BYTE	0x50: At least one DC is not synchronized
PrioLevel	BYTE	Priority of the OB82
OBNo	BYTE	82
Reserved1	BYTE	reserved
DatId	WORD	0x0000
Info1	WORD	0x0000

Structural element	Data type	Description
Info2	WORD	Diagnostic address of the master
Info3	WORD	0: DC status change master 1: DC status change slave station

Update SSL data

In the SSL xy94 the corresponding bits are updated for each slave. Each to the CPU reported state change as interrupt event generates a diagnostics buffer entry and may be read in the SSL 0xE0.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	DatId	Info1	Info2	Info3
0xEC50 0xED50	PrioLevel of OB82	82	0x0000	0x0000	Diagnostics address master	0: DC status change master 1: DC status change slave station

7.5.2.10 BUS_CYCLE_STATUS (0x8006) - OB 82

Properties

In the case that the bus cycle time can not be complied with, when reaching a maximum value (penalty) BUS_CYLCE_STATUS_CYCLE_DURATION_TOO_LONG interrupt is triggered. If the bus cycle is OK again, a BUS_CYLCE_STATUS_CYCLE_DURATION_OK interrupt is triggered. The monitoring and recording of bus cycle time violations follows these rules:

- If a bus cycle time violation occurs, *Penalty* is incremented by 3.
- If there is no violation, *Penalty* is decremented by 1.
- If *Penalty* reaches the value 9, the interrupt BUS_CYCLE_STATUS with cause BUS_CYLCE_STATUS_CYCLE_DURATION_TOO_LONG is sent to the CPU. After that no further interrupts of this type are reported on other bus cycle time violations.
- If *Penalty* reaches the value 0, the interrupt BUS_CYCLE_STATUS with cause BUS_CYLCE_STATUS_CYCLE_DURATION_OK is sent to the CPU.

OB82_EV_CLASS	OB82_FLT_ID	Description
B#16#EC	B#16#40	Bus cycle time violation resolved
OB82_Z1:	Diagnostics address of the EtherCAT IO system	
OB82_Z2:	0	
OB82_Z3:	0	

OB82_EV_CLASS	OB82_FLT_ID	Description
B#16#ED	B#16#40	Bus cycle time violation occurred
OB82_Z1:	Diagnostics address of the EtherCAT IO system	
OB82_Z2:	0	
OB82_Z3:	0	

7.5.2.11 BUS_STATUS (0x8007) - OB 86

In the case that multiple telegrams can not be received in the master, when reaching a maximum value (penalty) BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE interrupt is triggered. If telegrams can be received again, BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE_OK interrupt is triggered. The monitoring and recording of receive time-out in the master follows these rules:

- If a receive time-out occurs, *Penalty* is incremented by 3.
- If there is no receive time-out, *Penalty* is decremented by 1.
- If *Penalty* reaches the value 9, the interrupt BUS_STATUS with cause BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE_TIMEOUT is sent to the CPU. After that no further interrupts of this type may be reported on other receive time-outs.
- If *Penalty* reaches the value 0, the interrupt BUS_STATUS with cause BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE_OK is sent to the CPU. This interrupt is only sent once an interrupt BUS_STATUS with the cause BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE_TIMEOUT has been reported to the CPU, before.

OB86_EV_CLASS	OB86_FLT_ID	Description
B#16#EC	B#16#80	Bus error resolved
OB86_Z1:	Diagnostics address of the EtherCAT IO system	
OB86_Z2:	0	
OB86_Z3:	Bit 11 to 14: System ID EtherCAT network - 100	
Bit 15: 1 - flag bit for EtherCAT		

OB86_EV_CLASS	OB86_FLT_ID	Description
B#16#ED	B#16#80	Bus error occurred (receive timeout)
OB86_Z1:	Diagnostics address of the EtherCAT IO system	
OB86_Z2:	0	
OB86_Z3:	Bit 11 to 14: System ID EtherCAT network - 100	
Bit 15: 1 - flag bit for EtherCAT		

7.6 Firmware update

EtherCAT master

Chapter 4.12 'Firmware update' on page 83

EtherCAT slave station

Firmware update via *SPEED7 EtherCAT Manager*. More may be found in the according manual respectively online help. Chapter 7.9.11 'Firmware update - VIPA System SLIO IM 053-1EC00' on page 195

7.7 Accessing the object dictionary

7.7.1 Overview

Blocks

With the following blocks you have at run-time access to the object dictionary of the EtherCAT slave stations and EtherCAT master:

- FB 52 - Read SDO - Read access to object dictionary
- FB 53 - Write SDO - Write access to object dictionary

These are VIPA-specific blocks. More information about the usage of these blocks may be found in the manual "Operation list".

Please consider when accessing the object dictionary, depending on your master system, the byte order can be rotated!

7.8 Object dictionary

7.8.1 Object overview

Index	Object Dictionary Area
0x0000 ... 0x0FFF	Data Type Area Objects
0x1000 ... 0x1FFF	CoE Communication Area Objects
0x2000 ... 0x20FF	Generic Master Area Objects
0x2100 ... 0x21FF	Distributed Clocks Objects
0x3000 ... 0x3FFF	Slave Configuration / Information Objects
0x4000 ... 0x7FFF	Reserved Area
0x8000 ... 0x8FFF	CoE Slave Configuration Objects
0x9000 ... 0x9FFF	CoE Slave Information Objects
0xA000 ... 0xAFFF	CoE Slave Diagnosis Data Objects
0xB000 ... 0xEFFF	Reserved Area
0xF000 ... 0xFFFF	CoE Device Area Objects

7.8.2 CoE Communication Area Objects: 0x1000-0x1FFF

Index	Object Type	Name	Type
0x1000	VAR	Device Type	Unsigned32
0x1001	VAR	Error Register	Unsigned8
0x1008	VAR	Manufacturer Device Name String	VisibleString
0x1009	VAR	Manufacturer Hardware Version String	VisibleString
0x100A	VAR	Manufacturer Software Version String	VisibleString
0x1018	RECORD	Identity Object	Identity (0x23)
0x10F3	RECORD	History Object	History (0x26)

7.8.2.1 Device Type 0x1000

Subindex	Name	Type	Access	Value	Meaning
0x00	Device Type	Unsigned32	ro	0x00001389	0x00001389 means MDP

7.8.2.2 Device Name 0x1008

Subindex	Name	Type	Access	Value	Meaning
0x00	Device name	Visible string	ro	VIPA 31x	Name of the EtherCAT device

7.8.2.3 Hardware Version 0x1009

Sub-index	Name	Type	Access	Value	Meaning
0x00	Hardware version	Visible string	ro	"V MM.mm.ss.bb"" MM = Major Version mm = Minor Version ss = Service Pack bb = Build e.g. "V 01.05.02.02"	Hardware version of the EtherCAT device

7.8.2.4 Software Version 0x100A

Sub-index	Name	Type	Access	Value	Meaning
0x00	Software version	Visible string	ro	"V MM.mm.ss.bb"" MM = Major Version mm = Minor Version ss = Service Pack bb = Build e.g. "V 01.05.02.02"	Software version of the EtherCAT device

7.8.2.5 Identity Object 0x1018

Sub-index	Name	Type	Access	Value	Meaning
0x00	Number of Entries	Unsigned8	ro	0x04 (default)	
0x01	Vendor ID	Unsigned32	ro	0x0000022B (default)	Vendor ID of the EtherCAT device
0x02	Product Code	Unsigned32	ro	0x00001636 (default)	Product Code of the EtherCAT device
0x03	Revision Number	Unsigned32	ro	0x00000000 (default)	Revision Number (EtherCAT master software version)
0x04	Serial Number	Unsigned32	ro	0x00000000 (default)	Serial Number of the EtherCAT device

7.8.2.6 History Object 0x10F3

Sub-index	Name	Type	Access	Value	Meaning
0	Number of Entries	Unsigned8	ro		
1	Maximum number of Diag messages	Unsigned8	ro		
2	Subindex of newest Diag message	Unsigned8	ro		
3	Subindex of newest acknowledged Diag message	Unsigned8	r/w		
4	New Diag messages available	BOOL32	ro		

Sub-index	Name	Type	Access	Value	Meaning
5	Flags (UINT16, r/w)	Unsigned16	r/w	0	Bit 0 = 1: Enable Emergency sending (default = 0) Bit 1 = 1: Disable Storing Info Messages (default = 0) Bit 2 = 1: Disable Storing Warning Messages (default = 0) Bit 3 = 1: Disable Storing Error Messages (default = 0) Bit 4...15: reserved for future use
6 ... 255			ro		

7.8.2.6.1 Diagnosis Messages Object 0x10F3: 6-255

Byte-Offset	Name	Type	Access	Value	Meaning
0	Diag-Number	Unsigned32	ro		Bit 0...11: free use Bit 12...15 = 14: to be comp. with Emergency Error Bit 16...31 = 0: reserved Bit 16...31 = 0xFFFE: free use Bit 16...31 = 0xFFFF: reserved
4	Flags	Unsigned16	ro		Bit 0...3: Diag type (0 = Info, 1 = warning, 2 = error) Bit 4...15: reserved
6	Text ID	Unsigned16	ro		0 = no Text ID 1-65535 = Reference to a Text ID with formatted string
8	Time Stamp in ns (from DC)	Unsigned64	ro		
16	Flags parameter 1	Unsigned16	ro		
18	Parameter 1	several	ro		
N	Flags parameter n	Unsigned16	ro		
N+2	Parameter n	several	ro		

7.8.3 Generic Master Objects: 0x2000-0x20FF

Index	Object Type	Name	Type
0x2000	VAR	Master State Change Command Register	Unsigned32
0x2001	VAR	Master State Summary	Unsigned32
0x2002	RECORD	Bus Diagnosis Object	BusDiagnostic (0x40)
0x2005	RECORD	MAC Address	MACAddress (0x41)
0x2010	VAR	Debug Register	Unsigned48
0x2020	RECORD	Master Init. Parameters	MasterInitParm (0x42)

7.8.3.1 Master State Change Command Register 0x2000

Sub-index	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	r/w	0 = invalid 1 = init 2 = pre-operational 3 = bootstrap mode 4 = safe operational 8 = operational	

7.8.3.2 Master State Summary 0x2001

Sub-index	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	ro		Bit 0: = 1 Master OK Bit 1...3: reserved Bit 4...7: Master State Bit 8: Slaves in requested State Bit 9: Master in requested State Bit 10: Bus Scan Match Bit 11: reserved Bit 12: DC is enabled Bit 13: DC In-Sync Bit 14: DC Busy Bit 15: Reserved Bit 16: Link Up Bit 17...31: reserved

Master is OK if topology is Ok (Mismatch if slave exists, which is not configured). Master must be in *Op* state, slaves must be in *Op* state and *Distributed Clocks* must be *insync* if activated.

Parameter Flags Bit 12...15	Parameter Flags Bit 0...11	Type of Data	Data
0	CoE DataType e.g. 0x0007 = UINT32	Data Type	Data defined through CoE DataType
1	Length in Byte	Byte Array	Byte stream byData[Size]
2	Length in Byte	ASCII-String	String szString[Length] (not '\0' terminated)
3	Length in Byte	Unicode String	String wszString[Length/2] (not L'\0' terminated)
4	0	Text Id	Text Id (Word)

7.8.3.3 Bus Diagnosis Object 0x2002

Object Type: RECORD, Manufacturer Specific Identity 0x40

Subindex	Description	Type	Access
0x00	Number of Entries	Unsigned8	ro
0x01	Reserved	Unsigned16	ro
0x02	Configuration Checksum CRC32	Unsigned32	ro
0x03	Number of found Slave	Unsigned32	ro
0x04	Number of found DC Slave	Unsigned32	ro
0x05	Number of Slaves in Configuration	Unsigned32	ro
0x06	Number of Mailbox Slaves in Configuration	Unsigned32	ro
0x07	Counter: TX frames	Unsigned32	ro
0x08	Counter: RX frames	Unsigned32	ro
0x09	Counter: Lost frames	Unsigned32	ro
0x10	Counter: Cyclic frames	Unsigned32	ro
0x11	Counter: Cyclic datagrams	Unsigned32	ro
0x12	Counter: Acyclic frames	Unsigned32	ro

Subindex	Description	Type	Access
0x13	Counter: Acyclic datagrams	Unsigned32	ro
0x14	Clear Counters by writing 1 to bit(s) Bit 0: Clear all Counters Bit 1: Clear Tx Frame Counter (Idx 7) Bit 2: Clear Rx Frame Counter (Idx 8) Bit 3: Clear Lost Frame Counter (Idx 9) Bit 4: Clear Cyclic Frame Counter (Idx 10) Bit 5: Clear Cyclic Datagram Counter (Idx 11) Bit 6: Clear Acyclic Frame Counter (Idx 12) Bit 7: Clear Acyclic DataGram Counter (Idx 13) Bit 8...31: Reserved	Unsigned32	r/w

7.8.3.4 MAC Address 0x2005

Object Type: RECORD, Manufacturer Specific Identity 0x41

Subindex	Description	Type	Access
0x00	Number of Entries	Unsigned8	ro
0x01	Hardware	Unsigned48	ro
0x02	Red Hardware	Unsigned48	ro
0x03	Configuration Source	Unsigned48	ro
0x04	Configuration Destination	Unsigned48	

7.8.3.5 Debug Register 0x2010

Sub-index	Name	Type	Access	Value	Meaning
0x00	Debug Register	Unsigned38	r/w	Upper 16bit: 0: activate LinkError Messages 1...15: reserved Lower 32bit: Definition of parameter dwStateChangeDebug in structure EC_T_MASTER_CONFIG	

7.8.3.6 Master Init Parameters 0x2020

Object Type: RECORD, Manufacturer Specific Identity 0x42

Sub-index	Description	Type	Access
00	Number of Entries	Unsigned8	ro
01	EC_T_INITMASTERPARMS.dwVersion Application	Unsigned32	ro
02	dwVersion Master	Unsigned32	ro
03	EC_T_MASTER_CONFIG.nSlaveMultiplier	Unsigned32	ro
04	EC_T_MASTER_CONFIG.dwEcatCmdTimeout in millisec	Unsigned32	ro
05	EC_T_MASTER_CONFIG.dwEcatCmdMaxRetries	Unsigned32	ro
06	EC_T_MASTER_CONFIG.dwCycTimeout in millisec	Unsigned32	ro
07	EC_T_MASTER_CONFIG.dwEoeTimeout in millisec	Unsigned32	ro
08	EC_T_MASTER_CONFIG.dwFoeBusyTimeout in millisec	Unsigned32	ro
09	EC_T_MASTER_CONFIG.dwMaxQueuedEthFrames	Unsigned32	ro
10	EC_T_MASTER_CONFIG.dwMaxSlaveCmdPerFrame	Unsigned32	ro
11	EC_T_MASTER_CONFIG.dwMaxQueuedCoeSlaves	Unsigned32	ro
12	EC_T_MASTER_CONFIG.dwMaxQueuedCoeCmds	Unsigned32	ro
13	EC_T_MASTER_CONFIG.dwStateChangeDebug	Unsigned32	ro
14	EC_T_LINK_DEV_PARAM.szDriverIdent	VisibleString	ro
15	EC_T_LINK_DEV_PARAM.bPollingModeActive	Bool32	ro
16	EC_T_LINK_DEV_PARAM.bAllocSendFrameActive	Bool32	ro

7.8.4 Distributed Clocks Objects: 0x2100-0x21FF

Index	Object Type	Name	Type
0x2100	VAR	DC Slave Sync Deviation Limit	Unsigned32
0x2101	VAR	DC Current Deviation	Signed32
0x2102	VAR	DC Reserved	Unsigned32
0x2103	VAR	DC Reserved	Unsigned32

7.8.4.1 Distributed Clocks Slave Sync Deviation Limit 0x2100

Sub-index	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	ro	dwDevLimit	

7.8.4.2 Distributed Clocks Current Deviation 0x2101

Sub-index	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	ro	dwDeviation	

7.8.4.3 Reserved: 0x2102 / 0x2103

This value is reserved.

7.8.5 Slave specific objects

Slave Configuration / Information Objects: 0x3000-0x3FFF

Index	Object Type	Name	Type
0x3000 ... 0x3FFF	RECORD	Slave Configuration and Information Objects	SlaveCfgInfo (0x43)

CoE Slave Configuration Objects: 0x8000-0x8FFF

Index	Object Type	Name	Type
0x8000 ... 0x8FFF	RECORD	One index entry for each configured slave (from ESI)	SlaveCfg (0x45)

CoE Slave Information Objects: 0x9000-0x9FFF

Index	Object Type	Name	Type
0x9000 ... 0x9FFF	RECORD	One index entry for each connected BUS-slave (updated during BUS scan)	SlaveInfo (0x46)

CoE Slave Diagnosis Data Objects: 0xA000-0xAFFF

Index	Object Type	Name	Type
0xA000 ... 0xAFFF	RECORD	One subindex entry for each connected BUS-slave (cyclic updated)	SlaveDiag (0x47)

7.8.5.1 Slave Configuration and Information Object 0x3000-0x3FFF

Object Type: RECORD, Manufacturer Specific Identity 0x43

Subindex	Description	Type	Access
0	Number of Entries	Unsigned8	ro
1	Entry Valid	Bool32	ro
2	VendorId (Bus)	Unsigned32	ro
3	ProductCode (Bus)	Unsigned32	ro
4	Revision No (Bus)	Unsigned32	ro
5	Serial No (Bus)	Unsigned32	ro
6	Device Name (Config)	Visible_String[80]	ro
7	Auto Increment Address (Bus)	Unsigned16	ro
8	Physical Address (Bus)	Unsigned16	ro
9	Config Physical Address (Config)	Unsigned16	ro
10	Alias Address (Bus)	Unsigned16	ro
11	PortState (Bus)	Unsigned16	ro
12	DC Support (Bus)	Bool32	ro
13	DC Support 64Bit (Bus)	Bool32	ro
14	Mailbox Support (Config)	Bool32	ro
15	Requested State (slave instance)	Unsigned32	r/w
16	Current State (slave instance)	Unsigned32	ro
17	Error Flag Set (slave instance)	Bool32	ro
18	Enable Linkmessages (slave instance)	Bool32	r/w
19	Error code (slave instance)	Unsigned32	ro
20	Sync Pulse active (Config, slave instance)	Bool32	ro
21	DC Sync 0 Period (Config, slave instance)	Unsigned32	ro
22	DC Sync 1 Period (Config, slave instance)	Unsigned32	ro
23	SB Error Code (Bus Topology)	Unsigned32	ro
24	RX Error Counter Port 0 (Bus)	Unsigned16	ro
25	RX Error Counter Port 1 (Bus)	Unsigned16	ro
26	RX Error Counter Port 2 (Bus)	Unsigned16	ro
27	RX Error Counter Port 3 (Bus)	Unsigned16	ro
28	Forwarded RX Error Counter Port 0 (Bus)	Unsigned8	ro

Subindex	Description	Type	Access
29	Forwarded RX Error Counter Port 1 (Bus)	Unsigned8	ro
30	Forwarded RX Error Counter Port 2 (Bus)	Unsigned8	ro
31	Forwarded RX Error Counter Port 3 (Bus)	Unsigned8	ro
32	EtherCAT Processing Unit Error Counter (Bus)	Unsigned8	ro
33	PDI Error Counter (Bus)	Unsigned8	ro
34	Reserved	Unsigned16	ro
35	Lost Link Counter Port 0 (Bus)	Unsigned8	ro
36	Lost Link Counter Port 1 (Bus)	Unsigned8	ro
37	Lost Link Counter Port 2 (Bus)	Unsigned8	ro
38	Lost Link Counter Port 3 (Bus)	Unsigned8	ro
39	FMMU's supported (Bus)	Unsigned8	ro
40	Sync Managers supported (Bus)	Unsigned8	ro
41	RAM Size in kByte (Bus)	Unsigned8	ro
42	Port Descriptor (Bus)	Unsigned8	ro
43	ECS Type (Config)	Unsigned8	ro
44	Slave is optional (Config)	Bool32	ro
45	Slave is present (Bus)	Bool32	ro
46	Hot connect group ID	Unsigned32	ro

7.8.5.2 CoE Slave Configuration Objects: 0x8000-0x8FFF

Object Type: RECORD, Manufacturer Specific Identity 0x45

The configuration data contain information about the EtherCAT slaves.

Subindex	Description	Type	Access
0	Number of Entries	Unsigned8	ro
1	Fixed Station Address	Unsigned16	ro
2	Type	Visible_String[64]	ro
3	Name	Visible_String[64]	ro
4	Device Type	Unsigned32	ro
5	Vendor ID	Unsigned32	ro
6	Product Code	Unsigned32	ro
7	Revision Number	Unsigned32	ro
8	Version Number	Unsigned32	ro
33	Mailbox Out Size (if mailbox slave)	Unsigned16	ro
34	Mailbox In Size (if mailbox slave)	Unsigned16	ro

7.8.5.3 CoE Slave Information Objects: 0x9000-0x9FFF

Object Type: RECORD, Manufacturer Specific Identity 0x46

Information about the connected EtherCAT-Slaves can be found in the information data. They are available when the scan command has been executed.

Subindex	Description	Type	Access
0	Number of Entries	Unsigned8	ro
1	**Fixed Station Address** of the Nth EtherCAT slave found (same value as 0xF040: 01)	Unsigned16	ro
5	**Vendor ID** of the Nth EtherCAT slave found (entry 0x1018: 01 of the EtherCAT slave)	Unsigned32	ro
6	**Product Code** of the Nth EtherCAT slave found (entry 0x1018: 02 of the EtherCAT slave)	Unsigned32	ro
7	**Revision Number** of the first EtherCAT slave found (entry 0x1018: 03 of the EtherCAT slave)	Unsigned32	ro
8	**Version Number** of the first EtherCAT slave found (entry 0x1018: 04 of the EtherCAT slave)	Unsigned32	ro
32	**DL Status** (Register 0x110-0x111) of the Nth EtherCAT slave found.	Unsigned16	ro

7.8.5.4 CoE Slave Diagnosis Data Objects: 0xA000-0xAFFF

Object Type: RECORD, Manufacturer Specific Identity 0x47

The diagnostics data contain status and diagnostics information of the EtherCAT slaves or the connections of the EtherCAT slaves.

Subindex	Description	Type	Access
0	**Number of Entries**	Unsigned8	ro
1	**AL Status** (Register 0x130-0x131) of the Nth EtherCAT slave configured.	Unsigned16	ro
2	**AL Control** (Register 0x120-0x121) of the Nth EtherCAT slave configured.	Unsigned16	r/w

7.8.6 CoE Device Area Objects: 0xF000-0xFFFF

Index	Object Type	Name	Type
0xF000	RECORD	Modular Device Profile	DeviceProfile (0x48)
0xF002	RECORD	Detect Modules Command	DetectCmd (0x49)
0xF020 ... 0xF02F	RECORD	Configured Address List	ConfAddrList (0x50)
0xF040 ... 0xF04F	RECORD	Detected Address List	ConnAddrList (0x51)

7.8.6.1 Modular Device Profile Object 0xF000

Object Type: RECORD, Manufacturer Specific Identity 0x48

Subindex	Description	Type	Access
0	**Number of Entries**	Unsigned8	ro
1	**Index distance** between two modules. This value is always read as 1.	Unsigned16	ro
2	**Maximum number of EtherCAT slaves** connected to the EtherCAT bus. This value is read as 512.	Unsigned16	ro
3	**Available entries in objects 0x8xxx** (number of configured slaves).	Unsigned32	ro
4	**Available entries in objects 0x9xxx** (number of connected slaves).	Unsigned32	ro

7.8.6.2 Configured Address List Object 0xF020-0xF02F

Object Type: RECORD, Manufacturer Specific Identity 0x50

Subindex	Description	Type	Access
0	**Number of Entries**	Unsigned8	ro
1	Fixed Station Address of the first EtherCAT slave configured.	Unsigned16	ro
2	Fixed Station Address of the second EtherCAT slave configured.	Unsigned16	ro
…	…		ro
255	Fixed Station Address of the 255. EtherCAT slave configured.	Unsigned16	ro
0	**Number of Entries**	Unsigned8	ro

Subindex	Description	Type	Access
1	Fixed Station Address of the 256. EtherCAT slave configured.	Unsigned16	ro
…	...		

7.8.6.3 Detected Address List Object 0xF040-0xF04F

Object Type: RECORD, Manufacturer Specific Identity 0x51

Subindex	Description	Type	Access
0	**Number of Entries**	Unsigned8	ro
1	Fixed Station Address of the first EtherCAT slave detected.	Unsigned16	ro
2	Fixed Station Address of the second EtherCAT slave detected.	Unsigned16	ro
...	...		ro
255	Fixed Station Address of the 255. EtherCAT slave detected.	Unsigned16	ro
0	**Number of Entries**	Unsigned8	ro
1	Fixed Station Address of the 256. EtherCAT slave detected.	Unsigned16	ro
...	...		

7.9 Deployment *SPEED7 EtherCAT Manager*

7.9.1 Overview

Properties

- Serves to configure EtherCAT master.
- Is called within the *SPEED7 Studio*.
- Synchronizes the address areas with the *SPEED7 Studio*.
- Saves the configuration in the *SPEED7 Studio* project.
- Expanded functionality by choose-able *'Expert'* mode.

Functions

- Automatic configuration
- Manual configuration
- Diagnosis

Starting the *SPEED7 EtherCAT Manager*

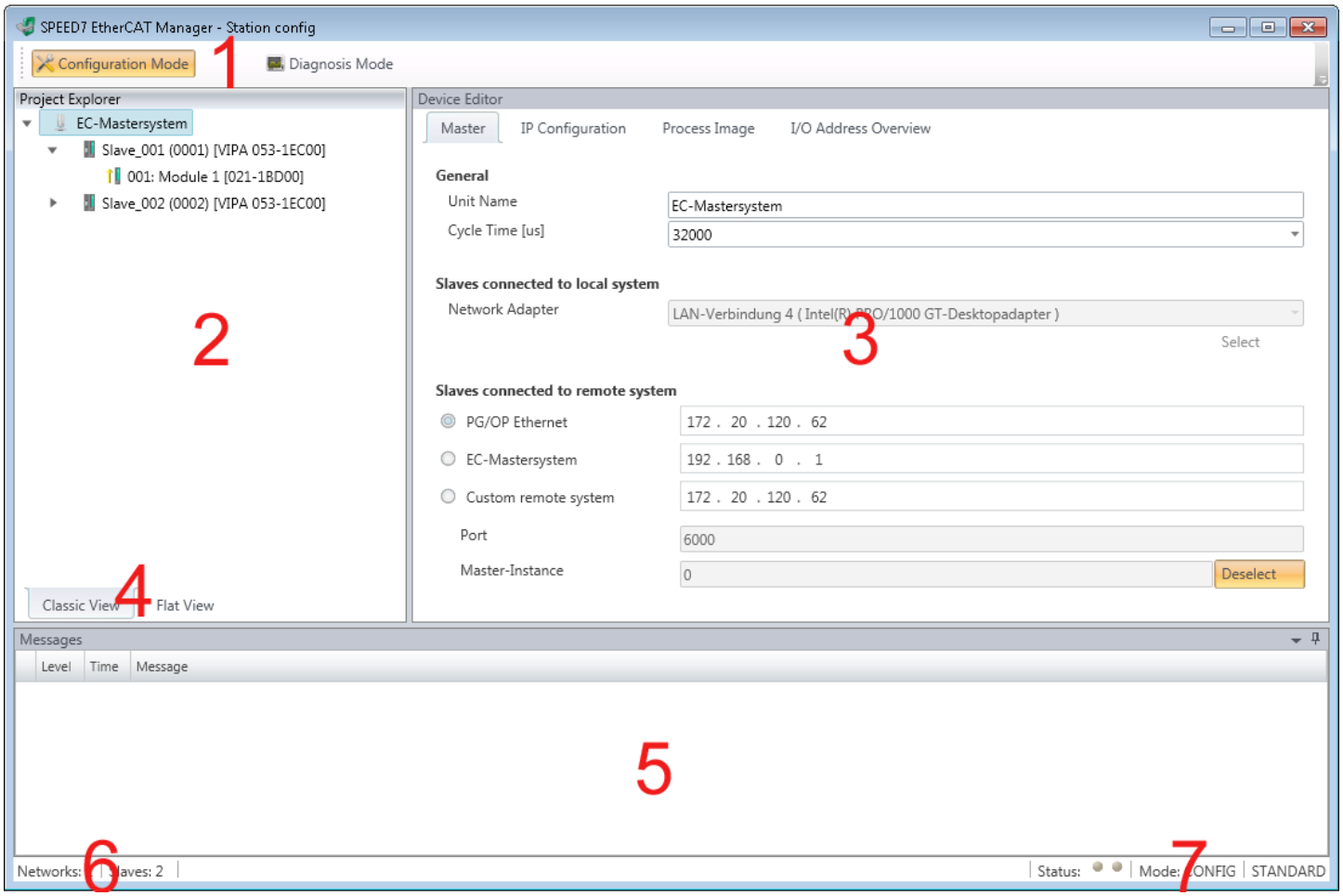
In *SPEED7 Studio* you can call via the *'Project tree'*, *'Field periphery'* of the EtherCAT CPU the *SPEED7 EtherCAT Manager* with *'Bus system properties'*.

Exit the *SPEED7 EtherCAT Manager*

By clicking at [X] in the *SPEED7 EtherCAT Manager*, the dialog is closed and the configuration is taken to the *SPEED7 Studio*.

Work environment of the *SPEED7 EtherCAT Manager*

The work environment of the *SPEED7 EtherCAT Manager* is divided into the following parts:



1 Tool bar: Here you can switch between *Configuration* and *Diagnosis*.
2 Project explorer: Here master and slave stations of your system are listed.
3 Device editor: Properties dialog of a device (parameter) respectively information area.
4 Selection of the view: In *Classic View* all the subordinate stations are shown indented. In *Flat View* all the subordinate stations are shown at the same level.
5 Here all the messages are listed.
6 In this section you can find the number of networks and slave stations.
7 Status area: With an online connection the 2 *Status* LEDs flash alternately. At *Modus* it is shown whether you are in operating mode *Diagnosis* or *Configuration*, followed by the selected dialog sight *Standard* respectively *Expert*.

'Expert mode'

In *SPEED7 Studio* you can call via the *'Project tree'*, *'Field periphery'* of the EtherCAT CPU the *SPEED7 EtherCAT Manager* with *'Bus system properties (Expert)'*. When enabled, the properties dialogs are extended accordingly. In *'Expert mode'* you will have the full scope of the *SPEED7 EtherCAT Manager*. Additionally in the status area *'Expert'* is shown.

Input area - numeric format

Some input fields have [Dec] respectively [Hex] buttons. By selecting the corresponding button you can select the input format *decimal* respectively *hexadecimal* for the input field.

7.9.2 Automatic configuration of a slave system

Precondition

The automatic configuration assumes that your EtherCAT system is mounted and can be reached on-line.

There are the following possibilities for on-line connection:

- Slaves connected to the local system
  - You are directly connected to a slave station via EtherCAT by means of a separate network adapter. Here the on-line connection is established by specifying the *Network Adapter*.
- Slaves connected to remote system
  - You are connected to the PG/OP channel of your CPU and can use this to access the EtherCAT master. The on-line connection is established by specifying *IP Address*, *Port* and *Master Instance*. With VIPA *Port* 6000 and *Master Instance* 0 is to be set.

Proceeding

1. Open if not already done the *SPEED7 EtherCAT Manager*
2. Click in the *'Project Explorer'* at *'EC-Mastersystem'*
3. Set depending on the on-line access in the *'Device Editor > Master'* as follows:
   - If you are directly locally connected to a slave station via EtherCAT by means of a separate network adapter, select your *Network adapter* and click at [Select].
   - If you are connected to the PG/OP channel of you CPU, please enter *IP Address*, *Port* and *Master Instance* and click at [Select]. With VIPA set *Port* to 6000 and *Master Instance* to 0.

   ⇨ The *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.

   When called from the SPEED7 Studio the IP address is taken from your project. If you change the IP address you need to adjust this in your project and start the SPEED7 EtherCAT Manager again!
4. Click in the *'Project Explorer'* at *'EC-Mastersystem'* and select from the context menu *'Scan EtherCAT network'*

   ⇨ You might be asked if you want to delete the existing slaves. Confirm with [Yes].

   Then the master is listed with its slaves and the associated PDO configuration in the *'Project Explorer'*, which was found by the network scan. The system can now be configured accordingly.

If there is no connection possible with the local master, the an anti virus software could block the connection. Then disabling the packet filter of the protocols of the network card in the anti virus software could help.

7.9.3 Manual configuration of a slave system

Precondition

With the manual configuration the system need not be built and connected online. The system can freely be configured in the *SPEED7 EtherCAT Manager*.

Proceeding

1. Open if not already done the *SPEED7 EtherCAT Manager*.
2. Click in the *'Project Explorer'* at *'EC-Mastersystem'* and select from the context menu *'Insert Slave'*.
   ⇨ A dialog opens to insert slave systems
3. Select the according slave from the list, enter the number of slaves and confirm with [OK].
   ⇨ The corresponding slave systems are inserted and can be configured now.

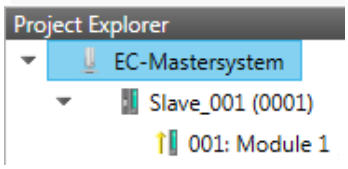
7.9.4 Configuration - EC-Mastersystem

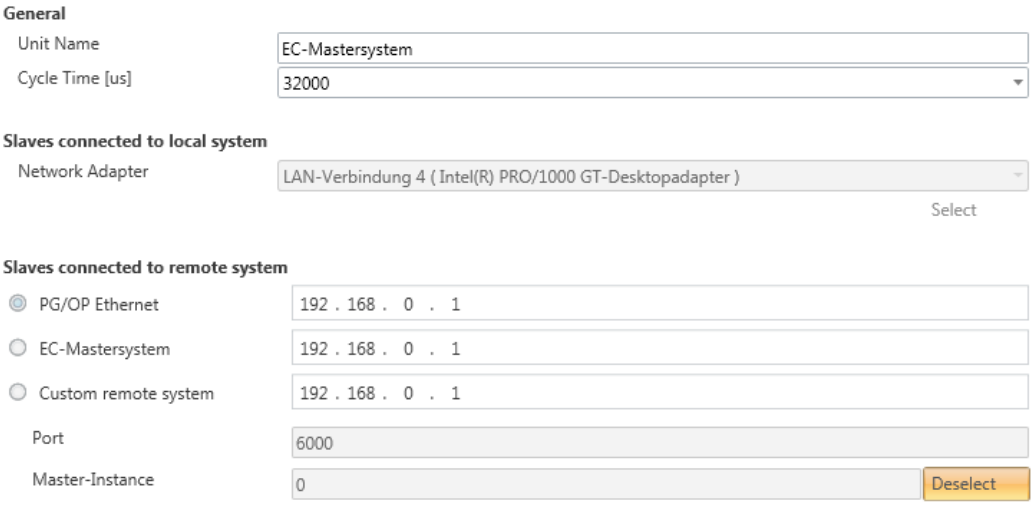
7.9.4.1 Preparation

Click in the Toolbar at [Configuration] and select *'EC-Mastersystem'* in the *'Project Explorer'*. As soon you have configured at least one slave station, the following registers are available:

↳ *Chapter 7.9.4.2 'Master' on page 163*
↳ *Chapter 7.9.4.5 'Process Image' on page 167*
↳ *Chapter 7.9.4.3 'Advanced Options (Expert mode)' on page 164*
↳ *Chapter 7.9.4.4 'Distributed Clocks (Expert mode)' on page 166*
↳ *Chapter 7.9.4.6 'I/O Address Overview' on page 168*

7.9.4.2 Master





Here you can perform master and bus-specific settings.

- General
  - Unit Name: Name of the master
  - Cyclic time: Interval in µs, in which the process data are read and written (PDO cycle time).
- Slaves connected to the local system
  - You are directly connected to a slave station via EtherCAT by means of a separate network adapter. Here the on-line connection is established by specifying the *Network Adapter*.
- Slaves connected to remote system
  - You are connected to the PG/OP channel of your CPU and can use this to access the EtherCAT master. The on-line connection is established by specifying *IP Address*, *Port* and *Master Instance*.

    IP Address: Enter the IP Address of the PG/OP channel of the remote CPU.

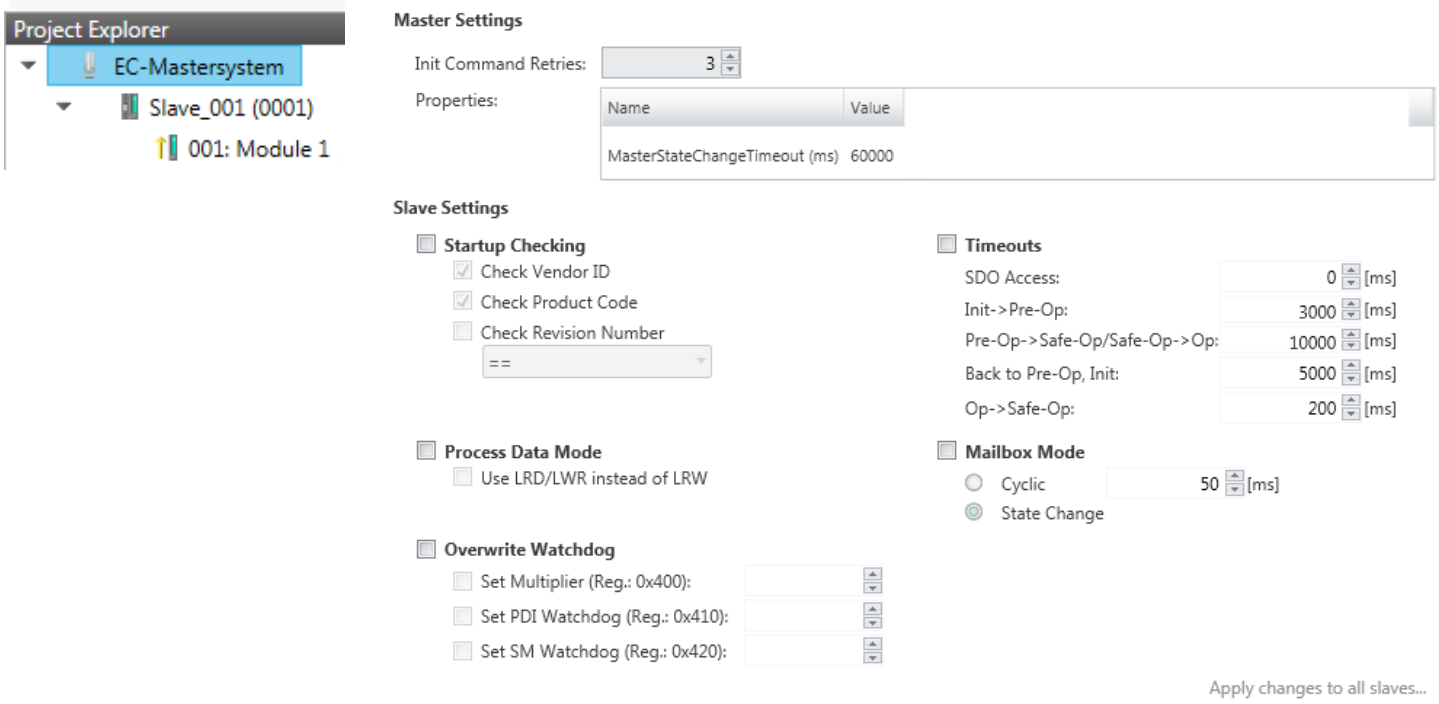
    Port: Port, over which the communication takes place with the remote CPU. With VIPA use Port 6000.

    Master-Instance: Serves for the master instance of the remote system. With VIPA the master instance is 0.

With [Select] the *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.

When called from the SPEED7 Studio the IP address is taken once from your project. If you change the IP address you need to adjust this in your project and then start the SPEED7 EtherCAT Manager again.

7.9.4.3 Advanced Options (Expert mode)



This dialog is only visible in the *'Expert mode'*! In this dialog the parameters of the master system can be adjusted and the default settings for all the slave stations can be defined.

- Master Settings
  - Init Command Retries: Number of retries, beyond which a transmission error is returned. (default: 3)
  - MasterStateChangeTimeout: Here you can define a timeout for the state change of the master and its slave stations (default: 60000ms). If the *MasterStateChangeTimeout* is too short, the EtherCAT master reports the error message 0xED21.
- Slave Settings
  - In this area default parameters can be applied for all the slave stations. The settings are applied for all slave stations as default stetting by clicking on [Apply changes (to all slaves)]. By selecting the slave station in the *'Project Explorer'* you always have the possibility to customize the slave parameters via the register *'Advanced Options'*.

- Slave Settings - Parameters
  - Start-up checking: Here you can define the items, the EtherCAT master has to check during the transition *'Init→Pre-Op'*.
  - Process Data Mode: Here you specify the command that should be used for process data access.

    'LRD/LWR:' Read access with **L**ogical-**R**ea**d** command to inputs and write access with **L**ogical-**Wr**ite command to outputs. This needs 2 frames.

    LRW: With one **L**ogical-**R**ea**d**-**L**ogical-**Wr**ite command inputs are read and also outputs are set. This needs 1 frame.

    Use *'LRD/LWR'* for cable redundancy and *'LRW'* for slave-to-slave-copy.
  - Overwrite Watchdog: Writes the configured value in the relevant register of the slave station. Here among others you can set the time of the *'SM Watchdog'* (SyncManager-Watchdog).

    Please note that even if a watchdog is present, this need not be indicated in the ESI file and this is shown as inactive!
  - Timeouts:

    'SDO Access': Internal master timeout for SDO access

    'Init→Pre-Op': Internal master timeout for slave state change from *Init* to *Pre-Op*

    'Pre-Op→Safe-Op/Safe-Op→Op': Internal master timeout for slave state change from *Pre-Op* to *Safe-Op* and then to *Op.*

    'Back to Pre-Op, Init': Internal master timeout for slave state change to *Pre-Op* and *Init*

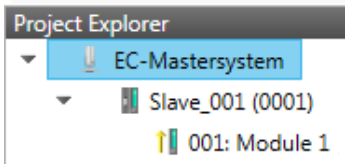
    'Op→Safe-Op': Internal master timeout for slave state change from *Op* to *Safe-Op* ↬ *Chapter 7.1.2 'EtherCAT State Machine' on page 122*
  - Mailbox Mode: The *'Mailbox'* is an a-cyclic communication channel. Here mostly *'Emergencies'* messages and *'SDOs'* are buffered. The way of accessing the just unread mailbox data can be specified here.

    'Cyclic': Interval in ms within which the mailbox is to be read (polling mode). If you want short interrupt response times, you should select the mode *'Cyclic'* and set a short time e.g. 1ms.

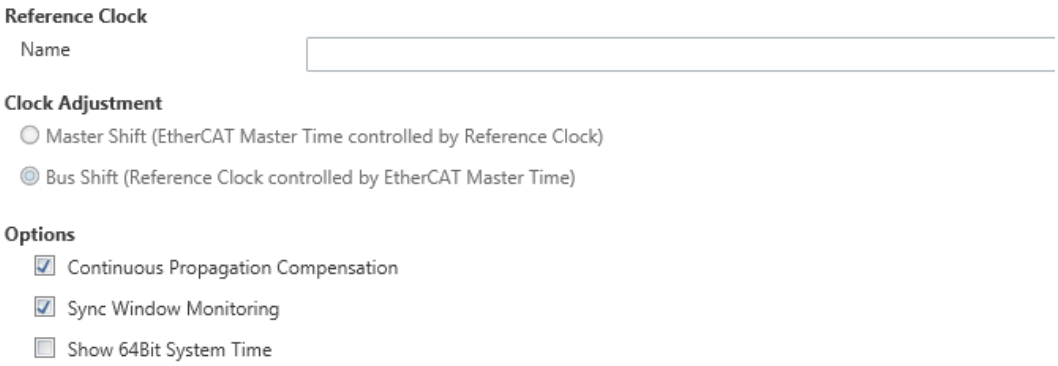
    'State change': The mailbox is read only on a state bit change.

- *When changing the 'Process Data Mode' you have to refresh the addresses in the Register 'Process Image'.*
- *If the Process Data Mode 'LRW' is used, the input and the output address of the EtherCAT process image must be identical. Here address leaks can occur between slave stations. If an EtherCAT address exceeds the maximum address area of the CPU, the current configuration gets invalid. You need to reduce the configuration or change to process data mode 'LRD/LWR'.*
- *If you use long cycle times (> 100ms) you should always accordingly raise the 'SM Watchdog'. Otherwise your slave station changes after laps of 'SM Watchdog' time to Safe-Op and releases OB 86. From now on you can only manually set the slave to Op!*

7.9.4.4 Distributed Clocks (Expert mode)



Due to the hardware, with local connections the function 'distributed clocks' is not supported.

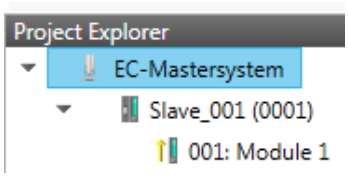


This dialog is only visible in the *'Expert mode'*! Here you can adjust the clock functionality accordingly. In EtherCAT "Distributed Clocks" means a logical combination of "clocks", which are located in the EtherCAT devices. With this there is the possibility to

locally provide a synchronized time in each bus device. If an EtherCAT device supports the Distributed Clocks functionality, it has its own clock. After PowerON this first locally works, based on an own pulse generator. By selecting an EtherCAT slave station, which has to provide the reference time, the distributed clocks can be synchronized. This reference clock so represents the system time.

- Reference clock: Here you get information about the clock, which provides the reference time.
  - Name: Name of the reference clock. Per default this is always the 1. slave station, which supports the "Distributed Clock (DC)" functionality.
- Clock adjustment
  - Master Shift: The EtherCAT master time is synchronized by the reference clock.
  - Bus Shift: The reference clock is synchronized by the EtherCAT master time.
- Options
  - Continuous Propagation Compensation: A command (datagram) will be inserted in the cyclic frame which allows the EtherCAT master to measure and compensate the propagation delay time by time.
  - Sync Window Monitoring: A command (datagram) will be inserted in the cyclic frame to read the ESC registers 0x092C. If this is selected the master will throw a notification about the state (*sync* respectively *out-of-sync*) of your system.
  - 64bit system time: Master supports slaves with 32bit and 64bit system time register (0x0910). If this is selected he will interpret it as 64bit system time.

7.9.4.5 Process Image



I/O addresses

Input addresses

Start address:

End Address:

Inputaddresses assigned (Byte). 20

Output addresses

Start address:

End Address:

Outputaddresses assigned (Byte). 0

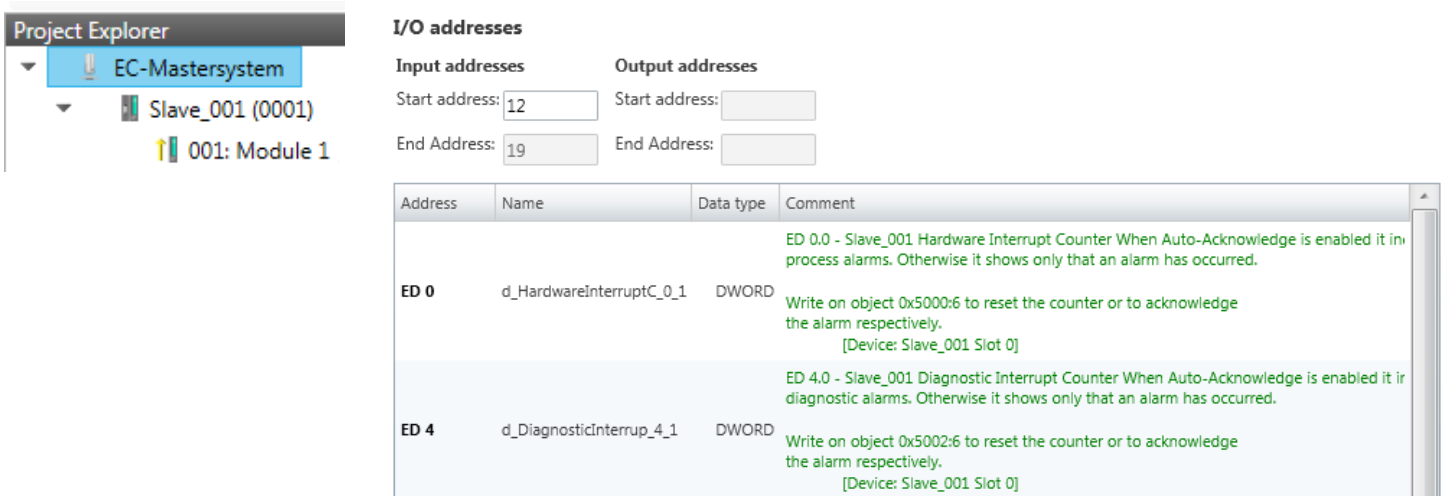
No.	Bus address	Slave	Module	Slot	S7 Input address	S7 Output address	EtherCAT input address	EtherCAT output address	T
1	1	Slave_001			0 - 7		9 - 16		V
2	1	Slave_001	Module 1	1	8		17		V
3	2	Slave_002			12 - 19		0 - 7		V

Here you have a list of S7 respectively EtherCAT addresses, which are used by the modules of all the slave stations. The *'S7 address'* corresponds to the address in the address area of the CPU. By entering a new *'Start Address'* you can adjust the S7 addressing of the input and output areas of the modules accordingly.

Information about the assignment of the in/output area can be found in the manual of your module.

The *'I/O Addresses EtherCAT'* are only visible in *'Expert mode'*! *'I/O Addresses EtherCAT'* are the offset addresses, which are used within the EtherCAT process image. You cannot change the address. You can use the addresses e.g. for EtherCAT network analysis.

7.9.4.6 I/O Address Overview



Here you have a list of addresses that are used by the I/O components of all the modules in the address area of the CPU. By entering a new *'Start address'* you can adjust the addressing of the input and output areas accordingly. You can edit *'Name'* and *'Comment'* by clicking at the corresponding entry.

Information about the assignment of the in/output area can be found in the manual of your module.

7.9.5 Configuration - slave station

7.9.5.1 Preparation

Click in the Toolbar at [Configuration] and select the Slave-Station *'Slave_...'* in the *'Project Explorer'*. The following registers are available now:

↳ *Chapter 7.9.5.2 'General' on page 169*

↳ *Chapter 7.9.5.3 'Modules' on page 170*

↳ *Chapter 7.9.5.4 'PDO Mapping' on page 171*

Group - if a group exists for this slave station ↳ *Chapter 7.9.9 'Grouping logic' on page 189*

↳ *Chapter 7.9.5.5 'Advanced Options (Expert mode)' on page 172*

↳ *Chapter 7.9.5.6 'Distributed Clocks (Expert mode)' on page 174*

↳ *Chapter 7.9.5.7 'Init Commands (Expert mode)' on page 175*

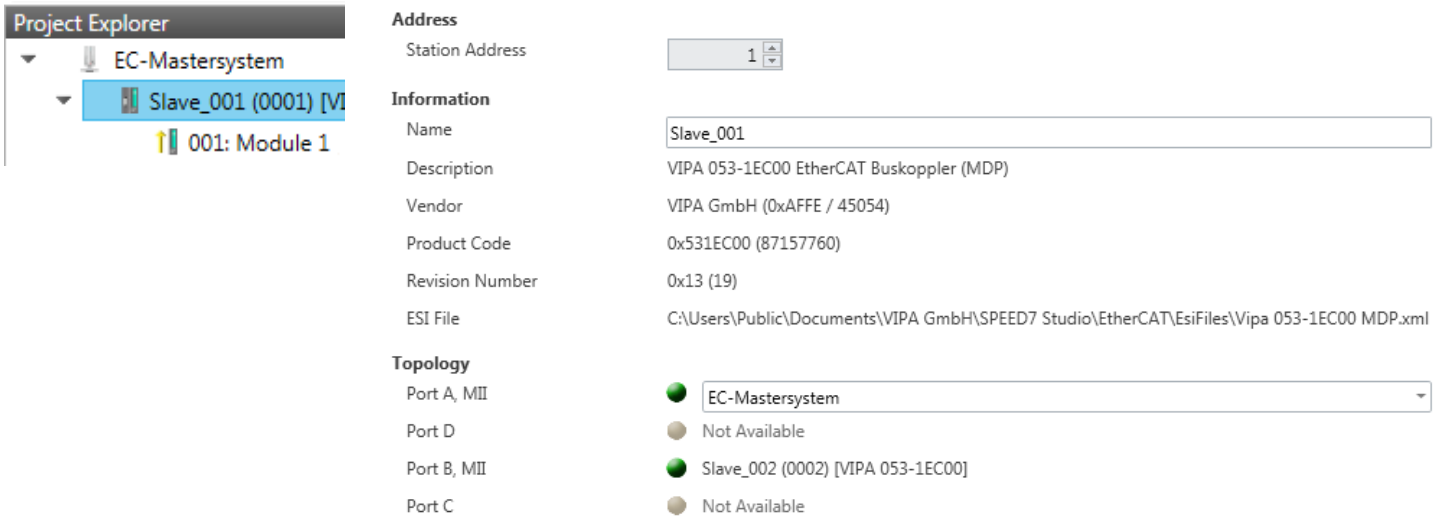
↳ *Chapter 7.9.5.8 'CoE Object Dictionary (Expert mode)' on page 177*

↳ *Chapter 7.9.5.9 'Process Image' on page 177*

↳ *Chapter 7.9.5.10 'I/O Address Overview' on page 178*

↳ *Chapter 7.9.5.11 'Parameter' on page 178*

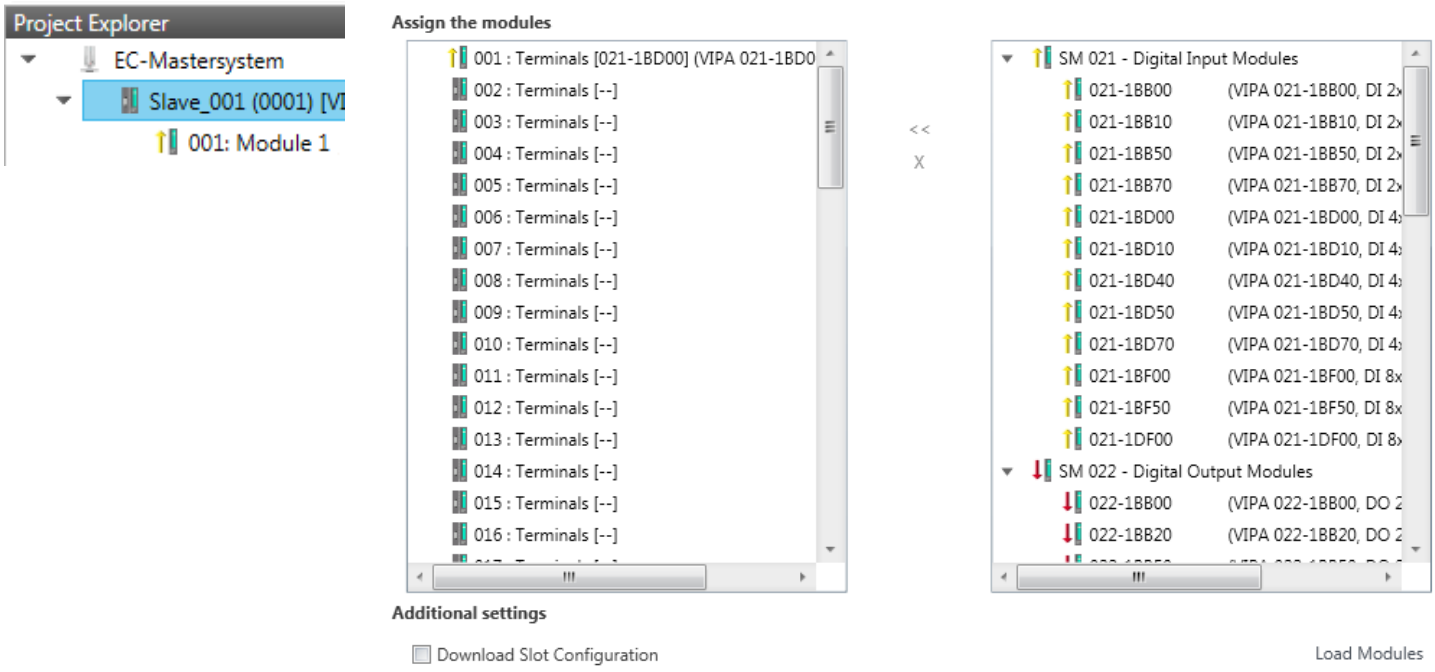
7.9.5.2 General



Here you can perform slave-specific settings such as assignment of name and address to a station. It is also possible to change the connection to the station.

- Address:
  - Station Address: EtherCAT address of the slave station.
- Information
  - Name: Name of the slave station can be assigned accordingly
  - Description: Description of the slave station
  - Vendor: Name of the vendor
  - Product Code: Internal product code of the slave station
  - Revision Number: Internal revision number of the slave station
  - ESI file: Path and name of the device file, in which the data of the slave station is stored.
- Topology
  - Port A / Port B: Here you will find the device that is connected to the corresponding port.

7.9.5.3 Modules



With an E-Bus slave this dialog is hidden. ↳ *Chapter 7.9.9 'Grouping logic' on page 189*

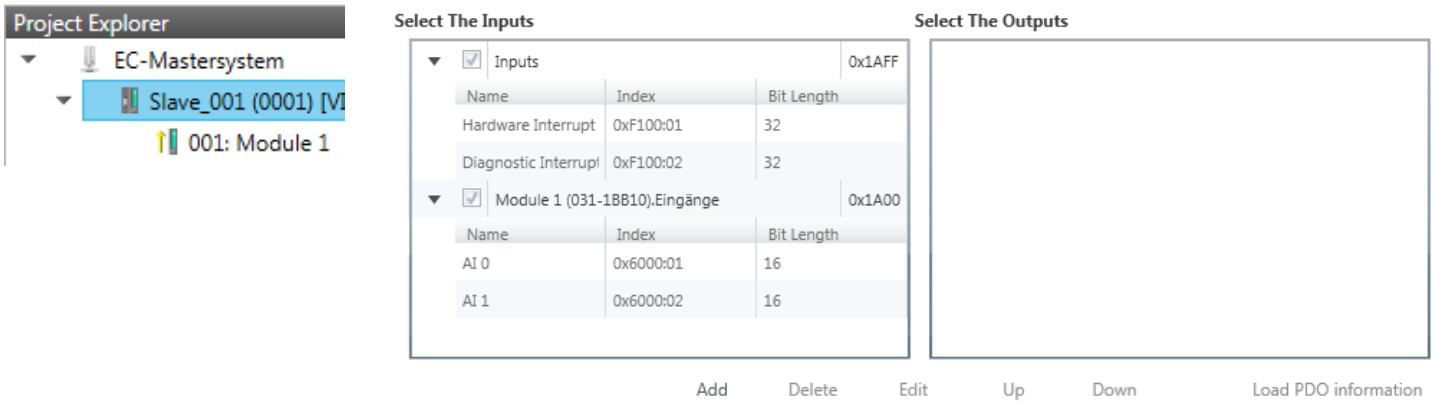
In this dialog you can assign modules to the appropriate slot.

- Assign module to a slot ("<<")
  Select your module from the list on the right and add it to a selected slot *'Terminals'* in the left list by clicking [<<]. This takes place according to the following rules:
  – If no modules are configured, the module is connected to the highlighted slot. Each additional module is inserted below.
  – If modules are already exist, the module is added to the highlighted slot and the following modules are moved accordingly.
- Remove module from a slot ("X")
  – Select from the left list the appropriate slot, which you want to disconnect from the module again and click at ["X"].

There is also the possibility to add or remove slave stations via the context menu of the 'Project Explorer'.

- Option field - *'Download slot configuration'*
  When enabled, an Init Command is created, which contains the slot configuration with the unique module identifier. During start-up of the slave station the slot configuration serves for comparison between configured and inserted modules. This can prevent misconfigurations.
- Button - [Load modules]
  With this function you can load the configuration from the EtherCAT master for the selected slave station.

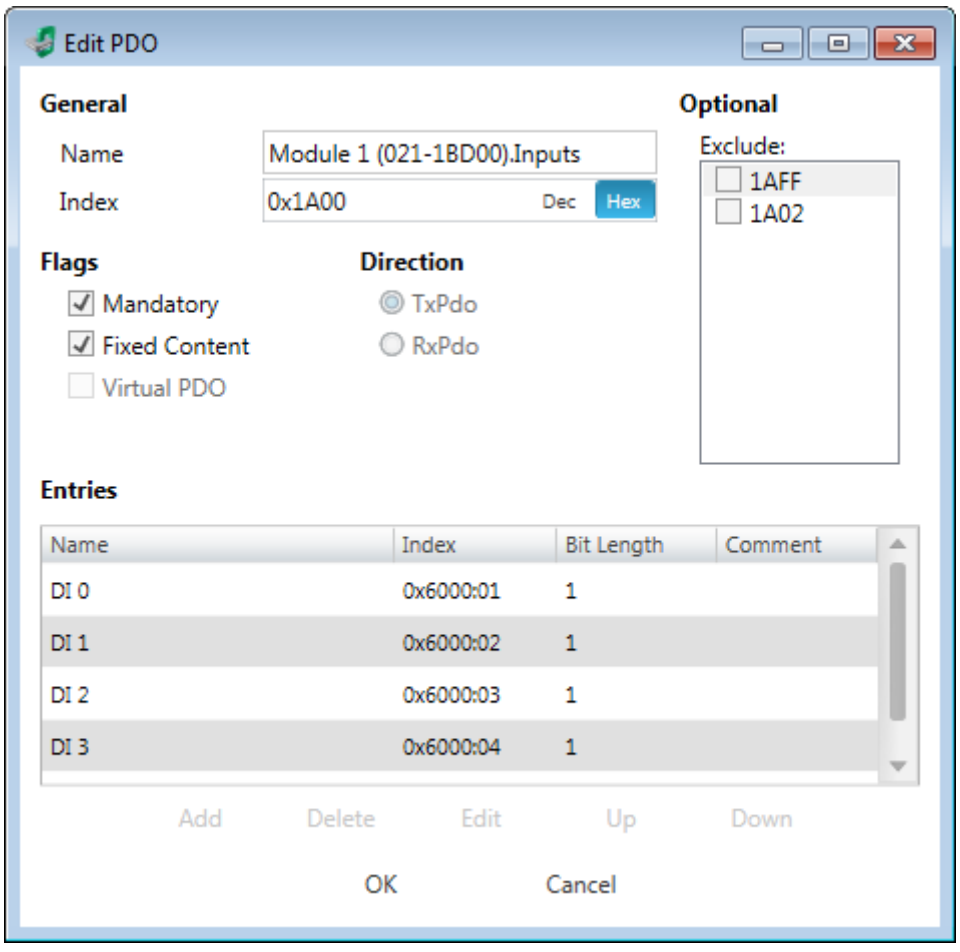
7.9.5.4 PDO Mapping



This dialog shows a list of the assigned PDOs. With some slave stations it is possible to activate respectively de-activate certain PDO configurations.

- Select the Inputs
  - If your slave station supports it, you can hide the corresponding input PDO from the configuration by disabling the checkbox.
- Select the Outputs
  - If your slave station supports it, you can hide the corresponding output PDO from the configuration by disabling the checkbox.

7.9.5.4.1 Edit PDO (Expert mode)

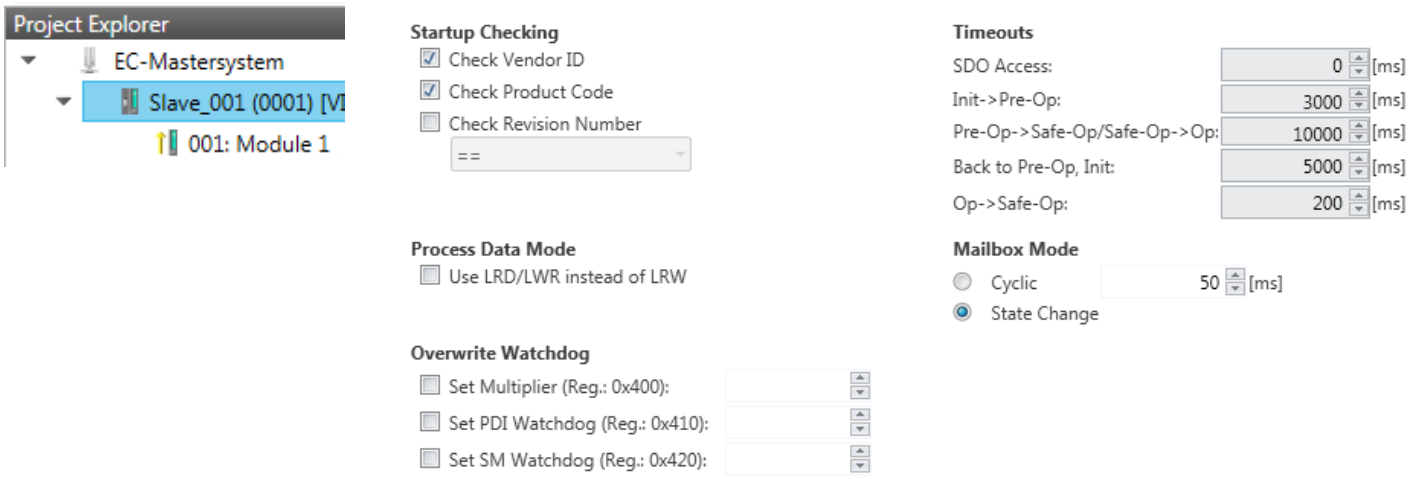


PDOs can only be edited in the *'Expert mode'*! Otherwise, the buttons are hidden.
With [Edit] the dialog *'Edit PDO'* opens.

- General
  - Name: Name of the PDO
  - Index: Index of the PDO (can be entered in hexadecimal or decimal)
- Flags
  - Mandatory: If activated the PDO cannot be deleted.
  - Fixed Content: If activated the content of the PDO is write protected. to create new or to edit existing PDOs you have to disable *'Fixed Content'*.
  - Virtual PDO: If activated the PDO has no entries.
- Direction
  - TxPDO: Send PDO of the slave station for input data.
  - RxPDO: Receive PDO of the slave station for output data.
- Optional
  - Exclude: Select the PDOs which cannot be activated if this PDO is activated.
- Entries
  - Here is the list of configured PDO entries shown.

After editing the PDOs, the addresses need to be re-calculated! For this jump to register 'Process Image' and click at [Recalculate].

7.9.5.5 Advanced Options (Expert mode)



This dialog is only visible in the *'Expert mode'*! Here you can make further adjustments to the slave station.

- Slave Settings - Parameters
  - Start-up checking: Here you can define the items, the EtherCAT master has to check during the transition *'Init→Pre-Op'*.
  - Process Data Mode: Here you specify the command that should be used for process data access.

    'LRD/LWR:' Read access with **L**ogical-**R**ea**d** command to inputs and write access with **L**ogical-**Wr**ite command to outputs. This needs 2 frames.

    LRW: With one **L**ogical-**R**ea**d**-**L**ogical-**Wr**ite command inputs are read and also outputs are set. This needs 1 frame.

    Use *'LRD/LWR'* for cable redundancy and *'LRW'* for slave-to-slave-copy.
  - Overwrite Watchdog: Writes the configured value in the relevant register of the slave station. Here among others you can set the time of the *'SM Watchdog'* (SyncManager-Watchdog).

    Please note that even if a watchdog is present, this need not be indicated in the ESI file and this is shown as inactive!
  - Timeouts:

    'SDO Access': Internal master timeout for SDO access

    'Init→Pre-Op': Internal master timeout for slave state change from *Init* to *Pre-Op*

    'Pre-Op→Safe-Op/Safe-Op→Op': Internal master timeout for slave state change from *Pre-Op* to *Safe-Op* and then to *Op.*

    'Back to Pre-Op, Init': Internal master timeout for slave state change to *Pre-Op* and *Init*

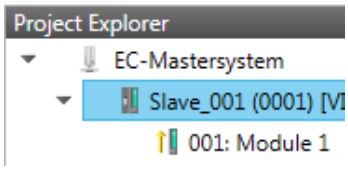
    'Op→Safe-Op': Internal master timeout for slave state change from *Op* to *Safe-Op* *Chapter 7.1.2 'EtherCAT State Machine' on page 122*
  - Mailbox Mode: The *'Mailbox'* is an a-cyclic communication channel. Here mostly *'Emergencies'* messages and *'SDOs'* are buffered. The way of accessing the just unread mailbox data can be specified here.

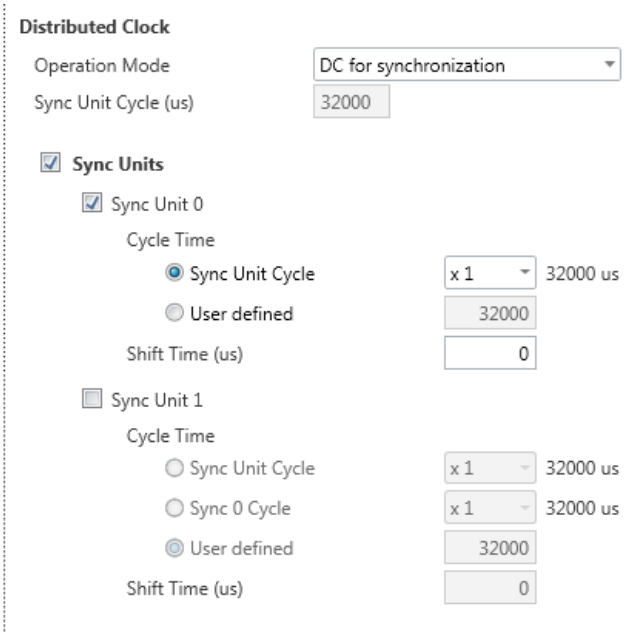
    'Cyclic': Interval in ms within which the mailbox is to be read (polling mode). If you want short interrupt response times, you should select the mode *'Cyclic'* and set a short time e.g. 1ms.

    'State change': The mailbox is read only on a state bit change.

- *When changing the 'Process Data Mode' you have to refresh the addresses in the Register 'Process Image'.*
- *If the Process Data Mode 'LRW' is used, the input and the output address of the EtherCAT process image must be identical. Here address leaks can occur between slave stations. If an EtherCAT address exceeds the maximum address area of the CPU, the current configuration gets invalid. You need to reduce the configuration or change to process data mode 'LRD/LWR'.*
- *If you use long cycle times (> 100ms) you should always accordingly raise the 'SM Watchdog'. Otherwise your slave station changes after laps of 'SM Watchdog' time to Safe-Op and releases OB 86. From now on you can only manually set the slave to Op!*

7.9.5.6 Distributed Clocks (Expert mode)





This dialog is only visible in the *'Expert mode'* if this is supported by your slave station! Here you can adjust the clock functionality accordingly. In EtherCAT "Distributed Clocks" (DC) means a logical combination of "clocks", which are located in the EtherCAT devices. With this there is the possibility to locally provide a synchronized time in each bus device. If an EtherCAT device supports the *Distributed Clocks* functionality, it has its own clock. After PowerON this first locally works, based on an own pulse generator. By selecting an EtherCAT slave station, which has to provide the reference time, the distributed clocks can be synchronized. This *reference clock* so represents the system time.

- Reference clock
  - Operation Mode: Here you can set the operation mode of the reference clock. More may be found in the manual of your slave station.
  - Sync Unit Cycle: Cycle time of the master. *Chapter 7.9.4 'Configuration - EC-Mastersystem' on page 163*
- Sync Units
  - Sync Unit 0

    - Cycle Time: Here you can specify the cycle time in relation to the *'Master Cycle'* or *'User defined'*.

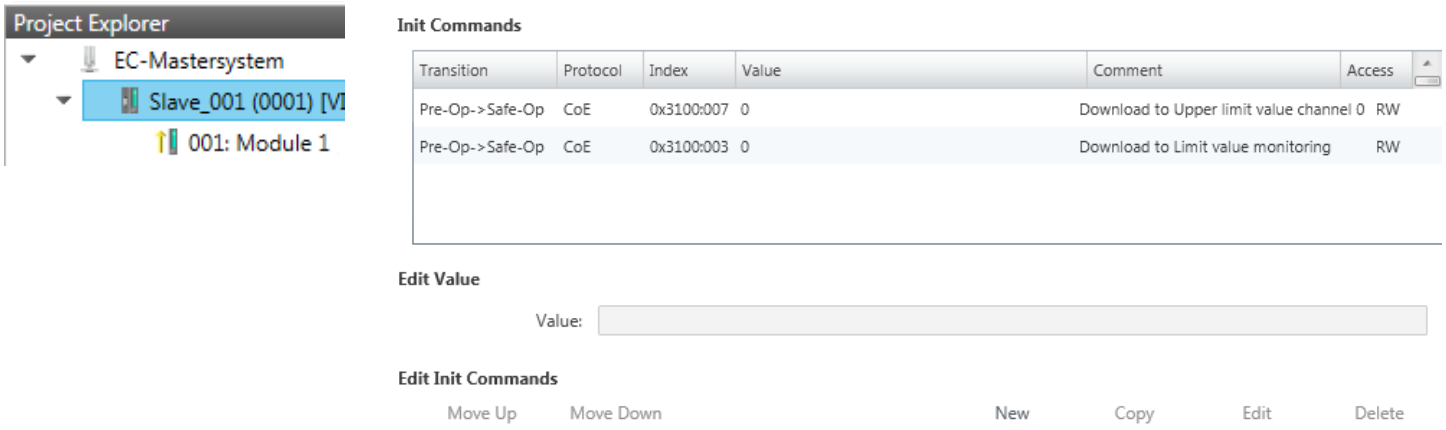
    - Time Shift: Specify here a time shift. This is used for fine adjustment.
  - Sync Unit 1

    - Cycle Time: Here you can specify the cycle time in relation to the *'Master Cycle'*, to the cycle of Sync Unit 0 *'Sync 0 Cycle'* or *'User defined'*.

    - Time Shift: Specify here a time shift. This is used for fine adjustment.

Due to the hardware with a local connection Distributed Clocks (connection via network adapter) is not supported!

7.9.5.7 Init Commands (Expert mode)



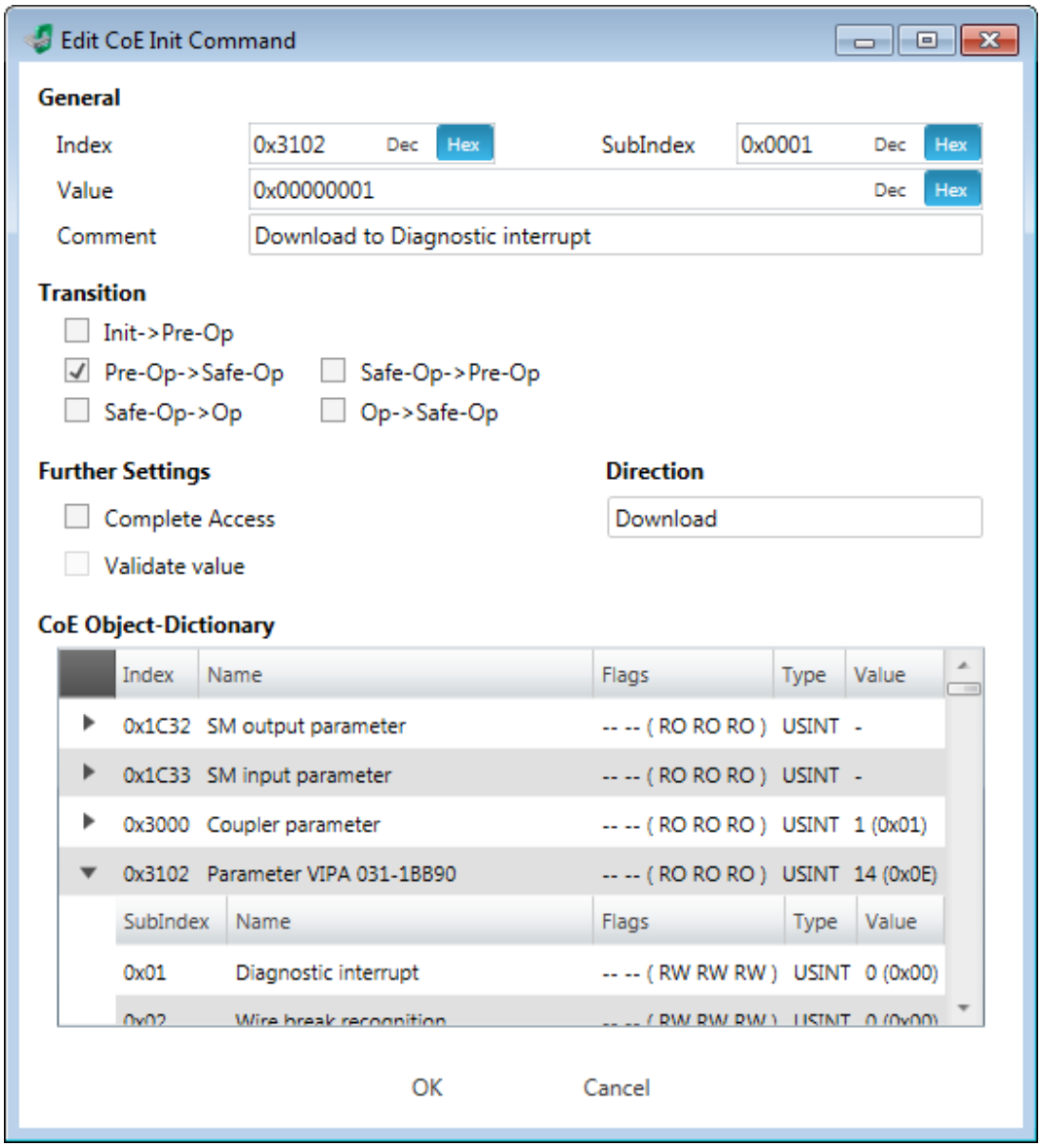
This dialog is only visible in the *'Expert mode'*!

- *For each parameter of a slave station or module, which differs from the standard setting you have to create an Init command!*
- *If a write access to an object in the configuration mode is performed, and the written value does not reflect to the default value of the object, so this command is automatically added to the 'Init Commands'.* *Chapter 7.9.5.8 'CoE Object Dictionary (Expert mode)' on page 177*

Here you can see a list of the current configured Init Commands and if it is allowed you can also add/edit/delete the commands.

- Init Commands: Init Commands come from the ESI file or are automatically generated on write access to CoE objects or can be created by the user. You either have full-access (RW = Read/Write) or only read access (RO = Read-only). Init commands from ESI files are automatically listed here. These cannot be changed or deleted.
- Button
  - New, Copy, Edit, Delete: Used for changing Init Commands.
  - Move Up, Move Down: Moving the selected Init Command up or down.

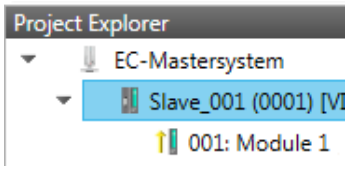
7.9.5.7.1 CoE Init Command (Expert mode)

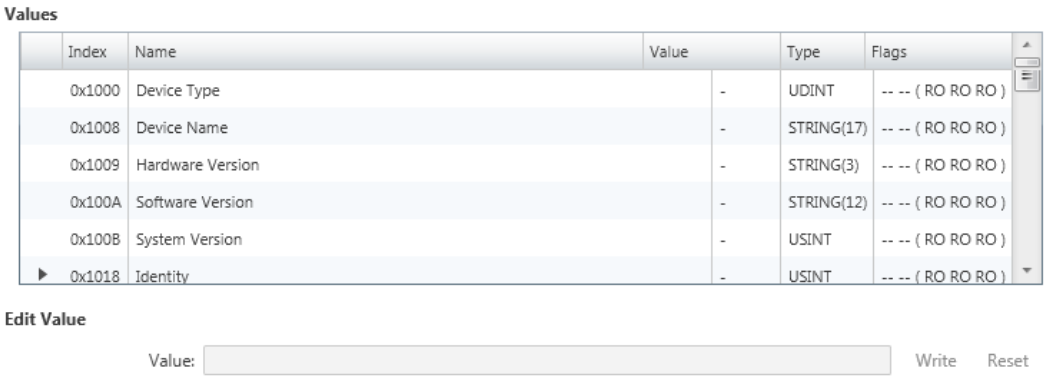


This dialog is only visible in the *'Expert mode'*! With [New] the dialog *'Add CoE Init Command'* opens. This dialog also opens to edit CoE Init Commands, which just exist.

- General
  - Index/Subindex: CoE-Index respectively Subindex of the Init Command
  - Value: Value of the Init Command, which should be written in the chose transition (only available if *'Direction'* is set to *'Download'*). If type of data is unknown, the hex format must be used. (Example: "0011 2233 …").
  - Comment: Here you can comment your Init Command.
- Transition
  - Determines in which transition the Init Command will be executed.
- Further Settings
  - Complete Access: Determines if the complete SDO object should be written/read.
- Direction
  - Download: Writes value to slave station.
  - Upload: Reads value from slave.
- CoE Object Dictionary: Select here the value in the CoE Object Dictionary of the slave station, you want to edit.

7.9.5.8 CoE Object Dictionary (Expert mode)





This dialog is only visible in the *'Expert mode'*! Here you will have read and write access to the CoE Object Dictionary of the slave station. This can be changed if your slave station permits. It is indicated by the *'Flags'* of each object, if write access is permitted. Information about the structure of the Object Dictionary can be found in the manual of your slave station.

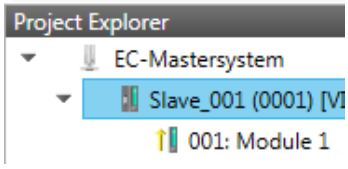
Description of the flags: "AA BB (CC DD EE)"

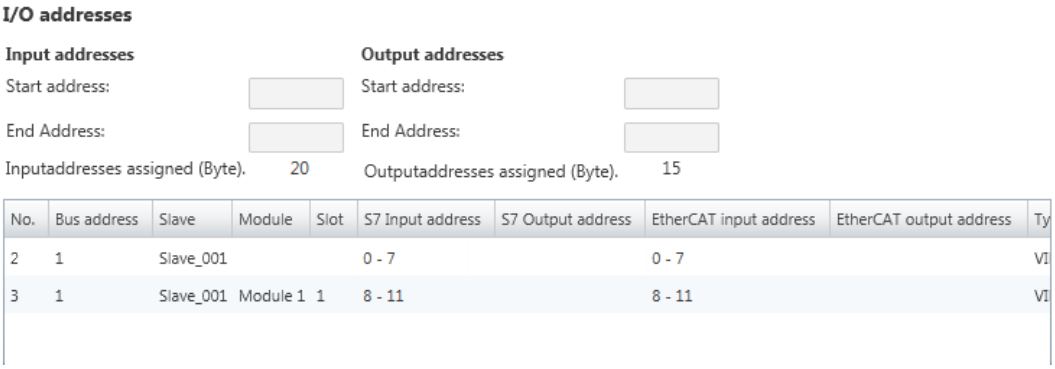
- AA, BB
  - Rx: Mapping as receive PDO
  - Tx: Mapping as send PDO
  - --: Mapping not allowed
- CC:
  - Access rights for state *PreOp* (RO, WO, RW)
- DD:
  - Access rights for state *SafeOp* (RO, WO, RW)
- EE:
  - Access rights for state *Op* (RO, WO, RW)

↳ *Chapter 7.1.2 'EtherCAT State Machine' on page 122*

If a write access to an object in the configuration mode is performed, and the written value does not reflect to the default value of the object, so this command is automatically added to the 'Init Commands'. ↳ *Chapter 7.9.5.7 'Init Commands (Expert mode)' on page 175*

7.9.5.9 Process Image



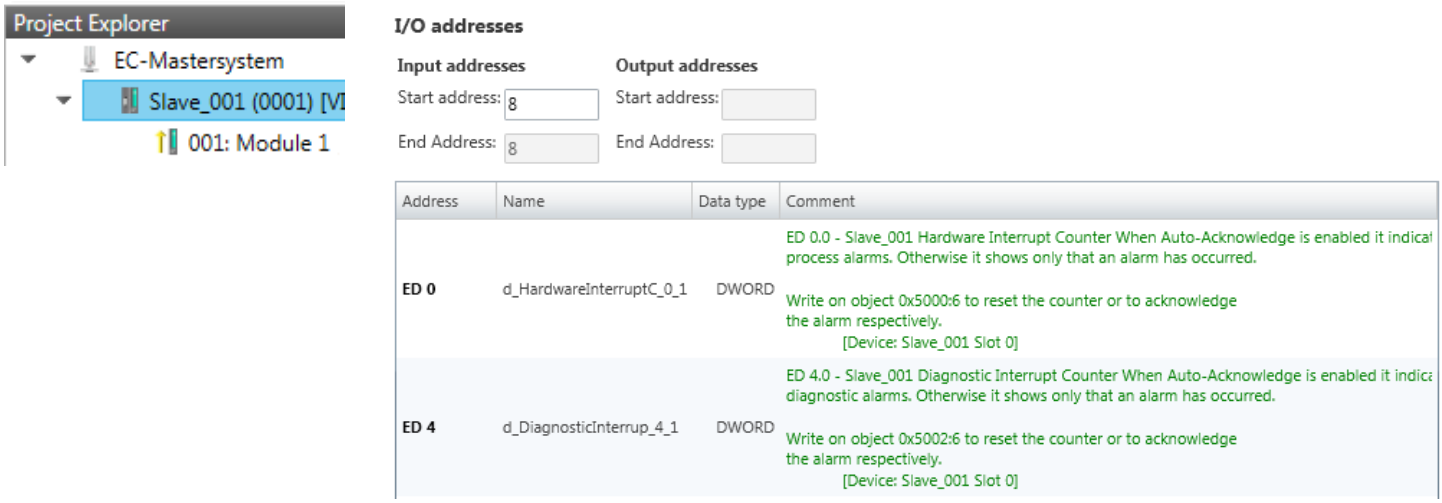


Here you have a list of S7 respectively EtherCAT addresses, which are used by the modules of the slave system. The *'S7 address'* corresponds to the address in the address area of the CPU. By entering a new *'Start address'* you can adjust the S7 addressing of the input and output areas of the modules accordingly.

Information about the assignment of the in/output area can be found in the manual of your module.

The *'I/O addresses EtherCAT'* are only visible in *'Expert mode'*! *'I/O addresses EtherCAT'* are the addresses, which are used within the EtherCAT bus. You cannot change the address. You can use the addresses e.g. for EtherCAT network analysis.

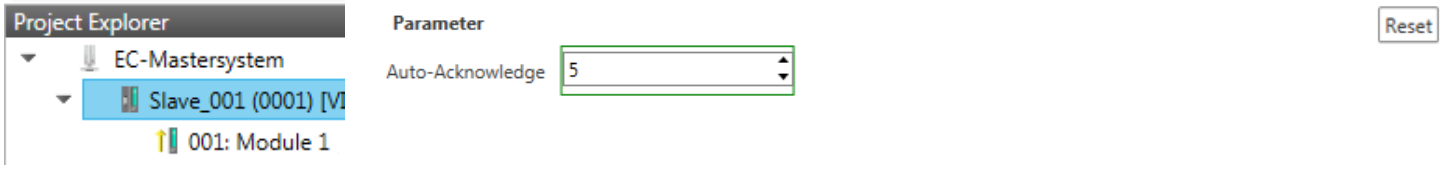
7.9.5.10 I/O Address Overview



Here you have a list of addresses, which are used by the I/O components of the modules of the selected slave system in the address area of the CPU. By entering a new *'Start address'* you can adjust the addressing of the input and output areas accordingly. You can edit *'Name'* and *'Comment'* by clicking at the corresponding entry.

Information about the assignment of the in/output area can be found in the manual of your module.

7.9.5.11 Parameter



If the parameters of the slave station can be determined such as a System SLIO slave station, the slave parameters can be set here. With [Reset], the parameters of the slave station are reset to their default values.

More information about the parameters can be found in the manual of you slave station.

7.9.6 Configuration - modules

With an E-Bus slave the dialogs of the module configuration are hidden!
Chapter 7.9.9 'Grouping logic' on page 189

7.9.6.1 Preparation

Select in the configuration mode in the *'Project Explorer'* the module of the according slave station. The following registers are available now:

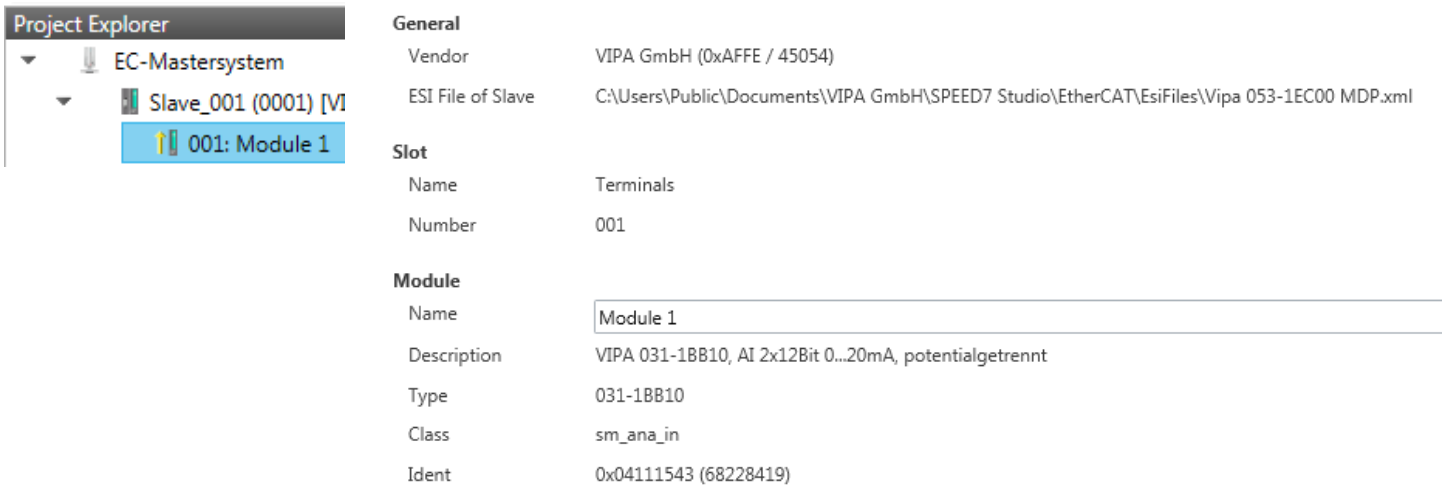
Chapter 7.9.6.2 'MDP Slot Properties' on page 179

Chapter 7.9.6.3 'Process Image' on page 180

Chapter 7.9.6.4 'I/O Address Overview' on page 180

Chapter 7.9.6.5 'Parameter' on page 181

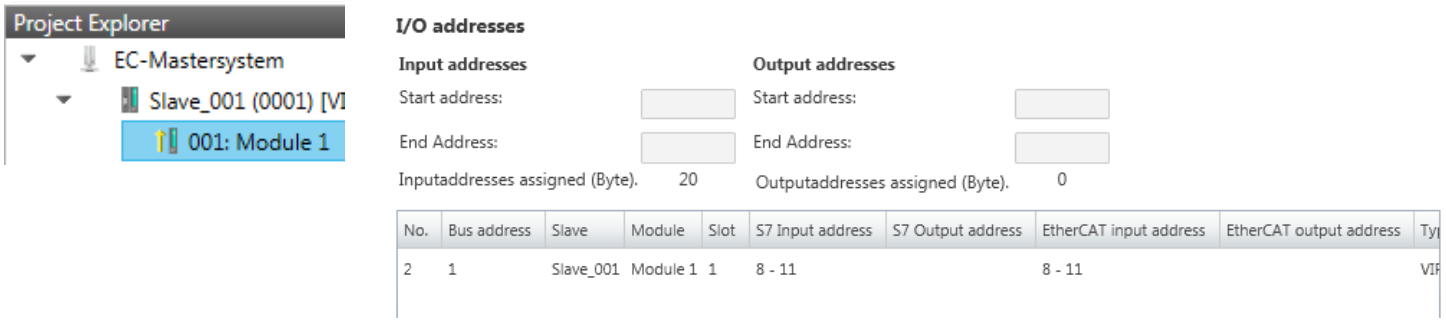
7.9.6.2 MDP Slot Properties



Here you can see the MDP Slot Properties of the corresponding module. This dialog serves for information. You cannot change something.

- General
  - Vendor: Name of the vendor of the module.
  - ESI file: Path and name of the device file, in which the data of the module and the associated slave station is stored.
- Slot
  - Name: Name of the slot
  - Number: Number of the slot
- Module
  - Name: Name of the module
  - Type: Order number of the module
  - Class: Module class
  - Identificator: Identification number of the according module class.

7.9.6.3 Process Image

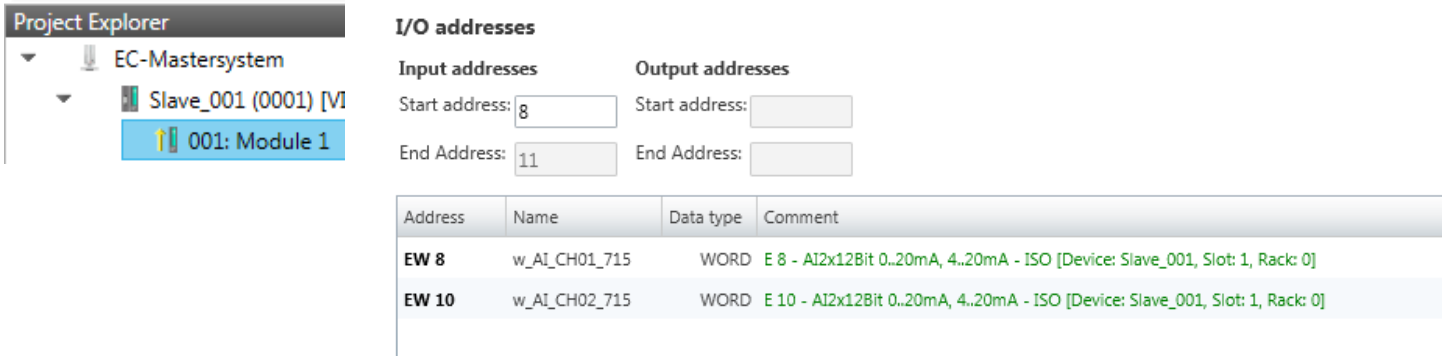


Here you have a list of S7 respectively EtherCAT addresses, which are used by the modules of all the slave stations. The *'S7 address'* corresponds to the address in the address area of the CPU. By entering a new *'Start Address'* you can adjust the S7 addressing of the input and output areas of the modules accordingly.

Information about the assignment of the in/output area can be found in the manual of your module.

The *'I/O Addresses EtherCAT'* are only visible in *'Expert mode'*! *'I/O Addresses EtherCAT'* are the addresses, which are used within the EtherCAT bus. You cannot change the addresses. You can use the addresses e.g. for EtherCAT network analysis.

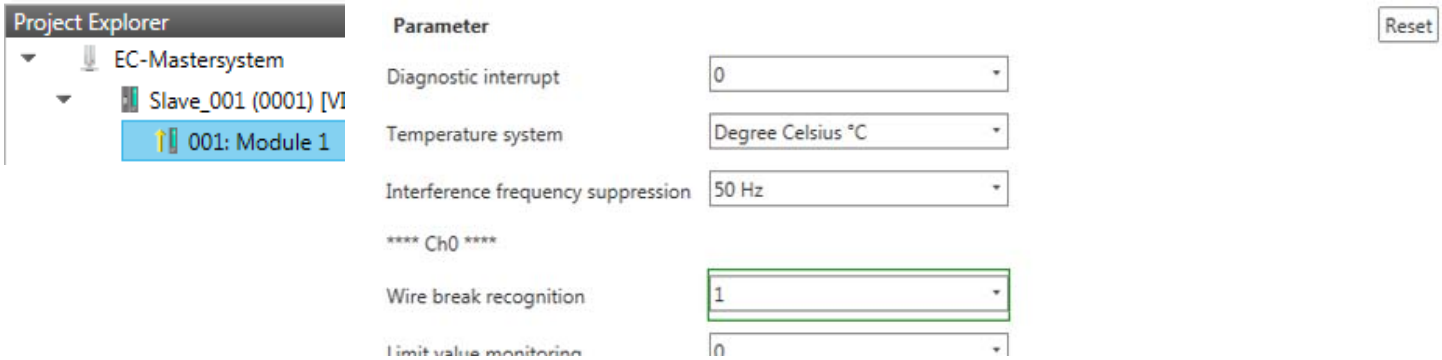
7.9.6.4 I/O Address Overview



Here you have a list of addresses that are used by the module in the address area of the CPU. By entering a new *'Start address'* you can adjust the addressing of the input and output areas accordingly. You can edit *'Name'* and *'Comment'* by clicking at the corresponding entry.

Information about the assignment of the in/output area can be found in the manual of your module.

7.9.6.5 Parameter



If there is a parametrizable module, whose parameters can be determined such as a System SLIO module, the module parameters can be set here. Here also the necessary Init command for the EtherCAT slave station is generated ↳ *Chapter 7.9.5.7 'Init Commands (Expert mode)' on page 175*. With [Reset], the parameters of the module are reset to their default values.

More information about the parameters can be found in the manual of you module.

7.9.7 Diagnostics - EC-Mastersystem

7.9.7.1 Preparation

To use the *'Diagnostics'* functions, you must be connected online with your EtherCAT system.

1. Click in the Toolbar at [Configuration] and select *'EC-Mastersystem'* in the *'Project Explorer'*.
2. Activate in the *'Device editor'* the register *'Master'*.
3. Set depending on the on-line access in the *'Device Editor > Master'* as follows:
   - If you are directly connected to a slave station via EtherCAT by means of a separate network adapter, select your *Network Adapter* and click at [Select].
   - If you are connected to the PG/OP channel of you CPU, please enter *IP Address*, *Port* and *Master Instance* and click at [Select]. With VIPA *Port* 6000 and *Master Instance* 0 is to be set.

   ⇨ The *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.

4. Click in the Toolbar at [Diagnosis Mode].

   ⇨ An online connection to your EtherCAT system is established via the preset communication channel and the current project configuration in the *'Project Explorer'*.

   With an online connection the 2 LEDs flash alternately in the *'Status area'*. In addition *'Modus'* switches to *'Diagnosis'*.

5. Click in the *'Project Explorer'* at Master.

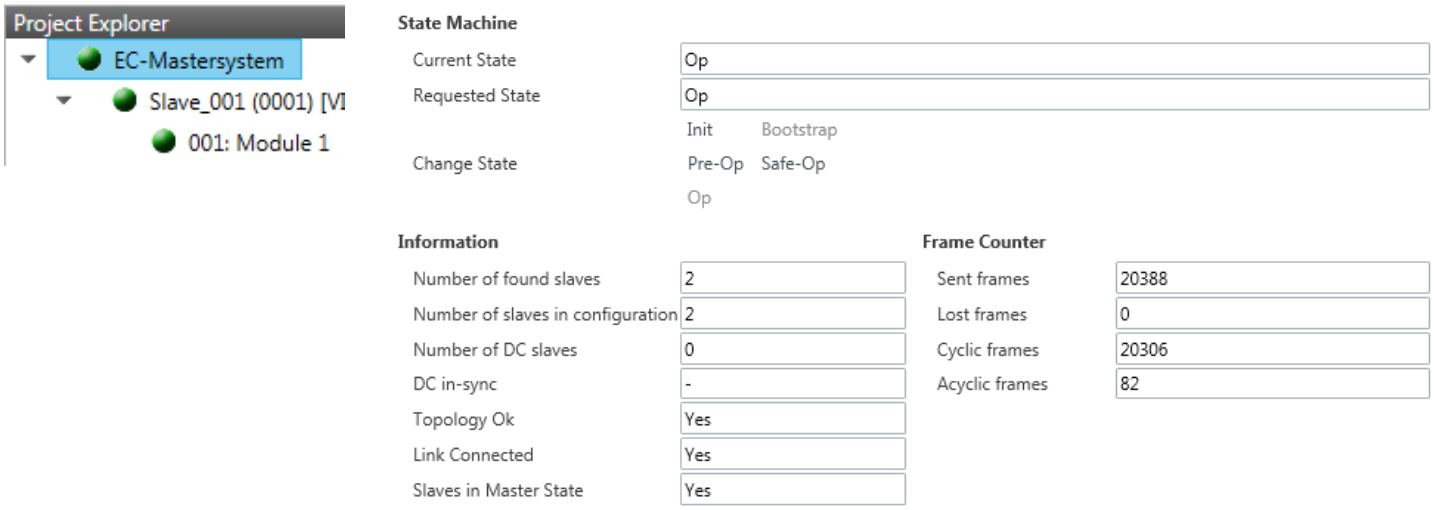
   ⇨ The following registers are available now:

   ↳ *Chapter 7.9.7.2 'General' on page 182*

   ↳ *Chapter 7.9.7.3 'CoE Object Dictionary' on page 183*

   ↳ *Chapter 7.9.7.4 'History (Expert mode)' on page 184*

7.9.7.2 General



Colors and states

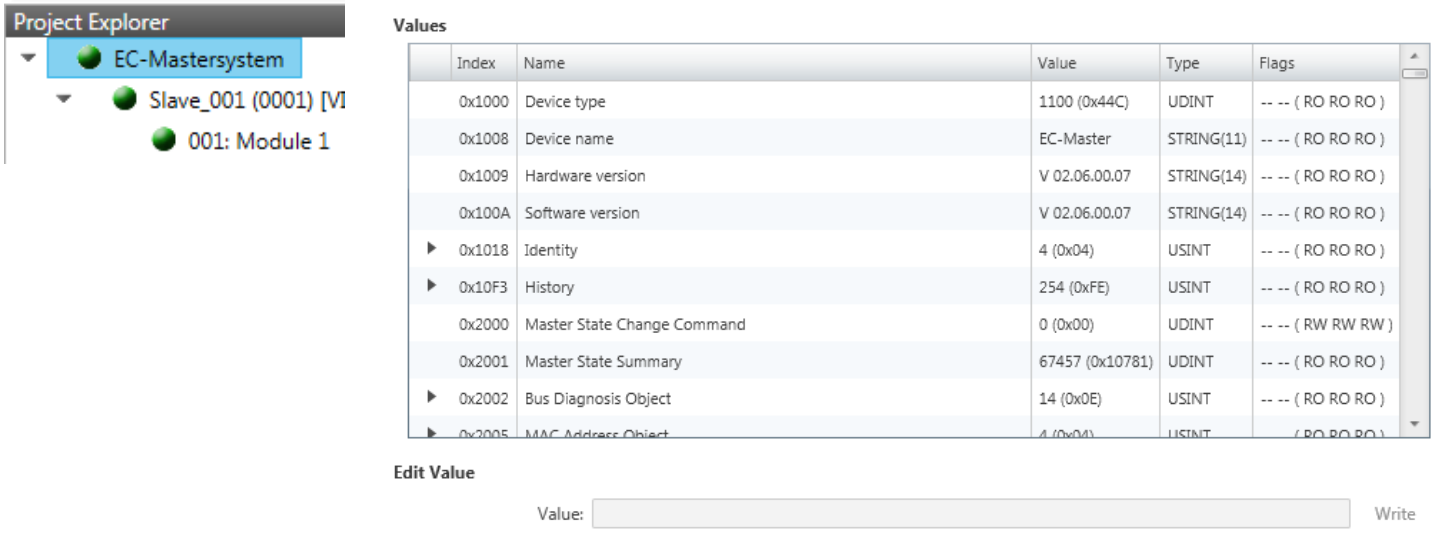
The state of the state machine can be determined via the color according to the following specifications:

Color	State of the state machine
- red	Init / Bootstrap
- blue	Pre-Op
- yellow	Safe-Op
- green	Op

Here you will get master and bus-specific information.

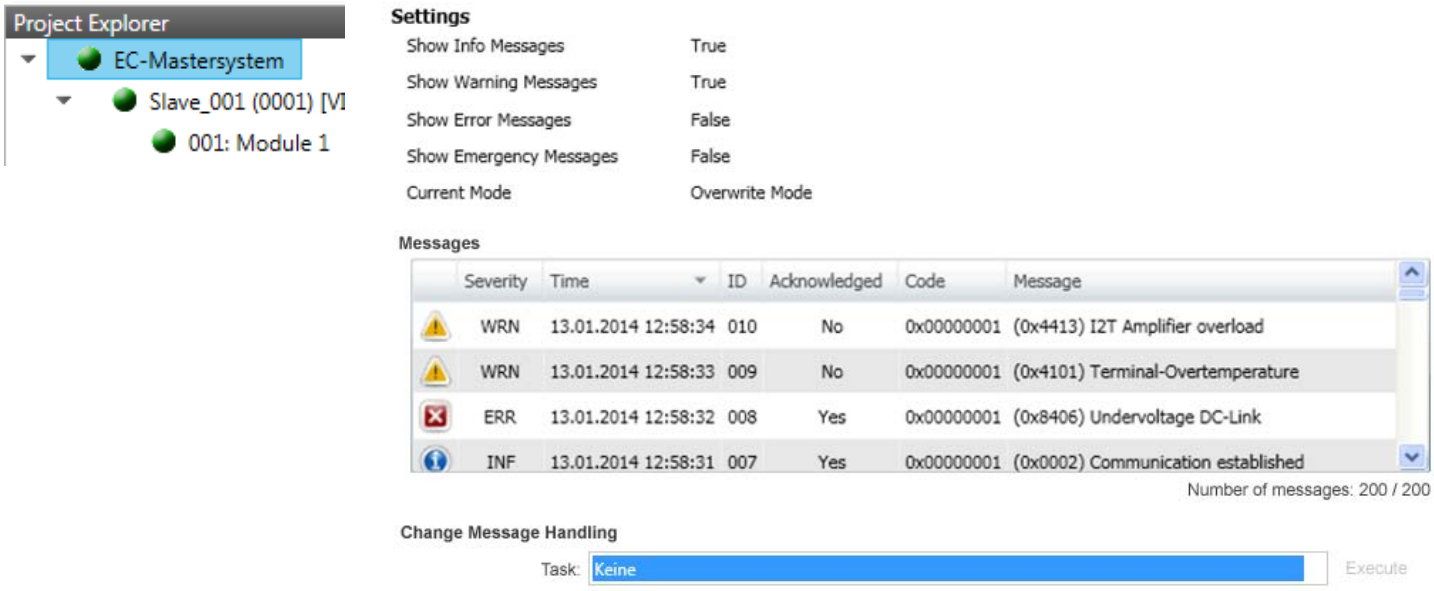
- State Machine
  - Current State: Shows the current state of master. *Chapter 7.1.2 'EtherCAT State Machine' on page 122*
  - Requested State: Shows the currently requested state of the master which was requested by *'Change State '*.
  - Change State: Here you can change the state of the master.
- Information
  - Number of found slaves: Shows number of found slave stations at the bus.
  - Number of slaves in configuration: Shows number of configured slave stations at the bus.
  - Number of DC slaves: Shows the number of slave stations, which support distributed clocks functionality (DC).
  - DC in-sync: If distributed clocks is configured you can find here information about the synchronization status of the system.
  - Topology OK: The *'Topology'* is OK (*'Yes'*), if the number of configured matches the number of found slave stations. Here only the mandatory slaves stations are considered.
  - Link Connected: Here you will find *'Yes'*, if there is a physical connection to the configured slave stations.
  - Slaves in Master State: Here you will find *'Yes'*, if every configured slave station is in master state.
- Frame Counter
  - Sent frames: Number of sent frames since the last power cycle.
  - Lost frames: Number of lost frames since the last power cycle.
  - Cyclic frames: Number of cyclic frames since the last power cycle.
  - Acyclic frames: Number of acyclic frames since the last power cycle.

7.9.7.3 CoE Object Dictionary



Here you will have read and write access to the CoE Object Dictionary of the slave station. This can be changed if your slave station permits. It is indicated by the *'Flags'* of each object, if write access is permitted. Information about the structure of the Object Dictionary can be found in the manual of your slave station.

7.9.7.4 History (Expert mode)



In this dialog box, you can access all the diagnostic messages in the master and edit them if necessary. Via *'Settings'* they may be filtered accordingly.

7.9.8 Diagnostics - slave station

7.9.8.1 Preparation

To use the *'Diagnostics'* functions, you must be connected online with your EtherCAT system.

1. Click in the Toolbar at [Configuration] and select *'EC-Mastersystem'* in the *'Project Explorer'*.
2. Activate in the *'Device editor'* the register *'Master'*.
3. Set depending on the on-line access in the *'Device Editor > Master'* as follows:
   - If you are directly connected to a slave station via EtherCAT by means of a separate network adapter, select your *Network Adapter* and click at [Select].
   - If you are connected to the PG/OP channel of you CPU, please enter *IP Address*, *Port* and *Master Instance* and click at [Select]. With VIPA *Port* 6000 and *Master Instance* 0 is to be set.

   ⇨ The *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.
4. Click in the Toolbar at [Diagnosis Mode].

   ⇨ An online connection to your EtherCAT system is established via the preset communication channel and the current project configuration in the *'Project Explorer'*

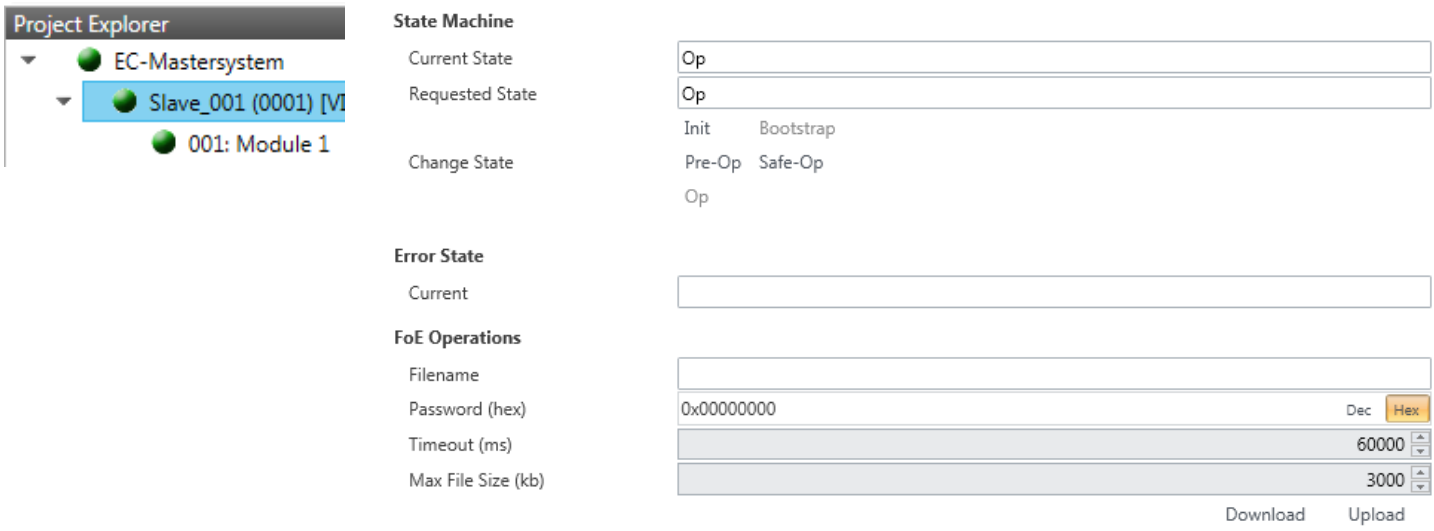
   With an online connection the 2 LEDs flash alternately in the *'Status area'*. In addition *'Modus'* switches to *'Diagnosis'*.

5. Click in the *'Project Explorer'* at the according slave station *'Slave_...'*

The following registers are available now:

↳ *Chapter 7.9.8.2 'General' on page 185*
↳ *Chapter 7.9.8.3 'ESC Register (Expert mode)' on page 186*
↳ *Chapter 7.9.8.4 'EEPROM (Expert mode)' on page 187*
↳ *Chapter 7.9.8.5 'Extended Diagnosis (Expert mode)' on page 187*
↳ *Chapter 7.9.8.6 'DC Diagnosis (Expert mode)' on page 188*
↳ *Chapter 7.9.8.7 'CoE Object Dictionary' on page 188*

7.9.8.2 General



Colors and states

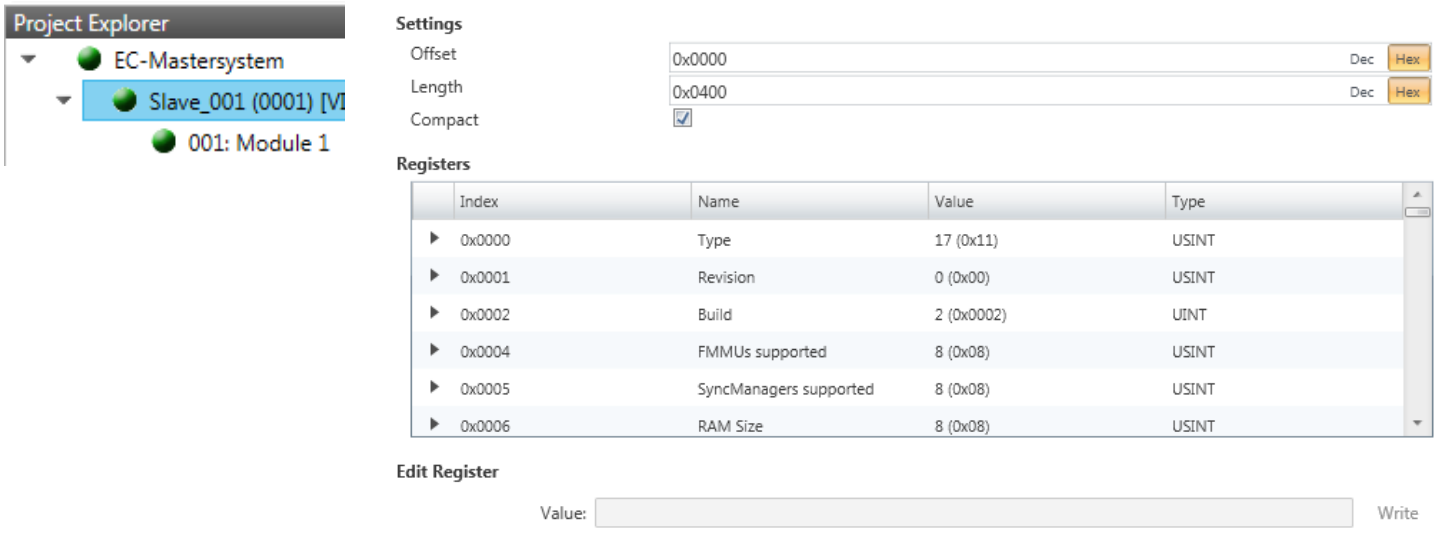
The state of the state machine can be determined via the color according to the following specifications:

Color	State of the state machine
- red	Init / Bootstrap
- blue	Pre-Op
- yellow	Safe-Op
- green	Op

- State Machine
  - Current State: Shows the current state of the state machine of the slave station. *Chapter 7.1.2 'EtherCAT State Machine' on page 122*
  - Requested State: Shows the requested state of the slave station.
  - Change State: Here you can change the state of the state machine of the slave station.
- Error State
  - Current: Slave error which occurred during state transition.
- FoE Operations (file access via EtherCAT)

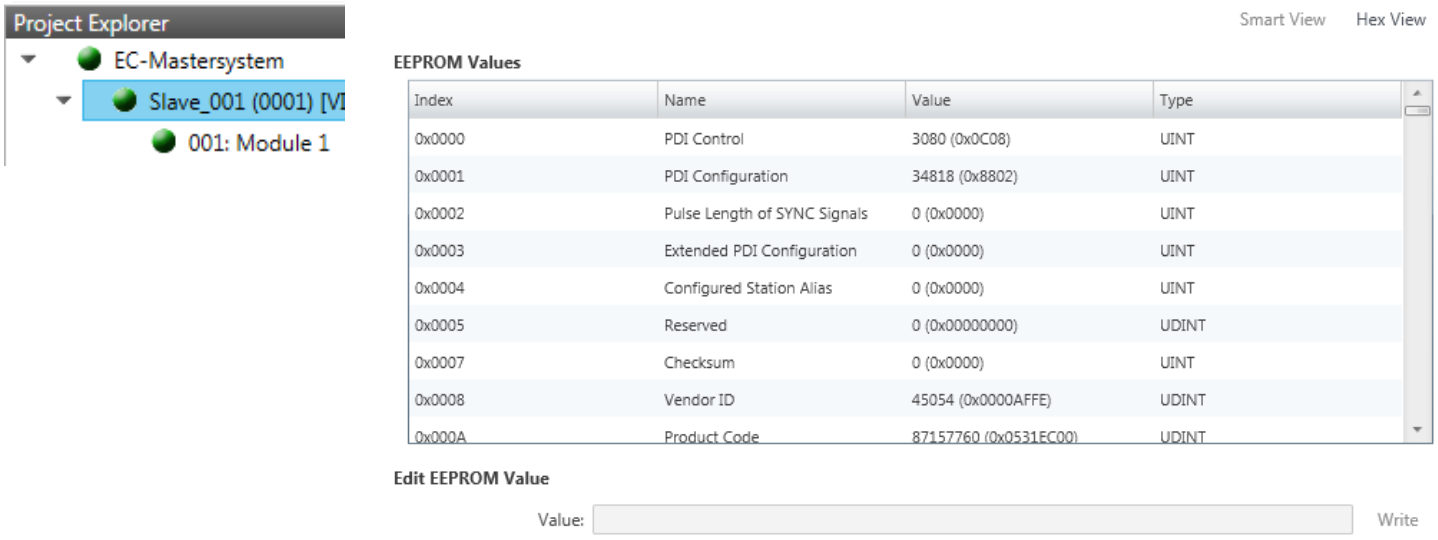
  With this function you have the possibility to transfer files between PC and slave station (if this is supported by the device). If the slave station is in state *Bootstrap*, a firmware update of the slave station can be established via FoE. Here you have to enter the file name without extension. *Chapter 7.9.11 'Firmware update - VIPA System SLIO IM 053-1EC00' on page 195*
  - Filename: Name of the file
  - Password: Password for the access of the slave station
  - Timeout: Maximum time for data transfer
  - Max. File Size: Maximum File Size.

7.9.8.3 ESC Register (Expert mode)



This dialog is only visible in the *'Expert mode'*! Here you can directly access the registers of the EtherCAT ASIC. You should not make any changes here!

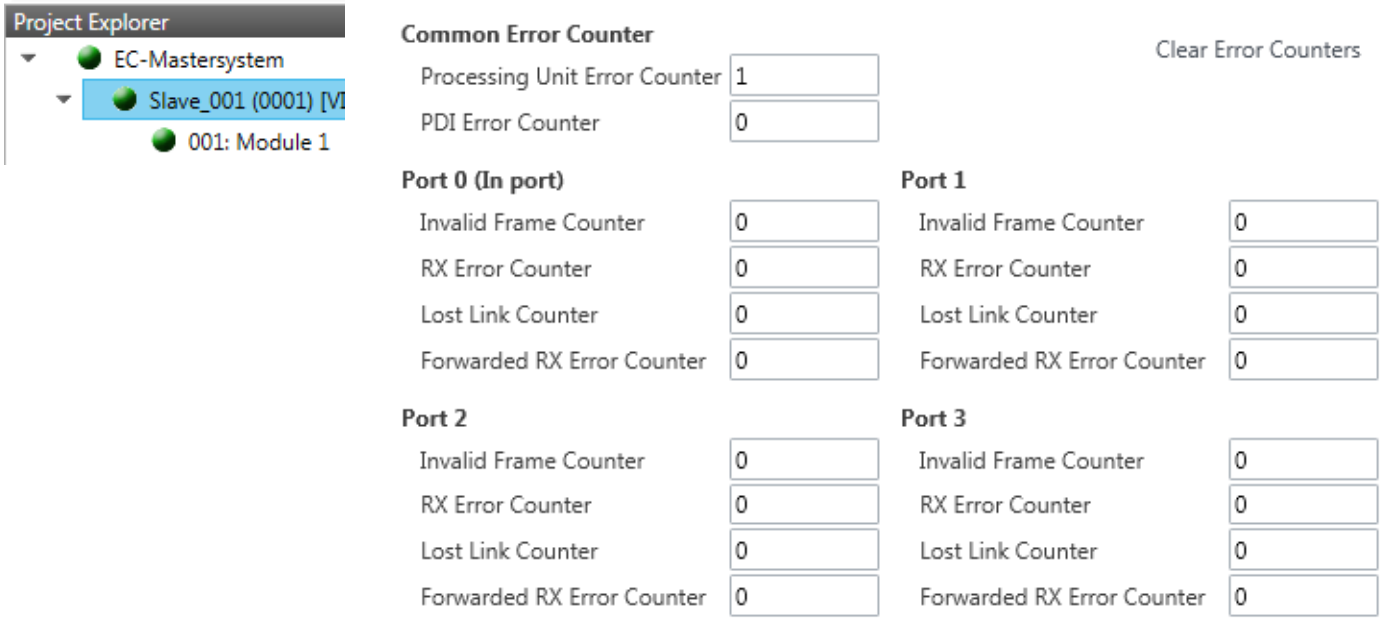
7.9.8.4 EEPROM (Expert mode)



This dialog is only visible in the *'Expert mode'*! Here you can access the contents of the EEPROM of the slave station. Currently you can only change the parameter *'Configured Station Alias'*. This can be used for forming groups. *Chapter 7.9.9 'Grouping logic' on page 189*

CAUTION!
Please regard that your slave station can get unusable by entering incorrect values especially in *'Hex view'*! In this case, any warranty of the vendor is excluded!

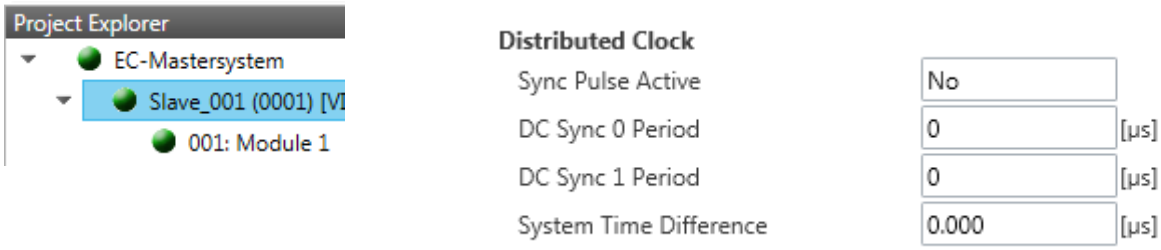
7.9.8.5 Extended Diagnosis (Expert mode)



This dialog is only visible in the *'Expert mode'*!

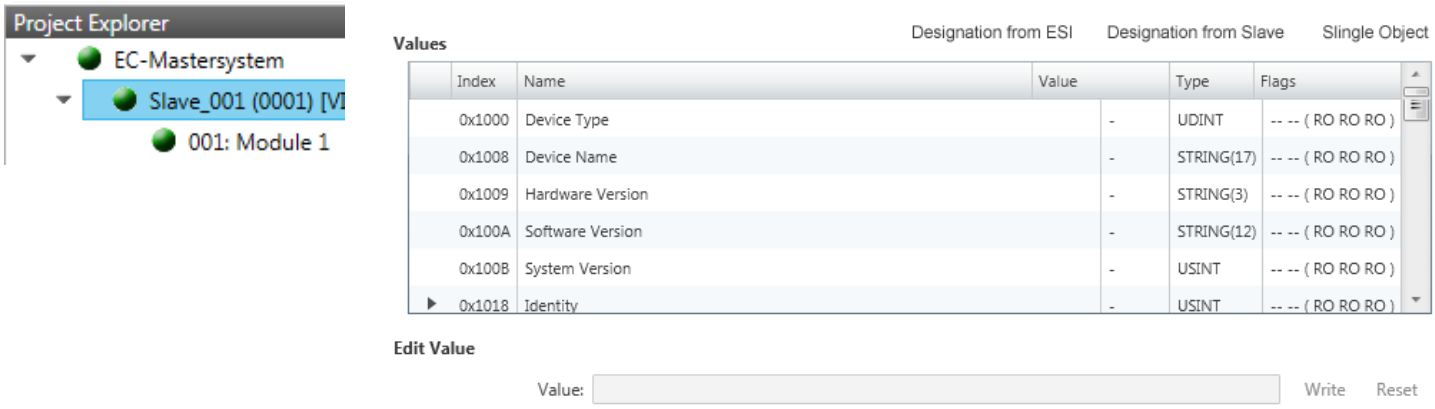
- Common Error Counter
  - Processing Unit Error Counter: Number of received frames by the slave station, which are no EtherCAT frames.
  - PDI Error Counter: Number of PDI access errors (**P**rocess **D**ata **I**nterface). These are physical errors, which were detected by the PDI at the EtherCAT bus.
  - With [Clear Error Counters] the error counters can be reset.
- Port 0...3
  - Invalid Frame Counter: Number of invalid frames from *Port* y (access at register 0x300+y*2)
  - RX Error Counter: Number of RX errors from *Port* y (access at register 0x300+y*2+8bit)
  - Lost Link Counter: Number of lost connections from *Port* y (access at register 0x310+y)
  - Forwarded RX Error Counter: Number of forwarded RX errors from *Port* y (access at register 0x380+y)

7.9.8.6 DC Diagnosis (Expert mode)



This dialog is only visible in the *'Expert mode'*! Here status information for the distributed clock of your slave station is shown. More may be found in the manual of the slave station.

7.9.8.7 CoE Object Dictionary



Here you will have read and write access to the CoE Object Dictionary of the slave station. This can be changed if your slave station permits. It is indicated by the *'Flags'* of each object, if write access is permitted. Information about the structure of the Object Dictionary can be found in the manual of your slave station.

If a write access to an object in the diagnosis mode is performed, and the written value does not reflect to the default value of the object, so this command is automatically added to the 'Init commands'. *Chapter 7.9.5.7 'Init Commands (Expert mode)' on page 175*

These buttons areonly visible in the *'Expert mode'*:

- Designation from ESI
  - By selecting this button the designations are loaded from the ESI file.
- Designation from slave
  - By selecting this button the designations are loaded from the slave station.
- Single object
  - With this button you have read and write access to a single object in the object dictionary by specifying index and subindex.

7.9.9 Grouping logic

7.9.9.1 Overview

Slave types

With EtherCAT, the following slave types are distinguished:

MII slave - MII corresponds to **M**edia **I**ndependant **I**nterface. An MII slave has an EtherCAT interface to connect to EtherCAT for integration into a system bus (backplane bus) for connecting peripheral modules. The MII slave receives data via EtherCAT and passes them through its backplane to the according peripheral module. Conversely, it reads the input data and passes it via EtherCAT. The System SLIO 053-1EC00 e.g. is a MII-Slave.

E-Bus slave - In an E-Bus slave the EtherCAT protocol is used for communication on the backplane bus For this reason, the attached peripheral modules are also shown as a slave station in the *SPEED7 EtherCAT Manager*

Possibilities

The EtherCAT Manager supports the following ways to group the individual slave stations. Each group may consist of 1 .. n slave stations. Group nesting is not supported:

Chapter 7.9.9.2 'Create group with pinned process data offset' on page 190

'Hot Connect group with Dynamic Position in Topology' on page 193

'Hot Connect group with Fixed Position in Topology' on page 193

'Hot Connect group with Pinned or Dynamic Process Data Offset' on page 193

Please consider that Hot Connect groups are not possible with E-Bus slaves!

Create Group

1. Click in the Toolbar of the *SPEED7 EtherCAT Manager* at [Configuration].
2. Click in the *Project Explorer* at the slave station and select *'Context menu ➔ Create Group'*.

   ⇨ The dialog *'Create Group'* opens. Here always the 1. slave station is selected. You can either select more slave stations or depending on the group type selection, the necessary save stations are automatically selected.

With the *'Create Group'* functionality you have two different functions:

- You can create a new group if the selected slave station is not yet part of a group.
- If the selected slave station is already part of a group, the current group is divided into two sub-groups from the selected slave station.

Remove Group

To remove a group click in the *SPEED7 EtherCAT Manager* at a slave station and select *'Context menu ➔ Remove Group'*.

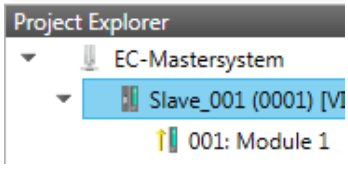
⇨ The group is removed. Depending on the group, the previously grouped slave stations are reintegrated into the topology or remain at the current position.

Edit Group

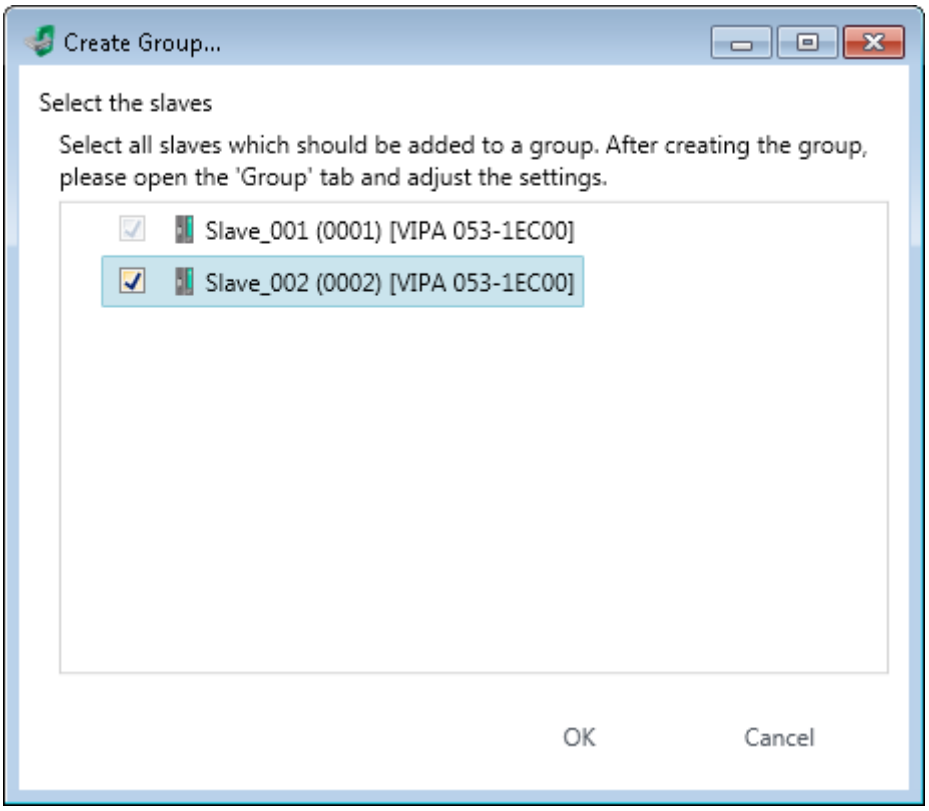
After creating a group, the *'Device Editor'* of the slave station is extended with the register *'Group'*. Here you can adjust the group properties accordingly.

7.9.9.2 Create group with pinned process data offset

Proceeding



This group may start at any slave station and either end at himself, at a following slave station, at a following group or at the last slave station. The group functionality is possible with each slave type. The slave stations of this group are pinned at a fix position in the topology.

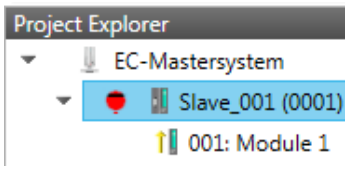


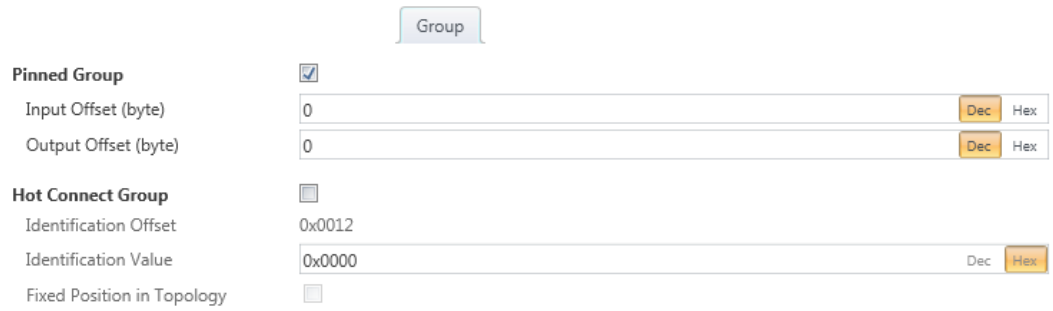
1. Click in the *'Project Explorer'* at the according slave station and select *'Context menu ➔ Create group'*.

   ⇨ A dialog opens to create a group.

2. Choose from *'Select the slaves'* the slave stations, which you want to include in the *'Pinned group'*.

   ⇨ The dialog is closed, the slave station is marked as group in the *'Project Explorer'* and a tab "Group" is created in the *'Device Editor'*.



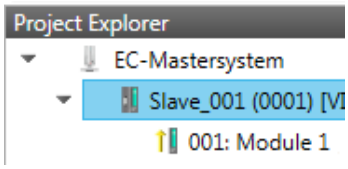


3. Enable the option *'Pinned Group'*.
4. Enable the option *'Input Offset = Output Offset'* if the input and output addresses are identical.

   ⇨ The group is now defined as *Pinned Group*.

7.9.9.3 Create Hot Connect group

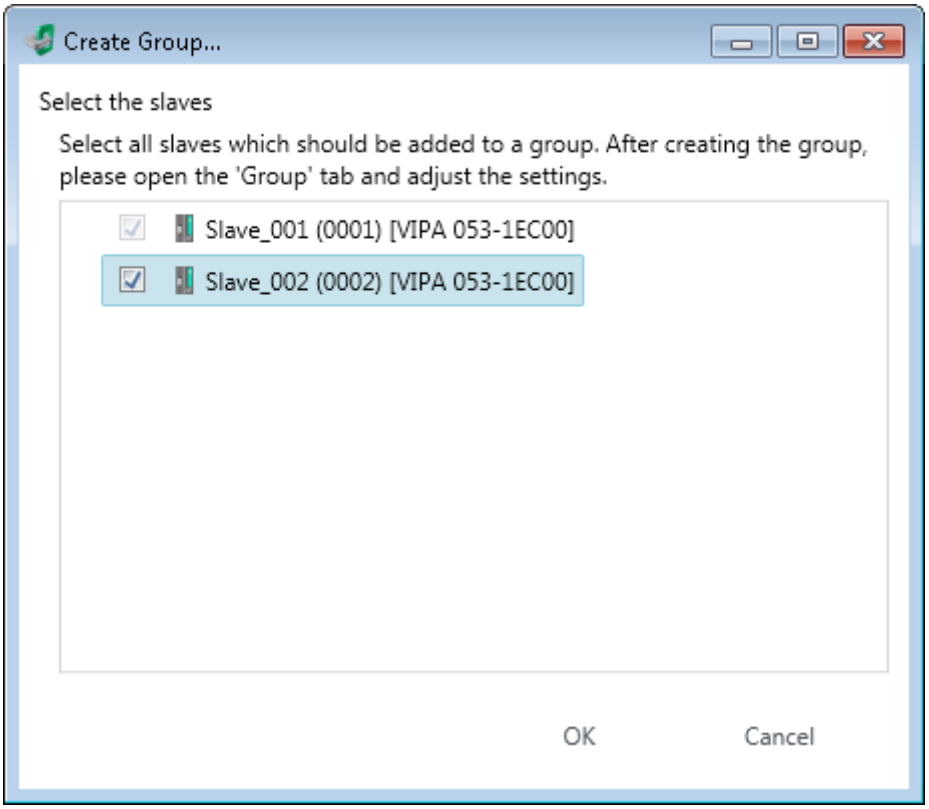
Proceeding



In a *Hot Connect group* several slave stations can be located, which must only optional be available at the EtherCAT bus. So you have the possibility to take or add pre-configured sections from the traffic before starting the system or during the operation. This can be done by disconnecting/connecting the communication path or enabling/disabling the participant.

Please consider that the first slave station after the EtherCAT master must not be optional!

To use the hot connect function with E-Bus slave stations, the E-Bus head station and the connected slave stations must be in the same group! ↳ *Chapter 7.9.9 'Grouping logic' on page 189*

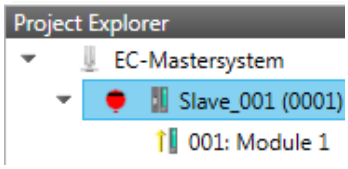


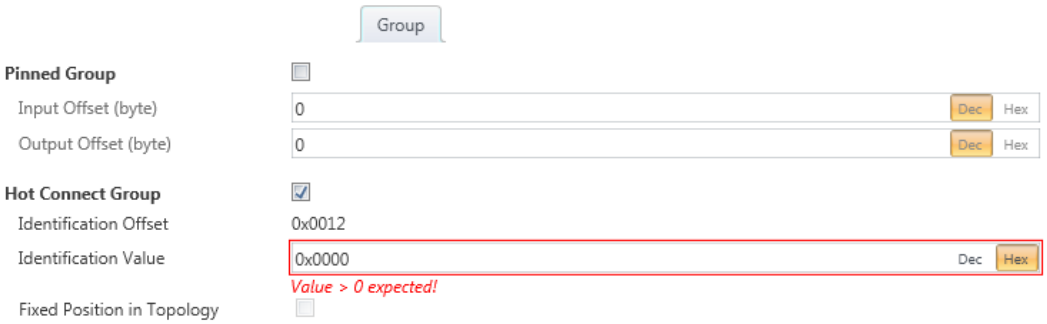
1. Click in the *'Project Explorer'* at the according slave station and select *'Context menu ➔ Create group'*.

   ⇨ A dialog opens to create a group.

2. Choose from *'Select the slaves'* the slave stations, which you want to include in the *'Hot connect group'*.

   ⇨ The dialog is closed, the slave station is marked as group in the *'Project Explorer'* and a tab "Group" is created in the *'Device Editor'*.





3. Enable the option *'Hot connect group'*.
4. Enter an *'Identification value'*: This is the *Station-Alias-Address*, which you have to assign before to the slave station in the *'Diagnosis'* Mode. *Chapter 7.9.8.4 'EEPROM (Expert mode)' on page 187*

   Please regard that the slave station takes the new address after a power-cycle.
5. For a fix position of the group in the topology the option *'Pinned group'* can be enabled.

7.9.9.3.1 Combination possibilities

Hot Connect group with Dynamic Position in Topology

The group must start with a MII slave. Here, all slave stations below the selected are automatically added to the group. This group ends at himself, at a following slave station, at a following group or at the last slave station.

Hot Connect group with Fixed Position in Topology

The group is fix coupled to a predecessor slave station and its port. You always have the possibility to change the link to the previous slave station via the dialog box. If the group is removed, the slave stations remain in place.

A Hot Connect group with Fixed Position in Topology cannot be removed, if the slave stations before are a part of another Hot Connect group with Fixed Position in Topology!

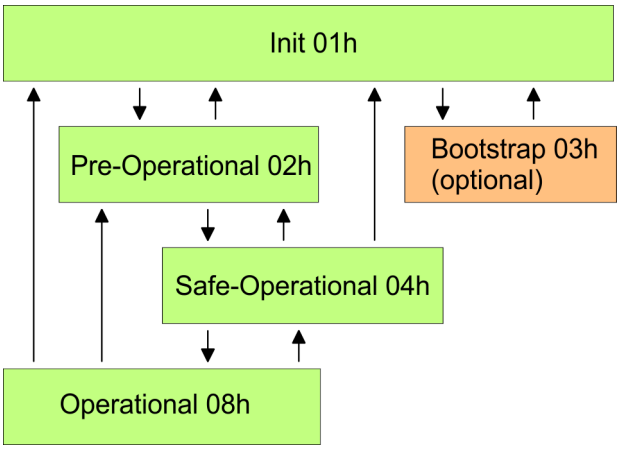
Hot Connect group with Pinned or Dynamic Process Data Offset

This group does not depend on slave station or port. The group has no predecessor slave station and is moved to the end of the tree when created. When the group is removed it is searched for a suited free port starting from the end of the main tree. If there is no suited slave station available, the group will be rejected! Due to the system the group has no predecessor slave station, the connection cannot be changed via the dialog box.

7.9.10 EtherCAT State Machine

States

In each EtherCAT communication device a *state machine* is implemented. For each state there is defined which communication service is active via EtherCAT. The state machine of the slave station is controlled by the state machine of the EtherCAT master.



Init - 01h

After power-on the EtherCAT members are in state *Init*. There is neither mailbox nor process data communication possible. The EtherCAT master initializes the SyncManager channels 0 and 1 for the mailbox communication.

Pre-Operational (Pre-Op) - 02h

The EtherCAT master initializes the SyncManager channels for process data (starting with SyncManager channel 2), the FMMU channels and the PDO mapping respectively the SyncManager PDO assignment. Further in this state the settings for process data transfer and the module-specific parameters, which deviate from the default values are transferred. During the transition from *Init* to *Pre-Op* the EtherCAT slave checks whether the mailbox was correctly initialized. In the state *Pre-Op* mailbox communication and Ethernet over EtherCAT (EoE) are possible but the process data communication is blocked.

Safe-Operational (Safe-Op) - 04h

In *Safe-Op* the input data are cyclically updated but the outputs are de-activated. With the transition from *Pre-Op* to *Safe-Op* the EtherCAT slave checks if the SyncManager channels for process data communication are correct. Before it acknowledges the state change, the EtherCAT slave copies current input data to the corresponding DP RAM areas of the EtherCAT slave controller. In the state *Safe-Op* mailbox and process data communication is possible.

Operational (Op) - 08h

In the state *Op* the input data are cyclically updated and the EtherCAT master sends output data to the EtherCAT slave. The EtherCAT slave copies the output data of the master to its outputs and return input data to the EtherCAT master. In this state process data and mailbox communication is possible.

Bootstrap - option (Boot) - 03h

In state *Boot* the firmware of an EtherCAT slave may be updated via the EtherCAT master. This state may only be reached via Init. In the state *Boot* is mailbox communication via the protocol File-Access over EtherCAT (FoE) possible. Other mailbox and process data communications are de-activated.

7.9.11 Firmware update - VIPA System SLIO IM 053-1EC00

Current firmware at www.vipa.com

The latest firmware versions are to be found in the service area at www.vipa.com.

For example the following files are necessary for the firmware update of the System SLIO IM 053-1EC00 with hardware release 1: Px000106.pkg. Load this file from the VIPA service area.

CAUTION!
When installing a new firmware you have to be extremely careful. Under certain circumstances you may destroy the slave station, for example if the voltage supply is interrupted during transfer or if the firmware file is defective. In this case, please call the VIPA Hotline!

Please regard that the version of the update firmware has to be different from the existing firmware otherwise no update is executed.

Precondition

- There is an Ethernet respectively remote connection between the PC and the VIPA EtherCAT slave station, where a firmware update is to be established.

Proceeding

Below the proceeding is shown by the example of the VIPA System SLIO slave station. For other devices, please follow the procedures described in the according manual.

1. Open if not already done the *SPEED7 EtherCAT Manager*.
2. Click in the *'Project Explorer'* at *'EC-Mastersystem'*.
3. Select in *'Device Editor > Master'* at *'Network Adapter'* your network card and enter at *'IP Address'* the IP address of the PG/OP channel of the CPU and click at [Select].
4. Click in the Toolbar at [Diagnosis Mode].
   ⇨ An online connection to your EtherCAT system is established via the preset communication channel and the current project configuration in the *'Project explorer'*.
5. Click in the *'Project explorer'* at the master.
6. Select in the register *'General'* at *'State Machine'* the state *'Init'*. Wait, until all slave station response the state *'Init'*.
7. Click in the *'Project explorer'* at the slave, where the firmware update is to be established.
8. Select in the register *'General'* at *'State Machine'* the state *'Bootstrap'*.
9. Enter in the register *'General'* at *'FoE Operations'* as follows:
   - Filename: Px000106
   - Password (hex): 0x0000000
   - Timeout (ms): 60000
   - Max File Size (kb): 3000
10. Click at [Download].
    ⇨ A dialog for file selection opens.
11. Select the file. The transfer starts with [OK].
    ⇨ There will be a progress bar displayed, which informs you about the transfer state.

12. After successful download bring your slave in the *'Init'* state.
    ⇨ With this operation the firmware file is taken.

8 *Option:* Deployment PROFIBUS communication

8.1 Overview

Enable bus functionality via VSC

To switch the MPI(PB) interface X3 to PROFIBUS functionality, you have to enable the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is enabled.

'Overview' on page 85

PROFIBUS DP

- PROFIBUS is an international standard applicable to an open and serial field bus for building, manufacturing and process automation that can be used to create a low (sensor-/actuator level) or medium (process level) performance network of programmable logic controllers.
- PROFIBUS comprises an assortment of compatible versions. The following details refer to PROFIBUS DP.
- PROFIBUS DP is a special protocol intended mainly for automation tasks in a manufacturing environment. DP is very fast, offers Plug'n'Play facilities and provides a cost-effective alternative to parallel cabling between PLC and remote I/O. PROFIBUS DP was designed for high-speed data communication on the sensor-actuator level.
- The data transfer referred to as "Data Exchange" is cyclical. During one bus cycle, the master reads input values from the slaves and writes output information to the slaves.

CPU with DP master

The PROFIBUS DP master is to be configured in the hardware configurator from Siemens. Here the configuration happens by the sub module X1 (MPI/DP) of the Siemens CPU. After the transmission of the data to the CPU, the configuration data are internally passed on to the PROFIBUS master part. During the start-up the DP master automatically includes his data areas into the address range of the CPU. Project engineering in the CPU is not required.

Deployment of the DP master with CPU

Via the PROFIBUS DP master PROFIBUS DP slaves may be coupled to the CPU. The DP master communicates with the DP slaves and links up its data areas with the address area of the CPU. At every POWER ON respectively overall reset the CPU fetches the I/O mapping data from the master. At DP slave failure, the OB 86 is requested. If this is not available, the CPU switches to STOP and BASP is set. As soon as the BASP signal comes from the CPU, the DP master is setting the outputs of the connected periphery to zero. The DP master remains in the operating mode RUN independent from the CPU.

DP slave operation

For the deployment in a super-ordinated master system you first have to project your slave system as Siemens CPU in slave operation mode with configured in-/output areas. Afterwards you configure your master system. Couple your slave system to your master system by dragging the CPU 31x from the hardware catalog at *Configured stations* onto the master system, choose your slave system and connect it.

Operating mode DP slave: Test, commissioning, routing (active/passive)

There is the possibility to enable the option *'Test, commissioning, routing'* in the hardware configuration by means of the properties dialog of the PROFIBUS via the register *'Operating mode'* at *'DP slave'*. The activation affects as follows:

- The PROFIBUS interface gets an "active" PROFIBUS node, this means it is involved in the token rotation.
- Via this interface you have PG/OP functions (programming, status request, control, test).

- The PROFIBUS interface serves as a gateway (S7 routing).
- The bus rotation time can exceed.

When disabled, the PROFIBUS interface operates as passive DP slave with the following characteristics:

- The PROFIBUS interface gets an "passive" PROFIBUS node, this means it is not involved in the token rotation.
- Bus rotation time is not influenced.
- S7 routing is not possible.

8.2 Fast introduction

Overview

The PROFIBUS DP master is to be configured in the hardware configurator. Here the configuration happens by means of the sub module X1 (MPI/DP) of the Siemens CPU.

Enable bus functionality via VSC

To switch the MPI(PB) interface X3 to PROFIBUS functionality, you have to enable the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is enabled.

'Overview' on page 85

Steps of configuration

For the configuration of the PROFIBUS DP master please follow the following approach:

- **Enable bus functionality via VSC**
- **Hardware configuration - CPU**
- **Deployment as DP master or DP slave**
  - With activating the bus function *'PROFIBUS DP master'* by means of the VSC, the bus function *'PROFIBUS DP slave'* is also unlocked.
- **Transfer of the complete project to CPU**

With the Siemens SIMATIC Manager, the CPU 015-CEFNR00 from VIPA is to be configured as

CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2)

The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X1 (MPI/DP).

8.3 Enable bus functionality via VSC

Enabling

'Overview' on page 85

8.4 Hardware configuration - CPU

Precondition

The configuration of the CPU takes place at the Siemens *'hardware configurator'*. The hardware configurator is part of the Siemens SIMATIC Manager. It serves for project engineering. Please use for configuration the Siemens SIMATIC Manager V 5.5 SP2 and up. The modules, which may be configured here are listed in the hardware catalog. If necessary you have to update the hardware catalog with *'Options ➔ Update Catalog'*.

For project engineering a thorough knowledge of the Siemens SIMATIC Manager and the Siemens hardware configurator is required!

Proceeding

With the Siemens SIMATIC Manager the following steps should be executed:

1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at *'Slot'*-Number 2 the CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).

Slot	Module
1	
2	**CPU 31...**
X1	*MPI/DP*
X2	*PN-IO*
X2...	*Port 1*
X2...	*Port 2*
3	

The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X1 (MPI/DP).

8.5 Deployment as PROFIBUS DP master

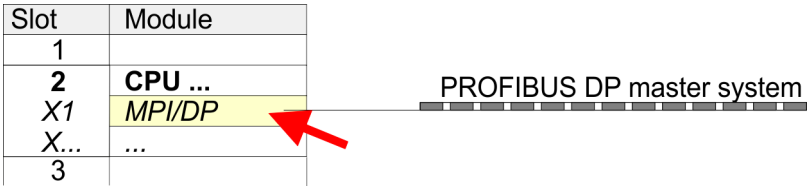
Precondition

The hardware configuration described before was established.

Proceeding

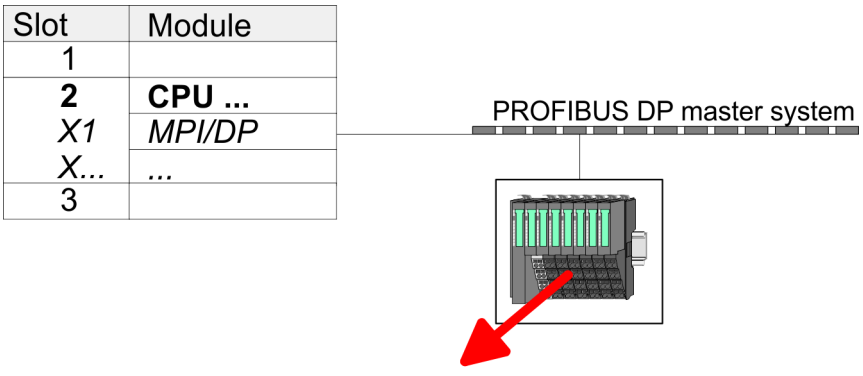
1. Open the properties dialog of the DP interface of the CPU by means of a double-click at *'MPI/DP'*.
2. Set at Interface: Type "PROFIBUS".
3. Connect to PROFIBUS and preset an address (preferably 2). Confirm your input with [OK].
4. Switch at Operating mode to "DP master" and confirm the dialog with [OK].

   ⇨ A PROFIBUS DP master system is inserted:



Now the project engineering of your PROFIBUS DP master is finished. Please link up now your DP slaves with periphery to your DP master.

1. For the project engineering of PROFIBUS DP slaves you search the concerning PROFIBUS DP slave in the hardware catalog and drag&drop it in the subnet of your master.
2. Assign a valid PROFIBUS address to the DP slave.
3. Link up the modules of your DP slave system in the plugged sequence and add the addresses that should be used by the modules.
4. If needed, parametrize the modules.
5. Save, compile and transfer your project.



Slot	Module	Order number	
1	...		
2	Modules		
3	...		
4			
5			
...			

8.6 Deployment as PROFIBUS DP slave

Fast introduction

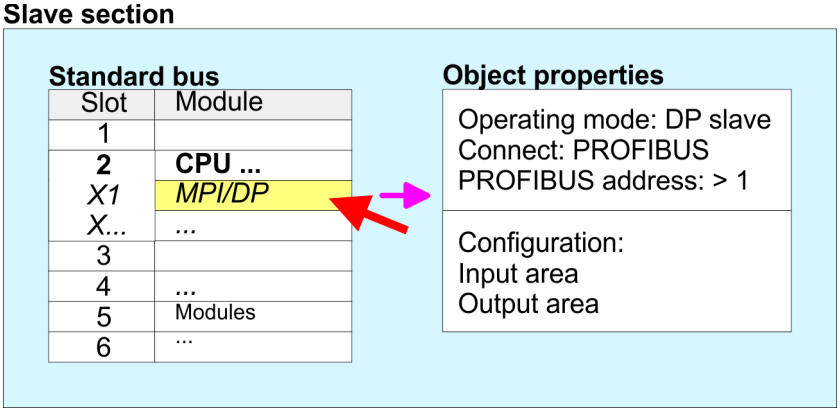
In the following the deployment of the PROFIBUS section as "intelligent" DP slave on master system is described, which exclusively may be configured in the Siemens SIMATIC Manager. The following steps are required:

1. Configure a station with a CPU with operating mode DP slave.
2. Connect to PROFIBUS and configure the in-/output area for the slave section.
3. Save and compile your project.
4. Configure another station with another CPU with operating mode DP master.
5. Connect to PROFIBUS and configure the in-/output ranges for the master section.
6. Save, compile and transfer your project to your CPU.

Project engineering of the slave section

1. Start the Siemens SIMATIC Manager and configure a CPU as described at "Hardware configuration - CPU".
2. Designate the station as "...DP slave".
3. Add your modules according to the real hardware assembly.
4. Open the properties dialog of the DP interface of the CPU by means of a double-click at *'MPI/DP'*.
5. Set Interface type to "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 3) and confirm with [OK].
7. Switch at Operating mode to "DP slave" .
8. Via Configuration you define the in-/output address area of the slave CPU, which are to be assigned to the DP slave.
9. Save, compile and transfer your project to your CPU.

Slave section

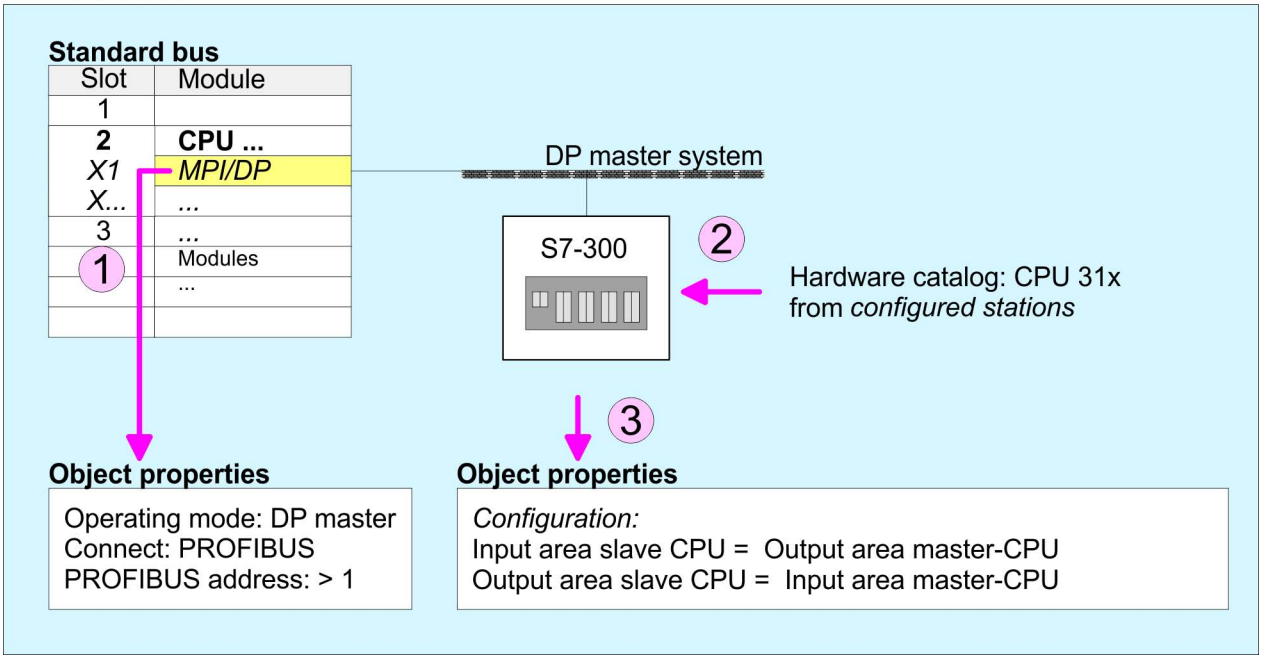


Project engineering of the master section

1. Insert another station and configure a CPU.
2. Designate the station as "...DP master".
3. Add your modules according to the real hardware assembly.
4. Open the properties dialog of the DP interface of the CPU by means of a double-click at *'MPI/DP'*.
5. Set Interface: type to "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 2) and confirm with [OK].
7. Switch at Operating mode to "DP master" and confirm the dialog with [OK].

8. Connect your slave system to this master system by dragging the "CPU 31x" from the hardware catalog at *Configured stations* onto the master system and select your slave system to be coupled.
9. Open the *Configuration* at *Object properties* of your slave system.
10. Via double click to the according configuration line you assign the according input address area on the master CPU to the slave output data and the output address area to the slave input data.
11. Save, compile and transfer your project to your CPU.

Master section



8.7 PROFIBUS installation guidelines

PROFIBUS in general

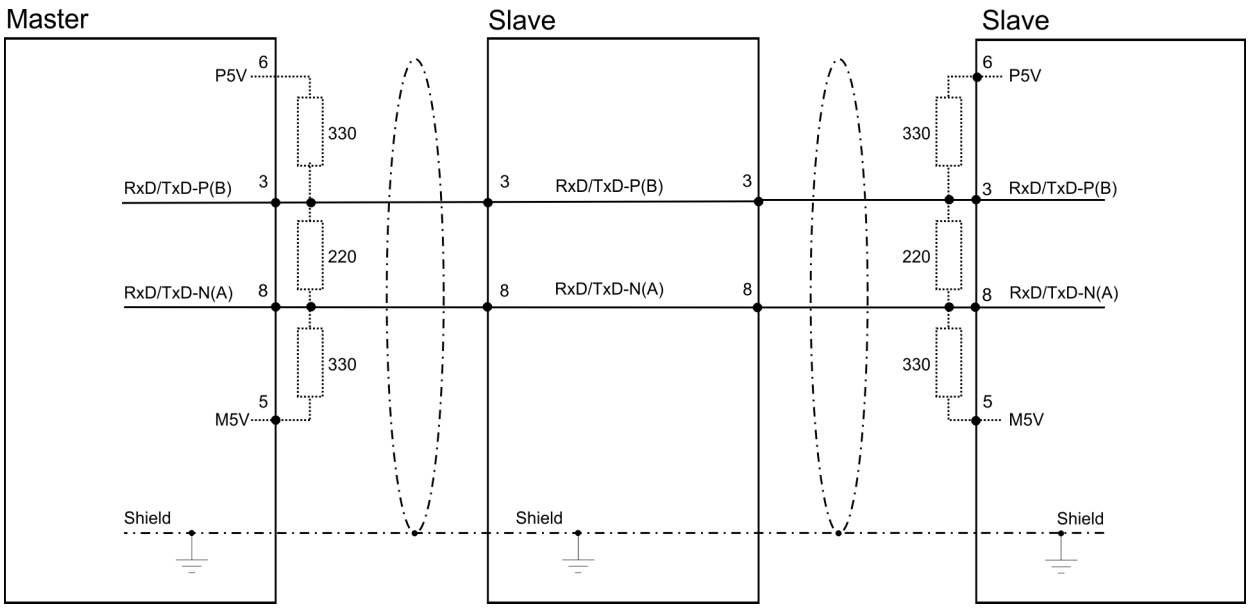
- A PROFIBUS DP network may only be built up in linear structure.
- PROFIBUS DP consists of minimum one segment with at least one master and one slave.
- A master has always been deployed together with a CPU.
- PROFIBUS supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.
- The max. segment length depends on the transfer rate:
  9.6 ... 187.5bit/s → 1000m
  500kbit/s → 400m
  1.5Mbit/s → 200m
  3 ... 12Mbit/s → 100m
- Max. 10 segments may be built up. The segments are connected via repeaters. Every repeater counts for one participant.
- The bus respectively a segment is to be terminated at both ends.
- All participants are communicating with the same transfer rate. The slaves adjust themselves automatically on the transfer rate.

Transfer medium

- As transfer medium PROFIBUS uses an isolated twisted-pair cable based upon the RS485 interface.
- The RS485 interface is working with voltage differences. Though it is less irritable from influences than a voltage or a current interface. You are able to configure the network as well linear as in a tree structure.
- Max. 32 participants per segment are permitted. Within a segment the members are linear connected. The segments are connected via repeaters. The maximum segment length depends on the transfer rate.
- PROFIBUS DP uses a transfer rate between 9.6kbit/s and 12Mbit/s, the slaves are following automatically. All participants are communicating with the same transfer rate.
- The bus structure under RS485 allows an easy connection res. disconnection of stations as well as starting the system step by step. Later expansions don't have any influence on stations that are already integrated. The system realizes automatically if one partner had a fail down or is new in the network.

Bus connection

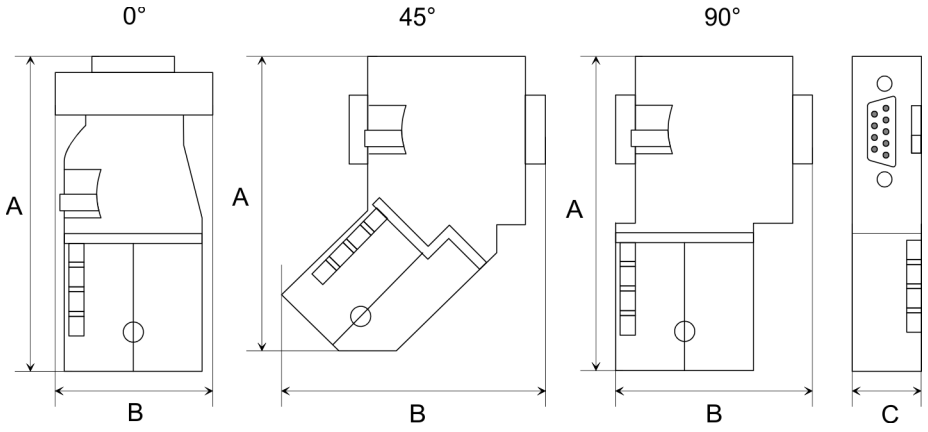
The following picture illustrates the terminating resistors of the respective start and end station.



The PROFIBUS line has to be terminated with its ripple resistor. Please make sure to terminate the last participants on the bus at both ends by activating the terminating resistor.

EasyConn bus connector

In PROFIBUS all participants are wired parallel. For that purpose, the bus cable must be feed-through. Via the order number 972-0DP10 you may order the bus connector "EasyConn" from VIPA. This is a bus connector with switchable terminating resistor and integrated bus diagnostic.



Dimensions in mm	0°	45°	90°
A	64	61	66
B	34	53	40
C	15.8	15.8	15.8

To connect this EasyConn plug, please use the standard PROFIBUS cable type A (EN50170). Starting with release 5 you also can use highly flexible bus cable:

Lapp Kabel order no: 2170222, 2170822, 2170322.

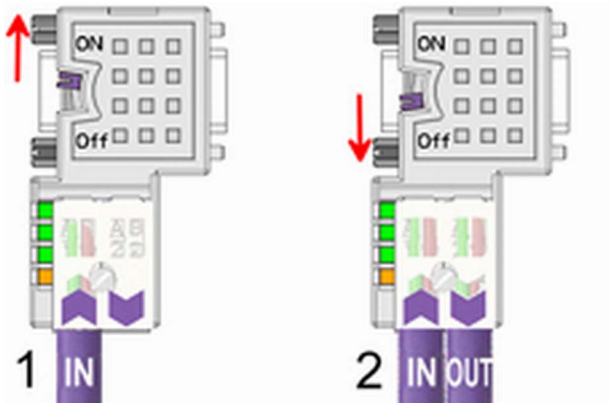
With the order no. 905-6AA00 VIPA offers the "EasyStrip" de-isolating tool that makes the connection of the EasyConn much easier.

Dimensions in mm

Termination with "EasyConn"

The "EasyConn" bus connector is provided with a switch that is used to activate a terminating resistor.

Wiring



[1] 1./last bus participant
[2] further participants

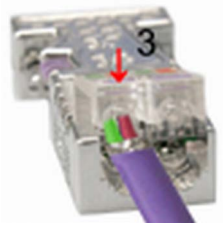
CAUTION!
The terminating resistor is only effective, if the connector is installed at a bus participant and the bus participant is connected to a power supply.

The tightening torque of the screws to fix the connector to a device must not exceed 0.02Nm!

A complete description of installation and deployment of the terminating resistors is delivered with the connector.

Assembly

1. Loosen the screw.
2. Lift contact-cover.
3. Insert both wires into the ducts provided (watch for the correct line colour as below!)
4. Please take care not to cause a short circuit between screen and data lines!



5. Close the contact cover.
6. Tighten screw (max. tightening torque 0.08Nm).

The green line must be connected to A, the red line to B!

8.8 Commissioning and Start-up behavior

Start-up on delivery

In delivery the CPU is overall reset. The PROFIBUS part is deactivated and its LEDs are off after Power ON.

Online with bus parameter without slave project

The DP master can be served with bus parameters by means of a hardware configuration. As soon as these are transferred the DP master goes online with his bus parameter. This is shown by the RUN LED. Now the DP master can be contacted via PROFIBUS by means of his PROFIBUS address. In this state the CPU can be accessed via PROFIBUS to get configuration and DP slave project.

Slave configuration

If the master has received valid configuration data, he switches to *Data Exchange* with the DP slaves. This is indicated by the DE-LED.

CPU state controls DP master

After PowerON respectively a receipt of a new hardware configuration the configuration data and bus parameter were transferred to the DP master. Dependent on the CPU state the following behavior is shown by the DP master:

Master behavior at CPU STOP

- The global control command "Clear" is sent to the slaves by the master. Here the DE-LED is blinking.
- DP slaves with fail safe mode were provided with output telegram length "0".
- DP slaves without fail safe mode were provided with the whole output telegram but with output data = 0.
- The input data of the DP slaves were further cyclically transferred to the input area of the CPU.

Master behavior at CPU RUN

- The global control command "Operate" is sent to the slaves by the master. Here the DE-LED is on.
- Every connected DP slave is cyclically attended with an output telegram containing recent output data.
- The input data of the DP slaves were cyclically transferred to the input area of the CPU.

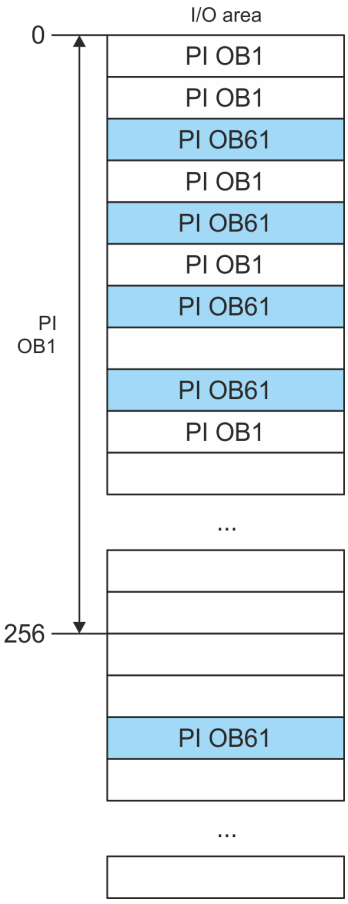
9 *Option:* Deployment - Isochronous

Isochronous only via SPEED7 Studio
Please note that the use of Isochronous together with motion functions is only possible via the SPEED7 Studio.

Activate additional functions by means of VSC in the CPU
In order to use the additional functions, you must activate them by means of a VSC storage card from VIPA. By inserting the VSC storage card and and then an overall reset, the corresponding additional functions are activated.

'Overview' on page 85

9.1 Process Image



The CPU determines by reading the input values the current state of a system and achieves by selectively controlling of output values the required system behavior (functionality). If the operand areas of the process data are accessed by the user program, so an area of the system memory is accessed. This memory area is called *Process image* (PI). Direct access to the process image has the advantage that for the duration of cyclic program processing the CPU has a consistent image of process signals. The update of the process image can be interrupted by an organization block with higher priority. This is only possible on the consistency points, which are defined by module limits. The OB 61 has the highest priority.

Process images

The CPU has an I/O data area to store process images. There are the following PIs:

PI	Remark
PI OB 1	■ Triggering: Internal ■ Assignment: OB 1 ■ Start address: 0 ■ End address configurable ■ Gaps: Usage by PI OB 61 permitted
PI OB 61	■ Triggering: Internal ■ Assignment: OB 61 ■ For system SLIO and EtherCAT bus ■ Any address can be mapped

Here is valid:

- Each address can be assigned to only one PI.
- The data of a PI are consistent for the duration of the OB for which the process image has been configured.
- The input data of the configured process images for the corresponding OB are read before starting the OB and the output data are written after completion of the OB.
- The data of a PI can be accessed from any OB.

– *By assigning the address ranges of the corresponding modules and EtherCAT slaves (S7 addresses) to the process image of the OB 61, these are synchronized in clock cycles. The other addresses may be assigned to the PI OB 1.*

– *In principle, the clock synchronicity does not support analog modules on the SLIO bus system. You can add analog modules to the process image of the OB 61 record, tape. However, their input and output data are not processed asynchronously. If you use modules on the System SLIO bus system, which do not support isochronous, you will receive the diagnostic message 0xEB05 (bus configuration for isochronous process diagram not suitable). The module's error LED blinks.*

9.2 Isochronous

Isochronous and Sync-signal

The detection or output of input or output signals synchronized with a reference signal in the central system and decentralized via a connected field bus system is called *Isochronous*. In decentralized automation structures many processing cycles are not synchronized to each other. In the process input signals are detected, evaluated in the user program and the according reactions are connected to the output components. Here the cycles correlate to each other. Due to the telegram runtime of the corresponding bus, the process reaction time can vary respectively the process data are not transferred at a consistent time.

For the synchronization of I/O data, a basic clock is required. This is derived as *Sync-signal* from the EtherCAT system. With each *Sync-Signal*, all input data are buffered and the output data are output, i.e. all data of the process image belong together logically and temporally. The *Sync-signal* serves as a clock generator within its cycle the following functions are performed:

- The current centralized and decentralized input data are latched.
- The output data, which were cached *Sync-signal* cycle are centralized and decentralized output. All output data get simultaneously active.

The isochronous functionality on EtherCAT is called Distributed Clocks (DC). For synchronization on EtherCAT DC-capable EtherCAT slaves are required, where DC is also activated. If only modules on the System SLIO backplane bus are to be synchronized, so to generate the Sync-signal, you have to configure EtherCAT without slaves.

Synchronous cycle interrupt OB 61

With OB 1 no isochronous is possible. For this the high-priority OB 61 is to be used. For isochronous use the OB 61 is started in a defined time interval. The OB 61 is executed according to the following steps, whereby the processing of these steps must be within one cycle so that it is ensured that the output data can be output at the next *Sync-Signal*.

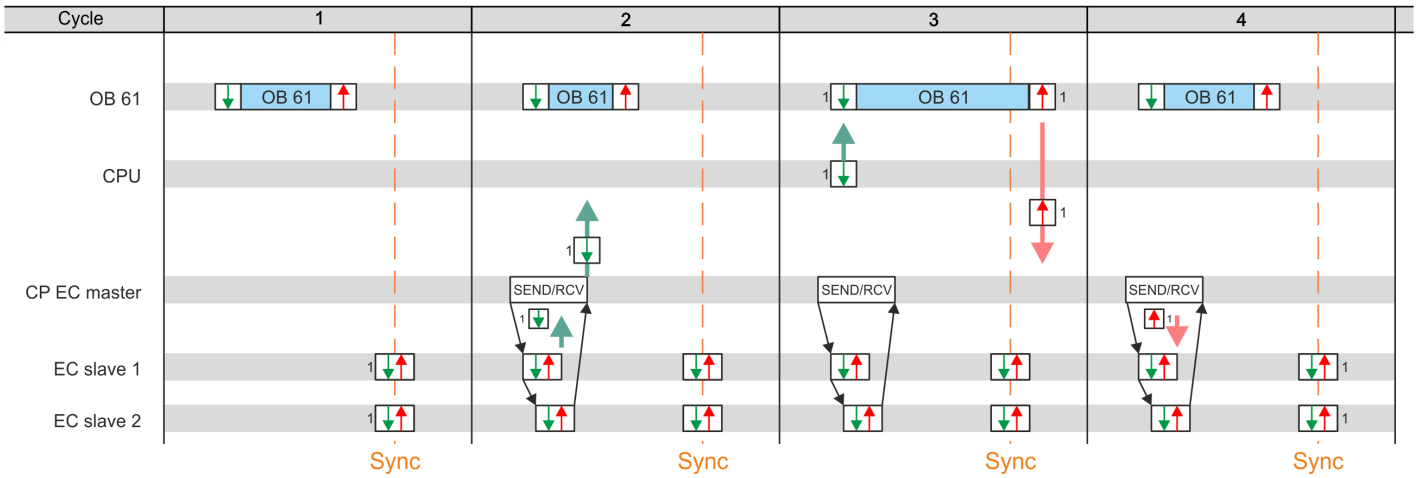
1. The input process image of the OB 61 is refreshed.
2. The user program of the OB 61 is executed.
3. The output process image of the OB 61 is refreshed.

A maximum of 1 EtherCAT cycle can elapse before data changes can be detected to the next following *Sync-Signal*.

CAUTION!
If a cycle time is exceeded in the OB 61 based on the user program, you can configure an OB 80 (time error). It is called on time error. If this is not available, the CPU switches to STOP.

Sequence OB 61 with EtherCAT master system

The following figure shows a sequence on EtherCAT, how the input data from cycle 1 are buffered and transferred and the resulting output data are output.



Input process image with the input data from cycle 1

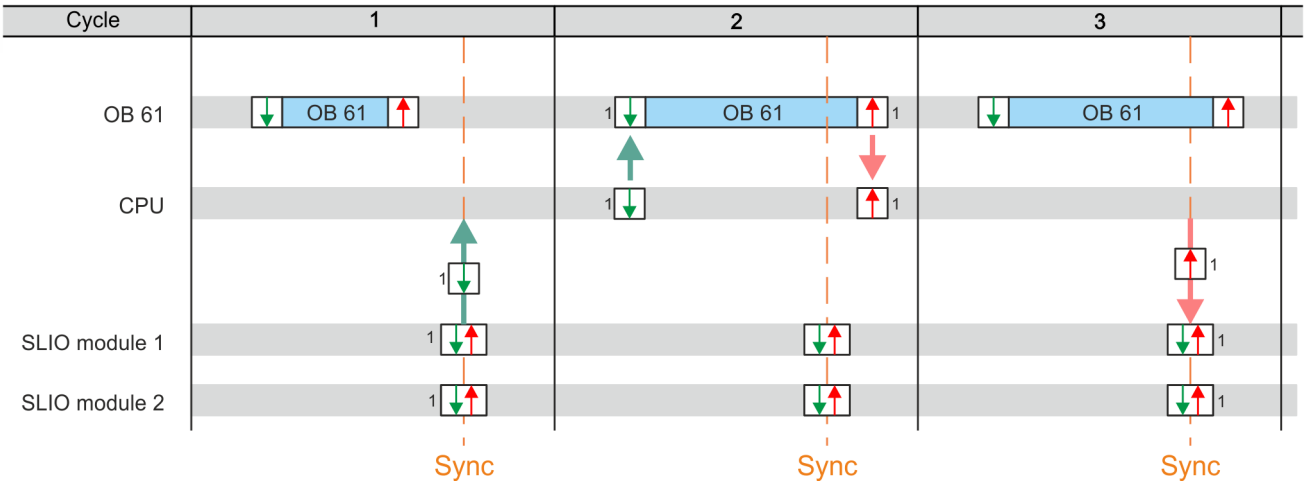
Output process image resulting from the input data from cycle 1

Sync *Sync-Signal* - Sync / Freece I / O

- Cycle 1: The input signals are buffered at the time of the *Sync-Signals* at the EtherCAT input modules and forwarded to the CPU.
- Cycle 2: The input process image is forwarded to the CPU via SEND / RECEIVE via the EtherCAT master.
- Cycle 3: The input process image is transferred to OB 61, OB 61 is processed, and the output process data is transferred from the CPU to the EtherCAT master.
- Cycle 4: The output process image is transferred to the EtherCAT slave via SEND / RECEIVE and is switched to the outputs of the EtherCAT output modules at the time of the *Sync-Signals*.

Sequence OB 61 with System SLIO modules

The following figure shows a sequence in the System SLIO as to how the input data from cycle 1 are buffered and transferred and the resulting output data are output.

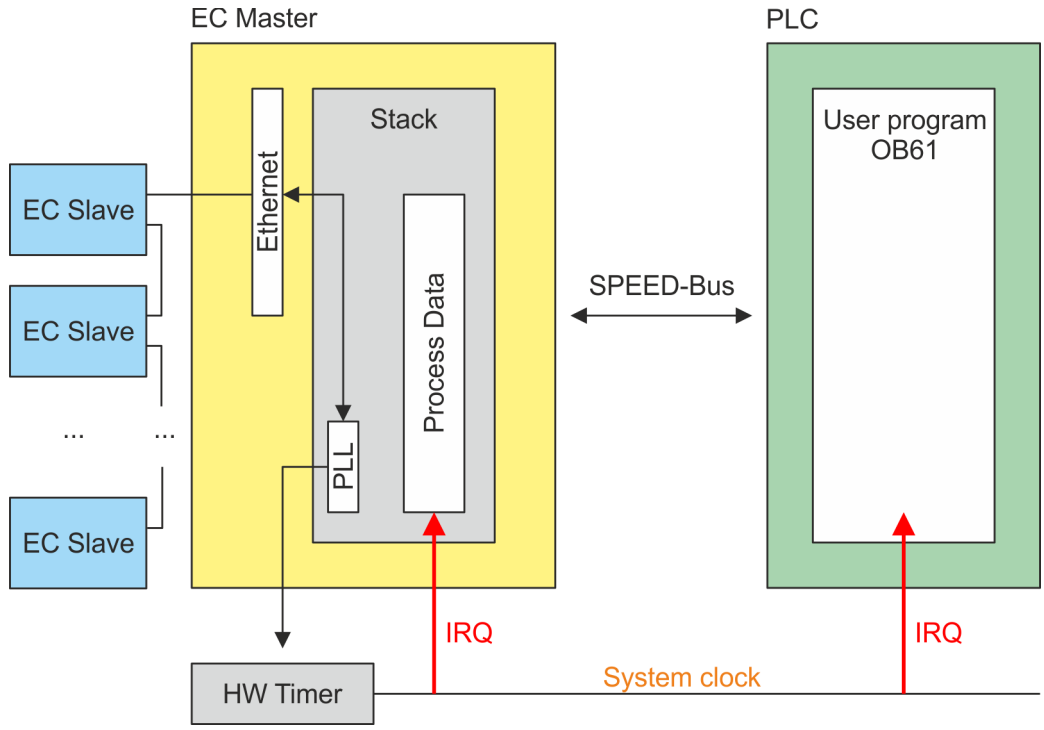


Input process image with the input data from cycle 1
Output process image resulting from the input data from cycle 1
Sync *Sync-Signal* - Sync / Freece I / O

- Cycle 1: The input signals are read from the input modules with the *Sync-signal* and forwarded to the CPU.
- Cycle 2: The input process image is transferred to the OB 61, the OB 61 is processed and then the output process image is forwarded to the SLIO modules.
- Cycle 3: At the time of the *Sync-Signals* the outputs are enabled on the SLIO output modules.

Mechanism of synchronization

The CPU components PLC and EtherCAT Master are synchronized by an interrupt. This interrupt is generated from the SLIO bus timer and the EtherCAT bus cycle time. The synchronization of EtherCAT slaves happens by DC. EtherCAT slaves that do not support DC are not synchronized. With VIPA the 1. DC-capable EtherCAT slave in the network always has the DC reference time. The synchronization between the DC-reference time and the EtherCAT master takes place in the EtherCAT master. Here also the System SLIO bus timer is synchronized. The EtherCAT bus cycle time can be configured in *SPEED7 Studio*.

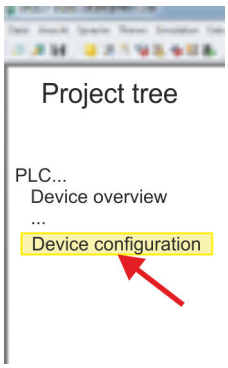


9.3 Configuration

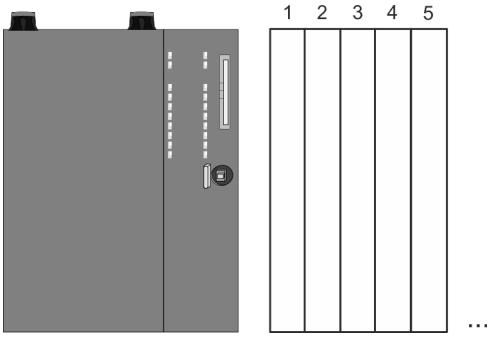
9.3.1 Hardware configuration CPU

Proceeding

1. Start the *SPEED7 Studio*.
2. Create a new project in the *Work area* with *'New project'*.
   ⇨ A new project is created and the view *'Devices and networking'* is shown.
3. Click in the *Project tree* at *'Add new device ...'*.
   ⇨ A dialog for device selection opens.



4. Select from the *'Device templates'* your CPU and click at [OK].
   ⇨ The CPU is inserted in *'Devices and networking'* and the *'Device configuration'* is opened.



Device configuration

Slot	Module	...	...	...	...
0	CPU 015-CEFNR00				

-X1	PG_OP_Ethernet				
-X3	MPI interface				
...	...			...	

9.3.2 Activate isochronous

Please consider the additional functions in the SPEED7 Studio can only be activated, if you have valid license for these functions!

Proceeding

1. Click at the CPU in the *'Device configuration'* and select *'Context menu ➔ Components properties'*.

   ⇨ The properties dialog of the CPU is opened

2. Click at *'Feature Sets'* and activate at *'Motion Control'* the parameter *'EtherCAT Master functionality+Motion Control+...'*.

3. Confirm your input with [OK].

   ⇨ The additional functions are now available in your project. More information about the usage may be found in the online help of the *SPEED7 Studio*.

CAUTION!
Please note that with any change in the feature set settings in the *SPEED7 Studio*, due to the system, the EtherCAT fieldbus system together with the motion control configuration are removed from your project!

Activate additional functions by means of VSC in the CPU
In order to use the additional functions, you must activate them by means of a VSC storage card from VIPA. By inserting the VSC storage card and and then an overall reset, the corresponding additional functions are activated.

'Overview' on page 85

OB 60

By activating of the function *'Motion Control'* in the *SPEED7 Studio* the OB 60 is automatically created. The OB is used internally and can not be edited. It used to manage the service data objects (SDO) and diagnostic data. The OB 60 has a higher priority than OB 1. The cycle time for this OB can be configured in the *SPEED7 Studio*. Please note that no peripheral direct access is possible in OB 60.

OB 61

By activating of the function *'Motion Control'* in the *SPEED7 Studio* the OB 61 is automatically created. For the OB a separate process image PI OB 61 is created, which data are consistent during the execution of the OB. OB 61 has a higher priority than OB 60. Please note that no peripheral direct access is possible in OB 61. Within the OB 61 should be the functions which are synchronously should be executed. The following blocks are permitted in the OB 61:

- FBs and FCs without restrictions
- SFCs
  - SFC 20 - BLKMOV
  - SFC 21 - FILL
  - SFC 47 - WAIT
  - SFC 53 - uS_Tick
  - SFC 64 - TIME_TCK
- SFBs
  - SFB 0 - CTU
  - SFB 1 - CTD
  - SFB 2 - CTUD
  - SFB 3 - TP
  - SFB 4 - TON
  - SFB 5 - TOF
  - SFB 7 - TIMEMESS

10 Configuration with Siemens SIMATIC Manager

10.1 SIMATIC Manager - General

In this part the project engineering of the VIPA CPU in the Siemens SIMATIC Manager is shown. Here only the basic usage of the Siemens SIMATIC Manager together with a VIPA CPU is shown. In the SIMATIC Manager your VIPA PLCs may be configured and linked. For diagnostics online tools are available.

More information can be found in the online help respectively in documentation of the Siemens SIMATIC Manager.

10.2 SIMATIC Manager - Hardware configuration - CPU

Precondition

- The configuration of the CPU takes place at the Siemens *'hardware configurator'*. The hardware configurator is part of the Siemens SIMATIC Manager. It serves for project engineering.
- Please use for configuration the Siemens SIMATIC Manager V 5.5 SP2 and up.
- The configuration of the System SLIO CPU happens in the Siemens SIMATIC Manager by means of a virtual PROFINET IO device *'VIPA SLIO CPU'*. The *'VIPA SLIO System'* is to be installed in the hardware catalog by means of the GSDML.

For project engineering a thorough knowledge of the Siemens SIMATIC Manager and the Siemens hardware configurator is required!

Installing the IO device *'VIPA SLIO System'*

The installation of the PROFINET IO devices *'VIPA SLIO CPU'* happens in the hardware catalog with the following approach:

1. Go to the service area of www.vipa.com.
2. Load from the download area at *'Config files ➔ PROFINET'* the according file for your System SLIO.
3. Extract the file into your working directory.
4. Start the Siemens hardware configurator.
5. Close all the projects.
6. Select *'Options ➔ Install new GSD file'*
7. Navigate to your working directory and install the according GSDML file.
   ⇨ After the installation according PROFINET IO device can be found at *'PROFINET IO ➔ Additional field devices ➔ I/O ➔ VIPA SLIO System'*

Proceeding

In the Siemens SIMATIC Manager the following steps should be executed:

1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at *'Slot'*-Number 2 the CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).

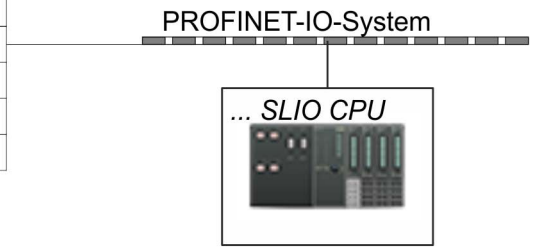
Slot	Module
1	
2	**CPU 31...**
X1	*MPI/DP*
X2	*PN-IO*
X2...	*Port 1*
X2...	*Port 2*
3	

4. Click at the sub module *'PN-IO'* of the CPU.
5. Select *'Context menu ➔ Insert PROFINET IO System'*.

Slot	Module
1	
2	**CPU ...**
X...	*PN-IO*
3	

6. Create with [New] a new sub net and assign valid address data
7. Click at the sub module *'PN-IO'* of the CPU and open with *'Context menu ➔ Properties'* the properties dialog.
8. Enter at *'General'* a device name. The device name must be unique at the Ethernet subnet.

Slot	Module
1	
2	**CPU ...**
X...	*PN-IO*
3	



Slot	Module	Order number	
0	***... SLIO CPU ...***	***015-...***	
X2	*015-...*		
1			
2			
3			
...			

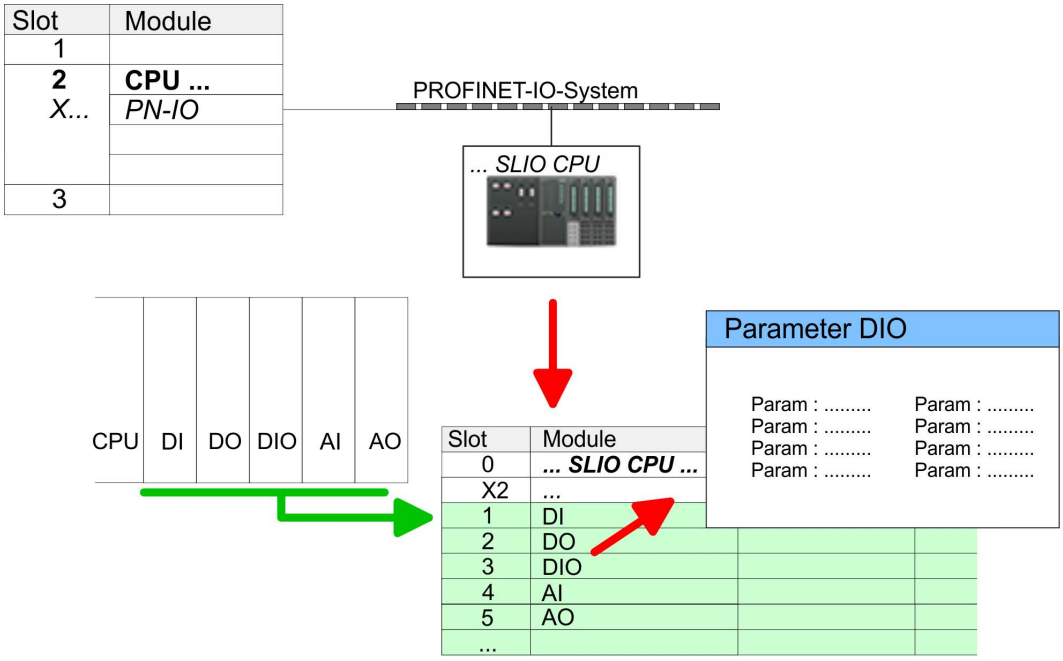
9. Navigate in the hardware catalog to the directory *'PROFINET IO ➔ Additional field devices ➔ I/O ➔ VIPA SLIO System'* and connect the IO device *'015-CEFNR00 CPU'* to your PROFINET system.

   ⇨ In the slot overview of the PROFINET IO device *'VIPA SLIO CPU'* the CPU is already placed at slot 0. From slot 1 you can place your System SLIO modules.

10.3 SIMATIC Manager - Hardware configuration - I/O modules

Hardware configuration of the modules

Starting with slot 1 place in the slot overview of the PROFINET IO device *'VIPA SLIO CPU'* your System SLIO modules in the plugged sequence. To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU.



Parametrization

For parametrization double-click during the project engineering at the slot overview on the module you want to parametrize. In the appearing dialog window you may set the wanted parameters. Here you can make your parameter settings.

Parametrization during runtime

By using the SFCs 55, 56 and 57 you may alter and transfer parameters for wanted modules during runtime. For this you have to store the module specific parameters in so called "record sets". More detailed information about the structure of the record sets is to find in the according module description.

10.4 SIMATIC Manager - Hardware configuration - Ethernet PG/OP channel

Overview

The CPU has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU.

- The Ethernet PG/OP channel (X1/X5) is designed as switch. This enables PG/OP communication via the connections X1 and X5.
- The Ethernet PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc.
- At the first commissioning respectively after a factory reset the Ethernet PG/OP channel has no IP address.
- For online access to the CPU via the Ethernet PG/OP channel, valid IP address parameters have to be assigned to this. This is called "initialization".
- This can be done with the *SPEED7 Studio*.

Assembly and commissioning

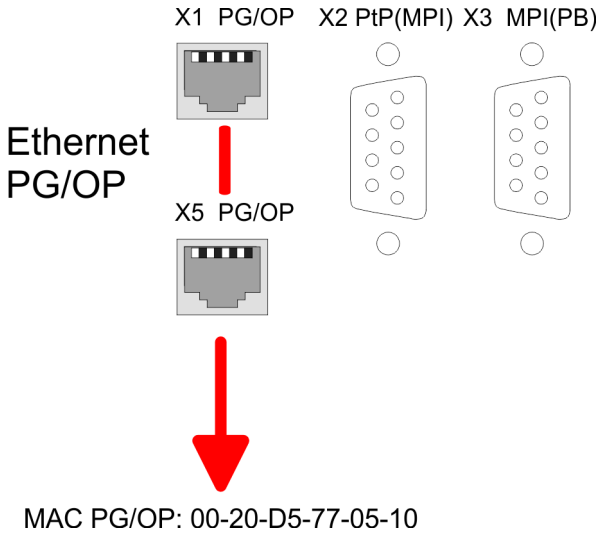
1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.

3. Connect the one of the Ethernet jacks (X1, X5) of the Ethernet PG/OP channel to Ethernet.
4. Switch on the power supply.
   ⇨ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization" via Online functions

The initialization via the Online functions takes place with the following proceeding:

Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".



MAC PG/OP: 00-20-D5-77-05-10

Assign IP address parameters

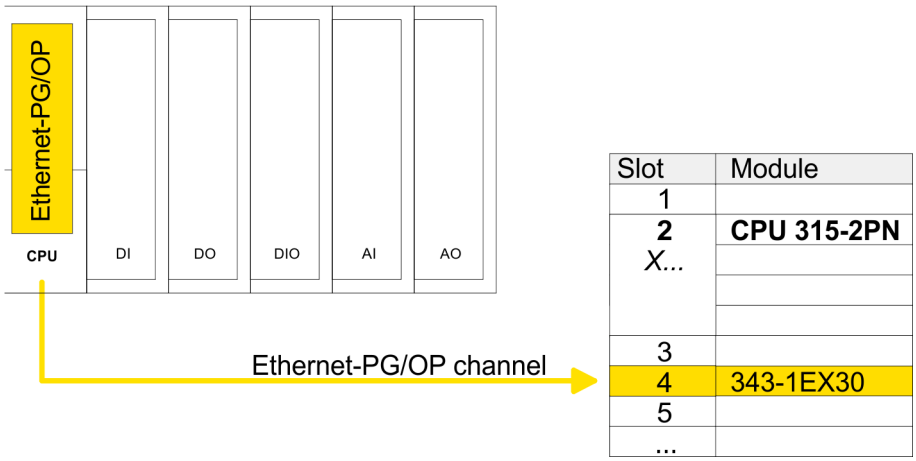
You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the Siemens SIMATIC Manager starting with version V 5.3 & SP3 with the following proceeding:

1. Start the Siemens SIMATIC Manager and set via *'Options ➔ Set PG/PC interface'* the access path to *'TCP/IP -> Network card ....'*.
2. Open with *'PLC ➔ Edit Ethernet Node n'* the dialog window with the same name.
3. To get the stations and their MAC address, use the [Browse] button or type in the MAC Address. The Mac address may be found at the 1. label beneath the front flap of the CPU.
4. Choose if necessary the known MAC address of the list of found stations.
5. Either type in the IP configuration like IP address, subnet mask and gateway.
6. Confirm with [Assign IP configuration].
   ⇨ Direct after the assignment the Ethernet PG/OP channel may be reached online by these address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.

Take IP address parameters in project

1. Open the Siemens hardware configurator and configure the Siemens CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).
2. For the Ethernet PG/OP channel you have to configure at slot 4 a Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
3. Open the property window via double-click on the CP 343-1EX30 and enter for the CP at *'Properties'* the IP address data, which you have assigned before.
4. Assign the CP to a *'Subnet'*. Without assignment the IP address data are not used!

5. Transfer your project.

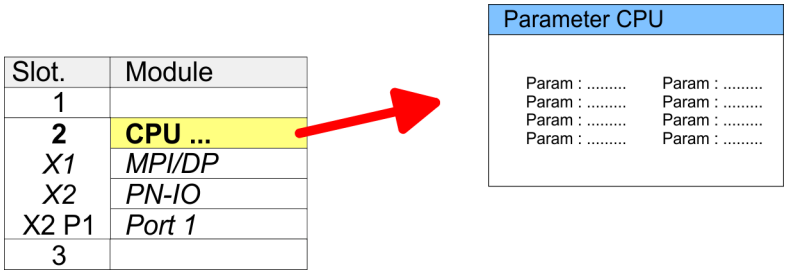


10.5 SIMATIC Manager - Hardware configuration - Parametrization

10.5.1 Standard CPU parameters

Parametrization via Siemens CPU 315-2 PN/DP

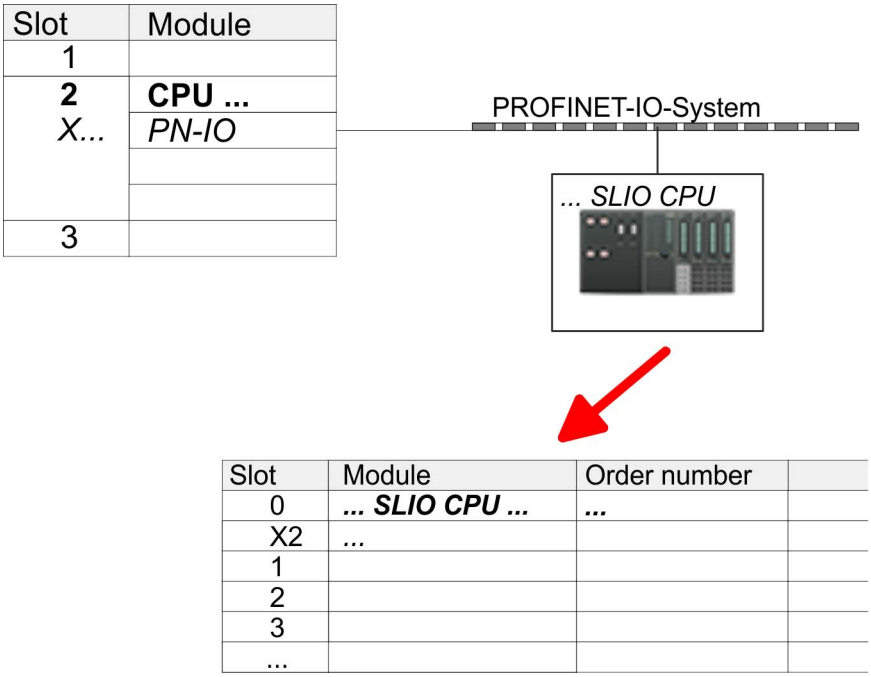
Since the CPU is to be configured as Siemens CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2) in the hardware configurator, the standard parameters of the VIPA CPU may be set with "Object properties" of the CPU 315-2 PN/DP during hardware configuration. Via a double-click on the CPU 315-2 PN/DP the parameter window of the CPU may be accessed. Using the registers you get access to every standard parameter of the CPU.
Chapter 4.7.1 'Parameter CPU' on page 64



10.5.2 VIPA specific CPU parameters

Except of the VIPA specific CPU parameters the CPU parametrization takes place in the parameter dialog of the CPU 315-2 PN/DP from Siemens. After the hardware configuration of the CPU you can set the parameters of the CPU in the virtual IO device *'VIPA SLIO CPU'*. Via double-click at the *'VIPA SLIO CPU'* the properties dialog is opened Here the following parameters may be accessed:

- Function X2 (PtP/MPI)
- MPI address X2
- MPI Baud rate X2
- Number of retentive memory/timer/counter



10.5.3 Parameters for MPI/DP

The properties dialog of the MPI(PB) interface X3 is opened via a double click to the sub module MPI/DP

To switch the interface to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is activated. *Chapter 4.14 'Deployment storage media - VSD, VSC' on page 85*

10.6 SIMATIC Manager - Project transfer

Overview

There are the following possibilities for project transfer into the CPU:

- Transfer via MPI (optional via PROFIBUS)
- Transfer via Ethernet
- Transfer via memory card

To switch the interface X3 MPI(PB) to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is activated. *Chapter 4.14 'Deployment storage media - VSD, VSC' on page 85*

10.6.1 Transfer via MPI / optional PROFIBUS

General

For transfer via MPI / optional PROFIBUS there are the following 2 interface:

‘X3: MPI(PB) interface’ on page 42

‘X2: PtP(MPI) interface’ on page 42

With an overall reset CPU the configuration via X2 PtP(MPI) is not possible!

Net structure

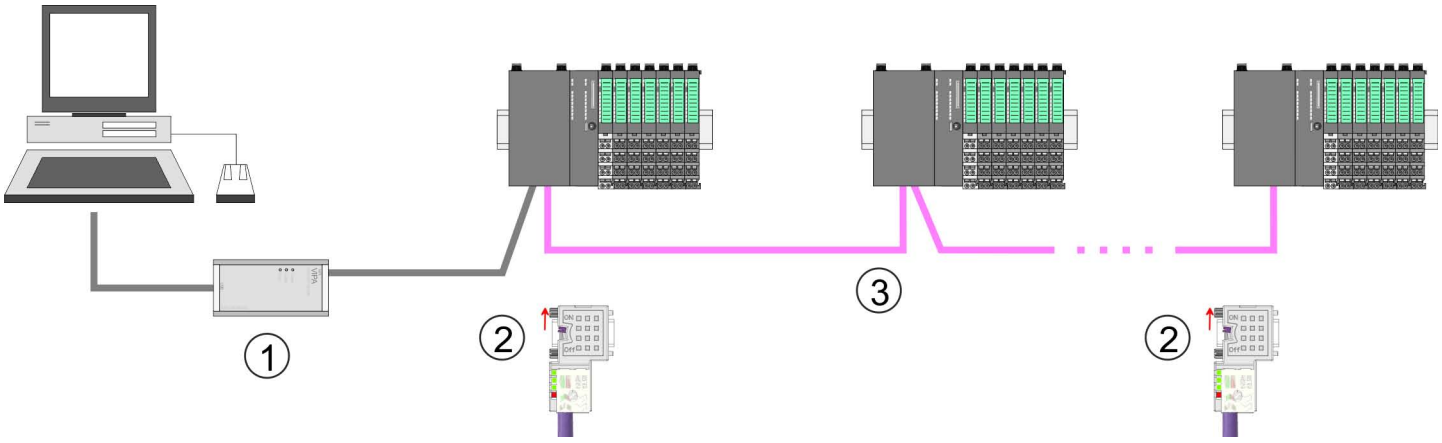
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs with 187.5kbaud. VIPA CPUs are delivered with MPI address 2.

MPI programming cable

The MPI programming cables are available at VIPA in different variants. The cables provide a RS232 res. USB plug for the PC and a bus enabled RS485 plug for the CPU. Due to the RS485 connection you may plug the MPI programming cables directly to an already plugged plug on the RS485 jack. Every bus participant identifies itself at the bus with an unique address, in the course of the address 0 is reserved for programming devices.

Terminating resistor

A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment. Please make sure that the participants with the activated terminating resistors are always power supplied. Otherwise it may cause interferences on the bus.



1 MPI programming cable
2 Activate the terminating resistor via switch
3 MPI/PROFIBUS network

Approach transfer via MPI interface

1. Connect your PC to the MPI jack of your CPU via a MPI programming cable.
2. Load your project in the SIMATIC Manager from Siemens.
3. Choose in the menu *‘Options ➔ Set PG/PC interface’*.
4. Select in the according list the "PC Adapter (MPI)"; if appropriate you have to add it first, then click on [Properties].

5. Set in the register MPI the transfer parameters of your MPI net and type a valid *address.*
6. Switch to the register *Local connection*.
7. Set the COM port of the PCs and the transfer rate 38400baud for the MPI programming cable from VIPA.
8. Transfer your project via *'PLC ➔ Load to module'* via MPI to the CPU and save it with *'PLC ➔ Copy RAM to ROM'* on a memory card if one is plugged.

Proceeding Transfer via PROFIBUS interface

To switch the interface to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is activated.

1. Connect your PC to the MPI(PB) jack X3 of your CPU via a MPI programming cable.
2. Load your project in the Siemens SIMATIC Manager.
3. Choose in the menu *'Options ➔ Set PG/PC interface'*.
4. Select in the according list the "PC Adapter (PROFIBUS)"; if appropriate you have to add it first, then click at [Properties].
5. Set in the register PROFIBUS the transfer parameters of your PROFIBUS net and enter a valid *PROFIBUS address*. The *PROFIBUS address* must be assigned to the DP master by a project before.
6. Switch to the register *Local connection*.
7. Set the COM port of the PCs and the transfer rate 38400baud for the MPI programming cable from VIPA.
8. Transfer your project via *'PLC ➔ Load to module'* via PROFIBUS to the CPU and save it with *'PLC ➔ Copy RAM to ROM'* on a memory card if one is plugged.

Transfer via PROFIBUS is available by DP master, if projected as master and assigned with a PROFIBUS address before. In slave operation you have also to enable the option 'Test , Commissioning, Routing ' when selecting the slave mode.

10.6.2 Transfer via Ethernet

For transfer via Ethernet the CPU has the following interfaces:

- X1, X5: Ethernet PG/OP channel (switch)

Initialization

So that you may the according Ethernet interface, you have to assign IP address parameters by means of the "initialization". *Chapter 10.4 'SIMATIC Manager - Hardware configuration - Ethernet PG/OP channel' on page 217*

Transfer

1. For the transfer, connect, if not already done, the appropriate Ethernet jack to your Ethernet.
2. Open your project with the Siemens SIMATIC manager.

3. Set via *'Options ➔ Set PG/PC interface'* the access path to "TCP/IP → Network card .... ".
4. Click to *'PLC ➔ Download'*. The dialog "Select target module" is opened. Select your target module and enter the IP address parameters of the corresponding Ethernet interface. Provided that no new hardware configuration is transferred to the CPU, the entered Ethernet connection is permanently stored in the project as transfer channel.
5. With [OK] the transfer is started.

Due to the system you may get a message that the projected system differs from target system. This message may be accepted by [OK].

→ Your project is transferred and may be executed in the CPU after transfer.

10.6.3 Transfer via memory card

The memory serves as external transfer and storage medium. There may be stored several projects and sub-directories on a memory card. Please regard that your current project is stored in the root directory and has one of the following file names:

- S7PROG.WLD
- AUTOLOAD.WLD

With *'File ➔ Memory Card File ➔ New'* in the Siemens SIMATIC Manager a new wld file may be created. After the creation copy the blocks from the project blocks folder and the *System data* into the wld file.

Transfer memory card → CPU

The transfer of the application program from the memory card into the CPU takes place depending on the file name after an overall reset or PowerON.

- *S7PROG.WLD* is read from the memory card after overall reset.
- *AUTOLOAD.WLD* is read from the memory card after PowerON.

A short lightning up of the SD LED of the CPU marks the active transfer. Please regard that your user memory serves for enough space for your user program, otherwise your user program is not completely loaded and the SF LED gets on.

Transfer CPU → memory card

When a memory card has been installed, the write command stores the content of the RAM as S7PROG.WLD on the memory card.

The write command is controlled by means of the block area of the Siemens SIMATIC Manager *'PLC ➔ Copy RAM to ROM'*. The SD LED lights up during the write access. When the LED expires, the write process is finished.

If this project is to be loaded automatically from the memory card with PowerON, you have to rename this to on the memory card to *AUTOLOAD.WLD*.

Checking the transfer operation

After accessing the memory card you can find a diagnostics entry in the CPU. To monitor the diagnostics entries you choose in the Siemens SIMATIC manager *'PLC ➔ Module information'*. Via the register "Diagnostic Buffer" you reach the diagnostic window. ↳ *Chapter 4.18 'Diagnostic entries' on page 92*

10.7 SIMATIC Manager - Access to diagnostics entries

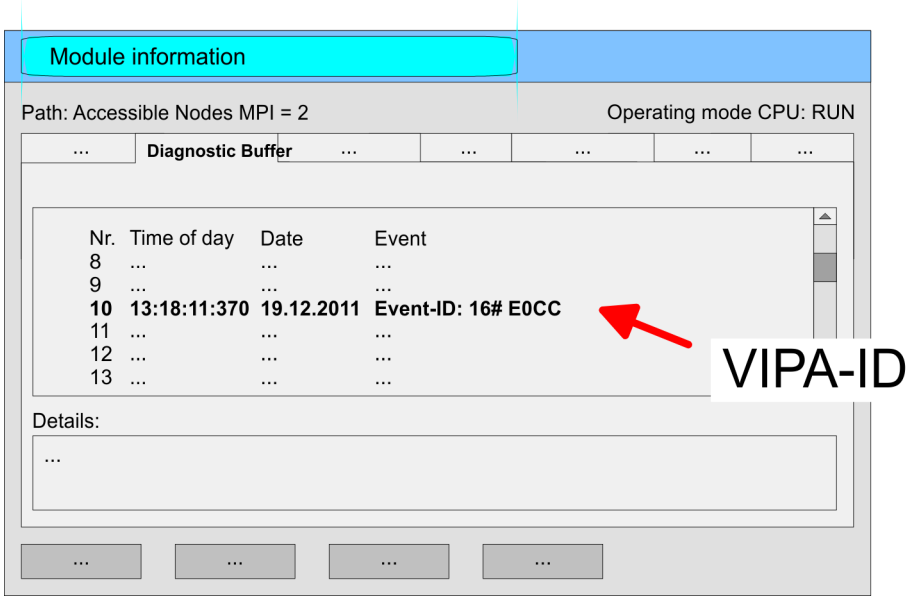
Entries in the diagnostics buffer

- You may read the diagnostics buffer of the CPU via the Siemens SIMATIC Manager.
- Besides of the standard entries in the diagnostics buffer, the VIPA CPUs support some additional specific entries as Event-IDs.
- The current content of the diagnostics buffer is stored at the memory card by means of the CMD DIAGBUF.

Every register of the module information is supported by the VIPA CPUs. More information may be found at the online help of the Siemens SIMATIC Manager.

Monitoring the diagnostic entries

To monitor the diagnostics entries you choose in the Siemens SIMATIC manager *'PLC ➔ Module information'*. Via the register "Diagnostics Buffer" you reach the diagnostics window:



The diagnostic is independent from the operating mode of the CPU. Max. 100 diagnostic entries can be stored in the CPU. *Appendix 'System specific event IDs' on page 242*

10.8 SIMATIC Manager - Deployment PROFIBUS

Chapter 8 'Option: Deployment PROFIBUS communication' on page 197

10.9 SIMATIC Manager - Deployment EtherCAT

Precondition

The configuration of the EtherCAT masters happens in the Siemens SIMATIC Manager by means of a virtual PROFINET IO device *'EtherCAT network'*. The *'EtherCAT network'* is to be installed in the hardware catalog by means of the GSDML and can be configured with the VIPA tool *SPEED7 EtherCAT Manager*.

The following preconditions must be fulfilled for the configuration of the EtherCAT master:

- GSDML for *'EtherCAT network'* is installed
- *SPEED7 EtherCAT Manager* for EtherCAT configuration is installed

Installing the IO device EtherCAT network

The installation of the PROFINET IO devices *'EtherCAT Network'* happens in the hardware catalog with the following approach:

1. Go to the service area of www.vipa.com
2. Load from the download area at *'Config files ➔ EtherCAT'* the GSDML file for your EtherCAT master.
3. Extract the files into your working directory.
4. Start the Siemens hardware configurator.
5. Close all the projects.
6. Select *'Options ➔ Install new GSD file'*.
7. Navigate to your working directory and install the according GSDML file.
   ⇨ After the installation the *'EtherCAT Network'* can be found at *'PROFINET IO ➔ Additional field devices ➔ I/O ➔ VIPA VIPA EtherCAT System'*.

Installing the *SPEED7 EtherCAT Manager*

The configuration of the PROFINET IO device *'EtherCAT Network'* happens by means of the *SPEED7 EtherCAT Manager* from VIPA. This may be found in the service area of www.vipa.com at *'Service/Support ➔ Downloads ➔ SPEED7'*.

The installation happens with the following proceeding:

1. Close the Siemens SIMATIC Manager.
2. Go to the service area of www.vipa.com
3. Load the *SPEED7 EtherCAT Manager* and unzip it on your PC.
4. For installation start the file EtherCATManager_v... .exe.
5. Select the language for the installation.
6. Accept the licensing agreement.
7. Select the installation directory and start the installation.
8. After installation you have to reboot your PC.
   ⇨ The *SPEED7 EtherCAT Manager* is installed and can now be called via the context menu of the Siemens SIMATIC Manager.

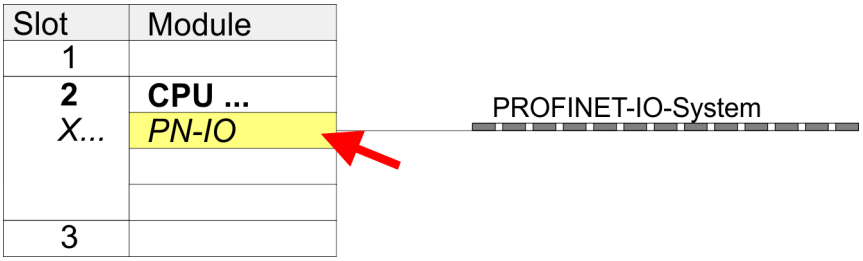
Configuration of the CPU

Slot	Module
1	
2	**CPU 31...**
X1	*MPI/DP*
X2	*PN-IO*
X2...	*Port 1*
X2...	*Port 2*
3	

To be compatible with the Siemens SIMATIC manager the following steps should be executed:

1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at *'Slot'* number 2 the CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).
4. The integrated PROFIBUS DP master (jack X3) is to be configured and connected via the sub module *'X1 MPI/DP'*.
5. The integrated EtherCAT master is to be configured via the sub module *'X2 PN-IO'* as a virtual PROFINET network.
6. Click at the sub module *'PN-IO'* of the CPU.

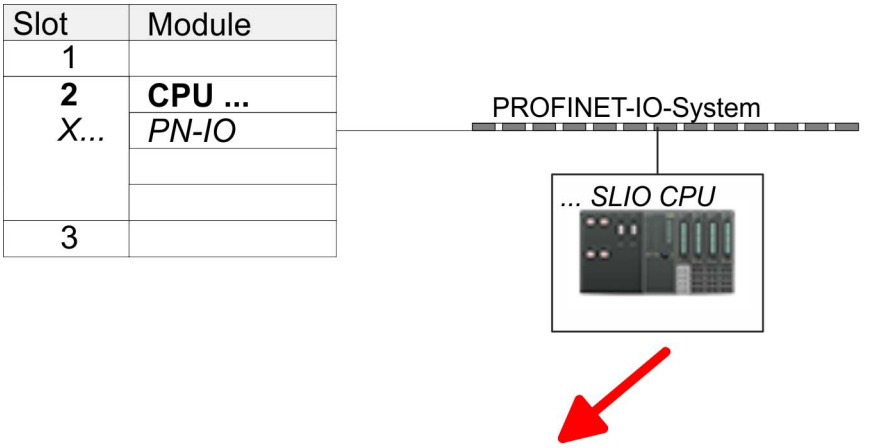
7. Select *'Context menu ➔ Insert PROFINET IO System'*.



8. Create with [New] a new sub net and assign valid address data

9. Click at the sub module *'PN-IO'* of the CPU and open with *'Context menu ➔ Properties'* the properties dialog.

10. Enter at *'General'* a *'Device name'*. The device name must be unique at the Ethernet subnet.



Slot	Module	Order number	
0	***... SLIO CPU ...***	***015-...***	
X2	*015-...*		
1			
2			
3			
...			

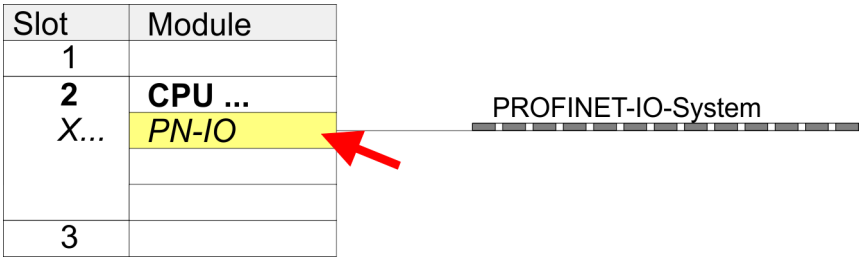
11. Navigate in the hardware catalog to the directory *'PROFINET IO ➔ Additional field devices ➔ I/O ➔ VIPA SLIO System'* and connect the IO device *'015-CFFNR00 CPU'* to your PROFINET system.

⇨ In the Device overview of the PROFINET IO device *'VIPA SLIO CPU'* the CPU is already placed at slot 0. From slot 1 you can place your System SLIO modules.

Configuration EtherCAT master

1. Click at the sub module *'PN-IO'* of the CPU.

2. Select *'Context menu ➔ Insert PROFINET IO System'*.



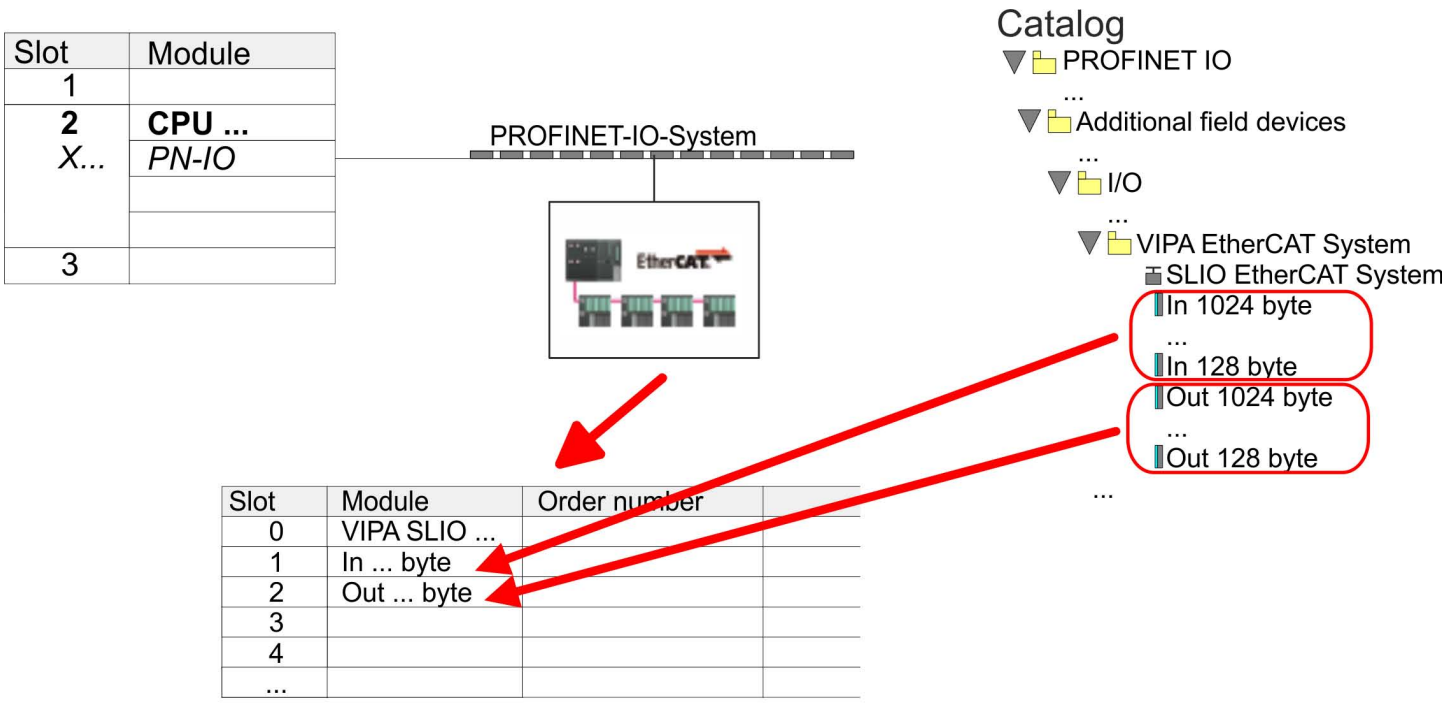
3. Create with [New] a new sub net and assign valid address data

4. Navigate in the hardware catalog to the directory *'PROFINET IO ➔ Additional field devices ➔ I/O ➔ VIPA EtherCAT System'* and connect the IO device *'SLIO EtherCAT System'* to your PROFINET system.

5. Click at the inserted IO device *'EtherCAT Network'* and define the areas for in and output by drag and dropping the according *'Out'* or *'In'* area to a slot.

   Here the following rules must be observed:

   - Input and output areas can be mixed.
   - You have a maximum of 4096byte EtherCAT process data for input and output respectively.
   - Data must be consistent in the Siemens hardware configurator, i.e. with PROFINET the maximum number of bytes may not be exceeded. Otherwise you have to connect another *'SLIO EtherCAT System'* to your PROFINET system. In the *SPEED7 EtherCAT Manager* all the areas are automatically detected and combined



6. Select *'Station ➔ Save and compile'*

   ⇨ Now you can configure your EtherCAT system with the *SPEED7 EtherCAT Manager*.

Before calling the SPEED7 EtherCAT Manager you have always to save you project with 'Station ➔ Save and compile'.

7. Click at an inserted IO device *'EtherCAT Network'* and select *'Context menu ➔ Start Device-Tool ➔ SPEED7 EtherCAT Manager'*.

   ⇨ The *SPEED7 EtherCAT Manager* opens. Here you can configure the EtherCAT master system.

   More information about the usage of the *SPEED7 EtherCAT Manager* may be found in the according manual or online help.

8. By closing the *SPEED7 EtherCAT Manager* the EtherCAT configuration is taken to the project and the *SPEED7 EtherCAT Manager* is closed. You can always edit your EtherCAT configuration in the *SPEED7 EtherCAT Manager*, since the configuration is stored in your project.

9. Choose the Siemens SIMATIC manager and transfer your project into the CPU.

   The transfer can only be done by the Siemens SIMATIC Manager - not hardware configurator!

Since slave and module parameters are transmitted by means of SDO respectively SDO Init command, the configuration remains active, until a power cycle is performed or new parameters for the same SDO objects are transferred.

With an overall reset the slave and module parameters are not reset!

11 Configuration with TIA Portal

11.1 TIA Portal - Work environment

11.1.1 General

General

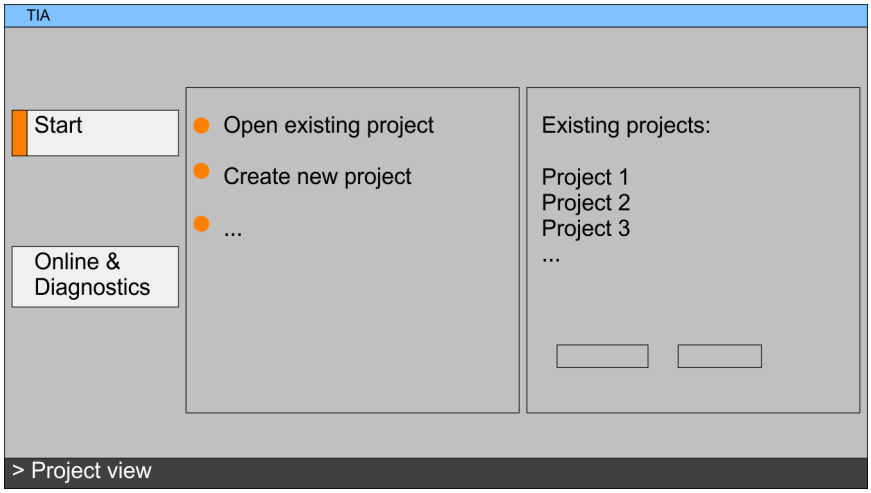
In this chapter the project engineering of the VIPA CPU in the Siemens TIA Portal is shown. Here only the basic usage of the Siemens TIA Portal together with a VIPA CPU is shown. Please note that software changes can not always be considered and it may thus be deviations to the description. TIA means **T**otally **i**ntegrated **A**utomation from Siemens. Here your VIPA PLCs may be configured and linked. For diagnostics online tools are available.

Information about the Siemens TIA Portal can be found in the online help respectively in the according online documentation.

Starting the TIA Portal

To start the Siemens TIA Portal with Windows select *'Start ➔ Programs ➔ Siemens Automation ➔ TIA ...'*

Then the TIA Portal opens with the last settings used.



Exiting the TIA Portal

With the menu *'Project ➔ Exit'* in the *'Project view'* you may exit the TIA Portal. Here there is the possibility to save changes of your project before.

11.1.2 Work environment of the TIA Portal

Basically, the TIA Portal has the following 2 views. With the button on the left below you can switch between these views:

Portal view

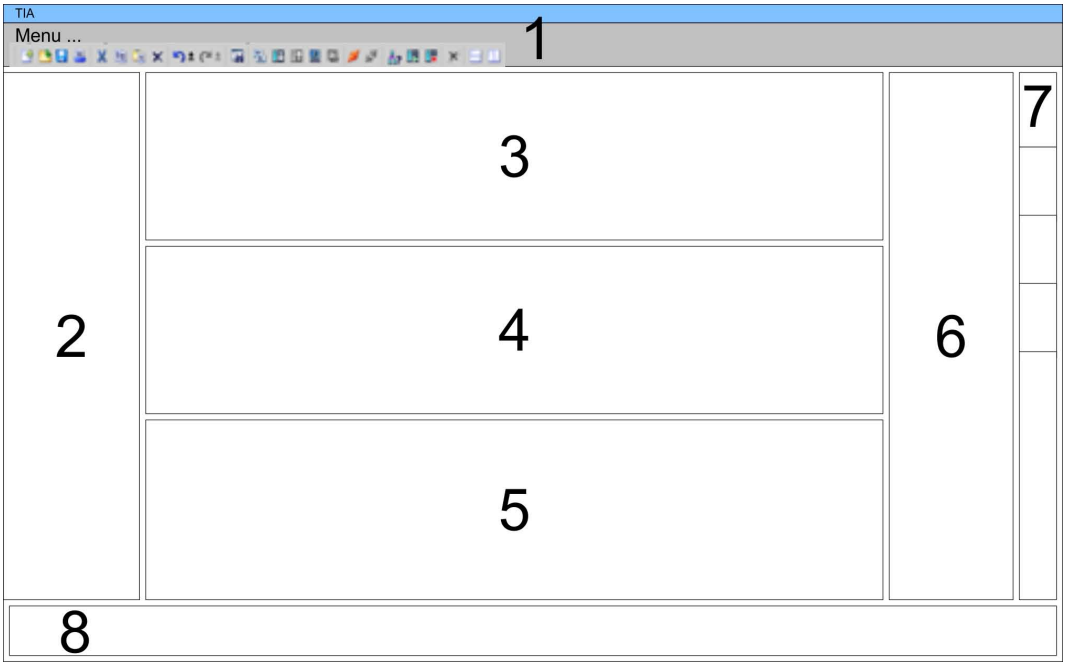
The *'Portal view'* provides a "task oriented" view of the tools for processing your project. Here you have direct access to the tools for a task. If necessary, a change to the Project view takes place automatically for the selected task.

Project view

The *'Project view'* is a "structured" view to all constituent parts of your project.

Areas of the Project view

The Project view is divided into the following areas:



1 Menu bar with toolbars
2 Project tree with Details view
3 Project area
4 Device overview of the project respectively area for block programming
5 Properties dialog of a device (parameter) respectively information area
6 Hardware catalog and tools
7 "Task-Cards" to select hardware catalog, tasks and libraries
8 Jump to Portal or Project view

11.2 TIA Portal - Hardware configuration - CPU

Overview

The hardware configuration of the CPU and its plugged modules happens in the Siemens TIA Portal by means of a virtual PROFINET IO device. For the PROFINET interface is standardized software sided, the functionality is guaranteed by including a GSDML file into the Siemens TIA Portal.

The hardware configuration of the CPU is divided into the following parts:

- Installation GSDML SLIO CPU PROFINET
- Configuration Siemens CPU
- Connection SLIO CPU as PROFINET IO device

Installation GSDML SLIO CPU for PROFINET

The installation of the PROFINET IO devices *'VIPA SLIO CPU'* happens in the hardware catalog with the following approach:

1. Go to the service area of www.vipa.com.
2. Load from the download area at *'Config files ➔ PROFINET'* the according file for your System SLIO.
3. Extract the file into your working directory.
4. Start the Siemens TIA Portal.
5. Close all the projects.
6. Switch to the *Project view*.
7. Select *'Options ➔ Install general station description file (GSD)'*.

8. Navigate to your working directory and install the according GSDML file.

   ⇨ After the installation the hardware catalog is refreshed and the Siemens TIA Portal is finished.

   After restarting the Siemens TIA Portal the according PROFINET IO device can be found at *Other field devices* > *PROFINET* > *IO* > *VIPA GmbH* > *VIPA SLIO System*.

Thus, the VIPA components can be displayed, you have to deactivate the "Filter" of the hardware catalog.

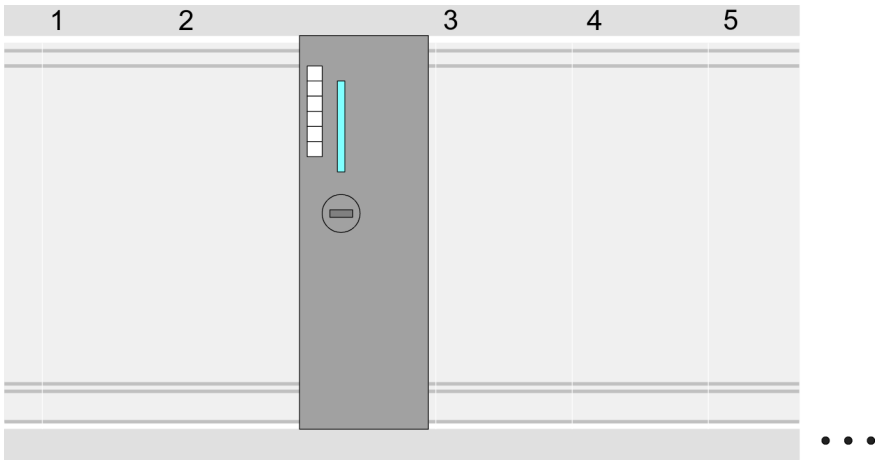
Configuration Siemens CPU

In the Siemens TIA Portal, the CPU from VIPA is to be configured as CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2) from Siemens.

1. Start the Siemens TIA Portal.
2. Create a new project in the *Portal view* with *'Create new project'*.
3. Switch to the *Project view*.
4. Click in the *Project tree* at *'Add new device'*.
5. Select the following CPU in the input dialog:

   SIMATIC S7-300 > CPU 315-2 PN/DP > 6ES7 315-2EH14-0AB0 V3.2

   ⇨ The CPU is inserted with a profile rail.



Device overview

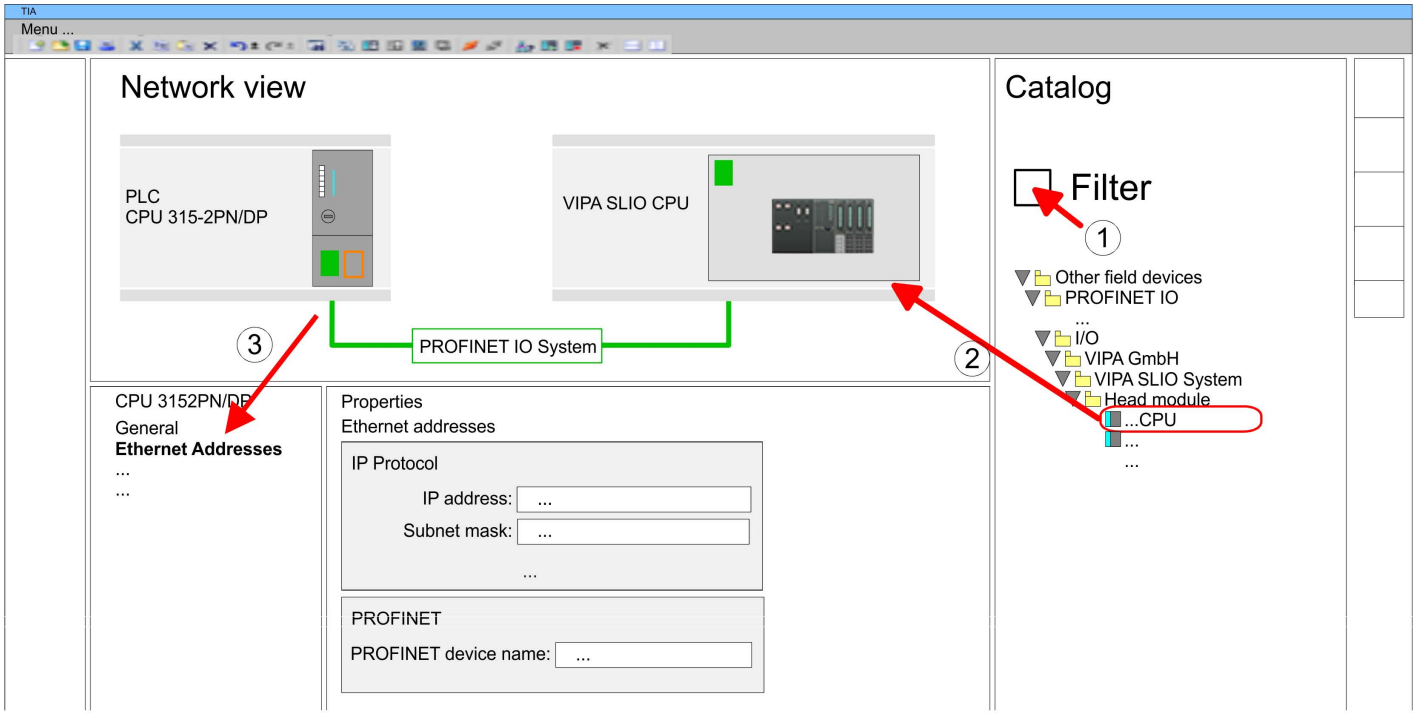
Module	...	Slot	...	Type	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	

Setting standard CPU parameters

Since the CPU from VIPA is configured as Siemens CPU, so the setting of the non- VIPA specific parameters takes place via the Siemens CPU. For parametrization click in the *Project area*respectively in the *Device overview* at the CPU part. Then the parameters of the CPU part are shown in the *Properties dialog*. Here you can make your parameter settings. ↳ *Chapter 4.7.1 'Parameter CPU' on page 64*

Connection SLIO CPU as PROFINET IO device

1. Switch in the *Project area* to *'Network view'*.
2. After installing the GSDML the IO device for the SLIO CPU may be found in the hardware catalog at *Other field devices > PROFINET > IO > VIPA GmbH > VIPA SLIO System*. Connect the slave system to the CPU by dragging&dropping it from the hardware catalog to the *Network view* and connecting it via PROFINET to the CPU.
3. Click in the *Network view* at the PROFINET part of the Siemens CPU and enter a valid IP address data in *'Properties'* at *'Ethernet address'* in the area *'IP protocol'*.
4. Enter at *'PROFINET'* a *'PROFINET device name'*. The device name must be unique at the Ethernet subnet.



5. Select in the *Network view* the IO device *'VIPA SLIO CPU...'* and switch to the *Device overview*.

   ⇨ In the *Device overview* of the PROFINET IO device *'VIPA SLIO CPU'* the CPU is already placed at slot 0. From slot 1 you can place your system SLIO modules.

Setting VIPA specific CPU parameters

For parametrization click at the CPU at slot 0 in the *Device overview* of the PROFINET IO device *'VIPA SLIO CPU'*. Then the parameters of the CPU part are shown in the *Properties dialog*. Here you can make your parameter settings. ↳ *Chapter 4.7.1 'Parameter CPU' on page 64*

11.3 TIA Portal - Hardware configuration - Ethernet PG/OP channel

Overview

The CPU has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU.

- The Ethernet PG/OP channel (X1/X5) is designed as switch. This enables PG/OP communication via the connections X1 and X5.
- The Ethernet PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc.
- At the first commissioning respectively after a factory reset the Ethernet PG/OP channel has no IP address.
- For online access to the CPU via the Ethernet PG/OP channel, valid IP address parameters have to be assigned to this. This is called "initialization".
- This can be done with the Siemens TIA Portal.

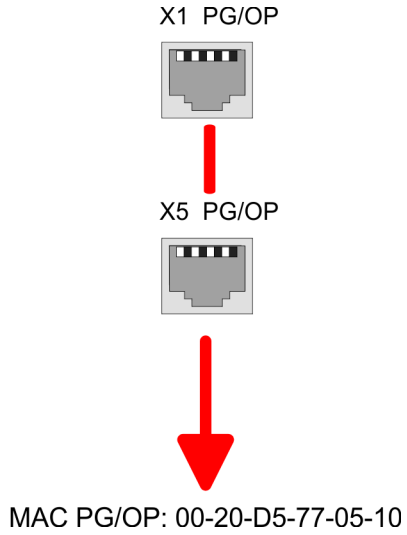
Assembly and commissioning

1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the one of the Ethernet jacks (X1, X5) of the Ethernet PG/OP channel to Ethernet.
4. Switch on the power supply.
   ⇨ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization" via Online functions

The initialization via the Online functions takes place with the following proceeding:

Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".

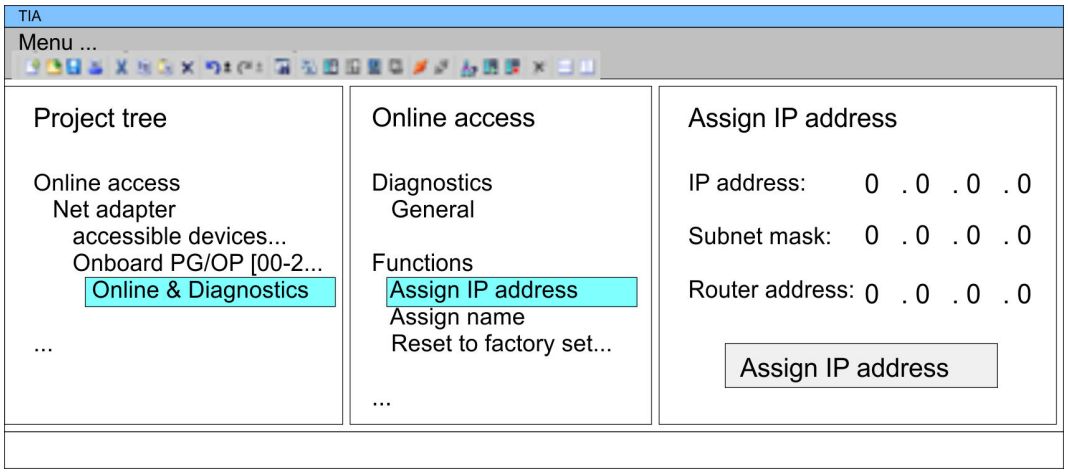


Assign IP address parameters

You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the Siemens TIA Portal with the following proceeding:

1. Start the Siemens TIA Portal.
2. Switch to the *'Project view'*.
3. Click in the *'Project tree'* at *'Online access'* and choose here by a doubleclick your network card, which is connected to the Ethernet PG/OP channel.

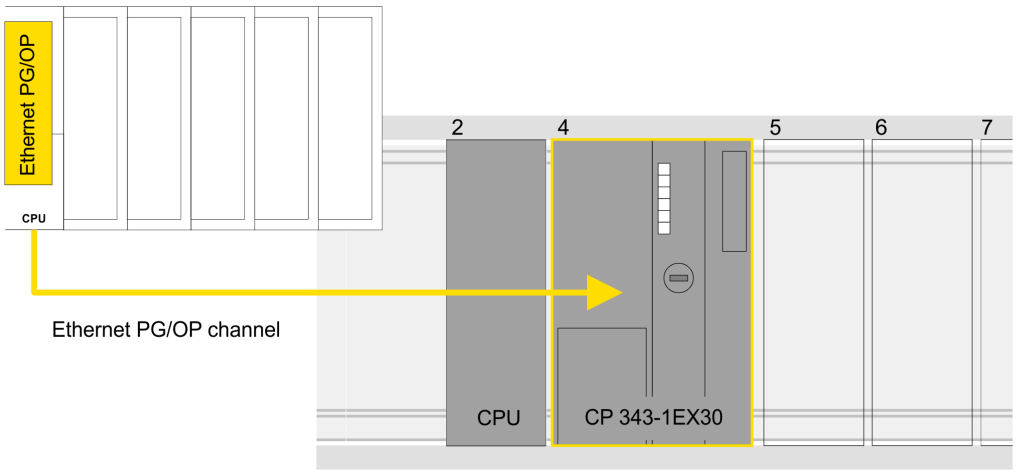
4. To get the stations and their MAC address, use the *'Accessible device'*. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".
5. Choose from the list the module with the known MAC address (Onboard PG/OP [MAC address]) and open with "Online & Diagnostics" the diagnostics dialog in the Project area.
6. Navigate to *Functions* > *Assign IP address*. Type in the IP configuration like IP address, subnet mask and gateway.
7. Confirm with [Assign IP configuration].
   ⇨ Directly after the assignment the Ethernet PG/OP channel is online reachable using the set IP address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.



Due to the system you may get a message that the IP address could not be assigned. This message can be ignored.

Take IP address parameters in project

1. Open your project.
2. If not already done, configure in the *'Device configuration'* a Siemens CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).
3. As Ethernet PG/OP channel place at slot 4 the Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).
4. Open the "Property" dialog by clicking on the CP 343-1EX30 and enter for the CP at "Properties" at "Ethernet address" the IP address data, which you have assigned before.
5. Transfer your project.



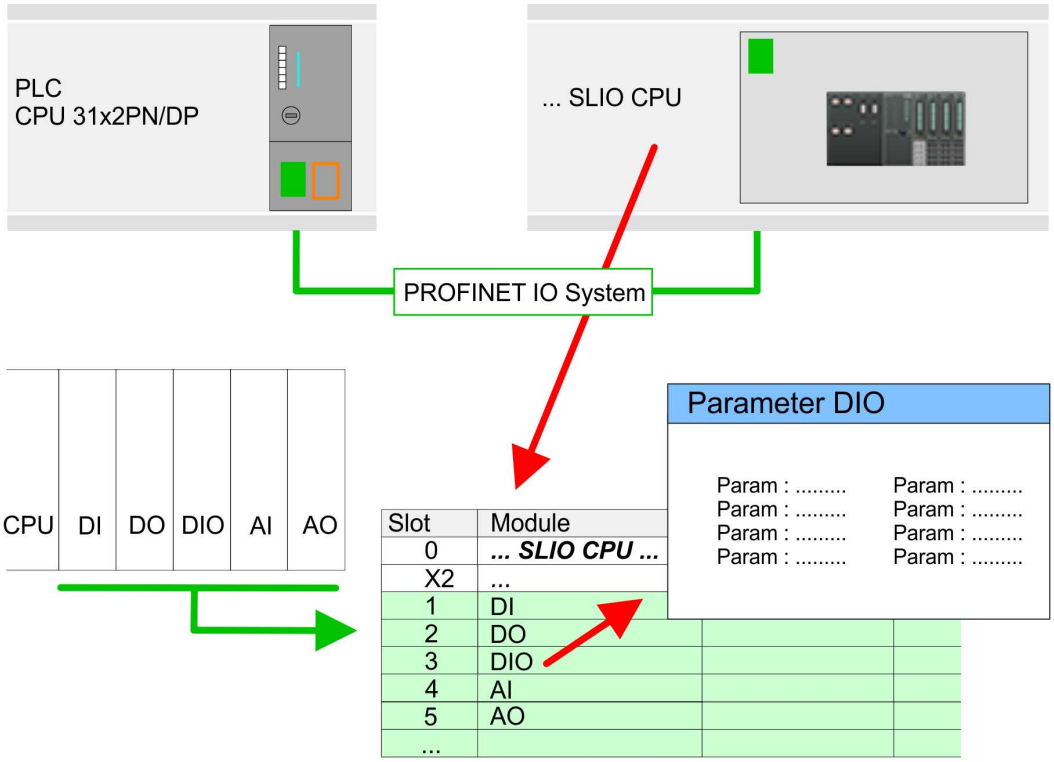
Device overview

Module	...	Slot	...	Type	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	
CP 343-1		4		CP 343-1	
...		...		...	

11.4 TIA Portal - Hardware configuration - I/O modules

Hardware configuration of the modules

Starting with slot 1 place in the *Device overview* of the PROFINET IO device *'VIPA SLIO CPU'* your System SLIO modules in the plugged sequence. For this drag from the hardware catalog the corresponding module to the corresponding position in the *Device overview*.



Parametrization

To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU. For parametrization click in the *Device overview* at the module you want to parametrize. Then the parameters of the module are shown in the *Properties* dialog. Here you can make your parameter settings.

11.5 TIA Portal - VIPA-Include library

Overview

- The VIPA specific blocks can be found in the "Service" area of www.vipa.com as library download file at *Downloads* > *VIPA LIB*.
- The library is available as packed zip file for the corresponding TIA Portal version.
- As soon as you want to use VIPA specific blocks you have to import them into your project.
  Execute the following steps:
  – Load an unzip the file ...TIA_V*xx*.zip (note TIA Portal version)
  – Open library and transfer blocks into the project

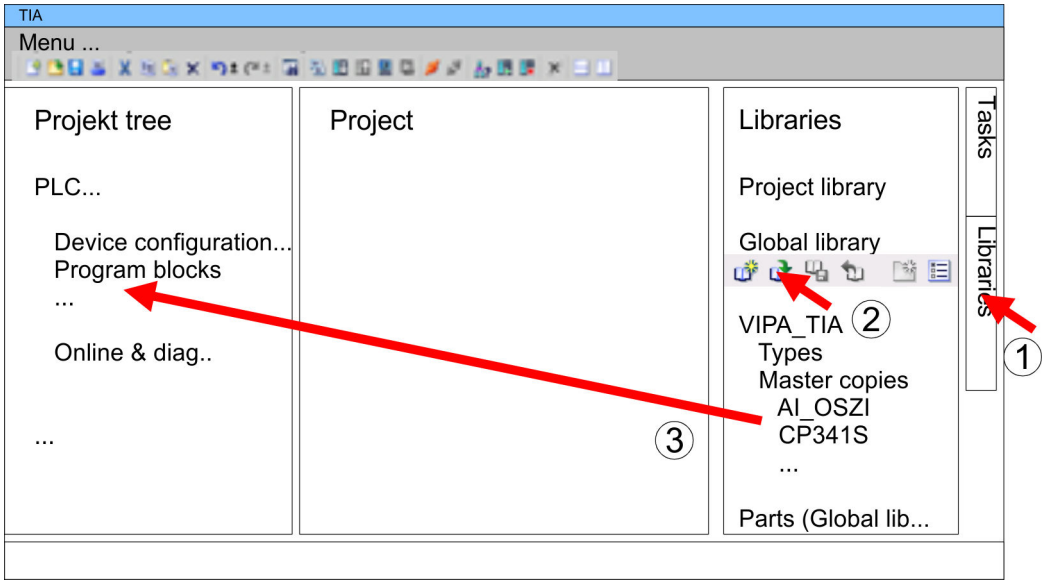
Unzip ...TIA_Vxx.zip

Start your un-zip application with a double click on the file TIA_V*xx*.zip and copy all the files and folders in a work directory for the Siemens TIA Portal.

Open library and transfer blocks into the project

1. Start the Siemens TIA Portal with your project.
2. Switch to the *Project view*.
3. Choose "Libraries" from the task cards on the right side.
4. Click at "Global libraries".
5. Click at "Open global libraries".

6. Navigate to your directory and load the file ...TIA.al*xx*.



7. Copy the necessary blocks from the library into the "Program blocks" of the *Project tree* of your project. Now you have access to the VIPA specific blocks via your user application.

11.6 TIA Portal - Project transfer

Overview

There are the following possibilities for project transfer into the CPU:

- Transfer via MPI
- Transfer via Ethernet
- Transfer via memory card

Transfer via MPI

Currently the VIPA programming cables for transfer via MPI are not supported. This is only possible with the programming cable from Siemens.

1. Establish a connection to the CPU via MPI with an appropriate programming cable. Information may be found in the corresponding documentation of the programming cable.
2. Switch-ON the power supply of your CPU and start the Siemens TIA Portal with your project.
3. Select in the *Project tree* your CPU and choose *'Context menu ➔ Download to device ➔ Hardware configuration'* to transfer the hardware configuration.
4. To transfer the PLC program choose *'Context menu ➔ Download to device ➔ Software'*. Due to the system you have to transfer hardware configuration and PLC program separately.

Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- X1 Ethernet PG/OP channel

Initialization

So that you may the according Ethernet interface, you have to assign IP address parameters by means of the "initialization". *Chapter 11.3 'TIA Portal - Hardware configuration - Ethernet PG/OP channel' on page 233*

Please consider to use the same IP address data in your project for the CP 343-1.

Transfer

1. For the transfer, connect, if not already done, the appropriate Ethernet jack to your Ethernet.
2. Open your project with the Siemens TIA Portal.
3. Click in the *Project tree* at *Online access* and choose here by a double-click your network card, which is connected to the Ethernet PG/OP interface.
4. Select in the *Project tree* your CPU and click at [Go online].
5. Set the access path by selecting "PN/IE" as type of interface, your network card and the according subnet. Then a net scan is established and the corresponding station is listed.
6. Establish with [Connect] a connection.
7. Click to *'Online ➔ Download to device'*.
   ⇨ The according block is compiled and by a request transferred to the target device. Provided that no new hardware configuration is transferred to the CPU, the entered Ethernet connection is permanently stored in the project as transfer channel.

Transfer via memory card

The memory card serves as external storage medium. There may be stored several projects and sub-directories on a memory card. Please regard that your current project is stored in the root directory and has one of the following file names:

- S7PROG.WLD
- AUTOLOAD.WLD

1. Create in the Siemens TIA Portal a wld file with *'Project ➔ Memory card file ➔ New'*.
   ⇨ The wld file is shown in the *Project tree* at "SIMATIC Card Reader" as "Memory card file".
2. Copy the blocks from the *Program blocks* to the wld file. Here the hardware configuration data are automatically copied to the wld file as "System data".

Transfer memory card → CPU

The transfer of the application program from the memory card into the CPU takes place depending on the file name after an overall reset or PowerON.

- *S7PROG.WLD* is read from the memory card after overall reset.
- *AUTOLOAD.WLD* is read from the memory card after PowerON.

The blinking of the SD LED of the CPU marks the active transfer. Please regard that your user memory serves for enough space for your user program, otherwise your user program is not completely loaded and the SF LED gets on.

Transfer CPU → Memory card

When a memory card has been installed, the write command stores the content of the RAM as S7PROG.WLD on the memory card. The write command can be found in the Siemens TIA Portal in the Task card "Online tools" in the command area at "Memory" as button [Copy RAM to ROM]. The SD LED blinks during the write access. When the LED expires, the write process is finished. If this project is to be loaded automatically from the memory card with PowerON, you have to rename this to on the memory card to *AUTOLOAD.WLD*.

Please note that in the Siemens TIA Portal with some CPU types the [Copy RAM to ROM] button is not available.

Checking the transfer operation

After accessing the memory card you can find a diagnostics entry in the CPU. To monitor the diagnostics entries, you select *Online & Diagnostics* in the Siemens TIA Portal. Here you can access the "Diagnostics buffer". ↳ *Chapter 4.18 'Diagnostic entries' on page 92*

Appendix

Content

A System specific event IDs

Event IDs ↳ *Chapter 4.18 'Diagnostic entries' on page 92*

Event ID	Description
0x115C	Vendor-specific interrupt (OB 57) at EtherCAT
	OB : OB number
	ZINFO1 : Logical address of the slave station that triggered the interrupt
	ZINFO2 : Interrupt type
	0: Reserved
	1: Diagnostic interrupt (incoming)
	2: Process interrupt
	3: Pull interrupt
	4: Plug interrupt
	5: Status interrupt
	6: Update interrupt
	7: Redundancy interrupt
	8: Controlled by the supervisor
	9: Enabled
	10: Wrong sub module plugged
	11: Recurrence of the sub module
	12: Diagnostic interrupt (outgoing)
	13: Cross traffic connection message
	14: Neighbourhood change message
	15: Synchronisation message (bus)
	16: Synchronisation message (device)
	17: Network component message
	18: Clock synchronisation message (bus)
	31: Pull interrupt component
	32: Vendor-specific interrupt min
	33: Vendor-specific interrupt topology change
	127: Vendor-specific interrupt max
	ZINFO3 : CoE error code
0xE003	Error in access to periphery
	ZINFO1 : Transfer type
	ZINFO2 : Periphery address
	ZINFO3 : Slot
0xE004	Multiple configuration of a periphery address
	ZINFO1 : Periphery address
	ZINFO2 : Slot

Event ID	Description
0xE005	Internal error - Please contact the hotline!
	ZINFO1 : Not user relevant
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
0xE007	Configured input/output bytes do not fit in the periphery area
0xE008	Internal error - Please contact the hotline!
0xE009	Error on accessing the standard backplane bus
0xE010	Non-defined component recognised at the backplane bus
	ZINFO2 : Slot
	ZINFO3 : Type identifier
0xE011	Master project engineering at slave CPU not possible or wrong slave configuration
0xE012	Error at configuration
0xE013	Error at shift register access to standard bus digital modules
0xE014	Error in Check_Sys
0xE015	Error in access to master
	ZINFO2 : Slot of the master
	ZINFO2 : Page frame master
0xE016	Maximum block size exceeded in master transfer
	ZINFO1 : Periphery address
	ZINFO2 : Slot
0xE017	Error in access to integrated slave
0xE018	Error in mapping the master periphery
0xE019	Error on standard backplane bus system detection
0xE01A	Error at detection of the operating mode (8 / 9 bit)
0xE01B	Error - maximum number of plug-in components exceeded
0xE020	Error - interrupt information undefined
	ZINFO2 : Slot
	ZINFO3 : Not user relevant
	DatID : Interrupt type
0xE030	Error of the standard bus
0xE033	Internal error - Please contact the hotline!
0xE0B0	SPEED7 is not stoppable (e.g. undefined BCD value at timer)
	ZINFO1 : Not user relevant
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xE0C0	Not enough memory space in the working memory for code block (block too large)
0xE0CB	Error on SSL access

Event ID	Description
	ZINFO1 : Error
	4: SSL wrong
	5: Sub-SSL wrong
	6: Index wrong
	ZINFO2 : SZL-ID
	ZINFO3 : Index
0xE0CC	Communication error
	ZINFO1 : Error code
	1: Wrong priority
	2: Buffer overrun
	3: Telegram format error
	4: Wrong SSL request (SSL-ID invalid)
	5: Wrong SSL request (SSL-Sub-ID invalid)
	6: Wrong SSL request (SSL-Index invalid)
	7: Wrong value
	8: Wrong return value
	9: Wrong SAP
	10: Wrong connection type
	11: Wrong sequence number
	12: Faulty block number in the telegram
	13: Faulty block type in the telegram
	14: Inactive function
	15: Wrong size in the telegram
	20: Error in writing on MMC
	90: Faulty buffer size
	98: Unknown error
	99: Internal error
0xE0CD	Error at DP-V1 job management
	ZINFO1 : Not user relevant
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xE0CE	Error: Time out when sending i-slave diagnostics
0xE100	Memory card access error
0xE101	Memory card error file system
0xE102	Memory card error FAT
0xE104	Memory card error at saving
	ZINFO3 : Not user relevant

Event ID	Description
0xE200	Memory card writing finished (Copy Ram2Rom)
	OB : Not user relevant
	PK : Not user relevant
0xE210	Memory card reading finished (reload after memory reset)
	OB : Not user relevant
	PK : Not user relevant
	ZINFO1 : Not user relevant
0xE21D	Memory card reading: Error on reload (after memory reset), error in the block header
	ZINFO1 : Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO2 : Block number
	ZINFO3 : Block length
0xE21E	Memory card reading: error in recharging (after memory reset), “Protect.wld” file too large
	OB : Not user relevant
0xE21F	Memory card reading: Error at reload (after memory reset), checksum error when reading
	OB : Not user relevant
	PK : Not user relevant
	ZINFO1 : Not user relevant
	ZINFO2 : Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB

Event ID	Description
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO3 : Block number
0xE300	Internal flash writing completed (copy Ram2Rom)
0xE310	Internal flash reading completed (recharging after battery failure)
0xE400	FSC card was plugged
	OB : FSC activated from this slot (PK)
	OB : The inserted FSC is the activated FSC
	OB : The inserted FSC is compatible with the CPU
	PK : FSC source
	0: CPU
	1: Card
	ZINFO1 : FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00

Event ID	Description
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2 : FSC serial number (high word)
	ZINFO3 : FSC serial number (low word)
0xE401	FSC card was removed
	OB : Action after the end of the trial time
	0: No action
	1: CPU STOP
	2: CPU STOP and FSC deactivated
	3: Factory reset
	255: FSC was not activated
	PK : FSC source
	0: CPU
	1: Card
	ZINFO1 : FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30

Event ID	Description
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2 : FSC serial number (high word)
	ZINFO3 : FSC serial number (low word)
	DatID : FeatureSet Trialtime in minutes
0xE402	A configured functionality is not activated
	ZINFO1 : Required FSC: PROFIBUS
	ZINFO1 : Required FSC: MOTION

Event ID	Description
	ZINFO2 : Number of released axes
	ZINFO3 : Number of configured axes
0xE403	FSC can not be activated in this CPU
	OB : FCS error code
	PK : FSC source
	0: CPU
	1: Card
	ZINFO1 : FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00

Event ID	Description
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2 : FSC serial number (high word)
	ZINFO3 : FSC serial number (low word)
0xE404	Feature set deleted due to CRC error
0xE405	The trial time of a feature set / MMC has expired
	OB : Action after the end of the trial time
	0: No action
	1: CPU STOP
	2: CPU STOP and FSC deactivated
	3: Factory reset
	255: FSC was not activated
	PK : FSC source
	0: CPU
	1: Card
	ZINFO1 : FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00

Event ID	Description
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2 : FSC serial number (high word)
	ZINFO3 : FSC serial number (low word)
	DatID : FeatureSet Trialtime in minutes
0xE406	The inserted feature set is corrupt
	PK : FSC source
	0: CPU
	1: Card
0xE410	A CPU feature set was activated
	PK : FSC source
	0: CPU
	1: Card
	ZINFO1 : FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30

Event ID	Description
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2 : FSC serial number (high word)
	ZINFO3 : FSC serial number (low word)
0xE500	Memory management: Deleted block without corresponding entry in BstList
	ZINFO2 : Block type

Event ID	Description
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO3 : Block number
0xE501	Parser error
	ZINFO1 : Error code
	1: Parser error: SDB structure
	2: Parser error: SDB is not a valid SDB type.
	ZINFO2 : SDB type
	ZINFO3 : SDB number
0xE502	Invalid block type in protect.wld
	ZINFO2 : Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO3 : Block number

Event ID	Description
0xE503	Inconsistency of code sizes and block sizes in the working memory
	ZINFO1 : Code size
	ZINFO2 : Block size (high word)
	ZINFO3 : Block size (low word)
0xE504	Additional information for CRC error in the working memory
	ZINFO2 : Block address (high word)
	ZINFO3 : Block address (low word)
0xE505	Internal error - Please contact the hotline!
	ZINFO1 : Cause for MemDump
	0: Unknown
	1: Manual request
	2: Invalid OP value
	3: CRC code error
	4: Processor exception
	5: Processor exception with dump after reboot
	6: Block-CRC error
0xE604	Multiple configuration of a periphery address for Ethernet PG/OP channel
	ZINFO1 : Periphery address
	ZINFO3 : 0: periphery address is input, 1: periphery address is output
0xE605	Too many productive connections configured
	ZINFO1 : Interface slot
	ZINFO2 : Number of configured connections
	ZINFO3 : Number of admissible connections
0xE610	On-board PROFIBUS/MPI: bus error removed
	PK : Not user relevant
	ZINFO1 : Interface
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xE701	Internal error - Please contact the hotline!
	ZINFO1 : Not user relevant
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xE703	Internal error - Please contact the hotline!
	PK : Not user relevant
	ZINFO1 : Master system ID
	ZINFO2 : Slave address

Event ID	Description
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xE705	Too many PROFIBUS slaves configured
	ZINFO1 : Diagnostic address of the PROFIBUS master
	ZINFO2 : Number of configured slaves
	ZINFO3 : Number of admissible slaves
0xE710	On-board PROFIBUS/MPI: bus error occurred
	PK : Not user relevant
	ZINFO1 : Interface
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xE720	Internal error - Please contact the hotline!
	ZINFO1 : Slave no
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Master system ID
0xE721	Internal error - Please contact the hotline!
	ZINFO1 : Not user relevant
	ZINFO2 : Master system ID
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xE722	Internal error - Please contact the hotline!
	ZINFO1 : Channel-Event
	0: Channel offline
	1: Bus error
	2: Internal error
	ZINFO2 : Master system ID
	DatID : Not user relevant
0xE723	Internal error - Please contact the hotline!
	ZINFO1 : Error code
	1: Parameter error
	2: Configuration error
	ZINFO2 : Master system ID
	DatID : Not user relevant
0xE780	Internal error - Please contact the hotline!
0xE801	CMD - auto command: CMD_START recognized and successfully executed
0xE802	CMD - auto command: CMD_End recognized and successfully executed

Event ID	Description
0xE803	CMD - auto command: WAIT1SECOND recognized and successfully executed
0xE804	CMD - auto command: WEBPAGE recognized and successfully executed
0xE805	CMD - auto command: LOAD_PROJECT recognized and successfully executed
0xE806	CMD - auto command: SAVE_PROJECT recognized and successfully executed
	ZINFO3 : Status
	0: Error
	1: OK
	32768: Wrong password
0xE807	CMD - auto command: FACTORY_RESET recognized and successfully executed
0xE808	Internal error - Please contact the hotline!
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
0xE809	Internal error - Please contact the hotline!
	ZINFO3 : Not user relevant
0xE80A	Internal error - Please contact the hotline!
	ZINFO3 : Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80B	CMD - auto command: DIAGBUF recognized and successfully executed
	ZINFO3 : Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80C	Internal error - Please contact the hotline!
	ZINFO3 : Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80D	Internal error - Please contact the hotline!
0xE80E	CMD - auto command: SET_NETWORK recognized and successfully executed
0xE80F	Internal error - Please contact the hotline!
	ZINFO3 : Status
	0: OK
	65153: File create error

Event ID	Description
	65185: File writing error
	65186: Odd address for reading
0xE810	Internal error - Please contact the hotline!
0xE811	Internal error - Please contact the hotline!
0xE812	Internal error - Please contact the hotline!
0xE813	Internal error - Please contact the hotline!
0xE814	CMD - auto command: SET_MPI_ADDRESS identified
0xE816	CMD - auto command: SAVE_PROJECT recognized but not executed, because the CPU memory is empty
0xE817	Internal error - Please contact the hotline!
	ZINFO3 : Not user relevant
0xE820	Internal message
0xE821	Internal message
0xE822	Internal message
0xE823	Internal message
0xE824	Internal message
0xE825	Internal message
0xE826	Internal message
0xE827	Internal message
0xE828	Internal message
0xE829	Internal message
0xE82A	CMD - auto command: CPUTYPE_318 recognized and successfully executed
	ZINFO3 : Error code:
0xE82B	CMD - auto command: CPUTYPE_ORIGINAL recognized and successfully executed
	ZINFO3 : Error code:
0xE8FB	CMD - auto command: error: Initialization of the Ethernet PG/OP channel by means of SET_NETWORK is faulty
0xE8FC	CMD - auto command: error: Some IP parameters missing in SET_NETWORK
0xE8FE	CMD - auto command: error: CMD_START not found
0xE8FF	CMD - auto command: Error while reading CMD file (memory card error)
0xE901	Checksum error
	ZINFO1 : Not user relevant
	ZINFO2 : Not user relevant
	DatID : Not user relevant
0xE902	Internal error - Please contact the hotline!
	ZINFO1 : Not user relevant
	ZINFO2 : Not user relevant
	DatID : Not user relevant
0xE904	PG/OP: Multiple parametrization of a peripheral address

Event ID	Description
	ZINFO1 : Peripheral address
	ZINFO2 : Slot
	ZINFO3 : Data width
	DatID : 0x54 Peripheral address is input address
	DatID : 0x55 Peripheral address is output address
0xE910	PG/OP: Input peripheral address out of peripheral area
	ZINFO1 : Peripheral address
	ZINFO2 : Slot
	ZINFO3 : Data width
0xE911	PG/OP: Output peripheral address out of peripheral area
	ZINFO1 : Peripheral address
	ZINFO2 : Slot
	ZINFO3 : Data width
0xEA00	Internal error - Please contact the hotline!
	PK : Not relevant to user
	DatID : Not user relevant
0xEA01	Internal error - Please contact the hotline!
	PK : Not user relevant
	ZINFO1 : Slot
	DatID : Not user relevant
0xEA02	SBUS: Internal error (internal plugged sub module not recognized)
	PK : Not user relevant
	ZINFO1 : Slot
	ZINFO2 : Type identifier target
	ZINFO3 : Type identifier
	DatID : Not user relevant
0xEA03	SBUS: Communication error between CPU and IO controller
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN

Event ID	Description
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK : Not user relevant
	ZINFO1 : Slot
	ZINFO2 : Status
	0: Ok
	1: Error
	2: Empty
	3: Busy
	4: Time out
	5: Internal blocking
	6: Too many telegrams
	7: Not Connected
	8: Unknown
	DatID : Not user relevant
0xEA04	SBUS: multiple configuration of a periphery address
	ZINFO1 : Periphery address
	ZINFO2 : Slot
	ZINFO3 : Data width
0xEA05	Internal error - Please contact the hotline!
0xEA07	Internal error - Please contact the hotline!
0xEA08	SBUS: configured input data width not the same as the connected input data width
	ZINFO1 : Configured input data width
	ZINFO2 : Slot
	ZINFO3 : Input data width of the connected component
0xEA09	SBUS: configured output data width not the same as the connected output data width
	ZINFO1 : Configured output data width
	ZINFO2 : Slot
	ZINFO3 : Output data width of the plugged component
0xEA10	SBUS: Input periphery address outside the periphery area
	ZINFO1 : Periphery address

Event ID	Description
	ZINFO2 : Slot
	ZINFO3 : Data width
0xEA11	SBUS: Output periphery address outside the periphery area
	ZINFO1 : Periphery address
	ZINFO2 : Slot
	ZINFO3 : Data width
0xEA12	SBUS: error in writing dataset
	ZINFO1 : Slot
	ZINFO2 : Dataset number
	ZINFO3 : Dataset length
0xEA14	SBUS: multiple configuration of a periphery address (diagnostic address)
	ZINFO1 : Periphery address
	ZINFO2 : Slot
	ZINFO3 : Data width
0xEA15	Internal error - Please contact the hotline!
	ZINFO2 : Slot of the master
0xEA18	SBUS: error in mapping the master periphery
	ZINFO2 : Slot of the master
0xEA19	Internal error - Please contact the hotline!
	PK : Not user relevant
	ZINFO2 : HW slot
	ZINFO3 : Interface type
	DatID : Not user relevant
0xEA1A	SBUS: error in access to SBUS FPGA address table
	PK : Not user relevant
	ZINFO2 : HW slot
	ZINFO3 : Table
	0: Read
	1: Writing
	DatID : Not user relevant
0xEA20	Error - RS485 interface is not pre-set to PROFIBUS DP master bus a PROFIBUS DP master is configured.
0xEA21	Error - configuration RS485 interface X2/X3: PROFIBUS DP master is configured but missing.
	ZINFO2 : Interface X is configured incorrectly
0xEA22	Error - RS485 interface X2 - value is outside the limits
	ZINFO2 : Configuration for X2
0xEA23	Error - RS485 interface X3 - value is outside the limits
	ZINFO2 : Configuration for X3
0xEA24	Error - configuration RS485 interface X2/X3: Interface/protocol missing, default settings are used.

Event ID	Description
	ZINFO2 : Configuration for X2
	ZINFO3 : Configuration for X3
0xEA30	Internal error - Please contact the hotline!
	ZINFO1 : Status
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
0xEA40	Internal error - Please contact the hotline!
	OB : Slot of the CP
	PK : File number
	ZINFO1 : Version of the CP
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Line
0xEA41	Internal error - Please contact the hotline!
	OB : Slot of the CP
	PK : File number
	ZINFO1 : Version of the CP
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Line
0xEA50	PROFINET-IO controller: Error in the configuration
	OB : Not user relevant
	PK : Not user relevant
	ZINFO1 : Rack/slot of the controller
	ZINFO2 : Device number
	ZINFO3 : Slot at the device
	DatID : Not user relevant
0xEA51	PROFINET-IO controller: There is no PROFINET IO controller at the configured slot
	PK : Not user relevant
	ZINFO1 : Rack/slot of the controller
	ZINFO2 : Recognized type identifier at the configured slot
	DatID : Not user relevant
0xEA52	PROFINET-IO controller: Too many configured PROFINET-IO controllers
	PK : Not user relevant
	ZINFO1 : Number of configured controllers
	ZINFO2 : Slot of the excessively configured controller
	DatID : Not user relevant
0xEA53	PROFINET-IO controller: PROFINET configuration: There are too many configured PROFINET IO devices

Event ID	Description
	ZINFO1 : Number of configured devices
	ZINFO2 : Slot
	ZINFO3 : Maximum possible number of devices
0xEA54	PROFINET: Consistent block size exceeded for IDevice
	PK : Not user relevant
	ZINFO1 : Logical address of the IO system
	ZINFO2 : Rack/slot of the controller
	ZINFO3 : Base address of the block which is too large
	DatID : Not user relevant
0xEA55	PROFINET error in configuration: too many slots configured
	ZINFO1 : Rack/slot of the controller
	ZINFO2 : Device number
	ZINFO3 : Number of configured slots
0xEA56	PROFINET error in configuration: too many subslots configured
	ZINFO1 : Rack/slot of the controller
	ZINFO2 : Device number
	ZINFO3 : Number of configured subslots
0xEA61	Internal error - Please contact the hotline!
	OB : File number
	PK : Slot of the controller
	ZINFO1 : Firmware major version
	ZINFO2 : Firmware minor version
	DatID : Line
0xEA62	Internal error - Please contact the hotline!
	OB : File number.
	PK : Slot of the controller
	ZINFO1 : Firmware major version
	ZINFO2 : Firmware minor version
	DatID : Line
0xEA63	Internal error - Please contact the hotline!
	OB : File number
	PK : Slot of the controller
	ZINFO1 : Firmware major version
	ZINFO2 : Firmware minor version
	DatID : Line
0xEA64	PROFINET IO controller/EtherCAT-CP: Error in configuration
	ZINFO1 : Assume IP address in another way. Is not supported for the IP address of the controller.
	ZINFO1 : CPU is configured as an I device

Event ID	Description
	ZINFO1 : Invalid device number
	ZINFO1 : Refresh interval too small
	ZINFO1 : Too many devices
	ZINFO1 : Too many devices per second
	ZINFO1 : Too many input bytes per millisecond
	ZINFO1 : Too many output bytes per millisecond
	ZINFO1 : Too many input bytes per device
	ZINFO1 : Too many output bytes per device
	ZINFO1 : Too many productive connections
	ZINFO1 : Too many input bytes in the process image
	ZINFO1 : Too many output bytes in the process image
	ZINFO1 : Configuration not available
	ZINFO1 : Configuration invalid
	ZINFO1 : Refresh interval too large
	ZINFO2 : Incompatible configuration (SDB version not supported)
	ZINFO2 : EtherCAT: EoE configured but not supported
	ZINFO2 : DC parameter invalid
	ZINFO2 : I device configuration invalid (slot gap)
	ZINFO2 : MRP configuration invalid (client)
0xEA65	Internal error - Please contact the hotline!
	PK : Platform
	0: none
	8: CP
	9: Ethernet CP
	10: PROFINET CP
	12: EtherCAT CP
	16: CPU
	ZINFO1 : ServiceID in which the error occurred
	ZINFO2 : Command in which the error occurred
	1: Request
	2: Connect
	3: Error
0xEA66	PROFINET error in the communication stack
	OB : StackError.Service
	PK : Rack/slot
	ZINFO1 : StackError.Error.Code
	ZINFO2 : StackError.Error.Detail
	ZINFO3 : StackError.Error.AdditionalDetail

Event ID	Description
	ZINFO3 : StackError.Error.AreaCode
	DatID : StackError.DeviceRef
0xEA67	PROFINET-IO controller: Error reading dataset
	OB : Rack/slot of the controller
	PK : Error type
	0: Dataset error local
	1: Dataset error stack
	2: Dataset error station
	ZINFO1 : Dataset number
	ZINFO2 : Dataset handle (caller)
	ZINFO3 : Internal error code from PN stack
	DatID : Device
0xEA68	PROFINET-IO controller: Error writing dataset
	OB : Rack/slot of the controller
	PK : Error type
	0: Dataset error local
	1: Dataset error stack
	2: Dataset error station
	ZINFO1 : Dataset number
	ZINFO2 : Dataset handle (caller)
	ZINFO3 : Internal error code from PN stack
	DatID : Device
0xEA69	Internal error - Please contact the hotline!
	ZINFO1 : Minimum version for the FPGA
	ZINFO2 : Loaded FPGA version
0xEA6A	PROFINET-IO controller: Service error in the communication stack
	OB : Service ID
	PK : Rack/slot
	ZINFO1 : ServiceError.Code
	ZINFO2 : ServiceError.Detail
	ZINFO3 : ServiceError.AdditionalDetail
	ZINFO3 : ServiceError.AreaCode
0xEA6B	PROFINET-IO controller: Incorrect Vendor-ID
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)

Event ID	Description
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK : Rack/slot
	ZINFO1 : Device ID
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xEA6C	PROFINET-IO controller: Incorrect Device-ID
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised

Event ID	Description
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK : Rack/slot
	ZINFO1 : Device ID
0xEA6D	PROFINET-IO controller: No empty name
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK : Rack/slot
	ZINFO1 : Device ID
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xEA6E	PROFINET-IO controller: Wait for RPC response
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)

Event ID	Description
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK : Rack/slot
	ZINFO1 : Device ID
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xEA6F	PROFINET-IO controller: PROFINET module deviation
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP

Event ID	Description
	254: Watchdog
	255: Not set
	PK : Rack/slot
	ZINFO1 : Device ID
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xEA70	PROFINET stack configuration error
	OB : UnsupportedApiError.api
	PK : Rack/slot
	ZINFO1 : UnsupportedApiError.slot
	ZINFO2 : UnsupportedApiError.subslot
	DatID : UnsupportedApiError.deviceID
0xEA71	Internal error - Please contact the hotline!
	PK : Rack/slot
	ZINFO1 : functionIndex
	ZINFO2 : Not user relevant
0xEA72	PROFINET IO status message
	OB : Connection number
	PK : Slot of the controller
	ZINFO1 : Error cause
	129: PNIO
	207: RTA error
	218: AlarmAck
	219: IODConnectRes
	220: IODReleaseRes
	221: IOD/IOXControlRes
	222: IODReadRes
	223: IODWriteRes
	ZINFO2 : ErrorDecode
	128: PNIORW: Service Read Write
	129: PNIO: Other Service or internal e.g. RPC errors
	130: Vendor specific
	ZINFO3 : Error code (PN spec. V2.722 chapter 5.2.6)
	DatID : Device ID
0xEA81	Internal error - Please contact the hotline!
	OB : Not user relevant
	PK : Not user relevant

Event ID	Description
	ZINFO1 : Filenamehash[0-3 0:
	ZINFO2 : Filenamehash[4-7]
	ZINFO3 : Line
	DatID : SvnRevision
0xEA82	Internal error - Please contact the hotline!
	OB : Not user relevant
	PK : Not user relevant
	ZINFO1 : Filenamehash[0-3 0:
	ZINFO2 : Filenamehash[4-7]
	ZINFO3 : Line
	DatID : SvnRevision
0xEA83	Internal error - Please contact the hotline!
	OB : Not user relevant
	PK : Not user relevant
	ZINFO1 : Filenamehash[0-3 0:
	ZINFO2 : Filenamehash[4-7]
	ZINFO3 : Line
	DatID : SvnRevision
0xEA91	Internal error - Please contact the hotline!
	OB : Current OB number
	PK : Core status
	0: INIT
	1: STOP
	2: READY
	3: PAUSE
	4: RUN
	ZINFO1 : Filenamehash[0-3 0:
	ZINFO2 : Filenamehash[4-7]
	ZINFO3 : Line
	DatID : Current job number
0xEA92	Internal error - Please contact the hotline!
	OB : Current OB number
	PK : Core status
	0: INIT
	1: STOP
	2: READY
	3: PAUSE
	4: RUN

Event ID	Description
	ZINFO1 : Filenamehash[0-3 0:
	ZINFO2 : Filenamehash[4-7]
	ZINFO3 : Line
	DatID : Current job number
0xEA93	Internal error - Please contact the hotline!
	OB : Current OB number
	PK : Core status
	0: INIT
	1: STOP
	2: READY
	3: PAUSE
	4: RUN
	ZINFO1 : Filenamehash[0-3 0:
	ZINFO2 : Filenamehash[4-7]
	ZINFO3 : Line
	DatID : Current job number
0xEA97	Internal error - Please contact the hotline!
	ZINFO3 : Slot
0xEA98	Time-out when waiting for the reboot of a SBUS module (server)
	PK : Not user relevant
	ZINFO3 : Slot
	DatID : Not user relevant
0xEA99	Error in file reading via SBUS
	PK : Not user relevant
	ZINFO1 : File version on MMC/SD (if not 0)
	ZINFO2 : File version of the SBUS module (if not 0)
	ZINFO3 : Slot
	DatID : Not user relevant
0xEAA0	Internal error - Please contact the hotline!
	OB : Current operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)

Event ID	Description
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1 : Diagnostic address of the master
	ZINFO2 : Not user relevant
	ZINFO3 : Number of errors which occurred
0xEAB0	Invalid link mode
	OB : Current operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1 : Diagnostic address of the master
	ZINFO2 : Current connection mode
	1: 10Mbit half-duplex

Event ID	Description
	2: 10Mbit full-duplex
	3: 100Mbit half-duplex
	4: 100Mbit full-duplex
	5: Connection mode undefined
	6: Auto Negotiation
0xEAC0	Internal error - Please contact the hotline!
	ZINFO1 : Error code
	2: Internal error
	3: Internal error
	4: Internal error
	5: Internal error
	6: Internal error
	7: Internal error
	8: Internal error
	8: Internal error
0xEAD0	SyncUnit configuration error
0xEB02	System SLIO ERROR. preset configuration does not match actual configuration
	ZINFO1 : Bit mask slots 1-16
	ZINFO2 : Bit mask slots 17-32
	ZINFO3 : Bit mask slots 33-48
	DatID : Bit mask slots 49-64
0xEB03	System SLIO error: IO mapping
	PK : Not user relevant
	ZINFO1 : Error type
	1: SDB parser error
	2: Configured address already used
	3: Mapping error
	ZINFO2 : Slot (0=cannot be determined)
	DatID : Not user relevant
0xEB05	System SLIO error: Bus structure for Isochron process image not suitable
	PK : Not user relevant
	ZINFO2 : Slot (0=cannot be determined)
	DatID : Not user relevant
0xEB10	System SLIO error: bus error
	PK : Not user relevant
	ZINFO1 : Error type
	96: Bus enumeration error
	128: General error

Event ID	Description
	129: Queue execution error
	130: Error interrupt
	ZINFO2 : Error on bus enumeration error (Zinfo1)
	DatID : Not user relevant
0xEB11	SYSTEM SLIO error: error during bus initialization
	PK : Not user relevant
	DatID : Not user relevant
0xEB20	System SLIO error: interrupt information undefined
0xEB21	System SLIO error: accessing configuration data
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xEC03	EtherCAT: Configuration error
	PK : Not user relevant
	ZINFO1 : Error code
	1: Number of slave stations is not supported
	2: Master system ID invalid
	3: Slot invalid
	4: Master configuration invalid
	5: Master type invalid
	6: Slave diagnostic address invalid
	7: Slave address invalid
	8: Slave module IO configuration invalid.
	9: Logical address already in use.
	10: Internal error
	11: IO mapping error
	12: Error
	13: Error in initialising the EtherCAT stack (is entered by the CP)
	ZINFO2 : Error code higher 2 bytes
	ZINFO3 : Error code lower 2 bytes
	DatID : Not user relevant
0xEC04	EtherCAT: Multiple configuration of a periphery address
	PK : Not user relevant
	ZINFO1 : Periphery address
	ZINFO2 : Slot
	DatID : Not user relevant
0xEC05	EtherCAT: Check the set DC mode of the YASKAWA Sigma 5/7 drive
	OB : Operating mode

Event ID	Description
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK : Not user relevant
	ZINFO1 : Station address of the EtherCAT device
	ZINFO2 : Error code
	1: WARNING: For the drive the DC Beckhoff mode is recommended (DC reference clock is not in Beckhoff Mode)
	2: NOTE: For the drive the DC Hilscher mode is recommended (DC reference clock is not in Beckhoff Mode)
	3: The station address could not be determined for checking (station address in Zinfo1 is accordingly 0)
	4: The slave information could not be determined for checking (station address in Zinfo1 is accordingly 0)
	5: The EtherCAT status of the drive could not be determined
	6: Error when sending the SDO request (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	7: Drive returns error in the SDO response (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	8: SDO time out, DC mode could not be determined (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xEC10	EtherCAT: Recurrence bus with all slaves
	ZINFO1 : Previous status
	0: Undefined/Unkown
	1: Init

Event ID	Description
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 : New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2 : Diagnostic address of the station
	ZINFO3 : Number of stations, which are not in the same state as the master
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xEC11	EtherCAT: Recurrence bus with missing slaves
	ZINFO1 : Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 : New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2 : Diagnostic address of the master
	ZINFO3 : Number of stations which are not in the same state as the master
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available

Event ID	Description
0xEC12	EtherCAT: Recurrence slave
	ZINFO1 : Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 : New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2 : Diagnostic address of the station
	ZINFO3 : AL status code
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xEC30	EtherCAT: Topology OK
	ZINFO2 : Diagnostic address of the master
0xEC40	Bus cycle time infringement resolved
	ZINFO2 : Logical address of the IO system
0xEC50	EtherCAT: Distributed clocks (DC) out of sync
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT

Event ID	Description
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO2 : Diagnostic address of the master
	ZINFO3 : DC state change
	0: DC master out of sync
	1: DC slave stations out of Sync
0xEC80	Bus error resolved
	ZINFO1 : Logical address of the IO system
	ZINFO3 : Station number
	ZINFO3 : IO system ID
	ZINFO3 : System ID DP/PN
0xED10	EtherCAT: Breakdown bus
	ZINFO1 : Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 : New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2 : Diagnostic address of the master
	ZINFO3 : Number of stations which are not in the same state as the master
	DatID : Station available
	DatID : Station not available
	DatID : Input address
	DatID : Output address

Event ID	Description
0xED12	EtherCAT: Breakdown slave
	ZINFO1 : New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 : Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2 : Diagnostic address of the station
	ZINFO3 : AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state

Event ID	Description
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED20	EtherCAT: Bus state change without calling OB86
	ZINFO1 : Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 : New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2 : Diagnostic address of the master
	ZINFO3 : Number of stations which are not in the same state as the master
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED21	EtherCAT: Incorrect bus status change

Event ID	Description
	ZINFO1 : Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 : New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2 : Diagnostic address of the master
	ZINFO3 : Error code
	4: Cancel (master state change)
	8: Busy
	11: Invalid parameters
	14: Invalid status
	16: Time out
	DatID : Station available
	DatID : Station not available
	DatID : Output address
	DatID : Input address
0xED22	EtherCAT: Slave status change that does not generate an OB86
	ZINFO1 : New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 : Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap

Event ID	Description
	4: SafeOp
	8: Op
	ZINFO2 : Diagnostic address of the station
	ZINFO3 : AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT
	DatID : Input address

Event ID	Description
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED23	EtherCAT: Time out while changing the master state to OP, after CPU has changed to RUN
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1 : Master status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2 : EtherCAT configuration present
	0: There is no EC configuration
	1: There is an EC configuration
	ZINFO3 : DC in sync
	0: not in sync
	1: in sync
0xED30	EtherCAT: Topology deviation

Event ID	Description
	ZINFO2 : Diagnostic address of the master
0xED31	EtherCAT: Overflow of the interrupt queue
	ZINFO2 : Diagnostic address of the master
0xED40	Bus cycle time infringement occurred
	ZINFO1 : Logical address of the IO system
0xED50	EtherCAT: Distributed clocks (DC) in sync
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO2 : Diagnostic address of the master
	ZINFO3 : DC state change
	0: Master
	1: Slave
0xED60	EtherCAT: Diagnostic buffer CP: Slave status change
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)

Event ID	Description
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1 : New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2 : Slave address
	ZINFO3 : AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration

Event ID	Description
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT
	DatID : Cause for slave status change
	0: Regular slave status change
	1: Slave failure
	2: Recurrence slave
	3: Slave is in an error state
	4: Slave has unexpectedly changed its status
0xED61	EtherCAT: Diagnostic buffer CP: CoE emergency
	OB : EtherCAT station address (high byte)
	PK : EtherCAT station address (low byte)
	ZINFO1 : Error register
	ZINFO1 : MEF-Byte1
	ZINFO2 : MEF-Byte2
	ZINFO2 : MEF-Byte3
	ZINFO3 : MEF-Byte4
	ZINFO3 : MEF-Byte5
	DatID : Error code
0xED62	EtherCAT: Diagnostic buffer CP: Error on SDO access
	OB : EtherCAT station address (high byte)
	PK : EtherCAT station address (low byte)
	ZINFO1 : Index
	ZINFO2 : SDO error code (high word)

Event ID	Description
	ZINFO3 : SDO error code (low word)
	DatID : Sub index
0xED63	EtherCAT: Diagnostic buffer CP: Error in the response to an INIT command
	OB : EtherCAT station address (high byte)
	PK : EtherCAT station address (low byte)
	ZINFO1 : Error type
	0: Not defined
	1: No response
	2: Validation error
	3: INIT command failed, requested station could not be reached
0xED70	EtherCAT: Diagnostic buffer CP: Twofold hot connect group recognised
	OB : Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1 : Diagnostic address of the master
	ZINFO2 : EtherCAT station address
0xED80	Bus error occurred (receive time-out)
	ZINFO1 : Logical address of the IO system
	ZINFO3 : IO system ID
	ZINFO3 : System ID DP/PN
	ZINFO3 : Station number

Event ID	Description
0xEE00	Additional information at UNDEF_OPCODE
	OB : Not user relevant
	ZINFO1 : Not user relevant
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xEE01	Internal error - Please contact the hotline!
	ZINFO3 : SFB number
0xEEEE	CPU was completely deleted, since after PowerON the start-up could not be finished
0xEF00	Internal error - Please contact the hotline!
	DatID : Not user relevant
0xEF01	Internal error - Please contact the hotline!
	ZINFO1 : Not user relevant
	ZINFO2 : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xEF11	Internal error - Please contact the hotline!
0xEF12	Internal error - Please contact the hotline!
0xEF13	Internal error - Please contact the hotline!
0xEFFE	Internal error - Please contact the hotline!
	PK : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant
0xEFFF	Internal error - Please contact the hotline!
	PK : Not user relevant
	ZINFO3 : Not user relevant
	DatID : Not user relevant

B Integrated blocks

More information about this may be found in the manual "SPEED7 Operation List" from VIPA.

OB	Name	Description
OB 1	CYCL_EXC	Program Cycle
OB 10	TOD_INT0	Time-of-day Interrupt
OB 20	DEL_INT0	Time delay interrupt
OB 21	DEL_INT1	Time delay interrupt
OB 28	CYC_INT_250us	Cyclic interrupt
OB 29	CYC_INT_500us	Cyclic interrupt
OB 32	CYC_INT2	Cyclic interrupt
OB 33	CYC_INT3	Cyclic interrupt
OB 34	CYC_INT4	Cyclic interrupt
OB 35	CYC_INT5	Cyclic interrupt
OB 40	HW_INT0	Hardware interrupt
OB 55	DP: STATUS ALARM	Status interrupt
OB 56	DP: UPDATE ALARM	Update interrupt
OB 57	DP: MANUFACTURE ALARM	Vendor specific interrupt
OB 60	CYCL_EXC_FOR_SYNC_1	Multi-computing interrupt
OB 61	SYNC_1	Synchronous cycle interrupt
OB 80	CYCL_FLT	Time error
OB 81	PS_FLT	Power supply error
OB 82	I/O_FLT1	Diagnostics interrupt
OB 83	I/O_FLT2	Insert / remove module
OB 85	OBNL_FLT	Priority class error
OB 86	RACK_FLT	Slave failure / restart
OB 100	COMPLETE RESTART	Start-up
OB 102	COLD RESTART	Start-up
OB 121	PROG_ERR	Programming error
OB 122	MOD_ERR	Periphery access error

SFB	Name	Description
SFB 0	CTU	Up-counter
SFB 1	CTD	Down-counter

SFB	Name	Description
SFB 2	CTUD	Up-down counter
SFB 3	TP	Create pulse
SFB 4	TON	On-delay
SFB 5	TOF	Create turn-off delay
SFB 7	TIMEMESS	Time measurement
SFB 12	BSEND	Sending data in blocks
SFB 13	BRCV	Receiving data in blocks:
SFB 14	GET	Remote CPU read
SFB 15	PUT	Remote CPU write
SFB 32	DRUM	Realize a step-by-step switch
SFB 47	COUNT	Control counter
SFB 48	FREQUENC	Frequency measurement
SFB 49	PULSE	Pulse width modulation
SFB 52	RDREC	Read record set
SFB 53	WRREC	Write record set
SFB 54	RALRM	Receiving an interrupt from a periphery module
SFB 238	EC_RWOD	Function is used internally

SFC	Name	Description
SFC 0	SET_CLK	Set system clock
SFC 1	READ_CLK	Read system clock
SFC 2	SET_RTM	Set run-time meter
SFC 3	CTRL_RTM	Control run-time meter
SFC 4	READ_RTM	Read run-time meter
SFC 5	GADR_LGC	Logical address of a channel
SFC 6	RD_SINFO	Read start information
SFC 7	DP_PRAL	Triggering a hardware interrupt on the DP master
SFC 12	D_ACT_DP	Activating and deactivating of DP slaves
SFC 13	DPNRM_DG	Read diagnostic data of a DP salve
SFC 14	DPRD_DAT	Read consistent data
SFC 15	DPWR_DAT	Write consistent data
SFC 17	ALARM_SQ	ALARM_SQ
SFC 18	ALARM_SQ	ALARM_S
SFC 19	ALARM_SC	Acknowledgement state last alarm
SFC 20	BLKMOV	Block move
SFC 21	FILL	Fill a field
SFC 22	CREAT_DB	Create a data block

SFC	Name	Description
SFC 23	DEL_DB	Deleting a data block
SFC 24	TEST_DB	Test data block
SFC 28	SET_TINT	Set time-of-day interrupt
SFC 29	CAN_TINT	Cancel time-of-day interrupt
SFC 30	ACT_TINT	Activate time-of-day interrupt
SFC 31	QRY_TINT	Query time-of-day interrupt
SFC 32	SRT_DINT	Start time-delay interrupt
SFC 33	CAN_DINT	Cancel time-delay interrupt
SFC 34	QRY_DINT	Query time-delay interrupt
SFC 36	MSK_FLT	Mask synchronous errors
SFC 37	MSK_FLT	Unmask synchronous errors
SFC 38	READ_ERR	Read error register
SFC 39	DIS_IRT	Disabling interrupts
SFC 40	EN_IRT	Enabling interrupts
SFC 41	DIS_AIRT	Delaying interrupts
SFC 42	EN_AIRT	Enabling delayed interrupts
SFC 43	RE_TRIGR	Re-trigger the watchdog
SFC 44	REPL_VAL	Replace value to ACCU1
SFC 46	STP	STOP the CPU
SFC 47	WAIT	Delay the application program
SFC 49	LGC_GADR	Read the slot address
SFC 51	RDSYSST	Read system status list SSL
SFC 52	WR_USMSG	Write user entry into diagnostic buffer
SFC 53	µS_TICK	Time measurement
SFC 54	RD_DPARM	Reading predefined parameters
SFC 55	WR_PARM	Write dynamic parameter
SFC 56	WR_DPARM	Write default parameter
SFC 57	PARM_MOD	Parametrize module
SFC 58	WR_REC	Write record set
SFC 59	RD_REC	Read record set
SFC 64	TIME_TCK	Read system time tick
SFC 65	X_SEND	Sending data
SFC 66	X_RCV	Receiving data
SFC 67	X_GET	Read data
SFC 68	X_PUT	Write data
SFC 69	X_ABORT	Disconnect

SFC	Name	Description
SFC 70	GEO_LOG	Determining the start address of a module
SFC 71	LOG_GEO	Determining the slot belonging to a logical address
SFC 81	UBLKMOV	Copy data area without gaps
SFC 101	HTL_RTM	Handling runtime meters
SFC 102	RD_DPARA	Reading predefined parameters
SFC 105	READ_SI	Reading dynamic system resources
SFC 106	DEL_SI	Releasing dynamic system resources
SFC 107	ALARM_DQ	ALARM_DQ
SFC 108	ALARM_DQ	ALARM_DQ

C SSL partial list

More information about this may be found in the manual "SPEED7 Operation List" from VIPA.

SSL-ID	SSL partial list
xy11h	Module identification
xy12h	CPU characteristics
xy13h	User memory areas
xy14h	System areas
xy15h	Block Types
xy19h	Status of all LEDs
xy1Ch	Identification of the component
xy22h	Interrupt status
xy32h	Communication status data
xy37h	Ethernet details of the module
xy3Ah	Status of the TCON Connections
xy74h	Status of the LEDs
xy91h	Status information CPU
xy92h	Stations status information (DPM)
xy94h	Stations status information (DPM, PROFINET-IO and EtherCAT)
xy96h	Module status information (PROFIBUS DP, PROFINET-IO, EtherCAT)
xyA0h	Diagnostic buffer of the CPU
xyB3h	Module diagnostic information (record set 1) via logical address
xyB4h	Diagnostic data of a DP slave
xyE0h	Information EtherCAT master/slave
xyE1h	EtherCAT bus system
xyFAh	Statistics information to OBs
xyFCh	Status of the VSC features from the System SLIO CPU