

VIPA System SLIO

CPU | 013-CCF0R00 | Manual

HB300 | CPU | 013-CCF0R00 | en | 19-30

SPEED7 CPU 013C



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

Table of contents

| | | |
|----------|--|-----------|
| 1 | General | 9 |
| | 1.1 Copyright © VIPA GmbH | 9 |
| | 1.2 About this manual..... | 10 |
| | 1.3 Safety information..... | 11 |
| 2 | Basics and mounting | 12 |
| | 2.1 Safety information for users..... | 12 |
| | 2.2 System conception..... | 13 |
| | 2.2.1 Overview..... | 13 |
| | 2.2.2 Components..... | 14 |
| | 2.2.3 Accessories..... | 16 |
| | 2.2.4 Hardware revision..... | 17 |
| | 2.3 Dimensions..... | 17 |
| | 2.4 Mounting..... | 20 |
| | 2.4.1 Mounting CPU 01xC..... | 20 |
| | 2.5 Wiring..... | 22 |
| | 2.5.1 Wiring CPU 01xC..... | 22 |
| | 2.5.2 Wiring periphery modules..... | 25 |
| | 2.5.3 Wiring power modules..... | 27 |
| | 2.6 Demounting..... | 32 |
| | 2.6.1 Demounting CPU 01xC..... | 32 |
| | 2.6.2 Demounting periphery modules..... | 33 |
| | 2.7 Trouble shooting - LEDs..... | 37 |
| | 2.8 Installation guidelines..... | 38 |
| | 2.9 General data..... | 40 |
| 3 | Hardware description | 42 |
| | 3.1 Properties..... | 42 |
| | 3.2 Structure..... | 43 |
| | 3.2.1 Compact CPU..... | 43 |
| | 3.2.2 Interfaces..... | 43 |
| | 3.2.3 Memory management..... | 47 |
| | 3.2.4 Slot for storage media..... | 47 |
| | 3.2.5 Buffering mechanisms..... | 47 |
| | 3.2.6 Operating mode switch..... | 48 |
| | 3.2.7 LEDs..... | 49 |
| | 3.3 Technical data..... | 53 |
| 4 | Deployment CPU 013-CCF0R00 | 66 |
| | 4.1 Please note!..... | 66 |
| | 4.2 Assembly..... | 66 |
| | 4.3 Start-up behavior..... | 66 |
| | 4.4 Addressing..... | 67 |
| | 4.4.1 Overview..... | 67 |
| | 4.4.2 Default address assignment of the I/O part..... | 67 |
| | 4.4.3 Addressing periphery modules..... | 68 |
| | 4.5 Hardware configuration - CPU..... | 69 |
| | 4.6 Hardware configuration - System SLIO modules..... | 71 |
| | 4.7 Hardware configuration - Ethernet PG/OP channel..... | 72 |
| | 4.7.1 <i>Take IP address parameters in project</i> | 73 |

| | | |
|----------|--|------------|
| 4.8 | Setting standard CPU parameters..... | 78 |
| 4.8.1 | Parameterization via Siemens CPU..... | 78 |
| 4.8.2 | Parameter CPU..... | 78 |
| 4.8.3 | Parameter for MPI/DP | 81 |
| 4.9 | Setting VIPA specific CPU parameters..... | 82 |
| 4.9.1 | Free Module Mapping..... | 85 |
| 4.9.2 | Access settings..... | 90 |
| 4.10 | Project transfer..... | 90 |
| 4.10.1 | Transfer via MPI / optional PROFIBUS..... | 91 |
| 4.10.2 | Transfer via Ethernet..... | 92 |
| 4.10.3 | Transfer via memory card..... | 93 |
| 4.11 | Accessing the web server..... | 93 |
| 4.11.1 | Device web page CPU..... | 94 |
| 4.12 | Operating modes..... | 102 |
| 4.12.1 | Overview..... | 102 |
| 4.12.2 | Function security..... | 104 |
| 4.13 | Overall reset..... | 105 |
| 4.14 | Firmware update..... | 106 |
| 4.14.1 | Firmware update online..... | 107 |
| 4.14.2 | Firmware update via memory card..... | 108 |
| 4.15 | Reset to factory settings..... | 110 |
| 4.16 | Deployment storage media - VSD, VSC..... | 111 |
| 4.17 | Extended know-how protection..... | 113 |
| 4.18 | CMD - auto commands..... | 114 |
| 4.19 | Control and monitoring of variables with test functions..... | 116 |
| 4.20 | Diagnostic entries..... | 117 |
| 5 | Deployment I/O periphery..... | 118 |
| 5.1 | Overview..... | 118 |
| 5.2 | Address assignment..... | 119 |
| 5.3 | Analog input..... | 120 |
| 5.3.1 | Properties..... | 120 |
| 5.3.2 | Analog value representation..... | 120 |
| 5.3.3 | Wiring..... | 121 |
| 5.3.4 | Parametrization..... | 122 |
| 5.4 | Digital input..... | 123 |
| 5.4.1 | Properties..... | 123 |
| 5.4.2 | Wiring..... | 123 |
| 5.4.3 | Parametrization..... | 124 |
| 5.4.4 | Status indication..... | 125 |
| 5.5 | Digital output..... | 126 |
| 5.5.1 | Properties..... | 126 |
| 5.5.2 | Wiring..... | 126 |
| 5.5.3 | Parametrization..... | 127 |
| 5.5.4 | Status indication..... | 127 |
| 5.6 | Counting..... | 128 |
| 5.6.1 | Properties..... | 128 |
| 5.6.2 | Wiring..... | 128 |
| 5.6.3 | Proceeding..... | 130 |
| 5.6.4 | Parametrization..... | 130 |

| | | |
|----------|---|------------|
| 5.6.5 | Counter operating modes..... | 135 |
| 5.6.6 | Counter - Additional functions..... | 142 |
| 5.6.7 | Diagnostics and interrupt..... | 148 |
| 5.7 | Frequency measurement..... | 148 |
| 5.7.1 | Properties..... | 148 |
| 5.7.2 | Wiring..... | 149 |
| 5.7.3 | Proceeding..... | 150 |
| 5.7.4 | Parametrization..... | 150 |
| 5.7.5 | Status indication..... | 152 |
| 5.8 | Pulse width modulation - PWM..... | 153 |
| 5.8.1 | Properties..... | 153 |
| 5.8.2 | Wiring..... | 153 |
| 5.8.3 | Proceeding..... | 154 |
| 5.8.4 | Parametrization..... | 154 |
| 5.8.5 | Status indication..... | 156 |
| 5.9 | Pulse train..... | 157 |
| 5.9.1 | Properties..... | 157 |
| 5.9.2 | Wiring..... | 158 |
| 5.9.3 | Proceeding..... | 158 |
| 5.9.4 | Parametrization..... | 159 |
| 5.9.5 | Status indication..... | 160 |
| 5.10 | Diagnostic and interrupt..... | 161 |
| 5.10.1 | Overview..... | 161 |
| 5.10.2 | Process interrupt..... | 161 |
| 5.10.3 | Diagnostic interrupt..... | 163 |
| 6 | Deployment PtP communication..... | 170 |
| 6.1 | Fast introduction..... | 170 |
| 6.2 | Principle of the data transfer..... | 171 |
| 6.3 | Enable PtP functionality..... | 172 |
| 6.4 | Deployment of RS485 interface for PtP..... | 173 |
| 6.5 | Parametrization..... | 174 |
| 6.5.1 | FC/SFC 216 - SER_CFG - Parametrization PtP..... | 174 |
| 6.6 | Communication..... | 174 |
| 6.6.1 | FC/SFC 217 - SER_SND - Send to PtP..... | 174 |
| 6.6.2 | FC/SFC 218 - SER_RCV - Receive from PtP..... | 175 |
| 6.7 | Protocols and procedures..... | 175 |
| 6.8 | Modbus - Function codes | 179 |
| 7 | Deployment OPC UA..... | 183 |
| 7.1 | General..... | 183 |
| 7.2 | Basics OPC UA..... | 184 |
| 7.2.1 | OPC UA..... | 184 |
| 7.2.2 | Information modeling..... | 185 |
| 7.2.3 | OPC UA data types and their conversion..... | 187 |
| 7.2.4 | Integrated security concept..... | 189 |
| 7.3 | Activate OPC UA functionality..... | 192 |
| 7.4 | Usage in VIPA SPEED7 Studio..... | 192 |
| 7.5 | Usage in Siemens SIMATIC Manager..... | 192 |
| 7.5.1 | Installation OPC UA Configurator..... | 193 |
| 7.5.2 | Steps of the OPC UA configuration..... | 195 |

| | | |
|-----------|---|------------|
| 7.6 | Usage in Siemens TIA Portal..... | 195 |
| 7.6.1 | Installation <i>OPC UA Configurator</i> | 196 |
| 7.6.2 | Steps of the <i>OPC UA</i> configuration..... | 198 |
| 7.7 | Usage <i>OPC UA Configurator</i> | 199 |
| 7.7.1 | <i>OPC UA Configurator</i> | 199 |
| 7.7.2 | Project tree  | 200 |
| 7.7.3 | Device properties  | 201 |
| 7.7.4 | Server settings - Connection  | 203 |
| 7.7.5 | Server settings - Certificate  | 205 |
| 7.7.6 | Data access  | 206 |
| 7.7.7 | User management  | 207 |
| 7.7.8 | Role management  | 208 |
| 7.7.9 | Output  | 208 |
| 8 | Deployment <i>WebVisu</i> - Web visualization..... | 209 |
| 8.1 | <i>WebVisu</i> editor..... | 209 |
| 8.1.1 | Working environment..... | 210 |
| 8.1.2 | Creating a <i>WebVisu</i> project..... | 210 |
| 8.2 | Activate <i>WebVisu</i> functionality..... | 212 |
| 8.3 | Start-up of the <i>WebVisu</i> project..... | 212 |
| 8.4 | Access to the <i>WebVisu</i> | 213 |
| 8.4.1 | Status of the <i>WebVisu</i> | 213 |
| 9 | Deployment PG/OP communication - productive..... | 214 |
| 9.1 | Basics - Industrial Ethernet in automation..... | 214 |
| 9.2 | Basics - ISO/OSI reference model..... | 215 |
| 9.3 | Basics - Terms..... | 217 |
| 9.4 | Basics - Protocols..... | 218 |
| 9.5 | Basics - IP address and subnet..... | 219 |
| 9.6 | Fast introduction..... | 221 |
| 9.7 | Hardware-Konfiguration..... | 221 |
| 9.8 | Configure Siemens S7 connections..... | 221 |
| 9.9 | Configure Open Communication..... | 227 |
| 10 | Deployment PG/OP communication - PROFINET..... | 230 |
| 10.1 | Basics PROFINET..... | 230 |
| 10.2 | PROFINET installation guidelines..... | 232 |
| 10.3 | Deployment as PROFINET IO controller..... | 233 |
| 10.3.1 | Steps of configuration..... | 233 |
| 10.3.2 | Commissioning and initialization..... | 234 |
| 10.3.3 | Configuration PROFINET IO controller..... | 235 |
| 10.3.4 | Configuration PROFINET IO device..... | 237 |
| 10.4 | Deployment as PROFINET I-Device..... | 238 |
| 10.4.1 | Steps of configuration..... | 238 |
| 10.4.2 | Installing the GSDML file..... | 239 |
| 10.4.3 | Configuration as I-Device..... | 239 |
| 10.4.4 | Configuration in the higher-level IO controller..... | 241 |
| 10.4.5 | Error behavior and interrupts..... | 242 |
| 10.5 | MRP..... | 245 |
| 10.6 | Topology..... | 246 |
| 10.7 | Device replacement without exchangeable medium/PG..... | 247 |
| 10.8 | Commissioning and start-up behavior..... | 248 |

| | | |
|-----------|---|------------|
| 10.9 | PROFINET diagnostics..... | 249 |
| 10.9.1 | Overview..... | 249 |
| 10.9.2 | Diagnostics with the configuration and engineering tool..... | 249 |
| 10.9.3 | Diagnostics during runtime in the user program..... | 250 |
| 10.9.4 | Diagnostics via OB start information..... | 251 |
| 10.9.5 | Diagnostics status indication via SSLs..... | 251 |
| 10.10 | PROFINET system limits..... | 253 |
| 11 | Option: Deployment PROFIBUS communication..... | 254 |
| 11.1 | Fast introduction..... | 254 |
| 11.2 | PROFIBUS communication..... | 255 |
| 11.3 | Enable bus functionality via VSC..... | 256 |
| 11.4 | Hardware configuration - CPU..... | 256 |
| 11.5 | Deployment as PROFIBUS DP master..... | 257 |
| 11.6 | Deployment as PROFIBUS DP slave..... | 258 |
| 11.6.1 | Diagnostic functions..... | 260 |
| 11.7 | PROFIBUS installation guidelines..... | 263 |
| 11.8 | Commissioning and Start-up behavior..... | 265 |
| 12 | Configuration with VIPA SPEED7 Studio..... | 267 |
| 12.1 | <i>SPEED7 Studio</i> - Overview..... | 267 |
| 12.2 | <i>SPEED7 Studio</i> - Work environment..... | 268 |
| 12.2.1 | Project tree | 270 |
| 12.2.2 | Catalog | 271 |
| 12.3 | <i>SPEED7 Studio</i> - Hardware configuration - CPU..... | 273 |
| 12.4 | <i>SPEED7 Studio</i> - Hardware configuration - Ethernet PG/OP channel..... | 274 |
| 12.5 | <i>SPEED7 Studio</i> - Hardware configuration - I/O modules..... | 277 |
| 12.6 | Deployment I/O periphery..... | 277 |
| 12.6.1 | Overview..... | 277 |
| 12.6.2 | Analog input..... | 278 |
| 12.6.3 | Digital input..... | 278 |
| 12.6.4 | Digital output..... | 279 |
| 12.6.5 | Counter..... | 279 |
| 12.6.6 | Frequency measurement..... | 283 |
| 12.6.7 | Pulse width modulation - PWM..... | 285 |
| 12.6.8 | Pulse Train..... | 287 |
| 12.7 | Deployment <i>OPC UA</i> | 289 |
| 12.8 | Deployment <i>WebVisu - Web visualization</i> | 289 |
| 12.9 | <i>SPEED7 Studio</i> - Project transfer..... | 289 |
| 12.9.1 | Transfer via MPI..... | 289 |
| 12.9.2 | Transfer via Ethernet..... | 291 |
| 12.9.3 | Transfer via memory card..... | 292 |
| 13 | Configuration with TIA Portal..... | 294 |
| 13.1 | TIA Portal - Work environment | 294 |
| 13.1.1 | General..... | 294 |
| 13.1.2 | Work environment of the TIA Portal..... | 295 |
| 13.2 | TIA Portal - Functional limitations..... | 296 |
| 13.3 | TIA Portal - Hardware configuration - CPU..... | 296 |
| 13.4 | TIA Portal - Hardware configuration - Ethernet PG/OP channel..... | 299 |
| 13.4.1 | <i>Take IP address parameters in project</i> | 301 |
| 13.5 | TIA Portal - Hardware configuration - I/O modules..... | 304 |

| | | |
|---------|---|------------|
| 13.6 | TIA Portal - Deployment PG/OP communication - PROFINET..... | 305 |
| 13.6.1 | Deployment as PROFINET IO controller..... | 305 |
| 13.6.2 | Deployment as PROFINET I-Device..... | 310 |
| 13.7 | TIA Portal - Option: Deployment PROFIBUS communication..... | 316 |
| 13.7.1 | Fast introduction..... | 316 |
| 13.7.2 | Activating bus functionality by means of a VSC..... | 316 |
| 13.7.3 | Hardware configuration - CPU..... | 316 |
| 13.7.4 | Deployment as PROFIBUS DP master..... | 316 |
| 13.7.5 | Deployment as PROFIBUS DP slave..... | 318 |
| 13.8 | Deployment <i>OPC UA</i> | 319 |
| 13.9 | TIA Portal - VIPA-Include library..... | 320 |
| 13.10 | TIA Portal - Project transfer..... | 320 |
| 13.10.1 | Transfer via MPI..... | 321 |
| 13.10.2 | Transfer via Ethernet..... | 321 |
| 13.10.3 | Transfer via memory card..... | 322 |
| | Appendix | 323 |
| A | System specific event IDs..... | 325 |
| B | Integrated blocks..... | 377 |
| C | SSL partial list..... | 380 |

1 General

1.1 Copyright © VIPA GmbH

All Rights Reserved

This document contains proprietary information of VIPA and is not to be disclosed or used except in accordance with applicable agreements.

This material is protected by the copyright laws. It may not be reproduced, distributed, or altered in any fashion by any entity (either internal or external to VIPA), except in accordance with applicable agreements, contracts or licensing, without the express written consent of VIPA and the business management owner of the material.

For permission to reproduce or distribute, please contact: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

E-Mail: 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.

EC 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, S7-400 and S7-1500 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

E-Mail: 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)

E-Mail: support@vipa.de

1.2 About this manual

Objective and contents

This manual describes the CPU 013-CCF0R00 of the System SLIO from VIPA. It contains a description of the construction, project implementation and usage.

| Product | Order number | as of state: | |
|----------|--------------|--------------|--------|
| | | CPU-HW | CPU-FW |
| CPU 013C | 013-CCF0R00 | 01 | V3.0.0 |

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.



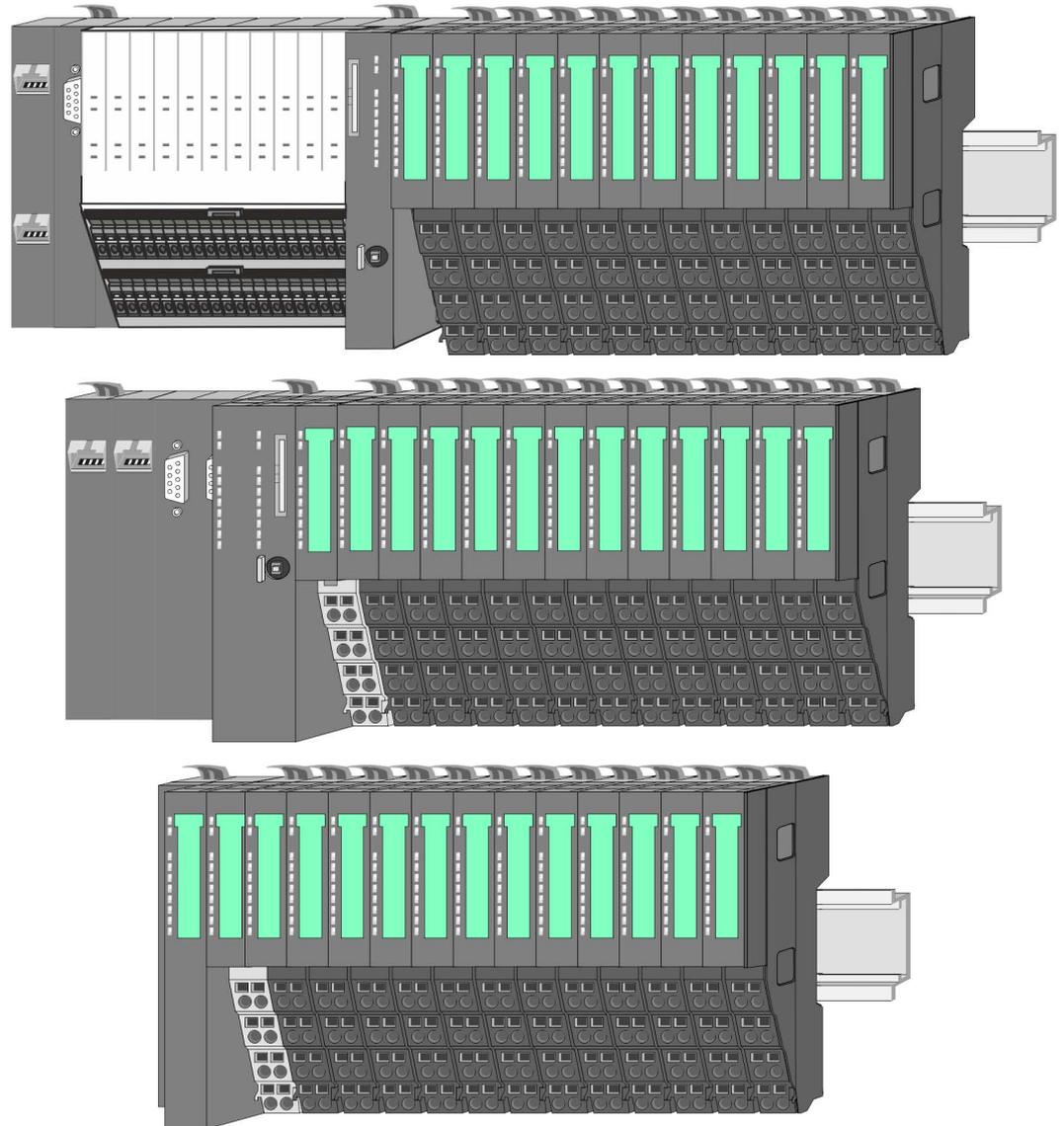
CAUTION!

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

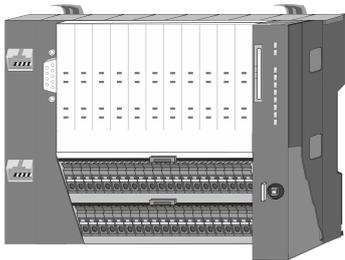
- CPU (head module)
- Bus coupler (head module)
- Line extension
- Periphery modules
- Accessories



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



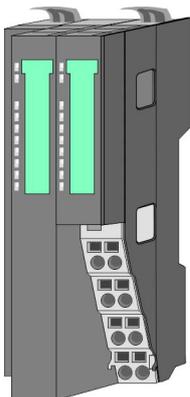
With this CPU 01x, the CPU electronic and power supply are integrated to one casing. As head module, via the integrated power module for power supply, CPU electronic and the electronic of the connected periphery modules are supplied. The DC 24 power section supply for the linked periphery modules is 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



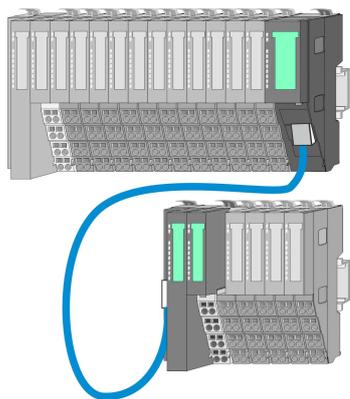
With a bus coupler bus interface and power module is integrated to one casing. With the bus interface you get access to a subordinated bus system. As head module, via the integrated power module for power supply, bus interface and the electronic of the connected periphery modules are supplied. The DC 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!

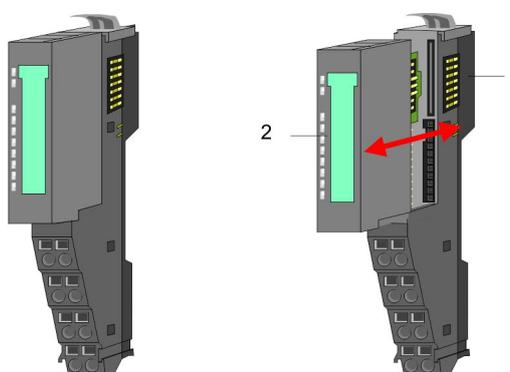
Line extension



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.

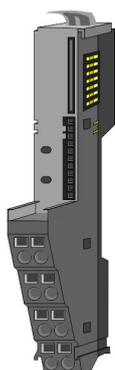
Periphery modules

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



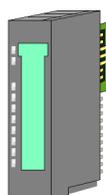
- 1 Terminal module
- 2 Electronic module

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



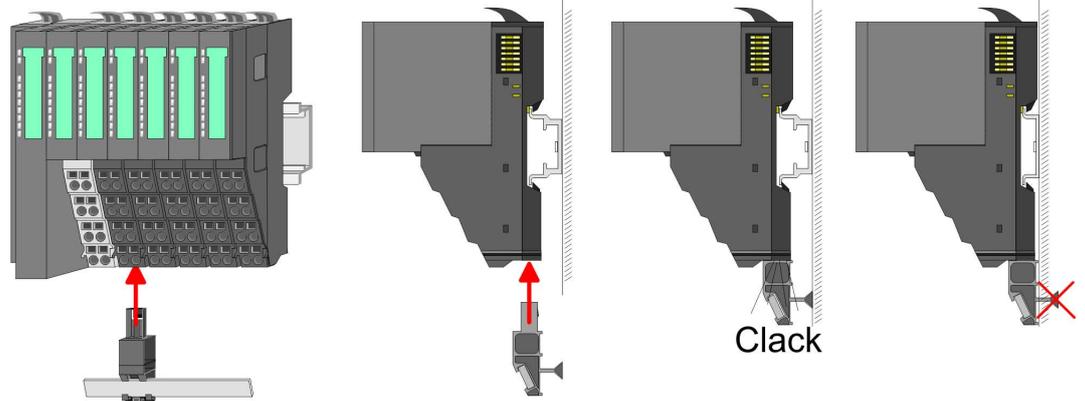
The functionality of a SLIO periphery module is defined by the *electronic* module, which is mounted to the terminal module by a sliding mechanism. With an error the defective module may be exchanged for a functional module with standing installation. At the front side there are LEDs for status indication.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

2.2.3 Accessories

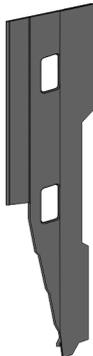
Shield bus carrier



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

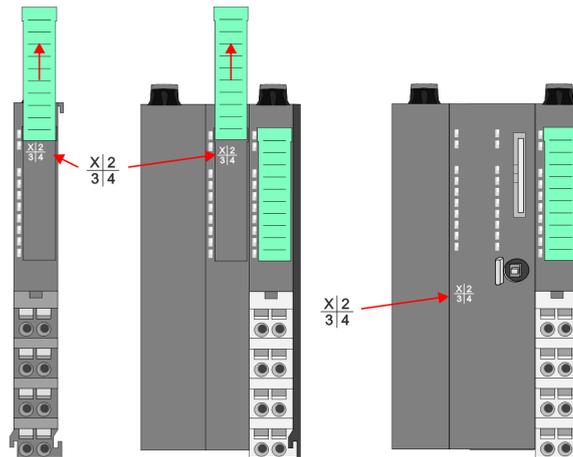


There is the possibility to fix the assignment of electronic and terminal module. Here coding pins (order number 000-0AC00) from VIPA can be used. The coding pin consists of a coding jack and a coding plug. By combining electronic and terminal module with coding pin, the coding jack remains in the electronic 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.

2.2.4 Hardware revision

Hardware revision on the front

- The hardware revision version is printed on every System SLIO module.
- Since a System SLIO module consists of a terminal and electronics module, you will find a hardware revision on each of them.
- Authoritative for the hardware revision of a System SLIO module is the hardware revision of the electronic module. This is always located under the labeling strip of the corresponding electronic module.
- On modules without labeling strip, such as CPUs, the hardware revision is printed on the front.



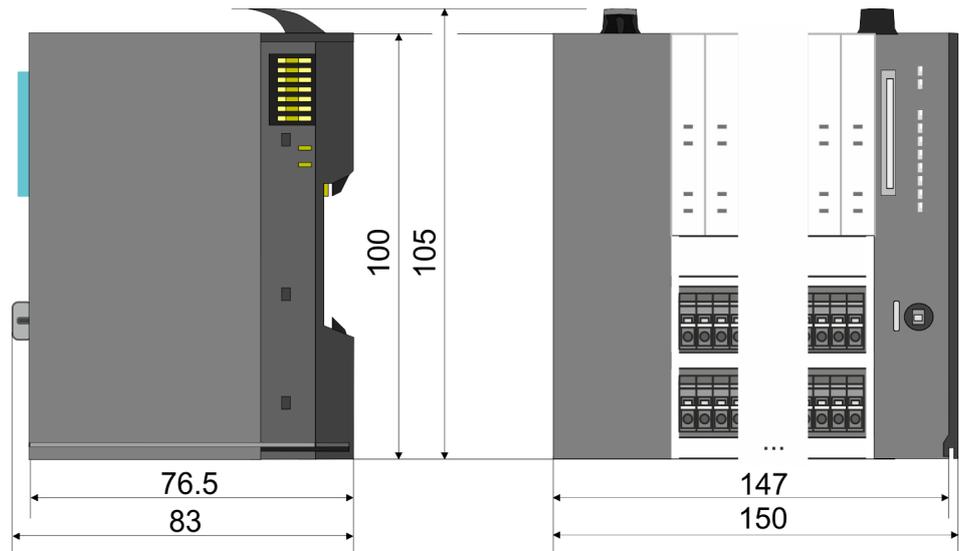
The example here shows the hardware revision 1. 1 is marked with 'X'.

Hardware revision via web server

On the CPUs and some bus couplers, you can output the hardware revision 'HW Revision' via the integrated web server.

2.3 Dimensions

Dimensions CPU 01xC

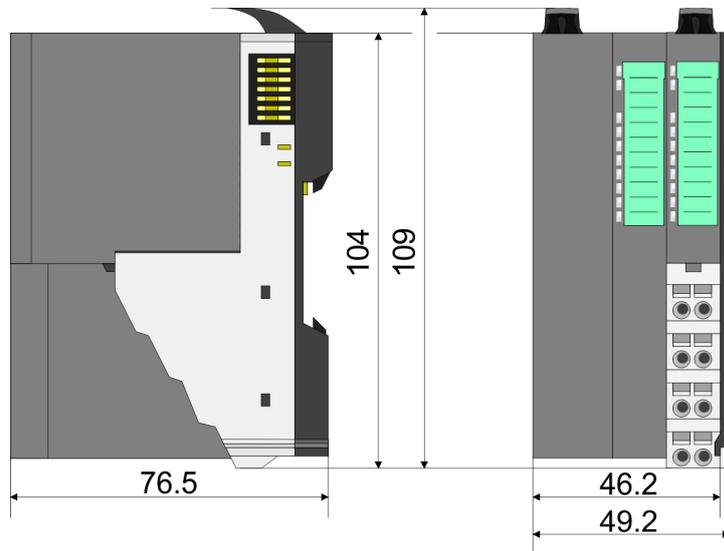


Dimensions

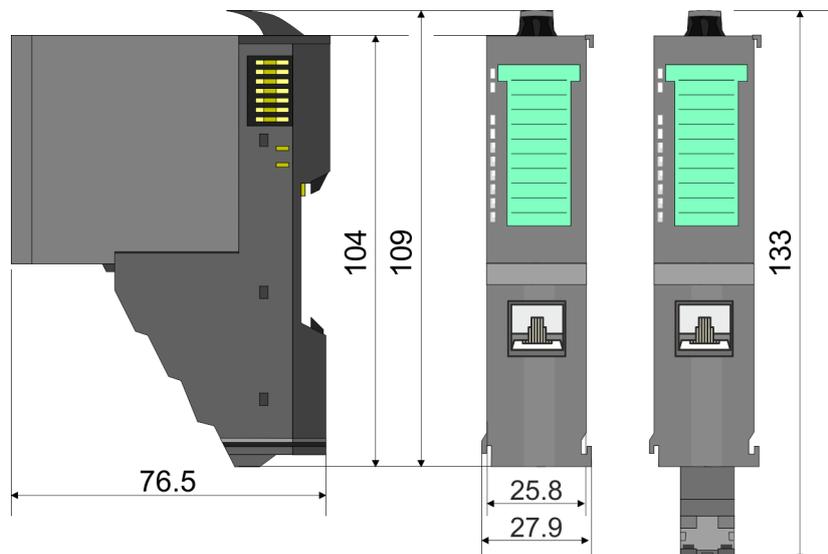
Dimensions CPU 01x



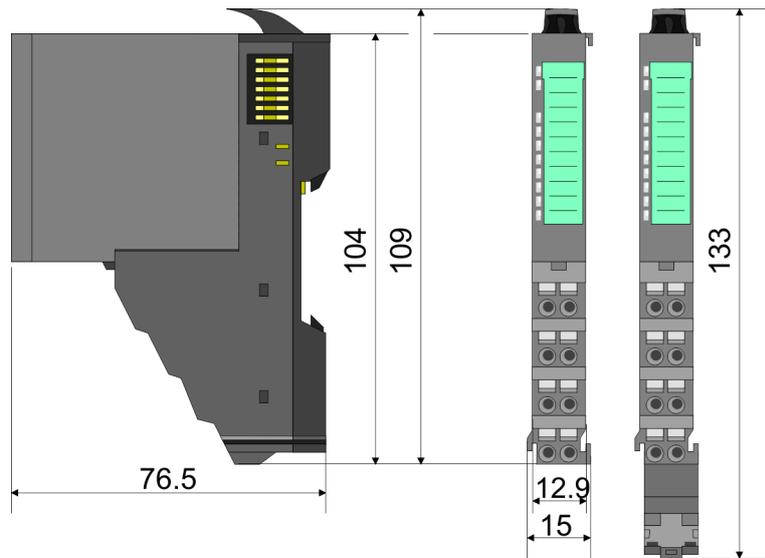
Dimensions bus coupler and line extension slave



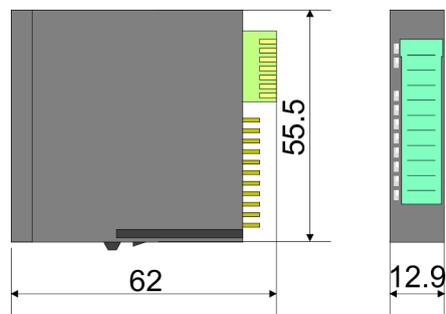
Dimensions line extension master



Dimension periphery module



Dimensions electronic module



Dimensions in mm

2.4 Mounting

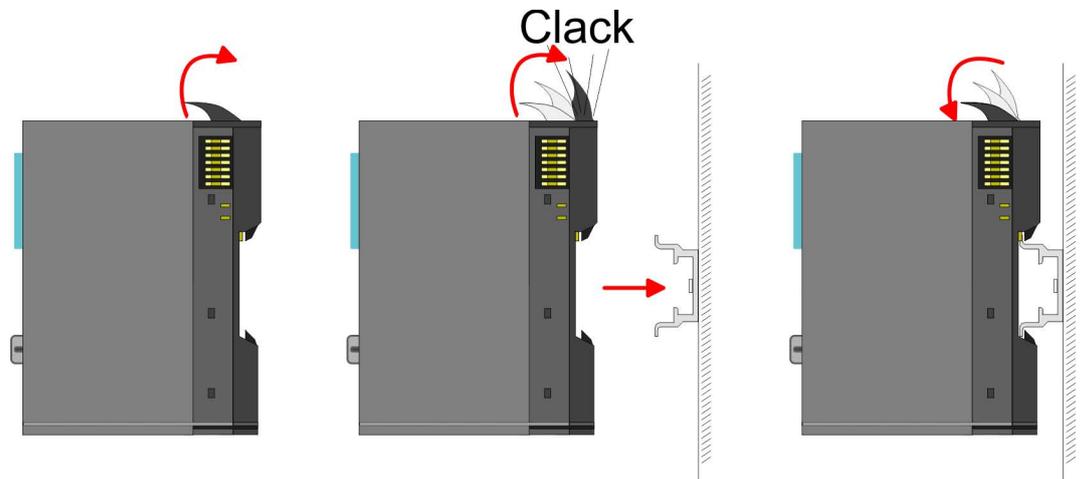


Requirements for UL compliance use

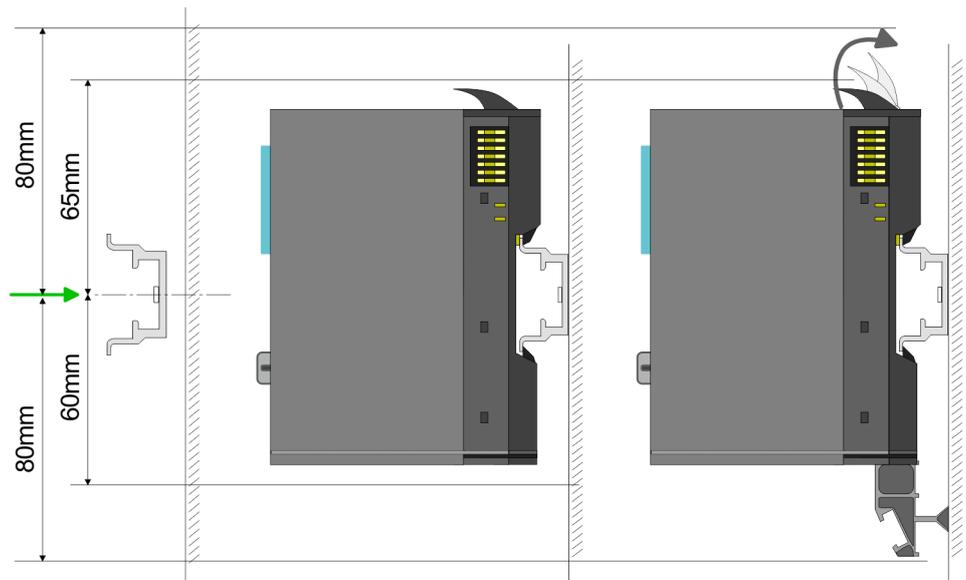
- Use for power supply exclusively SELV/PELV power supplies.
- The System SLIO must be installed and operated in a housing according to IEC 61010-1 9.3.2 c).

2.4.1 Mounting CPU 01xC

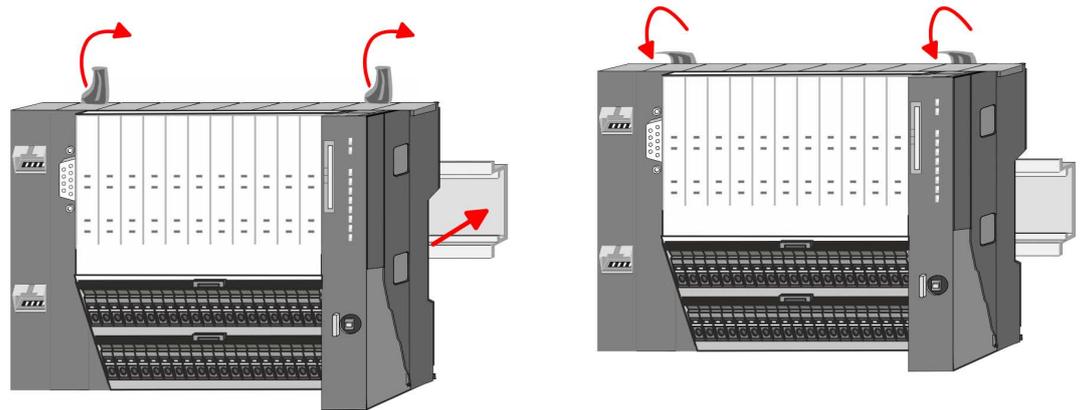
There are locking lever at the top side of the CPU. For mounting and demounting these locking lever are to be turned upwards until these engage. Place the CPU at the mounting rail. The CPU is fixed to the mounting rail by pushing downward the locking levers. The CPU is directly mounted at a mounting 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 1A. By means of the power module 007-1AB10 the current of the electronic power supply may be expanded accordingly.



Proceeding

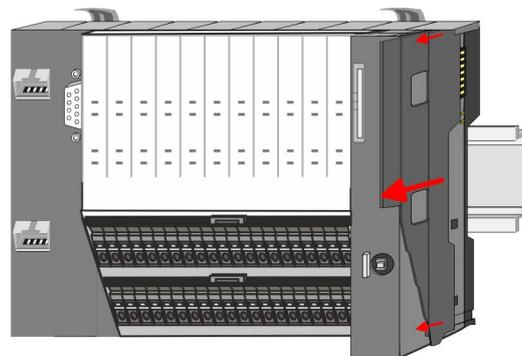


1. → Mount the mounting rail! Please consider that a clearance from the middle of the mounting rail of at least 80mm above and 60mm below, respectively 80mm by deployment of shield bus carriers, exist.



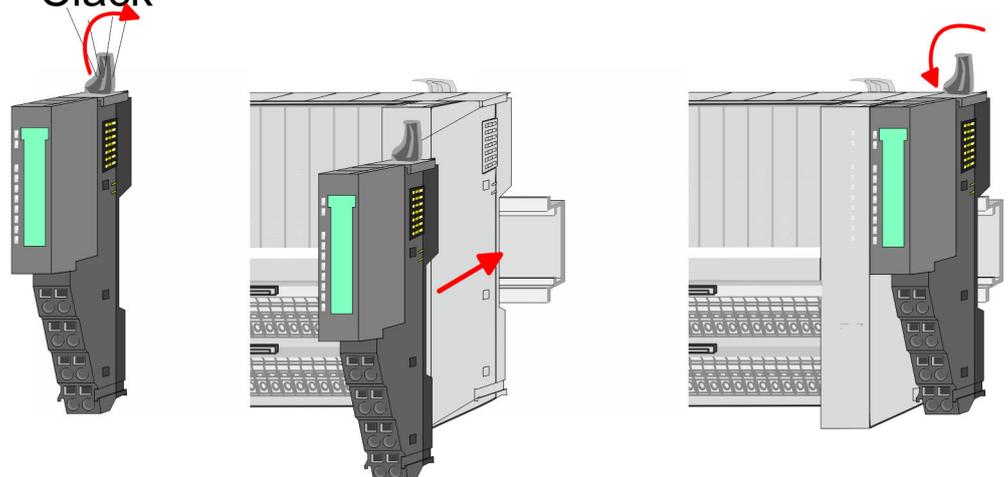
2. Turn the locking lever upwards, place the CPU at the mounting rail and turn the lever downward.
 - ⇒ If you want to use the CPU without periphery modules, the mounting is now complete.

Mounting periphery modules

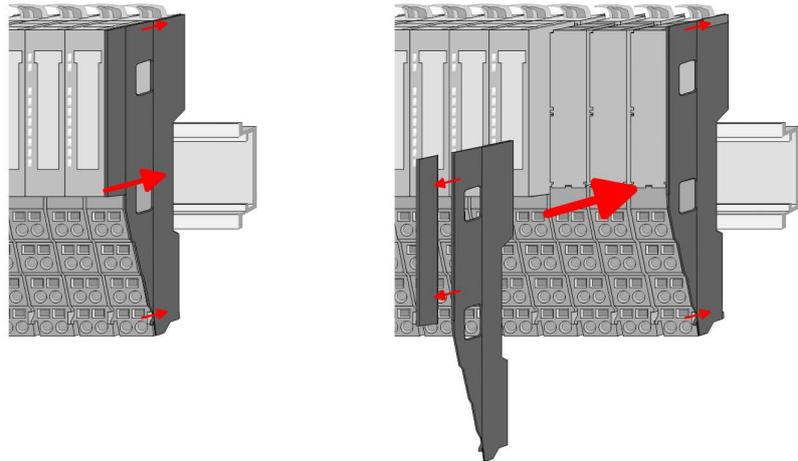


1. Before mounting the periphery modules you have to remove the bus cover at the right side of the CPU by pulling it forward. Keep the cover for later mounting.

Clack



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!

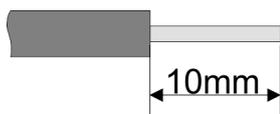
The system is specified for SELV/PELV environment. Devices, which are attached to the system must meet these specifications. Installation and cable routing other than SELV/PELV specification must be separated from the system's equipment!

2.5.1 Wiring CPU 01xC

CPU connector

For wiring the CPU 01xC has removable connectors. With the wiring of the connectors a "push-in" spring-clip technique is used. This allows a quick and easy connection of your signal and supply lines. The clamping off takes place by means of a screwdriver.

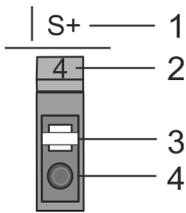
Data



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

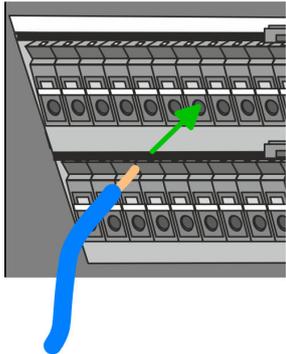
Use for wiring rigid wires respectively use wire sleeves. When using stranded wires you have to press the release button with a screwdriver during the wiring.

Wiring procedure



- 1 Labeling on the casing
- 2 Pin no. at the connector
- 3 Release button
- 4 Connection hole for wire

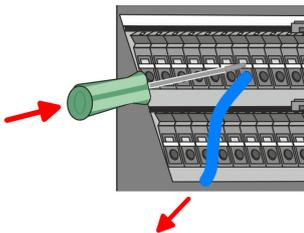
Insert wire



The wiring happens without a tool.

- ➔ Determine according to the casing labelling the connection position and insert through the round connection hole of the according contact your prepared wire until it stops, so that it is fixed.
 - ⇒ By pushing the contact spring opens, thus ensuring the necessary contact pressure.

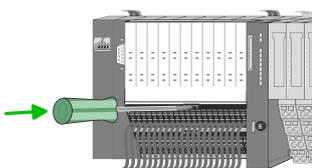
Remove wire



The wire is to be removed by means of a screwdriver with 2.5mm blade width.

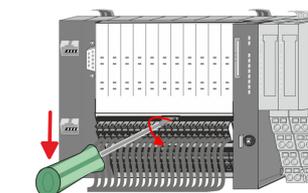
- 1. ➔ Press with your screwdriver vertically at the release button.
 - ⇒ The contact spring releases the wire.
- 2. ➔ Pull the wire from the round hole.

Remove connectors (module replacement)



By means of a screwdriver there is the possibility to remove the connectors e.g. for module exchange with a fix wiring. For this each connector has a release lever centrally on its top side. Unlocking takes place by the following proceeding:

- 1. ➔ Remove connector:
 - Push your screwdriver horizontally into the slot between connector and release lever, until it stops.
- 2. ➔ Push the screwdriver down:
 - ⇒ The connector is unlocked and can be removed by turning downwards.

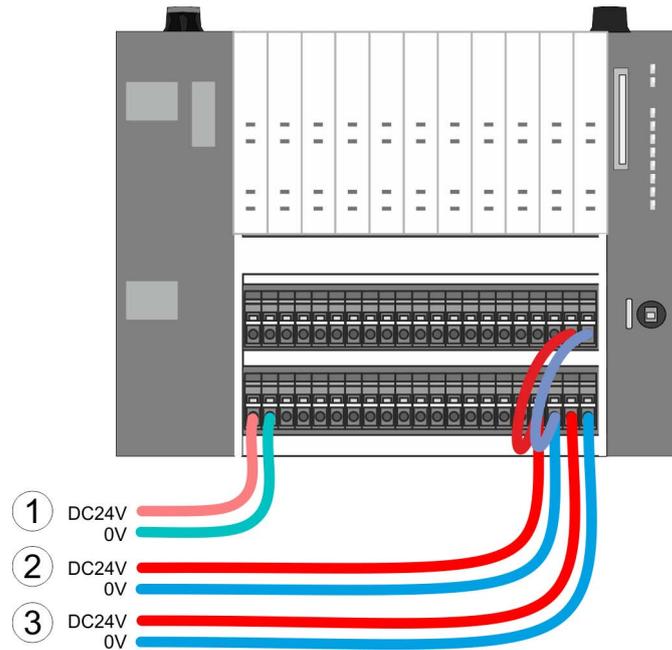


CAUTION!

Via wrong operation such as pressing, the screwdriver upward the release lever may be damaged.

- 3. ➔ Plug connector:
 - The connector is plugged by setting it at the bottom line and engage with a with a slight twist upwards into the release lever.

Standard wiring



- (1) DC 24V for electronic section supply of the CPU, the internal I/Os and SLIO bus
- (2) DC 24V for power section supply integrated I/Os
- (3) DC 24V for power section supply SLIO bus



The electronic power section supply is internally protected against higher voltage by fuse. The fuse is located inside the CPU and can not be changed by the user.

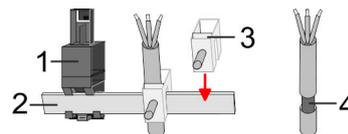
Fusing

- It is recommended to externally protect the electronic power supply for CPU and SLIO bus with a 3A fuse (fast) respectively by a line circuit breaker 3A characteristics Z.
- The power section supply of the internal I/Os is to be externally protected with a 6A fuse (fast) respectively by a line circuit breaker 6A characteristics Z!
- The power section supply of the SLIO bus is to be externally protected with a 6A fuse (fast) respectively by a line circuit breaker 6A characteristics Z! For UL-compliant operation, use a fuse that is UL listed or "UL-recognized".

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 1A. With a sum current greater than 1A the LEDs may not be activated. Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

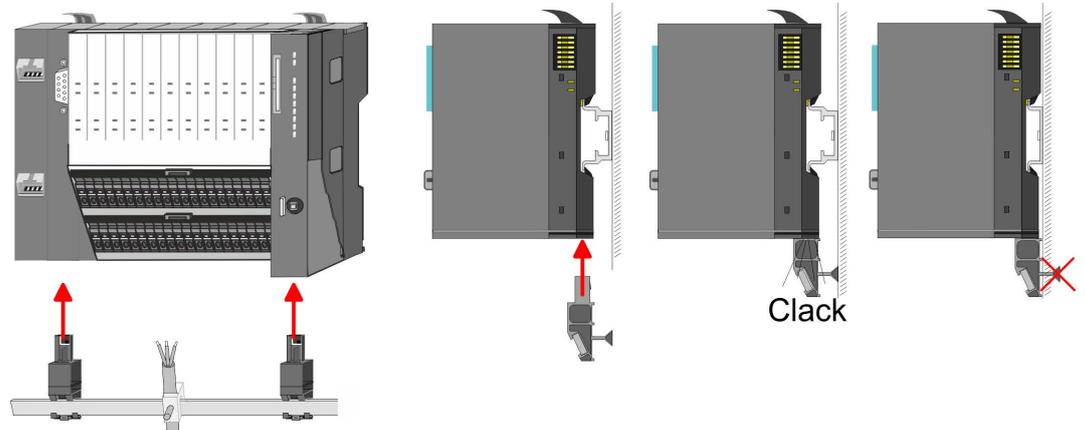
Shield attachment



- 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



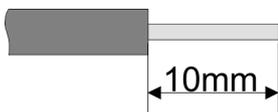
CAUTION!

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!

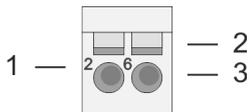
With wiring the terminal modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

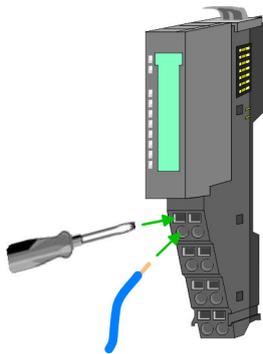
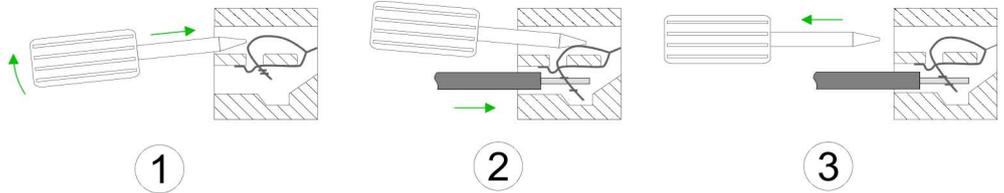


| | |
|------------------|---|
| U_{max} | 240V AC / 30V DC |
| I_{max} | 10A |
| Cross section | 0.08 ... 1.5mm ² (AWG 28 ... 16) |
| Stripping length | 10mm |

Wiring procedure

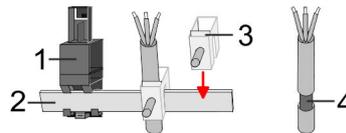


- 1 Pin number at the connector
- 2 Opening for screwdriver
- 3 Connection hole for wire



- 1. Insert a suited screwdriver at an angle 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.

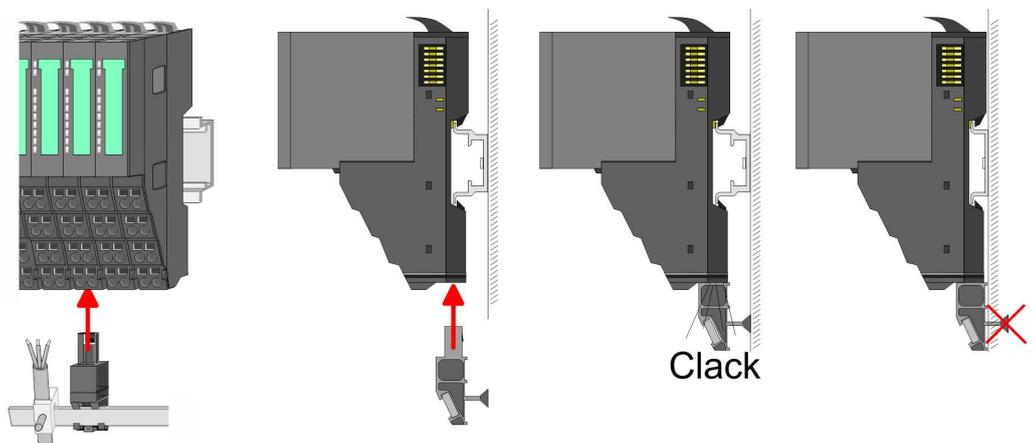
Shield attachment



- 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

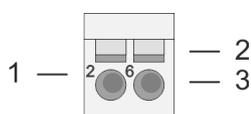
Terminal module terminals Power modules are either integrated to the head module or may be installed between the periphery modules. With power modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines.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

Data

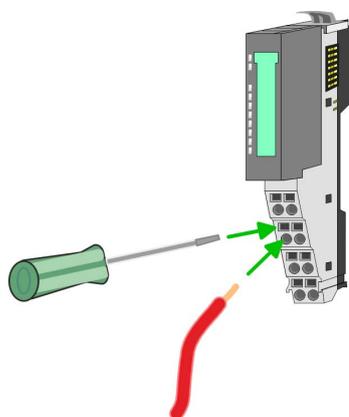
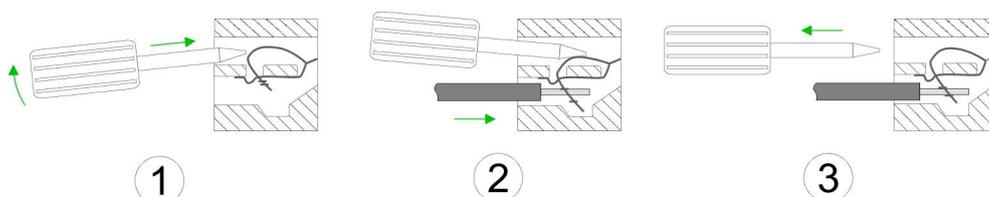


| | |
|------------------|---|
| U_{\max} | 30V DC |
| I_{\max} | 10A |
| Cross section | 0.08 ... 1.5mm ² (AWG 28 ... 16) |
| Stripping length | 10mm |

Wiring procedure

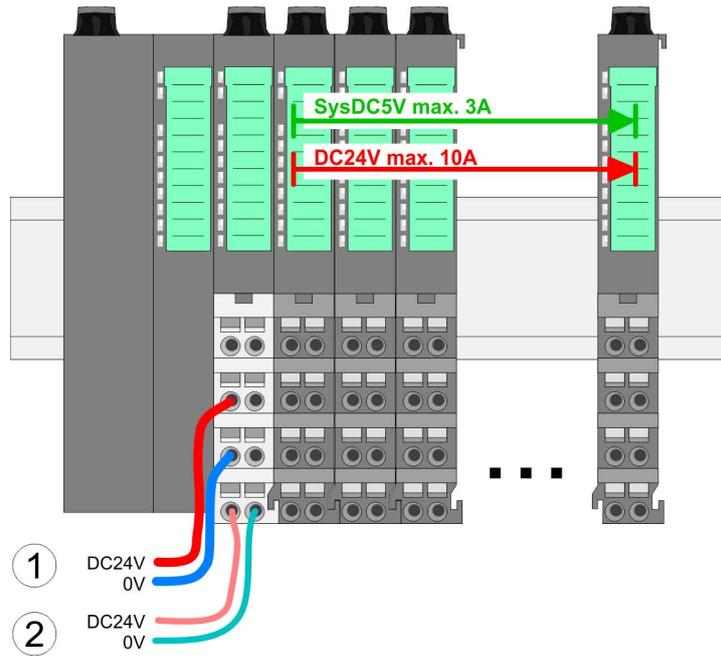


- 1 Pin number at the connector
- 2 Opening for screwdriver
- 3 Connection hole for wire



1. Insert a suited screwdriver at an angle 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.

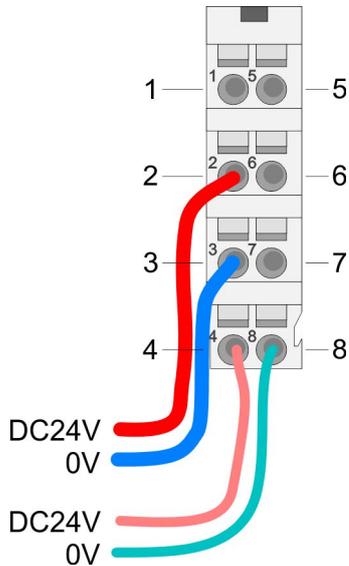
Standard wiring



- (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.

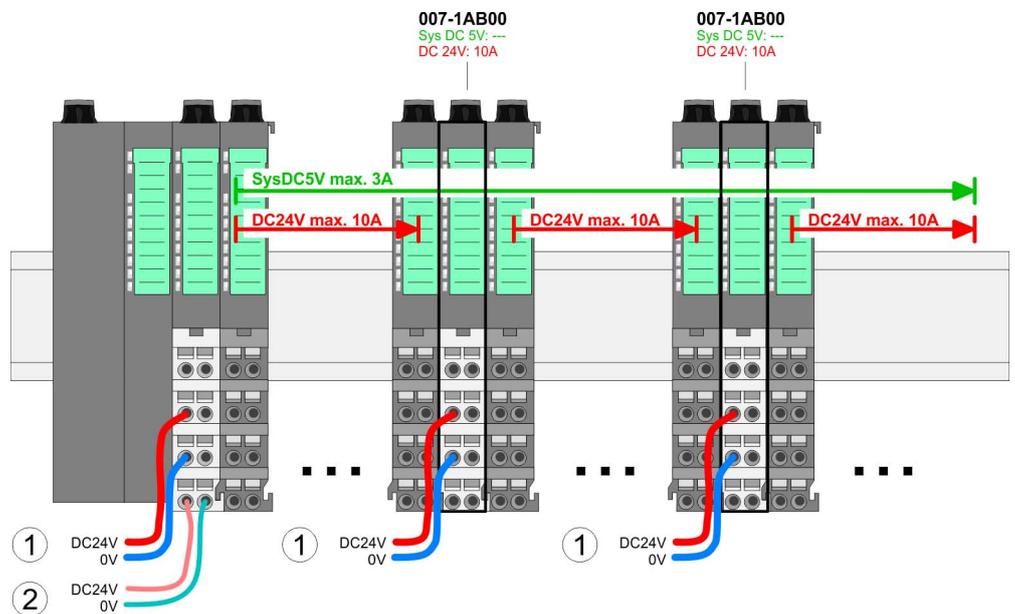
State of the electronic power supply via LEDs

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

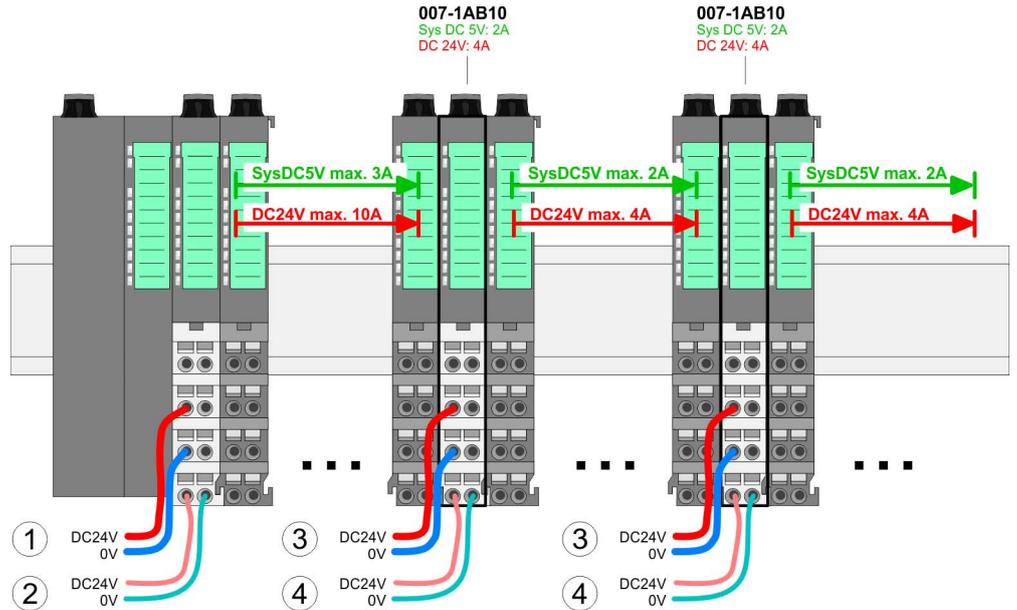
Deployment of the power modules

- 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

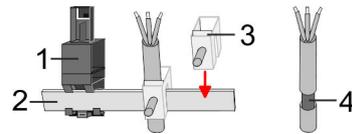


Power module 007-1AB10



- (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

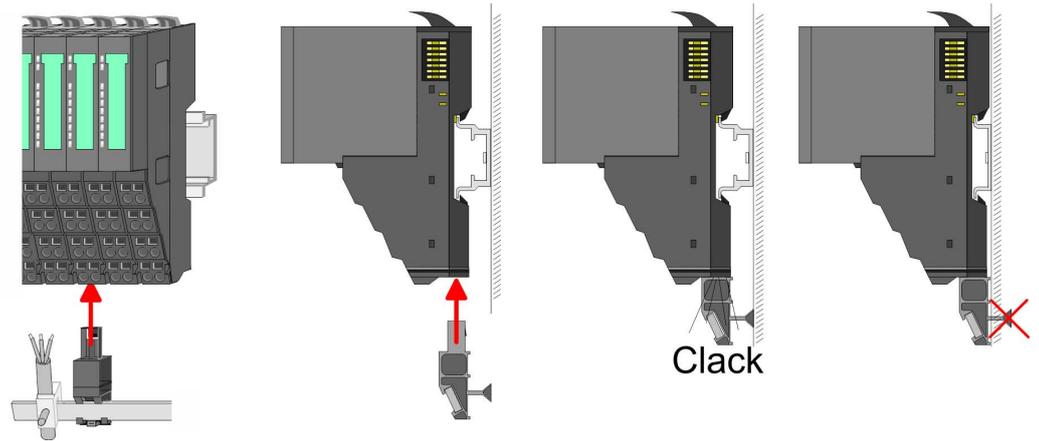
Shield attachment



- 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.

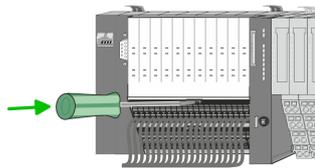
2.6 Demounting

2.6.1 Demounting CPU 01xC

Proceeding

Remove connector

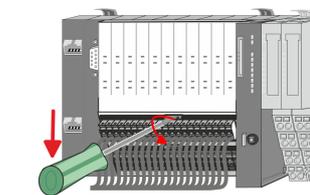
By means of a screwdriver there is the possibility to remove the connectors e.g. for module exchange with a fix wiring. For this each connector has a release lever centrally on its top side. Unlocking takes place by the following proceeding:



1. Power-off your system.

2. Remove connector:

Push your screwdriver horizontally into the slot between connector and release lever, until it stops.



3. Push the screwdriver down

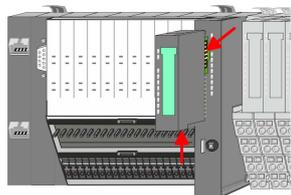
⇒ The connector is unlocked and can be removed by turning downwards.



CAUTION!

Via wrong operation such as pressing, the screwdriver upward the release lever may be damaged.

CPU replacement

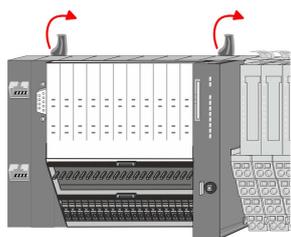


1.

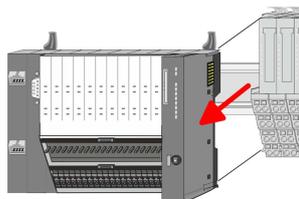


For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.

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

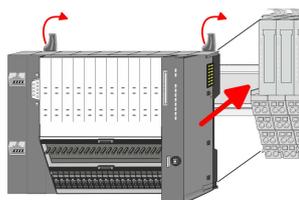


2. Turn all the locking lever of the CPU to be exchanged upwards.



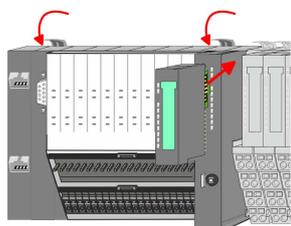
3. Pull the CPU forward.

4. For mounting turn all the locking lever of the CPU to be mounted upwards.



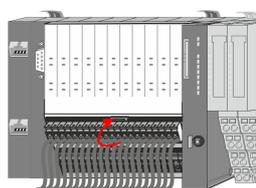
5. ➤ To mount the CPU put it to the periphery module and push it, guided by the stripes, to the mounting rail.

6. ➤ Turn all the locking lever downward, again.



7. ➤ Plug again the electronic module, which you have removed before. For installation plug the electronic module guided by the stripes at the lower side until this engages to the terminal module.

Plug connector



➤ Put the connector on the bottom edge and push it, as shown in the figure, with a rotation upwards into the release lever until it engages.

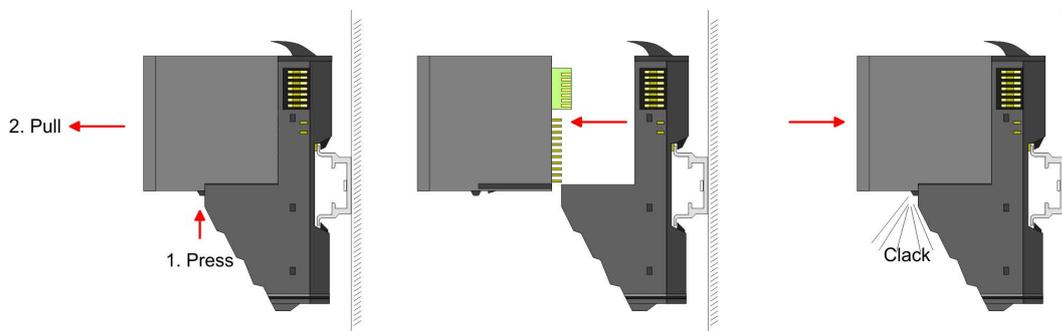
⇒ Now you can bring your system back into operation.

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.

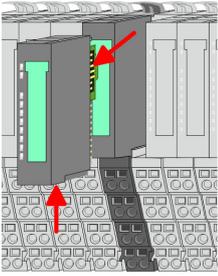
3. ➤ For installation plug the new electronic module guided by the stripes at the lower side until this engages to the terminal module.

⇒ Now you can bring your system back into operation.

Exchange of a periphery module

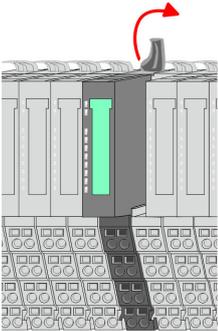
1. ➤ Power-off your system.

2. ➤ Remove if exists the wiring of the module.

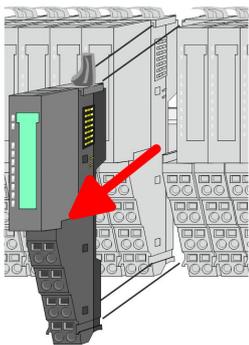


3.  For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.

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

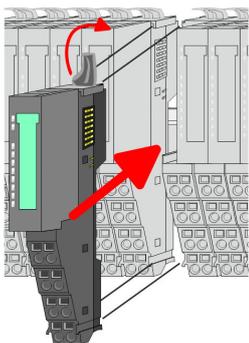


4. Turn the locking lever of the module to be exchanged upwards.



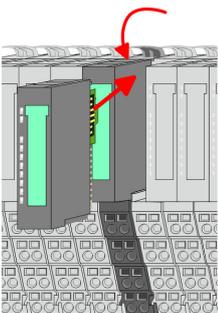
5. Pull the module.

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.

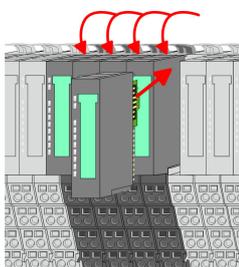
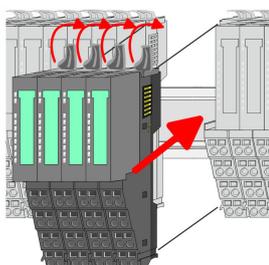
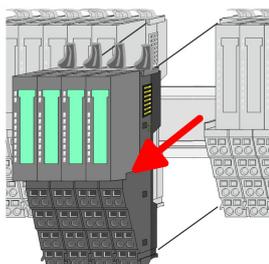
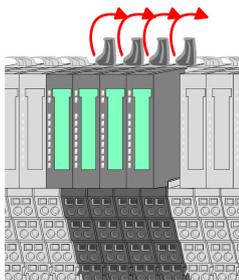
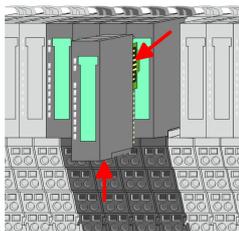


9. Plug again the electronic module, which you have removed before.

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.

2.6.2.1 Easy Maintenance

Overview

Easy Maintenance means the support for adding and removing modules during operation without having to restart the system. There are the following behaviors:

- Electronic module is removed
 - The CPU detects a module failure on the backplane bus.
 - Diagnostic message ‘*System SLIO bus failure*’ (0x39D0) is triggered.
 - OB 86 is called. If this is not available, the CPU switches to STOP otherwise it remains in RUN.
 - The SF LED of the CPU lights up.
 - The I/O data of all modules become invalid.
- Identical electronic module is plugged
 - The CPU detects the module return on the backplane bus.
 - The SF-LED of the CPU gets off.
 - All RUN LEDs on the modules get on and the MF LEDs get off.
 - Diagnostic message ‘*System SLIO bus recovery*’ (0x38D0) is triggered.
 - OB 86 is called. If this is not available, the CPU switches to STOP otherwise it remains in RUN.
 - The I/O data of all modules become valid again.
- Wrong electronic module is plugged
 - The CPU detects the wrong module.
 - Diagnostic message ‘*System SLIO bus recovery, but expected configuration does not match actual configuration*’ (0x38D1) is triggered.
 - The SF LED of the CPU remains on.
 - The MF LED of the wrong module flashes.
 - OB 86 is called. If this is not available, the CPU switches to STOP otherwise it remains in RUN.
 - With the exception of the wrong module, the I/O data of all modules become valid again.



Please note that the CPU switches to STOP, if there is no OB 86 configured when adding or removing System SLIO modules!

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

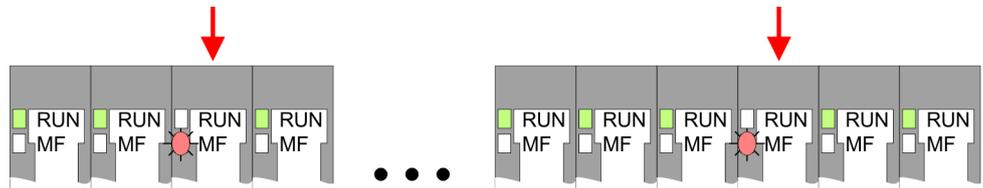


Behaviour: After PowerON the RUN LED of each module is off and the MF LED of each module is sporadically on.

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. ↪ Chap. 2.5.3 'Wiring power modules' page 27

Error in configuration

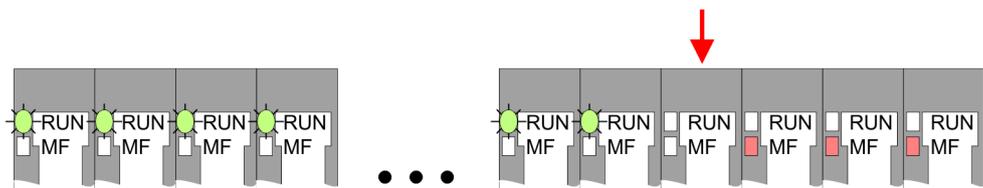


Behaviour: After PowerON the MF LED of one module respectively more modules flashes. The RUN LED remains off.

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 | <p>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.</p> |
| What does EMC mean? | <p>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.</p> <p>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.</p> |
| Possible interference causes | <p>Electromagnetic interferences may interfere your control via different ways:</p> <ul style="list-style-type: none">■ Electromagnetic fields (RF coupling)■ Magnetic fields with power frequency■ Bus system■ Power supply■ Protected earth conductor <p>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.</p> <p>There are:</p> <ul style="list-style-type: none">■ galvanic coupling■ capacitive coupling■ inductive coupling■ radiant coupling |
| Basic rules for EMC | <p>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.</p> <ul style="list-style-type: none">■ Take care of a correct area-wide grounding of the inactive metal parts when installing your components.<ul style="list-style-type: none">– 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.<ul style="list-style-type: none">– 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

General data

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...+60°C |
| Vertical installation | EN 61131-2 | 0...+60°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 |

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 * | |

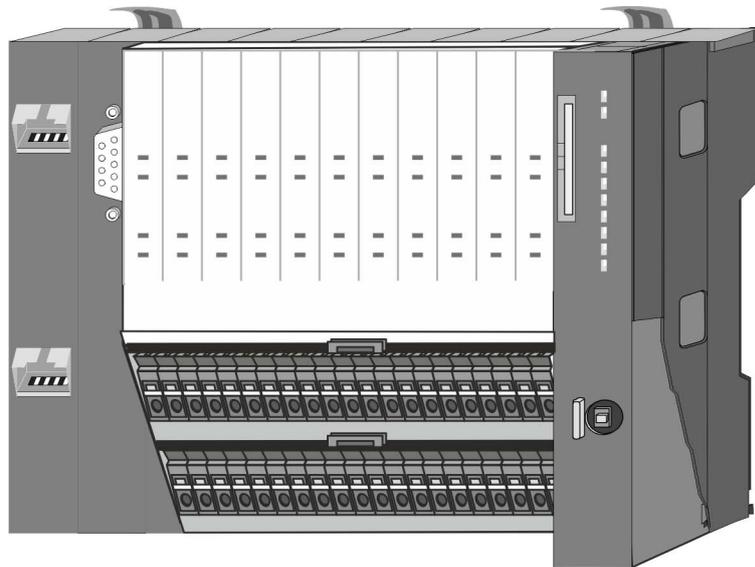
*) Due to the high-energetic single pulses with Surge an appropriate external protective circuit with lightning protection elements like conductors for lightning and overvoltage is necessary.

3 Hardware description

3.1 Properties

CPU 013-CCF0R00

- SPEED7 technology integrated
- Programmable via VIPA SPEED7 Studio, Siemens SIMATIC Manager or TIA Portal
- Integrated work memory 64kbyte (32kbyte code, 32kbyte data)
- Work memory expandable up to 128kbyte (64kbyte code, 64kbyte data)
- 128kbyte load memory integrated
- Slot for external storage media (lockable)
- Status LEDs for operating state and diagnostics
- X1/X2: Ethernet PG/OP channel (switch) for active and passive communication integrated
- X3: MPI(PB) interface: MPI interface with via VSC unlock able field bus functions
- Integrated Digital I/Os: DI 16xDC24V; DO 12xDC24V, 0,5A
- Integrated Analog Input : AI 2x12Bit (single ended)
- 4 channels for counter, frequency measurement and 2 channels for pulse width modulation
- Pulse Train via SFB 49 (PULSE)
- PROFINET IO controller and I-Device via Ethernet PG/OP channel
- OPC UA project via Ethernet PG/OP channel
- WebVisu project via Ethernet PG/OP channel
- Up to 64 SLIO modules placeable
- I/O address area digital/analog 2048byte
- 512 timer/counter, 8192 flag byte



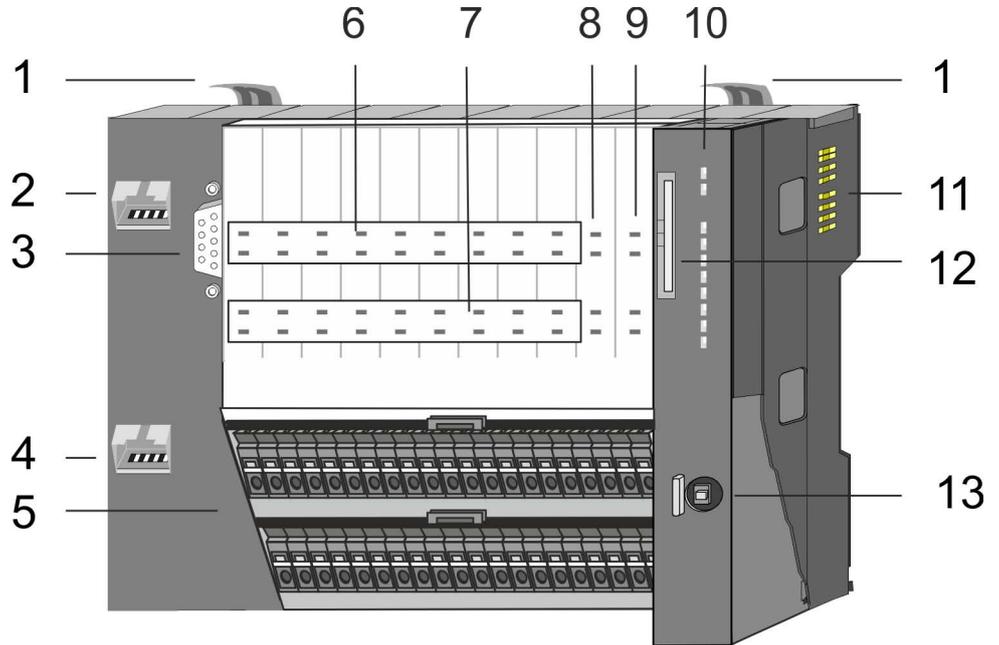
Ordering data

| Type | Order number | Description |
|----------|--------------|--|
| CPU 013C | 013-CCF0R00 | Compact CPU 013C with options to extend work memory and field bus interface with DI 16xDC24V, DO 12xDC24V 0.5A, AI 2x12Bit and 4 channels technological function |

3.2 Structure

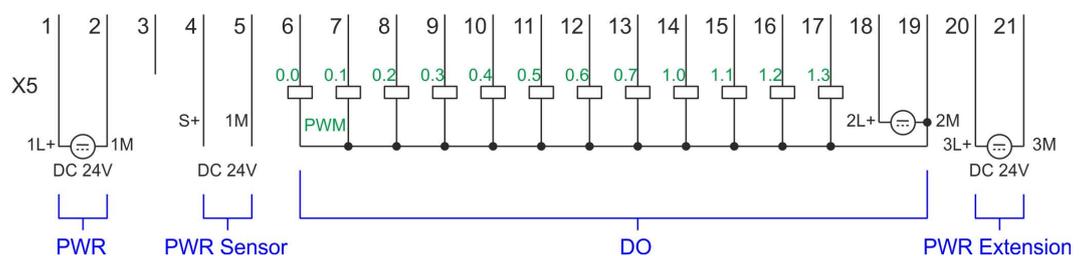
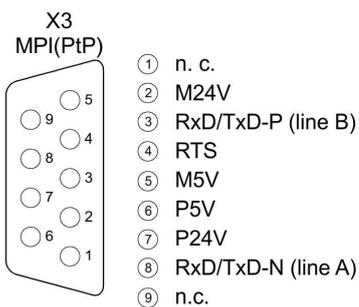
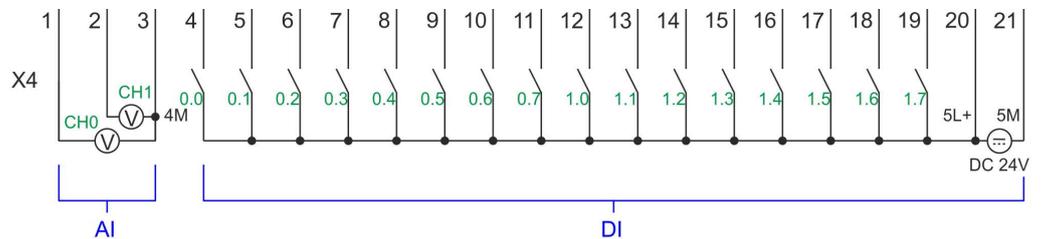
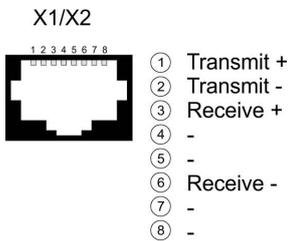
3.2.1 Compact CPU

CPU 013C



- 1 Locking lever
- 2 X1: Ethernet PG/OP channel 1
- 3 X3: MPI(PtP) interface
- 4 X2: Ethernet PG/OP channel 2
- 5 X4, X5: Connector IO part
- 6 DI +x: LEDs integrated DI
- 7 DO +x: LEDs integrated DO
- 8 xL+: LEDs power supply
- 9 xF: LEDs error
- 10 LEDs CPU part
- 11 Backplane bus
- 12 Slot for external storage media (lockable)
- 13 Operating mode switch CPU

3.2.2 Interfaces



X1/X2: 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 Ethernet PG/OP channel (X1/X2) is designed as switch. This enables PG/OP communication via the connections X1 and X2.
- Configurable connections are possible.
- DHCP respectively the assignment of the network configuration with a DHCP server is supported.
- Default diagnostics addresses: 2025 ... 2040
- 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 by means of your configuration tool. This is called "initialization".
- Via the Ethernet PG/OP channel, you have access to:
 - Device website, where you can find information on firmware status, connected peripherals, current cycle times, etc.
 - *OPC UA* project, which is to be created in the *OPC UA Configurator*.
 - *WebVisu* project, which is to be created in the *SPEED7 Studio*.
 - PROFINET IO controller or the PROFINET I-Device.

↪ *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 72*

↪ *Chap. 9 'Deployment PG/OP communication - productive' page 214*

X3: MPI(PtP) interface*9pin SubD jack: (isolated)*

The interface supports the following functionalities, which are switch able:

- MPI (default / after overall reset)

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.
- PtP

The RS485 interface can be switched to PtP functionality ↪ *Chap. 4.9 'Setting VIPA specific CPU parameters' page 82*. Using the *PtP* functionality the RS485 interface is allowed to connect via serial point-to-point connection to different source res. target systems.

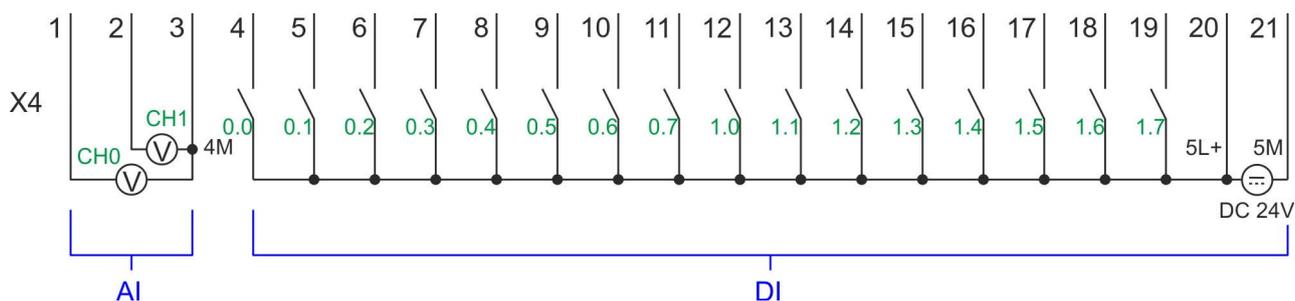
The following protocols are supported:

 - ASCII
 - STX/ETX
 - 3964R
 - USS
 - Modbus master (ASCII, RTU)
- PROFIBUS DP (option)

The PROFIBUS functionality of this interface can be activated by configuring the sub module X1 '*MPI/DP*' of the CPU in the hardware configuration. ↪ *Chap. 11 'Option: Deployment PROFIBUS communication' page 254*



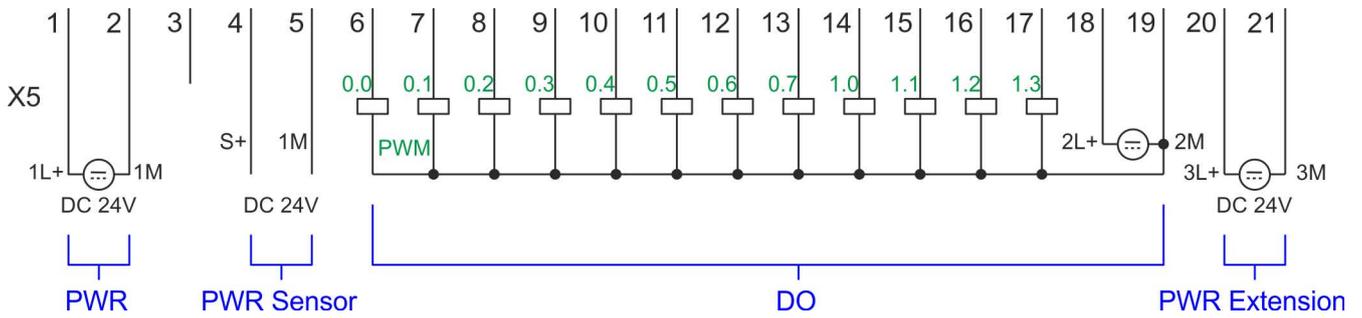
To switch the interface X3 MPI(PtP) to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is activated. ↪ Chap. 4.16 'Deployment storage media - VSD, VSC' page 111

X4: Connector

| X4 | Function | Type | Description |
|----|-----------|------|---|
| 1 | AI 0 | I | AI0: Analog input AI 0 |
| 2 | AI 1 | I | AI1: Analog input AI 1 |
| 3 | Analog 0V | I | 4M: GND for analog inputs |
| 4 | DI 0 | I | +0.0: Digital input DI 0 / Counter 0 (A) * |
| 5 | DI 1 | I | +0.1: Digital input DI 1 / Counter 0 (B) / Frequency 0 * |
| 6 | DI 2 | I | +0.2: Digital input DI 2 |
| 7 | DI 3 | I | +0.3: Digital input DI 3 / Counter 1 (A) * |
| 8 | DI 4 | I | +0.4: Digital input DI 4 / Counter 1 (B) / Frequency 1 * |
| 9 | DI 5 | I | +0.5: Digital input DI 5 |
| 10 | DI 6 | I | +0.6: Digital input DI 6 / Counter 2 (A) * |
| 11 | DI 7 | I | +0.7: Digital input DI 7 / Counter 2 (B) / Frequency 2 * |
| 12 | DI 8 | I | +1.0: Digital input DI 8 |
| 13 | DI 9 | I | +1.1: Digital input DI 9 / Counter 3 (A) * |
| 14 | DI 10 | I | +1.2: Digital input DI 10 / Counter 3 (B) / Frequency 3 * |
| 15 | DI 11 | I | +1.3: Digital input DI 11 / Gate 3 * |
| 16 | DI 12 | I | +1.4: Digital input DI 12 |
| 17 | DI 13 | I | +1.5: Digital input DI 13 |
| 18 | DI 14 | I | +1.6: Digital input DI 14 |
| 19 | DI 15 | I | +1.7: Digital input DI 15 / Latch 3 * |
| 20 | DC 24V | I | 5L+: DC 24V for onboard DI power section supply |
| 21 | 0 V | I | 5M: GND for onboard DI power section supply |

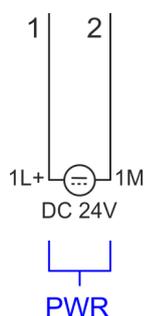
*) Max. input frequency 100kHz otherwise 1kHz.

X5: Connector



| X5 | Function | Type | Description |
|----|------------|------|--|
| 1 | Sys DC 24V | I | 1L+: DC 24V for electronic section supply |
| 2 | Sys 0V | I | 1M: GND for electronic section supply |
| 3 | --- | --- | reserved |
| 4 | DC 24V | O | S+: DC 24V for sensor |
| 5 | 0V | O | 1M: GND for sensor |
| 6 | DO 0 | O | +0.0: Digital output DO 0 / PWM 0 / Output channel counter 0 |
| 7 | DO 1 | O | +0.1: Digital output DO 1 / PWM 1 / Output channel counter 1 |
| 8 | DO 2 | O | +0.2: Digital output DO 2 / Output channel counter 2 |
| 9 | DO 3 | O | +0.3: Digital output DO 3 / Output channel counter 3 |
| 10 | DO 4 | O | +0.4: Digital output DO 4 |
| 11 | DO 5 | O | +0.5: Digital output DO 5 |
| 12 | DO 6 | O | +0.6: Digital output DO 6 |
| 13 | DO 7 | O | +0.7: Digital output DO 7 |
| 14 | DO 8 | O | +1.0: Digital output DO 8 |
| 15 | DO 9 | O | +1.1: Digital output DO 9 |
| 16 | DO 10 | O | +1.2: Digital output DO 10 |
| 17 | DO 11 | O | +1.3: Digital output DO 11 |
| 18 | DC 24V | I | 2L+: DC 24V for onboard DO power section supply |
| 19 | 0 V | I | 2M: GND for onboard DO power section supply / GND PWM |
| 20 | DC 24V | I | 3L+: DC 24V for SLIO bus power section supply |
| 21 | 0 V | I | 3M: GND for SLIO bus power section supply |

X5: Electronic power supply



The CPU has an integrated power supply. The power supply has to be provided with DC 24V. Via the power supply not only the internal electronic of the CPU is provided with voltage, but also the electronic from the integrated IO modules and the sensor output. The power supply is protected against polarity inversion and over current.

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 128kbyte
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 64kbyte
 - There is the possibility to extend the work memory to its maximum capacity 128kbyte by means of a VSC.

3.2.4 Slot for storage media

Overview

In this slot you can insert the following storage media:

- VSD - **VIPA SD-Card**
 - External memory card for programs and firmware.
- VSC - **VIPASetCard**
 - 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. ↪ *Chap. 4.16 'Deployment storage media - VSD, VSC' page 111*
 - To activate the corresponding card is to be installed and a *Overall reset* is to be established. ↪ *Chap. 4.13 'Overall reset' page 105*



To avoid malfunctions, you should use memory cards of VIPA. These correspond to the industrial standard. 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. ↪ *Chap. 4.20 'Diagnostic entries' page 117*

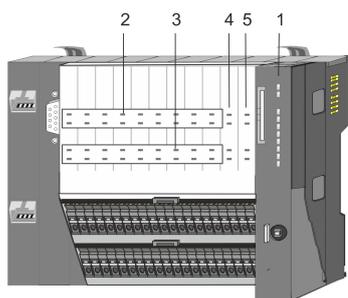
3.2.6 Operating mode switch

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



- 1 LEDs CPU part
- 2 DI +x: LEDs integrated DI (DI +0.0 ... DI +1.7)
- 3 DO +x: LEDs integrated DO (DO +0.0 ... DO +1.3)
- 4 xL+: LEDs power supply
- 5 xF: LEDs error

LEDs CPU

| LED | Color | Function |
|------|--|---|
| PW | ■ green | CPU - Power: The CPU is power supplied. |
| SF | ■ red | CPU - system fault: System error occurred. ↪ 51 |
| RN | ■ green | CPU - RUN: CPU is in RUN state. ↪ 51 |
| ST | ■ yellow | CPU - STOP: CPU is in STOP state. ↪ 51 |
| FC | ■ yellow | CPU - Forced: Variables are forced. ↪ 51 |
| SD | ■ yellow | CPU - SD memory card: Accessing the memory card. ↪ 51 |
| DE | ■ green | PROFIBUS - data exchange ↪ 52 |
| BF1 | ■ red | PROFIBUS - bus error ↪ 52 |
| L/A1 | ■ green | Ethernet PG/OP channel X1 - link/activity ↪ 52 |
| L/A2 | ■ green | Ethernet PG/OP channel X2 - link/activity ↪ 52 |

DI +x

| Digital input | LED | Description |
|---------------------|--|--|
| DI +0.0 ... DI +0.7 | ■ green | Digital input I+0.0 ... 0.7 has "1" signal |
| | <input type="checkbox"/> | Digital input I+0.0 ... 0.7 has "0" signal |
| DI +1.0 ... DI +1.7 | ■ green | Digital input I+1.0 ... 1.7 has "1" signal |
| | <input type="checkbox"/> | Digital input I+1.0 ... 1.7 has "0" signal |

DO +x

| Digital output | LED | Description |
|---------------------|--|---|
| DO +0.0 ... DO +0.7 | ■ green | Digital output Q+0.0 ... 0.7 has "1" signal |
| | <input type="checkbox"/> | Digital output Q+0.0 ... 0.7 has "0" signal |
| DO +1.0 ... DO +1.3 | ■ green | Digital output Q+1.0 ... 1.3 has "1" signal |
| | <input type="checkbox"/> | Digital output Q+1.0 ... 1.3 has "0" signal |

Structure > LEDs

xL+

| Power supply | LED  green | Description |
|--------------|--|--|
| 1L+ |  | DC 24V electronic section supply OK |
| | <input type="checkbox"/> | DC 24V electronic section supply not available |
| 2L+ |  | DC 24V power section supply outputs OK |
| | <input type="checkbox"/> | DC 24V power section supply outputs not available |
| 3L+ |  | DC 24V power section supply SLIO bus OK |
| | <input type="checkbox"/> | DC 24V power section supply SLIO bus not available |
| 5L+ |  | DC 24V power section supply inputs OK |
| | <input type="checkbox"/> | DC 24V power section supply inputs not available |

xF

| Error | LED  red | Description |
|-------|--|---|
| 1F |  | Error, overload respectively short circuit on power supply sensor |
| | <input type="checkbox"/> | no error |
| 2F |  | Error at overload respectively short circuit at the outputs |
| | <input type="checkbox"/> | no error |

LEDs CPU

| SF | RN | ST | FC | SD | Description |
|--|---|--|--|--|--|
|  red |  green |  yellow |  yellow |  yellow | |
| Boot-up after PowerON - as soon as the CPU is supplied with 5V, the  green PW-LED (Power) is on. | | | | | |
|  |  | X |  |  | Firmware is loaded, here the SF-LED flickers. |
|  |  |  |  |  | Initialization: Phase 1 |
|  |  |  |  |  | Initialization: Phase 2 |
|  |  |  |  |  | Initialization: Phase 3 |
|  |  |  |  |  | Initialization: Phase 4 |
| Operation | | | | | |
| X |  |  | X | X | CPU is in STOP state. |
| X |  2Hz |  | X | X | CPU is in start-up state. During the start-up (OB 100) the RUN LED blinks for at least 3s. |
| X |  |  10Hz | 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. ↪ <i>Chap. 4.20 'Diagnostic entries' page 117</i> |
| X | X | X |  | X | Variables are forced. |
| X | X | X | X |  | Accessing the memory card. |
| X | X |  10Hz | X | X | Configuration is loaded. |
| Overall reset | | | | | |
| X |  |  1Hz | X | X | Overall reset is requested. |
| X |  |  2Hz | X | X | Overall reset is executed. |
| X |  |  10Hz | X | X | Overall reset with none hardware configuration respectively hardware configuration from memory card. |
| Reset to factory setting | | | | | |
|  |  |  |  |  | Reset to factory setting is executed. |
|  |  |  |  |  | Reset to factory setting was 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 was finished without error. |
|  10Hz |  |  10Hz |  10Hz |  10Hz | Error during firmware update. |
| not relevant: X | | | | | |

LEDs Ethernet PG/OP channel

| L/A1 L/A2  green | Description |
|--|--|
|  | The corresponding Ethernet PG/OP channel is physically connected to the Ethernet. |
| <input type="checkbox"/> | The corresponding Ethernet PG/OP channel is not physically connected to the Ethernet. |
|  | The corresponding Ethernet PG/OP channel indicates Ethernet activity, here the LED flickers. |

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:

| DE  green | BF1  red | Description |
|--|--|--|
| Master operation | | |
| <input type="checkbox"/> | <input type="checkbox"/> | Master has no project, this means the interface is deactivated respectively the master configured without slaves with no errors. |
|  2Hz | <input type="checkbox"/> | CPU is in STOP state, the master is in "clear" state. All the slaves are in DE (data exchange) and the outputs of the slaves are disabled. |
|  | <input type="checkbox"/> | 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 at least 1 slave is in DE. |
| <input type="checkbox"/> |  | PROFIBUS is interrupted (no communication possible). |
| <input type="checkbox"/> |  2Hz | At least 1 slave is missing and no slave is in DE. |
| X |  2Hz | At least 1 slave is not in DE. |
| Slave operation | | |
| <input type="checkbox"/> | <input type="checkbox"/> | Slave has no configuration. |
| <input type="checkbox"/> |  | There is a bus error. |
|  2Hz | <input type="checkbox"/> | Slave exchanges data with the master. Slave CPU is in state STOP. |
|  | <input type="checkbox"/> | Slave exchanges data with the master. Slave CPU is in state RUN. |
| not relevant: X | | |

3.3 Technical data

| Order no. | 013-CCF0R00 |
|---|------------------------------------|
| Type | CPU 013C |
| 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) | 120 mA |
| Current consumption (rated value) | 360 mA |
| Inrush current | 3 A |
| I^2t | 0.1 A ² s |
| Max. current drain at backplane bus | 1 A |
| Max. current drain load supply | 6 A |
| Power loss | 7 W |
| Technical data digital inputs | |
| Number of inputs | 16 |
| Cable length, shielded | 1000 m |
| Cable length, unshielded | 600 m |
| Rated load voltage | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ |
| Current consumption from load voltage L+ (without load) | 25 mA |
| Rated value | DC 24 V |
| Input voltage for signal "0" | DC 0...5 V |
| Input voltage for signal "1" | DC 15...28.8 V |
| Input voltage hysteresis | - |
| Signal logic input | Sinking input |
| Frequency range | - |
| Input resistance | - |
| Input current for signal "1" | 3 mA |
| Connection of Two-Wire-BEROs possible | ✓ |
| Max. permissible BERO quiescent current | 0.5 mA |
| Input delay of "0" to "1" | 3 μ s – 15 ms / 0.5 ms – 15 ms |
| Input delay of "1" to "0" | 3 μ s – 15 ms / 0.5 ms – 15 ms |
| Number of simultaneously utilizable inputs horizontal configuration | 16 |
| Number of simultaneously utilizable inputs vertical configuration | 16 |

Technical data

| | |
|---|---------------------|
| Order no. | 013-CCF0R00 |
| Input characteristic curve | IEC 61131-2, type 1 |
| Initial data size | 16 Bit |
| Technical data digital outputs | |
| Number of outputs | 12 |
| Cable length, shielded | 1000 m |
| Cable length, unshielded | 600 m |
| Rated load voltage | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ |
| Current consumption from load voltage L+ (without load) | 20 mA |
| Total current per group, horizontal configuration, 40°C | 6 A |
| Total current per group, horizontal configuration, 60°C | 6 A |
| Total current per group, vertical configuration | 6 A |
| Output voltage signal "1" at min. current | L+ (-0.8 V) |
| Output voltage signal "1" at max. current | L+ (-0.8 V) |
| Output current at signal "1", rated value | 0.5 A |
| Signal logic output | Sourcing output |
| Output current, permitted range to 40°C | 5 mA to 0.6 A |
| Output current, permitted range to 60°C | 5 mA to 0.6 A |
| Output current at signal "0" max. (residual current) | 0.5 mA |
| Output delay of "0" to "1" | 2 µs / 30 µs |
| Output delay of "1" to "0" | 3 µs / 175 µs |
| Minimum load current | - |
| Lamp load | 10 W |
| Parallel switching of outputs for redundant control of a load | not possible |
| Parallel switching of outputs for increased power | not possible |
| Actuation of digital input | ✓ |
| Switching frequency with resistive load | max. 1000 Hz |
| Switching frequency with inductive load | max. 0.5 Hz |
| Switching frequency on lamp load | max. 10 Hz |
| Internal limitation of inductive shut-off voltage | L+ (-45 V) |
| Short-circuit protection of output | yes, electronic |
| Trigger level | 1 A |
| Number of operating cycle of relay outputs | - |
| Switching capacity of contacts | - |
| Output data size | 12 Bit |
| Technical data analog inputs | |

| Order no. | 013-CCF0R00 |
|---|---------------|
| Number of inputs | 2 |
| Cable length, shielded | 200 m |
| Rated load voltage | - |
| Reverse polarity protection of rated load voltage | - |
| Current consumption from load voltage L+ (without load) | - |
| Voltage inputs | ✓ |
| Min. input resistance (voltage range) | 100 kΩ |
| Input voltage ranges | 0 V ... +10 V |
| Operational limit of voltage ranges | +/-3.5% |
| Operational limit of voltage ranges with SFU | - |
| Basic error limit voltage ranges | +/-3.0% |
| Basic error limit voltage ranges with SFU | - |
| Destruction limit voltage | max. 30V |
| Current inputs | - |
| Max. input resistance (current range) | - |
| Input current ranges | - |
| Operational limit of current ranges | - |
| Operational limit of current ranges with SFU | - |
| Basic error limit current ranges | - |
| Radical error limit current ranges with SFU | - |
| Destruction limit current inputs (electrical current) | - |
| Destruction limit current inputs (voltage) | - |
| Resistance inputs | - |
| Resistance ranges | - |
| Operational limit of resistor ranges | - |
| Operational limit of resistor ranges with SFU | - |
| Basic error limit | - |
| Basic error limit with SFU | - |
| Destruction limit resistance inputs | - |
| Resistance thermometer inputs | - |
| Resistance thermometer ranges | - |
| Operational limit of resistance thermometer ranges | - |
| Operational limit of resistance thermometer ranges with SFU | - |
| Basic error limit thermoresistor ranges | - |
| Basic error limit thermoresistor ranges with SFU | - |

Technical data

| Order no. | 013-CCF0R00 |
|---|--------------------------|
| Destruction limit resistance thermometer inputs | - |
| Thermocouple inputs | - |
| Thermocouple ranges | - |
| Operational limit of thermocouple ranges | - |
| Operational limit of thermocouple ranges with SFU | - |
| Basic error limit thermoelement ranges | - |
| Basic error limit thermoelement ranges with SFU | - |
| Destruction limit thermocouple inputs | - |
| Programmable temperature compensation | - |
| External temperature compensation | - |
| Internal temperature compensation | - |
| Technical unit of temperature measurement | - |
| Resolution in bit | 12 |
| Measurement principle | successive approximation |
| Basic conversion time | 2 ms |
| Noise suppression for frequency | 40 dB |
| Initial data size | 4 Byte |
| Technical data analog outputs | |
| Number of outputs | - |
| Cable length, shielded | - |
| Rated load voltage | - |
| Reverse polarity protection of rated load voltage | - |
| Current consumption from load voltage L+ (without load) | - |
| Voltage output short-circuit protection | - |
| Voltage outputs | - |
| Min. load resistance (voltage range) | - |
| Max. capacitive load (current range) | - |
| Max. inductive load (current range) | - |
| Output voltage ranges | - |
| Operational limit of voltage ranges | - |
| Basic error limit voltage ranges with SFU | - |
| Destruction limit against external applied voltage | - |
| Current outputs | - |
| Max. in load resistance (current range) | - |
| Max. inductive load (current range) | - |
| Typ. open circuit voltage current output | - |

| Order no. | 013-CCF0R00 |
|--|--|
| Output current ranges | - |
| Operational limit of current ranges | - |
| Radical error limit current ranges with SFU | - |
| Destruction limit against external applied voltage | - |
| Settling time for ohmic load | - |
| Settling time for capacitive load | - |
| Settling time for inductive load | - |
| Resolution in bit | - |
| Conversion time | - |
| Substitute value can be applied | - |
| Output data size | - |
| Technical data counters | |
| Number of counters | 4 |
| Counter width | 32 Bit |
| Maximum input frequency | 100 kHz |
| Maximum count frequency | 400 kHz |
| Mode incremental encoder | ✓ |
| Mode pulse / direction | ✓ |
| Mode pulse | ✓ |
| Mode frequency counter | ✓ |
| Mode period measurement | ✓ |
| Gate input available | ✓ |
| Latch input available | ✓ |
| Reset input available | - |
| Counter output available | ✓ |
| Load and working memory | |
| Load memory, integrated | 128 KB |
| Load memory, maximum | 128 KB |
| Work memory, integrated | 64 KB |
| Work memory, maximal | 128 KB |
| Memory divided in 50% program / 50% data | ✓ |
| Memory card slot | SD/MMC-Card with max. 2 GB |
| Hardware configuration | |
| Racks, max. | 5 |
| Modules per rack, max. | total max. 64 minus number line extensions |
| Number of integrated DP master | - |

Technical data

| | |
|--|------------------------|
| Order no. | 013-CCF0R00 |
| Number of DP master via CP | - |
| Operable function modules | 64 |
| Operable communication modules PtP | 64 |
| Operable communication modules LAN | - |
| Status information, alarms, diagnostics | |
| Status display | yes |
| Interrupts | yes |
| Process alarm | yes |
| Diagnostic interrupt | yes |
| Diagnostic functions | yes, parameterizable |
| Diagnostics information read-out | possible |
| Supply voltage display | green LED |
| Group error display | red SF LED |
| Channel error display | red LED per group |
| Isolation | |
| Between channels | ✓ |
| Between channels of groups to | 16 |
| Between channels and backplane bus | ✓ |
| Between channels and power supply | - |
| Max. potential difference between circuits | DC 75 V/ AC 50 V |
| Max. potential difference between inputs (U _{cm}) | - |
| Max. potential difference between Mana and Mintern (U _{iso}) | - |
| Max. potential difference between inputs and Mana (U _{cm}) | - |
| Max. potential difference between inputs and Mintern (U _{iso}) | - |
| Max. potential difference between Mintern and outputs | - |
| Insulation tested with | DC 500 V |
| Command processing times | |
| Bit instructions, min. | 0.02 µs |
| Word instruction, min. | 0.02 µs |
| Double integer arithmetic, min. | 0.02 µs |
| Floating-point arithmetic, min. | 0.12 µs |
| Timers/Counters and their retentive characteristics | |
| Number of S7 counters | 512 |
| S7 counter remanence | adjustable 0 up to 256 |
| S7 counter remanence adjustable | C0 .. C7 |

| Order no. | 013-CCF0R00 |
|---|------------------------|
| Number of S7 times | 512 |
| S7 times remanence | adjustable 0 up to 256 |
| 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 256 |
| Bit memories retentive characteristic preset | MB0 .. MB15 |
| Number of data blocks | 1024 |
| Max. data blocks size | 64 KB |
| Max. local data size per execution level | 4096 Byte |
| Blocks | |
| Number of OBs | 22 |
| Number of FBs | 1024 |
| Number of FCs | 1024 |
| Maximum nesting depth per priority class | 16 |
| Maximum nesting depth additional within an error OB | 4 |
| Time | |
| Real-time clock buffered | ✓ |
| Clock buffered period (min.) | 30 d |
| Accuracy (max. deviation per day) | 10 s |
| Number of operating hours counter | 8 |
| Clock synchronization | ✓ |
| Synchronization via MPI | Master/Slave |
| Synchronization via Ethernet (NTP) | no |
| Address areas (I/O) | |
| Input I/O address area | 2048 Byte |
| Output I/O address area | 2048 Byte |
| Input process image maximal | 2048 Byte |
| Output process image maximal | 2048 Byte |
| Digital inputs | 16224 |
| Digital outputs | 16256 |
| Digital inputs central | 528 |
| Digital outputs central | 524 |
| Integrated digital inputs | 16 |
| Integrated digital outputs | 12 |
| Analog inputs | 1015 |

Technical data

| | |
|---|---|
| Order no. | 013-CCF0R00 |
| Analog outputs | 1015 |
| Analog inputs, central | 514 |
| Analog outputs, central | 256 |
| Integrated analog inputs | 2 |
| Integrated analog outputs | - |
| Technical data encoder supply | |
| Number of outputs | 1 |
| Output voltage (typ) | L+ (-1.5 V) |
| Output voltage (rated value) | 300 mA |
| Short-circuit protection | yes, electronic |
| Binding of potential | Power supply of PLC |
| Communication functions | |
| PG/OP channel | ✓ |
| Global data communication | ✓ |
| Number of GD circuits, max. | 8 |
| Size of GD packets, max. | 54 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 |
| PWM data | |
| PWM channels | 2 |
| PWM time basis | 1 µs / 0.1 ms / 1 ms |
| Period length | 50µs...65.535ms / 0.1...87ms / 1...87ms |
| Minimum pulse width | 0...0.5 * Period duration |
| Type of output | Highside |
| Functionality Sub-D interfaces | |
| Type | X3 |
| Type of interface | RS485 |
| Connector | Sub-D, 9-pin, female |
| Electrically isolated | ✓ |
| MPI | ✓ |
| MP ² I (MPI/RS232) | - |

| Order no. | 013-CCF0R00 |
|--------------------------------------|--------------------------|
| 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 |
| Type | - |
| Type of interface | - |
| Connector | - |
| Electrically isolated | - |
| MPI | - |
| MP ² I (MPI/RS232) | - |
| DP master | - |
| DP slave | - |
| Point-to-point interface | - |
| 5V DC Power supply | - |
| 24V DC Power supply | - |
| Functionality MPI | |
| Number of connections, max. | 32 |
| PG/OP channel | ✓ |
| Routing | ✓ |
| 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 | |
| Number of connections, max. | 32 |
| PG/OP channel | ✓ |
| Routing | ✓ |
| S7 basic communication | ✓ |
| S7 communication | ✓ |
| S7 communication as server | ✓ |
| S7 communication as client | - |

Technical data

| Order no. | 013-CCF0R00 |
|---|-----------------------------|
| 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. | 32 |
| 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 | |
| Number of connections, max. | 32 |
| 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 |
| Transfer memory outputs, max. | 244 Byte |
| Address areas, max. | 32 |
| User data per address area, max. | 32 Byte |
| Functionality RJ45 interfaces | |
| Type | X1/X2 |
| Type of interface | Ethernet 10/100 MBit Switch |
| Connector | 2 x RJ45 |
| Electrically isolated | ✓ |
| PG/OP channel | ✓ |
| Number of connections, max. | 4 |
| Productive connections | ✓ |
| Fieldbus | - |
| | |

| Order no. | 013-CCF0R00 |
|---|----------------------|
| Type | - |
| Type of interface | - |
| Connector | - |
| Electrically isolated | - |
| PG/OP channel | - |
| Number of connections, max. | - |
| Productive connections | - |
| Fieldbus | - |
| Point-to-point communication | |
| PtP communication | ✓ |
| Interface isolated | ✓ |
| RS232 interface | - |
| RS422 interface | - |
| RS485 interface | ✓ |
| Connector | Sub-D, 9-pin, female |
| Transmission speed, min. | 1200 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 | - |
| Properties PROFINET I/O-Controller via PG/OP | |
| Realtime Class | - |
| Conformance Class | PROFINET IO |
| Number of PN IO devices | 8 |
| IRT support | - |
| Shared Device supported | ✓ |
| MRP Client supported | ✓ |
| Prioritized start-up | - |
| Number of PN IO lines | 1 |

Technical data

| | |
|--|---|
| Order no. | 013-CCF0R00 |
| Address range inputs, max. | 2 KB |
| Address range outputs, max. | 2 KB |
| Transmitting clock | 1 ms |
| Update time | 1 ms .. 512 ms |
| Isochronous mode | - |
| Parallel operation as controller and I-Device | ✓ |
| Properties PROFINET I-Device via PG/OP | |
| I/O Data range, max. | 768 Byte |
| Update time | 1 ms .. 512 ms |
| Mode as Shared I-Device | - |
| Management & diagnosis via PG/OP | |
| Protocols | ICMP DCP LLDP / SNMP NTP |
| Web based diagnosis | ✓ |
| NCM diagnosis | - |
| Ethernet communication via PG/OP | |
| Number of productive connections via PG/OP, max. | 2 |
| Number of productive connections by Siemens NetPro, max. | 2 |
| S7 connections | BSEND, BRCV, GET, PUT, Connection of active and passive data handling |
| User data per S7 connection, max. | 64 KB |
| TCP-connections | FETCH PASSIV, WRITE PASSIV, Connection of passive data handling |
| User data per TCP connection, max. | 8 KB |
| ISO on TCP connections (RFC 1006) | FETCH PASSIV, WRITE PASSIV, Connection of passive data handling |
| User data per ISO connection, max. | 8 KB |
| Ethernet open communication via PG/OP | |
| Number of configurable connections, max. | 2 |
| ISO on TCP connections (RFC 1006) | TSEND, TRCV, TCON, TDISCON |
| User data per ISO on TCP connection, max. | 32 KB |
| TCP-Connections native | TSEND, TRCV, TCON, TDISCON |
| User data per native TCP connection, max. | 32 KB |
| User data per ad hoc TCP connection, max. | 1460 Byte |
| UDP-connections | TUSEND, TURCV |

| | |
|--|---|
| Order no. | 013-CCF0R00 |
| User data per UDP connection, max. | 1472 Byte |
| WebVisu via PG/OP | |
| WebVisu is supported | ✓ |
| Max. number of connections WebVisu | 4 |
| WebVisu supports HTTP | ✓ |
| WebVisu supports HTTPS | ✓ |
| OPC UA server via PG/OP | |
| OPC UA server is supported | ✓ |
| Max. number of connections per interface | 4 |
| Services | Data Access (Read, Write, Subscribe) |
| Security policies | None, Basic128Rsa15, Basic256, Basic256Sha256 |
| Authentication | Anonymous, username and password |
| Housing | |
| Material | PPE / PPE GF10 |
| Mounting | Profile rail 35 mm |
| Mechanical data | |
| Dimensions (WxHxD) | 147 mm x 100 mm x 83 mm |
| Net weight | 320 g |
| Weight including accessories | 320 g |
| Gross weight | 355 g |
| Environmental conditions | |
| Operating temperature | 0 °C to 60 °C |
| Storage temperature | -25 °C to 70 °C |
| Certifications | |
| UL certification | yes |
| KC certification | yes |

4 Deployment CPU 013-CCF0R00

4.1 Please note!



The following descriptions always refer to the usage in the Siemens SIMATIC Manager. Information on usage in the VIPA SPEED7 Studio or Siemens TIA Portal can be found here:

↳ Chap. 12 'Configuration with VIPA SPEED7 Studio' page 267

↳ Chap. 13 'Configuration with TIA Portal' page 294

4.2 Assembly



Information about assembly and cabling ↳ Chap. 2 'Basics and mounting' page 12.

4.3 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 *.pkb 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. ↳ Chap. 4.14 'Firmware update' page 106
- The CPU checks if a previously activated VSC is inserted. If not, the SF 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. ↳ Chap. 4.20 'Diagnostic entries' page 117

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.

4.4 Addressing

4.4.1 Overview

To provide specific addressing of the installed peripheral 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 peripheral addresses for digital in-/output modules starting with 0 and analog modules are assigned to even addresses starting with 256.

4.4.2 Default address assignment of the I/O part

| Sub module | Input address | Access | Description |
|------------|---------------|--------|-----------------------------|
| AI5/AO2 | 800 | WORD | Analog input channel 0 (X4) |
| | 802 | WORD | Analog input channel 1 (X4) |

| Sub module | Input address | Access | Description |
|------------|---------------|--------|------------------------------------|
| DI24/DO16 | 136 | BYTE | Digital input I+0.0 ... I+0.7 (X4) |
| | 137 | BYTE | Digital input I+1.0 ... I+1.7 (X4) |

| Sub module | Input address | Access | Description |
|------------|---------------|--------|--|
| Counter | 816 | DINT | Channel 0: Counter value / Frequency value |
| | 820 | DINT | Channel 1: Counter value / Frequency value |
| | 824 | DINT | Channel 2: Counter value / Frequency value |
| | 828 | DINT | Channel 3: Counter value / Frequency value |

| Sub module | Output address | Access | Description |
|------------|----------------|--------|-------------------------------------|
| DI24/DO16 | 136 | BYTE | Digital output Q+0.0 ... Q+0.7 (X5) |
| | 137 | BYTE | Digital output Q+1.0 ... Q+1.3 (X5) |

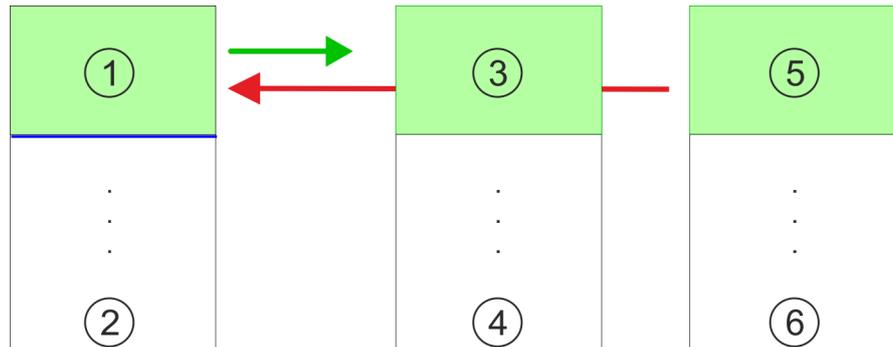
| Sub module | Output address | Access | Description |
|------------|----------------|--------|-------------|
| Counter | 816 | DWORD | reserved |
| | 820 | DWORD | reserved |
| | 824 | DWORD | reserved |
| | 828 | DWORD | reserved |

4.4.3 Addressing periphery modules

The CPU 013-CCF0R00 provides an periphery area (address 0 ... 2047) and a process image of the in- and outputs (each address default 0 ... 127). The size of the process image can be adjusted via the parametrization. ↪ *Chap. 4.8 'Setting standard CPU parameters' page 78*

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

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



- 1 Periphery area: 0 ... 127 (default)
- 2 Periphery area : 0 ... 2047
- 3 Process image of the inputs (PII): 0 ... 127
- 4 Process image of the inputs (PII): max. 2047
- 5 Process image of the outputs (PIQ): 0 ... 127
- 6 Process image of the outputs (PIQ): max. 2047

Max. number of 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 respectively write accesses to the peripheral 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.

4.5 Hardware configuration - CPU

Precondition

- The configuration of the CPU takes place at the *'hardware configurator'* of the Siemens SIMATIC Manager V 5.5 SP2 and up.
- The configuration of the System SLIO CPU happens by means of a virtual PROFINET IO device *'VIPA SLIO System'*. 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!

Installation IO device 'VIPA SLIO System'

The installation of the PROFINET IO devices *'VIPA SLIO System'* 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'*

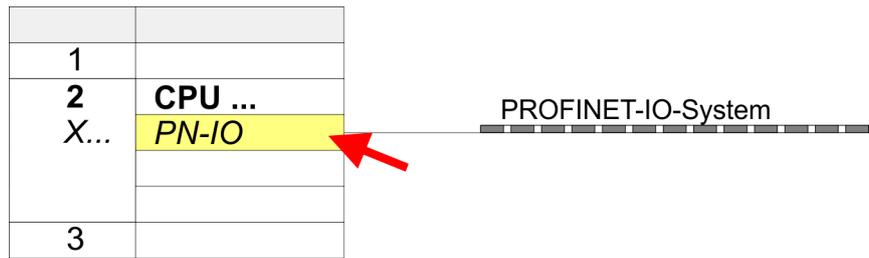
Proceeding

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 314C-2 PN/DP (314-6EH04-0AB0 V3.3).

| Slot | Module |
|----------|------------------------|
| 1 | |
| 2 | CPU 314C-2PN/DP |
| X1 | MPI/DP |
| X2 | PN-IO |
| X2... | Port 1 |
| X2... | Port 2 |
| ... | ... |
| 3 | |

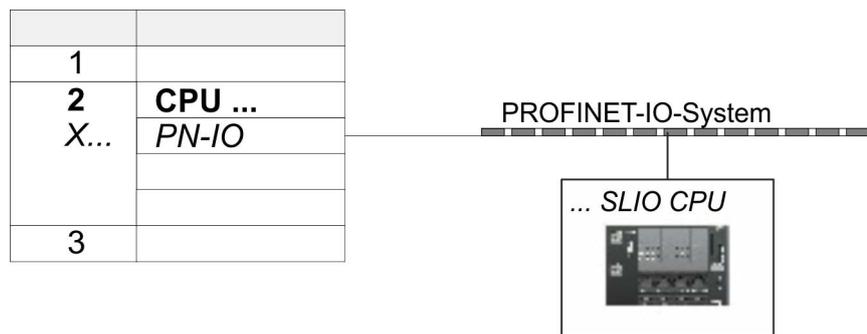
4. ➤ Click at the sub module *'PN-IO'* of the CPU.
5. ➤ Select *'Context menu → Insert PROFINET IO System'*.



- 6. Use [New] to create a new subnet and assign valid IP address data for your PROFINET system.

i With firmware version V2.4.0 and up, you can access the Ethernet PG/OP channel via this IP address data. The configuration via an additional CP is no longer required, but still possible. ↪ Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 72

- 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.



↘

| | | | |
|-----|-------------------------|--------------------|--|
| 0 | ... SLIO CPU ... | 013-CCF0R00 | |
| X2 | 013-CCF0R00 | | |
| 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 '013-CCF0R00 CPU' to your PROFINET system.
 - ⇒ In the slot overview of the PROFINET IO device 'VIPA SLIO System' the CPU is already placed at slot 0. From slot 1 you can place your System SLIO modules.

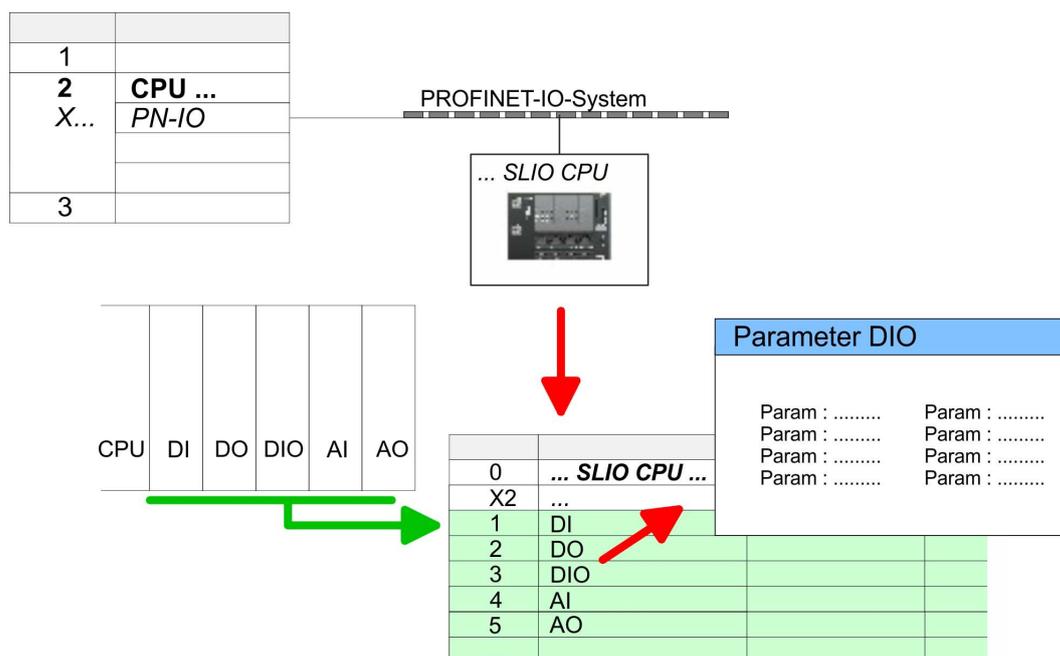
4.6 Hardware configuration - System SLIO modules

System SLIO backplane bus

To connect System SLIO modules, the CPU has a backplane bus, which must additionally to be supplied. Here up to 64 System SLIO modules can be connected.

Proceeding

1. ➔ Perform, if not already done, a hardware configuration for the CPU. ↪ *Chap. 4.5 'Hardware configuration - CPU' page 69*
2. ➔ Starting with slot 1 place in the slot overview of the PROFINET IO device 'VIPA SLIO System' your System SLIO modules in the plugged sequence.
3. ➔ Parametrize if necessary the modules and assign valid addresses, so that they can directly be addressed.



4.7 Hardware configuration - Ethernet PG/OP channel

Overview



Please note!

- *At the first commissioning respectively after a reset to factory setting the Ethernet interface has no IP address.*
- *For online access, you have to assign valid IP address data to it by means of "Initialization".*
- *After initialization, you can transfer the IP address data to your project.*

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/X2) is designed as switch. This enables PG/OP communication via the connections X1 and X2.
- Configurable connections are possible.
- DHCP respectively the assignment of the network configuration with a DHCP server is supported.
- Default diagnostics addresses: 2025 ... 2040
- Via the Ethernet PG/OP channel, you have access to:
 - Device website, where you can find information on firmware status, connected peripherals, current cycle times, etc.
 - *OPC UA* project, which is to be created in the *OPC UA Configurator*.
 - *WebVisu* project, which is to be created in the *SPEED7 Studio*.
 - PROFINET IO controller or the PROFINET I-Device.

Assembly and commissioning

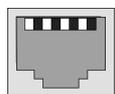
1. ➤ Install your System SLIO with your CPU.
2. ➤ Wire the system by connecting cables for voltage supply and signals.
3. ➤ Connect the one of the Ethernet jacks (X1, X2) of the Ethernet PG/OP channel to Ethernet.
4. ➤ Switch on the power supply.
 - ⇒ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization" via PLC functions

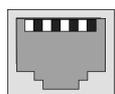
The initialization via PLC functions takes place with the following proceeding:

- ➔ Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found at the front of your CPU with the name "MAC PG/OP: ...".

X1 PG/OP



X2 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 5.5 & SP2 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.

4.7.1 Take IP address parameters in project**2 variants for configuration**

From firmware version V2.4.0 and up, you have the following options for configuring the Ethernet PG/OP channel:

- Configuration via integrated CPU interface (firmware version V2.4.0 and up only).
- Configuration via additional CP (all firmware versions).

4.7.1.1 Configuration via integrated CPU interface

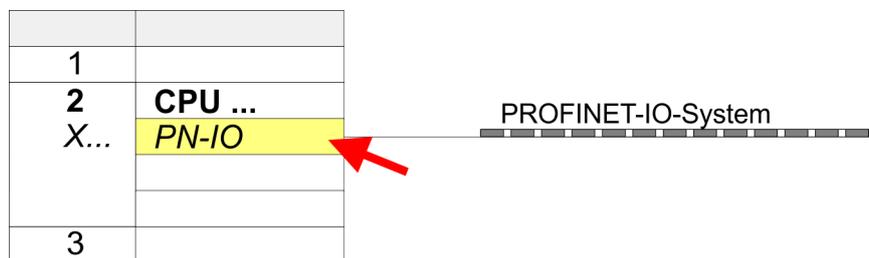
Proceeding

From firmware version V2.4.0 this variant for configuration is recommended. The following advantages result:

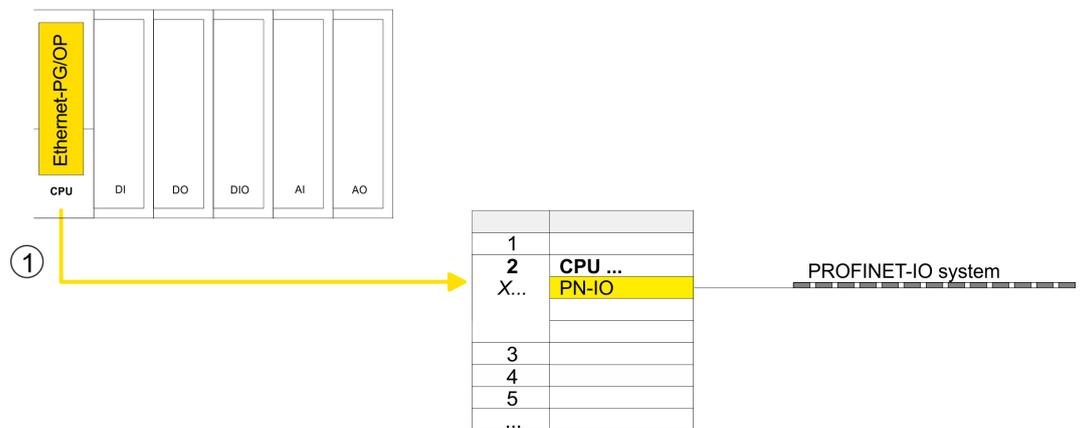
- The configuration becomes clearer, because the periphery modules and the PROFINET IO devices are configured on the PROFINET line of the CPU and no additional CP is to be configured.
- There are no address collisions, because the S7 addresses for all components are assigned from the address area of the CPU.

Unless during the hardware configuration of the CPU  69 there was no IP address data assigned yet or these are to be changed, the configuration happens to the following proceeding, otherwise the Ethernet PG/OP channel is configured.

1. ➤ Open the Siemens hardware configurator and, if not already done, configure the Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
2. ➤ Open the PROFINET Properties dialog box of the CPU by double-clicking 'PN-IO'.



3. ➤ Click at 'General'.
4. ➤ At 'Properties', enter the previously assigned IP address data and a subnet. The IP address data are not accepted without subnet assignment!
5. ➤ Transfer your project.

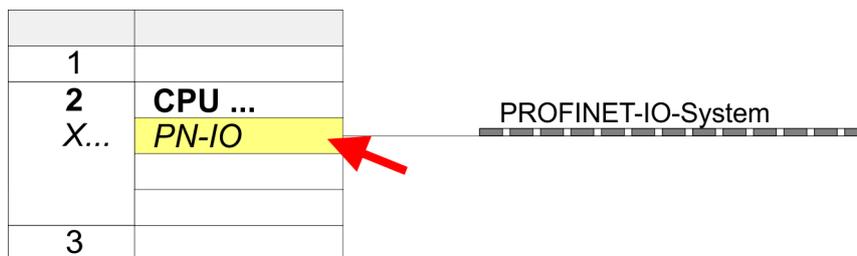


1 Ethernet PG/OP channel

4.7.1.1.1 Time-of-day synchronization

NTP method

In the NTP mode (**N**etwork **T**ime **P**rotocol) the module sends as client time-of-day queries at regular intervals to all configured NTP servers within the sub net. You can define up to 4 NTP server. Based on the response from the servers, the most reliable and most exact time-of-day is determined. Here the time with the lowest *stratum* is used. *Stratum 0* is the time standard (atomic clock). *Stratum 1* are directly linked to this NTP server. Using the NTP method, clocks can be synchronized over subnet boundaries. The configuration of the NTP servers is carried out in the Siemens SIMATIC Manager via the CP, which is already configured.



1. Open the Properties dialog by double-clicking 'PN-IO'.
2. Select the tab 'Time-of-day synchronization'.
3. Activate the NTP method by enabling 'Activate NTP time-of-day synchronization'.
4. Click at [Add] and add the corresponding NTP server.
5. Set the 'Update interval' you want. Within this interval, the time of the module is synchronized once.
6. Close the dialog with [OK].
7. Save and transfer your project to the CPU.
 - ⇒ After transmission, the NTP time is requested by each configured time server and the best response for the time synchronization is used.



Please note that although the time zone is evaluated, an automatic changeover from winter to summer time is not supported. Industrial systems with time-of-day synchronization should always be set in accordance to the winter time.

With the FC 61 you can determine the local time in the CPU. More information about the usage of this block may be found in the manual "SPEED7 Operation List" from VIPA.

4.7.1.2 Configuration via additional CP

Proceeding

This is the conventional variant for configuration and is supported by all firmware versions. If possible, always use the configuration via the internal interface, otherwise the following disadvantages result:

- Address overlaps are not recognized in the Siemens SIMATIC Manager.
- For PROFINET devices only the address range 0 ... 1023 is available.
- The addresses of the PROFINET devices are not checked with the address range of the CPU by the Siemens SIMATIC Manager for address overlaps.

The configuration happens according to the following procedure:

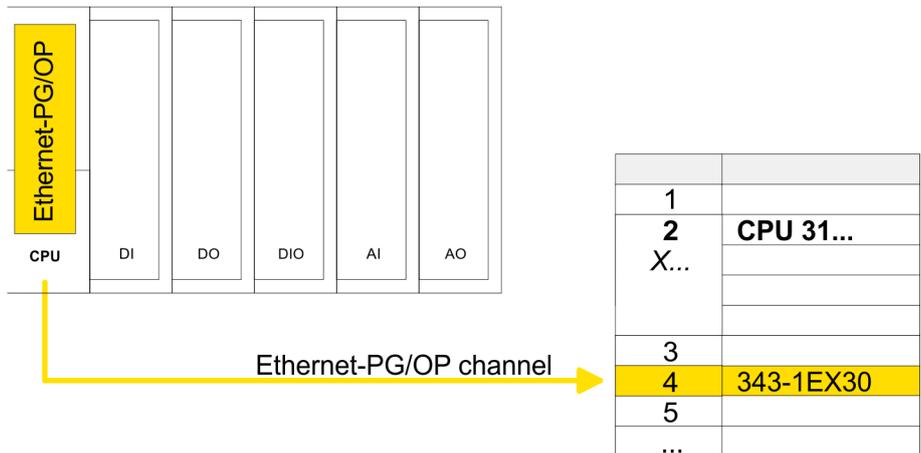
1. ➤ Open the Siemens hardware configurator and, if not already done, configure the Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
2. ➤ Place for the Ethernet PG/OP channel at slot 4 the Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).



CAUTION!

Please configure the diagnostic addresses of the CP343-1EX30 for 'PN-IO', 'Port1' and 'Port2' so that no overlaps occur in the periphery input area. Otherwise your CPU can not start-up and you receive the diagnostic entry 0xE904. These addresses overlaps are not recognized by the Siemens SIMATIC Manager.

3. ➤ Open the Properties dialog by double-clicking on 'PN-IO' of the CP 343-1EX30 and enter the previously assigned IP address data and a subnet for the CP at 'Properties'. The IP address data are not accepted without subnet assignment!
4. ➤ Transfer your project.



1 Ethernet PG/OP channel

4.7.1.2.1 Time-of-day synchronization

NTP method

In the NTP mode (**N**etwork **T**ime **P**rotocol) the module sends as client time-of-day queries at regular intervals to all configured NTP servers within the sub net. You can define up to 4 NTP server. Based on the response from the servers, the most reliable and most exact time-of-day is determined. Here the time with the lowest *stratum* is used. *Stratum 0* is the time standard (atomic clock). *Stratum 1* are directly linked to this NTP server. Using the NTP method, clocks can be synchronized over subnet boundaries. The configuration of the NTP servers is carried out in the Siemens SIMATIC Manager via the CP, which is already configured.

| | |
|------|-----------|
| 1 | |
| 2 | CPU 31... |
| X... | |
| | |
| 3 | |
| 4 | 343-1EX30 |
| 5 | |
| ... | |

1. ➤ Open the properties dialog via double-click on the CP 343-1EX30.
2. ➤ Select the register *'Time-of-day synchronization'*.
3. ➤ Activate the NTP method by enabling *'Activate NTP time-of-day synchronization'*.
4. ➤ Click at [Add] and add the corresponding NTP server.
5. ➤ Select your *'time zone'*. In the NTP method, UTC (**U**niversal **T**ime **C**oordinated) is generally transmitted; this corresponds to GMT (**G**reenwich **M**ean **T**ime). By configuring the local time zone, you can set a time offset to UTC.
6. ➤ Select your *'Update interval'*. Within this interval the time of the module is synchronized once.
7. ➤ Close the dialog with [OK].
8. ➤ Save and transfer your project to the CPU.
 - ⇒ After transmission, the NTP time is requested by each configured time server and the best response for the time synchronization is used.



Please note that although the time zone is evaluated, an automatic changeover from winter to summer time is not supported. Industrial systems with time-of-day synchronization should always be set in accordance to the winter time.

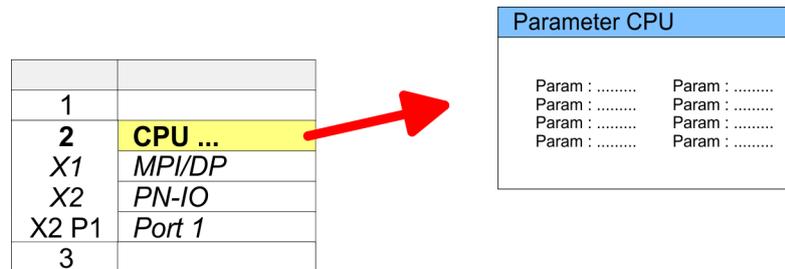
The FC 61 can be used to determine the local time in the CPU. More information about the usage of this block may be found in the manual "SPEED7 Operation List" from VIPA.

4.8 Setting standard CPU parameters

4.8.1 Parameterization via Siemens CPU

Parameterization via Siemens CPU 314C-2 PN/DP

Since the CPU from VIPA is to be configured as Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) in the Siemens hardware configurator, the standard parameters of the VIPA CPU may be set with "Object properties" of the CPU 314C-2 PN/DP during hardware configuration. Via a double-click on the CPU 314C-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.



4.8.2 Parameter CPU

Supported parameters

The CPU does not evaluate each parameter, which may be set at the hardware configuration. The parameters of the following registers are not supported: Synchronous cycle interrupts, communication and web. The following parameters are currently supported:

General

- Short description
 - The short description of the Siemens CPU is CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
- Order No. / Firmware
 - Order number and firmware are identical to the details in the "hardware catalog" window.
- Name
 - The Name field provides the short description of the CPU.
 - If you change the name the new name appears in the Siemens SIMATIC Manager.
- 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 81346-1.
- Location designation
 - The location designation is part of the resource designation.
 - Here the exact location of your module within a plant may be specified.
- Comment
 - In this field information about the module may be entered.

Startup

- Startup when expected/actual configuration differs
 - If the checkbox for '*Startup when expected/actual configuration differ*' is deselected and at least one module is not located at its configured slot or if another type of module is inserted there instead, then the CPU does not switch to RUN mode and remains in STOP mode.
 - If the checkbox for '*Startup when expected/actual configuration differ*' is selected, then the CPU starts even if there are modules not located in their configured slots or if another type of module is inserted there instead, such as during an initial system start-up.
- Monitoring time for ready message by modules [100ms]
 - This operation specifies the maximum time for the ready message of every configured module after PowerON.
 - Here connected PROFIBUS DP slaves are also considered until they are parameterized.
 - If the modules do not send a ready message to the CPU by the time the monitoring time has expired, the actual configuration becomes unequal to the preset configuration.
- Monitoring time for transfer of parameters to modules [100ms]
 - The maximum time for the transfer of parameters to parameterizable modules.
 - Here connected PROFINET IO devices also considered until they are parameterized.
 - If not every module has been assigned parameters by the time this monitoring time has expired; the actual configuration becomes unequal to the preset configuration.

Cycle / Clock memory

- Update OB1 process image cyclically
 - This parameter is not relevant.
- Scan cycle monitoring time
 - Here the scan cycle monitoring time in milliseconds may be set.
 - If the scan cycle time exceeds the scan cycle monitoring time, the CPU enters the STOP mode.
 - Possible reasons for exceeding the time are:
 - Communication processes
 - a series of interrupt events
 - an error in the CPU program
- Minimum scan cycle time
 - This parameter is not relevant.
- Scan cycle load from Communication
 - Using this parameter you can control the duration of communication processes, which always extend the scan cycle time so it does not exceed a specified length.
 - If the cycle load from communication is set to 50%, the scan cycle time of OB 1 can be doubled. At the same time, the scan cycle time of OB 1 is still being influenced by asynchronous events (e.g. hardware interrupts) as well.
- Size of the process image input/output area
 - Here the size of the process image max. 2048 for the input/output periphery may be fixed (default: 256).
- OB85 call up at I/O access error
 - The preset reaction of the CPU may be changed to an I/O access error that occurs during the update of the process image by the system.
 - The VIPA CPU is preset such that OB 85 is not called if an I/O access error occurs and no entry is made in the diagnostic buffer either.
- Clock memory
 - Activate the check box if you want to use clock memory and enter the number of the memory byte.



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

Retentive Memory

- Number of Memory bytes from MB0
 - Enter the number of retentive memory bytes from memory byte 0 onwards.
- Number of S7 Timers from T0
 - Enter the number of retentive S7 timers from T0 onwards. Each S7 timer occupies 2bytes.
- Number of S7 Counters from C0
 - Enter the number of retentive S7 counter from C0 onwards.
- Areas
 - This parameter is not supported.

Interrupts

- Priority
 - Here the priorities are displayed, according to which the hardware interrupt OBs are processed (hardware interrupt, time-delay interrupt, async. error interrupts).

Time-of-day interrupts

- Priority
 - This value is fixed to 2.
- Active
 - By enabling 'Active' the time-of-day interrupt function is enabled.
- Execution
 - Select how often the interrupts are to be triggered.
 - Intervals ranging from every minute to yearly are available. The intervals apply to the settings made for *start date* and *time*.
- Start date/time
 - Enter date and time of the first execution of the time-of-day interrupt.
- Process image partition
 - This parameter is not supported.

Cyclic interrupts

- Priority
 - Here the priorities may be specified according to which the corresponding cyclic interrupt is processed.
- Execution
 - Enter the time intervals in ms, in which the watchdog interrupt OBs should be processed.
 - The start time for the clock is when the operating mode switch is moved from STOP to RUN.
- Phase offset
 - Enter the delay time in ms for current execution for the watch dog interrupt. This should be performed if several watchdog interrupts are enabled.
 - Phase offset allows to distribute processing time for watchdog interrupts across the cycle.
- Process image partition
 - This parameter is not supported.

Diagnostics/Clock

- Report cause of STOP
 - Activate this parameter, if the CPU should report the cause of STOP to PG respectively OP on transition to STOP.
- Number of messages in the diagnostics buffer
 - This parameter is ignored. The CPU always has a diagnostics buffer (circular buffer) for 100 diagnostics messages.
- Synchronization type
 - Here you specify whether clock should synchronize 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 is to be carried out.
- Correction factor
 - Lose or gain in the clock time may be compensated within a 24 hour period by means of the correction factor in ms.
 - If the clock is 1s slow after 24 hours, you have to specify a correction factor of "+1000" ms.

Protection

- Level of protection
 - Here 1 of 3 protection levels may be set to protect the CPU from unauthorized access.
 - *Protection level 1 (default setting):*
No password adjustable, no restrictions
 - *Protection level 2 with password:*
Authorized users: read and write access
Unauthorized user: read access only
 - *Protection level 3:*
Authorized users: read and write access
Unauthorized user: no read and write access

4.8.3 Parameter for MPI/DP

The properties dialog of the MPI(PtP) interface X3 is opened via a double click to the sub module MPI/DP



To switch the interface X3 MPI(PtP) 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. ↗ Chap. 4.16 'Deployment storage media - VSD, VSC' page 111

General

- Short description
 - Here the short description "MPI/DP" for the interface is specified.
- Name
 - At *Name* "MPI/DP" is shown. If you change the name, the new name appears in the Siemens SIMATIC Manager.
- Type
 - Here you can choose between the functionalities MPI and PROFIBUS.
- Interface
 - Here the MPI respectively PROFIBUS address is shown.

Setting VIPA specific CPU parameters

- Properties
 - With this button the properties of the interface may be pre-set.
- Comment
 - You can enter the purpose of the interface.

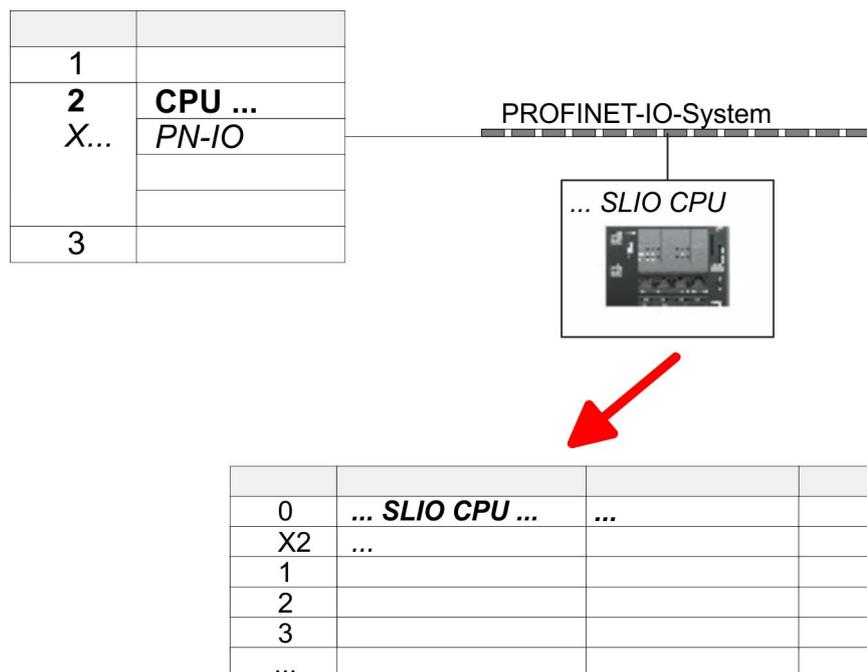
Address

- Diagnostics
 - A diagnostics address for the interface is to be pre-set here. In the case of an error the CPU is informed via this address.
- Operating mode
 - With the interface type 'PROFIBUS' here you can pre-set the 'Operating mode' DP Slave.
- Configuration, Clock
 - These parameters are not supported.

4.9 Setting VIPA specific CPU parameters

Overview

Except of the VIPA specific CPU parameters the CPU parametrization takes place in the parameter dialog of the CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) 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 System' 013-CCF0R00 CPU the properties dialog is opened.



Here the following parameters may be accessed:

- Diagnostic
 - Diagnostic interrupt in case of short circuit or overload of the power supplies
- MPI - function X3
 - Functionality of X3
- Retentive data
 - Additional retentive memory, timer respectively counter
- OB
 - Priority OB 57
 - Call OB 80 on cyclic interrupt

- Miscellaneous
 - Direct DX transition
 - PN MultipleWrite
 - Free Module Mapping ↗ 85
 - Reduced PDU size
- Access settings ↗ 90
 - CPU
 - Activation of interfaces and ports
 - Activation of protocols

Parameter

The following parameters may be accessed by means of the properties dialog of the CPU.

Diagnostics interrupt (default: deactivated)

- Diagnostics interrupt 5L+ (DC 24V DI power section supply)
- Diagnostics interrupt 2L+ (DC 24V DO power section supply)
- Diagnostics interrupt 3L+ (DC 24V System SLIO bus power section supply)
- Diagnostics interrupt DO short circuit/overload
- Diagnostics interrupt sensor short circuit/overload

MPI

- Function X3
 - Functionality of the MPI(PtP) interface X3
 - MPI/DP (default): In this operating mode parameters are active, which you set on sub module 'MPI/DP' of the Siemens CPU 314C-2 PN/DP. ↗ Chap. 11 'Option: Deployment PROFIBUS communication' page 254
 - PTP: 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. ↗ Chap. 6 'Deployment PtP communication' page 170

Retentive data

- Additional retentive memory
 - Here enter the number of retentive memory bytes. With 0 the value 'Retentive memory → Number of memory bytes starting with MB0' is set, which is pre-set at the parameters of the Siemens CPU.
 - Range of values: 0 (default) ... 8192
- Additional retentive timer
 - Enter the number of S7 timers. With 0 the value 'Retentive memory → Number S7 timers starting with T0' is set, which is pre-set at the parameters of the Siemens CPU.
 - Range of values: 0 (default) ... 512
- Additional retentive counter
 - Enter the number of S7 counter. With 0 the value 'Retentive memory → Number S7 counters starting with C0' is set, which is pre-set at the parameters of the Siemens CPU.
 - Range of values: 0 (default) ... 512

OB

- Priority OB 57
 - Here you can set the priority for the OB 57.
 - Range of values: 2 (default) ... 24
- OB 80 for cyclic interrupt error
 - Here you can set for which cyclic interrupt OB 80 (time error) should be called.
 - Range of values: Deactivated (default), selection of the corresponding OB

Miscellaneous

- Direct DX transition - if this parameter is activated, the integrated PROFIBUS DP master, if activated by VSC, shows the following behavior:
 - As long as a DP slave is in Data Exchange, i.e. in the DP slave standard diagnostic data byte 0, bit 1 and byte 1, bit 0 have the status 0, this DP slave is directly taken from the DP master in Data Exchange. The transition happens without a *SetPrm*- and *CheckConfig* telegram was being sent to the DP Slave.
 - When a DP slave is taken to Data Exchange, the outputs remain active and are not deactivated.
 - If the CPU goes from RUN to STOP, the DP master is deactivated for at least the duration of the *Response monitoring time*, which is to be set in the PROFIBUS parameters. Afterwards, the DP master becomes active again and takes the DP slaves back to Data Exchange. When the DP master transitions to the inactive state, the output data of the DP slaves are not zeroed nor deactivated. Standard-compliant DP slaves automatically will switch off the outputs or zero them if they do not receive DE telegrams from the master during the *Response monitoring time*.
 - If the power supply of the CPU fails, the outputs of the DP slaves are not zeroed and not deactivated. Standard-compliant DP slaves automatically will switch off the outputs or zero them if they do not receive DE telegrams from the master during the *Response monitoring time*.
- PN MultipleWrite
 - In the activated state, parameter record sets are combined at PROFINET to one or more Ethernet frames during the connection setup. This speeds up the connection setup, since a separate Ethernet frame is not used for each parameter record set.
- Free Module Mapping ↻ 85
 - When activated, you can use your CPU in different hardware variants.
 - You specify the mapping at runtime with record set 0x7F.
- Reduced PDU size
 - When activated, the PDU size is reduced accordingly.
 - For some protocols, some configuration tools require a reduced PDU size. For example, in the Siemens TIA portal for variable forcing, the PDU size must be reduced. For the processing of single steps at several breakpoints, the PDU size must also be reduced.

Access settings ↻ 90

- When activated, you have access to the interface or the corresponding communication protocol.
- By default, there is no access restriction.

4.9.1 Free Module Mapping

4.9.1.1 Overview

Free module mapping - FMM

- With *FMM* you can use your CPU in different hardware variants without adapting your user program. You only have to adapt the FMM configuration in the CPU when configuring the hardware variants. Here you have the following possibilities:
 - Modules from the target configuration can be divided in any order to the slots of the actual configuration.
 - Modules from the target configuration may be missing in the actual configuration.
 - Individual slots of the target configuration can be deactivated, on which modules are located in the actual configuration.
- FMM is a functionality of VIPA and is only supported by VIPA modules.
- By default, FMM is disabled. To use FMM mapping, you must enable the CPU parameter '*Free Module Mapping*'.
- For the *FMM* the mapping of the slots is to be specified via the record set 0x7F.
- For commissioning, you have to enable the parameter '*Startup when expected/actual configuration differs*' in your CPU.
- If FMM is activated and configured correctly, the system reacts as follows:
 - During start-up, no target/actual difference of the hardware is diagnosed.
 - Output data of missing modules are ignored and not output.
 - Input data of missing modules are set to 0.

4.9.1.2 FMM configuration

Configuration

- The mapping of the modules is defined as configuration by the 64byte record set 0x7F.
- The data record is retentively stored in the CPU.
- The record set must be transferred to the CPU by the user program by means of a write command.
- With the record set read command parts of the active configuration can be read. You have always to write the complete record set.
- Each written and valid configuration is only saved if a difference to the existing configuration exists.

Record set 0x7F

Record set 0x7F

| Byte | 0 | 1 | 2 | 3 | ... | 63 |
|---------|---|---|---|---|-----|----|
| Mapping | | | | | | |

- The record 0x7F has a length of 64Byte, where Byte 0 ... 63 corresponds to the slot 1 ... 64 of the target configuration.
- For the FMM configuration, you must specify for each used slot of the target configuration at '*Mapping*' the corresponding value that corresponds to the actual configuration.

The following values can be entered at *Mapping*:

- 0 (0x00) - module is ignored
 - If modules of the target configuration are to be ignored, the value 0x00 must be used. In this way, gaps can be projected.
- 1 ... 64 (0x01 ... 0x40) - position of the module in the actual configuration
 - '*Mapping*' corresponds to the value of Slot_{actual} i.e. the slot of the actual configuration on which the module of the target configuration is located.
- 255 (0xFF) - virtual module
 - If a module from the target configuration is missing, for Mapping the value 255 for "virtual module" is to be used.
 - Behaviour of a *virtual module*:
 - The input area always has the value 0, regardless of its size.
 - The writing to the output area has no effect.

Commissioning

The *target configuration* serves as template for the configuration of hardware variants.

1. ➤ Configure your system with a hardware configuration as target configuration and create your user program. The target configuration represents a superset of all available hardware variants.
2. ➤ Activate the parameter '*Free Module Mapping*' in your CPU.
3. ➤ Activate the parameter '*Startup when expected/actual configuration differs*' in your CPU.
4. ➤ Create the configuration by defining the deviation of the actual and target configuration for the current hardware configuration in record set 0x7F.
5. ➤
 - Transfer this record set via write command to your CPU.
 - For this use SFB 53 or SFB 58.
 - The address to be used is the diagnostic address of the CPU in the virtual IO device '*VIPA SLIO CPU*'.
 - ⇒ The configuration is permanently stored in the CPU and immediately active.

4.9.1.3 Examples

| (1): Target configuration | Slot _{target} |
|----------------------------------|------------------------|
| Slot: 1 2 3 4 5 6 | 1 |
| ① DI DO DIO AI AO CP | 2 |
| | 3 |
| | 4 |
| | 5 |
| | 6 |

Slot_{target} - The mapping always refers to the slot of the target configuration.

Based on the target configuration, the following examples show how to determine the mapping values for the hardware variants.

4.9.1.3.1 Examples of hardware variants

Variant 1: Same type and number of modules but reversed slots

| (1): Target configuration (2): Actual configuration | Slot _{target} | Slot _{actual} | Record set 0x7F | |
|--|------------------------|------------------------|-----------------|---------|
| | | | Byte | Mapping |
| Slot: 1 2 3 4 5 6 | 1 | 2 | 0 | 0x02 |
| ① DI DO DIO AI AO CP | 2 | 1 | 1 | 0x01 |
| | 3 | 3 | 2 | 0x03 |
| | 4 | 5 | 3 | 0x05 |
| | 5 | 6 | 4 | 0x06 |
| ② DO DI DIO CP AI AO | 6 | 4 | 5 | 0x04 |

Determination of Mapping values of record set 0x7F:

- Byte 0: The module of Slot_{target} = 1 is in the actual configuration at Slot_{actual} = 2 → Mapping = 0x02
- Byte 1: The module of Slot_{target} = 2 is in the actual configuration at Slot_{actual} = 1 → Mapping = 0x01
- Byte 2: The module of Slot_{target} = 3 is in the actual configuration at Slot_{actual} = 3 → Mapping = 0x03
- Byte 3: The module of Slot_{target} = 4 is in the actual configuration at Slot_{actual} = 5 → Mapping = 0x05
- Byte 4: The module of Slot_{target} = 5 is in the actual configuration at Slot_{actual} = 6 → Mapping = 0x06
- Byte 5: The module of Slot_{target} = 6 is in the actual configuration at Slot_{actual} = 4 → Mapping = 0x04

Slot_{target} - The mapping always refers to the slot of the target configuration.

Slot_{actual} - Slot of the actual configuration on which the module of the target configuration is located.

Mapping - For variant 1, Mapping corresponds to Slot_{actual}, i.e. slot of the actual configuration on which the module of the target configuration is located.

Variant 2: Reversed slots and modules are missing

| (1): Target configuration (2): Actual configuration | Slot _{target} | Slot _{actual} | Record set 0x7F | |
|--|------------------------|------------------------|-----------------|---------|
| | | | Byte | Mapping |
| | 1 | 1 | 0 | 0x01 |
| | 2 | - | 1 | 0xFF |
| | 3 | 2 | 2 | 0x02 |
| | 4 | 3 | 3 | 0x03 |
| | 5 | 4 | 4 | 0x04 |
| | 6 | - | 5 | 0xFF |

Determination of *Mapping* values of record set 0x7F:

- Byte 0: The module of $Slot_{target} = 1$ is in the actual configuration at $Slot_{actual} = 1 \rightarrow Mapping = 0x01$
- Byte 1: The module of $Slot_{target} = 2$ is not available in the actual configuration $\rightarrow Mapping = 0xFF$
- Byte 2: The module of $Slot_{target} = 3$ is in the actual configuration at $Slot_{actual} = 2 \rightarrow Mapping = 0x02$
- Byte 3: The module of $Slot_{target} = 4$ is in the actual configuration at $Slot_{actual} = 3 \rightarrow Mapping = 0x03$
- Byte 4: The module of $Slot_{target} = 5$ is in the actual configuration at $Slot_{actual} = 4 \rightarrow Mapping = 0x04$
- Byte 5: The module of $Slot_{target} = 6$ is not available in the actual configuration $\rightarrow Mapping = 0xFF$

Slot_{target} - The mapping always refers to the slot of the target configuration.

Slot_{actual} - Slot of the actual configuration on which the module of the target configuration is located.

Mapping - For variant 2, *Mapping* corresponds to the value of $Slot_{actual}$, i.e. slot of the actual configuration on which the module of the target configuration is located. If a module from the target configuration is missing, for *Mapping* the value 0xFF for "virtual module" is to be used.

Variant 3: Modules are ignored

| | Slot _{target} | Slot _{actual} | Record set 0x7F | |
|---|------------------------|------------------------|-----------------|---------|
| | | | Byte | Mapping |
| (1) Target configuration Slot: 1 2 3 4 5 6 DI DO DIO AI AO CP | 1 | empty | 0 | 0x00 |
| (2) Actual configuration DI DO DIO AI AO CP | 2 | empty | 1 | 0x00 |
| | 3 | 3 | 2 | 0x03 |
| | 4 | 4 | 3 | 0x04 |
| | 5 | 5 | 4 | 0x05 |
| | 6 | 6 | 5 | 0x06 |

Determination of *Mapping* values of record set 0x7F:

- Byte 0: The module of $Slot_{target} = 1$ is ignored in the actual configuration → Mapping = 0x00
- Byte 1: The module of $Slot_{target} = 2$ is ignored in the actual configuration → Mapping = 0x00
- Byte 2: The module of $Slot_{target} = 3$ is in the actual configuration at $Slot_{actual} = 3$ → Mapping = 0x03
- Byte 3: The module of $Slot_{target} = 4$ is in the actual configuration at $Slot_{actual} = 4$ → Mapping = 0x04
- Byte 4: The module of $Slot_{target} = 5$ is in the actual configuration at $Slot_{actual} = 5$ → Mapping = 0x05
- Byte 5: The module of $Slot_{target} = 6$ is in the actual configuration at $Slot_{actual} = 6$ → Mapping = 0x06

Slot_{target} - The mapping always refers to the slot of the target configuration.

Slot_{actual} - Slot of the actual configuration on which the module of the target configuration is located.

Mapping - For variant 3, *Mapping* corresponds to the value of $Slot_{actual}$, i.e. slot of the actual configuration on which the module of the target configuration is located. If modules of the target configuration are to be ignored, for *Mapping* the value 0x00 is to be used.



The presence of gaps in the System SLIO is not allowed! But you can place modules and define them via the configuration as empty slot for the target hardware configuration.

4.9.2 Access settings

Overview

The 'Access setting' allows you to disable access to ports or protocols. The CPU has an integrated *Device web page* that shows information about the access ways via 'Access Ways'. ↗ *Chap. 4.11.1 'Device web page CPU' page 94*

Ethernet Port

Here you can disable individual Ethernet interfaces.



Please note that by disabling e.g. the Ethernet PG/OP channel after transferring the hardware configuration, the CPU can no longer be configured via this Ethernet PG/OP channel. The access setting can be reset by an overall reset.

Ethernet protocol

Here you can disable Ethernet protocols. If a protocol is disabled, requests via the disabled protocol will be rejected.

- TCP/UDP/IP services
 - NTP protocol - protocol for time synchronization between the stations.
 - OPC UA - protocol for access to an OPC UA project in the CPU.
 - Open communication - protocol for communication via the user program when using handling blocks.
 - Device WebSite - protocol for access to the integrated web server.
 - Web Visu - protocol for access to the Web visualization in the CPU, which can be configured accordingly.
- S7 connections
 - PG/OP protocol - protocol for PG/OP communication via Siemens S7 connections.
 - PG/OP Routing - routing requests via Siemens S7 connections.
 - NetPro connections - protocol for communication between PLC systems based on Siemens STEP®7 by means of configured communication connections.
- Other services
 - DCP - frame for determining accessible nodes on PROFINET
 - LLDP - frame for determining the topology on PROFINET
 - Field bus PN - protocols for communication via PROFINET

MPI/PB protocol via X3

Here you can disable protocols for the MPI interface X3. If a protocol is disabled, requests via the disabled protocol will be rejected.

- MPI(PB) PG/OP protocol - protocol for PG/OP communication via the MPI(PB) interface X3.
- MPI(PB) Routing - routing requests via the MPI(PB) interface X3.
- Global data communication - protocol for cyclic data exchange between CPUs via the MPI interface.

4.10 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(PtP) 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. ↪ Chap. 4.16 'Deployment storage media - VSD, VSC' page 111

4.10.1 Transfer via MPI / optional PROFIBUS

General

For transfer via MPI the CPU has the following interface:

↪ 'X3: MPI(PtP) interface' page 44

Net structure

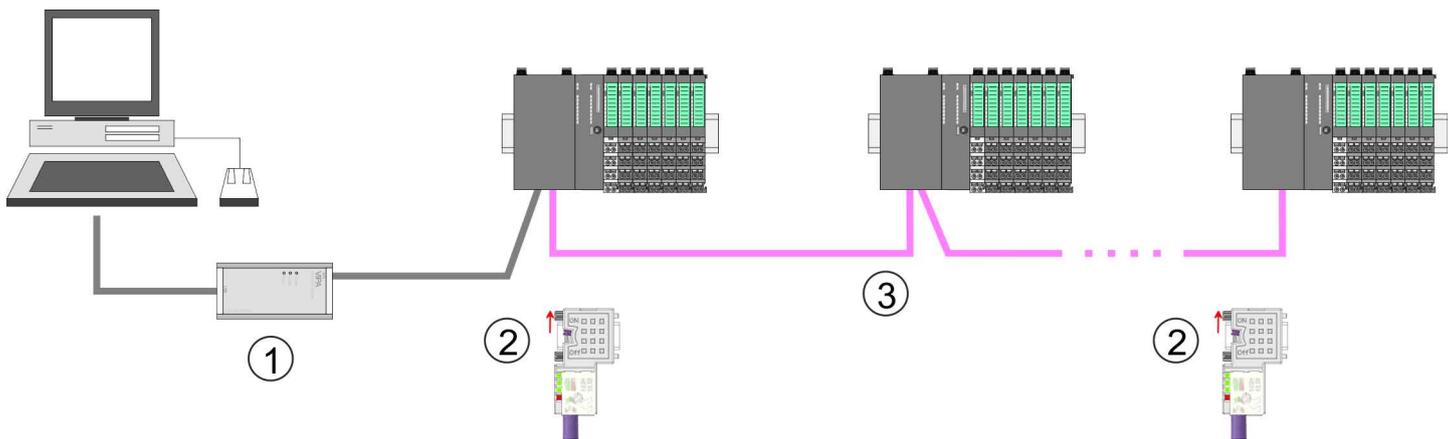
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs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

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 X3 MPI(PtP) 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(PtP) 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.

4.10.2 Transfer via Ethernet

Initialization

So that you may access the according Ethernet interface you have to assign IP address parameters by means of the "initialization".

- X1/X2: Ethernet PG/OP channel
 - ↪ Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 72

Transfer

1. ➤ For the transfer, connect, if not already done, the appropriate Ethernet port 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' Download → the dialog "Select target module" is opened. Select your target module and enter the IP address parameters of the Ethernet PG/OP channel for connection. 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.



System dependent you 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.

4.10.3 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 Siemens SIMATIC Manager with your project
2. ➤ Create with 'File → Memory Card File → New' a new wld file.
3. ➤ Copy the blocks from the project blocks folder and the *System data* into the wld file.
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.

4.11 Accessing the web server

Overview

The CPU has a web server integrated. This provides access to:

- Device web page
- OPC UA project ↗ *Chap. 4.11.1.1.1 'Tab: 'OPC UA'' page 97*
- WebVisu project ↗ *Chap. 4.11.1.1.2 'Tab: 'WebVisu'' page 98*

Accessing the web server > Device web page CPU

4.11.1 Device web page CPU

Overview

- Dynamic web page, which exclusively outputs information.
- On the *device web page* you will find information about your CPU, the connected modules and your *WebVisu* project.
 - CPU
 - the connected modules
 - *OPC UA* project
 - *WebVisu* project
- The shown values cannot be changed.
- Access is via the IP address of the Ethernet PG/OP channel.
 - ↳ *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 72*
- You can access the IP address with a web browser.



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

4.11.1.1 Web page with selected CPU

Tab: 'Info'

| Name | Value |
|---------------|-------|
| Ordering Info | |
| Serial | |
| Version | |
| HW Revision | |
| Software | |
| Package | |

[Expert View ...]

| Name | Value | |
|---------------|--------------|-----------------------------------|
| Ordering Info | 013-CCF0R00 | Order number of the CPU |
| Serial | ... | Serial number of the CPU |
| Version | 01V... | Version number of the CPU |
| HW Revision | 01 | CPU hardware revision |
| Software | 3.0.0 | CPU firmware version |
| Package | Pb000265.pkb | File name for the firmware update |

[Expert View] takes you to the advanced "Expert View".

| Runtime Information | | CPU |
|---------------------|--|--|
| Operation Mode | RUN | Mode |
| Mode Switch | RUNP | |
| System Time | 14.03.19 08:34:14:486 | Date, time |
| Up Time | 0 days 02 hrs 07 min 08 sec | Time to change the operating mode |
| Last Change to RUN | n/a | |
| Last Change to STOP | 14.03.19 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 | | | Interface |
|-----------------------|-----------------------|--------------------|--|
| X1 | PG/OP Ethernet Port 1 | Address 2025..2040 | Ethernet PG/OP channel |
| X2 | PG/OP Ethernet Port 2 | Address 2025..2040 | |
| X3 | MPI | Address 2047 | Operating mode RS485 MPI: MPI operation PtP operation PROFIBUS DP operation |
| X4 | DI 16 | Address 136..137 | Digital input |
| | AI 2 | Address 800..803 | Analog input |
| | Counter | Address 816..831 | Counter |
| X5 | DO 12 | Address 136..137 | Digital output |
| | Counter | Address 816..831 | Counter |

| Card Information | | |
|-------------------|--------------|-----------------------------------|
| Type | SD | Information about the memory card |
| Manufacturer ID | 9 | |
| Application ID | 16720 | |
| Card Name | AF SD | |
| Card Revision | 16 | |
| Card S/N | 64C34010 | |
| Manufacture Month | 8 | |
| Manufacture Year | 2013 | |
| Size | 470.73 MByte | |
| Free | 454.70 MByte | |

| VSC Information | | |
|--------------------|-------------|---------------------|
| VSC Product Number | 955-C000M20 | Information via VSC |
| VSC Product S/N | 00001055 | |

Accessing the web server > Device web page CPU

| VSC Information | |
|------------------|-----------|
| Memory Extension | 64 kByte |
| Profibus | PB Master |

| Active Feature Set Information | |
|--------------------------------|---------------|
| Status | Media present |
| VSC Product Number | 955-C000M20 |
| VSC Product S/N | 00001055 |
| Memory Extension | 64 kByte |
| Profibus | PB Master |

| Memory Usage | | | | CPU |
|--------------|-------------|--------|-------------|---|
| | free | used | max | Information on the memory expansion |
| LoadMem | 128.0 kByte | 0 byte | 128.0 kByte | Load memory, working memory (code/data) |
| WorkMemCode | 64.0 kByte | 0 byte | 64.0 kByte | |
| WorkMemData | 64.0 kByte | 0 byte | 64.0 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 Mode and speed |
| Link Mode X2 | Not Available | |

| CPU Firmware Information | CPU | |
|--------------------------|--|---------------------------------|
| File System | V1.0.2 | Name, firmware version, package |
| PRODUCT | VIPA 013-CCF0R00 V3.0.0 Px000265.pkg | |
| HARDWARE | V0.1.0.0 5846A-V12 MX000303.004 | |
| BOOTLOADER | Bx000715 V126 | Information for the support |
| Bx000501 | V2.2.5.0 | |
| Ax000136 | V1.0.6.0 | |
| fx000018.wld | V1.0.2.0 | |
| syslibex.wld | n/a | |
| Protect.wld | n/a | |

| ARM Processor Load | | CPU |
|---------------------------|--------|-----------------------------|
| Measurement Cycle Time | 100 ms | Information for the support |
| Last Value | 9% | |
| Average Of Last 10 Values | 10% | |
| Minimum Load | 9% | |
| Maximum Load | 23% | |

Tab: 'IP' Here the IP address data of your Ethernet PG/OP channel are shown.

Tab: 'Firmware' As of CPU firmware version V3.0.0, you can transfer the firmware file online to the CPU via the 'Firmware' tab. The firmware update in the CPU is triggered by means of the operating mode switch. ↪ *Chap. 4.14 'Firmware update' page 106*

Tab: 'Access Ways' As of CPU firmware version V3.0.0, information about access settings is shown here. In the delivery state, there are no restrictions. You can specify access to interfaces, ports and protocols via the parametrization. ↪ *Chap. 4.9 'Setting VIPA specific CPU parameters' page 82*

4.11.1.1.1 Tab: 'OPC UA'

As of CPU firmware version V3.0.0, information about the OPC UA project is shown here. ↪ *Chap. 7 'Deployment OPC UA' page 183*

| Start Conditions | |
|------------------------|--------------------------------|
| Conflicting projects | no conflicting project on card |
| Feature Set activated | yes |
| Allowed by User | yes |
| Allowed by Access Ways | yes |
| Hardware configured | yes |
| OPC UA project | loaded |

| Server | |
|--------------|-------------------------------|
| State | running |
| Endpoint URL | opc tcp://190.165.72.180:4840 |



For your CPU can process a OPC UA project, you have to activate the OPC UA functionality. ↪ Chap. 7.3 'Activate OPC UA functionality' page 192

Start Conditions

Here the start conditions for the *OPC UA* server are listed:

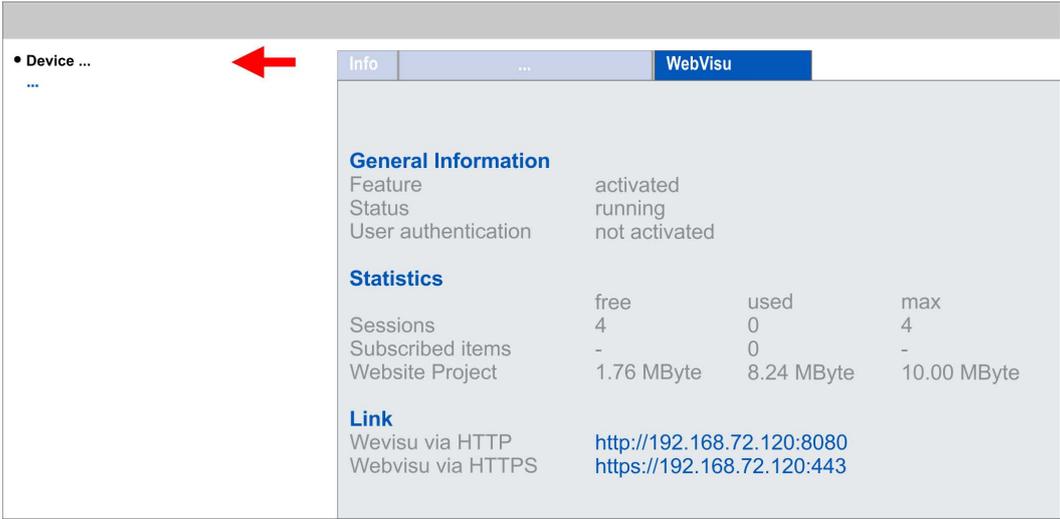
- Conflicting projects
 - Simultaneous use of an *OPC UA* and *WebVisu* project via the same interface is not permitted and results in the message *'conflicting project on card'*.
 - The status *'no conflicting project on card'* indicates that there is no conflict with a *WebVisu* project on the same interface.
- Feature Set activated
 - yes: The *OPC UA* functionality is activated.
 - no: The *OPC UA* functionality is not activated.
 - ↪ *Chap. 7.3 'Activate OPC UA functionality' page 192*
- Allowed by User
 - yes: The *OPC UA* server is activated and access to it is allowed. As soon as an *OPC UA* project is found on the memory card, it is automatically started and enabled for access.
 - no: You can disable and stop the *OPC UA* server by means of the CMD - auto command *'OPCUA_PGOP_DISABLE'*. With *'OPCUA_PGOP_ENABLE'* you can enable to restart the *OPC UA* server.
 - ↪ *Chap. 4.18 'CMD - auto commands' page 114*
- Allowed by Access Ways
 - yes: By default, the *OPC UA* protocol is enabled.
 - no: The *OPC UA* protocol is disabled. You can deactivate or activate the *OPC UA* protocol by means of the parameter *'OPC UA'* in the *'Access settings'*.
- Hardware configured
 - yes: A hardware configuration as a precondition for the *OPC UA* communication is loaded. The hardware configuration is checked for validity elsewhere.
 - no: A hardware configuration is not loaded e.g. after an overall reset.
 - ↪ *Chap. 4.5 'Hardware configuration - CPU' page 69*
- OPC UA project
 - loaded: An *OPC UA* project is loaded.
 - not loaded: An *OPC UA* project is not loaded.
 - ↪ *Chap. 7 'Deployment OPC UA' page 183*

Server

- State
 - Running: The start conditions are fulfilled and the *OPC UA* server is started.
 - Stopped: The *OPC UA* server is stopped.
 - Startup failure: The *OPC UA* server can not be started.
 - Starting: The *OPC UA* server currently starts up.
 - Stopping: The *OPC UA* server currently stops.
- Endpoint URL
 - As soon as the *OPC UA* server is started, the endpoint URL of the *OPC UA* server is listed here.

4.11.1.1.2 Tab: 'WebVisu'

Information about the web visualization (*'WebVisu'*) are shown here. The creation of a *'WebVisu'* project is only possible with the *SPEED7 Studio V1.7.0* and up. ↪ *Chap. 8 'Deployment WebVisu - Web visualization' page 209*



• Device ...

Info ... **WebVisu**

General Information

| | |
|---------------------|---------------|
| Feature | activated |
| Status | running |
| User authentication | not activated |

Statistics

| | free | used | max |
|------------------|------------|------------|-------------|
| Sessions | 4 | 0 | 4 |
| Subscribed items | - | 0 | - |
| Website Project | 1.76 MByte | 8.24 MByte | 10.00 MByte |

Link

| | |
|-------------------|---|
| Wevisu via HTTP | http://192.168.72.120:8080 |
| Webvisu via HTTPS | https://192.168.72.120:443 |



For your CPU can process a WebVisu project, you have to activate the WebVisu functionality. ↪ Chap. 8.2 'Activate WebVisu functionality' page 212

General Information

- Feature
 - activated: The *WebVisu* functionality is activated.
 - not activated: The *WebVisu* functionality is not activated.
- Status
 - The status of your *WebVisu* project is shown here. ↪ Chap. 4.11.1.1.2.1 'Status of the WebVisu' page 100
- User authentication
 - activated: User authentication is activated. Access to the *WebVisu* happens via a login by user name and password.
 - not activated: User authentication is de-activated. Access to the *WebVisu* is unsecured.

Statistics

Statistical information about your *WebVisu* project are shown here.

- Sessions: Number of sessions, i.e. online connections to this *WebVisu* project. A session corresponds to an open window or tab in a web browser.
 - free: Number of sessions still possible.
 - used: Number of active sessions. For the number of active sessions, it is not relevant whether the sessions were started by the same or different users.
 - max.: Number of sessions still possible. The maximum number of sessions is device specific and specified in the technical data.
- Subscribed items: Number of variables.
 - free: Here nothing is shown.
 - used: Number of variables used.
 - max.: Here nothing is shown.
- WebVisu Project: Information on the memory allocation for the *WebVisu* project.
 - free: Still free space for the *WebVisu* project.
 - used: Size of the current *WebVisu* project.
 - max.: Maximum available space for a *WebVisu* project.

Link

In *Status 'running'* the links to access your *WebVisu* are listed here.

Accessing the web server > Device web page CPU

Status of the *WebVisu*

On the device web page at the tab '*WebVisu*' via '*Status*' you get the status of your *WebVisu* project.

| Status | Meaning |
|--------------------------------------|---|
| running | <i>WebVisu</i> is active / has started-up and can be opened |
| loading webvisu project | Loading <i>WebVisu</i> project |
| shutting down | <i>WebVisu</i> server shuts down |
| stop requested | <i>WebVisu</i> STOP requested |
| stopped | <i>WebVisu</i> server is down |
| webvisu feature not activated | <i>WebVisu</i> not activated, memory card is not inserted |
| webvisu is disabled by the user | <i>WebVisu</i> was disabled by the user |
| no webvisu project file found | No <i>WebVisu</i> project found |
| no hardware configuration loaded | No hardware configuration is loaded |
| invalid configuration | Invalid <i>WebVisu</i> configuration |
| internal error: filesystem | Error initializing the file system |
| webvisu project file too large | Error loading <i>WebVisu</i> project, project file too large |
| loading webvisu project file | Error loading <i>WebVisu</i> project, project file may be damaged |
| deleting webvisu project | Failed to delete the <i>WebVisu</i> project |
| internal error: file system - delete | <i>WebVisu</i> project to be deleted was not found in the memory |
| CRC mismatch | CRC of the <i>WebVisu</i> project file is not correct |
| webvisu stopped | <i>WebVisu</i> server has terminated unexpectedly |
| internal error 1 | Internal error - initialization failed step 1 |
| internal error 2 | Internal error - initialization failed step 2 |
| internal error 3 | Unexpected internal error |
| unknown error | General error |

4.11.1.1.3 Tab: '*Port Mirroring*'

Overview

- *Port Mirroring* offers the possibility to diagnose the communication without additional hardware effort.
- The Ethernet PG/OP interface is designed as switch.
- When *Port Mirroring* of the PG/OP2: X2 (Mirror Port) interface is activated, all telegrams received and sent via the PG/OP1: X1 interface are mirrored to the PG/OP2: X2 interface and vice versa.
- When *Port Mirroring* is activated, for diagnostics with diagnostics software such as Wireshark, you can connect your PC directly to the 2. interface.
- The next power-cycle will automatically disable *Port Mirroring*.

Enable *Port Mirroring*

When enabled, you can set the parameters for *Port Mirroring*.

- PG/OP1: X1** When enabled the frames of PG/OP2: X2 are mirrored to PG/OP1: X1.

- PG/OP2: X2** When enabled the frames of PG/OP1: X1 are mirrored to PG/OP2: X2.

- Disable communication on the Mirror Port** When enabled additional communication via the mirrored interface (Mirror Port) is prevented.

- Save** With *Save*, the setting are taken and activated. The next power-cycle will automatically disable *Port Mirroring*.

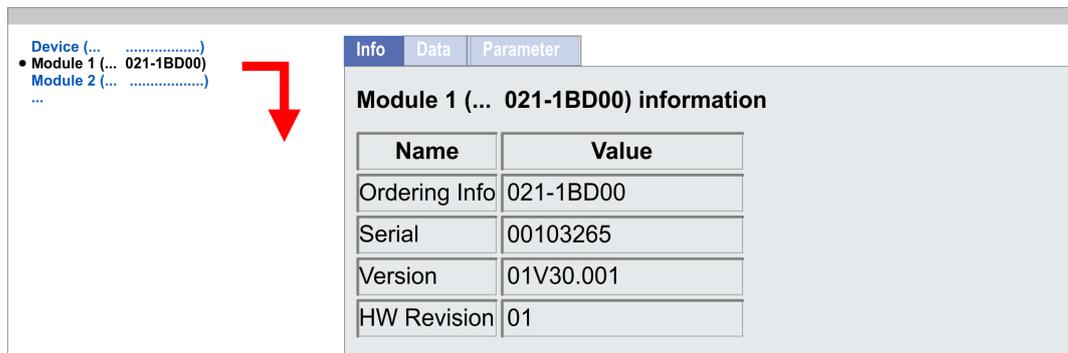
4.11.1.2 Web page with selected module

Structure 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.

Module



| Module 1 (... 021-1BD00) information | |
|--------------------------------------|-----------|
| Name | Value |
| Ordering Info | 021-1BD00 |
| Serial | 00103265 |
| Version | 01V30.001 |
| HW Revision | 01 |

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

Tab: '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.

Tab: 'Parameter' With parametrizable modules e.g. analog modules the parameter setting is shown here. These come from the hardware configuration.

4.12 Operating modes

4.12.1 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.

4.12.2 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- S perre, i.e. command output disable) is set. |
| | central digital outputs | The outputs are disabled. |
| | central analog outputs | The outputs are disabled. <ul style="list-style-type: none"> ■ 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

4.13 Overall reset

Overview

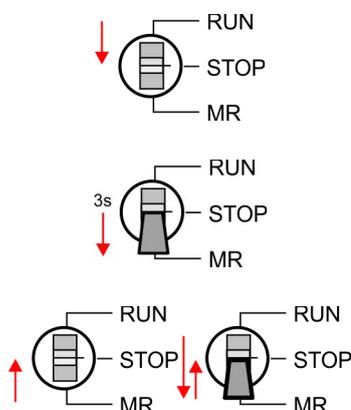
During the overall reset the entire user memory is erased. Data located in the memory card is not affected. You have 2 options to initiate an overall reset:

- Overall reset by means of the operating mode switch
- Overall reset by means of a configuration tool like e.g. the Siemens SIMATIC Manager



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.

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.

Overall reset by means of the Siemens SIMATIC Manager

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. You may place the CPU in STOP by the menu command 'PLC → Operating mode'.
2. You may request the overall reset by means of the menu command 'PLC → Clean/Reset'.
 - ⇒ 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 ST LED flashes. When the ST LED is on permanently the overall reset procedure has been completed.

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' page 111 ↗ Chap. 4.16 'Deployment storage media - VSD, VSC' page 111

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! ↗ Chap. 4.15 'Reset to factory settings' page 110

4.14 Firmware update

Overview

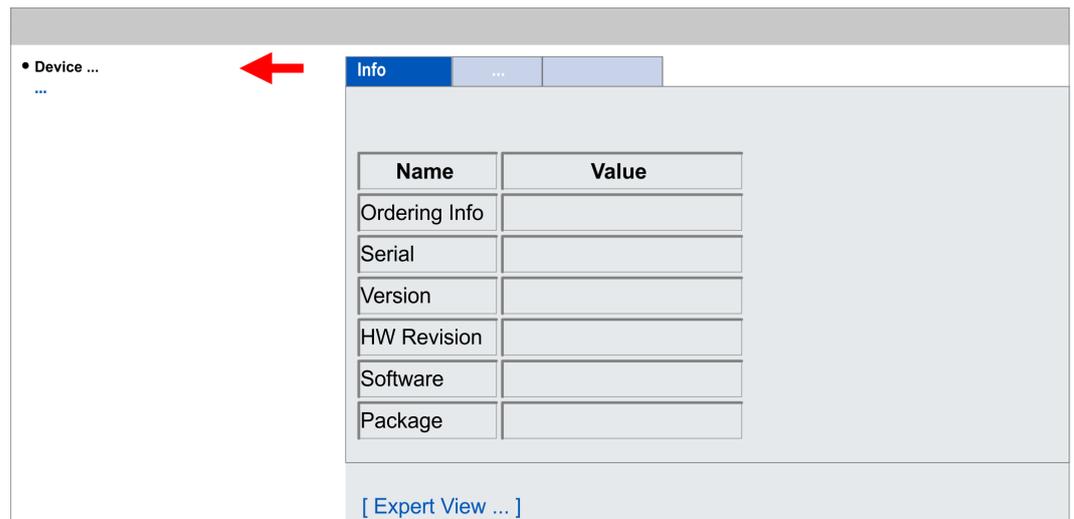
You can find current firmware versions at www.vipa.com in the service area. There are the following possibilities for the firmware update:

- Firmware update online - from FW V3.0.0 ↪ *Chap. 4.14.1 'Firmware update online' page 107*
 - Transfer of the firmware file to the CPU via the CPU web page.
 - Triggering the firmware update by means of the operating mode switch.
- Firmware update via memory card ↪ *Chap. 4.14.2 'Firmware update via memory card' page 108*
 - Transfer of the firmware file to a memory card.
 - The identification of a firmware file on the memory card takes place by means of a defined naming convention.
 - After PowerON and operating mode switch in the STOP position, the firmware update can be triggered by means of operating mode switch.

Show the firmware version via web page

The CPU has an integrated *Device web page* that also shows information about the firmware version via *'Info'*. Here you will also find information about the required firmware *'Package'*. With **[Expert View]** you can access the extended "Expert" overview. ↪ *Chap. 4.11.1 'Device web page CPU' page 94*

Tab: 'Info'



| Name | Value | |
|---------------|--------------|-----------------------------------|
| Ordering Info | 013-CCF0R00 | Order number of the CPU |
| Serial | ... | Serial number of the CPU |
| Version | 01V... | Version number of the CPU |
| HW Revision | 01 | CPU hardware revision |
| Software | 3.0.0 | CPU firmware version |
| Package | Pb000265.pkb | File name for the firmware update |

**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 file is necessary for the firmware update of the CPU 013-CCF0R00 and its components with hardware release 01:

- CPU 013C, Hardware release 01: Pb000265.pkb

**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 our hotline!

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

4.14.1 Firmware update online

Precondition

- Access is via the IP address of the Ethernet PG/OP channel.
↳ *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 72*
- You can access the IP address with a web browser.



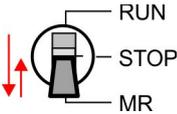
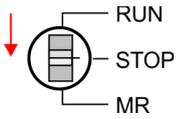
It is assumed that there is a connection via the Ethernet PG/OP channel between the PC with web browser and the CPU. This may be tested by Ping to the IP address of the Ethernet PG/OP channel.

**Load firmware and store it
in working directory**

1. ➤ Go to www.vipa.com.
2. ➤ Click at 'Service/Support → Downloads → Firmware'.
3. ➤ Via 'System SLIO → CPU' navigate to your CPU and download the zip file to your PC.
4. ➤ Extract the zip file into your working directory.

**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!

Perform firmware update

1. ➤ Switch the operating mode switch of your CPU in position STOP.
2. ➤ Execute an overall reset. [↪ Chap. 4.13 'Overall reset' page 105](#)
3. ➤ Open the CPU web page and select the 'Firmware' tab.
4. ➤ Click at 'Browse ...' and navigate to the firmware file in your working directory.
5. ➤ Click at 'Upload'.
 - ⇒ The firmware file is checked for plausibility and transmitted to the CPU. After the transfer, the firmware versions are matched and listed with the note if a firmware update is possible.
6. ➤ You start the firmware update by tipping the operating mode switch downwards to MR and then leaving the switch in STOP position.
 - ⇒ During the update process, the LEDs SF and FC are alternately blinking. This may last several minutes.
7. ➤ The update is successful finished when the LEDs PW, SF, ST, FC and SD are on. If they are blinking fast, an error occurred.
8. ➤ Turn power OFF and ON.
 - ⇒ After the start-up, the CPU is ready for operation with the new firmware. The current firmware version can be determined via the web page of the CPU.

4.14.2 Firmware update via memory card**Overview**

- For the firmware update via memory card 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 pkb file name is reserved for each hardware revision, which begins with "pb" and differs in a number with 6 digits.
- In the VIPASystem SLIO CPU, the pkb file name can be shown via the web page.
- After PowerON and operating mode switch of the CPU in STOP, the CPU checks if there is a pkb 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.



The procedure here describes the update from the CPU firmware version V2.4.0. The update of an older version to the firmware version V2.4.0 has to be done via pkg files. For this refer to the corresponding manual for your CPU version.

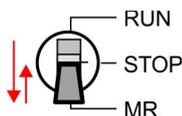
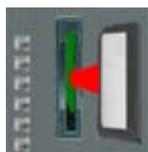
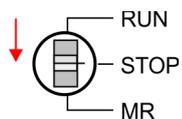
Load firmware and transfer it to memory card

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 pgb 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!

Transfer firmware from memory card into CPU



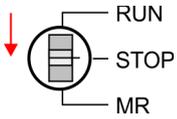
1. ➤ Switch the operating mode switch of your CPU in position STOP.
2. ➤ Turn off the power supply.
3. ➤ Plug the memory card with the firmware file into the CPU. Please take care of the correct plug-in direction of the memory card.
4. ➤ Turn on the power supply.
 - ⇒ After a short boot-up time, the alternate blinking of the LEDs SF and FC shows that 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, SF, ST, FC and SD are on. If they are blinking fast, an error occurred.
7. ➤ Turn power OFF and ON.
8. ➤ Now execute a *Reset to factory setting*. After that the CPU is ready for duty.
 - 🔗 *Chap. 4.15 'Reset to factory settings' page 110*

4.15 Reset to factory settings

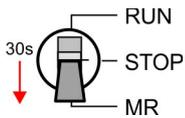
Proceeding

- 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`. ↪ *Chap. 4.18 'CMD - auto commands' page 114*

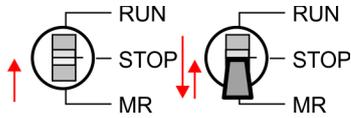
1. ➔ Switch the CPU to STOP.



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.

i *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.



4.16 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 - **VIPA SD-Card**
 - External memory card for programs and firmware.
- VSC - **VIPASetCard**
 - 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. ↪ *Chap. 4.13 'Overall reset' page 105*



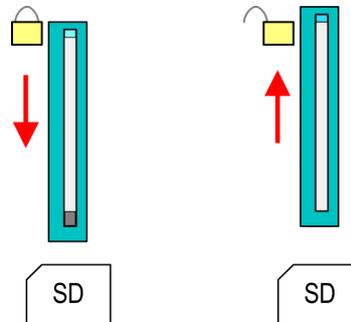
To avoid malfunctions, you should use memory cards of VIPA. These correspond to the industrial standard. A list of the currently available VSD respectively VSC can be found at www.vipa.com

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. ↪ *Chap. 4.11 'Accessing the web server' page 93*

**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 "TrialTime" 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 a 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 *.pkb file (firmware file). If so, this is shown by the CPU by blinking LEDs and the firmware may be installed by an update request. ↪ *Chap. 4.14 'Firmware update' page 106*

In STOP state when inserting a memory card

- If a memory card is plugged in STOP state, 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".

4.17 Extended know-how protection

Overview



Please note that this functionality is not supported by the Siemens TIA Portal!

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*. ↪ [Chap. 4.13 'Overall reset' page 105](#)
 - ⇒ 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.

4.18 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.
- 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 |
| 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 |
| WEBPAGE | Saves all information on the device web page (Expert-View) as <i>webpage.txt</i> on the memory card ↪ <i>Chap. 4.11 'Accessing the web server' page 93</i> | 0xE804 |
| WEBVISU_PGOP_ENABLE | Enable <i>WebVisu</i> project via Ethernet PG/OP channel | 0xE82C |
| WEBVISU_PGOP_DISABLE* | Disable <i>WebVisu</i> project via Ethernet PG/OP channel | 0xE82D |

| Command | Description | Diagnostics entry |
|---|--|-------------------|
| OPCUA_PGOP_ENABLE | Enable <i>OPC UA</i> project via Ethernet PG/OP channel | 0xE830 |
| OPCUA_PGOP_DISABLE | Disable <i>OPC UA</i> project via Ethernet PG/OP channel | 0xE831 |
| *) After a power cycle or loading a hardware configuration, the settings are retained. With <i>reset to the factory settings</i> or <i>over all reset</i> , the <i>WebVisu</i> project is set to the default value "enabled". | | |

Examples

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

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.

4.19 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 via the menu item **Debug** of the Siemens SIMATIC Manager.
- The status of the operands and the RLO can be displayed by means of the test function *'Debug → Monitor'*.
- The status of the operands and the RLO can be displayed by means of the test function *'PLC → Monitor/Modify Variables'*.

'Debug → Monitor'

- This test function displays the current status and the RLO of the different operands while the program is being executed.
- It is also possible to enter corrections to the program.
- The processing of the states may be interrupted by means of jump commands or by timer and process-related interrupts.
- At the breakpoint the CPU stops collecting data for the status display and instead of the required data it only provides the PG with data containing the value 0.
- The interruption of the processing of statuses does not change the execution of the program. It only shows that the data displayed is no longer valid.



When using the test function "Monitor" the PLC must be in RUN mode!

For this reason, jumps or time and process alarms can result in the value displayed during program execution remaining at 0 for the items below:

- the result of the logical operation RLO
- Status / AKKU 1
- AKKU 2
- Condition byte
- absolute memory address SAZ. In this case SAZ is followed by a "?".

**'PLC
→ Monitor/Modify
Variables'**

This test function returns the condition of a selected operand (inputs, outputs, flags, data word, counters or timers) at the end of program execution. 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.

- Control of outputs
 - Serves to check the wiring and proper operation of output modules.
 - If the CPU is in RUN mode, so only outputs can be controlled, which are not controlled by the user program. Otherwise values would be instantly overwritten.
 - If the CPU is in STOP - even without user program, so you need to disable the command output lock BASP ('Enable PO'). Then you can control the outputs arbitrarily
- Controlling variables
 - The following variables may be modified: I, Q, M, T, C and D.
 - The process image of binary and digital operands is modified independently of the operating mode of the CPU.
 - When the operating mode is RUN the program is executed with the modified process variable. When the program continues they may, however, be modified again without notification.
- Forcing variables
 - You can pre-set individual variables of a user program with fixed values so that they can not be changed or overwritten by the user program of the CPU.
 - By pre-setting of variables with fixed values, you can set certain situations for your user program and thus test the programmed functions.



CAUTION!

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

Even after a power cycle forced variables remain forced with its value, until the force function is disabled.

These functions should only be used for test purposes respectively for troubleshooting. More information about the usage of these functions may be found in the manual of your configuration tool.

4.20 Diagnostic entries

Accessing diagnostic data ↪ *Appendix A 'System specific event IDs' page 325*

- 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.
- 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 current content of the diagnostic buffer is stored at the memory card by means of the CMD DIAGBUF. ↪ *Chap. 4.18 'CMD - auto commands' page 114*
- 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 I/O periphery

5.1 Overview

Project engineering and parametrization

- On this CPU the connectors for digital respectively analog signal and *Technological functions* are combined in a one casing.
- The project engineering happens in the Siemens SIMATIC Manager as Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3). Here the CPU 013-CCF0R00 is parameterized via the '*Properties*' dialog of the Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
- For parametrization of the digital I/O periphery and the *technological functions* the corresponding sub modules of the CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) is to be used.
- The controlling of the operating modes of the *technological functions* happens by means of handling blocks of the user program.

I/O periphery

- The integrated I/Os of the CPU may be used for *technological functions* or as standard periphery.
- *Technological functions* and standard periphery may be used simultaneously with appropriate hardware.
- Read access to inputs used by *technological functions* is possible.
- Write access to used outputs is not possible.
- ↪ *Chap. 5.3 'Analog input' page 120*
 - 2xUx12Bit (0 ... 10V)
 - The analog channels of the module are not isolated to the electronic power supply.
 - The analog part has no status indication
- ↪ *Chap. 5.4 'Digital input' page 123*
 - 16xDC 24V
 - Interrupt functions parameterizable
 - Status indication via LEDs
- ↪ *Chap. 5.5 'Digital output' page 126*
 - DO 12xDC 24V, 0.5A
 - Status indication via LEDs

Technological functions

- ↪ *Chap. 5.6 'Counting' page 128*
 - 4 channels
 - Count once
 - Count continuously
 - Count Periodically
 - Control by the user program (SFB 47)
- ↪ *Chap. 5.7 'Frequency measurement' page 148*
 - 4 channels
 - Control by the user program (SFB 48)
- ↪ *Chap. 5.8 'Pulse width modulation - PWM' page 153*
 - 2 channels
 - Control by the user program (SFB 49)
- ↪ *Chap. 5.9 'Pulse train' page 157*
 - 2 channels
 - Control by the user program (SFB 49)

5.2 Address assignment

| Sub module | Input address | Access | Description |
|------------|---------------|--------|-----------------------------|
| AI5/AO2 | 800 | WORD | Analog input channel 0 (X4) |
| | 802 | WORD | Analog input channel 1 (X4) |

| Sub module | Input address | Access | Description |
|------------|---------------|--------|------------------------------------|
| DI24/DO16 | 136 | BYTE | Digital input I+0.0 ... I+0.7 (X4) |
| | 137 | BYTE | Digital input I+1.0 ... I+1.7 (X4) |

| Sub module | Input address | Access | Description |
|------------|---------------|--------|--|
| Counter | 816 | DINT | Channel 0: Counter value / Frequency value |
| | 820 | DINT | Channel 1: Counter value / Frequency value |
| | 824 | DINT | Channel 2: Counter value / Frequency value |
| | 828 | DINT | Channel 3: Counter value / Frequency value |

| Sub module | Output address | Access | Description |
|------------|----------------|--------|-------------|
| Counter | 816 | DWORD | reserved |
| | 820 | DWORD | reserved |
| | 824 | DWORD | reserved |
| | 828 | DWORD | reserved |

| Sub module | Output address | Access | Description |
|------------|----------------|--------|-------------------------------------|
| DI24/DO16 | 136 | BYTE | Digital output Q+0.0 ... Q+0.7 (X5) |
| | 137 | BYTE | Digital output Q+1.0 ... Q+1.3 (X5) |

5.3 Analog input

5.3.1 Properties

- 2xUx12Bit (0 ... 10V) fixed.
- The analog channels of the module are not isolated to the electronic power supply.
- The analog part has no status indication.



Temporarily not used analog inputs must be connected to the concerning ground.

5.3.2 Analog value representation

Number representation in Siemens S7 format

| Resolution | Analog value - twos complement | | | | | | | | | | | | | | | |
|------------|--------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|----------------|-------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | High byte (byte 0) | | | | | | | | Low byte (byte 1) | | | | | | | |
| Bit number | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | SG | 2 ¹⁴ | 2 ¹³ | 2 ¹² | 2 ¹¹ | 2 ¹⁰ | 2 ⁹ | 2 ⁸ | 2 ⁷ | 2 ⁶ | 2 ⁵ | 2 ⁴ | 2 ³ | 2 ² | 2 ¹ | 2 ⁰ |
| 11Bit+sign | SG | Measuring value | | | | | | | | | | | X* | X* | X* | X* |

*) The lowest value irrelevant bits of the output value (0) are marked with "X".

Sign bit (SG)

Here it is essential:

- Bit 15 = "0": → positive value
- Bit 15 = "1": → negative value

Behavior at error

As soon as a measured value exceeds the overdrive region respectively falls below the underdrive region, the following value is issued:

- Measuring value > end of overdrive region:
32767 (7FFFh)
- Measuring value < end of underdrive region:
-32768 (8000h)

At a parameterization error the value 32767 (7FFFh) is issued.

When leaving the defined range during analog output 0V respectively 0A is issued.

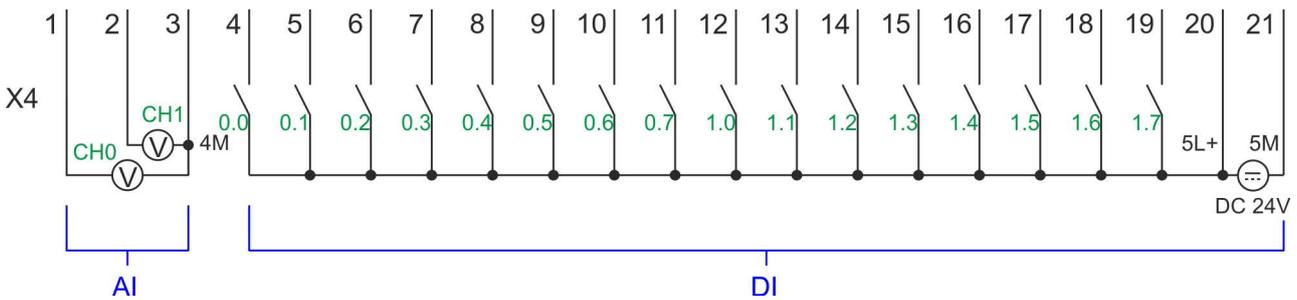
Voltage measurement

0 ... 10V

| Measuring range | Voltage (U) | Decimal (D) | Hex | Range | Formulas |
|-----------------|-------------|-------------|-------|------------------|---|
| 0 ... 10V | > 11.759V | 32767 | 7FFFh | overflow | $D = 27648 \cdot \frac{U}{10}$ $U = D \cdot \frac{10}{27648}$ |
| | 11.759V | 32511 | 7EFFh | overdrive range | |
| | 10V | 27648 | 6C00h | nominal range | |
| | 5V | 13824 | 3600h | | D: decimal value U: voltage value |
| | 0V | 0 | 0000h | | |
| | -0.8V | -2212 | F75Ch | underdrive range | |
| | < -0.8V | -32768 | 8000h | underflow | |

5.3.3 Wiring

X4: Connector



| X4 | Function | Type | Description |
|----|-----------|------|---------------------------|
| 1 | AI 0 | I | AI0: Analog input AI 0 |
| 2 | AI 1 | I | AI1: Analog input AI 1 |
| 3 | Analog 0V | I | 4M: GND for analog inputs |

Cables for analog signals

For the analog signals you have to use isolated cables. With this the interferences can be reduced. The shield of the analog cables should be grounded at both ends. If there are potential differences between the cables, a potential compensation current can flow, which could disturb the analog signals. In this case, you should only ground the shield at one end of the cable.



Temporarily not used analog inputs must be connected to the concerning ground.

5.3.4 Parametrization

5.3.4.1 Adress assignment

| Sub module | Input address | Access | Description |
|------------|---------------|--------|-----------------------------|
| AI5/AO2 | 800 | WORD | Analog input channel 0 (X4) |
| | 802 | WORD | Analog input channel 1 (X4) |

5.3.4.2 Filter

Parameter hardware configuration

The analog input part has a filter integrated. The parametrization of the filter happens in the Siemens SIMATIC Manager via the parameter *'Integration time'*. The default value of the filter is 1000ms. The following values can be entered:

- *'Input 0 ≙ Channel 0'*
'Input 1 ≙ Channel 1'
 - *'Integration time 2.5ms' ≙ 2ms* (no filter)
 - *'Integration time 16.6ms' ≙ 100ms* (small filter)
 - *'Integration time 20ms' ≙ 1000ms* (medium filter)

Parametrization during runtime

By using the record set 1 of the SFC 55 "WR_PARM" you may alter the parametrization in the module during runtime.



The time needed until the new parametrization is valid can last up to 2ms. During this time, the measuring value output is 7FFFFh.

Record set 1

| Byte | Bit 7 ... Bit 0 | Default |
|--------|--|---------|
| 0 | Bit 7...0: reserved | 00h |
| 1 | Filter <ul style="list-style-type: none"> ■ Bit 1, 0: Analog input channel 0 Bit 3, 2: Analog input channel 1 <ul style="list-style-type: none"> – 00b: <i>'Integration time 2.5ms' ≙ 2ms</i> (no filter) – 01b: <i>'Integration time 16.6ms' ≙ 100ms</i> (small filter) – 10b: <i>'Integration time 20ms' ≙ 1000ms</i> (medium filter) ■ Bit 7...4: reserved | 10h |
| 2...12 | Bit 7...0: reserved | |

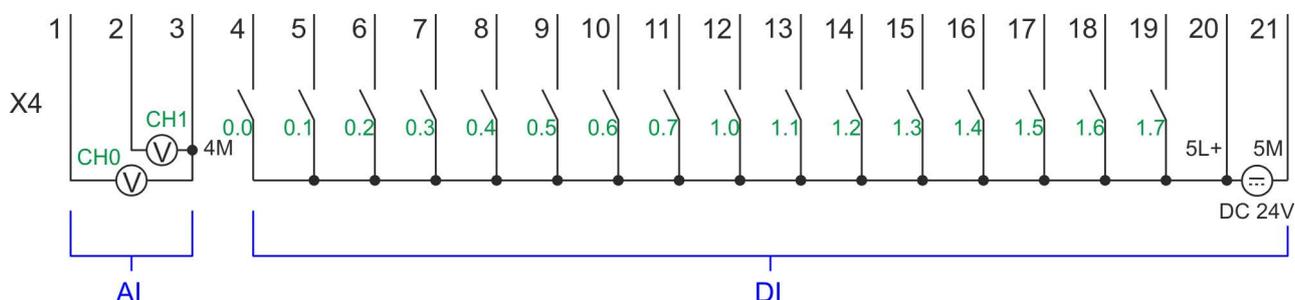
5.4 Digital input

5.4.1 Properties

- 16xDC 24V
- Maximum input frequency
 - 10 inputs: 100kHz
 - 6 inputs: 1kHz
- Interrupt functions parameterizable
- Status indication via LEDs

5.4.2 Wiring

X4: Connector



| X4 | Function | Type | Description |
|----|----------|------|--|
| 4 | DI 0 | I | +0.0: Digital input DI 0 |
| 5 | DI 1 | I | +0.1: Digital input DI 1 |
| 6 | DI 2 | I | +0.2: Digital input DI 2 |
| 7 | DI 3 | I | +0.3: Digital input DI 3 * |
| 8 | DI 4 | I | +0.4: Digital input DI 4 * |
| 9 | DI 5 | I | +0.5: Digital input DI 5 |
| 10 | DI 6 | I | +0.6: Digital input DI 6 * |
| 11 | DI 7 | I | +0.7: Digital input DI 7 * |
| 12 | DI 8 | I | +1.0: Digital input DI 8 |
| 13 | DI 9 | I | +1.1: Digital input DI 9 * |
| 14 | DI 10 | I | +1.2: Digital input DI 10 * |
| 15 | DI 11 | I | +1.3: Digital input DI 11 * |
| 16 | DI 12 | I | +1.4: Digital input DI 12 |
| 17 | DI 13 | I | +1.5: Digital input DI 13 |
| 18 | DI 14 | I | +1.6: Digital input DI 14 |
| 19 | DI 15 | I | +1.7: Digital input DI 15 * |
| 20 | DC 24V | I | 5L+: DC 24V power section supply for on-board DI |
| 21 | 0 V | I | 5M: GND power section supply for on-board DI |

*) Max. input frequency 100kHz otherwise 1kHz.

5.4.3 Parametrization

5.4.3.1 Adress assignment

| Sub module | Input address | Access | Description |
|------------|---------------|--------|------------------------------------|
| DI24/DO16 | 136 | BYTE | Digital input I+0.0 ... I+0.7 (X4) |
| | 137 | BYTE | Digital input I+1.0 ... I+1.7 (X4) |

5.4.3.2 Hardware interrupt

Parameter hardware configuration

With the parameter *'Hardware interrupt at ...'* you can specify a hardware interrupt for each input for the corresponding edge. The hardware interrupt is disabled, if nothing is selected (default setting). A diagnostics interrupt is only supported with *Hardware interrupt lost*. Select with the arrow keys the input and enable the according hardware interrupts.

Here is valid:

- Rising edge: Edge 0-1
- Falling edge: Edge 1-0

5.4.3.3 Input delay

Parameter hardware configuration

- The input delay can be configured per channel in groups of 4.
- An input delay of 0.1ms is only possible with "fast" inputs, which have a max. input frequency of 100kHz ↪ *Chap. 5.4 'Digital input' page 123*. Within a group, the input delay for slow inputs is limited to 0.5ms.
- Range of values: 0.1ms / 0.5ms / 3ms / 15ms

5.4.4 Status indication

| Digital input | LED  green | Description |
|---------------------|--|--|
| DI +0.0 DI +0.7 |  | Digital I+0.0 ... 0.7 has "1" signal |
| |  | Digital I+0.0 ... 0.7 has "0" signal |
| DI +1.0 ... DI +1.7 |  | Digital I+1.0 ... 1.7 has "1" signal |
| |  | Digital input I+1.0 ... 1.7 has "0" signal |

| Power supply | LED  green | Description |
|--------------|--|--|
| 1L+ |  | DC 24V electronic section supply |
| |  | DC 24V electronic section supply not available |
| 2L+ |  | DC 24V power section supply outputs OK |
| |  | DC 24V power section supply outputs OK |
| 3L+ |  | DC 24V power section supply SLIO bus OK |
| |  | DC 24V power section supply SLIO bus not available |
| 5L+ |  | DC 24V power section supply inputs OK |
| |  | DC 24V power section supply inputs not available |

| Error | LED  red | Description |
|-------|--|---|
| 1F |  | Error power supply sensor |
| |  | No error |
| 2F |  | Error at overload respectively short circuit at the outputs |
| |  | No error |

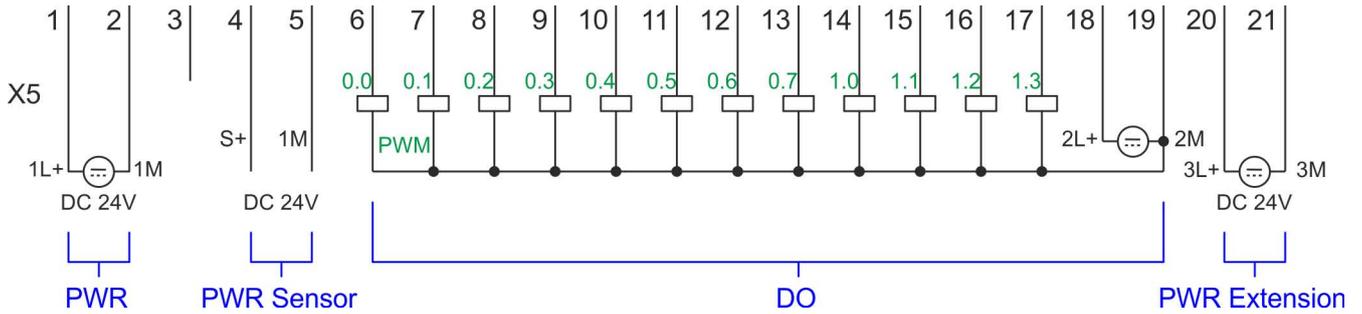
5.5 Digital output

5.5.1 Properties

- 12xDC 24V, 0.5A
- Status indication via LEDs

5.5.2 Wiring

X5: Connector



| X5 | Function | Type | Description |
|----|------------|------|--|
| 1 | Sys DC 24V | I | 1L+: DC 24V for electronic section supply |
| 2 | Sys 0V | I | 1M: GND for electronic section supply |
| 6 | DO 0 | O | +0.0: Digital output DO 0 |
| 7 | DO 1 | O | +0.1: Digital output DO 1 |
| 8 | DO 2 | O | +0.2: Digital output DO 2 |
| 9 | DO 3 | O | +0.3: Digital output DO 3 |
| 10 | DO 4 | O | +0.4: Digital output DO 4 |
| 11 | DO 5 | O | +0.5: Digital output DO 5 |
| 12 | DO 6 | O | +0.6: Digital output DO 6 |
| 13 | DO 7 | O | +0.7: Digital output DO 7 |
| 14 | DO 8 | O | +1.0: Digital output DO 8 |
| 15 | DO 9 | O | +1.1: Digital output DO 9 |
| 16 | DO 10 | O | +1.2: Digital output DO 10 |
| 17 | DO 11 | O | +1.3: Digital output DO 11 |
| 18 | DC 24V | I | 2L+: DC 24V power section supply for on-board DO |
| 19 | 0 V | I | 2M: GND power section supply for on-board DO |
| 20 | DC 24V | I | 3L+: DC 24V SLIO bus power section supply |
| 21 | 0 V | I | 3M: GND SLIO bus power section supply |

5.5.3 Parametrization

5.5.3.1 Address assignment

| Sub module | Output address | Access | Description |
|------------|----------------|--------|-------------------------------------|
| DI24/DO16 | 136 | BYTE | Digital output Q+0.0 ... Q+0.7 (X5) |
| | 137 | BYTE | Digital output Q+1.0 ... Q+1.3 (X5) |

5.5.4 Status indication

| Digital output | LED | Description |
|---------------------|---|---|
| |  green | |
| DO +0.0 ... DO +0.7 |  | Digital output Q+0.0 ... 0.7 has "1" signal |
| | <input type="checkbox"/> | Digital output Q+0.0 ... 0.7 has "0" signal |
| DO +1.0 ... DO +1.3 |  | Digital output Q+1.0 ... 1.3 has "1" signal |
| | <input type="checkbox"/> | Digital output Q+1.0 ... 1.3 has "0" signal |

| Power supply | LED | Description |
|--------------|---|--|
| |  green | |
| 1L+ |  | DC 24V electronic section supply |
| | <input type="checkbox"/> | DC 24V electronic section supply not available |
| 2L+ |  | DC 24V power section supply outputs OK |
| | <input type="checkbox"/> | DC 24V power section supply outputs OK |
| 3L+ |  | DC 24V power section supply SLIO bus OK |
| | <input type="checkbox"/> | DC 24V power section supply SLIO bus not available |
| 5L+ |  | DC 24V power section supply inputs OK |
| | <input type="checkbox"/> | DC 24V power section supply inputs not available |

| Error | LED | Description |
|-------|---|---|
| |  red | |
| 1F |  | Error power supply sensor |
| | <input type="checkbox"/> | no error |
| 2F |  | Error at overload respectively short circuit at the outputs |
| | <input type="checkbox"/> | no error |

5.6 Counting

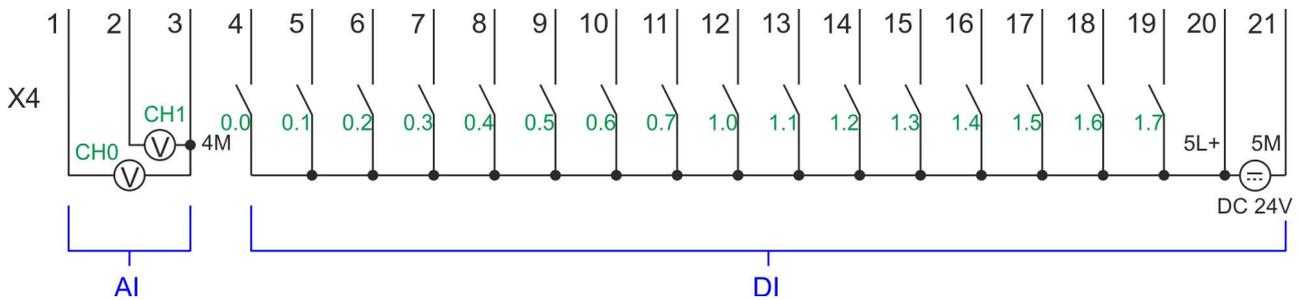
5.6.1 Properties

- 4 channels
- Various counting modes
 - once
 - continuously
 - periodically
- Control by the user program via blocks

5.6.2 Wiring

5.6.2.1 Counter inputs

X4: Connector



| X4 | Function | Type | Description |
|----|----------|------|--|
| 4 | DI 0 | I | +0.0: Counter 0 (A) * |
| 5 | DI 1 | I | +0.1: Counter 0 (B) * |
| 7 | DI 3 | I | +0.3: Counter 1 (A) * |
| 8 | DI 4 | I | +0.4: Counter 1 (B) * |
| 10 | DI 6 | I | +0.6: Counter 2 (A) * |
| 11 | DI 7 | I | +0.7: Counter 2 (B) * |
| 13 | DI 9 | I | +1.1: Counter 3 (A) * |
| 14 | DI 10 | I | +1.2: Counter 3 (B) * |
| 15 | DI 11 | I | +1.3: Gate 3 * |
| 19 | DI 15 | I | +1.7: Latch 3 * |
| 20 | DC 24V | I | 5L+: DC 24V power section supply for counter |
| 21 | 0 V | I | 5M: GND power section supply for counter |

*) Max. input frequency 100kHz otherwise 1kHz.

Input signals

The following sensors can be connected

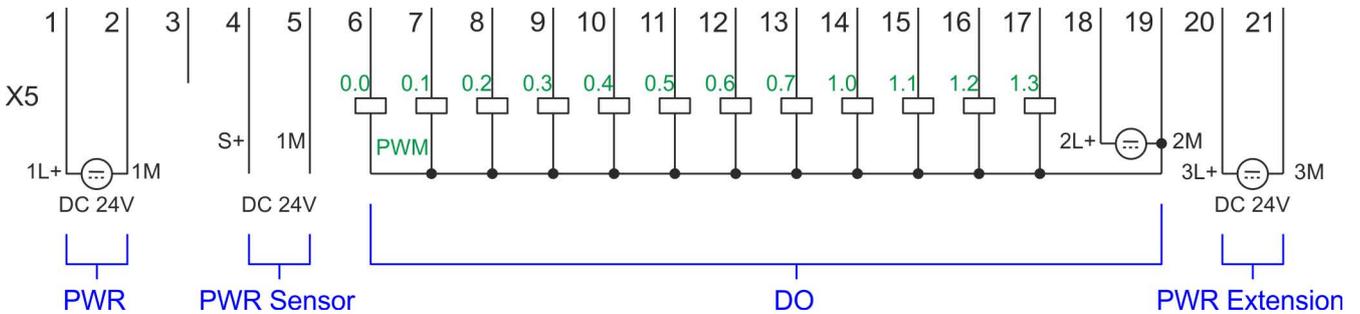
- 24V incremental encoders with two phase-shifted by 90 ° tracks
- 24V pulse encoder with direction signal
- 24V initiator as BERO or beam sensor

For not all inputs are available at the same time, for every counter you may define the input assignment via the parameterization for the following input signals:

- **Counter_x (A)**
 - Pulse input for counter signal respectively track A of an encoder for 1-, 2- or 4-fold evaluation.
- **Counter_x (B)**
 - Direction signal respectively track B of the encoder. Via the parameterization you may invert the direction signal.
- **Gate 3**
 - Via this input you can if parameterized open the HW gate of *Counter 3* with edge 0-1 and start counting.
- **Latch 3**
 - Via this input via edge 0-1 the current counter value of *Counter 3* is stored in a memory that you may read if needed.

5.6.2.2 Counter outputs

X5: Connector



| X5 | Function | Type | Description |
|----|------------|------|---|
| 1 | Sys DC 24V | I | 1L+: DC 24V for electronic section supply |
| 2 | Sys 0V | I | 1M: GND for electronic section supply |
| 6 | DO 0 | O | +0.0: Output channel counter 0 |
| 7 | DO 1 | O | +0.1: Output channel counter 1 |
| 8 | DO 2 | O | +0.2: Output channel counter 2 |
| 9 | DO 3 | O | +0.3: Output channel counter 3 |
| 18 | DC 24V | I | 2L+: DC 24V power section supply for on-board counter |
| 19 | 0 V | I | 2M: GND power section supply for on-board counter |
| 20 | DC 24V | I | 3L+: DC 24V SLIO bus power section supply |
| 21 | 0 V | I | 3M: GND SLIO bus power section supply |

Output channel Counter_x

Every counter has an assigned output channel. For each counter you can specify the behavior of the counter output via the parameterization with 'Characteristics of the output' and 'Pulse duration'. ↪ Chap. 5.6.4.3 'Counter' page 131

5.6.3 Proceeding

Hardware configuration

In the Siemens SIMATIC Manager the following steps should be executed:

1. ➤ Perform a hardware configuration for the CPU. ↪ *Chap. 4.5 'Hardware configuration - CPU' page 69*
2. ➤ Double-click the counter sub module of the CPU CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
⇒ The dialog 'Properties' is opened.
3. ➤ As soon as you select the operating mode for the corresponding channel, a dialog box with default values for this counter mode is created and shown.
4. ➤ Perform the required parameter settings.
5. ➤ Save your project with 'Station → Safe and compile'.
6. ➤ Transfer your project to your CPU.

User program

- The SFB 47 should cyclically be called (e.g. OB 1) for controlling the counter functions.
- The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.
- Among others the SFB 47 contains a request interface. Hereby you get read and write access to the registers of the appropriate counter.
- So that a new job may be executed, the previous job must have be finished with JOB_DONE = TRUE.
- Per channel you may call the SFB in each case with the same instance DB, since the data necessary for the internal operational are stored here.
- Writing accesses to outputs of the instance DB is not permissible.
- Starting, stopping and interrupting a count function of *Counter 0* to *Counter 2* exclusively happens via the SW gate by setting the SW gate of the SFB 47. You can also activate input 'Gate 3' via the parametrization for *Counter 3*.



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

5.6.4 Parametrization

5.6.4.1 Address assignment

| Sub module | Input address | Access | Description |
|----------------|---------------|--------|--------------------------|
| <i>Counter</i> | 816 | DINT | Channel 0: Counter value |
| | 820 | DINT | Channel 1: Counter value |
| | 824 | DINT | Channel 2: Counter value |
| | 828 | DINT | Channel 3: Counter value |

| Sub module | Output address | Access | Description |
|------------|----------------|--------|-------------|
| Counter | 816 | DWORD | reserved |
| | 820 | DWORD | reserved |
| | 824 | DWORD | reserved |
| | 828 | DWORD | reserved |

5.6.4.2 Interrupt selection

Via 'Basic parameters' you can reach 'Select interrupt'. Here you can define the interrupts the CPU will trigger. The following parameters are supported:

- None: The interrupt function is disabled.
- Process: The following events of the counter can trigger a hardware interrupt (selectable via 'Count'):
 - Hardware gate opening
 - Hardware gate closing
 - On reaching the comparator
 - on Counting pulse
 - on overflow
 - on underflow
- Diagnostics+process: A diagnostics interrupt is only triggered when a hardware interrupt was lost.

5.6.4.3 Counter

Parameter hardware configuration

Default values and structure of this dialog box depend on the selected 'Operating mode'.



Please consider that the range of values could be limited due to the used projecting tool. With the VIPA SPEED7 Studio there are no limitations.
 ↪ Chap. 12 'Configuration with VIPA SPEED7 Studio' page 267

Parameter overview

| Operating parameters | Description | Assignment |
|----------------------|--|--|
| Main count direction | <ul style="list-style-type: none"> ■ <i>None</i> No restriction of the counting range ■ <i>Up</i>: Restricts the up-counting range. The counter starts from 0 or <i>load value</i>, counts in positive direction up to the declaration <i>end value</i> -1 and then jumps back to <i>load value</i> at the next positive transducer pulse. ■ <i>Down</i>: Restricts the down-counting range. The counter starts from the declared <i>start value</i> or <i>load value</i> in negative direction, counts to 1 and then jumps to <i>start value</i> at the next negative encoder pulse. Function is disable with <i>count continuously</i>. | <ul style="list-style-type: none"> ■ None |
| Gate function | <ul style="list-style-type: none"> ■ <i>Cancel count</i>: The count starts when the gate opens and resumes at the <i>load value</i> when the gate opens again. ■ <i>Stop count</i>: The count is interrupted when the gate closes and resumed at the last actual counter value when the gate opens again. <p>🔗 <i>Chap. 5.6.6.2 'Gate function' page 142</i></p> | Abort count process |
| Start value | <i>Start value</i> with counting direction backward. | 2147483647 ($2^{31}-1$) |
| End value | <i>End value</i> with main counting direction forward. Range of values: 2...2147483647 ($2^{31}-1$) | |
| Comparison value | <p>The count value is compared with the <i>comparison value</i>. See also the parameter "Characteristics of the output":</p> <ul style="list-style-type: none"> ■ No main counting direction <ul style="list-style-type: none"> – Range of values: -2^{31} to $+2^{31}-1$ ■ Main counting direction forward <ul style="list-style-type: none"> – Range of values: -2^{31} to end value-1 ■ Main counting direction backward <ul style="list-style-type: none"> – Range of values: 1 to $+2^{31}-1$ | 0 |
| Hysteresis | <p>The <i>hysteresis</i> serves the avoidance of many toggle processes of the output, if the counter value is in the range of the <i>comparison value</i>.</p> <p>0, 1: <i>Hysteresis</i> disabled</p> <p>Range of values: 0 to 255</p> | 0 |

| Input | Description | Assignment |
|--------------------------|---|-----------------|
| Signal evaluation | Specify the signal of the connected encoder: <ul style="list-style-type: none"> ■ Pulse/direction At the input count and direction signal are connected ■ At the input there is an encoder connected with the following evaluation: <ul style="list-style-type: none"> – Rotary encoder single – Rotary encoder double – Rotary encoder quadruple | Pulse/direction |
| Hardware gate | Gate control exclusively via channel 3: <ul style="list-style-type: none"> ■ enabled: The gate control for channel 3 happens via SW and HW gate ■ disabled: The gate control for channel 3 exclusively happens via SW gate ↗ <i>Chap. 5.6.6.2 'Gate function' page 142</i> | disabled |
| Count direction inverted | Invert the input signal ' <i>Direction</i> ': <ul style="list-style-type: none"> ■ enabled: The input signal is inverted ■ disabled: The input signal is not inverted | disabled |

| Output | Description | Assignment |
|-------------------------------|---|---------------|
| Characteristics of the output | The output and the "Comparator" (STS_CMP) status bit are set, dependent on this parameter. <ul style="list-style-type: none"> ■ No comparison: The output is used as normal output and STS_CMP remains reset. ■ Comparator <ul style="list-style-type: none"> – Counter value \geq Comparison value – Counter value \leq Comparison value ■ Pulse at <i>comparison value</i> <ul style="list-style-type: none"> – To adapt the used actuators you can specify a <i>pulse duration</i>. The output is set for the specified <i>pulse duration</i> when the counter value reaches the <i>comparison value</i>. When you've set a main counting direction the output is only set at reaching the <i>comparison value</i> from the main counting direction. | No comparison |
| Pulse duration | Here you can specify the <i>pulse duration</i> for the output signal. <ul style="list-style-type: none"> ■ The <i>pulse duration</i> starts with the setting of the according digital output. ■ The inaccuracy of the <i>pulse duration</i> is less than 1ms. ■ There is no past triggering of the <i>pulse duration</i> when the <i>comparison value</i> has been left and reached again during pulse output. ■ If the <i>pulse duration</i> is changed during operation, it will take effect with the next pulse. ■ If the <i>pulse duration</i> = 0, the output is set until the comparison condition is not longer fulfilled. Range of values: 0...510ms in steps of 2ms | 0 |

| Hardware interrupt | Description | Assignment |
|------------------------|--|------------|
| Hardware gate opening | Hardware interrupt by edge 0-1 exclusively at HW gate channel 3 <ul style="list-style-type: none"> ■ enabled: Process interrupt by edge 0-1 exclusively at HW gate channel 3 with open SW gate ■ disabled: no hardware interrupt | disabled |
| Hardware gate closing | Hardware interrupt by edge 1-0 exclusively at HW gate channel 3 <ul style="list-style-type: none"> ■ enabled: Process interrupt by edge 1-0 exclusively at HW gate channel 3 with open SW gate ■ disabled: no hardware interrupt | disabled |
| On reaching comparator | Hardware interrupt on reaching <i>comparator</i> <ul style="list-style-type: none"> ■ enabled: Hardware interrupt when comparator is triggered, can be configured via '<i>Characteristics of the output</i>' ■ disabled: no hardware interrupt | disabled |
| Overflow | Hardware interrupt overflow <ul style="list-style-type: none"> ■ enabled: Hardware interrupt on overflow the upper counter limit ■ disabled: no hardware interrupt | disabled |
| Underflow | Hardware interrupt on underrun <ul style="list-style-type: none"> ■ enabled: Hardware interrupt on underflow the lower counter limit ■ disabled: no hardware interrupt | disabled |

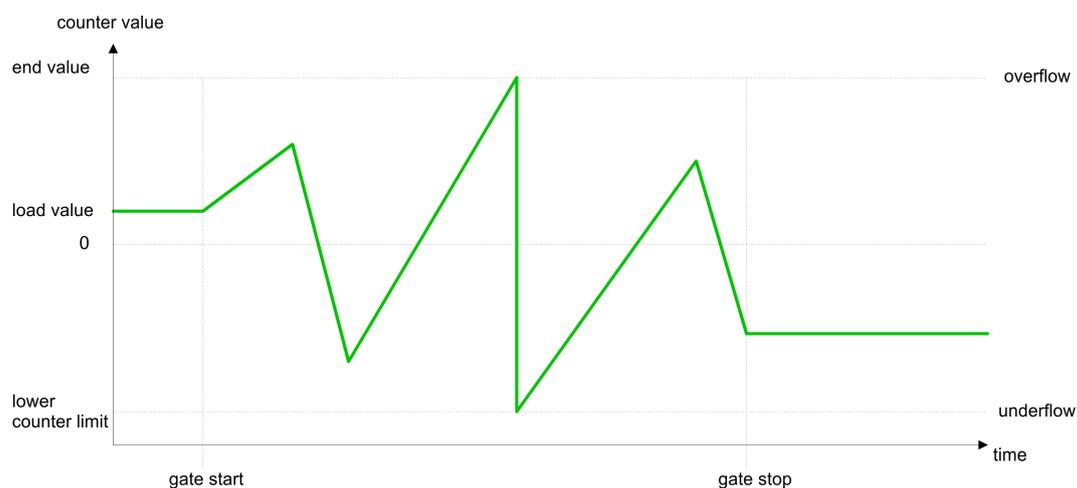
| Max. frequency | Description | Assignment | |
|--------------------------|---|------------|----------------------------------|
| Counting signals/HW gate | Specify the max. frequency for track A/pulse, track B/direction and HW gate | 60kHz | |
| | Frequency | | shortest permissible count pulse |
| | 1kHz | | 400µs |
| | 2kHz | | 200µs |
| | 5kHz | | 80µs |
| | 10kHz | | 40µs |
| | 30kHz | | 13µs |
| | 60kHz | | 6.7µs |
| Latch | Specify the max. frequency for the latch signal | 10kHz | |
| | Frequency | | shortest permissible latch pulse |
| | 1kHz | | 400µs |
| | 2kHz | | 200µs |
| | 5kHz | | 80µs |
| | 10kHz | | 40µs |
| | 30kHz | | 13µs |
| | 60kHz | | 6.7µs |

5.6.5 Counter operating modes

5.6.5.1 Count continuously

- In this operating mode the counter counts starting with the *load value*.
- When the counter counts forward and reaches the upper count limit and another counting pulse in positive direction arrives, it jumps to the lower count limit and counts from there on.
- When the counter counts backwards and reaches the lower count limit and another counting pulse in negative direction arrives, it jumps to the upper count limit and counts from there on.
- The counter limits are fix set to maximum range.
- With overflow or underflow the status bits STS_OFLW respectively STS_UFLW are set in SFB 47. These bits remain set until these are reset with RES_STS. If enabled additionally a hardware interrupt is triggered.

| Limits | Valid range of values |
|-------------------|---------------------------------|
| Lower count limit | -2 147 483 648 (-2^{31}) |
| Upper count limit | +2 147 483 647 ($2^{31} - 1$) |



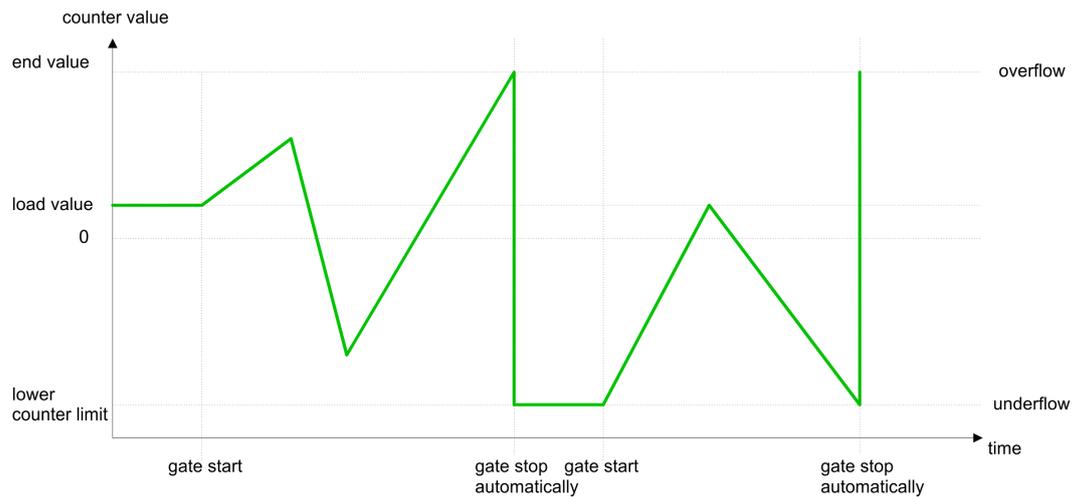
5.6.5.2 Count once

5.6.5.2.1 No main counting direction

- The counter counts once starting with *load value*.
- It is counted forward or backward.
- The counter limits are fix set to maximum range.
- At over- or underflow at the count limits, the counter jumps to the according other count limit and the gate is automatically closed.
- To restart the count process, you have to generate an edge 0-1 at the gate ↪ Chap. 5.6.6.2 'Gate function' page 142.
- With the configured 'Gate function' 'Interrupt count' the counting is continued with current *Counter value*.
- With configured 'Gate function' 'Cancel count' the counter starts with the *Load value*.

| Limits | Valid range of values |
|-------------------|-------------------------------------|
| Lower count limit | -2 147 483 648 (-2 ³¹) |
| Upper count limit | +2 147 483 647 (2 ³¹ -1) |

Interrupting gate control



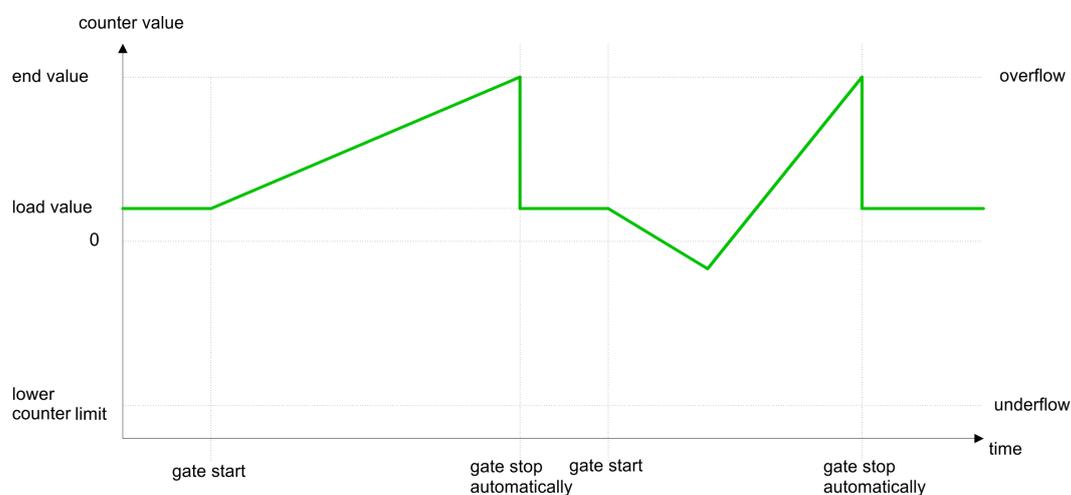
Aborting gate control



5.6.5.2.2 Main counting direction forward

- The counter counts forward starting with the *load value*.
- When the counter reaches the *End value* -1 in positive direction, it jumps to the *load value* at the next count pulse and the gate is automatically closed.
- To restart the count process, you have to generate an edge 0-1 at the gate ↪ *Chap. 5.6.6.2 'Gate function' page 142*. The counter counts starting with the *load value*.
- You may exceed the lower count limit.

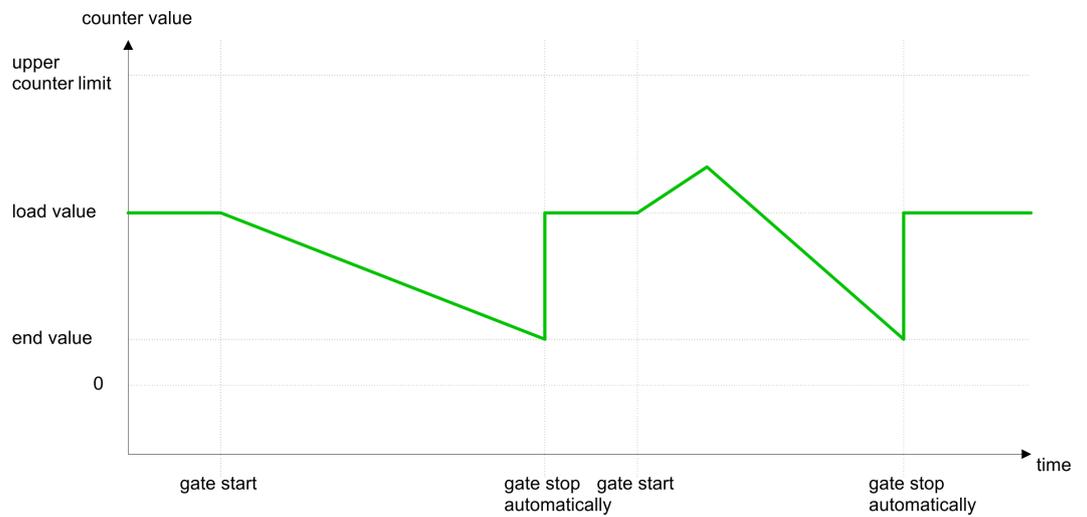
| Limits | Valid range of values |
|-------------------|---|
| End value | -2 147 483 647 ($-2^{31} + 1$) up to +2 147 483 647 ($2^{31} - 1$) |
| Lower count limit | -2 147 483 648 (-2^{31}) |



5.6.5.2.3 Main counting direction backward

- The counter counts backward starting with the *load value*.
- When the counter reaches the *End value* +1 in positive direction, it jumps to the *load value* at the next count pulse and the gate is automatically closed.
- To restart the count process, you have to generate an edge 0-1 at the gate ↪ Chap. 5.6.6.2 'Gate function' page 142. The counter counts starting with the *load value*.
- You may exceed the upper count limit.

| Limits | Valid range of values |
|-------------------|---|
| End value | -2 147 483 648 (-2^{31}) up to +2 147 483 646 ($2^{31} - 2$) |
| Upper count limit | +2 147 483 647 ($2^{31} - 1$) |



5.6.5.3 Count periodically

5.6.5.3.1 No main counting direction

- The counter counts forward or backwards starting with the *load value*.
- At over- or underrun at the count limits, the counter jumps to the *load value* and continues counting. If enabled additionally a hardware interrupt is triggered.
- The counter limits are fix set to maximum range.

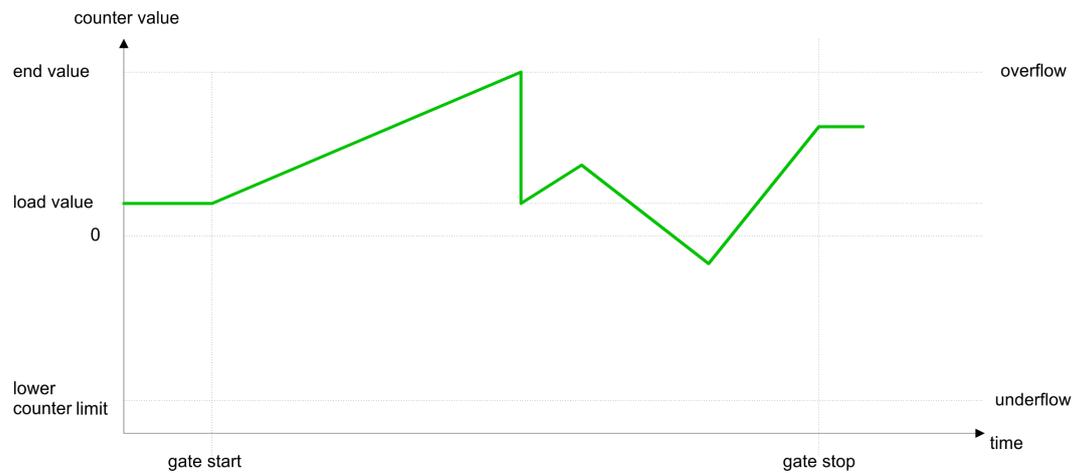
| Limits | Valid range of values |
|-------------------|---------------------------------|
| Lower count limit | -2 147 483 648 (-2^{31}) |
| Upper count limit | +2 147 483 647 ($2^{31} - 1$) |



5.6.5.3.2 Main counting direction forward

- The counter counts forward starting with the *load value*.
- When the counter reaches the end value -1 in positive direction, it jumps to the *load value* at the next positive count pulse and continues counting. If enabled additionally a hardware interrupt is triggered.
- You may exceed the lower count limit.

| Limits | Valid range of values |
|-------------------|---|
| End value | -2 147 483 647 ($-2^{31} + 1$) up to +2 147 483 647 ($2^{31} - 1$) |
| Lower count limit | -2 147 483 648 (-2^{31}) |

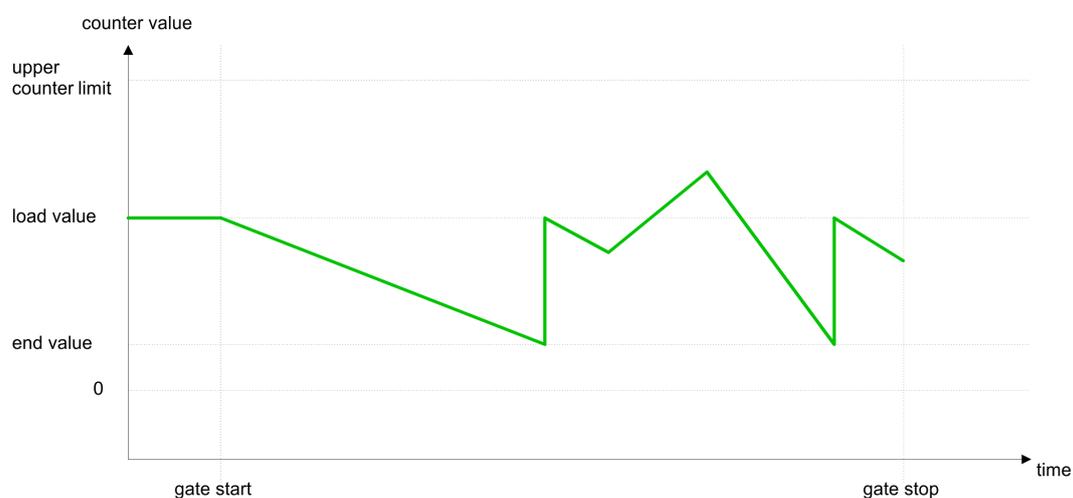


5.6.5.3.3 Main counting direction backward

Main counting direction backward

- The counter counts backward starting with the *load value*.
- When the counter reaches the *end value* +1 in positive direction, it jumps to the *load value* at the next negative count pulse and continues counting. If enabled additionally a hardware interrupt is triggered.
- You may exceed the upper count limit.

| Limits | Valid range of values |
|-------------------|---|
| End value | -2 147 483 648 (-2^{31}) up to +2 147 483 646 ($2^{31} - 2$) |
| Upper count limit | +2 147 483 647 ($2^{31} - 1$) |

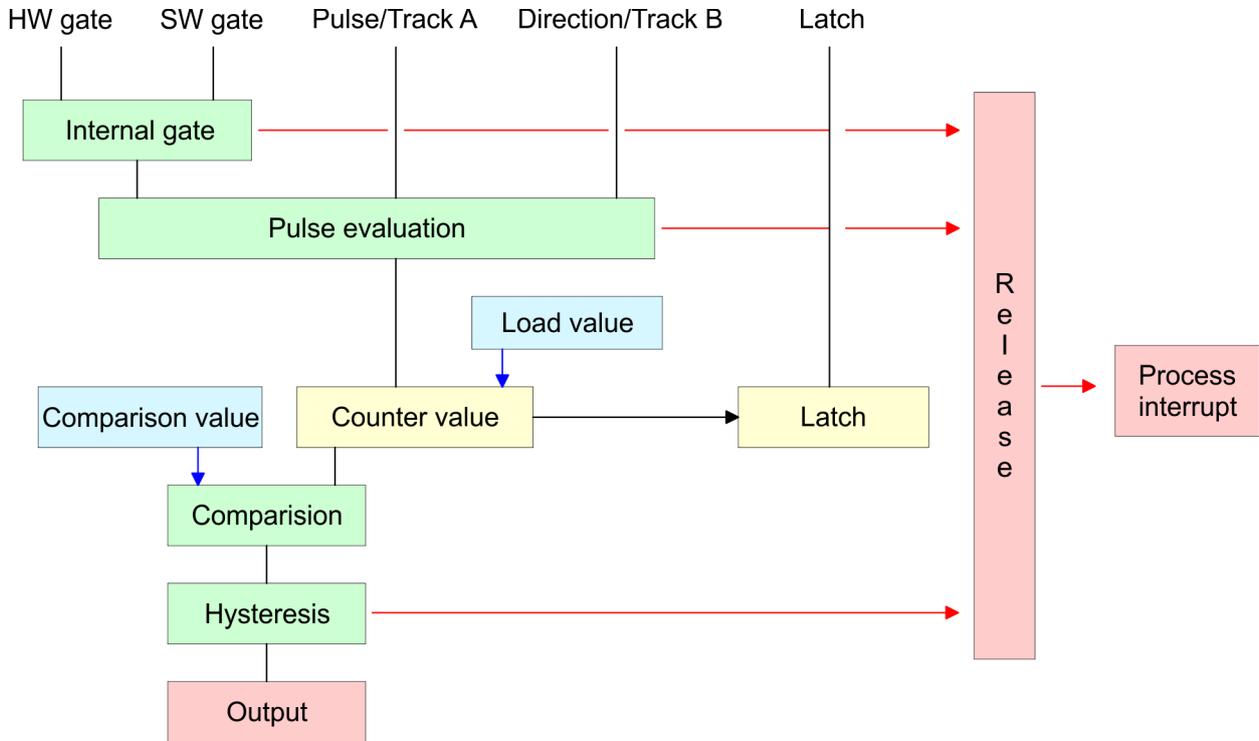


5.6.6 Counter - Additional functions

5.6.6.1 Overview

Schematic structure

The illustration shows how the additional functions influence the counting behavior. The following pages describe these additional functions in detail:



5.6.6.2 Gate function

Function

- Starting, stopping and interrupting a count function of *counter 0* to *counter 2* exclusively happens via the SW gate by setting the SW gate of SFB 47.
- Starting, stopping and interrupting a count function of *counter 3* happens via the internal gate (I gate). The i gate is the result of logic operation of HW gate and SW gate. The HW gate evaluation of the connection 'Gate 3' may be deactivated by the parametrization. With a de-activated HW gate evaluation the triggering exclusively happens by setting the SW gate of SFB 47.

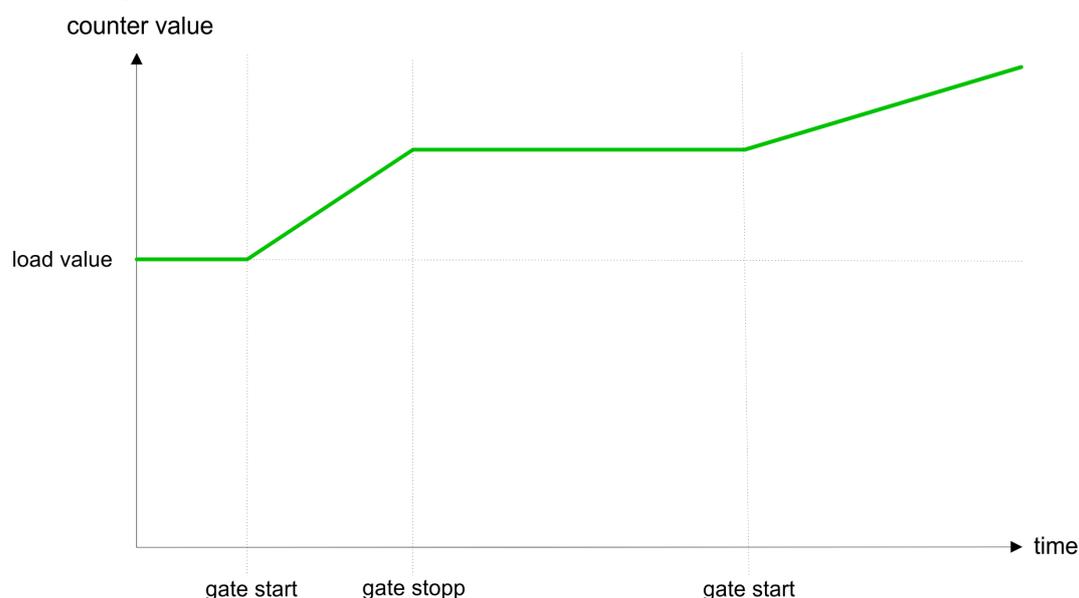
Gate function abort and interrupt

The parametrization defines if the gate interrupts or aborts the counter process.

- At *abort function* the counter starts counting with the *load value* after gate restart.



- At *interrupt function*, the counter starts counting with the last recent counter value after gate restart.



Counter 0 ... 2

| SW gate | Gate function | Reaction counter 0 ... 2 |
|----------|-------------------------|--------------------------------|
| Edge 0-1 | Abort count process | Restart with <i>load value</i> |
| Edge 0-1 | Interrupt count process | Continue |

5.6.6.3 Comparator

Function

In the CPU a *comparison value* may be stored. During the counting procedure the counter value is compared with the *comparative value*. Depending on the result of the comparison the output channel of the counter and the status bit of STS_CMP of SFB 47 can be set. In addition, you can configure a hardware interrupt. A *comparison value* can be specified via the parametrization respectively the job interface of SFB 47.

5.6.6.4 Additional functions counter 3

Exclusively counter 3 has the following additional functions:

- HW gate via *Gate 3*
- Latch function

5.6.6.4.1 HW gate via *Gate 3*

Starting, stopping and interrupting a count function of counter 3 happens via the internal gate (I gate). The i gate is the result of logic operation of HW gate and SW gate. The HW gate evaluation of the connection '*Gate 3*' may be deactivated by the parametrization. With a de-activated HW gate evaluation the triggering exclusively happens by setting the SW gate of SFB 47.

Counter 3:

| SW gate | HW gate | Gate function | Reaction counter 3: |
|----------|--------------|-------------------------|-------------------------|
| Edge 0-1 | de-activated | Abort count process | Restart with load value |
| Edge 0-1 | de-activated | Interrupt count process | Continue |
| Edge 0-1 | 1 | Abort count process | Continue |
| 1 | Edge 0-1 | Abort count process | Restart with load value |
| Edge 0-1 | 1 | Interrupt count process | Continue |
| 1 | Edge 0-1 | Interrupt count process | Continue |

Counter 3 - count once

If the internal gate has been closed automatically it may only be opened again under the following conditions:

| SW gate | HW gate | I gate |
|--------------------------------------|----------|--------|
| 1 | Edge 0-1 | 1 |
| Edge 0-1 (after edge 0-1 at HW gate) | Edge 0-1 | 1 |

5.6.6.4.2 Latch function

Function

- As soon as during a count process an edge 0-1 is recognized at the "Latch" input of counter 3, the current counter value is stored in the according latch register.
- You may access the latch value via the parameter LATCHVAL of SFB 47.
- A just in LATCHVAL loaded value remains after a STOP-RUN transition.

5.6.6.5 Counter output channel

Characteristics of the output

Each counter has an output channel. You pre-define the behavior of the counter output via the parametrization:

- no comparison:
 - The output is used as normal output.
 - SFB 47:
 - The input parameter CTRL_DO is effect less.
 - The status bits STS_DO and STS_CMP (status comparator in the instance DB) remain reset.
- Counter value \geq comparison value respectively counter value \leq comparison value
 - The output remains set as long as the counter value is higher or equal *comparison value* respectively lower or equal *comparison value*.
 - SFB 47:
 - Control bit CTRL_DO must be set.
 - The comparison result is shown by the status bit STS_CMP. This status bit may only be reset if the comparison condition is no longer fulfilled.
- Pulse at comparison value
 - When the counter reaches the *comparison value* the output is set for the parametrized *pulse duration*. When you've set a main counting direction the output is only set at reaching the *comparison value* from the main counting direction. If the *pulse duration* = 0, the output is set until the comparison condition is not longer fulfilled.
 - SFB 47:
 - Control bit CTRL_DO must be set.
 - The status of the digital output may be shown by the status bit ST_DO.
 - The comparison result is shown by the status bit STS_CMP. The bit may only be reset if the *pulse duration* has expired.
- Pulse duration
 - The *pulse duration* starts with the setting of the according digital output.
 - The inaccuracy of the *pulse duration* is less than 1ms.
 - There is no past triggering of the *pulse duration* when the *comparison value* has been left and reached again during pulse output.
 - If the *pulse duration* is changed during operation, it will take effect with the next pulse.
 - If the *pulse duration* = 0, the output is set until the comparison condition is not longer fulfilled.
 - Range of values: 0...510ms in steps of 2ms

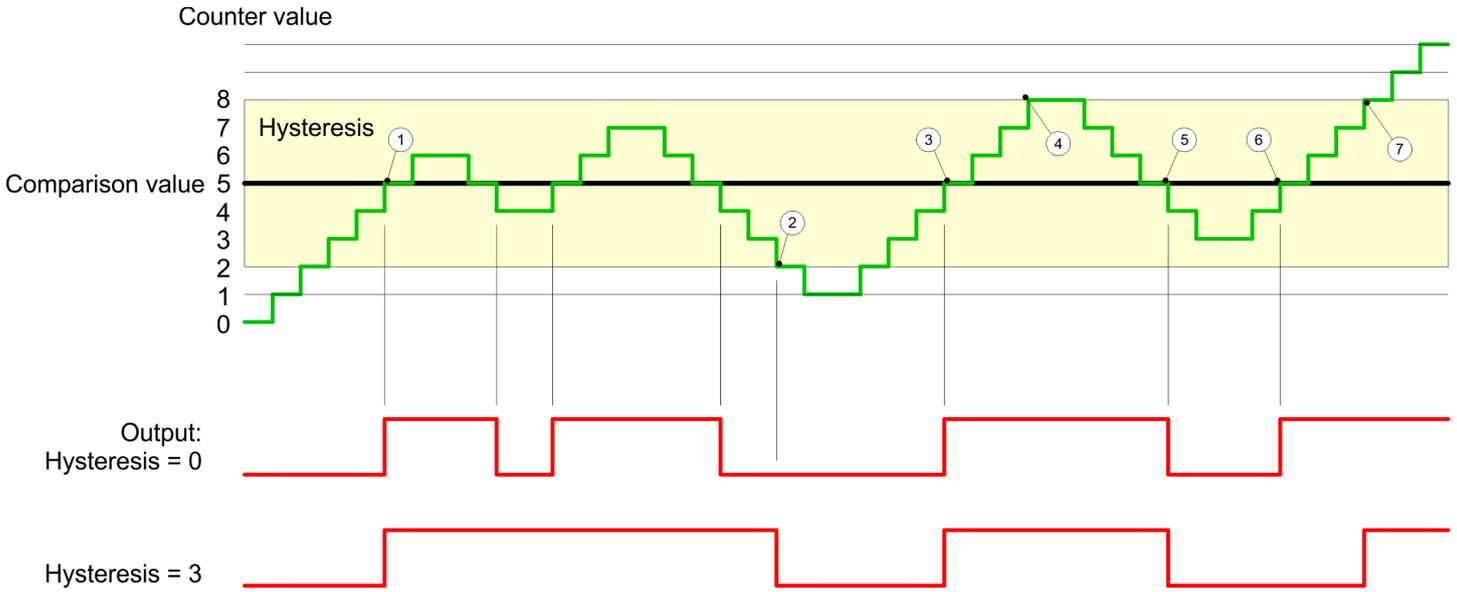
5.6.6.6 Hysteresis function

Hysteresis

- The *hysteresis* serves the avoidance of many toggle processes of the output and the interrupt, if the *counter value* is in the range of the *comparison value*.
- For the *hysteresis* you may set a range of 0 to 255.
- The settings 0 and 1 deactivate the *hysteresis*.
- The *hysteresis* influences zero run, comparison, over- and underflow.
- An activated *hysteresis* remains active after a change. The new *hysteresis* range is activated with the next *hysteresis* event.

The following pictures illustrate the output behavior for *hysteresis* 0 and *hysteresis* 3 for the according conditions:

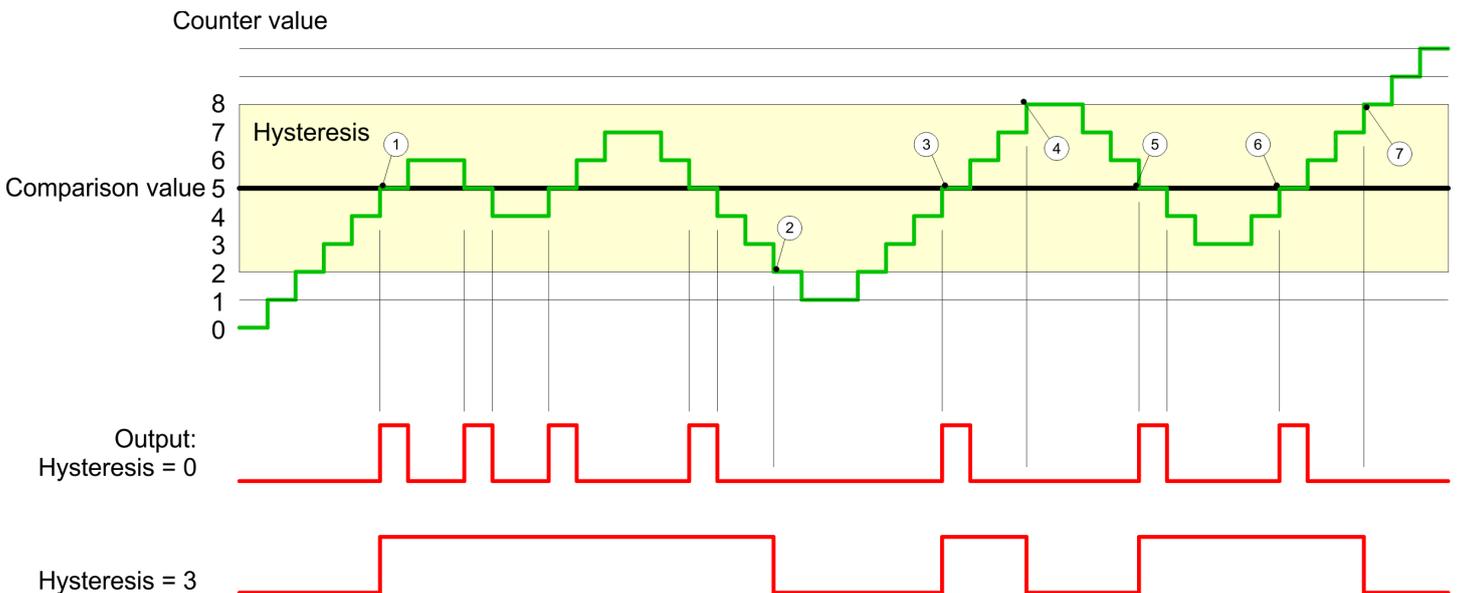
Effect at counter value \geq comparison value



- 1 Counter value \geq comparison value \rightarrow output is set and *hysteresis* activated
- 2 Leave *hysteresis* range \rightarrow output is reset
- 3 Counter value \geq comparison value \rightarrow output is set and *hysteresis* activated
- 4 Leave *hysteresis* range, output remains set for counter value \geq comparison value
- 5 counter value $<$ comparison value and *hysteresis* active \rightarrow output is reset
- 6 counter value \geq comparison value \rightarrow output is not set for *hysteresis* active
- 7 Leave *hysteresis* range, output remains set for counter value \geq comparison value

With reaching the comparison condition the *hysteresis* gets active. At active *hysteresis* the comparison result remains unchanged until the *counter value* leaves the set *hysteresis* range. After leaving the *hysteresis* range a new *hysteresis* is only activated with again reaching the comparison conditions.

Effect at pulse at comparison value with pulse duration Zero

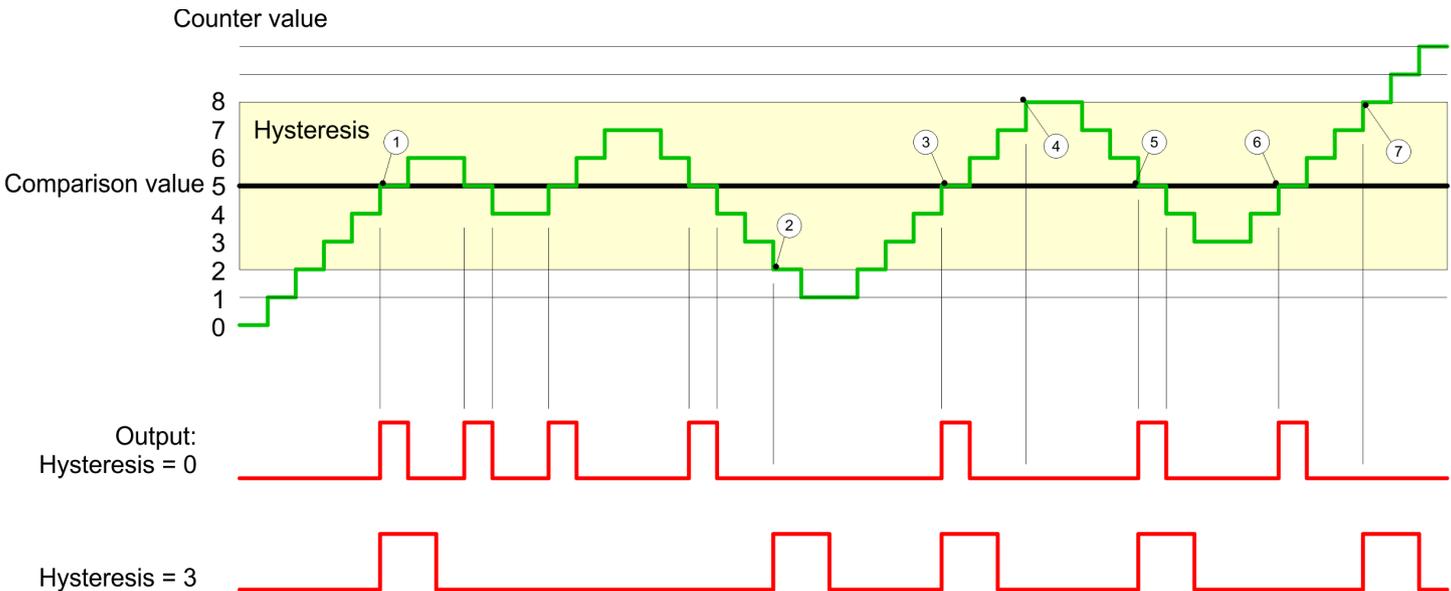


- 1 Counter value = comparison value \rightarrow output is set and *hysteresis* activated
- 2 Leave *hysteresis* range \rightarrow output is reset and counter value $<$ comparison value

- 3 Counter value = comparison value → output is set and hysteresis activated
- 4 Output is reset for leaving hysteresis range and counter value > comparison value
- 5 Counter value = comparison value → output is set and hysteresis activated
- 6 Counter value = comparison value and hysteresis active → output remains set
- 7 Leave hysteresis range and counter value > comparison value → output is reset

With reaching the comparison condition the hysteresis gets active. At active hysteresis the comparison result remains unchanged until the counter value leaves the set hysteresis range. After leaving the hysteresis range a new hysteresis is only activated with again reaching the comparison conditions.

Effect at pulse at comparison value with pulse duration not zero



- 1 Counter value = comparison value → pulse of the parameterized pulse duration is put out, the hysteresis is activated and the counting direction stored
- 2 Leaving the hysteresis range contrary to the stored counting direction → pulse of the parameterized pulse duration is put out, the hysteresis is de-activated
- 3 Counter value = comparison value → pulse of the parameterized pulse duration is put out, the hysteresis is activated and the counting direction stored
- 4 Leaving the hysteresis range without changing counting direction → hysteresis is de-activated
- 5 Counter value = comparison value → pulse of the parameterized pulse duration is put out, the hysteresis is activated and the counting direction stored
- 6 Counter value = comparison value and hysteresis active → no pulse
- 7 Leaving the hysteresis range contrary to the stored counting direction → pulse of the parameterized pulse duration is put out, the hysteresis is de-activated

With reaching the comparison condition the hysteresis gets active and a pulse of the parameterized duration is put out. As long as the counter value is within the hysteresis range, no other pulse is put out. With activating the hysteresis the counting direction is stored in the module. If the counter value leaves the hysteresis range contrary to the stored counting direction, a pulse of the parameterized duration is put out. Leaving the hysteresis range without direction change, no pulse is put out.

5.6.7 Diagnostics and interrupt

Overview

GSDML

- Edge at a digital interrupt input

Via the hardware configuration you can define the following trigger for a hardware interrupt that can trigger a diagnostics interrupt:

- Reaching the comparison value
- Overflow respectively at overrun upper counter limit
- Underflow respectively at underrun lower counter limit
- Opening the HW gate with open SW gate - except for counter 3
- Closing the HW gate with open SW gate - except for counter 3

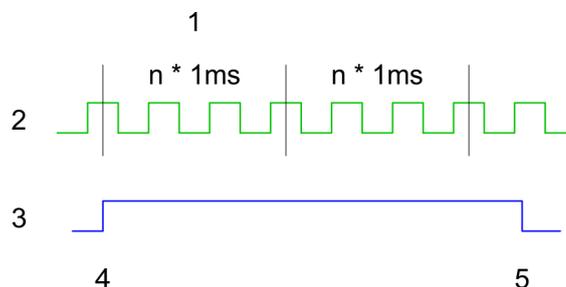
5.7 Frequency measurement

5.7.1 Properties

- In this operating mode the CPU counts the incoming pulses during a specified integration time and outputs them as frequency value.
- Integration time 10ms ... 10000ms in steps of 1ms configurable
- Control by the user program via SFB 48



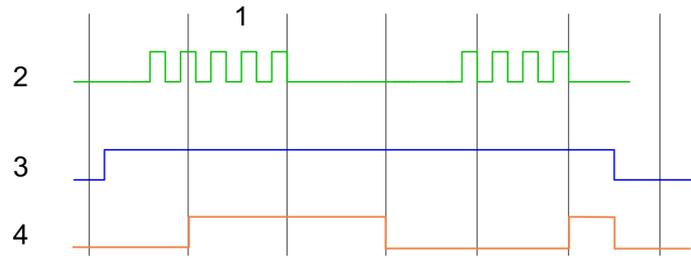
More information about the usage of this block may be found in the manual "SPEED7 Operation List" from VIPA.



- 1 Integration time
- 2 Counting pulse
- 3 SW gate
- 4 Frequency measurement start
- 5 Frequency measurement stop

Measuring procedure

- The measurement is carried out during the integration time and is updated after the integration time has expired.
- If the period of the measured frequency exceeds the assigned integration time, this means there was no edge 0-1 during the measurement, the measurement value 0 is returned.
- The calculated frequency value is supplied in "mHz" units.
- The measurement value can be read with `MEAS_VAL` from SFB 48.
- As long as the SW gate is open, you can request the calculated frequency.
- The number of activated channels does not influence the max. frequency, which is defined in the technical data.



- 1 Integration time
- 2 Counting pulse
- 3 SW gate
- 4 Evaluated frequency



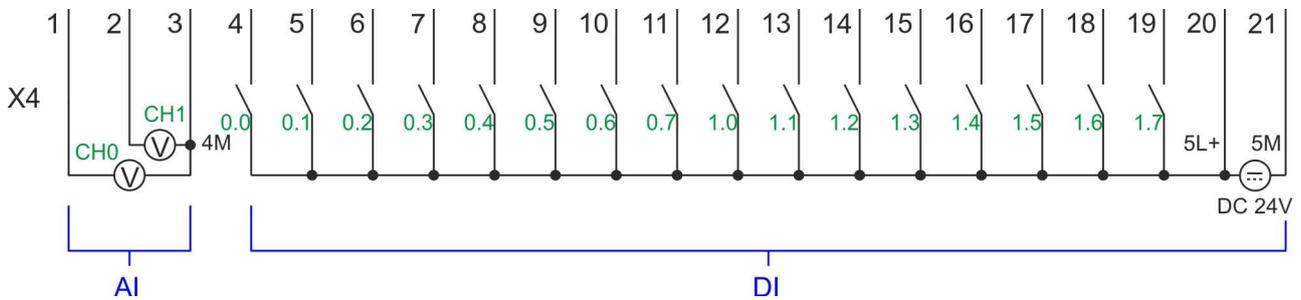
The counting function is disabled during the pulse width modulation on the same channel.

5.7.2 Wiring

5.7.2.1 Frequency measurement inputs

For frequency measurement, connect the signal to be measured to the B input of the corresponding counter.

X4: Connector



| X4 | Function | Type | Description |
|----|----------|------|--|
| 5 | DI 1 | I | +0.1: Frequency 0 * |
| 8 | DI 4 | I | +0.4: Frequency 1 * |
| 11 | DI 7 | I | +0.7: Frequency 2 * |
| 14 | DI 10 | I | +1.2: Frequency 3 * |
| 20 | DC 24V | I | 5L+: DC 24V power supply for frequency measurement |
| 21 | 0 V | I | 5M: GND power supply for frequency measurement |

*) Max. input frequency 100kHz otherwise 1kHz.

5.7.3 Proceeding

Hardware configuration

In the Siemens SIMATIC Manager the following steps should be executed:

1. ➤ Perform a hardware configuration for the CPU. ↪ *Chap. 4.5 'Hardware configuration - CPU' page 69*
2. ➤ Double-click the counter sub module of the CPU 314C-2 PN/DP.
⇒ The dialog '*Properties*' is opened.
3. ➤ As soon as you select the operating mode for the corresponding channel, a dialog box with default values for this counter mode is created and shown. Select for the corresponding channel the operating mode '*Frequency counting*'.
4. ➤ Perform the required parameter settings.
5. ➤ Save your project with '*Station → Safe and compile*'.
6. ➤ Transfer your project to your CPU.

User program

- The SFB 48 should cyclically be called (e.g. OB 1) for controlling the frequency measurement.
- The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.

5.7.4 Parametrization

5.7.4.1 Address assignment

| Sub module | Input address | Access | Description |
|----------------|---------------|--------|--|
| <i>Counter</i> | 816 | DINT | Channel 0: Counter value / Frequency value |
| | 820 | DINT | Channel 1: Counter value / Frequency value |
| | 824 | DINT | Channel 2: Counter value / Frequency value |
| | 828 | DINT | Channel 3: Counter value / Frequency value |

| Sub module | Output address | Access | Description |
|----------------|----------------|--------|-------------|
| <i>Counter</i> | 816 | DWORD | reserved |
| | 820 | DWORD | reserved |
| | 824 | DWORD | reserved |
| | 828 | DWORD | reserved |

5.7.4.2 Interrupt selection

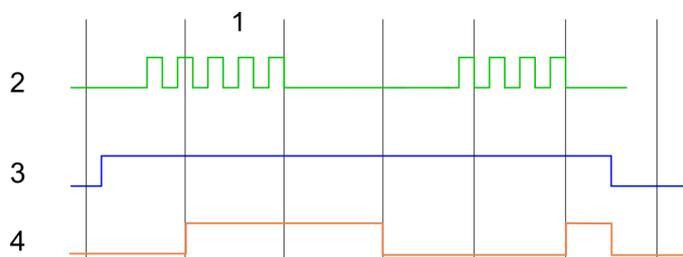
Via 'Basic parameters' you can reach 'Select interrupt'. Here you can define the interrupts the CPU will trigger. The following parameters are supported:

- None: The interrupt function is de-activated.
- Process: The following events of the frequency measurement can trigger a hardware interrupt (selectable via 'Frequency counting'):
 - End of measurement
- Diagnostics and process: A diagnostics interrupt is only triggered when a hardware interrupt was lost.

5.7.4.3 Frequency measurement

Parameter hardware configuration

Default values and structure of this dialog box depend on the selected 'Operating mode'. The following parameters are relevant for frequency measurement, which must be specified or determined:



- 1 Integration time
- 2 Counting pulse
- 3 SW gate
- 4 Evaluated frequency

Parameter overview

| Operating parameters | Description | Assignment |
|-------------------------|---|----------------------------------|
| Integration time | Specify the integration time Range of values: 10ms ... 10000ms in steps of 1ms | 100ms |
| Max. counting frequency | Specify the max. frequency for the corresponding input | 60kHz |
| | Frequency | shortest permissible count pulse |
| | 1kHz | 400µs |
| | 2kHz | 200µs |
| | 5kHz | 80µs |
| | 10kHz | 40µs |
| | 30kHz | 13µs |
| | 60kHz | 6.7µs |
| 100kHz | 4µs | |
| Hardware interrupt | Description | Assignment |
| End of measurement | Hardware interrupt at end of measurement | de-activated |

5.7.5 Status indication

| Digital input | LED  green | Description |
|---------------------|--|---|
| DI +0.0 ... DI +0.7 |  | Digital input I+0.0 ... 0.7 has "1" signal |
| |  | Digital input I+0.0 ... 0.7 has "0" signal |
| DI +1.0 ... DI +1.7 |  | Digital input I+1.0 ... 1.7 has "1" signal |
| |  | Digital input I+1.0 ... 1.7 has "0" signal |
| Digital output | LED  green | Description |
| DO +0.0 ... DO +0.7 |  | Digital output Q+0.0 ... 0.7 has "1" signal |
| |  | Digital output Q+0.0 ... 0.7 has "0" signal |
| DO +1.0 ... DO +1.3 |  | Digital output Q+1.0 ... 1.3 has "1" signal |
| |  | Digital output Q+1.0 ... 1.3 has "0" signal |
| Power supply | LED  green | Description |
| 1L+ |  | DC 24V electronic section supply OK |
| |  | DC 24V electronic section supply not available |
| 2L+ |  | DC 24V power section supply outputs OK |
| |  | DC 24V power section supply outputs not available |
| 3L+ |  | DC 24V power section supply SLIO bus OK |
| |  | DC 24V power section supply SLIO bus not available |
| 5L+ |  | DC 24V power section supply inputs OK |
| |  | DC 24V power section supply inputs not available |
| Error | LED  red | Description |
| 1F |  | Error, overload respectively short circuit on power supply sensor |
| |  | no error |
| 2F |  | Error at overload respectively short circuit at the outputs |
| |  | no error |

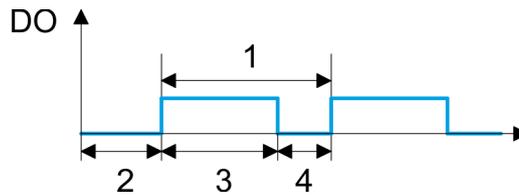
5.8 Pulse width modulation - PWM

5.8.1 Properties

- By presetting of time parameters, the CPU evaluates a pulse sequence with according pulse/pause ratio and outputs it via the according output channel.
- Channel 0 and 1 are supported
- Control by the user program via SFB 49



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



- 1 Period
- 2 On-delay
- 3 Pulse duration
- 4 Pulse pause

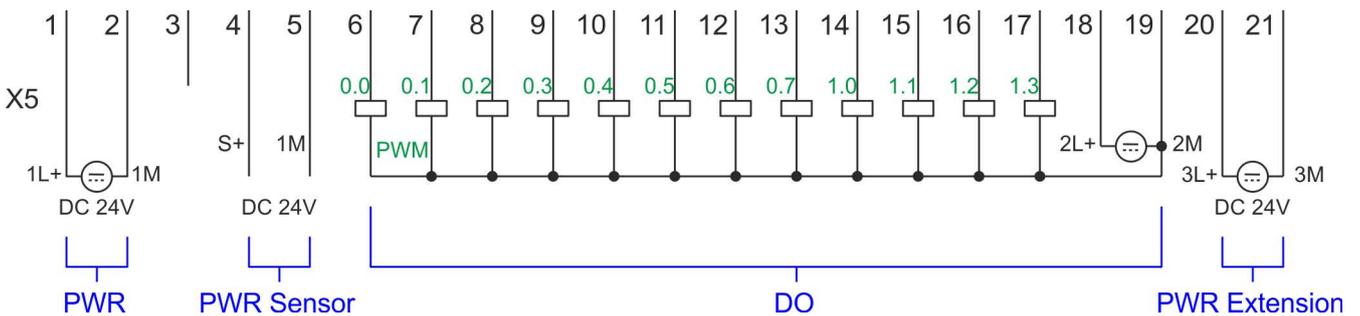


The counting function is disabled during the pulse width modulation on the same channel.

5.8.2 Wiring

5.8.2.1 Pulse width modulation outputs

X5: Connector



| X5 | Function | Type | Description |
|----|------------|------|--|
| 1 | Sys DC 24V | I | 1L+ DC 24V for electronic section supply |
| 2 | Sys 0V | I | 1M: GND for electronic section supply |
| 6 | DO 0.0 | O | PWM 0 |
| 7 | DO 0.1 | O | PWM 1 |
| 18 | DC 24V | I | 2L+ 24V DC power section supply for PWM |

| X5 | Function | Type | Description |
|----|----------|------|---|
| 19 | 0V | I | 2M: GND for PWM |
| 20 | DC 24V | I | 3L+: DC 24V SLIO bus power section supply |
| 21 | 0V | I | 3M: GND SLIO bus power section supply |

5.8.3 Proceeding

Hardware configuration

PWM and *pulse train* output use the same hardware configuration. Switching between these modes is done within the SFB 49. In the Siemens SIMATIC Manager the following steps should be executed:

1. ➤ Perform a hardware configuration for the CPU. ↪ *Chap. 4.5 'Hardware configuration - CPU' page 69*
2. ➤ Double-click the counter sub module of the CPU 314C-2 PN/DP.
⇒ The dialog '*Properties*' is opened.
3. ➤ As soon as you select the operating mode for the corresponding channel, a dialog box with default values for this counter mode is created and shown. For *PWM* respectively *pulse train* output select for the corresponding channel the operating mode '*Pulse width modulation - PWM*'.
4. ➤ Perform the required parameter settings.
5. ➤ Save your project with '*Station → Safe and compile*'.
6. ➤ Transfer your project to your CPU.

User program

- The SFB 49 should cyclically be called (e.g. OB 1) for controlling the pulse width modulation.
 - The SFB 49 is used for *PWM* and *pulse train* output.
 - The switching between the modes takes place by the presetting of the *pulse number* (JOB_ID = 08h/09h). As soon as you specify a *pulse number* > 0, you switch to the *pulse train* mode, otherwise *PWM* is active.
- The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.

5.8.4 Parametrization

5.8.4.1 Address assignment

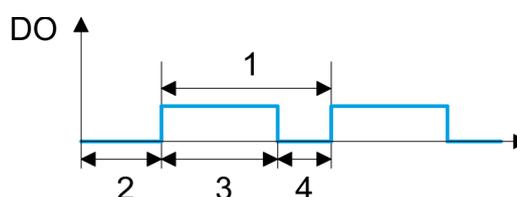
| Sub module | Input address | Access | Description |
|----------------|---------------|--------|-------------|
| <i>Counter</i> | 816 | DINT | reserved |
| | 820 | DINT | reserved |
| | 824 | DINT | reserved |
| | 828 | DINT | reserved |

| Sub module | Output address | Access | Description |
|------------|----------------|--------|-------------|
| Counter | 816 | DWORD | reserved |
| | 820 | DWORD | reserved |
| | 824 | DWORD | reserved |
| | 828 | DWORD | reserved |

5.8.4.2 Pulse width modulation

Parameter hardware configuration

Default values and structure of this dialog box depend on the selected 'Operating mode'. The following parameters are relevant for PWM, which must be specified or determined:



- 1 Period
- 2 On-delay
- 3 Pulse duration
- 4 Pulse pause

Parameter overview

| Operating parameters | Description | Assignment |
|----------------------|--|------------|
| Output format | <p>Here specify the range of values for the output. The CPU hereby determines the pulse duration:</p> <ul style="list-style-type: none"> ■ Per mil <ul style="list-style-type: none"> – Output value is within 0 ... 1000 – Pulse duration = (Output value / 1000) x Period ■ S7 Analog value: <ul style="list-style-type: none"> – Output value is Siemens S7 analog value 0 ... 27648 – Pulse duration = (Output value / 27648) x Period | Per mil |
| Time base | <p>Here you can set the time base, which will apply for resolution and range of values of the period duration, minimum pulse duration and on-delay.</p> <ul style="list-style-type: none"> ■ 1ms: Die Time base is 1ms ■ 0.1ms: Time base is 0.1ms | 0.1ms |
| On-delay | <p>Enter here a value for the time to expire from the start of the output sequence to the output of the pulse. The pulse sequence is output at the output channel, on expiration of the on-delay.</p> <p>Range of values: 0 ... 65535 from this there are the following effective values:</p> <ul style="list-style-type: none"> ■ Time base 1ms: 0 ... 65535ms ■ Time base 0.1ms: 0 ... 6553.5ms | 0 |

Pulse width modulation - PWM > Status indication

| Operating parameters | Description | Assignment |
|------------------------|--|------------|
| Period | <p>With the period you define the length of the output sequence, which consists of pulse duration and pulse pause.</p> <p>Range of values:</p> <p>Time base 1ms: 1 ... 87ms</p> <p>Time base 0.1ms: 0.4 ... 87.0ms</p> | 50* |
| Minimum pulse duration | <p>With the minimum pulse duration you can suppress short output pulses and short pulse pauses. All pulses or pauses, which are smaller than the minimum pulse duration, are suppressed. This allows you to filter very short pulses (spikes), which can not be recognized by the periphery.</p> <p>Range of values:</p> <p>Time base 1ms: 0 ... Period / 2 · 1ms</p> <p>Time base 0.1ms: 2 ... Period / 2 · 0.1ms</p> | 2 |

*) This value can vary depending on the configuration tool and can be out of range. Values, which are out of range are invalid and must be adjusted accordingly!

5.8.5 Status indication

| Digital output | LED | Description |
|----------------|---|----------------------|
| | <input checked="" type="checkbox"/> green | |
| DO +0.0 | <input checked="" type="checkbox"/> | PWM 0 has "1" signal |
| | <input type="checkbox"/> | PWM 0 has "0" signal |
| DO +0.1 | <input checked="" type="checkbox"/> | PWM 1 has "1" signal |
| | <input type="checkbox"/> | PWM 1 has "0" signal |

| Power supply | LED | Description |
|--------------|---|--|
| | <input checked="" type="checkbox"/> green | |
| 1L+ | <input checked="" type="checkbox"/> | DC 24V electronic section supply OK |
| | <input type="checkbox"/> | DC 24V electronic section supply not available |
| 2L+ | <input checked="" type="checkbox"/> | DC 24V power section supply outputs OK |
| | <input type="checkbox"/> | DC 24V power section supply outputs not available |
| 3L+ | <input checked="" type="checkbox"/> | DC 24V power section supply SLIO bus OK |
| | <input type="checkbox"/> | DC 24V power section supply SLIO bus not available |
| 5L+ | <input checked="" type="checkbox"/> | DC 24V power section supply inputs OK |
| | <input type="checkbox"/> | DC 24V power section supply inputs not available |

| Error | LED | Description |
|-------|--|---|
| | ■ red | |
| 1F | ■ | Error power supply sensor |
| | <input type="checkbox"/> | no error |
| 2F | ■ | Error at overload respectively short circuit at the outputs |
| | <input type="checkbox"/> | no error |

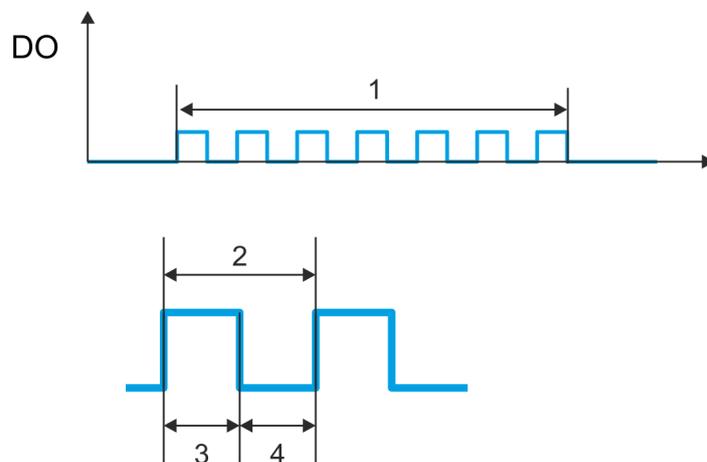
5.9 Pulse train

5.9.1 Properties

- By presetting of time parameters, the CPU evaluates a pulse sequence with according pulse/pause ratio and outputs it via the according output channel.
- The output is a pulse-direction command (P/D).
 - Output frequency pattern via pulse train channel
 - To output the direction, an additional output is to be used, which is to be controlled via your user program.
- Channel 0 and 1 are supported
- Control by the user program via SFB 49



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



- 1 Number of pulses
- 2 Period duration
- 3 Pulse duration
- 4 Pulse pause

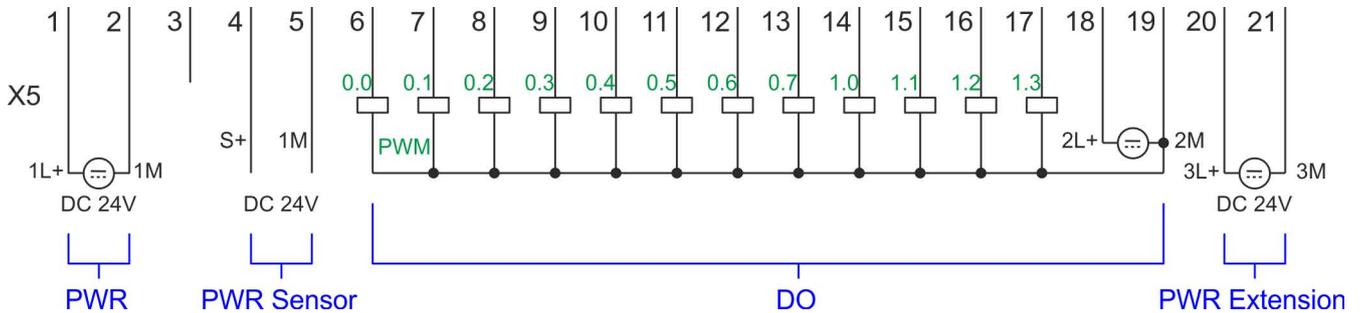


The counting function is disabled during the pulse train output on the same channel.

5.9.2 Wiring

5.9.2.1 Pulse train outputs

X5: Connector



| X5 | Function | Type | Description |
|----|------------|------|---|
| 1 | Sys DC 24V | I | 1L+ DC 24V for electronic section supply |
| 2 | Sys 0V | I | 1M: GND for electronic section supply |
| 6 | DO 0.0 | O | Pulse train 0 |
| 7 | DO 0.1 | O | Pulse train 1 |
| 18 | DC 24V | I | 2L+ 24V DC power section supply for pulse train |
| 19 | 0V | I | 2M: GND for pulse train |
| 20 | DC 24V | I | 3L+: DC 24V SLIO bus power section supply |
| 21 | 0V | I | 3M: GND SLIO bus power section supply |

5.9.3 Proceeding

Hardware configuration

PWM and *pulse train* output use the same hardware configuration. Switching between these modes is done within SFB 49. In the Siemens SIMATIC Manager the following steps should be executed:

1. ➤ Perform a hardware configuration for the CPU. ↪ *Chap. 4.5 'Hardware configuration - CPU' page 69*
2. ➤ Double-click the counter sub module of the CPU 314C-2 PN/DP.
⇒ The dialog '*Properties*' is opened.
3. ➤ As soon as you select the operating mode for the corresponding channel, a dialog box with default values for this counter mode is created and shown. For *PWM* respectively *pulse train* output select for the corresponding channel the operating mode '*Pulse width modulation - PWM*'.
4. ➤ Perform the required parameter settings.
5. ➤ Save your project with '*Station → Safe and compile*'.
6. ➤ Transfer your project to your CPU.

User program

- The SFB 49 should cyclically be called (e.g. OB 1) for controlling the pulse train output.
 - The SFB 49 is used for *PWM* and *pulse train* output.
 - The switching between the modes takes place by the presetting of the *pulse number* (JOB_ID = 08h/09h). As soon as you specify a *pulse number* > 0, you switch to the *pulse train* mode, otherwise *PWM* is active.
- The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.

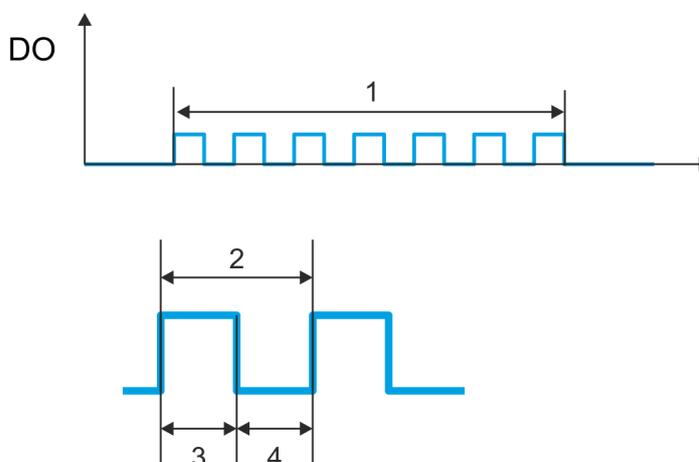
5.9.4 Parametrization**5.9.4.1 Address assignment**

| Sub module | Input address | Access | Description |
|------------|---------------|--------|-------------|
| Counter | 816 | DINT | reserved |
| | 820 | DINT | reserved |
| | 824 | DINT | reserved |
| | 828 | DINT | reserved |

| Sub module | Output address | Access | Description |
|------------|----------------|--------|-------------|
| Counter | 816 | DWORD | reserved |
| | 820 | DWORD | reserved |
| | 824 | DWORD | reserved |
| | 828 | DWORD | reserved |

5.9.4.2 Pulse train output**Parameter hardware configuration**

Default values and structure of this dialog box depend on the selected 'Operating mode'. For *pulse train* following parameters are relevant, to be specified or determined:



- 1 Number of pulses
- 2 Period duration
- 3 Pulse duration
- 4 Pulse pause

Parameter overview

| Operating parameters | Description | Assignment |
|------------------------|--|------------|
| Output format | <p>Here specify the range of values for the output. The CPU hereby determines the pulse duration:</p> <ul style="list-style-type: none"> ■ Per mil <ul style="list-style-type: none"> – Output value is within 0 ... 1000 – Pulse duration = (Output value / 1000) x period duration ■ S7 Analog value: <ul style="list-style-type: none"> – Output value is Siemens S7 analog value 0 ... 27648 – Pulse duration = (Output value / 27648) x period duration | Per mil |
| Time base | <p>Here you can set the time base, which will apply for resolution and range of values of the period duration, minimum pulse duration and on-delay.</p> <ul style="list-style-type: none"> ■ 1ms: The time base is 1ms ■ 0.1ms: Time base is 0.1ms | 0.1ms |
| On-delay | This parameter is ignored. | 0 |
| Period duration | <p>With <i>period duration</i> you define the length of the output sequence, which consists of pulse duration and pulse pause.</p> <p>Range of values:</p> <p>Time base 1ms: 1 ... 87ms</p> <p>Time base 0.1ms: 0.4 ... 87.0ms</p> | 50* |
| Minimum pulse duration | <p>With the <i>minimum pulse duration</i> you can suppress short output pulses and short pulse pauses. All pulses or pauses, which are smaller than the <i>minimum pulse duration</i>, are suppressed. This allows you to filter very short pulses (spikes), which can not be recognized by the periphery.</p> <p>Range of values:</p> <p>Time base 1ms: 0 ... Period duration / 2 · 1ms</p> <p>Time base 0.1ms: 2 ... Period duration / 2 · 0.1ms</p> | 2 |

*) This value can vary depending on the configuration tool and can be out of range. Values, which are out of range are invalid and must be adjusted accordingly!

5.9.5 Status indication

| Digital output | LED | Description |
|----------------|---|------------------------------|
| |  green | |
| DO +0.0 |  | Pulse train 0 has "1" signal |
| | <input type="checkbox"/> | Pulse train 0 has "0" signal |
| DO +0.1 |  | Pulse train 1 has "1" signal |
| | <input type="checkbox"/> | Pulse train 1 has "0" signal |

| Power supply | LED <input checked="" type="checkbox"/> green | Description |
|--------------|--|--|
| 1L+ | <input checked="" type="checkbox"/> | DC 24V electronic section supply OK |
| | <input type="checkbox"/> | DC 24V electronic section supply not available |
| 2L+ | <input checked="" type="checkbox"/> | DC 24V power section supply outputs OK |
| | <input type="checkbox"/> | DC 24V power section supply outputs not available |
| 3L+ | <input checked="" type="checkbox"/> | DC 24V power section supply SLIO bus OK |
| | <input type="checkbox"/> | DC 24V power section supply SLIO bus not available |
| 5L+ | <input checked="" type="checkbox"/> | DC 24V power section supply inputs OK |
| | <input type="checkbox"/> | DC 24V power section supply inputs not available |

5.10 Diagnostic and interrupt

5.10.1 Overview

Hardware interrupt

The parametrization allows you to define the following trigger for a hardware interrupt:

- Edge at a digital interrupt input
- Reaching the comparison value
- Overflow respectively at overrun upper counter limit
- Underflow respectively at underrun lower counter limit
- Opening the HW gate with open SW gate - except for counter 3
- Closing the HW gate with open SW gate - except for counter 3

Diagnostics interrupt

The VIPA specific parameters allow you to define the following trigger for a diagnostics interrupt ↗ *Chap. 4.9 'Setting VIPA specific CPU parameters' page 82:*

- Hardware interrupt lost
- Error: 2L+ DC 24V DO power section supply
- Error: 3L+: DC 24V SLIO bus power section supply
- Error: 5L+: DC 24V DI power section supply
- Short circuit overload: Sensor
- Short circuit overload: DO

5.10.2 Process interrupt



An interrupt for the corresponding channel operating mode can only be triggered if you have additionally parameterized 'Diagnostics+Process' at 'Select interrupt' of the 'Basic parameters'.

A process interrupt causes a call of the OB 40. Within the OB 40 you may find the logical basic address of the module that initialized the process interrupt by using the Local word 6. More detailed information about the initializing event is to find in the *local double word 8*. The assignment of *local double word 8* depends on the parameterized operating mode of each channel.

Local double word 8 of OB 40 at Alarm Inputs

| Local byte | Bit 7...0 |
|------------|--|
| 8 | <ul style="list-style-type: none"> ■ Bit 0: Edge at I+0.0 ■ Bit 1: Edge at I+0.1 ■ Bit 2: Edge at I+0.2 ■ Bit 3: Edge at I+0.3 ■ Bit 4: Edge at I+0.4 ■ Bit 5: Edge at I+0.5 ■ Bit 6: Edge at I+0.6 ■ Bit 7: Edge at I+0.7 |
| 9 | <ul style="list-style-type: none"> ■ Bit 0: Edge at I+1.0 ■ Bit 1: Edge at I+1.1 ■ Bit 2: Edge at I+1.2 ■ Bit 3: Edge at I+1.3 ■ Bit 4: Edge at I+1.4 ■ Bit 5: Edge at I+1.5 ■ Bit 6: Edge at I+1.6 ■ Bit 7: Edge at I+1.7 |
| 10...11 | <ul style="list-style-type: none"> ■ Bit 7 ... 0: 0 (fix) |

Local double word 8 of OB 40 at counter function

| Local byte | Bit 7...0 |
|------------|---|
| 8 | <ul style="list-style-type: none"> ■ Bit 1, 0: 0 (fix) ■ Bit 2: Over-/underflow value counter 0 ■ Bit 3: Counter 0 reached comparison value ■ Bit 7 ... 4: 0 (fix) |
| 9 | <ul style="list-style-type: none"> ■ Bit 1, 0: 0 (fix) ■ Bit 2: Over-/underflow value counter 1 ■ Bit 3: Counter 1 reached comparison value ■ Bit 7 ... 4: 0 (fix) |
| 10 | <ul style="list-style-type: none"> ■ Bit 1, 0: 0 (fix) ■ Bit 2: Over-/underflow value counter 2 ■ Bit 3: Counter 2 reached comparison value ■ Bit 7 ... 4: 0 (fix) |
| 11 | <ul style="list-style-type: none"> ■ Bit 0: Gate counter 3 open (activated) ■ Bit 1: Gate counter 3 closed ■ Bit 2: Over-/underflow value counter 3 ■ Bit 3: Counter 3 reached comparison value ■ Bit 4: Counter 3 new latch value ■ Bit 7 ... 5: 0 (fix) |

Local double word 8 of OB 40 at *frequency measurement*

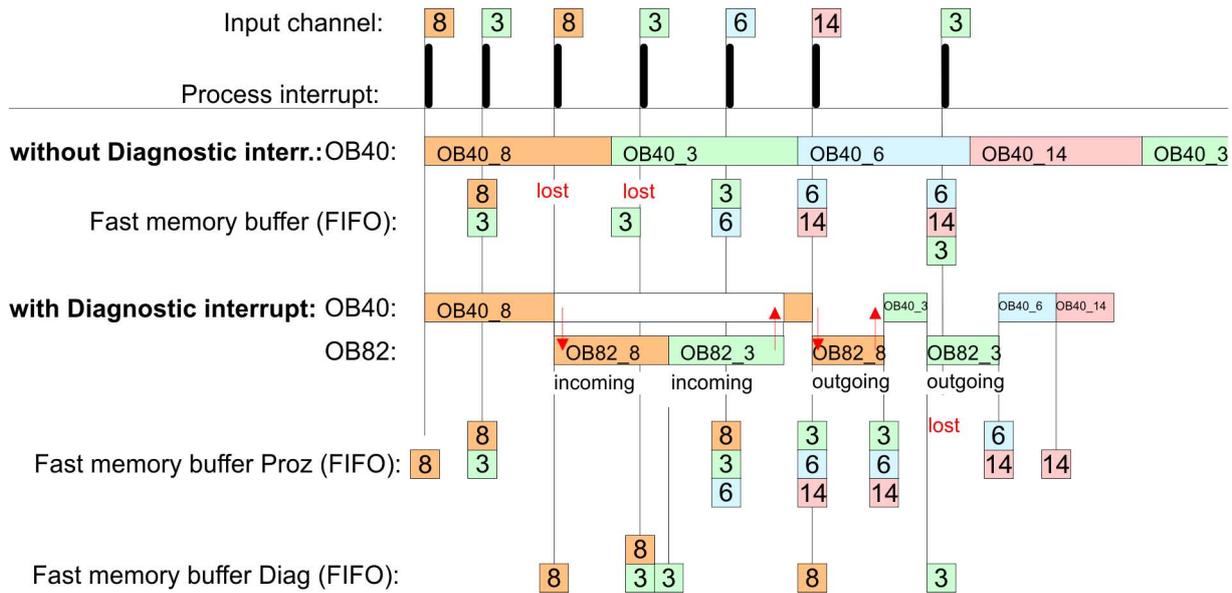
| Local byte | Bit 7...0 |
|------------|---|
| 8 | <ul style="list-style-type: none"> ■ Bit 0: End of measurement channel 0 (end of the integration time) ■ Bit 7 ... 1: 0 (fix) |
| 9 | <ul style="list-style-type: none"> ■ Bit 0: End of measurement channel 1 (end of the integration time) ■ Bit 7 ... 1: 0 (fix) |
| 10 | <ul style="list-style-type: none"> ■ Bit 0: End of measurement channel 2 (end of the integration time) ■ Bit 7 ... 1: 0 (fix) |
| 11 | <ul style="list-style-type: none"> ■ Bit 0: End of measurement channel 3 (end of the integration time) ■ Bit 7 ... 1: 0 (fix) |

5.10.3 Diagnostic interrupt**Function**

An interrupt for the corresponding channel operating mode can only be triggered if you have additionally parameterized 'Diagnostics+Process' at 'Select interrupt' of the 'Basic parameters'.

Via the parameterization (record set 7Fh) you may activate a global diagnostic interrupt for the module. A diagnostic interrupt occurs when during a process interrupt execution in OB 40 another process interrupt is thrown for the same event. The initialization of a diagnostic interrupt interrupts the recent process interrupt execution in OB 40 and branches in OB 82 to diagnostic interrupt processing_{incoming}. If during the diagnostic interrupt processing other events are occurring at other channels that may also cause a process res. diagnostic interrupt, these are interim stored. After the end of the diagnostic interrupt processing at first all interim stored diagnostic interrupts are processed in the sequence of their occurrence and then all process interrupts. If a channel where currently a diagnostic interrupt_{incoming} is processed res. interim stored initializes further process interrupts, these get lost. When a process interrupt for which a diagnostic interrupt_{incoming} has been released is ready, the diagnostic interrupt processing is called again as diagnostic interrupt_{outgoing}. All events of a channel between diagnostic interrupt_{incoming} and diagnostic interrupt_{outgoing} are not stored and get lost. Within this time window (1. diagnostic interrupt_{incoming} until last diagnostic interrupt_{outgoing}) the SF-LED of the CPU is on. Additionally for every diagnostic interrupt_{incoming/outgoing} an entry in the diagnostic buffer of the CPU occurs.

Example:



Diagnostic interrupt processing

Every OB 82 call causes an entry in the diagnostic buffer of the CPU containing error cause and module address. By using the SFC 59 you may read the diagnostic bytes. At de-activated diagnostic interrupt you have access to the last recent diagnostic event. If you've activated the diagnostic function in your hardware configuration, the contents of record set 0 are already in the local double word 8 when calling the OB 82. The SFC 59 allows you to also read the record set 1 that contains additional information. After leaving the OB 82 a clear assignment of the data to the last diagnostic interrupt is not longer possible. The record sets of the diagnostic range have the following structure:

Record set 0
Diagnostic_{incoming}

| Byte | Bit 7...0 |
|------|---|
| 0 | <ul style="list-style-type: none"> ■ Bit 0: set at module failure <ul style="list-style-type: none"> – Counter/Frequency measurement: Process interrupt lost – Digital input: Process interrupt lost – Missing power supply DI or DO – Digital output: short circuit/overload – Output Sensor: short circuit/overload – SLIO bus: missing supply fieldbus – Diagnostic interrupt from SLIO modules ■ Bit 1: set at internal error <ul style="list-style-type: none"> – Missing power supply DI or DO – Digital output: short circuit/overload – Output Sensor: short circuit/overload ■ Bit 2: set at external error <ul style="list-style-type: none"> – SLIO bus: missing supply fieldbus ■ Bit 3: set at channel error ■ Bit 4: set at missing external power supply <ul style="list-style-type: none"> – SLIO bus: missing supply fieldbus ■ Bit 7 ... 5: 0 (fix) |
| 1 | <ul style="list-style-type: none"> ■ Bit 3 ... 0: Module class <ul style="list-style-type: none"> – 1111b: Digital module or – 1000b: Function module ■ Bit 4: Channel information present <ul style="list-style-type: none"> – Counter/Frequency measurement: Process interrupt lost – Digital input: Process interrupt lost – Missing power supply DI or DO – Digital output: short circuit/overload – Output Sensor: short circuit/overload – SLIO bus: missing supply fieldbus – Diagnostic interrupt from SLIO modules ■ Bit 7 ... 5: 0 (fix) |
| 2 | <ul style="list-style-type: none"> ■ Bit 3 ... 0: 0 (fix) ■ Bit 4: set at missing internal power supply <ul style="list-style-type: none"> – Missing power supply DI or DO ■ Bit 7 ... 5: 0 (fix) |
| 3 | <ul style="list-style-type: none"> ■ Bit 5 ... 0: 0 (fix) ■ Bit 6: Process interrupt lost ■ Bit 7: 0 (fix) |

Record set 0
Diagnostic_{outgoing}

After the removing error a diagnostic message_{outgoing} takes place if the diagnostic interrupt release is still active.

| Byte | Bit 7...0 |
|------|---|
| 0 | <ul style="list-style-type: none"> ■ Bit 0: set at module failure <ul style="list-style-type: none"> – Counter/Frequency measurement: Process interrupt lost – Digital input: Process interrupt lost – Missing power supply DI or DO – Digital output: short circuit/overload – Output Sensor: short circuit/overload – SLIO bus: missing supply fieldbus – Diagnostic interrupt from SLIO modules ■ Bit 1: set at internal error <ul style="list-style-type: none"> – Missing power supply DI or DO – Digital output: short circuit/overload – Output Sensor: short circuit/overload ■ Bit 2: set at external error <ul style="list-style-type: none"> – SLIO bus: missing supply fieldbus ■ Bit 3: set at channel error ■ Bit 4: set at missing external power supply <ul style="list-style-type: none"> – SLIO bus: missing supply fieldbus ■ Bit 7 ... 5: 0 (fix) |
| 1 | <ul style="list-style-type: none"> ■ Bit 3 ... 0: Module class <ul style="list-style-type: none"> – 1111b: Digital module or – 1000b: Function module ■ Bit 4: Channel information present <ul style="list-style-type: none"> – Counter/Frequency measurement: Process interrupt lost – Digital input: Process interrupt lost – Missing power supply DI or DO – Digital output: short circuit/overload – Output Sensor: short circuit/overload – SLIO bus: missing supply fieldbus – Diagnostic interrupt from SLIO modules ■ Bit 7 ... 5: 0 (fix) |
| 2 | <ul style="list-style-type: none"> ■ Bit 3 ... 0: 0 (fix) ■ Bit 4: set at missing internal power supply <ul style="list-style-type: none"> – Missing power supply DI or DO ■ Bit 7 ... 5: 0 (fix) |
| 3 | <ul style="list-style-type: none"> ■ Bit 7 ... 0: 0 (fix) |



The record set 0 of the alarm interrupts, counter function, frequency measurement and pulse width modulation has the same structure. There are differences in the structure of record set 1.

Diagnostic record set 1 at Alarm Inputs

The record set 1 contains the 4byte of the record set 0 and additionally 12byte module specific diagnostic data. The diagnostic bytes have the following assignment:

| Byte | Bit 7...0 |
|---------|---|
| 0 ... 3 | Content record set 0 ↗ 'Record set 0 Diagnostic _{incoming} ' page 165 |
| 4 | <ul style="list-style-type: none"> ■ Bit 6 ... 0: Channel type (here 70h) <ul style="list-style-type: none"> – 70h: Digital input ■ Bit 7: More channel types present <ul style="list-style-type: none"> – 0: no – 1: yes |
| 5 | Number of diagnostic bits per channel (here 08h) |
| 6 | Number of channels of a module (here 08h) |
| 7 | <ul style="list-style-type: none"> ■ Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) ■ Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) ■ Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) ■ Bit 3: Error in channel group 2 (I+1.4 ... I+1.7) ■ Bit 7 ... 4: reserved |
| 8 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+0.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.3 ■ Bit 7: 0 (fix) |
| 9 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+0.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.7 ■ Bit 7: 0 (fix) |
| 10 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+1.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.3 ■ Bit 7: 0 (fix) |

| Byte | Bit 7...0 |
|----------|---|
| 11 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+1.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.7 ■ Bit 7: 0 (fix) |
| 12 ...15 | ■ Bit 7 ... 0: reserved |

Diagnostic record set 1 at counter function

The record set 1 contains the 4byte of the record set 0 and additionally 12byte module specific diagnostic data. The diagnostic bytes have the following assignment:

| Byte | Bit 7...0 |
|---------|---|
| 0 ... 3 | Content record set 0 ↪ 'Record set 0 Diagnostic _{incoming} ' page 165 |
| 4 | <ul style="list-style-type: none"> ■ Bit 6 ... 0: Channel type (here 76h) <ul style="list-style-type: none"> – 76h: Function module ■ Bit 7: More channel types present <ul style="list-style-type: none"> – 0: no – 1: yes |
| 5 | Number of diagnostic bits per channel (here 08h) |
| 6 | Number of channels of a module (here 04h) |
| 7 | <ul style="list-style-type: none"> ■ Bit 0: Error in channel group 0 (Counter 0) ■ Bit 1: Error in channel group 1 (Counter 1) ■ Bit 2: Error in channel group 2 (Counter 2) ■ Bit 3: Error in channel group 3 (Counter 3) ■ Bit 7 ... 4: reserved |
| 8 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 1, 0: reserved ■ Bit 2: Over-/underflow/end value counter 0 ■ Bit 3: Counter 0 reached comparison value ■ Bit 7 ... 4: 0 (fix) |
| 9 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 1, 0: reserved ■ Bit 2: Over-/underflow/end value counter 1 ■ Bit 3: Counter 1 reached comparison value ■ Bit 7 ... 4: 0 (fix) |
| 10 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 1, 0: reserved ■ Bit 2: Over-/underflow/end value counter 2 ■ Bit 3: Counter 2 reached comparison value ■ Bit 7 ... 4: 0 (fix) |

| Byte | Bit 7...0 |
|-----------|--|
| 11 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: Gate counter 3 open (activated) ■ Bit 1: Gate counter 3 closed ■ Bit 2: Over-/underflow/end value counter 3 ■ Bit 3: Counter 3 reached comparison value ■ Bit 4: Counter 3 new latch value ■ Bit 7 ... 5: 0 (fix) |
| 12 ... 15 | ■ Bit 7 ... 0: reserved |

Diagnostic Record set 1 at frequency measurement

The record set 1 contains the 4byte of the record set 0 and additionally 12byte module specific diagnostic data. The diagnostic bytes have the following assignment:

| Byte | Bit 7...0 |
|-----------|--|
| 0 ... 3 | Content record set 0 ↪ 'Record set 0 Diagnostic _{incoming} ' page 165 |
| 4 | <ul style="list-style-type: none"> ■ Bit 6 ... 0: Channel type (here 76h) <ul style="list-style-type: none"> – 76h: Function module ■ Bit 7: More channel types present <ul style="list-style-type: none"> – 0: no – 1: yes |
| 5 | Number of diagnostic bits per channel (here 08h) |
| 6 | Number of channels of a module (here 04h) |
| 7 | <ul style="list-style-type: none"> ■ Bit 0: Error in channel group 4 (Frequency meter 0) ■ Bit 1: Error in channel group 5 (Frequency meter 1) ■ Bit 2: Error in channel group 6 (Frequency meter 2) ■ Bit 3: Error in channel group 7 (Frequency meter 3) ■ Bit 7 ... 4: 0 (fix) |
| 8 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: End of measurement channel 0 (End of integration time) ■ Bit 7 ... 1: 0 (fix) |
| 9 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: End of measurement channel 1 (End of integration time) ■ Bit 7 ... 1: 0 (fix) |
| 10 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: End of measurement channel 2 (End of integration time) ■ Bit 7 ... 1: 0 (fix) |
| 11 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: End of measurement channel 3 (End of integration time) ■ Bit 7 ... 1: 0 (fix) |
| 12 ... 15 | 0 (fix) |

6 Deployment PtP communication

6.1 Fast introduction

General

The CPU has the interface X3 MPI(PtP) with a fix pinout. After an overall reset the interface has MPI functionality. By appropriate configuration the PtP function (**point to point**) can be enabled:

- 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.

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 instead SFCs

Please regard that the special VIPA SFCs are not shown in the SLIO CPU. Please use for programming tools e.g. Siemens SIMATIC Manager and TIA Portal the according FCs of the VIPA library.

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 |



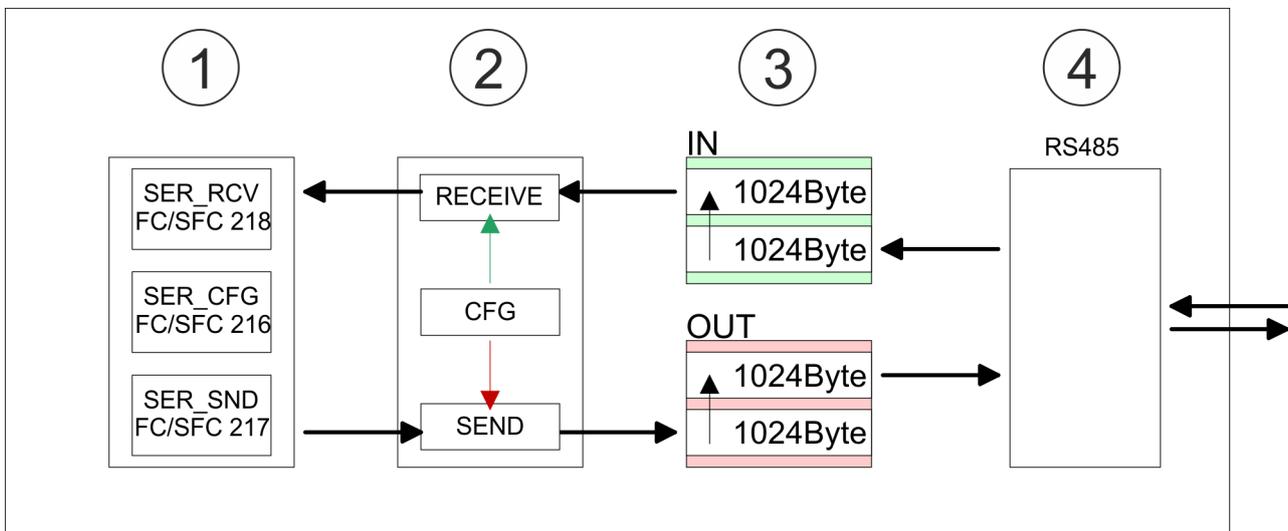
More information about the usage of these blocks may be found in the manual "SPEED7 Operation List" from VIPA.

6.2 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.



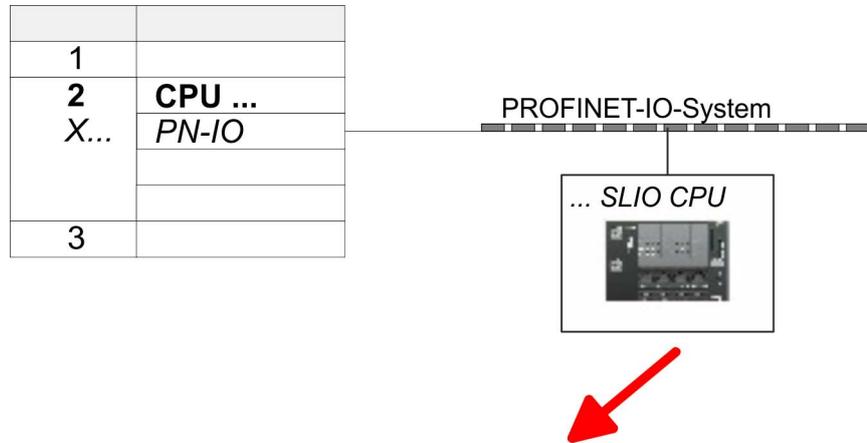
- 1 Program
- 2 Protocol
- 3 FIFO buffer
- 4 Interface

Enable PtP functionality

6.3 Enable PtP functionality

Proceeding

After the ↗ *Chap. 4.5 'Hardware configuration - CPU' page 69* of the CPU you can set the parameters of the CPU in the virtual IO device 'VIPA SLIO CPU'.



| | | | |
|-----|------------------|-----|--|
| 0 | ... SLIO CPU ... | ... | |
| X2 | ... | | |
| 1 | | | |
| 2 | | | |
| 3 | | | |
| ... | | | |

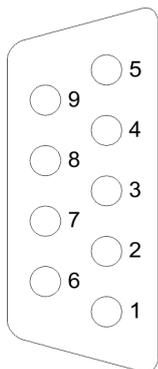
1. ➤ Open the properties dialog by a double-click at 'VIPA SLIO CPU'.
 - ⇒ The VIPA specific parameters may be accessed by means of the properties dialog.
2. ➤ Select at 'Function X3' the value 'PTP'.
3. ➤ Save and transfer your project to the CPU.
 - ⇒ After a short boot time the interface X3 MPI(PtP) is ready for PtP communication.

6.4 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

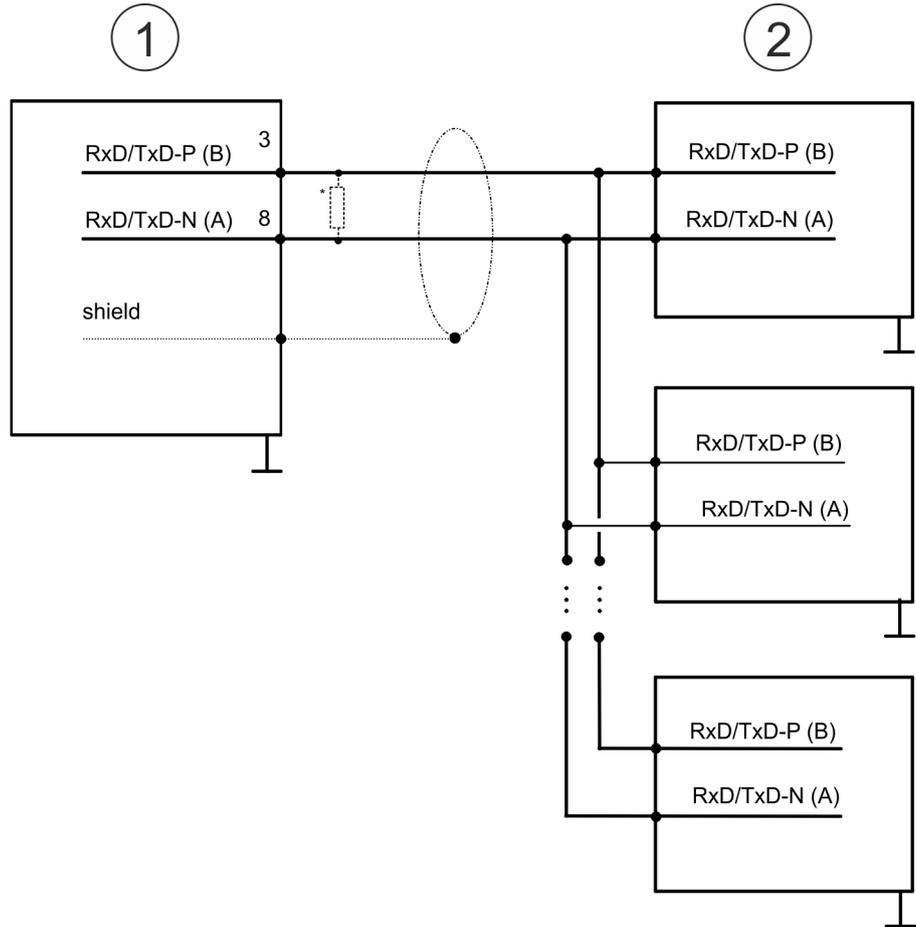
RS485



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



*) For traffic-free data transfer use a terminating resistor of approximately 120Ω .

6.5 Parametrization

6.5.1 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.

6.6 Communication

6.6.1 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.

6.6.2 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.

6.7 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.

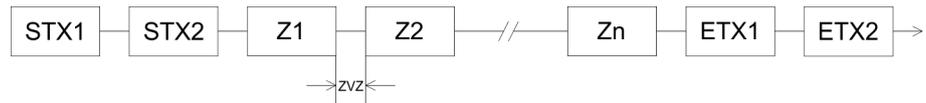
STX/ETX

STX/ETX is a simple protocol with start and end ID, where STX stands for **Start of Text** and ETX for **End of Text**.

- 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

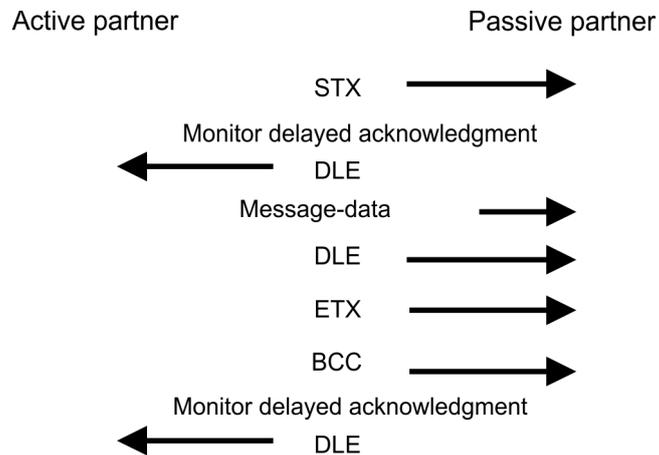
The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free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

The procedure employs the following control characters:

- STX: **S**tart of **T**ext
- DLE: **D**ata **L**ink **E**scape
- ETX: **E**nd of **T**ext
- BCC: **B**lock **C**heck **C**haracter
- NAK: **N**egative **A**cknowledge

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

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | 1 | | | | | |

|
Broadcast

A request can be directed to a certain slave or be sent 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

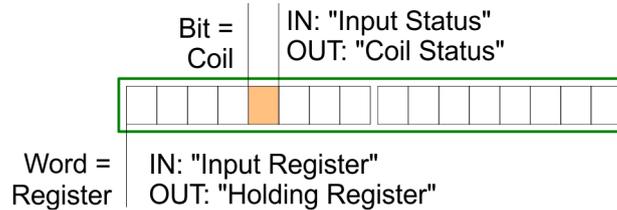
The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

6.8 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".

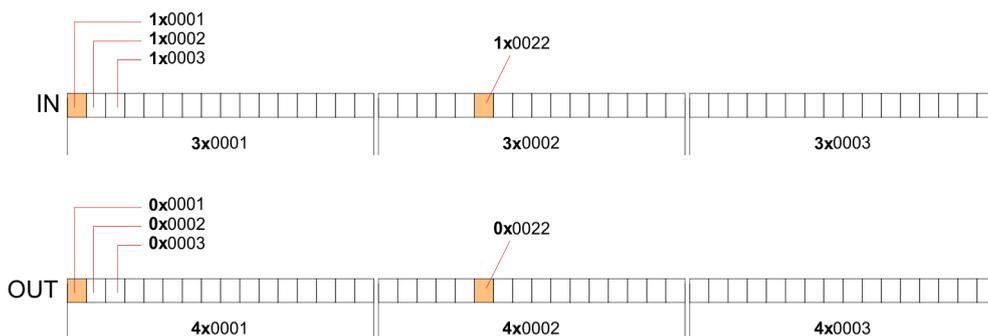
Range definitions

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. The description always takes place from the point of view of the master:

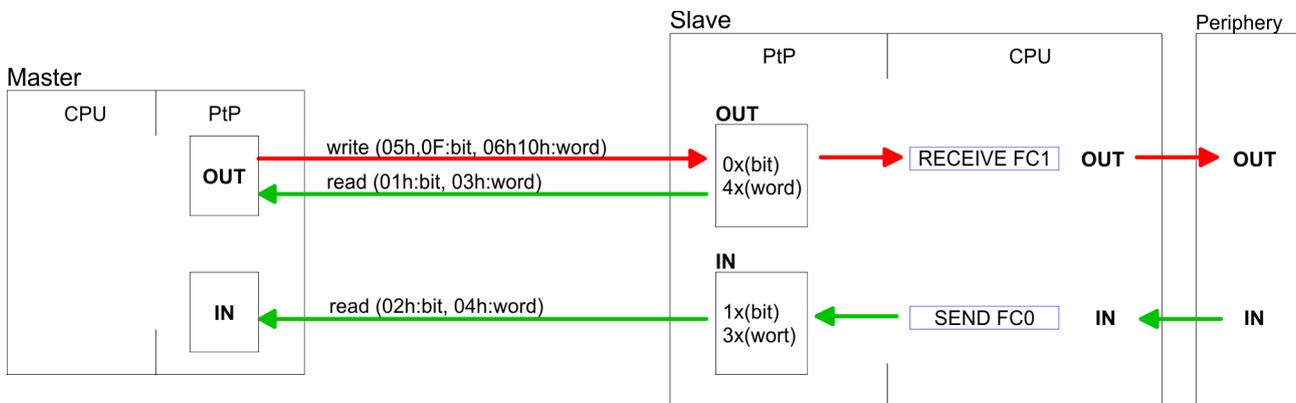
| Code | Command | Description |
|------|-------------|--------------------------------------|
| 01h | Read n bits | Read n bits of master output area 0x |
| 02h | Read n bits | Read n bits of master input area 1x |

Modbus - Function codes

| Code | Command | Description |
|------|---------------|--|
| 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).

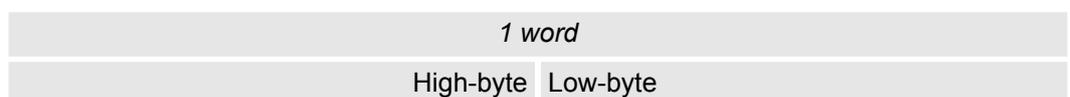


Respond of the slave

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



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 |

7 Deployment *OPC UA*



Please note that the simultaneous use of OPC UA and WebVisu on the same interface is not supported! When attempting to activate them, both servers are stopped and the diagnostic message 0xE989 or 0xE9AB is output.

- With an *OPC UA* project there is the possibility to configure an *OPC UA* server on your CPU.
- The configuration of an *OPC UA* project is only possible with the *SPEED7 Studio* V1.8.6 and up.
- Since a *OPC UA* project is only executable by memory card, a memory card of VIPA (VSD, VSC) must be plugged. Please note that you must always use a VSC card suitable for your CPU. ↪ *Chap. 4.16 'Deployment storage media - VSD, VSC' page 111*
- If the memory card is removed for a short time, the SF LED gets on. This indicates that a feature is missing and the *OPC UA* server will quit after 72 hours.
- When the project is transferred from the *SPEED7 Studio*, the *OPC UA* project is automatically transferred as TAR file to the inserted memory card.
- The *OPC UA* server must be activated in the CPU. ↪ *Chap. 8.2 'Activate WebVisu functionality' page 212*
- The access is made with an *OPC UA* client via the '*Endpoint URL*' of the corresponding interface. The '*Endpoint URL*' can be found on the *Device web page* of the CPU. ↪ *Chap. 4.11.1.1.1 'Tab: 'OPC UA'' page 97*

7.1 General

Term definitions

- **OPC - Open Platform Communications**
 - *OPC* is an interoperability standard for secure and reliable data exchange in industrial automation.
 - *OPC* is platform-independent and ensures a seamless flow of information between devices from different manufacturers.
- **UA - Unified Architecture**
 - *UA* specifies security features and data modeling based on a service-oriented architecture (SOA).

Precondition

- VIPA *SPEED7 Studio* from Version V1.8.6
 - The functionality for the *OPC UA* configuration is integrated in the *SPEED7 Studio*.
- Siemens SIMATIC Manager from version V5.5 and VIPA *SPEED7 Studio* from version V1.8.6
 - The *OPC UA* configuration is done with the *OPC UA Configurator*. This is part of the *SPEED7 Studio* from VIPA from version V1.8.6.
 - When calling the *OPC UA Configurator*, the *SPEED7 Studio* opens with functionality limited to *OPC UA* configuration.
 - The *OPC UA Configurator* is to be called from the Siemens SIMATIC Manager as external device tool.
 - To be able to call the *OPC UA Configurator* as an external device tool, you must first register it in the Siemens SIMATIC Manager. This is done with *SPEED7 Tools Integration*, which is automatically installed during the installation of the *SPEED7 Studio*.
 - The *OPC UA Configurator* is to be called from the Siemens SIMATIC Manager after project creation and online configuration.
 - The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens SIMATIC Manager.
 - The *OPC UA* configuration is transferred online from the *OPC UA Configurator*. The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens SIMATIC Manager.
- Siemens TIA Portal from version V15.0 and VIPA *SPEED7 Studio* from version V1.8.6
 - The *OPC UA* configuration is done with the *OPC UA Configurator*. This is part of the *SPEED7 Studio* from VIPA from version V1.8.6.
 - When calling the *OPC UA Configurator*, the *SPEED7 Studio* opens with functionality limited to *OPC UA* configuration.
 - The *OPC UA Configurator* is to be called from the Siemens TIA Portal as external device tool.
 - To be able to call the *OPC UA Configurator* as an external device tool, you must first register it in the Siemens TIA Portal. This is done with *SPEED7 Tools Integration*, which is automatically installed during the installation of the *SPEED7 Studio*.
 - The *OPC UA Configurator* is to be called from the Siemens TIA Portal after project creation and online configuration.
 - The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens TIA Portal.
 - The *OPC UA* configuration is transferred online from the *OPC UA Configurator*. The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens TIA Portal.

7.2 Basics OPC UA**7.2.1 OPC UA****Standard for data and information exchange**

OPC UA defines a common standard for data and information exchange in an 'Industry 4.0' environment. Due to the platform independence, the integrated security concept and the data type information supplied with the data, *OPC UA* provides the basis for machine-readable and cross-level communication.

OPC - Open Platform Communications

- Classic variant not scalable and exclusively for Microsoft Windows®
- For each type of data transfer, such as real-time data, history data, interrupts, events, etc., a separate solution with its own semantics is required, such as OPC DA, OPC HDA, OPC A&E, etc.
- Separate and complex effort required for security settings.

- *OPC* requires a complex *DCOM* configuration.
- *OPC* requires separate complex firewall settings.

OPC UA - Open Platform Communications Unified Architecture

- Scalable and platform-independent communication standard specified in IEC 62541.
- Standardization of classic *OPC* specifications with integrated security concept.
- The *OPC UA* security concept includes user and application authentication, message signing, and encryption of transmitted data.
- IP-based, optimized, binary protocol for Internet and firewall communication via one port (4840).
- With *OPC UA*, any type of information is available at any time and place for any authorized application and authorized person. For example, raw data and preprocessed information can be transported safely from the sensor and field level to the control system and into the production planning systems.
- SOA (**S**ervice **O**riented **A**rchitecture) replaces the Microsoft *DCOM* technology with open, platform-independent protocols with integrated security mechanisms.
 - Communication takes place via standardized services based on the *Information model* of *OPC UA*.
 - The services are divided into different task groups.
 - Based on a basic model, arbitrarily complex, object-oriented extensions of the services can be carried out, without affecting interoperability.

OPC UA server

- An *OPC UA server* provides information within a network that can be retrieved from an *OPC UA client*.
- The data exchange can take place via security certificates, which have to be stored accordingly in the server.
- The *OPC UA server* provides basic services such as data exchange or navigation through the address space.
- The *OPC UA* configuration is used to define the variables or contents that an *OPC UA server* should provide.
- The *OPC UA* configuration is done via an external tool such as for CPUs from VIPA the *OPC UA Configurator* from VIPA.

OPC UA client

OPC UA clients are programs with the following functionality:

- Read or write access to information of the *OPC UA server*.
- Access is controlled by access rights.
- Execute services on the *OPC UA server*.

Communication types

- Client/Server
 - An *OPC UA client* accesses information from the *OPC UA server* via services, provided by the *OPC UA server*. Here a fix defined connection is used.
 - Example: *OPC UA client* retrieves status of an input in the CPU.
- Publisher/Subscriber
 - A *Publisher* sends to unknown *Subscriber* (clients) without a fixed connection.
 - Example: Sensors send data to the cloud.

7.2.2 Information modeling

Information model

- *Information models* are used to describe devices and their data.
- The basis is the *Core specification*. The *Core specification* describes the structure of the address range and of the services, such as the entry points for the clients in the address space of an *OPC UA server*.

- In an *information model*, the content of the address space of the *OPC UA* server is described.
- The *Information models* are structured in layers. Each higher-order type is based on certain basic rules. Thus, clients who only know the basic rules can still edit complex information models, e.g. navigate through the *address space* and read or write data variables.
- In the *address space*, all information is represented by *Nodes*, which are interconnected via *references*.
- A node is always an instance of a *NodeClass*.
- *OPC UA* offers basic services such as data exchange or navigation through the address space. The services are grouped in *Service Sets*.

Node classes

The following *NodeClasses* are defined in the *OPC UA* specification:

- Variable - class of variables
- Method - class of functions
- Object - class of objects
- View - Class of view of a subset of nodes
- DataType - Class of the data types of the value of a variable
- VariableType - Class of the data types of a variable
- ObjectType - class of object types
- ReferenceType - class of reference types

Node attributes

Each *node* consists of attributes and references. Some attributes may also be optional. The following attributes of each *NodeClass* must be published:

- NodeID - Unique identifier of a *nodes* in the *address space*
- NodeClass - class of *node* instance
- BrowseName - name of the *node* in plain text
- DisplayName - display name of the *Node* for the user
- Description - Description of the *node* (optional)

OPC UA services

- *OPC UA* services are abstract descriptions defined by request and response messages.
- The available services of an *OPC UA* server are defined in the server profile and grouped together in service sets.

Basic service sets

- Discovery Service Set
 - Services for discovering existing servers and endpoints.
- SecureChannel Service Set
 - Services for opening and closing secure communication channels.
- Session Service Set
 - Services for the client to create and manage a session.
- NodeManagement Service Set
 - Services for creating and deleting nodes and references.
- View Service Set
 - Services for the client to navigate in the address space or in the view.
- Query Service Set
 - Services for search queries in the address space.
- Attribute Service Set
 - Services for accessing attributes of nodes.
- Method Service Set
 - Service for calling a method of an object.

- MonitoredItem Service Set
 - Services for the client to create and manage monitored items.
 - Monitored items are used to log in for data and event notifications.
- Subscription Service Set
 - Services for the client to create and manage subscriptions.
 - Subscriptions control the way of the data and event notification.

Access

- To access an *OPC UA* server, the *endpoint* must be known.
- You can navigate via the *endpoint* using the navigation function through the address space of the *OPC UA* server. Here you receive information about the *OPC UA* server and the CPU and have access to the objects created in the *OPC UA* configuration, such as tags, data blocks, etc.
- Lower network load through '*subscriptions*'
 - If variables are to be transmitted only if their value has changed, you have to use *subscriptions*.
 - To activate a *subscription*, enter the transmission interval "Publishing Interval" in the *OPC UA* client.
 - When the *subscription* is created, tell the server which variables to monitor. Among other things, you can specify the amount by which a value must change in order a transfer takes place.
 - Since only a change in value of a transmission takes place, the use of *subscriptions* leads to a reduced network load.
- Fast access through '*registration*'
 - Normally the addressing takes place by means of identifier strings. By using a numeric identifier access can be accelerated. For this reason, you should use the *registration* for regular access to certain variables.
 - During *registration*, the *OPC UA* client assigns the variable to the *OPC UA* server. The *OPC UA* server then generates a numeric identifier and sends it back to the *OPC UA* client.
 - The numerical identifier is valid for the duration of the session.
 - In the properties of the CPU, you can set the maximum number of registered nodes. This must be taken into account by the *OPC UA* clients.
 - Since the registration takes time, you should put them in the start-up phase of the *OPC UA* server.



Due to the system, access to data in complex structures is not consistent.



Setting the sampling intervals (sampling interval, publishing interval) too short may cause too much network load. Always choose intervals that are still sufficient for your application.

7.2.3 OPC UA data types and their conversion

Siemens S7 data types are mapped in the namespace via SPEED7 PLC OPC UA data types. Siemens S7 data types do not always match the *OPC UA* built-in data types. The CPU provides variables to the *OPC UA* server as an *OPC UA* built-in data type so that *OPC UA* clients can access these variables with *OPC UA* built-in data types via the server interface. A client can read the "DataType" attribute from such a variable and reconstruct the original data type.

Data type mapping

| Siemens S7 data type | | SPEED7 PLC OPC UA data type | | OPC UA built-in data type |
|----------------------|---|-----------------------------|---|----------------------------|
| BOOL | | BOOL | | Boolean |
| BYTE | | BYTE | | Byte |
| WORD | | WORD | | UInt16 |
| DWORD | | DWORD | | UInt32 |
| INT | | INT | | Int16 |
| DINT | | DINT | | Int32 |
| REAL | | REAL | | Float |
| S5TIME | | S5TIME | | UInt16 |
| TIME | → | TIME | → | Int32 |
| DATE | | DATE | | UInt16 |
| TIME_OF_DAY (TOD) | | TIME_OF_DAY | | UInt32 |
| CHAR | | CHAR | | Byte |
| COUNTER | | COUNTER | | UInt16 (Only valid values) |
| TIMER | | TIMER | | UInt16 (Only valid values) |
| STRING | | STRING | | String |
| DT | | DT | | Byte[8] |

Particularities

- String
 - The data type *STRING* in Siemens S7 is a byte array in which the maximum length and the current length are stored in the first 2 bytes. The other bytes store the string.
 - The *OPC UA* data type *String* should be defined in the same way.
- Array
 - A read or write job in *OPC UA* is always an *Array* access, i.e. always provided with index and length.
 - A single variable is a special case of an *Arrays* (index 0 and length 1). On the line, the data type is simply sent several times in succession. For the variable, the *DataType* attribute points to the base data type. The *ValueRank* and *ArrayDimensions* attributes determine if it is an array and how large the array is.
- Structure
 - A structure describes a complex data type.
 - You can describe your own structures as a subtype of the abstract data type *Structures*, which inherits from the data type *BaseDataType*.
 - Since a client may not know user-specific structures, the variables of the data type of this structure are uniformly published in an *ExtensionObject*. The structure *ExtensionObject* can be read by any client and also publishes the *DataTypeId* of the user-specific structure.
 - All structures that are not described by structures of the basic data types are published on the server in a *TypeDictionary*.
 - With the description of the structure by the *TypeDictionary* and the *DataTypeId*, which is published by the *ExtensionObject*, the structure of the *ExtensionObject* can be decoded by a client.
 - If a client knows in advance the description of a user-specific structure, it can be decoded without reading the *TypeDictionary*. In this approach, a client needs to read and decode the entire tree to access individual elements.

7.2.4 Integrated security concept

Generals to data security

The topic of data security and access protection have become increasingly important in the industrial environment. The increased networking of entire industrial systems to the network levels within the company together with the functions of remote maintenance have all served to increase vulnerability. Threats can arise from internal manipulation like technical errors, operator and program errors respectively from external manipulation like software viruses and worms, trojans and password phishing.

The most important precautions to prevent manipulation and loss of data security in the industrial environment are:

- Encrypting the data traffic by means of certificates.
- Filtering and inspection of the traffic by means of VPN - "Virtual Private Networks".
- Identification of the nodes by "Authentication" via save channels.
- Segmenting in protected automation cells, so that only devices in the same group can exchange data.

Guidelines for information security

With the "VDI/VDE 2182 sheet 1", Information Security in the Industrial Automation - General procedural model, VDI guidelines, the VDI/VDE society for measuring and automation engineering has published a guide for implementing a security architecture in the industrial environment. The guideline can be found at www.vdi.de PROFIBUS & PROFINET International (PI) can support you in setting up security standards by means of the "PROFINET Security Guideline". More concerning this can be found at the corresponding web site such as www.profibus.com

Security mechanisms in OPC UA

- Verifying the identity of *OPC UA* servers and clients.
- Checking the identity of the users.
- Signed and encrypted data exchange between *OPC UA* server and clients.
- In the connection settings in the *OPC UA Configurator*, you can specify how a user of an *OPC UA* client must legitimize access to the *OPC UA* server.

Safety rules:

- Only activate 'Anonymous-Login' or 'Unsecured data traffic' in exceptional cases.
- Only allow access to variables and data blocks via *OPC UA* if it is actually required.



Activate only security guidelines that are compatible with the protection concept for your machine or Application. Deactivate all other security guidelines.

X.509 certificates

OPC UA has integrated security mechanisms in multiple layers. An important component here are X.509 certificates, which are also used in the PC world. When using certificates, the *OPC UA* server delivers data to the client only if the security certificate has been accepted as valid on both sides. An X.509 certificate includes the following information:

- Version and serial number of the certificate.
- Name of the certification authority.
- Information about the algorithm used by the certification authority to sign the certificate.
- Start and end of the validity of the certificate.
- Name of the program, person, or organization for which the certificate was signed by the certification authority.
- The public key of the program, person or organization.

OPC UA uses three types of X.509 certificates when establishing a client-to-server connection:

- *OPC UA* application certificates
- *OPC UA* software certificates
- *OPC UA* user certificates

- Check when establishing a connection
 - When establishing a connection between client and server, the participants check all information from the certificate that is required to establish integrity.
 - Among other things, the period of validity which is stored in the certificate is checked. Please ensure that the date and time are set correctly for the participants, otherwise no communication can take place.
- Sign and encrypt
 - To avoid tampering, certificates are signed.
 - Within the *OPC UA Configurator*, you can use the 'Server settings' to import certificates or create and sign them yourself.
- Self-signed certificate
 - Each participant generates his own certificate and signs it.
 - Self-signed certificates are to be transferred to the CPU.
 - From a self-signed certificate no new certificates can be derived.
 - Sample applications: Static configuration with limited number of communication participants.
- *CA certificate*:
 - All certificates are created and signed by a certification authority.
 - Only the derived and signed certificate of the certification authority is to be transferred to the CPU.
 - The certification authority can generate new certificates. Adding partner devices is possible at any time.
 - Sample applications: Dynamically growing plants.

Digital signature

The signature can be used to prove the integrity and origin of a message.

1. ➤ The sender forms a hash value as a check value from the clear message.
2. ➤ The hash value and a private key result in the digital signature.
3. ➤ The clear message is sent to the recipient together with the digital signature.
4. ➤ The recipient decrypts the received signature with the public key and thus gets back the original hash value.
5. ➤ The receiver also forms a hash value from the clear message and checks it with the original hash value. The public key and hash method are included in the X.509 certificate.
 - ⇒ ■ If both hash values are identical, sender and clear message were not manipulated.
 - If both hash values are not identical, the clear message was manipulated or falsified during transmission.

Encrypting

- X.509 certificates are not encrypted; they are public and anyone can see them.
- Encrypting data prevents unauthorized users from knowing the content.
- When encrypting, the sender encrypts the clear message with the recipient's public key from the X.509 certificate.
- The recipient decrypts the message with his private key. Each owner of the private key can decrypt a received message.

Secure Channel

- *OPC UA* uses private and public keys to establish secure channels between client and server
- Once a secure connection is established, the client and server generate a shared private key for signing and encrypting messages.

Security policies

OPC UA uses the following security policies to protect messages:

- *No security*
All messages are unsecured. To use these security policies, connect to a "None" endpoint of a server.
- *Sign*
All messages are signed. This allows the integrity of the received messages to be checked. Manipulations are detected. To use these security policies, connect to a "Sign" endpoint of a server.
- *Sign & encrypt*
All messages are signed and encrypted. This allows the integrity of the received messages to be checked. Manipulations are detected. Due to the encryption, no attacker can read the content of the message. To use these security policies, connect to a "Sign & Encrypt" endpoint of a server.
The security guidelines are additionally named according to the algorithms used.
Example: "Basic256Sha256 - Sign & Encrypt" means: Secure Endpoint, supports a set of algorithms for 256-bit hashing and 256-bit encryption.



Please note that the encryption of the communication can affect CPU performance and therefore the response time of the entire system!

7.3 Activate OPC UA functionality

Proceeding

For your CPU can process a *OPC UA* project, you have to activate the *OPC UA* functionality.

1. ➤ Insert a memory card from VIPA (VSD, VSC) into your CPU. Please note that you must always use a VSC card suitable for your CPU. ↪ *Chap. 4.16 'Deployment storage media - VSD, VSC' page 111*
2. ➤ Turn on the CPU, to activate the *OPC UA* functionality, you have to establish an *Overall reset*.
 - ⇒ As long as the memory card is inserted, the *OPC UA* functionality remains activated even after a power cycle. When the project is transferred from the *OPC UA Configurator*, the *OPC UA* project is automatically transferred to the inserted memory card.



Please regard that the memory card must remain plugged when you've executed activated the OPC UA functionality. Otherwise the SF LED is on and the OPC UA functionality is deactivated after 72 hours. As long as an activated memory card is not inserted, the SF LED is on and the "TrialTime" timer counts from 72 hours down to 0. After that the OPC UA functionality is de-activated. By inserting the memory card, the LED goes off and the CPU runs again without restrictions.



Please note that the use of a OPC UA project, depending on the scope of the OPC UA project and the PLC project, can influence the performance and thus the response time of your application.

7.4 Usage in VIPA SPEED7 Studio

Precondition

- VIPA *SPEED7 Studio* from Version V1.8.6
 - The functionality for the *OPC UA* configuration is integrated in the *SPEED7 Studio*. For more information, see the according online help.

7.5 Usage in Siemens SIMATIC Manager

Precondition

Siemens SIMATIC Manager from V5.5 and VIPA *SPEED7 Studio* from V1.8.6

- The *OPC UA* configuration happens by the external *OPC UA Configurator* from VIPA.
- The *OPC UA Configurator* is the *SPEED7 Studio* reduced to *OPC UA* functionality.
- The *OPC UA Configurator* can be registered in the Siemens SIMATIC Manager by means of the *SPEED7 Tools Integration*.
- The *OPC UA Configurator* is to be called from the Siemens SIMATIC Manager after project creation and online configuration.
- The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens SIMATIC Manager.
- The *OPC UA* configuration is transferred online from the *OPC UA Configurator*. The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens SIMATIC Manager.



Please note that only the objects of the LD, FBD and IL languages can be transferred to the OPC UA Configurator.

7.5.1 Installation OPC UA Configurator

Proceeding

Installation and activation of SPEED7 Studio

The *OPC UA Configurator* is part of the *SPEED7 Studio* with *OPC UA* functionality. With the *SPEED7 Tools Integration*, which is also installed when installing the *SPEED7 Studio* the *OPC UA Configurator* is to be registered in the Siemens SIMATIC Manager as external tool.

1. The latest version of the *SPEED7 Studio* can be found in the download area of www.vipa.com. Double-click on the installation program and follow the instructions on the monitor.



The use of the *SPEED7 Studio* requires that you agree with the license agreement. During installation, you must confirm this.

Further components are required in order to operate *SPEED7 Studio*. If the following programs are not already present on your PC, they are automatically installed:

- Microsoft .NET Framework 4.52
- Microsoft SQL Server® 2014 SP1
- WinPcap

2. You can use a 30-day demo version or activate a license.

In order to use *SPEED7 Studio* without restrictions, you require a licence, which you can obtain from your local VIPA customer service organisation.

If the PC, on which you would like to use the *SPEED7 Studio*, is connected to the Internet, you can activate the licence online. If no license is activated, the dialog box for activating the license opens with each new start of *SPEED7 Studio*.

Click on 'Yes'.

⇒ The '*Product activation*' dialog window will open.

3. Enter the serial number that you received with your order of *SPEED7 Studio* in the '*Licence key*' input field.

4. Enter your name in the '*Your name*' input field.

5. If you enter your e-mail address in the '*E-mail address*' input field, you receive an e-mail confirmation regarding the product activation.

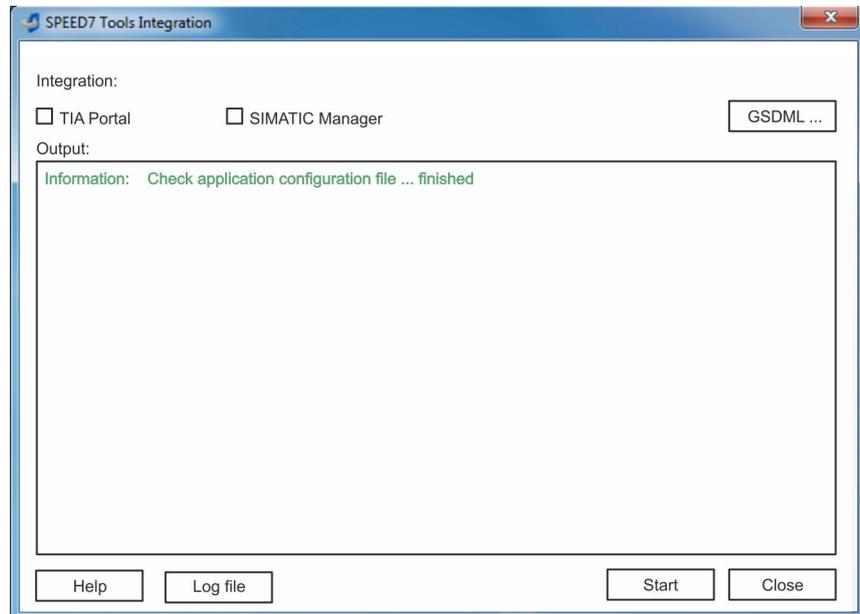
6. Click at '*Activate*'.

⇒ The licence is activated and the *SPEED7 Studio* is started.

Registration of SPEED7 Studio in the Siemens SIMATIC Manager as OPC UA Configurator.

SPEED7 Tools Integration is automatically listed in the Windows Start menu during the installation of the *SPEED7 Studio*.

1. ➤ To start the *SPEED7 Tools Integration*, click in the Windows Start menu on 'VIPA GmbH ➔ *SPEED7 Tools Integration*'.
- ⇒ For *SPEED7 Tools Integration* can start, you have to acknowledge the security prompt to change the data on your computer with 'Yes'. Afterwards *SPEED7 Tools Integration* will be started.



2. ➤ Click at 'GSDML ...'.
3. ➤ Navigate to your GSDML file of your VIPA-CPU, which you also use for your configuration in the Siemens '*SIMATIC Manager*'. Select these and click at '*Confirm*'. You can also select and use several GSDML files.
 - ⇒ The identified GSDML files are listed and the selection for the configuration tools is enabled.
4. ➤ Select the Siemens '*SIMATIC Manager*', in which the *SPEED7 Studio* is to be registered as *OPC UA Configurator*.
5. ➤ Click on '*Start*'.
 - ⇒ ■ *SPEED7 Studio* is registered in the Windows registry as *OPC UA Configurator*.
 - In the Siemens SIMATIC Manager the *OPC UA Configurator* is registered as externally callable program.
 - All changes are recorded in a log file, which you can output via '*Log file*'.
6. ➤ '*Close*' closes *SPEED7 Tools Integration*.
 - ⇒ With the next start of the Siemens hardware configurator, the *SPEED7 Studio* can be called as *OPC UA Configurator* with to *OPC UA* configuration limited functions. More information about the usage can be found in the in the online help of the *OPC UA Configurator*.

7.5.2 Steps of the OPC UA configuration

Steps of configuration

When using the Siemens SIMATIC Manager, the *OPC UA* configuration happens by the following steps:

1.  Create your project in the Siemens SIMATIC Manager with the corresponding hardware configuration.  *Chap. 4.5 'Hardware configuration - CPU' page 69*
2.  Configure the corresponding Ethernet connection for PG/OP communication and establish an online connection.  *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 72*
3.  Save translate and transfer your project.  *Chap. 4.10 'Project transfer' page 90*
4.  Call the external *OPC UA Configurator* from the Siemens SIMATIC Manager. For this click in the hardware configurator on the CPU from VIPA and select '*Start Device Tool → VIPA Framework → OPC UA Configurator*'.
5.  Confirm to start an external program with [YES].



NOTICE!

Data exchange between platforms of different vendors

If you allow access, you permit the exchange of data between *OPC UA Configurator* and your project data of the Siemens SIMATIC Manager.

- Ensure that the necessary security guidelines are complied with.

⇒ The *OPC UA Configurator* is started. For the *OPC UA* configuration, the data is taken from the Siemens SIMATIC Manager project and listed in the table for the *OPC UA* configuration.



Please note that only the objects of the LD, FBD and IL languages can be transferred to the *OPC UA Configurator*.

6.  Configure the *OPC UA* server and the data for the *OPC UA* communication.
7.  In the *OPC UA Configurator* switch to the online dialog and transfer the *OPC UA* configuration. For communication the IP address data are taken from the Siemens SIMATIC Manager project.
 - ⇒ The *OPC UA* configuration is now complete. For check you will find information about your *OPC UA* configuration on the device web page at '*OPC UA*'.
 *Chap. 4.11.1.1.1 'Tab: 'OPC UA'' page 97*

7.6 Usage in Siemens TIA Portal

Precondition

Siemens TIA Portal from version V15.0 and VIPA *SPEED7 Studio* from V1.8.6

- The *OPC UA* configuration happens by the external *OPC UA Configurator* from VIPA.
- The *OPC UA Configurator* is the *SPEED7 Studio* reduced to *OPC UA* functionality.
- The *OPC UA Configurator* can be registered in the Siemens TIA Portal by means of the *SPEED7 Tools Integration*.
- The *OPC UA Configurator* is to be called from the Siemens TIA Portal after project creation and online configuration.
- The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens TIA Portal.
- The *OPC UA* configuration is transferred online from the *OPC UA Configurator*. For the communication the *OPC UA Configurator* automatically uses the IP address data of the Siemens TIA Portal project.



Please note that only the objects of the LD, FBD and IL languages can be transferred to the OPC UA Configurator.

7.6.1 Installation OPC UA Configurator

Proceeding

Installation and activation of SPEED7 Studio

The *OPC UA Configurator* is part of the *SPEED7 Studio* with *OPC UA* functionality. With the *SPEED7 Tools Integration*, which is also installed when installing the *SPEED7 Studio* the *OPC UA Configurator* is to be registered in the Siemens TIA Portal as external tool.

1. The latest version of the *SPEED7 Studio* can be found in the download area of www.vipa.com. Double-click on the installation program and follow the instructions on the monitor.



The use of the *SPEED7 Studio* requires that you agree with the license agreement. During installation, you must confirm this.

Further components are required in order to operate *SPEED7 Studio*. If the following programs are not already present on your PC, they are automatically installed:

- Microsoft .NET Framework 4.52
- Microsoft SQL Server® 2014 SP1
- WinPcap

2. You can use a 30-day demo version or activate a license.

In order to use *SPEED7 Studio* without restrictions, you require a licence, which you can obtain from your local VIPA customer service organisation.

If the PC, on which you would like to use the *SPEED7 Studio*, is connected to the Internet, you can activate the licence online. If no licence is activated, the dialog box for activating the licence opens with each new start of *SPEED7 Studio*.

Click on 'Yes'.

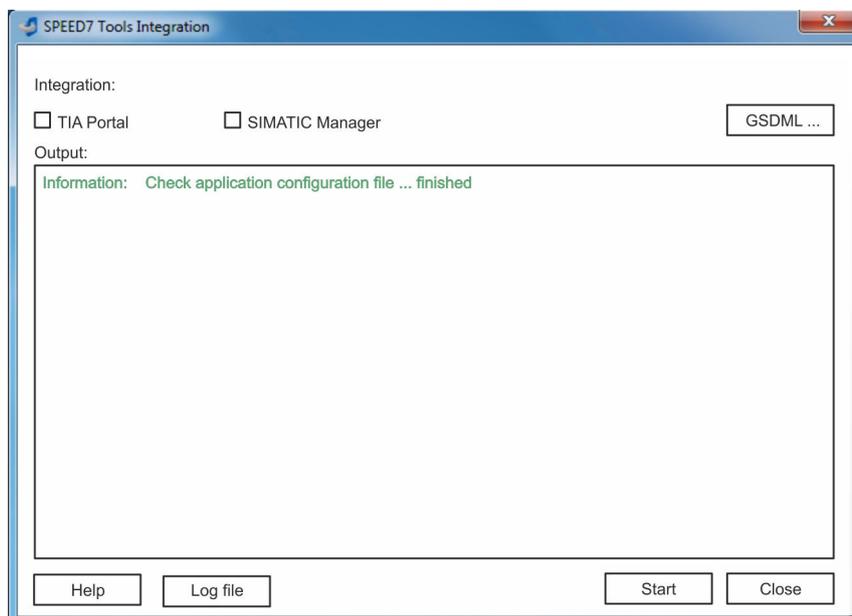
⇒ The 'Product activation' dialog window will open.

3. Enter the serial number that you received with your order of *SPEED7 Studio* in the 'Licence key' input field.
4. Enter your name in the 'Your name' input field.
5. If you enter your e-mail address in the 'E-mail address' input field, you receive an e-mail confirmation regarding the product activation.
6. Click at 'Activate'.
⇒ The licence is activated and the *SPEED7 Studio* is started.

Registration of *SPEED7 Studio* in the Siemens TIA Portal as *OPC UA Configurator*

SPEED7 Tools Integration is automatically listed in the Windows Start menu during the installation of the *SPEED7 Studio*.

1. ➤ To start the *SPEED7 Tools Integration*, click in the Windows Start menu on 'VIPA GmbH ➔ *SPEED7 Tools Integration*'.
- ⇒ For *SPEED7 Tools Integration* can start, you have to acknowledge the security prompt to change the data on your computer with 'Yes'. Afterwards *SPEED7 Tools Integration* will be started.



2. ➤ Click at 'GSDML ...'.
3. ➤ Navigate to your GSDML file of your VIPA-CPU, which you also use for your configuration in the Siemens 'TIA Portal'. Select these and click at 'Confirm'. You can also select and use several GSDML files.
 - ⇒ The identified GSDML files are listed and the selection for the configuration tools is enabled.
4. ➤ Select 'TIA Portal', in which the *SPEED7 Studio* is to be registered as *OPC UA Configurator*.
5. ➤ Click on 'Start'.
 - ⇒
 - *SPEED7 Studio* is registered in the Windows registry as *OPC UA Configurator*.
 - In the Siemens TIA Portal the *OPC UA Configurator* is registered as externally callable program.
 - The current Windows user is registered in the user group *Siemens TIA Openness* of the Siemens TIA Portal.
 - All changes are recorded in a log file, which you can output via 'Log file'.
6. ➤ 'Close' closes *SPEED7 Tools Integration*.
 - ⇒ With the next start of the Siemens TIA Portal, the *SPEED7 Studio* can be called as *OPC UA Configurator* with to *OPC UA* configuration limited functions. More information about the usage can be found in the in the online help of the *OPC UA Configurator*.

7.6.2 Steps of the OPC UA configuration

Steps of configuration

When using the Siemens TIA Portal, the *OPC UA* configuration happens by the following steps:

1. ➤ Create your project in the Siemens TIA Portal with the corresponding hardware configuration. ↪ *Chap. 13.3 'TIA Portal - Hardware configuration - CPU' page 296*
2. ➤ Configure the corresponding Ethernet connection for PG/OP communication and establish an online connection. ↪ *Chap. 13.4 'TIA Portal - Hardware configuration - Ethernet PG/OP channel' page 299*
3. ➤ Save translate and transfer your project. ↪ *Chap. 13.10 'TIA Portal - Project transfer' page 320*
4. ➤ Call the external *OPC UA Configurator* from the Siemens TIA Portal. For this click at *'Devices & networks'* on the CPU of VIPA and select *'Start device tool'*.
⇒ A dialog window opens. Select *'OPC UA Configurator'* and click [Start].
5. ➤ Ignore the query *'Set interface'* with [OK]
⇒ The *OPC UA Configurator* is started.
6. ➤ If not yet confirmed, you will now receive an access request in the TIA Portal.



Please note that due to the software the access request does not appear in the foreground. To show the access request, you must again bring the Siemens TIA Portal to the foreground. Once the access has been selected, you must again bring the 'OPC UA Configurator' to the foreground.

You have the following options for access:

- *'No'*: Deny access - the *OPC UA Configurator* is not started.
- *'Yes'*: Access is permitted once and the *OPC UA Configurator* is started.
- *'Yes to all'*: Access is permitted and the *OPC UA Configurator* is started. At the next call, the access request is no longer shown.

Allow access with *'Yes'* respectively *'Yes to all'*.

⇒



NOTICE!

Data exchange between platforms of different vendors

If you allow access, you permit the exchange of data between *OPC UA Configurator* and your project data of the Siemens TIA Portal.

- Ensure that the necessary security guidelines are complied with.

For the *OPC UA* configuration, the data is taken from the Siemens TIA Portal project and listed in the table for the *OPC UA* configuration.



Please note that only the objects of the LD, FBD and IL languages can be transferred to the OPC UA Configurator.

7. ➤ Configure the *OPC UA* server and the data for the *OPC UA* communication.

8. In the *OPC UA Configurator* switch to the online dialog and transfer the *OPC UA* configuration. For communication the IP address data are taken from the TIA Portal project.
- ⇒ The *OPC UA* configuration is now complete. For check you will find information about your *OPC UA* configuration on the device web page at 'OPC UA'.
 ↪ Chap. 4.11.1.1.1 'Tab: 'OPC UA'' page 97

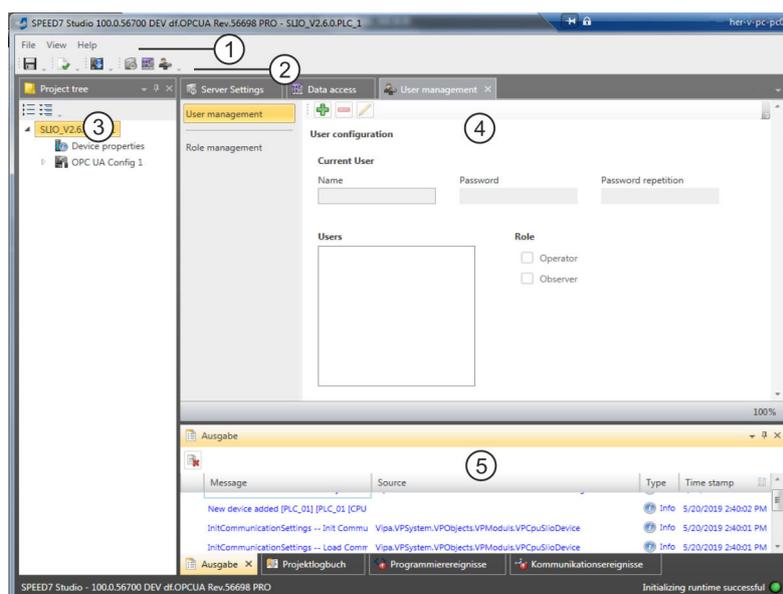


You can use the *CMD* auto commands *OPCUA_PGOP_ENABLE* and *OPCUA_PGOP_DISABLE* to enable or disable *OPC UA*. After a power cycle or loading a hardware configuration, the settings are retained. With reset to the factory settings or over all reset, the *OPC UA* project is set to the default value "enabled". ↪ Chap. 4.18 '*CMD* - auto commands' page 114

7.7 Usage *OPC UA* Configurator

7.7.1 *OPC UA* Configurator

The user interface of the *OPC UA* Configurator is divided into the following areas:



- 1 Menu bar
- 2 Toolbar
- 3 Project tree
- 4 Workspace
- 5 Output area

Menu bar

In the menu bar you will find a few general commands on the *OPC UA* Configurator. Further commands can be called up via context menus with the right mouse button, e.g. functions for an object in the project tree.

Toolbar

-  Store *OPC UA* configuration
-  Compile *OPC UA* configuration



Transfer *OPC UA* configuration into the control

Project tree

The *Project tree* gives you access to the '*Device properties*' and to the following areas of the '*OPC UA configuration*':

- Server settings
- Data access
- User management

Workspace

In the *Work space*, you can edit the settings in the following areas of the *OPC UA* configuration:

- Device properties - General
 - Information about the CPU such as device name, name and firmware version.
- Device properties - Communication
 - Configuration of the interface for data exchange.
 - The IP address data are automatically imported from the project when the *OPC UA Configurator* is called and can be viewed here.
- Device properties - Server configuration
 - Administration and interface assignment of the *OPC UA* server in the *Project tree*
- Server settings - Connection
 - Legitimation of the user for access to the *OPC UA* server.
 - Port for communication.
 - Security policy for encryption and corresponding exceptions.
- Server settings - Certificate
 - Create, view, import or export X.509 ITU-T standard certificate.
 - Re-creating or importing replaces an existing certificate.
- Data access
 - Selection of the variables that can be accessed via *OPC UA*.
 - Filter option to limit the selection.
- User management
 - Creation of a user list with password and role assignment.

Output area

The output area shows information about activities performed and background operations.

7.7.2 Project tree 

You can edit the *OPC UA* configuration via the project tree. The project tree contains the *OPC UA* configurations, which you have created. You can create a maximum of two *OPC UA* configurations: One configuration for the CPU and one configuration for the CP (if exists).

Show project tree

If the project tree is not shown select '*View* → *Project tree*' or press *[Strg]+[Shift]+[P]*.

Show/hide objects

The objects in the project tree are arranged in a tree structure. You can show or hide objects:



Hide all objects ('*Project* → *Collapse project tree*')



Show all objects ('*Project* → *Expand project tree*')



Hide slave objects / close folder

- ▼ Show slave objects / open folder

Edit configurations and OPC UA configuration

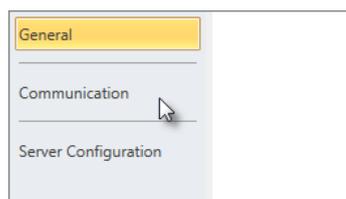
| Device properties | |
|--|---|
|  Device properties | <ul style="list-style-type: none"> ■ Edit device name and comment ↗ <i>Chap. 7.7.3.2 'General device properties' page 201</i> ■ Perform communication settings ↗ <i>Chap. 7.7.3.3 'Communication settings' page 202</i> ■ Create OPC UA configuration ↗ <i>Chap. 7.7.3.4 'Server configuration' page 203</i> |
| OPC UA | |
|  Server settings | <ul style="list-style-type: none"> ↗ <i>Chap. 7.7.4 'Server settings - Connection' page 203</i> ↗ <i>Chap. 7.7.5 'Server settings - Certificate' page 205</i> |
|  Data access | ↗ <i>Chap. 7.7.6 'Data access' page 206</i> |
|  User management | <ul style="list-style-type: none"> ↗ <i>Chap. 7.7.7 'User management' page 207</i> ↗ <i>Chap. 7.7.8 'Role management' page 208</i> |

7.7.3 Device properties

7.7.3.1 Overview

Here you can edit the device name and the comment, perform the communication settings as well as create the *OPC UA* configuration.

- ➔ Click in the project tree at *'Device properties'*.
- ⇒ The *'Device properties'* editor opens.



The *'Device properties'* editor is divided into several sections:

- ↗ *Chap. 7.7.3.2 'General device properties' page 201*
- ↗ *Chap. 7.7.3.3 'Communication settings' page 202*
- ↗ *Chap. 7.7.3.4 'Server configuration' page 203*

7.7.3.2 General device properties

To show or change the device properties, proceed as follows:



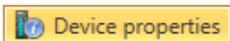
- ➔ Click in the project tree at *'Device properties'*.
⇒ The editor of the *'Device properties'* opens.
- ➔ Select the area *'General'*.

- 'Device type'* - Name of the CPU
- 'Firmware'* - Firmware version of the CPU
- 'Name'* - Device name: This name is shown in the project tree.
- 'Author'* - Name of the responsible person who created the device
- 'Comment'* - Any comment, e.g. an annotation or explanation

- ➔ Click on the input field and enter any comment, e.g. an annotation or explanation. With the *[Enter]* key, you can add a new line to the input field.

7.7.3.3 Communication settings

The communication settings are used to configure the interface for the data exchange between programming device and destination station. Since the IP address parameters for the *OPC UA* configuration are imported from the project, you simply have to set the interface via which you are connected to the destination station.



1. ➔ Click in the project tree at *'Device properties'*.
⇒ The editor of the *'Device properties'* opens.
2. ➔ Select the area *'Communication settings'*.

Communication configurations

Active pc interface:

Properties of Serial interface

PC interface:

COM port Baudrate

CPU interface:

Properties of ethernet interface

PC interface:

IP address:

CPU interface:

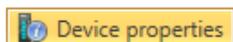
192.168.10.100

Setting the Ethernet interface

1. ➔ *'Active PC interface'*: Select *'Ethernet interface'*.
2. ➔ *'PC interface'*: Select the network adapter for the communication connection from the list.
⇒ If an IP address is already configured in the network adapter, it is shown under the input field *'IP address'*. If necessary, select a different IP address.
3. ➔ *'CPU interface'*: Select the interface of the control from the list.
⇒ Since the IP address is imported from the project, it is shown below the input field.
4. ➔ To configure further settings of the interface, click on *'Interface configuration'*.
⇒ The *'Interface properties'* dialog window will open.

5. ➤ In order to check whether a connection between the programming device and the control can be established with the selected communication settings, click on *'Verify connection'*.
 - ⇒ You can see in the status line, whether the connection could be established successfully.
6. ➤ In order to check whether your programming device is connected with the correct control, you can retrieve information from the connected control. For this click on *'Accessible partners'*.
 - ⇒ The *'Search for accessible partners'* dialog window will open.

7.7.3.4 Server configuration



Here you can create the *OPC UA* configurations.

1. ➤ Click in the project tree at *'Device properties'*.
 - ⇒ The editor of the *'Device properties'* opens.
2. ➤ Select the area *'Server configuration'*.

You can create a maximum of two *OPC UA* configurations: One configuration for the CPU and one configuration for the CP (if exists).

Create configuration

1. ➤ Click at on *'Add Server'*.
 - ⇒ A new *OPC UA* configuration is created and listed in the project tree.
2. ➤ Click in the selection field *'Active server CP'* or *'Active server CPU'* and choose which configuration is to be assigned. With the selection *'None'* the configuration remains saved in the project. However, it is not transferred to the device.

To swap the two configurations for CP and CPU, click on the button .

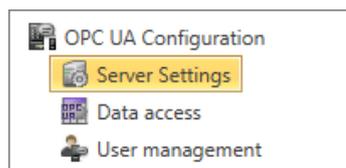
You can create a maximum of two *OPC UA* configurations.

Remove server

- Right-click on the *OPC UA* configuration (PLC) in the project tree and select *'Remove OPC UA server'*.

7.7.4 Server settings - Connection

Here you can perform the connection settings of the *OPC UA* server.



1. ➤ Under *Project tree* at *'OPC UA configuration'* click on *'Server settings'*.
 - ⇒ The *'Server settings editor'* editor opens.
2. ➤ Select the area *'Connection'*.

General

You can set for the *OPC UA* server how a user of an *OPC UA* client must prove their identity for access to the server. Select at least one of the following login methods. You can also combine the two login methods with each other.

- *'Activate anonymous login'*
 - The *OPC UA* server does not check the authorisation of the *OPC UA* client.
- *'Activate user/password login'*
 - The *OPC UA* server checks using the user name and password whether the access of the *OPC UA* client is authorised. To do this, the server evaluates the role assigned to the user. ↗ *Chap. 7.7.8 'Role management' page 208*
- *'Allow obsolete security guideline'*
 - Allows the selection of the two obsolete security guidelines *'Basic128Rsa15'* and *'Basic256'* (not recommended)
- *'Application name'*
 - Clear identification of the application in the OPC name space.

Network

- *'End point port'*
 - TCP port for binary data exchange (standard: 4840).

Security

Activate only security guidelines that are compatible with the protection concept for your machine or system. Deactivate all other security guidelines.

- *'None'*
 - Insecure data traffic between server and client.
- *'Basic128Rsa15'*
 - Secured data traffic, 128-bit encoding with key wrap algorithm RSA-15, (allow option with *'Allow obsolete security guideline'* see above).
- *'Basic256'*
 - Secured data traffic, 256-bit encoding (allow option with *'Allow obsolete security guideline'* see above).
- *'Basic256Sha256'*
 - Secured data traffic, 256-bit encoding with hash algorithm SHA-256 (recommended).

Encoding:

- *'Sign'*
 - Endpoint secures the integrity of the data through signing.
- *'SignAndEncrypt'*
 - Endpoint secures the integrity and confidentiality of the data through signing and encoding.
- *'Both'*
 - The *OPC UA* server offers both encryption methods *'Sign'* and *'SignAndEncrypt'*. The *OPC UA* client can use one of the two encoding methods.

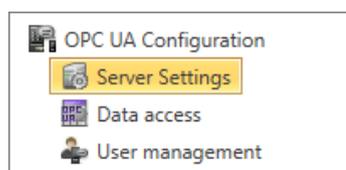
Security Check Overrides

Here you can allow various exceptions in the security check, in order to increase the error tolerance.

7.7.5 Server settings - Certificate

A secure connection between the *OPC UA* client and the server can only be established if the server classifies and accepts the client's digital certificate as trusted. Currently, the server accepts every valid client certificate. The server accepts self-signed certificates. In addition, the client also checks the server's certificate.

Here you can create, show, import or export an ITU-T standardized X.509 certificate for the *OPC UA* server. The certificate shown here is transferred into the *OPC UA* server.



1.  Under *Project tree* at '*OPC UA configuration*' click on '*Server settings*'.
⇒ The '*Server settings*' editor opens.
2.  Select the area '*Certificate*'.

The current X.509 certificate is shown in the work space. If you create or import a new certificate, the previously shown certificate is replaced.

Toolbar

-  **Create new certificate:** Opens the dialog window '*Create new certificate*'
-  **Display certificate:** Shows information on the current certificate
-  **Export certificate:** Opens the dialog window '*Save certificate*'
-  **Import certificate:** Opens the dialog window '*Open certificate*'

Create new certificate

1.  Click on  to create a new certificate.
⇒ The dialog window '*Create new certificate*' opens.
2.  Enter the data for the certificate and click on '*OK*'.
⇒ The previously shown certificate is replaced by the new certificate.

Display certificate

-  Click on  to show information about the current certificate.
⇒ The dialog window '*Certificate*' opens.

Export certificate

You can export the current certificate e.g. to use it on different computers.

1.  Click on .
⇒ The dialog window '*Save certificate*' opens.
2.  Select a directory and enter a file name.
3.  Click on '*Save*'.
⇒ The current certificate is saved in the export file (pfx file format).

Import certificate

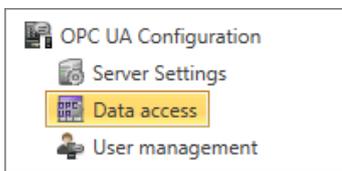
You can import a certificate, e.g. to use it for the current *OPC UA* configuration. For a successful import, the certificate must have the following characteristics:

- The certificate must be available as a PFX file.
- The '*Common name*' and '*Organization*' fields must be completed.

- The maximum key strength must not exceed 2048bit.
 - The certificate must contain a valid *Private key*.
1. Click at .
 - ⇒ The dialog window *'Open certificate'* opens.
 2. Select the desired certificate (pfx file format).
 3. Click on *'Open'*.
 - ⇒ The previously shown certificate is replaced by the imported certificate.

7.7.6 Data access

Here you can select the variables belonging to the CPU or CP (if exists) that can be accessed via *OPC UA*.



- Under *Project tree* at *'OPC UA configuration'* click on *'Data access'*.
 - ⇒ The editor for *'Data access'* opens.

Toolbar



Refresh variables: Apply changed filter settings to the result table.

Filter settings

Here you can select the operands and address ranges that will be shown in the results table.

1. Activate *'All operand areas'* or individual operand ranges to be shown in the result table.
2. To limit the addresses of an operand range, enter the start and end byte addresses in the two adjacent fields, e.g. 0 to 1000.
3. Click on  or activate *'Apply filter changes immediately'*.
 - ⇒ The result table is updated with the filter settings.

Result

In the results table, select the variables that are to be used in the *OPC UA* configuration. *OPC UA* clients may access these variables.

- Activate *'OPC UA'* of the desired variables.

Group operands

For a better overview, you can sort the table entries by groups.



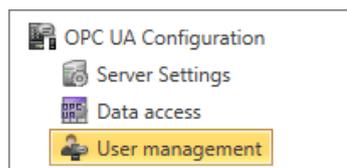
- (1) Select column (hold left mouse button down)
 - (2) Drag the column
 - (3) Drop column in the field (release mouse button)
1. Drag the desired column header into the field above the table.
 - ⇒ The contents of the column will be grouped. The number of lines is shown for each group.

2. Click on ▶ to open the group. Click on ▼ to close the group.

You can repeat steps 1 to 2 in order to structure the group into further sub-groups. In order to cancel a grouping, click on the close icon next to the group name.

7.7.7 User management

The user management allows you to create a user list. For each user, you can define a password and a role.



1. Under *Project tree* at '*OPC UA configuration*' click on '*User management*'.
⇒ The editor for '*User management*' opens.
2. Select the area '*User management*'.

Toolbar

-  **Add new user:** Input mode for new user
-  **Remove user:** Deletes the selected user
-  **Edit current user:** Input mode for selected user
-  **Save input:** Save input Save user settings
-  **Cancel input:** Cancel user settings without saving

Adding a user

1. Click on .
2. Enter the desired user name in the input field '*Name*'.
3. Enter the password in the input field '*Password*' and repeat the input under '*Re-enter password*'.
4. Select a role for the user. With this role, the access rights to the *OPC UA* server are established.
5. Click on .
⇒ The user will be entered in the user list.

Edit user

1. In the user list, select the user whose data you want to change.
2. Click on .
3. Enter the desired changes and click on .

Removing a user

1. In the user list, select the user you want to delete.
2. Click on .
⇒ A dialog box opens where you can choose whether the user should be deleted or not.

7.7.8 Role management

Here you establish the roles and access rights that you can assign to the users. When you activate the authentication via User/password login [Chap. 7.7.4 'Server settings - Connection' page 203](#), the access rights to the OPC UA server are issued using the logged-in user and the assigned role.

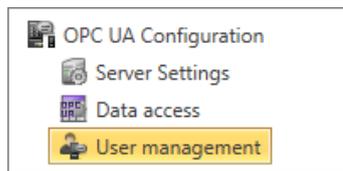
Example:

Role: Operator

Username: "I myself"

Server settings: User/password login activated

The user "Me Self" receives write permission and reading rights to the OPC UA server when he has successfully logged in with the password.



1. Under Project tree at 'OPC UA configuration' click on 'User management'.

⇒ The editor for 'User management' opens.

2. Select the area 'Role management'.

Configure roles

The following two roles are currently available for selection; further roles can not be added at the moment.

- Operator: Write permission and reading rights
- Observer: Reading rights only

7.7.9 Output

Information on executed activities and background operations are displayed in the "Output" window.



(1) Delete all messages in the output window

8 Deployment *WebVisu* - Web visualization



Please note that the simultaneous use of OPC UA and WebVisu on the same interface is not supported! When attempting to activate them, both servers are stopped and the diagnostic message 0xE989 or 0xE9AB is output.

- With a *WebVisu* project there is the possibility to configure a web visualization on your CPU.
- The configuration of a *WebVisu* project is only possible with the *SPEED7 Studio* V1.7.0 and up.
- Since a *WebVisu* project is only executable by memory card, a memory card of VIPA (VSD, VSC) must be plugged. ↪ *Chap. 4.16 'Deployment storage media - VSD, VSC' page 111*
- If the memory card is removed for a short time, the SF LED lights up. This indicates that a feature is missing and the *WebVisu* will quit after 72 hours.
- The *WebVisu* functionality must be activated in the CPU. ↪ *Chap. 8.2 'Activate Web-Visu functionality' page 212*
- When the project is transferred from the *SPEED7 Studio*, the *WebVisu* project is automatically transferred as TAR file to the inserted memory card.
- Access happens by the IP address of the Ethernet PG/OP channel and the correspondingly configured port or via the *device web page*
- You can access your web visualization via a web browser. Web browsers based on Windows CE are currently not supported.



Please note that the use of a WebVisu project, depending on the scope of the WebVisu project and the PLC project, can influence the performance and thus the response time of your application.

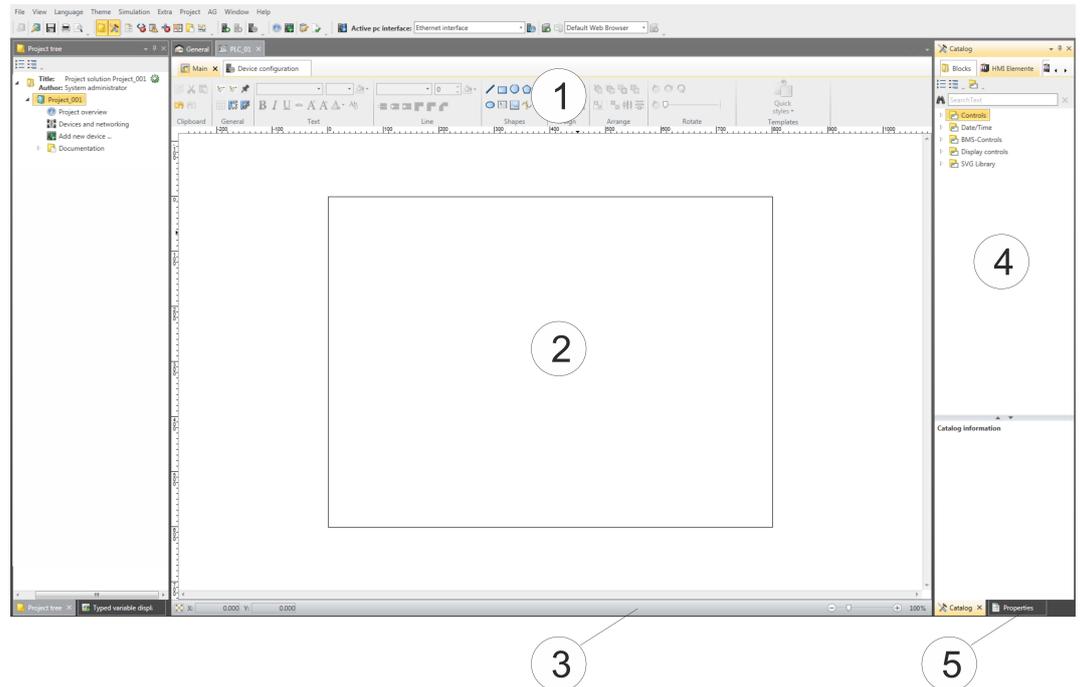
8.1 *WebVisu* editor

The configuration of a *WebVisu* project is shown below. This is only the basic use of the *WebVisu* editor in the *SPEED7 Studio* together with the VIPA CPU. Please note that software changes can not always be considered and it may thus be deviations to the description.



For more information on the SPEED7 Studio and how to use the Web-Visu editor, refer to its online help.

8.1.1 Working environment



- (1) Toolbar
- (2) Editor surface
- (3) Status bar
- (4) Catalog
- (5) Properties window

(1) Toolbar

The toolbar provides important commands for working with the *WebVisu* editor.

(2) Editor area

The editor area is your workspace. Here you can place and edit texts and graphics objects.

(3) Status bar

With a slider you can enlarge or reduce your view.

(4) Catalog

Via *Catalog* you can access all the *WebVisu* elements. You can use Drag & Drop to place them on the *Editor surface* and adjust them using properties.

(5) Properties window

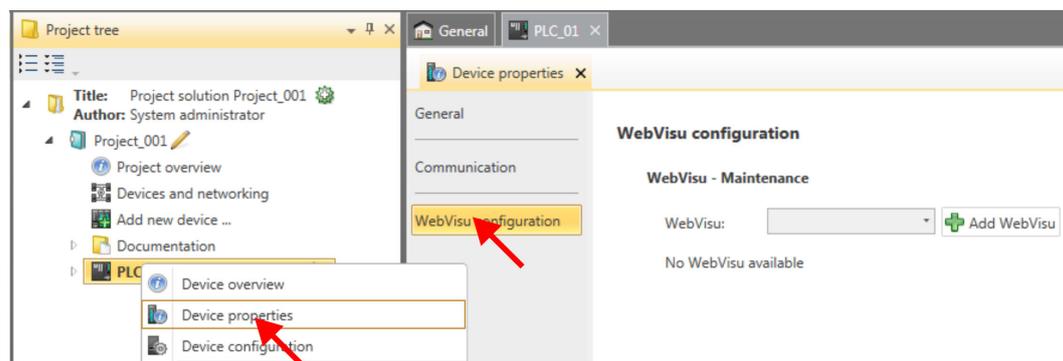
By enabling 'View → Properties' the 'Properties' are shown. The properties of the selected element are shown here. You can adjust these if necessary.

8.1.2 Creating a *WebVisu* project

Add *WebVisu*

1. ➤ Start the *SPEED7 Studio* with your project for the CPU for which a *WebVisu* project is to be created.
2. ➤ If not already done, add a CPU by clicking 'Add new device'.
3. ➤ Click in the 'Project tree' at the CPU and select 'Context menu → Device properties'.
 - ⇒ The 'Device properties' of your CPU opens.

4. Here click at '*WebVisu configuration*'
 ⇒ In this settings window, you can create a *WebVisu* project for your CPU.



5. To create a *WebVisu* project, click at [+ Add WebVisu].
 ⇒ A new *WebVisu* project is created and listed in the '*Project tree*'. At '*WebVisu - general configurations*' and '*WebVisu - SSL configurations*', you can make further settings.

WebVisu - general configurations

- Port number
 - Enter the port number under which the *WebVisu* should be accessible.
 - *Port number: 8080 (default):* The *WebVisu* can be accessed via the IP address and port 8080. The *Device web page* can be accessed via the IP address and port 80.
 - *Port number: 80:* The *WebVisu* can be accessed via the IP address and port 80. The *Device web page* can be accessed via the IP address and port 8080.
- Polling interval (ms)
 - Enter here the interval for the cyclical refresh of the web content.
- Execution device
 - Select '*CPU*' as device on which this *WebVisu* project is to be executed.
 - *WebVisu* projects for Ethernet CPs are not supported by this CPU.

WebVisu - SSL configurations

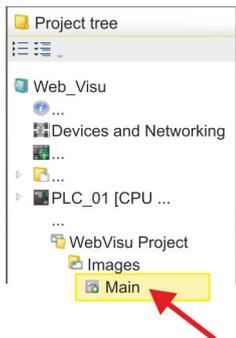
- Enabling encoding
 - When enabled, you have SSL-encrypted access to your *WebVisu*.
- Disable HTTP
 - When activated, the access happens via HTTPS.
- SSL port number
 - SSL port number 443 (default): The secure access to the *WebVisu* takes place via the IP address and port 443. The *Device web page* can be accessed via the IP address of the CPU and port 8080.
- Original path of the certificate used
 - Here you can upload a security certificate.
 - Only security certificates in PEM format are supported.
 - The file must contain the certificate and the private key.

Delete *WebVisu*

- Click in the '*Project tree*' at *WebVisu* Project and select '*Context menu*'
 → *Delete WebVisu*'.
 ⇒ The *WebVisu* project is removed from the configuration.

Start-up of the *WebVisu* project

Edit *WebVisu*



➔ In the 'Project tree', navigate to '*WebVisu Project > Images*' and click at '*Main*'. Select '*Context menu → Open image*'

⇒ The *WebVisu* editor opens. Here, you can configure your web visualization by dragging and dropping elements from the '*Catalog*' onto the editor area and using the '*Properties*' to interconnect them with a variable.

8.2 Activate *WebVisu* functionality

Proceeding

For your CPU can process a *WebVisu* project, you have to activate the *WebVisu* functionality.

1. ➔ Insert a memory card from VIPA (VSD, VSC) into your CPU. Please note that you must always use a VSC card suitable for your CPU. ↪ *Chap. 4.16 'Deployment storage media - VSD, VSC' page 111*
2. ➔ Turn on the CPU, to activate the *WebVisu* functionality, you have to establish an *Overall reset*.
 - ⇒ As long as the memory card is inserted, the *WebVisu* functionality remains activated even after a power cycle. When the project is transferred from the *SPEED7 Studio*, the *WebVisu* project is automatically transferred to the inserted memory card.



*Please regard that the memory card must remain plugged when you've executed activated the *WebVisu* functionality. Otherwise the SF LED is on and the *WebVisu* functionality is deactivated after 72 hours. As long as an activated memory card is not inserted, the SF LED is on and the "TrialTime" timer counts from 72 hours down to 0. After that the *WebVisu* functionality is de-activated. By inserting the memory card, the LED goes off and the CPU runs again without restrictions.*

8.3 Start-up of the *WebVisu* project

The following preconditions must be fulfilled for the *WebVisu* project to start-up:

1. ➔ Activate *WebVisu* functionality if not already done. ↪ *Chap. 8.2 'Activate *WebVisu* functionality' page 212*
2. ➔ Configure your CPU and perform a hardware configuration.
3. ➔ Configure your *WebVisu* project.
4. ➔ Save and translate your project.
5. ➔ If you are online connected to your CPU, you can transfer your project to the CPU with '*AG → Transfer all*'.
 - ⇒ Here the configuration is transferred in the CPU and the *WebVisu* project is transferred to the memory card. Immediately after the transfer you have access to your *WebVisu*.



You can use the CMD auto commands `WEBVISU_PGOP_ENABLE` and `WEBVISU_PGOP_DISABLE` to enable or disable the *WebVisu*. After a power cycle or loading a hardware configuration, the settings are retained. With reset to the factory settings or over all reset, the *WebVisu* project is set to the default value "enabled". ↪ Chap. 4.18 'CMD - auto commands' page 114

8.4 Access to the *WebVisu*

- When connected via Ethernet PG/OP channel, you have access to the *WebVisu* and *Device web page* of the CPU, controlled via ports.
- Access to the *WebVisu* can be password-protected and encrypted by means of SSL certificates. If you want to use SSL certificates, you must integrate them in the *SPEED7 Studio* accordingly.
- You can create users in the *SPEED7 Studio* via '*WebVisu project* > *User administration*', who can access the *WebVisu*.
- Via '*Device properties* > *WebVisu configuration*' you can specify the port, the *WebVisu* can be accessed and upload SSL certificates. This changes the port for accessing the device web page.
 - *Port number*: 8080 (default): The *WebVisu* can be accessed via the IP address and port 8080. The *Device web page* can be accessed via the IP address and port 80.
 - *Port number*: 80: The *WebVisu* can be accessed via the IP address and port 80. The *Device web page* can be accessed via the IP address and port 8080.
 - SSL port number 443 (default): The secure access to the *WebVisu* takes place via the IP address and port 443. The *Device web page* can be accessed via the IP address and port 8080.



- Please note that once you have made adjustments to the user management, you will need to restart your web browser. Otherwise you receive system-related error messages about invalid user information!
- Please note that the encryption of the communication can affect CPU performance and therefore the response time of the entire system!

8.4.1 Status of the *WebVisu*

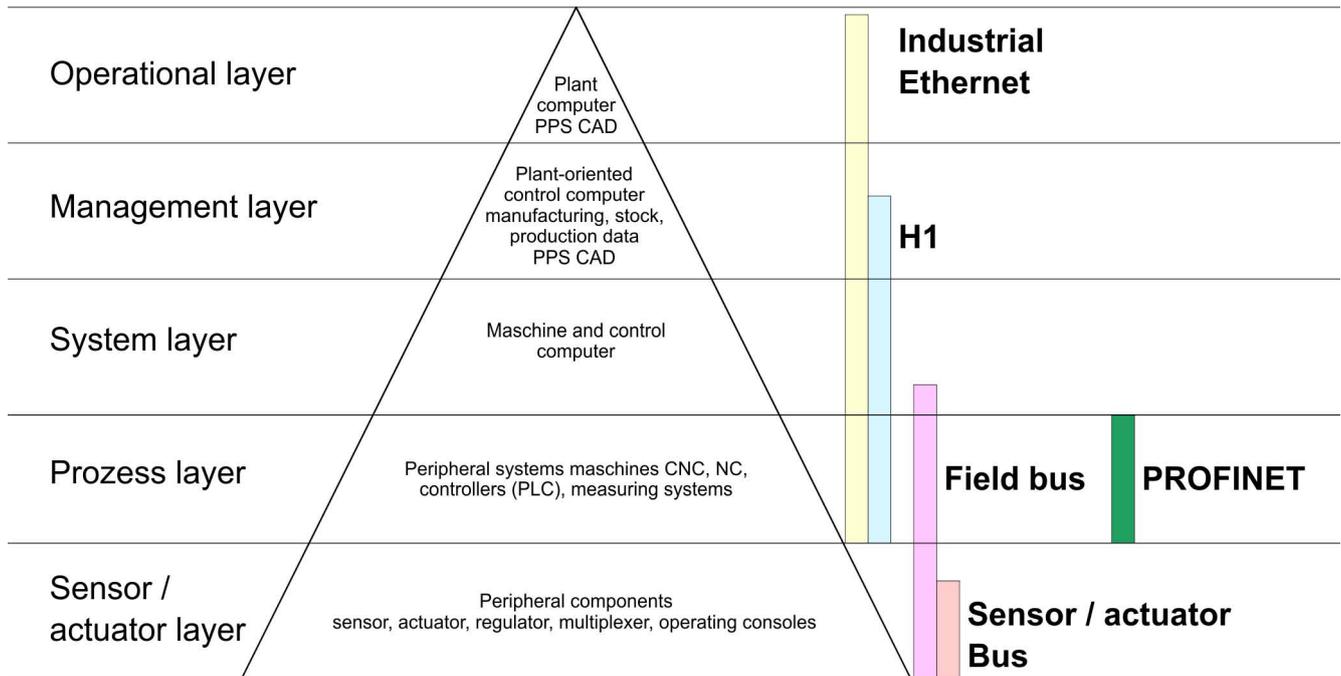
On the device web page at the tab '*WebVisu*' via '*Status*' you get the status of your *WebVisu* project. ↪ Chap. 4.11.1.1.2 'Tab: '*WebVisu*'' page 98

9 Deployment PG/OP communication - productive

9.1 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.

9.2 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.

9.3 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 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.

9.4 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 computer communication is the so called ISO/OSI layer model, a model based upon seven layers with rules for the usage of hardware and software ↪ *Chap. 9.2 'Basics - ISO/OSI reference model' page 215*

The following protocols are used:

- Siemens S7 connections
- Open communication
 - TCP native 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. Here NetPro from Siemens may be used.

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 also part of the Siemens SIMATIC Manager. You will find these in the *'Standard Library'* at *'Communication 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. 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 accord. to RFC 1006:*

During data transmission, information on the length and the end of the message is also transmitted. 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 accord. 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. 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.

9.5 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 **Network-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**.

Basics - IP address and subnet

| | | | |
|------------------------------|----------------|----------------|-------------|
| 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 start-up of the CPU, the Ethernet PG/OP channel does not have an IP address.

Information about the assignment of IP address data to the Ethernet PG/OP channel may be found in [Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel'](#) page 72.

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. <u>0.0.0</u> | 10. <u>255.255.255</u> | 255. <u>0.0.0</u> |
| B | 172.16. <u>0.0</u> | 172.31. <u>255.255</u> | 255.255. <u>0.0</u> |
| C | 192.168.0. <u>0</u> | 192.168.255. <u>255</u> | 255.255.255. <u>0</u> |

(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.)

9.6 Fast introduction

Overview

At the first commissioning respectively after an overall reset with PowerON again of the CPU, the Ethernet PG/OP channel has no IP address. This can only be reached by its MAC address. By means of the MAC address, which is printed at the front as 'MAC PG/OP:...', you can assign IP address data. The assignment takes place directly via the hardware configuration of the Siemens SIMATIC Manager.

Steps of configuration

For the configuration of the Ethernet PG/OP channel for productive connections please follow the following approach:

- Hardware configuration - CPU
- Hardware configuration - Ethernet PG/OP channel
- Configure connections
 - Siemens S7 connections
(Configuration via Siemens NetPro, communication via VIPA handling blocks)
 - Open communication
(Configuration and communication happens by standard handling blocks)
- Transfer of the complete project to CPU

9.7 Hardware-Konfiguration

Overview

At the first commissioning respectively after an overall reset with PowerON again of the CPU, the Ethernet PG/OP channel has no IP address. This can only be reached by its MAC address. By means of the MAC address, which is printed at the front as 'MAC PG/OP:...', you can assign IP address data. The assignment takes place directly via the hardware configuration of the Siemens SIMATIC Manager.

- CPU
 - ↳ *Chap. 4.5 'Hardware configuration - CPU' page 69*
- Ethernet PG/OP channel
 - ↳ *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 72*

9.8 Configure Siemens S7 connections

Overview

The project engineering of connections i.e. the "link-up" between stations happens in NetPro from Siemens. NetPro is a graphical user interface for the link-up of stations. A communication connection enables the program controlled communication between two participants at the Industrial Ethernet. The communication partners may here be part of the same project or - at multi projects - separated within related part projects. Communication connections to partners outside of a project are configured via the object "In unknown project" or via deputy objects like "Other stations" or Siemens "SIMATIC S5 Station". 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.

- ↳ *'Link-up stations' page 223*
- ↳ *'Projecting connections' page 224*
- ↳ *'Siemens S7 connection - Communication functions' page 226*

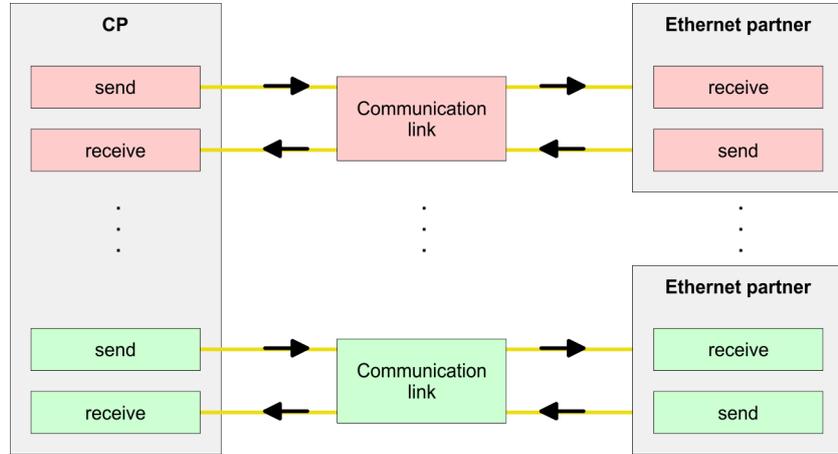
Properties communication connection

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)

Configure Siemens S7 connections

- 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.



Requirements

- Siemens SIMATIC Manager V 5.5 SP2 or higher and SIMATIC NET are installed.
- With the hardware configuration the according CP was assigned with IP address data by its properties.

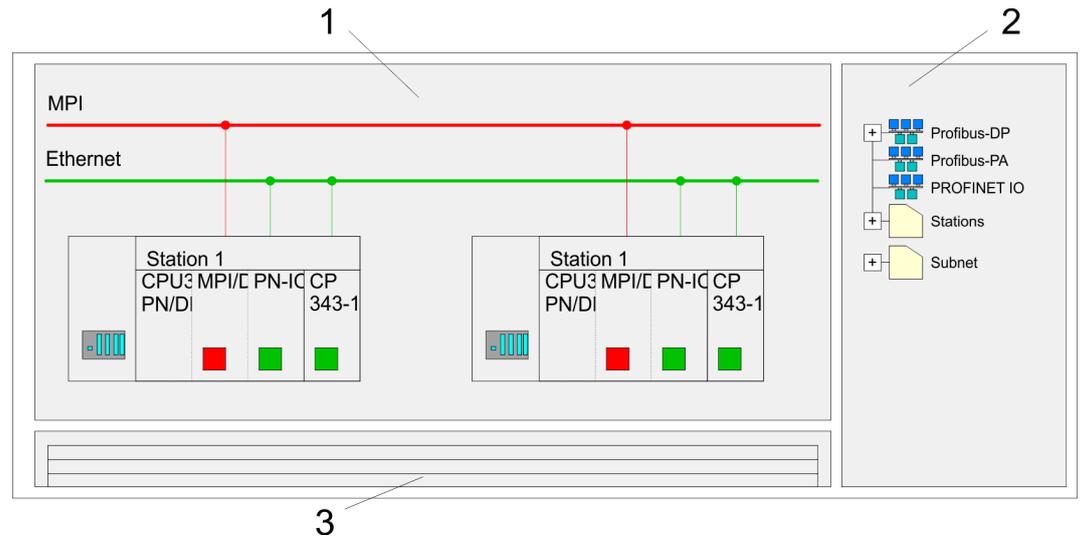


Every station outside of the recent project must be configured as replacement objects like e.g. Siemens "SIMATIC S5" or "other station" or with the object "In unknown project". When creating a connection you may also choose the partner type "unspecified" and set the required remote parameter directly in the connection dialog.

Work environment of NetPro

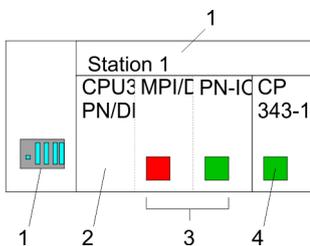
For the project engineering of connections, a thorough knowledge with NetPro from Siemens is required! The following passage only describes the basic usage of NetPro. More detailed information about NetPro is to be found in the according online manual res. documentation. Start NetPro by clicking on a "net" in the Siemens SIMATIC Manager or on "connections" within the CPU.

The environment of NetPro has the following structure:



- 1 **Graphic 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 **Net objects:** This area displays all available net objects in a directory view. By dragging a wanted object to the net view you may include further net objects and open them in the hardware configurator.
- 3 **Connection table:** The connection table lists all connections in a table. This list is only shown when you highlighted a connectable module like e.g. a CPU. You may insert new connections into this table with the according command.

PLC stations

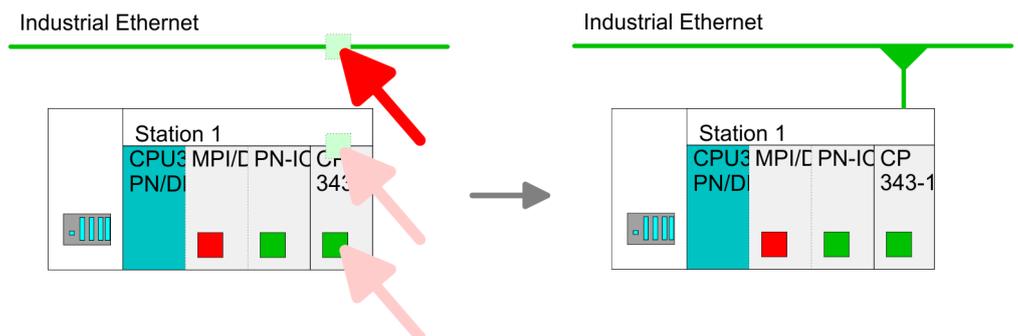


You receive the following graphical display for every PLC station and their component. By selecting the single components, the context menu offers you several functions:

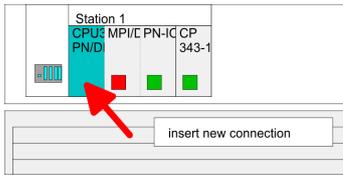
- 1 **Station:** This includes a PLC station with rack, CPU and communication components. Via the context menu you may configure a station added from the net objects and its concerning components in the hardware configurator. After returning to NetPro, the new configured components are shown.
- 2 **CPU:** A click onto the CPU shows the connection table. The connection table shows all connections that are configured for the CPU.
- 3 **Internal communication components:** This displays the communication components that are available in your CPU. The PROFINET IO controller is to be configured by the PN-IO component.
- 4 **Ethernet PG/OP channel:** The internal Ethernet PG/OP channel must always be configured as external CP in the hardware configuration.

Link-up stations

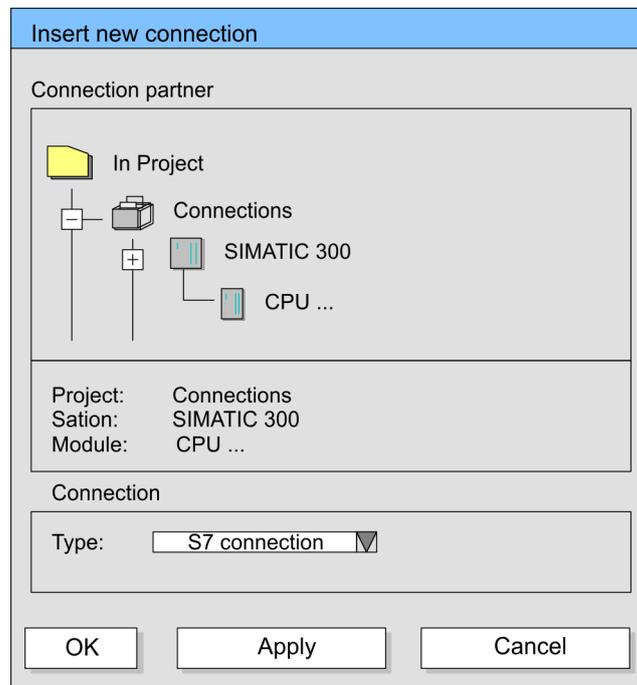
NetPro offers you the option to link-up the communicating stations. You may link-up the stations via the properties in the hardware configuration or graphically via NetPro. For this you point the mouse on the coloured net mark of the according CP and drag and drop it to the net you want to link. Now the CP is linked up to the wanted net by means of a line.



Projecting connections



1. ➔ For the project engineering of connections, open the connection list by selecting the according CPU. Choose *Insert new connection* in the context menu:
 - **Connection partner (partner station)**
A dialog window opens where you may choose the connection partner and the *connection type*.
 - **Specified connection partner**
Each station configured in the Siemens SIMATIC Manager is listed in the table of connection partner. These stations are unique specified by an IP address and a subnet mask.
 - **Unspecified connection partner**
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 connection name, which is to be used in the projects of both stations. Due to this allocation the connection remains *unspecified*.
2. ➔ Choose the connection partner and the type of connection and confirm with [OK].
 - ⇒ If activated, a properties dialog for the according connection opens as link to your PLC user program.



3. ➔ After every connection was configured by this way, you may save and compile your project and exit NetPro.

Connection types

With this CPU exclusively Siemens S7 connection may be configured with Siemens NetPro.

Siemens S7 connection

- For data transfer with Siemens S7 connections the FB/SFB VIPA 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:

The following parameters define a connection end point:

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

Combination options with deployment of the FB/SFB VIPA handling blocks

| Connection partner | Connection establishing | Connection |
|---|-------------------------|---|
| specified in NetPro (in the current project) | active/passive | specified |
| unspecified in NetPro (in the current project) | active | specified |
| | passive | unspecified |
| unspecified in NetPro (in the unknown project) | active/passive | specified (connection name in an other project) |

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

- **Local connection end point:**
Here you may define how the connection is to be established. Since the Siemens SIMATIC Manager can identify the communication options by means of the end points, some options are already preset and may not be changed.
 - **Establish an active connection:**
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 establish a connection. Here the job is to be made by the partner station.
 - **One-way:**
If activated only one-way communication blocks like PUT and GET may be used for communication in the user program. Here the partner station acts as server, which neither may send active nor receive active
- **Block parameters**
 - **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 VIPA handling block.
 - **[Default]:**
As soon as you click at [Default], the ID is reset to system generated ID.
- **Connection path:**
In this part of the dialog window the connection path between the local and the partner station may be set. Depending on the linking of the modules the possible interfaces for communication are listed in a selection field.
 - **[Address details]:**
With this button a dialog window is opened, which shows address information about the local and partner station. The parameters may also be changed.
 - **TSAP:**
With Siemens S7 connections a TSAP is automatically generated of the connection resource (one-way/two-way) and state of place (rack/slot respectively system internal ID at PC stations).
 - **Connection resource:**
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.

Siemens S7 connection - Communication functions

With the SPEED7 CPUs of VIPA there are two possibilities for the deployment of the communication functions:

- **Siemens S7-300 communication functions:**
By integration of the function blocks FB 12 ... FB 15 from VIPA you may access the Siemens S7-300 communication functions.
- **Siemens S7-400 communication functions:**
For the Siemens S7-400 communication functions the SFB 12 ... SFB 15 are to be used, which were integrated to the operating system of the CPU. Here copy the interface description of the SFBs from the standard library at system function block to the directory container, generate an instance data block for each call and call the SFB with the associated instance data block.

Function blocks

| FB/SFB | Label | Description |
|-----------|-------|--|
| FB/SFB 12 | BSEND | <p>Sending data in blocks:</p> <p>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. With this type of data transfer, more data can be transported between the communications partners than is possible with all other communication FBs/SFBs for configured S7 connections, namely 65534bytes.</p> |
| FB/SFB 13 | BRCV | <p>Receiving data in blocks:</p> <p>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.</p> |
| FB/SFB 14 | GET | <p>Remote CPU read:</p> <p>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.</p> |
| FB/SFB 15 | PUT | <p>Remote CPU write:</p> <p>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.</p> |

9.9 Configure Open Communication

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 Siemens SIMATIC Manager. You will find these in the "Standard Library" at "Communication Blocks". 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. First you have to establish a hardware configuration of the CPU and Ethernet PG/OP channel before you can use the handling blocks.

Hardware configuration:

- CPU
 - ↳ *Chap. 4.5 'Hardware configuration - CPU' page 69*
- Ethernet PG/OP channel
 - ↳ *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 72*

To specify the Ethernet PG/OP channel, the following values are defined in the UDT 65:

- *local_device_id*
 - 00h: Ethernet PG/OP channel of the CPU
- *next_staddr_len*
 - 01h: Ethernet PG/OP channel of the CPU
- *next_staddr*
 - 04h: Ethernet PG/OP channel of the CPU

Configure Open Communication

UDTs

| FB | Designation | Connection-oriented protocols: TCP native as per RFC 793, ISO on TCP as per RFC 1006 | Connectionless protocol: UDP according to 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 |

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

FBs

| FB | Designation | Connection-oriented protocols: TCP native as per RFC 793, ISO on TCP as per RFC 1006 | Connectionless protocol: UDP according to 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 |

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

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.

10 Deployment PG/OP communication - PROFINET



- With firmware version V2.4.0, there is a PROFINET IO controller available via the Ethernet PG/OP channel.
- As soon as you use the PROFINET functionality via the Ethernet PG/OP channel, this affects the performance and response time of your system and due to the system the cycle time of the OB 1 is extended by 2ms.

10.1 Basics PROFINET

General

- PROFINET is an open Industrial Ethernet Standard from PROFIBUS & PROFINET International (PI) for automation.
- PROFINET is standardized in the IEC 61158.
- PROFINET uses TCP/IP and IT standards and supplements the PROFIBUS technology for applications, where fast data communication with industrial IT functions is demanded.

There are 2 PROFINET function classes:

- PROFINET IO
- PROFINET CBA

These may be realized in 3 performance steps:

- TCP/IP communication
- RT communication
- IRT communication

PROFINET IO

- With PROFINET IO an I/O data sight to the distributed periphery is described.
- PROFINET IO describes the whole data transfer between IO controller and IO device.
- PROFINET is configured like PROFIBUS.
- PROFINET IO always contains the real time concept.
- Contrary to the master-slave procedure of PROFIBUS, PROFINET uses the provider-consumer model. This supports the communication relations (AR = Application Relation) between equal participants in the Ethernet. Here the provider sends its data without a request of the communication partner.
- Apart from the user data exchange also functions for parametrization and diagnostics are supported.

PROFINET CBA

- PROFINET CBA means **C**omponent **B**ased **A**utomation.
- This component model describes the communication between autonomously working stations.
- It makes a simple modularization of complex plants possible, by distributed intelligence by means of graphic configuration for communication of intelligent modules.

TCP/IP communication

This is the open communication via Ethernet TCP/IP without any demand on real-time.

RT Communication

- RT means **R**eal-**T**ime.
- The RT communication represents the basics for data transfer at PROFINET IO.
- Here RT data are handled with higher priority.

- IRT Communication**
- IRT means **I**sochronous **R**eal-**T**ime.
 - With the IRT communication the bus cycle begins clock-exactly i.e. with a maximum permissible tolerance and is again synchronized. Thereby the time-controlled and synchronous transfer of data is guaranteed.
 - Here sync telegrams of a sync master in the network serve for.
- Properties of PROFINET**
- PROFINET of IEC 61158 has the following properties:
- Full-duplex transfer with 100MBit/s via copper respectively fibre optics.
 - Switched Ethernet
 - Auto negotiation (negotiates the transfer parameters)
 - Auto crossover (transmission and receipt lines are crossed automatically if necessary)
 - Wireless communication via WLAN
 - UDP/IP is used as overlaid protocol. UDP means **U**ser **D**atagram **P**rotocol and contains the unprotected connectionless broadcast communication within IP.
- PROFINET devices**
- Like PROFIBUS DP also with PROFINET IO the following devices are classified according to their tasks:
- IO controller
 - The *IO controller* is equivalent to the master of PROFIBUS.
 - This is the PLC with PROFINET connection, in which the PLC program runs.
 - IO device
 - The *IO device* is a distributed I/O field device, which is connected to PROFINET.
 - The IO device is equal to the slave of PROFIBUS.
 - IO supervisor
 - The *IO supervisor* is an engineering station as e.g. programming unit, PC or HMI interface for commissioning and diagnostics.
- AR**
- AR (**A**pplication **R**elation) corresponds to a connection to an IO controller or IO supervisor.
- API**
- API means **A**pplication **P**rocess **I**dentifier and defines besides *Slot* and *Subslot* a further addressing level.
 - With this additional addressing mode with using of different applications, the overlapping of data areas can be prevented.
 - The following APIs are currently supported by the PROFINET IO devices from VIPA:
 - DEFAULT_API (0x00000000)
 - DRIVE_API (0x00003A00)
 - ENCODER_API (0x00003D00)
 - FIELDBUS_INTEGRATION_API (0x00004600)
 - PROFINET_IO_LINK_API (0x00004E01)
 - RFID_READER_API (0x00005B00)
 - BARCODE_READER_API (0x00005B10)
 - INTELLIGENT_PUMP_API (0x00005D00)
 - PROCESS_AUTOMATION_API (0x00009700)
- GSDML file**
- To configure a device I/O connection in your own configuration tool, you've got all the information about your PROFINET components in form of a GSDML file. This file may be found for System SLIO in the download area of www.vipa.com at '*Config files* → *PROFINET*'.
 - Please install the GSDML file in your configuration tool.

- More information about installing the GSDML file may be found at the manual of the according engineering tool.
- Structure and content of the GSDML file are defined by IEC 61158.

Addressing

In contrast to the PROFIBUS address, in PROFINET each device may be definitely identified with its PROFINET interface:

- Device name
- IP address respectively MAC address

Transfer medium

PROFINET is compatible to Ethernet in accordance with the IEEE standards. The connection of the PROFINET IO field devices is exclusively established via switches as network components. This is made either as star via multi-port switches or as line by means of switches, integrated to the field devices.

10.2 PROFINET installation guidelines

Generals to data security

The topic of data security and access protection have become increasingly important in the industrial environment. The increased networking of entire industrial systems to the network levels within the company together with the functions of remote maintenance have all served to increase vulnerability. Threats can arise from internal manipulation like technical errors, operator and program errors respectively from external manipulation like software viruses and worms, trojans and password phishing.

Precautions

The most important precautions to prevent manipulation and loss of data security in the industrial environment are:

- Encrypting the data traffic by means of certificates.
- Filtering and inspection of the traffic by means of VPN - "Virtual Private Networks".
- Identification of the nodes by "Authentication" via save channels.
- Segmenting in protected automation cells, so that only devices in the same group can exchange data.

Guidelines for information security

With the "VDI/VDE 2182 sheet 1", Information Security in the Industrial Automation - General procedural model, VDI guidelines, the VDI/VDE society for measuring and automation engineering has published a guide for implementing a security architecture in the industrial environment. The guideline can be found at www.vdi.de PROFIBUS & PROFINET International (PI) can support you in setting up security standards by means of the "PROFINET Security Guideline". More concerning this can be found at the corresponding web site e.g. www.profibus.com

Industrial Ethernet

- Due to the open standard of PROFINET standard Ethernet components may be used.
- For industrial environment and due to the high transfer rate of 100MBit/s your PROFINET system should consist of Industrial Ethernet components.
- All the devices interconnected by switches are located in one and the same network. All the devices in a network can communicate directly with each other.
- A network is physically limited by a router.
- If devices need to communicate beyond the limits of a network, you have to configure the router so that it allows this communication to take place.

Topology

- Linear
 - With the linear structure all the communication devices are connected via a linear bus topology.
 - Here the linear bus topology is realized with switches that are already integrated into the PROFINET device.
 - If a communication member fails, communication across the failed member is no longer possible.
- Star
 - If you connect communication devices to a switch with more than 2 PROFINET interfaces, you automatically create a star network topology.
 - If an individual PROFINET device fails, this does not automatically lead to failure of the entire network, in contrast to other structures.
 - It is only if a switch fails that part of the communication network will fail as well.
- Ring
 - In order to increase the availability of a network the both open ends of a linear bus topology may be connected by a switch.
 - By configuring the switch as redundancy manager on a break in the network it ensures that the data is redirected over an intact network connection.
- Tree
 - If you interconnect several star structures, you obtain a tree network topology.

10.3 Deployment as PROFINET IO controller

10.3.1 Steps of configuration



- With firmware version V2.4.0, there is a PROFINET IO controller available via the Ethernet PG/OP channel.
- As soon as you use the PROFINET functionality via the Ethernet PG/OP channel, this affects the performance and response time of your system and due to the system the cycle time of the OB 1 is extended by 2ms.



Range of functions

Please regard that the PROFINET IO controller supports only the PROFINET functions, which are described in this manual, even if the Siemens CPU, which is used for configuration, offers further functions! To use some described PROFINET functions, it is necessary to deploy another Siemens CPU for configuration. Here, however, is pointed to explicitly.

The configuration of the PROFINET IO controller for PROFINET communication should be done by the following procedure:

1. ➤ Commissioning and Initialization (assignment IP address data)
2. ➤ Hardware configuration - CPU
3. ➤ Configuration PROFINET IO controller
4. ➤ Configuration PROFINET IO devices



With the Siemens SIMATIC Manager, the CPU 013-CCF0R00 from VIPA is to be configured as
CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)!

10.3.2 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 your PROFINET IO controller with Ethernet.
4. ➤ Switch on the power supply.

⇒ After a short boot time, the CP is in idle.

At the first commissioning respectively after an overall reset of the CPU, the PROFINET IO controller and the Ethernet PG/OP channel have no IP address.

Assign IP address parameters

This function is supported only if the PROFINET IO controller is not yet configured. 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 5.5 & SP2 with the following proceeding:

1. ➤ Start the Siemens SIMATIC Manager.
2. ➤ Switch to "TCP/IP -> Network card " using 'Options → Set PG/PC interface → '.
3. ➤ Open the dialog for initialization of a station with 'PLC → Edit Ethernet node'.
4. ➤ 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 front of the CPU.
5. ➤ Choose if necessary the known MAC address of the list of found stations. To check this with [Blink] you may cause the MT LED to blink.
6. ➤ Either type in the IP configuration like IP address, subnet mask and gateway. Or your station is automatically provided with IP parameters by means of a DHCP server. Depending of the chosen option the DHCP server is to be supplied with MAC address, equipment name or client ID. The client ID is a numerical order of max. 63 characters. The following characters are allowed: Hyphen "-", 0-9, a-z, A-Z
7. ➤ Confirm with [Assign IP configuration].

Directly after the assignment the PROFINET IO controller is online reachable using the set IP address data. You can take the IP address data to your project by means of the hardware configuration. ↪ *Chap. 4.5 'Hardware configuration - CPU' page 69*

10.3.3 Configuration PROFINET IO controller

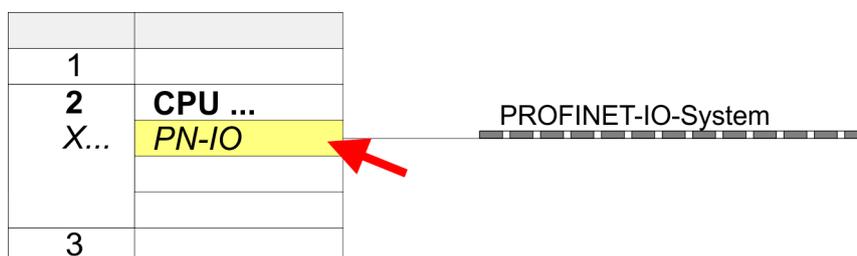
10.3.3.1 Precondition

To parametrize the PROFINET IO controller of the CPU, the following conditions must be fulfilled:

- The PROFINET IO controller is online reachable, this means an initialization was established.
- The hardware configuration described before was established and the PROFINET IO controller is networked.

Proceeding

➔ Open the properties dialog of the PROFINET IO controller by a double-click at PN-IO.



The PROFINET interface of the PROFINET IO controller is parameterized with PN-IO, the port with Port 1. In the following these parameters for PN-IO and Port 1 are described.

10.3.3.2 PN-IO

Tab: 'General'

Short description

Designation of the IO controller. For the IO controller of VIPA, the *short description* is "PN-IO".

Device name

The device name on the Ethernet subnet must be unique. For an integrated PROFINET interface the device name is derived from the short description. You can change this at any time.

Support device replacement without exchangeable medium

This parameter is not evaluated. With configured topology the *device replacement without exchangeable medium* is supported. ↪ Chap. 10.7 'Device replacement without exchangeable medium/PG' page 247

Properties

With *properties* you can enter the IP address, subnet mask and gateway for the PROFINET interface and select the subnet to be connected.

Tab: 'Addresses'

The CPU reports errors of the IO controller via the *interface address*, as soon as e.g. an error during synchronization of the IO controller occurs. With the *PROFINET IO system address* the CPU reports e.g. failure/return of the PROFINET IO system. This address is also used to identify the IO system to which the device belongs, if an IO device fails.

| | |
|---|--|
| Tab: 'PROFINET' | With the operation field "OB82 / I/O fault task..." you can cause the CPU to call the OB 82 at an error event of the PROFINET interface. An entry to the diagnostics buffer is always done. The other parameters here are not relevant for the use of the VIPA PROFINET CPU. |
| Tab: 'I-Device' | These settings are not required for the use of the PROFINET IO controller as an I-Device and should not be changed. ↪ <i>Chap. 10.4 'Deployment as PROFINET I-Device' page 238</i> |
| Tab: 'Synchronization' | These settings are not relevant and should not be changed. |
| Tab: 'Media Redundancy (MRP)' | MRP is supported exclusively as a <i>redundancy client</i> . ↪ <i>Chap. 10.5 'MRP' page 245</i> |
| Tab: 'Time-of-day synchronization' | In this area you can configure time-of-day master for time-of-day synchronization in the network. ↪ <i>Chap. 4.7.1.1.1 'Time-of-day synchronization' page 75</i> |
| Tab: 'Options' | |
| Interval | Here you can set the interval time with which "Keep-Alive" telegrams are to be sent to a connection partner. This ensures that a communication partner can still be reached because the connection resources are automatically released again after the expiration <i>interval</i> time. |
| 10.3.3.3 Port 1 | |
| Tab: 'General' | Shown is the short name "Port...". In the field Name another designation may be selected, which is also shown in the configuration table At <i>comment</i> you may describe your entry near more. The comment also appears in the configuration table. |
| Tab: 'Addresses' | Via the <i>port</i> address the diagnostics information of the IO controller may be accessed. |
| Tab: 'Topology' | These parameters serve for the port setting for topology ↪ <i>Chap. 10.6 'Topology' page 246</i> |
| Tab: 'Options' | These parameters serve for the port setting. Here the following parameters are supported: <ul style="list-style-type: none"> ■ Connection <ul style="list-style-type: none"> – Here you can make settings for transmission medium and type. Ensure that the setting for the local port and the partner port are identical. – PROFINET requires 100Mbit/s in duplex mode. ■ Boundaries <ul style="list-style-type: none"> – <i>Boundaries</i> are limitations for the transfer of certain Ethernet frames. The following <i>Boundaries</i> are supported: <ul style="list-style-type: none"> '<i>End of detection of accessible nodes</i>': DCP frames for detecting accessible nodes are not forwarded. When enabled, participants which are lying behind this port, are no longer recognized and can not be reached by the controller. '<i>End of topology discovery</i>': When enabled, this port does not support topology discovery, i.e. LLDP frames are not forwarded. |

10.3.4 Configuration PROFINET IO device

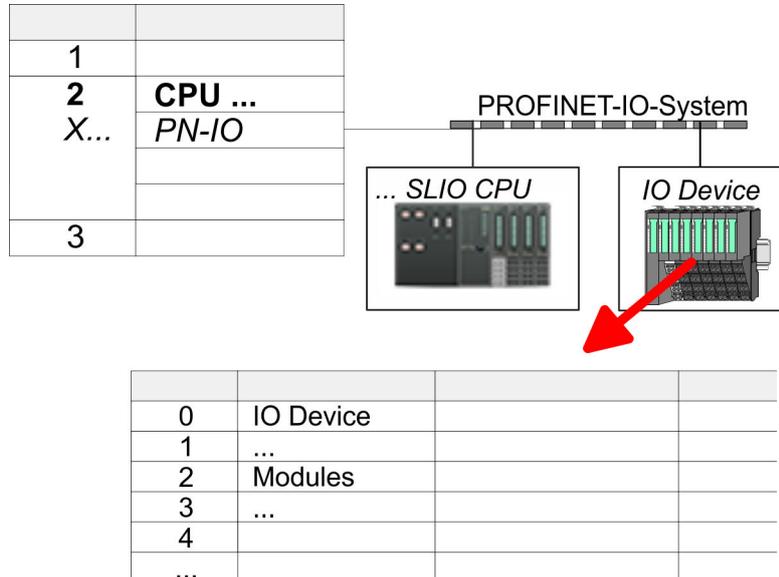
Precondition

- The modules, which may be configured here are listed in the hardware catalog.
- For the deployment of the PROFINET IO devices from VIPA you have to include the modules into the hardware catalog by means of the GSDML file from VIPA.
- After the installation of the GSDML file the PROFINET IO devices from VIPA may be found in the hardware catalog at 'PROFINET IO → Additional field devices → I/O → VIPA ...'

Configure IO devices

Now the project engineering of the PROFINET IO controller is finished. Please link up now your IO devices with periphery to your IO controller.

1. ➤ For the project engineering of PROFINET IO device you search the concerning PROFINET IO device in the hardware catalog at *PROFINET-IO* and drag&drop it in the subnet of your IO controller.
2. ➤ Assign a name to the IO device. The configured name must match the name of the device. Information about setting the device name can be found in the manual of the IO device.
3. ➤ Enter a valid IP address. The IP address is normally assigned automatically by the hardware configurator. If this is not desired, you can assign the IP address manually.
4. ➤ Link up the modules of your IO device in the plugged sequence and add the addresses that should be used by the modules.
5. ➤ If needed, parametrize the modules.
6. ➤ Save, compile and transfer your project. ↪ *Chap. 4.10 'Project transfer' page 90*



10.4 Deployment as PROFINET I-Device

10.4.1 Steps of configuration

Functionality



- With firmware version V2.4.0, there is a PROFINET IO controller available via the Ethernet PG/OP channel.
- As soon as you use the PROFINET functionality via the Ethernet PG/OP channel, this affects the performance and response time of your system and due to the system the cycle time of the OB 1 is extended by 2ms.



Range of functions

Please regard that the PROFINET IO controller supports only the PROFINET functions, which are described in this manual, even if the Siemens CPU, which is used for configuration, offers further functions! To use some described PROFINET functions, it is necessary to deploy another Siemens CPU for configuration. Here, however, is pointed to explicitly.

The I-Device (Intelligent IO device) functionality of a CPU allows data to be exchanged with an IO controller, which are preprocessed by the CPU. In this case, the I-Device is connected as an IO device to a higher-level IO controller. The process values recorded in central or decentralized periphery can be preprocessed via a user program and made available to the higher-level PROFINET IO controller by means of PROFINET.

- The configuration of the integrated PROFINET IO controller of the VIPA CPU as an I-Device is made via a virtual PROFINET devices, which is to be installed by means of a GSDML from VIPA in the hardware catalog.
- The communication takes place via input/output areas, which are defined in the I-Device.
- The size of the areas for input and output data is max. 768byte.
- The I-Device is made available to a deterministic PROFINET IO system via a PROFINET IO interface and thus supports the real-time communication *Real-Time*.
- The I-Device functionality meets the requirements of the RT class I (A) and corresponds to the PROFINET specification version V2.3.
- The configuration of a VIPA PROFINET CPU as an IO controller and at the same time as an I-Device is possible. The influence of the I-Device configuration on the system limits or performance of the PROFINET controller is equated with that of a device. This means that when the IO controller and I-Device are used at the same time on the PROFINET controller, the I-Device is to be regarded as an additional device for determining the system limits.
- In order for the higher-level IO controller to communicate with the VIPA I-Device, the following must be observed:
 - I-Device and higher-level IO controllers must be configured in different networks. Their IP addresses must be in the same IP circuit.
 - The device name of the PROFINET controller of the I-Device must match the device name of the I-Device at the higher-level IO controller.

Configuration

The configuration of the PROFINET IO controller as I-Device should be done by the following procedure:

1. ➤ Installation of the GSDML files
2. ➤ Configuration as I-Device
3. ➤ Configuration in the higher-level IO controller

10.4.2 Installing the GSDML file

The following GSDML files are required for configuring the integrated PROFINET IO controller of the VIPA CPU as I-Device in the Siemens SIMATIC Manager:

- GSDML for I-Device
- GSDML for I-Device at IO controller

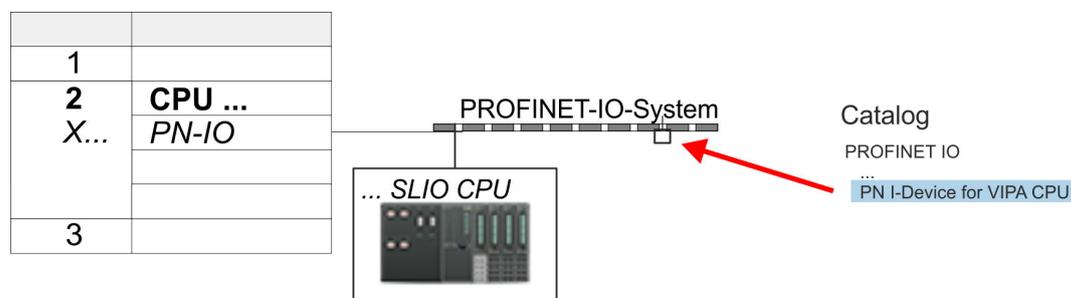
Proceeding

1. You can find the GSDML files in the download area of www.vipa.com. Load the file and unzip it on your PC.
2. Start the Siemens SIMATIC Manager and install via 'Options → Install new GSD file' both GSD files.
 - ⇒ After the installation you can find the following virtual devices in the *hardware catalog* at 'PROFINET IO → Additional field devices → ... → VIPA SLIO System':
 - PN I-Device for VIPA CPU
 - This allows you to configure the Input/output areas in the I-Device of the VIPA CPU.
 - PN I-Device for higher-level CPU
 - This allows you to connect the VIPA I-Device to the higher-level IO controller.

10.4.3 Configuration as I-Device

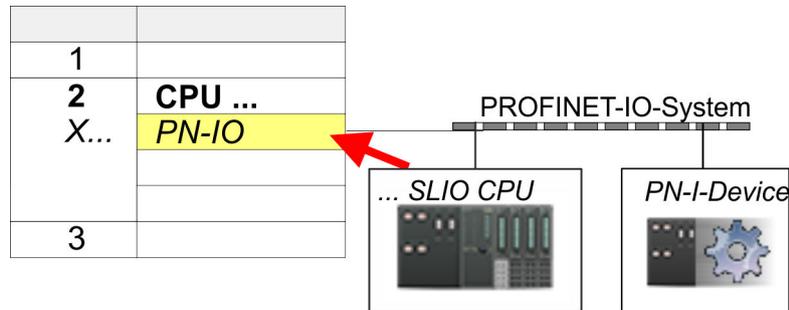
It is assumed that a hardware configuration of the CPU exists. ↪ *Chap. 4.5 'Hardware configuration - CPU' page 69*

1. For the project engineering of PROFINET I-Device you have to search the virtual device 'PN I-Device for VIPA CPU' in the hardware catalog at *PROFINET-IO* and drag&drop it in the PROFINET subnet.



- Open the properties dialog of the PROFINET IO controller of the CPU by a double-click at 'PN-IO' and assign the name for the I-Device.

i Write down the Name. This name must also be specified as the 'device name' of the I-Device for the higher-level IO controller.

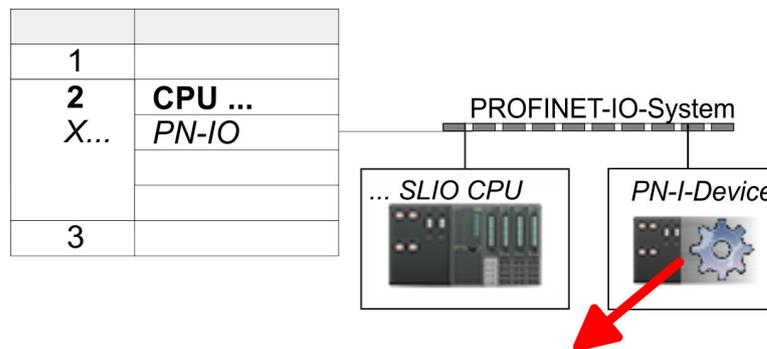


- For 'PN-IO' at 'slot' 'X...' assign an IP address via the properties dialog.
- Create the transfer areas by dragging them to the 'slots' as I/O areas from the hardware catalog. There must be no gaps in the slots. To create the transfer areas, the following input and output areas are available that can be assigned to the virtual I-Device:

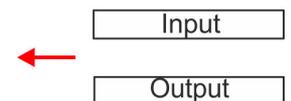
- Input: 1, 8, 16, 32, 64, 128, 256, 512 byte
- Output: 1, 8, 16, 32, 64, 128, 256, 512 byte

The data direction for *Input* or *Output* refers to the view of the I-Device.

- Input* areas define data that are sent from the higher-level IO controller to the I-Device and which are mapped to the input address area of the CPU.
- Output* areas define data that are sent to the higher-level IO controller and which are to be stored in the output address area of the CPU.



| | | | |
|-----|--------------|-------------------|--|
| 0 | PN-I-Device | SLIO PLC PN-I-D.. | |
| X2 | I-Device ... | | |
| 1 | | | |
| 2 | | | |
| 3 | | | |
| ... | | | |

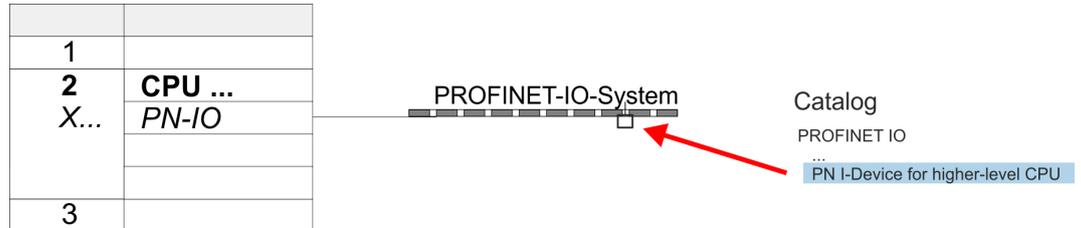


- Save and transfer your project to the CPU.

10.4.4 Configuration in the higher-level IO controller

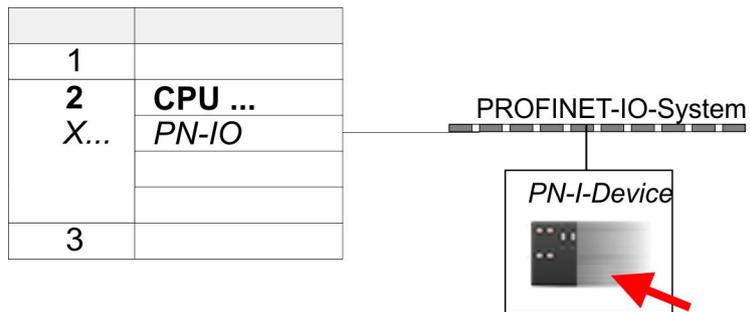
It is assumed that a CPU is configured with IP address with the higher-level IO controller. The IP address must be in the same IP circuit as the IP address of the I-Device.

1. ➤ Open the project of the CPU with the higher-level IO controller.
2. ➤ For the project engineering of VIPA I-Device in the high-level IO controller you have to search the device 'PN I-Device for high-level CPU' in the hardware catalog at PROFINET-IO and drag&drop it in the PROFINET subnet.



3. ➤ Open the properties dialog by double-clicking 'PN-I-Device' and enter at 'device name' the previously noted name of the VIPA I-Device.

i The configured name must match the name of the PROFINET IO controller 'PN-IO' of the I-Device CPU, which you have noted before! ↪ Chap. 10.4.3 'Configuration as I-Device' page 239



4. ➤ Configure an input area of the same size for each output area of the I-Device in the IO controller and vice versa. Here also no gaps may arise. In particular, make sure that the order of the transfer areas matches that of the I-Device configuration. The following transfer units are available:
 - Input: 1, 8, 16, 32, 64, 128, 256, 512 byte per slot
 - Output: 1, 8, 16, 32, 64, 128, 256, 512 byte per slot
5. ➤ Save and transfer your project to the CPU.
 - ⇒ Your VIPA PROFINET controller is now connected as an I-Device to the higher-level PROFINET IO controller.

i **I-Device with S7 routing**
 S7 routing is not possible with the procedure above. S7 routing is only possible if the I-Device and the higher-level I/O controller are configured in the same network. The device names must not be identical. By using identical names and extending the name of the I-Device with "-x", this is detected internally and used appropriately for S7 routing.

10.4.5 Error behavior and interrupts

Error behavior

The system shows the following error behavior ...

- ... at gaps in the 'slot' configuration:
 - If the configuration of the I-Device contains gaps in the 'slot' configuration (i.e. there are free 'slots' before used 'slots'), the configuration is rejected and 0xEA64 is returned as a configuration error in the diagnostic buffer.
 - If the configuration of the higher-level IO controller contains gaps in the 'slot' configuration (i.e. there are free 'slots' before used 'slots'), the connection is rejected with the PN IO Status *ErrorCode1* = 0x40 and *ErrorCode2* = 0x04 (AR_OUT_OF_RESOURCE).
- ... at modules, which differ from the configured:
 - A *ModuleDiffBlock* is generated and the wrong modules are not served.
- ... if the number of configured modules in the IO controller is greater than the number of configured modules in the I-Device:
 - The IO controller receives a *ModuleDiffBlock* with *ModuleStatus* "NoModule" for modules that are not configured in the I-Device. The I-Device sets the status of the non-configured modules to "bad".
- ... if the number of configured modules in the I-Device is greater than the number of configured modules in the IO controller:
 - The IO controller does not receive an error because the additional modules are unknown.

| Starting position | IO controller in RUN, I-Device in RUN |
|-------------------|---|
| Event | I-Device CPU goes to STOP |
| Reaction | <ul style="list-style-type: none"> ■ An OB 85 is called in the IO controller for each input and output transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. ↪ 79 ■ An OB 122 is triggered in the IO controller for each peripheral direct access to an input or output transfer area. |

| Starting position | IO controller in RUN, I-Device in RUN |
|-------------------|---|
| Event | IO controller goes to STOP |
| Reaction | <ul style="list-style-type: none"> ■ An OB 85 is called in the I-Device for each input transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. ↪ 79 ■ In the I-Device, an OB 122 is triggered for each peripheral direct access to an input transfer area. |
| | Note: Output transfer areas can still be accessed! |

| Starting position | IO controller in RUN, I-Device in RUN |
|-------------------|---|
| Event | Station failure I-Device, e.g. by bus interruption |
| Condition | I-Device must remain operational without a bus connection, i.e. the power supply must further exist. |
| Reaction | <ul style="list-style-type: none"> ■ An OB 86 (station failure) is called up in the IO controller. ■ An OB 85 is called in the IO controller for each input and output transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. ↪ 79 ■ An OB 122 is triggered in the IO controller for each peripheral direct access to an input or output transfer area. ■ An OB 86 (station failure) is called up in the I-device. ■ An OB 85 is called in the IO controller for each input and output transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. ↪ 79 ■ In the I-device, an OB 122 is triggered for each peripheral direct access to an input or output transfer area. |
| Starting position | IO controller in RUN, I-Device in RUN |
| Event | Station recovery |
| Reaction | <ul style="list-style-type: none"> ■ An OB 86 (recovery) is called in the IO controller. ■ An OB 85 is called in the IO controller until the OB 86 has been called, for each input and output transfer area, which is in the process diagram, if messages of process image transfer errors are parametrized. ↪ 79 ■ An OB 122 is triggered in the IO controller until the OB 86 is called, for each peripheral direct access to an input or output transfer area. ■ An OB 86 (return) is called in the I-Device. ■ An OB 83 (sub module recovery) is called for each input transfer area in the I-Device. ■ In the I-device, an OB 85 is called for each input transfer area, which is in the process image, if messages of process image transfer errors are parametrized and the corresponding OB 83 has not yet been called. ↪ 79 ■ An OB 122 is triggered in the I-Device for each peripheral direct access to an input transfer area, until the corresponding OB 83 has been called. |
| Starting position | Controller in RUN, I-Device in STOP |
| Event | I-Device starts |
| Reaction | <ul style="list-style-type: none"> ■ The OB 100 (start-up) is called in the I-Device. ■ The OB 83 (Return-of-Submodule) for input sub modules of the transfer areas to the higher-level IO controller is called in the I-Device. ■ An OB 85 is called in the I-device for each input transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. ↪ 79 ■ In the I-Device, an OB 122 is triggered for each peripheral direct access to an input transfer area. ■ OB 83 (Return-of-Submodule) for input and output sub modules of the transfer areas to the I-Device is called in the IO controller. ■ An OB 85 is called in the IO controller for each input and output transfer area, which is located in the process diagram, if messages of process image transfer errors are parametrized and the corresponding OB 83 has not yet been called. ↪ 79 ■ In the IO controller, an OB 122 is triggered for each peripheral direct access to an input or output transfer area until the corresponding OB 83 has been called. |

Deployment as PROFINET I-Device > Error behavior and interrupts

| Starting position | IO controller is in STOP, I-Device in RUN |
|-------------------|--|
| Event | IO controller starts |
| Reaction | <ul style="list-style-type: none">■ The OB 83 (Return-of-Submodule) for input sub modules of the transfer areas to the higher-level IO controller is called in the I-Device.■ An OB 85 is called for each transfer area, which is located in the process image, in the I-device if messages of process image transfer errors are parametrized and the corresponding OB 83 has not yet been called. ↪ 79■ An OB 122 is triggered in the I-Device for each peripheral direct access to an input transfer area, until the corresponding OB 83 has been called.■ The OB 100 (startup) is called in the IO controller. |

10.5 MRP

Overview

To increase the network availability of an industrial Ethernet network, you can connect a *line topology* together to a *ring topology*. To set up a ring topology with media redundancy, you have to bring together the two free ends of a linear bus topology in one device. Closing the linear bus topology to form a ring is achieved with two ports (ring ports) of a device in the ring. At least one device of the ring takes the role of the *redundancy manager*. All other devices in the ring are *redundancy clients*. A standard media redundancy method is MRP (Media Redundancy Protocol). Up to 50 devices per ring can participate. The MRP (**M**edia **R**edundancy **P**rotocol) is specified in the standard IEC 61158 Type 10 "PROFINET".

Precondition

- The ring in which you want to use MRP may consist only of devices that support this function.
- "MRP" must be activated for all devices in the ring.
- All devices must be connected via their ring ports.
- The ring may contain max. 50 devices.
- The connection setting (transmission medium/duplex) must be set to "full duplex" and at least 100Mbit/s for all ring ports. Otherwise there may be a loss of data traffic.

Function

- The data paths between the individual devices are automatically reconfigured if the ring is interrupted at any point. After reconfiguration, the devices are accessible again.
- In the redundancy manager, one of the both ring ports are blocked for uninterrupted network operation for normal communication so that no data telegrams are circulated. In terms of data transmission, the ring topology is a linear bus topology.
- The *redundancy manager* monitors the ring for interruptions. For this he sends test frames from both ring port 1 and ring port 2. The test frames run through the ring in both directions until they arrive at the other ring port of the redundancy manager.
- As soon as the interruption is removed, the original transmission paths are restored, the two ring ports of the redundancy manager are disconnected and the redundancy clients informed of the change. The redundancy clients then use the new paths to the other devices.

Reconfiguration time

The time between the ring interruption and restoration of a functional linear topology is known as the *reconfiguration time*. At MRP the *reconfiguration time* is typically 200ms.

VIPA PROFINET CP as redundancy client

MRP exclusively supported as *redundancy client*. If the ring is opened or closed, you will be informed via the OB 82 "Neighbourhood change". With SFB 54 you can get more information.



The use of MRP in the operating mode I-Device is not permissible and is rejected during the configuration!

10.6 Topology

Overview

By configuring the topology you specify for the PROFINET IO controller the physical connections between the stations in your PROFINET IO system. These "neighbourhood relations" are used among others at "Device replacement without exchangeable medium". Here by comparison of target and current topology, the IO device without a name is detected and automatically integrated to the user data traffic. By configuring the topology you have the following options:

- You can evaluate topological errors in your application program
- You have greater flexibility in planning and expansion of a plant



Support Topology editor is limited

Please consider that the support for the topology editor of the Siemens SIMATIC Manager is limited. Here you have only the possibility to configure the target topology offline. An online matching is currently not possible. An interconnection of the ports is also possible by means of the port properties!

Interconnection by means of the *Port* properties

1. ➤ Click in the hardware configurator at the according PROFINET port and open the properties dialog via 'Context menu → Object properties' and select the register 'Topology'.
 - ⇒ The properties dialog to interconnect the ports is opened.
2. ➤ Here you have the following parameters:
 - Port interconnection
 - Local port: Name of the local port
 - Medium: Specifying the line type (copper, fibre optic cable). Currently, this parameter is not evaluated.
 - Cable name Specifying a cable name
 - Partners
 - Partner port: Name of the port to which the selected port is interconnected.
 - Alternating partner ports: By specifying at 'Partner port' "Any partner", you can configure alternating partner ports for the I/O devices. Currently, this parameter is not evaluated.
 - Cable data
 - Cable length: Depending on the port medium you can set in the select list the cable length, if the medium between two stations does not change. Here the signal delay time is automatically calculated. Currently, this parameter is not evaluated.
 - Signal delay time: If the medium between two stations changes, a signal delay time can be defined here. Currently, this parameter is not evaluated.
3. ➤ Close the properties dialog with [OK] again.

10.7 Device replacement without exchangeable medium/PG

Overview

IO devices, which support the PROFINET function *Device replacement without exchangeable medium/PG* get their device name from the controller with the exchange. These can be replaced without installing an "exchangeable medium" (memory card) with the stored device name respectively without assigning a device name by a PG. To assign the device name the IO controller uses the configured *Topology* and the "neighbourhood relationship", which is determined by the IO devices.

Thus the *Device replacement without exchangeable medium/PG* is possible, the following requirements must be met:

- The *Topology* of your PROFINET IO system with the corresponding IO devices must be configured.
- The IO controller and the respective adjacent to the unit to be replaced IO device must support the functionality *Device replacement without exchangeable medium/PG*.
- In the IO controller in the '*Properties*' the option *Support device replacement without exchangeable medium* must be enabled.
- The replaced device must be reset to delivery state, before.

Configuring the function

The configuration of the function *Device replacement without exchangeable medium/PG* in your PROFINET IO system happens with the following approach:

1. ➤ Double-click at the PROFINET interface of the IO controller of the CPU.
⇒ The properties dialog of this PROFINET interface is opened
2. ➤ Enable in the register '*General*' the option '*Support device replacement without exchangeable medium*'.
3. ➤ Apply the settings with [OK].
4. ➤ Save and translate the hardware configuration.
5. ➤ Configure your *Topology*. ↪ *Chap. 10.6 'Topology' page 246*
6. ➤ Transfer your project to the CPU.

Prepare the replace device

For the replacement the "replace device" must be in "delivery state". If you have not received a new "replace device" from VIPA, you have to prepare this with the following approach:

1. ➤ For this connect your "replace device" local at your PG.
2. ➤ Start the Siemens SIMATIC Manager and execute '*PLC* ➔ *Edit Ethernet node*'
3. ➤ Click at '*Nodes accessible online*' at [Browse].
4. ➤ Select the according IO device, which you identify as your "replace device".
5. ➤ Click at '*Reset to factory settings*' at [Reset].
⇒ Your IO device is now reset and has then "delivery state".

Replace device

For the replacement the "replace device" must be in "delivery state".

1. ➤ Disconnect if not already done your device to be exchanged from power.
2. ➤ Replace this by your "replace device".
3. ➤ Connect the "replaced device" to power and turn it ON.
⇒ Here by comparison of target and current topology, the "replaced device" is automatically detected by the IO controller and automatically integrated to the user data traffic.

10.8 Commissioning and start-up behavior

- Start-up on delivery state** In the delivery state the CPU is overall reset. After power ON the PROFINET part has no configuration the PROFINET has no configuration and the LEDs of the PROFINET part are off. The PROFINET part is passive and can be found by the device search.
- Online with bus parameters without project**
- For the communication between IO controller and IO device the ways for the communication are to be defined before. For the clear specification of the communication ways, these are established during the start-up by the IO controller, based on the project data. Here the configuration takes place by a hardware configuration.
 - As soon as the project data were transmitted, the IO controller performs a new system start-up.
 - In this state the IO controller may be accessed and its CPU may be configured via Ethernet by the IO controller by means of the IP address.
- IO device configuration**
- The PROFINET IO controller is configured by a hardware configuration. After the transmission of the project into the IO controller, the IO controller has the whole information for the addressing of and the data exchange with the IO devices.
 - During the system start-up of the IO controller the IO devices are supplied with their configured IP address by means of the DCP protocol. After PowerON and after transmitting a new hardware configuration, due to the project data the system start-up of the IO controller is initialized and it runs automatically. During the system start-up the IO controller establishes a clear communication relation (CR) and an application relation (AR) to an IO device. Here the cyclic IO data, the acyclic R/W services and the expected modules/sub modules are specified.
 - The PROFINET IO controller does not have any physical LEDs to show the status. The status information are stored as virtual LED states. During runtime, you can determine their status using the SSL partial lists xy19h or xy74h. ↪ *Chap. 10.9.5 'Diagnostics status indication via SSLs' page 251*
 - The BF2 LED is on when a PROFINET IO device is configured as "linked" but o bus cable is connected.
 - If the IO controller has received a valid configuration with at least one IO device, the BS2-LED gets on.
 - With Ethernet interface parameters, which are unsuitable for PROFINET operation, the BS2-LED flashes at 1Hz.
 - If the IP address of the IO controller can not be used because e.g. it is duplicated, the BS2-LED flashes at 0.5Hz.
 - If at least one IO device is not in cyclic data exchange after the start-up, the BF2 LED blinks.
 - If all IO devices are in cyclic data exchange, the BF2 LED gets off. After a successful system start-up the system is ready for communication.
- CPU state influences the IO process data**
- After PowerON respectively a receipt of a new hardware configuration the configuration data are automatically transferred to the IO controller. Dependent on the CPU state the following behavior is shown by the IO controller:
- Behavior at CPU STOP
 - In the STOP state of the CPU an output telegram is further cyclically sent, but the contained data are marked as "invalid" and as output data zeros are transmitted.
 - The IO controller further receives the input data of the IO devices and transfers them cyclically to the input area of the CPU.
 - Behavior at CPU RUN
 - The IO controller cyclically reads the output data from the CPU and transfers these as telegram to the configured IO devices.
 - The IO controller receives the input data of the IO devices and transfers them cyclically to the input area of the CPU.

10.9 PROFINET diagnostics

10.9.1 Overview

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

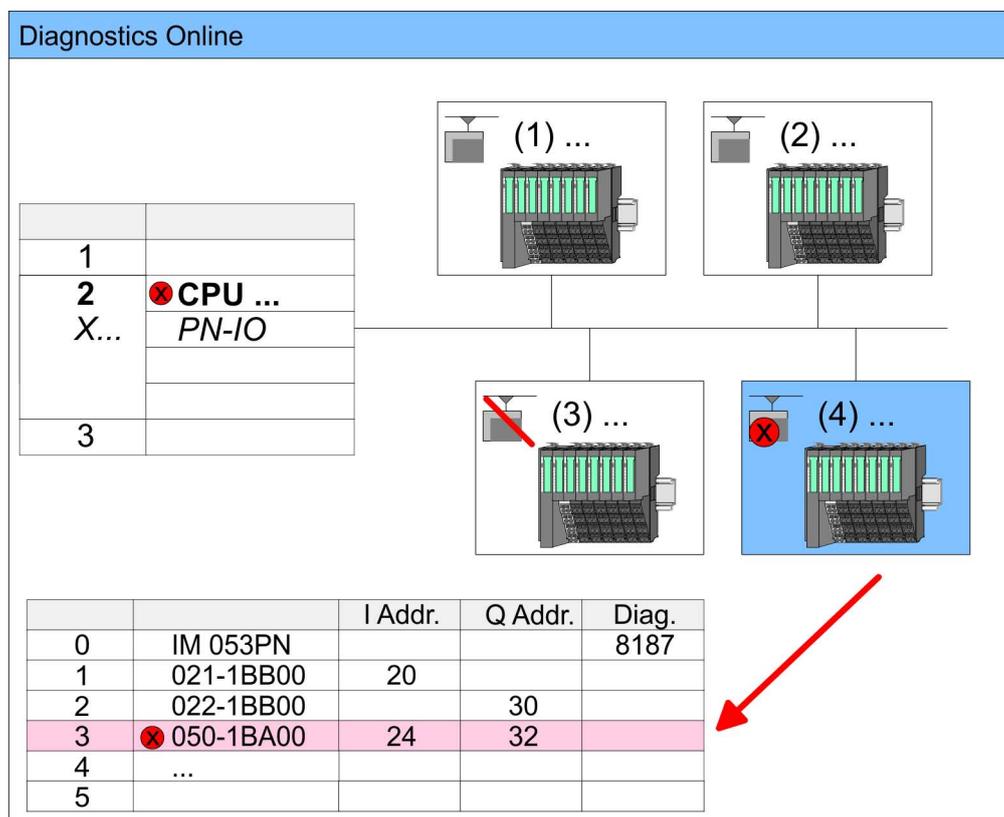
- Diagnostics with the configuration and engineering tool
- Diagnostics during runtime in the user program (OB 1, SFB 52)
- Diagnostics via OB start information
- Diagnostics status indication via SSLs

10.9.2 Diagnostics with the configuration and engineering tool

If you are connected from your configuration respectively engineering tool via Ethernet with the PROFINET IO controller, online diagnostics information may be accessed.

E.g. with 'Station → Open online' you get information about the state of your system. Here missing respectively faulty components are shown by symbols.

In the following figure e.g. there is shown that the configured device 3 is missing and device 4 reports an error.



10.9.3 Diagnostics during runtime in the user program

With SFB 52 RDREC (read record) you can access diagnostics data from your user program e.g. in OB 1. The SFB 52 RDREC operates asynchronously, that is, processing covers multiple SFB calls.



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

Example OB 1

For the cyclic access to the diagnostics data of the System SLIO counter module 050-1BA00 the following example may be used in the OB 1:

```

AN M10.3 'If the reading terminated (BUSY=0) and
AN M10.1 'there is no job triggered (REQ=0) then
S M10.1 'start transfer of record (REQ:=1)
L W#16#4000 'Number of record set (0x4000)
T MW12
CALL SFB 52, DB52 'Call SFB 52 with Instance DB
  REQ :=M10.1      'Trigger flag
  ID :=DW#16#0018 'Smaller addr. of mixed module
  INDEX :=MW12
  MLEN :=14       'Length record set 0x4000
                   'with 1 entry
  VALID :=M10.2   'Validity of the record set
  BUSY :=M10.3    'Flag 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 REQ

```

Diagnostics data

The counter module 050-1BA00 serves for 20byte diagnostics data. The diagnostics data of the System SLIO module 050-1BA00 have the following structure:

| Name: | Bytes | Function | Default |
|-----------------|-------|-------------------------------------|---------|
| ERR_A | 1 | Diagnostics | 00h |
| MODTYP | 1 | Module information | 18h |
| ERR_C | 1 | reserved | 00h |
| ERR_D | 1 | Diagnostics | 00h |
| CHTYP | 1 | Channel type | 76h |
| NUMBIT | 1 | Number diagnostics bits per channel | 08h |
| NUMCH | 1 | Number channels of the module | 01h |
| CHERR | 1 | Channel error | 00h |
| CH0ERR | 1 | Channel-specific error | 00h |
| CH1ERR...CH7ERR | 7 | reserved | 00h |
| DIAG_US | 4 | µs ticker | 00h |



More information about the diagnostics data may be found in the System SLIO manual HB300_FM_050-1BA00.

10.9.4 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 IO device (Diagnostics interrupt)
- OB 83 on inserting respectively removing a module on a IO device
- OB 86 on failure respectively return of a IO device



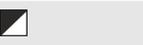
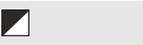
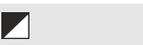
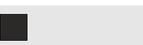
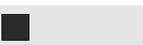
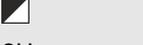
More information about the OBs and their start information may be found in the online help of your programming tool and in the manual "SPEED7 Operation List" from VIPA.

10.9.5 Diagnostics status indication via SSLs

The PROFINET IO controller does not have any physical LEDs to show the status. The status information are stored as virtual LED states. During runtime, you can determine their status using the SSL partial lists xy19h or xy74h. More can be found in the manual operation list (HB00_OPL_SP7) of your CPU.

Virtual LEDs PROFINET

| BF3 (bus error) | BS2 (Bus status) | MT2 (Maintenance) | Meaning |
|--|-------------------------------------|--------------------------|--|
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | PROFINET is not configured. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | PROFINET is configured with valid Ethernet interface parameter, valid IP address and at least one IO device. |
| <input checked="" type="checkbox"/> | X | X | <ul style="list-style-type: none"> ■ Bus error, no connection to sub net/switch. ■ Wrong transfer rate ■ Full-duplex-transmission is not activated. |
| <input checked="" type="checkbox"/> 2Hz | X | X | <ul style="list-style-type: none"> ■ Failure of a connected IO device. ■ At least one IO device is not access-able. ■ Faulty configuration ■ I device is configured, but no connection exists yet. |

| BF3 (bus error) | BS2 (Bus status) | MT2 (Maintenance) | Meaning |
|---|--|--|---|
| X |  1Hz | X | <ul style="list-style-type: none"> Ethernet interface parameter are not valid. I-Device is configured and <i>Link mode</i> does not correspond to '100 Mbps full duplex'. |
| X |  0.5Hz | X | There was no IP address assigned. |
| X | X |  | Maintenance event of an IO device is pending respectively an internal error happened. |
|  4s on, 1s off | X |  4s on, 1s off | Simultaneous blinking indicates a not valid configuration. |
|  4Hz |  |  4Hz | The alternate blinking indicates that a firmware update of the PROFINET IO controller is executed. |
|  |  |  | Firmware update of the PROFINET IO controller is finished without error. |
| X | X |  2Hz | With a suited configuration tool you can cause the MT LED to blink by means of the function ' <i>Member blink test</i> '. This can be useful for e.g. identification of the module. |

on:  | off:  | blinking:  | not relevant: X

Deployment BS LED - Bus status

- BS LED: off
 - PROFINET is not configured.
- BS LED: blinks with 1Hz
 - Ethernet interface parameter are not valid.
- BS LED: blinks with 0.5Hz
 - There was no IP address assigned.
- BS LED: on
 - PROFINET is configured with valid Ethernet interface parameter, valid IP address and at least one IO device.

Deployment of the MT LED - Maintenance

- MT LED: off
 - There is no maintenance event pending.
- MT LED: on
 - Maintenance event of an IO device is pending respectively an internal error happened.
 - Here in the diagnostic buffer of the CPU, an entry was created, where you can find more information about the maintenance event and to resolve it. ↪ *Chap. 4.20 'Diagnostic entries' page 117*
 - Resolve the error and execute PowerOFF/ON.
 - Currently you need to perform a power cycle, to switch the MT-LED off again.
- MT LED: blinks
 - With a suited configuration tool you can cause the LED to blink by means of the function '*Member blink test*'. This can be useful for e.g. identification of the module.
 - Simultaneous blinking together with BF2 LED (4s on, 1s off) indicates that the configuration is invalid.
 - The alternate blinking with BF2 LED with 4Hz indicates that a firmware update of the PROFINET IO controller is executed.

10.10 PROFINET system limits

Maximum number devices and configurable connections

$$D = \sum_{i=1}^n \frac{I}{A_i}$$

Based on the devices, which have to communicate with the IO controller per ms, you can determine the maximum number of devices. This also results in the maximum number of configurable connections. The *Devices per ms* can be determined by the sum formula of the individual refresh times (A).

- D Devices per ms
- n Number of devices
- A Refresh time device



Please note that the value D must always be rounded to the nearest smaller integer!

The PROFINET IO controller has the following system limits

| Devices per ms (D) | Max. number of devices | Max. number of configurable connections |
|--------------------|------------------------|---|
| 8 | 32 | 0 |
| 7 | 32 | 2 |
| 6 | 64 | 2 |
| 5 | 96 | 2 |
| 4 | 128 | 2 |
| 3 | 128 | 2 |
| 2 | 128 | 2 |
| 1 | 128 | 2 |
| 0 | 0 | 2 |

Output bytes per ms

$$O = \sum_{i=1}^n \frac{B_i}{A_i}$$

- O Output bytes per ms
- n Number of devices
- B Number output bytes per device
- A Refresh time per device

The PROFINET IO controller has the following system limits:

- Max. Number output bytes per ms: 800
- Max. Number output bytes per device: 768

Input bytes per ms

$$I = \sum_{i=1}^n \frac{C_i}{A_i}$$

- I Input bytes per ms
- n Number of devices
- C Number input bytes per device
- A Refresh time per device

The PROFINET IO controller has the following system limits:

- Max. number input bytes per ms: 800
- Max. number input bytes per device: 768

11 Option: Deployment PROFIBUS communication

11.1 Fast introduction

Overview

The PROFIBUS DP slave 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.



To switch the interface X3 MPI(PtP) 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. ↪ Chap. 4.16 'Deployment storage media - VSD, VSC' page 111

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 013-CCF0R00 from VIPA is to be configured as

CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)

The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X1 (MPI/DP).

11.2 PROFIBUS communication



To switch the interface X3 MPI(PtP) 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. ↪ Chap. 4.16 'Deployment storage media - VSD, VSC' page 111

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.

11.3 Enable bus functionality via VSC

Enabling

🔗 *'Overview' page 111*

11.4 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 314C-2 PN/DP (314-6EH04-0AB0 V3.3).

| 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).

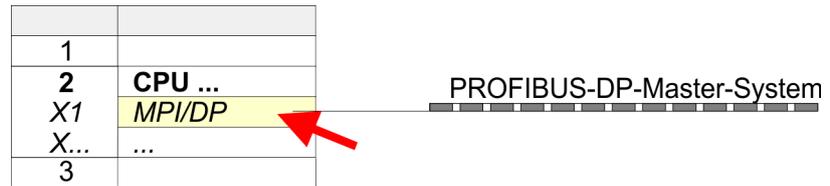
11.5 Deployment as PROFIBUS DP master

Precondition

The hardware configuration described before was established.

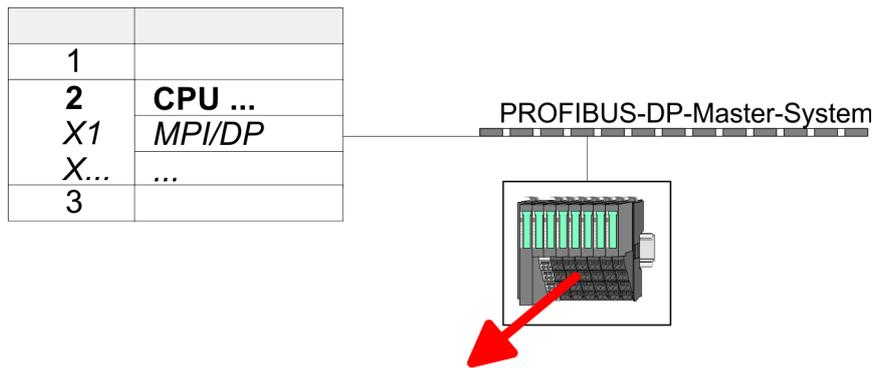
Proceeding

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.



| | | | |
|-----|--------|--|--|
| 1 | ... | | |
| 2 | Module | | |
| 3 | ... | | |
| 4 | | | |
| 5 | | | |
| ... | | | |

11.6 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

| Slot | Module |
|------|---------|
| 1 | |
| 2 | CPU ... |
| X1 | MPI/DP |
| X... | ... |
| 3 | |
| 4 | ... |
| 5 | Modules |
| 6 | ... |

Object properties

Operating mode: DP slave
 Connect: PROFIBUS
 PROFIBUS address: > 1

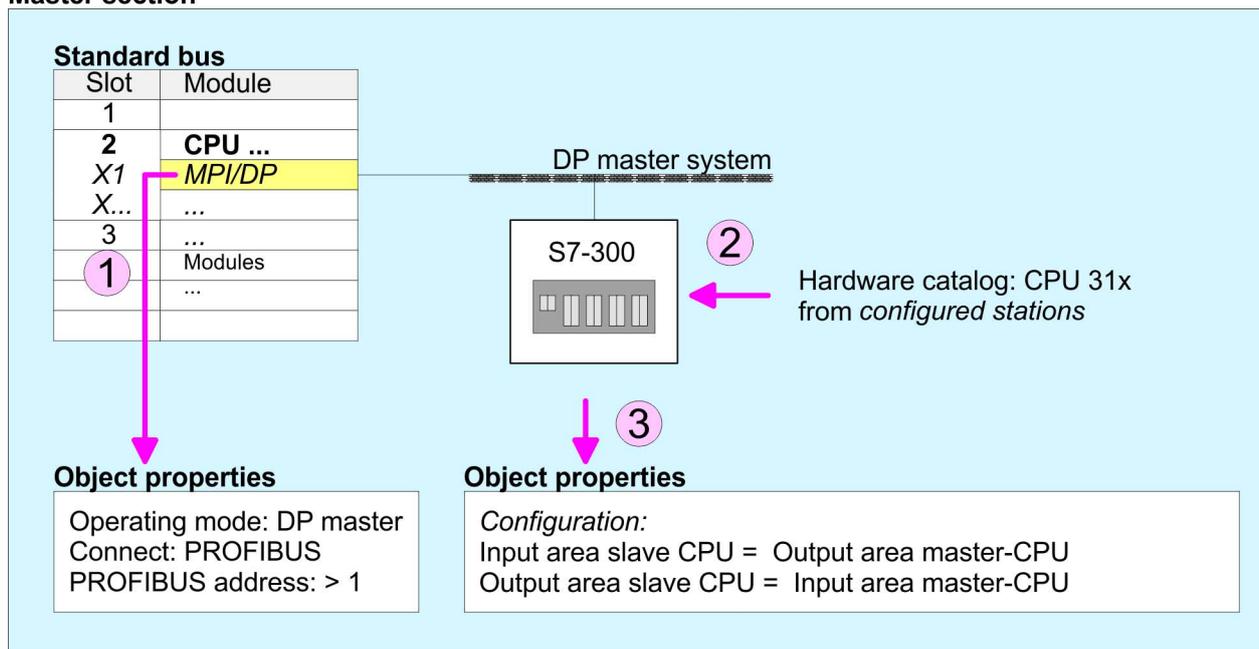
Configuration:
 Input area
 Output area

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



11.6.1 Diagnostic functions

Overview

PROFIBUS DP provides an extensive set of diagnostic functions for quick error localization. Diagnostic messages are transferred via the bus and collected by the master. In the operating mode DP slave the CPU sends diagnostic data when requested by the master or in case of an error. Since a part of the diagnostic data (Byte 11 ... 15) is located in the peripheral address area of the CPU, you may start the diagnostics and modify the diagnostic data. Diagnostic data consist of:

- Standard diagnostic data (Byte 0 ... 5),
- Device specific diagnostic data (Byte 6 ... 15).

Structure

The diagnostic data have the following structure:

Standard diagnostic data

| | |
|--------|--------------------|
| Byte 0 | Station status 1 |
| Byte 1 | Station status 2 |
| Byte 2 | Station status 3 |
| Byte 3 | Master address |
| Byte 4 | Ident number (low) |
| Byte 5 | Ident number High |

Device specific diagnostic data

| | |
|----------------------------|--|
| Byte 6 | Length and code device specific diagnostic |
| Byte 7 | Device specific diagnostic messages |
| Byte 8... | reserved |
| Byte 10 | |
| Byte 11 ... Byte 15 | User-specific diagnostic data is mapped into the peripheral addressing range of the CPU and may be modified and sent to the master. |

Standard diagnostic data

More detailed information to the structure of the slave standard diagnostic data can be found in the standard papers of the PROFIBUS User Organization. The slave diagnostic data have the following structure:

| Byte | Bit 7 ... Bit 0 |
|------|--|
| 0 | <ul style="list-style-type: none"> ■ Bit 0: 0 (fix) ■ Bit 1: Slave is not yet ready for data exchange ■ Bit 2: Configuration data are not identical ■ Bit 3: Slave has external diagnostic data ■ Bit 4: Slave does not provide this function ■ Bit 5: 0 (fix) ■ Bit 6: Wrong parametrization ■ Bit 7: 0 (fix) |
| 1 | <ul style="list-style-type: none"> ■ Bit 0: Slave needs new parametrization ■ Bit 1: Static diagnostic ■ Bit 2: 1 (fix) ■ Bit 3: Response monitoring active ■ Bit 4: Freeze command received ■ Bit 5: Sync command received ■ Bit 6: reserved ■ Bit 7: 0 (fix) |
| 2 | <ul style="list-style-type: none"> ■ Bit 0 ... Bit 6: reserved ■ Bit 7: Diagnostic data overflow |
| 3 | Master address after parametrization |
| 4 | ID number high byte |
| 5 | ID number low byte |

Device specific diagnostic data

The device related diagnostic data provide detailed information on the slave and the peripheral modules. The length of the device related diagnostic data is fixed at 10byte

| Byte | Bit 7 ... Bit 0 |
|-----------------|---|
| 6 | <ul style="list-style-type: none"> ■ Bit 0 ... 5: Length device specific diagnostic data <ul style="list-style-type: none"> – 001010: Length 10byte (fix) ■ Bit 6 ... 7: Code for device specific diagnostic <ul style="list-style-type: none"> – 00: Code 00 (fix) |
| 7 | <ul style="list-style-type: none"> ■ Device specific diagnostic message <ul style="list-style-type: none"> – 12h: Error: Data length parameters – 13h: Error: Data length configuration data – 14h: Error: Configuration entry – 15h: Error: VPC3 buffer calculation – 16h: Missing configuration data – 17h: Error: Comparison DP parametrization and configuration – 40h: User specific diagnostic data is valid |
| 8 ...10 | reserved |
| 11 ...15 | <p>User specific diagnostic data that are stored behind the diagnostic status byte in the process picture of the CPU.</p> <p>This data may be overwritten and forwarded to the master.</p> |

Release diagnostic

- In case of a diagnostic the contents of Byte 11 ... 15 of the device specific diagnostic data will be transferred to the process image of the CPU and this preceded by a status byte.
- Where this diagnostic block with a length of 6byte is located in the process image can be defined via the CPU parameters.
- You start diagnostics by means of a status change from 0 → 1 in the diagnostic status byte.
- This transmits the respective diagnostic message to the master.
- **A status of 0000 0011 is ignored!**

The diagnostic block of the CPU has the following structure:

| Byte | Bit 7 ... Bit 0 |
|---------|---|
| 0 | <p>Diagnostic status byte</p> <ul style="list-style-type: none"> ■ Bit 0: User specific diagnostic data <ul style="list-style-type: none"> – 0: Invalid diagnostic data – 1: Valid diagnostic data (starting a diagnostic) ■ Bit 1: Delete diagnostic <ul style="list-style-type: none"> – 0: Diagnostic deletion not valid – 1: Diagnostic deletion valid ■ Bit 2 ... Bit 7: reserved |
| 1 ... 5 | User specific diagnostic data equal to Byte 11 ... 15 of device specific diagnostic |

11.7 PROFIBUS installation guidelines

PROFIBUS in general

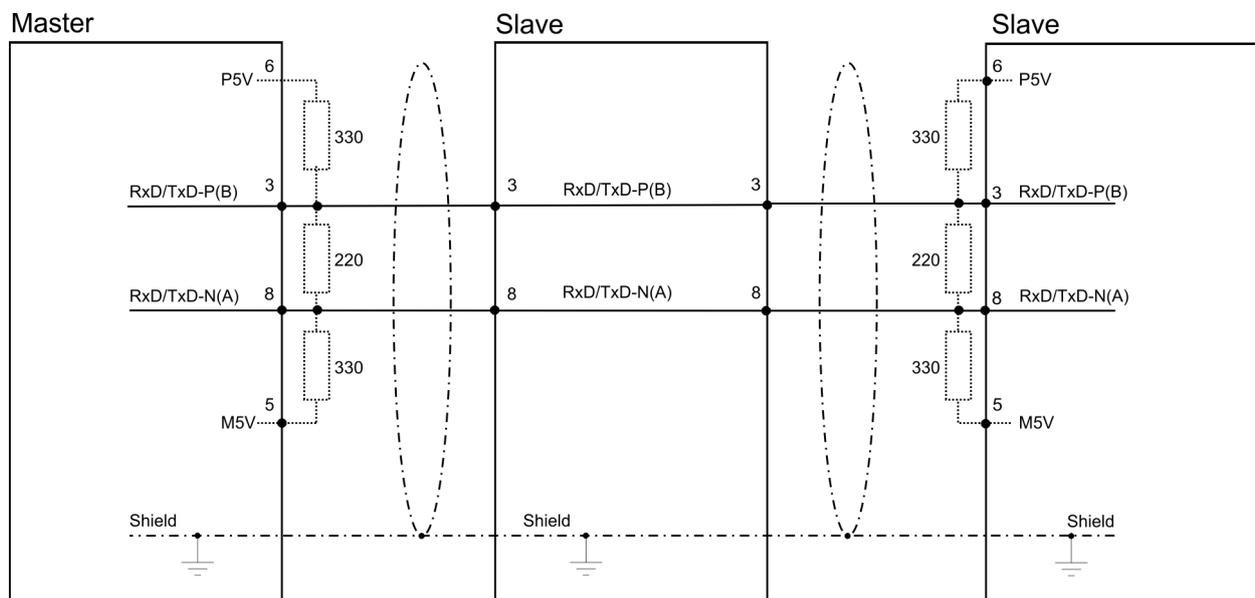
- A PROFIBUS DP network may only be built up in linear structure.
- PROFIBUS DP consists of minimum one segment with at least one master and one slave.
- A master has always been deployed together with a CPU.
- PROFIBUS supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.
- 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.



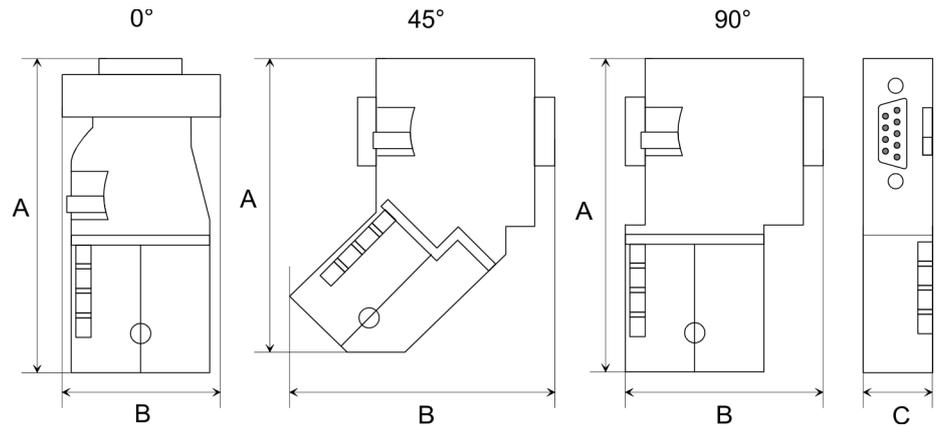


The PROFIBUS line has to be terminated with its ripple resistor. Please make sure to terminate the last participants on the bus at both ends by activating the terminating resistor.

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 cable 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.

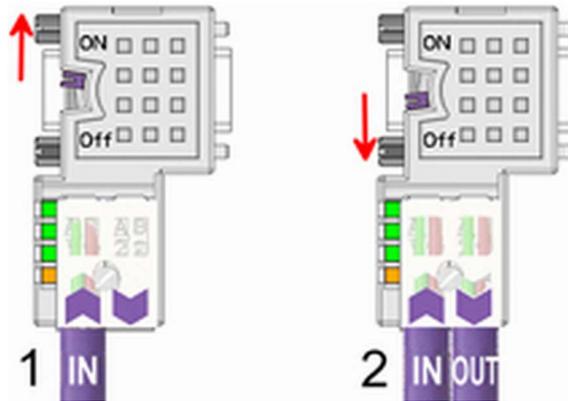


Dimensions in mm

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!

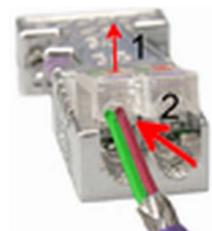
The terminating resistor is only effective, if the connector is installed at a bus participant and the bus participant is connected to a power supply.

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!

11.8 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 <i>Data Exchange</i> 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 | <ul style="list-style-type: none">■ The global control command "Clear" is sent by the master. Then the DP slaves disable the outputs.■ 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 | <ul style="list-style-type: none">■ The global control command "Operate" is sent by the master. Then the DP slaves enable the outputs.■ 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. |

12 Configuration with VIPA *SPEED7 Studio*

12.1 *SPEED7 Studio* - Overview

SPEED7 Studio - Working environment

In this part the project engineering of the VIPA CPU in the VIPA *SPEED7 Studio* is shown. Here only the basic usage of the *SPEED7 Studio* 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. In the *SPEED7 Studio* 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 *SPEED7 Studio*.*

Starting the *SPEED7 Studio*



- ➔ Click at the button. You can find *SPEED7 Studio* in Windows Start at 'VIPA'.
- ⇒ *SPEED7 Studio* is started. The *start page* is opened.

SPEED7 Studio

| Project Folder | Source | Last Access |
|----------------|--------|-----------------------|
| MyProject | local | 10/29/2013 5:59:03 PM |

- (1) Start You can create a new project, open a saved project, or delete projects.
- (2) Project If a project is open, you can open the 'Project overview' or add a new device.
- (3) Last projects Here recently opened projects are listed.



*You can repeatedly run *SPEED7 Studio* in order to work with different projects. You can not open the same project in the various instances of *SPEED7 Studio*.*

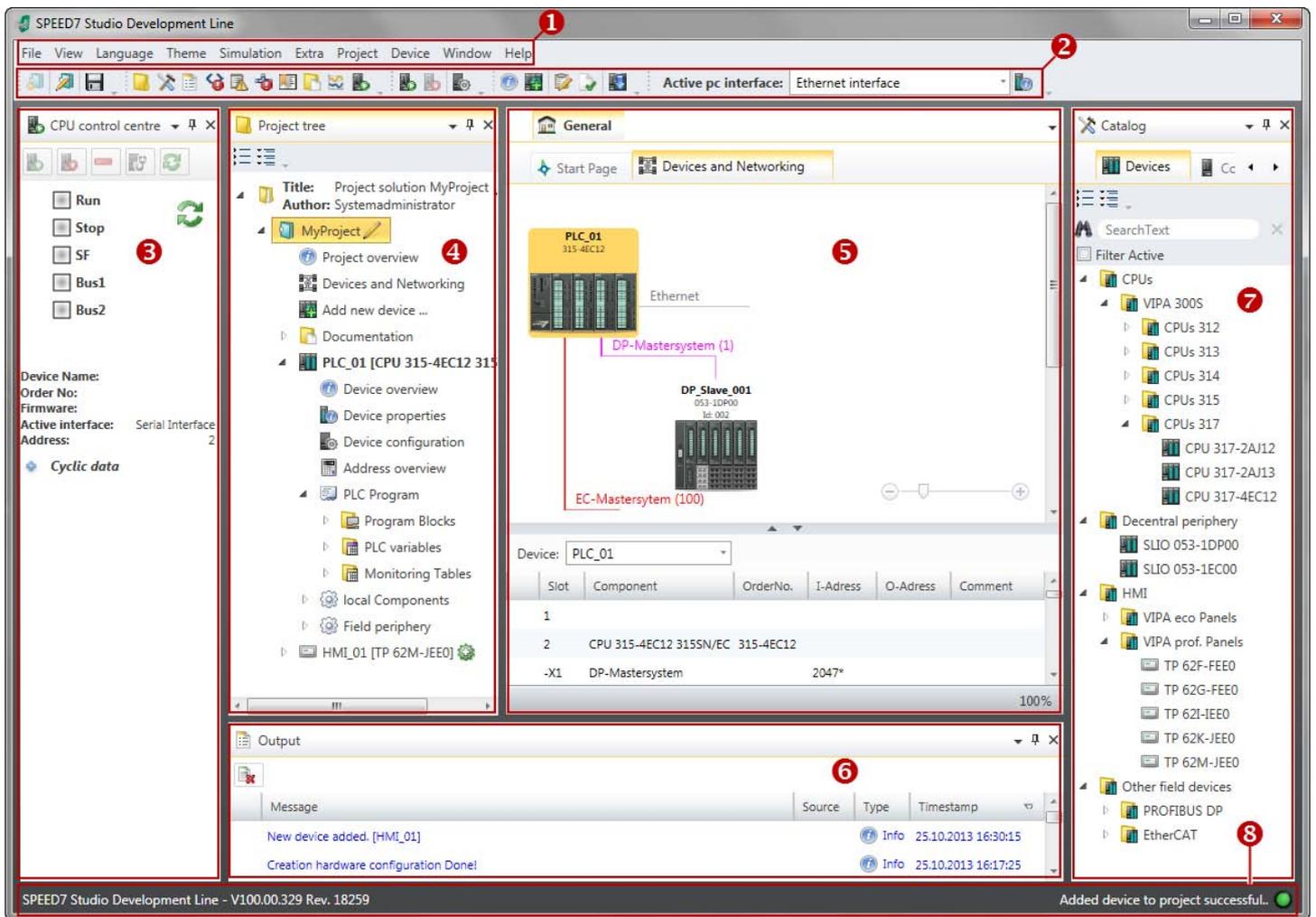
End *SPEED7 Studio*

- ➔ Select one of the following options if you want to end the program:
 - **Main window:** Click on the Close button of the *SPEED7 Studio* program window.
 - **Menu bar** Select '*File* ➔ *Exit*'.
 - **Keyboard:** Press [Alt] + [F4].

After you have made changes to the project, a dialogue window opens where you can select whether to save or ignore the changes.

⇒ *SPEED7 Studio* is ended.

12.2 *SPEED7 Studio* - Work environment

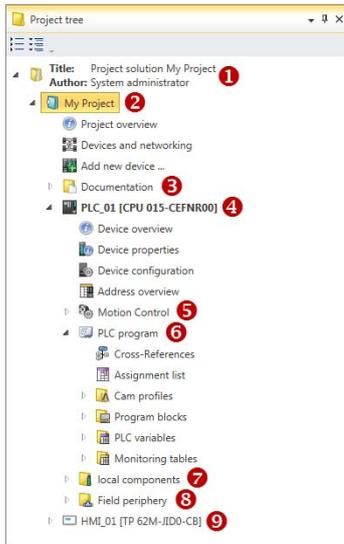


- (1) Menu bar
- (2) Toolbar
- (3) CPU control centre
- (4) Project tree
- (5) Area of operations
- (6) Output range
- (7) Catalog/properties
- (8) Status line

You can show and hide additional windows and the arrangement and size of the windows can be adjusted.

- (1) Menu bar** Most of the commands you need for working with *SPEED7 Studio* are provided in the menu bar. Further commands can be accessed via the context menus using the right mouse button, e.g. functions of a device in the project tree.
- The menu commands *'Project'* and *'Device'* are only shown if a project is open. The menu commands *'Image'* is only shown if a HMI image is open.
- You can use the menus with the mouse or the keyboard.
- (2) Toolbar** Important commands you need for working with *SPEED7 Studio* are provided in the toolbar. More commands can be accessed via the toolbars and push buttons of different editors.
- Some of the commands in the toolbar are only shown if a project is open.
- (3) CPU control centre** In the CPU control centre, you can view the current mode and other control data and control the CPU.
- (4) Project tree** Any project device and project data can be accessed via the project tree. The project tree includes any object which you have created in the project, e.g. devices, components, program blocks, HMI images. Here you can add or remove devices and components. Furthermore, you can open editors in order to edit settings, configurations, the control program and visualisation.
- (5) Area of operations** Devices and project data can be edited in the area of operations. You can open different editors for this purpose. The register in the area of operations is divided into two register levels. You can switch through the editors in the area of operations via the tabs.
- (6) Output range** Information on executed activities and background operations are displayed on the output range.
- (7) Catalog/properties** Devices and components which you want to add to the project can be selected in the catalog. You can also select objects which you want to add to the PLC program or to HMI images.
- (8) Status line** The version of *SPEED7 Studio* is displayed at the left edge of the status line. The progress bar for background operations and status messages is shown at the right edge. As long as there are no background operations, the status message created at last is shown.

12.2.1 Project tree



- (1) Title and author
- (2) Project
- (3) Documentation
- (4) PLC
- (5) Motion Control
- (6) PLC program
- (7) Local components
- (8) Field periphery
- (9) HMI

In the project tree, you can access commands in order to add or delete objects, e.g. add/delete devices or add/delete blocks.

You can open editors via the project tree if you want to edit settings, configurations, the control program and visualisation.

Moreover, you can retrieve information, e.g. project overview, device properties or properties of the bus system.

Show project tree

If the project tree is not displayed, you must select either 'View → Project tree' or press [Strg]+[Shift]+[P].

Show projects in the project tree

In order to display the project in the project tree, you must create a new project or open a stored project.

It is not possible to edit several projects at the same time. It is possible to run *SPEED7 Studio* simultaneously several times on one PC if you want to use it for various projects.

Show/hide objects

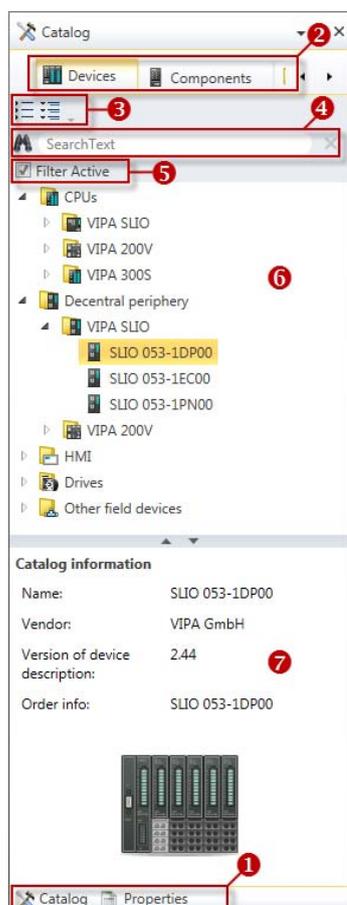
The objects in the project tree are arranged in a tree structure. You can show or hide objects:

-  Hide all objects ('Project → Reduce project tree')
-  Show all objects ('Project → Expand project tree')
-  Hide slave objects / close folder
-  Show slave objects / open folder

Recognise object state

Icons behind an object in the project tree provide indications of the object state.

12.2.2 Catalog



- (1) Switching to another view
- (2) Register
- (3) Show/hide objects
- (4) Search
- (5) Filter
- (6) Objects
- (7) Catalog information

Devices and components which you want to add to the project can be selected in the catalog. You can also select objects which you want to add to the PLC program or to HMI images.

Show catalog:

If the catalog is not displayed, you must select either 'View → Catalog' or press [Strg]+[Shift]+[C].

(1) Switch to another view

If the properties are displayed instead of the catalog, you must click on 'Catalog' at the lower screen edge.

(2) Register

Certain tabs are displayed in the catalog, depending on which editor window is opened in the foreground.

(3) Show/hide objects

The objects in the catalog are arranged in a tree structure. You can show or hide objects:

Hide all objects ('Project → Reduce project tree')

 Hide all objects ('Project → Reduce catalog tree')

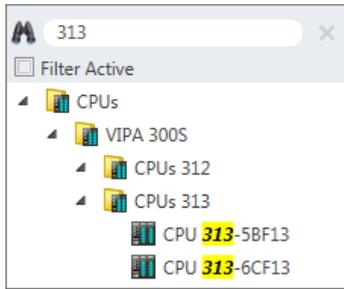
 Show all objects ('Project → Expand catalog tree')

 Hide slave objects / close folder

 Show slave objects / open folder

(4) Search

You can search for certain objects in the catalog.



1. Enter a search text in the input field.
 - ⇒ Only those objects are displayed in the catalog which contain the search text.
2. Click on to delete the search text.
 - ⇒ All objects are displayed in the catalog.

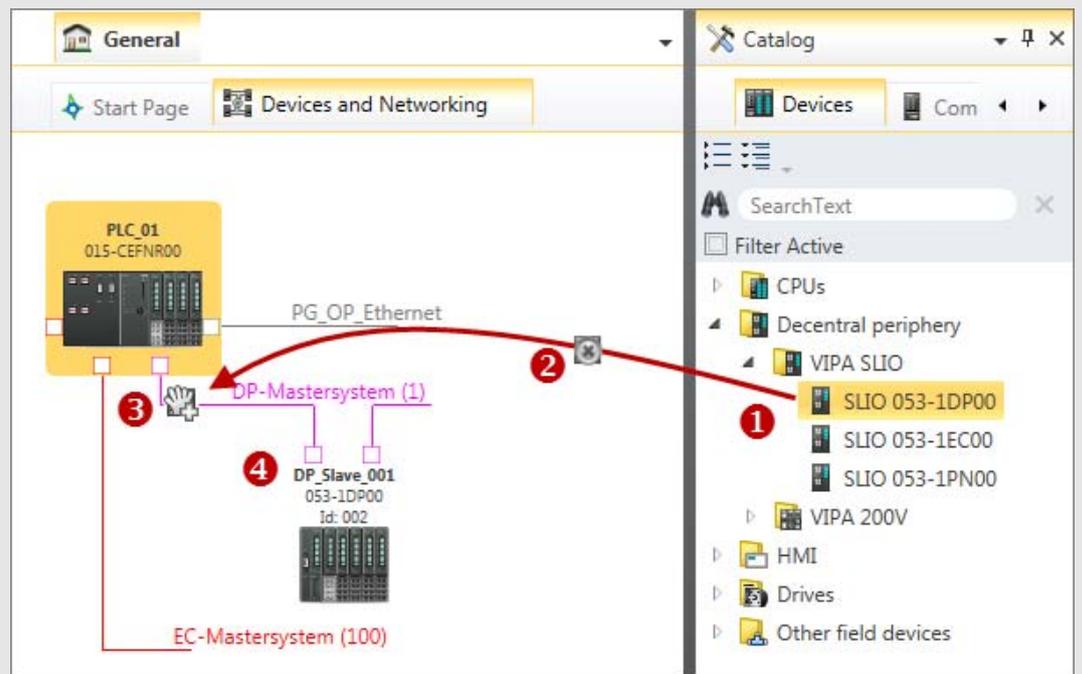
(5) Filter

With 'enabled' Filter, only these modules are shown in the *Catalog* which are relevant for configuration

(6) Add object

- ➔ Drag the desired object from the catalog to a suitable position.
 - ⇒ The object is added.

Example



- (1) Select the desired object (hold left mouse button down)
- (2) Drag the object
- (3) Drop the object at a suitable place (release the mouse button)
- (4) The object is added

(7) Catalog information

The catalog information shows detailed information of the selected object, e.g. name, producer, version and order information.

12.3 SPEED7 Studio - Hardware configuration - CPU

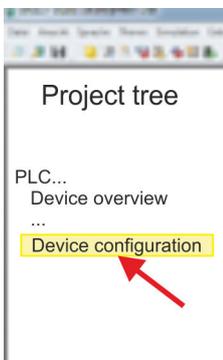
Precondition



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 013-CCF0R00 | | | | |
| -X1 | PG_OP_Ethernet | | | | |
| -X3 | MPI interface | | | | |
| ... | ... | | | ... | |

12.4 *SPEED7 Studio* - Hardware configuration - Ethernet PG/OP channel

Overview



Please note!

- *At the first commissioning respectively after a reset to factory setting the Ethernet interface has no IP address.*
- *For online access, you have to assign valid IP address data to it by means of "Initialization".*
- *After initialization, you can transfer the IP address data to your project.*

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/X2) is designed as switch. This enables PG/OP communication via the connections X1 and X2.
- Configurable connections are possible.
- DHCP respectively the assignment of the network configuration with a DHCP server is supported.
- Default diagnostics addresses: 2025 ... 2040
- Via the Ethernet PG/OP channel, you have access to:
 - Device website, where you can find information on firmware status, connected peripherals, current cycle times, etc.
 - *OPC UA* project, which is to be created in the *OPC UA Configurator*.
 - *WebVisu* project, which is to be created in the *SPEED7 Studio*.
 - PROFINET IO controller or the PROFINET I-Device.

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, X2) 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"

The assignment of IP address data takes place via the MAC address. The IP address of your Ethernet PG/OP channel for the interfaces X1 and X2 can be found on the front of your CPU with the name "MAC PG/OP: ...".

X1 PG/OP



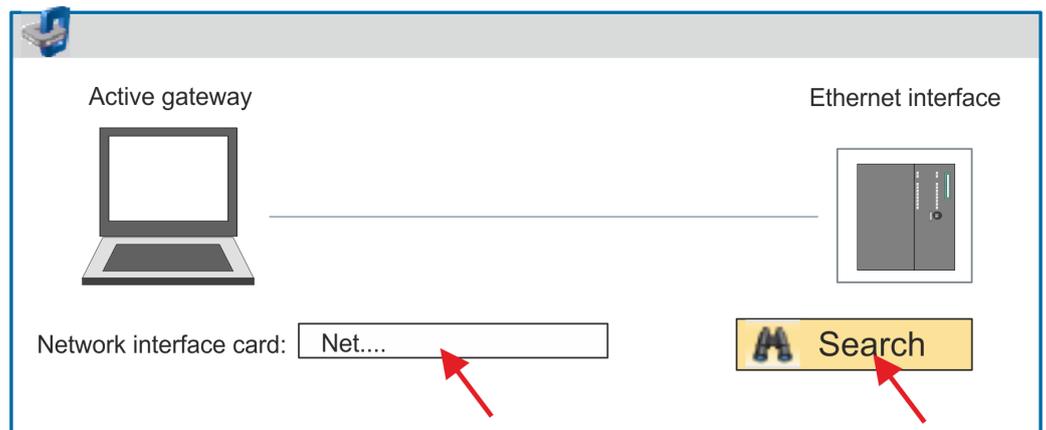
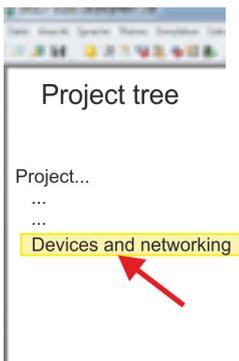
X2 PG/OP



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

The assignment of the IP address data happens online in the *SPEED7 Studio* with the 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* → *Determine accessible partner*'.
⇒ A dialog window opens.



5. 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.

| 6. ➔ | 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: ...".

7. ➔ Click at 'Set IP address'. Now set the IP configuration by entering 'IP address', 'Subnet mask' and 'Gateway'. You get valid IP address parameters from your system administrator.
8. ➔ 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.
9. ➔ 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.

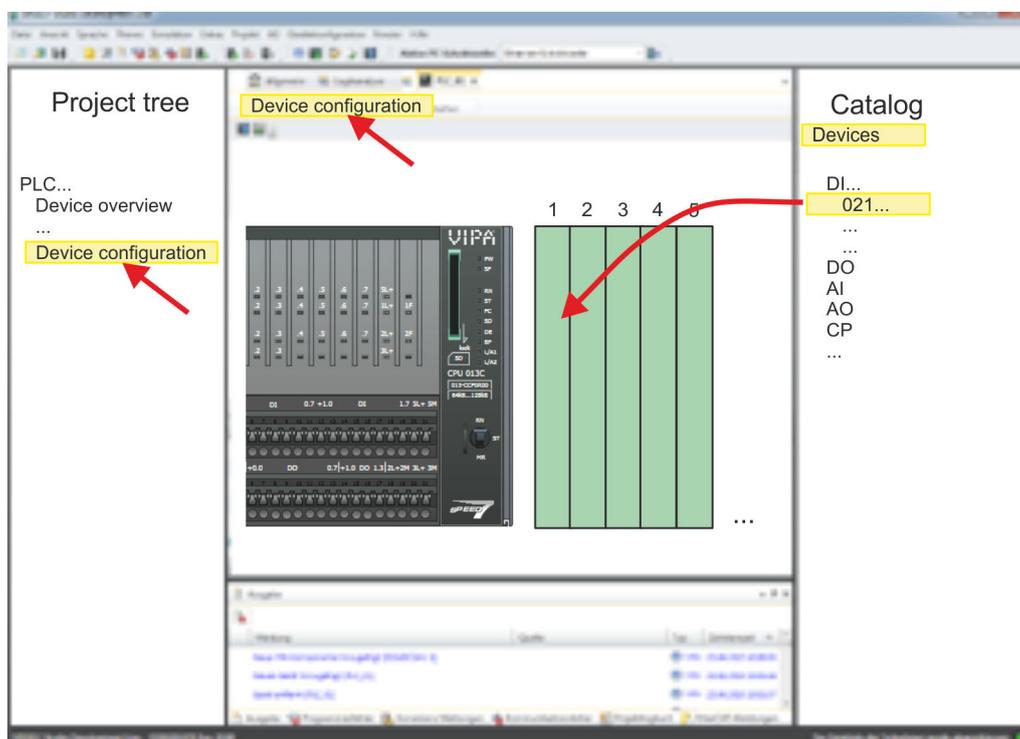
Local components

| Slot | Module | ... | ... | ...IP address | ... |
|------|-----------------|-----|-----|---------------|-----|
| 0 | CPU 013-CCF0R00 | | | ... | |
| -X1 | PG_OP_Ethernet | | | 172.20.120.40 | |
| -X3 | MPI interface | | | ... | |
| ... | ... | | | ... | |

12.5 SPEED7 Studio - 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 parametrize. 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.

12.6 Deployment I/O periphery

12.6.1 Overview

Project engineering and parametrization

- On this CPU the connectors for digital respectively analog signal and *Technological functions* are combined in a one casing.
- Die Project engineering happens in the VIPA SPEED7 Studio as CPU 013-CCF0R00.
- For parametrization of the digital I/O periphery and the *technological functions* the corresponding sub modules of the CPU013-CCF0R00 are to be used.
- The controlling of the operating modes of the *technological functions* happens by means of handling blocks of the user program.

12.6.2 Analog input

12.6.2.1 Overview

- 2xUx12Bit (0 ... 10V)
- Sub module 'A/2'
- ↪ *Chap. 5.3 'Analog input' page 120*

12.6.2.2 Parametrization in *SPEED7 Studio*

12.6.2.2.1 'I/O addresses'

| Sub module | Input address | Access | Assignment |
|------------|---------------|--------|-----------------------------|
| A/2 | 800 | WORD | Analog input channel 0 (X4) |
| | 802 | WORD | Analog input channel 1 (X4) |

12.6.2.2.2 'Parameter'

'Filtering channel 0/1'

The analog input part has a filter integrated. The parametrization of the filter happens via the parameter 'Filter channel 0/1'. The default value of the filter is 1000ms. The following values can be entered:

- 2ms: no filter
- 100ms: small filter
- 1000ms: medium filter
- 10000ms: maximum filter

12.6.3 Digital input

12.6.3.1 Overview

- 16xDC 24V
- Sub module 'DI16/DO12'
- ↪ *Chap. 5.4 'Digital input' page 123*

12.6.3.2 Parametrization in *SPEED7 Studio*

12.6.3.2.1 'I/O addresses'

| Sub module | Input address | Access | Assignment |
|------------|---------------|--------|------------------------------------|
| DI16/DO12 | 136 | BYTE | Digital input I+0.0 ... I+0.7 (X4) |
| | 137 | BYTE | Digital input I+1.0 ... I+1.7 (X4) |

12.6.3.2.2 'Inputs'

'Trigger for process interrupt'

Here you can specify a hardware interrupt for each input for the corresponding edge. The hardware interrupt is disabled, if nothing is selected (default setting). A diagnostics interrupt is only supported with *Hardware interrupt lost*.

Here is valid:

- Rising edge: Edge 0-1
- Falling edge: Edge 1-0

Input delay

- The input delay can be configured per channel in groups of 4.
- An input delay of 0.1ms is only possible with "fast" inputs, which have a max. input frequency of 100kHz ↪ *X4: Connector* page 45. Within a group, the input delay for slow inputs is limited to 0.5ms.
- Range of values: 0.1ms / 0.5ms / 3ms / 15ms

12.6.4 Digital output

12.6.4.1 Overview

- 12xDC 24V, 0.5A
- Sub module 'DI16/DO12'
- ↪ *Chap. 5.5 'Digital output'* page 126

12.6.4.2 Parametrization in *SPEED7 Studio*

12.6.4.2.1 'I/O addresses'

| Sub module | Output address | Access | Assignment |
|------------|----------------|--------|-------------------------------------|
| DI16/DO12 | 136 | BYTE | Digital output Q+0.0 ... Q+0.7 (X5) |
| | 137 | BYTE | Digital output Q+1.0 ... Q+1.3 (X5) |

12.6.5 Counter

12.6.5.1 Overview

- 4 channels
- Sub module: 'Counter'
- ↪ *Chap. 5.6 'Counting'* page 128

12.6.5.2 Parametrization in *SPEED7 Studio*

12.6.5.2.1 'I/O addresses'

| Sub module | Input address | Access | Assignment |
|------------|---------------|--------|--|
| Count | 816 | DINT | Channel 0: Counter value / Frequency value |
| | 820 | DINT | Channel 1: Counter value / Frequency value |
| | 824 | DINT | Channel 2: Counter value / Frequency value |
| | 828 | DINT | Channel 3: Counter value / Frequency value |

12.6.5.2.2 Basic parameters

Select interrupt

Via 'Basic parameters' you can reach 'Select interrupt'. Here you can define the interrupts the CPU will trigger. The following parameters are supported:

- None: The interrupt function is disabled.
- Process: The following events of the counter can trigger a hardware interrupt (selectable via 'Count'):
 - Hardware gate opening
 - Hardware gate closing
 - On reaching the comparator
 - on Counting pulse
 - on overflow
 - on underflow
- Diagnostics+process: A diagnostics interrupt is only triggered when a hardware interrupt was lost.

12.6.5.2.3 'Channel x'

Operating mode

Select via 'Channel' the channel select via 'Operating' the counter operating mode. The following counter operating modes are supported:

- Not parametrized: Channel is de-activated
- Count endless
- Count once
- Count periodical

Counter

Operating mode

Default values and structure of this dialog box depend on the selected 'Operating mode'.

Parameter overview

| Operating parameters | Description | Assignment |
|----------------------|--|--|
| Main count direction | <ul style="list-style-type: none"> ■ <i>None</i> No restriction of the counting range ■ <i>Up</i>: Restricts the up-counting range. The counter starts from 0 or <i>load value</i>, counts in positive direction up to the declaration <i>end value</i> -1 and then jumps back to <i>load value</i> at the next positive transducer pulse. ■ <i>Down</i>: Restricts the down-counting range. The counter starts from the declared <i>start value</i> or <i>load value</i> in negative direction, counts to 1 and then jumps to <i>start value</i> at the next negative encoder pulse. Function is disable with <i>count continuously</i>. | <ul style="list-style-type: none"> ■ None |
| Gate function | <ul style="list-style-type: none"> ■ <i>Cancel count</i>: The count starts when the gate opens and resumes at the <i>load value</i> when the gate opens again. ■ <i>Stop count</i>: The count is interrupted when the gate closes and resumed at the last actual counter value when the gate opens again. <p>🔗 Chap. 5.6.6.2 'Gate function' page 142</p> | Abort count process |

| Operating parameters | Description | Assignment |
|----------------------|---|---------------------------|
| Start value | <i>Start value</i> with counting direction backward. | 2147483647 ($2^{31}-1$) |
| End value | <i>End value</i> with main counting direction forward. Range of values: 2...2147483647 ($2^{31}-1$) | |
| Comparison value | The count value is compared with the <i>comparison value</i> . See also the parameter "Characteristics of the output": <ul style="list-style-type: none"> ■ No main counting direction <ul style="list-style-type: none"> – Range of values: -2^{31} to $+2^{31}-1$ ■ Main counting direction forward <ul style="list-style-type: none"> – Range of values: -2^{31} to end value-1 ■ Main counting direction backward <ul style="list-style-type: none"> – Range of values: 1 to $+2^{31}-1$ | 0 |
| Hysteresis | The <i>hysteresis</i> serves the avoidance of many toggle processes of the output, if the counter value is in the range of the <i>comparison value</i> . 0, 1: <i>Hysteresis</i> disabled Range of values: 0 to 255 | 0 |

| Input | Description | Assignment |
|--------------------------|---|-----------------|
| Signal evaluation | Specify the signal of the connected encoder: <ul style="list-style-type: none"> ■ Pulse/direction At the input count and direction signal are connected ■ At the input there is an encoder connected with the following evaluation: <ul style="list-style-type: none"> – Rotary encoder single – Rotary encoder double – Rotary encoder quadruple | Pulse/direction |
| Hardware gate | Gate control exclusively via channel 3: <ul style="list-style-type: none"> ■ enabled: The gate control for channel 3 happens via SW and HW gate ■ disabled: The gate control for channel 3 exclusively happens via SW gate <p>🔗 <i>Chap. 5.6.6.2 'Gate function' page 142</i></p> | disabled |
| Count direction inverted | Invert the input signal ' <i>Direction</i> ': <ul style="list-style-type: none"> ■ enabled: The input signal is inverted ■ disabled: The input signal is not inverted | disabled |

| Output | Description | Assignment |
|-------------------------------|--|---------------|
| Characteristics of the output | <p>The output and the "Comparator" (STS_CMP) status bit are set, dependent on this parameter.</p> <ul style="list-style-type: none"> ■ No comparison: The output is used as normal output and STS_CMP remains reset. ■ Comparator <ul style="list-style-type: none"> – Counter value \geq Comparison value – Counter value \leq Comparison value ■ Pulse at <i>comparison value</i> <ul style="list-style-type: none"> – To adapt the used actuators you can specify a <i>pulse duration</i>. The output is set for the specified <i>pulse duration</i> when the counter value reaches the <i>comparison value</i>. When you've set a main counting direction the output is only set at reaching the <i>comparison value</i> from the main counting direction. | No comparison |
| Pulse duration | <p>Here you can specify the <i>pulse duration</i> for the output signal.</p> <ul style="list-style-type: none"> ■ The <i>pulse duration</i> starts with the setting of the according digital output. ■ The inaccuracy of the <i>pulse duration</i> is less than 1ms. ■ There is no past triggering of the <i>pulse duration</i> when the <i>comparison value</i> has been left and reached again during pulse output. ■ If the <i>pulse duration</i> is changed during operation, it will take effect with the next pulse. ■ If the <i>pulse duration</i> = 0, the output is set until the comparison condition is not longer fulfilled. <p>Range of values: 0...510ms in steps of 2ms</p> | 0 |

| Frequency | Description | Assignment |
|-------------------------|--|------------|
| Max. counting frequency | Specify the max. frequency for track A/pulse, track B/direction, Latch and HW gate | 60kHz |
| Frequency | shortest permissible count pulse | |
| 1kHz | 400µs | |
| 2kHz | 200µs | |
| 5kHz | 80µs | |
| 10kHz | 40µs | |
| 30kHz | 13µs | |
| 60kHz | 6.7µs | |
| 100kHz | 4µs | |

| Hardware interrupt | Description | Assignment |
|------------------------|--|------------|
| Hardware gate opening | Hardware interrupt by edge 0-1 exclusively at HW gate channel 3 <ul style="list-style-type: none"> ■ enabled: Process interrupt by edge 0-1 exclusively at HW gate channel 3 with open SW gate ■ disabled: no hardware interrupt | disabled |
| Hardware gate closing | Hardware interrupt by edge 1-0 exclusively at HW gate channel 3 <ul style="list-style-type: none"> ■ enabled: Process interrupt by edge 1-0 exclusively at HW gate channel 3 with open SW gate ■ disabled: no hardware interrupt | disabled |
| On reaching comparator | Hardware interrupt on reaching <i>comparator</i> <ul style="list-style-type: none"> ■ enabled: Hardware interrupt when comparator is triggered, can be configured via '<i>Characteristics of the output</i>' ■ disabled: no hardware interrupt | disabled |
| Overflow | Hardware interrupt overflow <ul style="list-style-type: none"> ■ enabled: Hardware interrupt on overflow the upper counter limit ■ disabled: no hardware interrupt | disabled |
| Underflow | Hardware interrupt on underrun <ul style="list-style-type: none"> ■ enabled: Hardware interrupt on underflow the lower counter limit ■ disabled: no hardware interrupt | disabled |

12.6.6 Frequency measurement

12.6.6.1 Overview

- 4 channels
- Sub module '*Counter*'
- [↪ Chap. 5.7 'Frequency measurement' page 148](#)

12.6.6.2 Parametrization in *SPEED7 Studio*

12.6.6.2.1 '*I/O addresses*'

| Sub module | Input address | Access | Assignment |
|--------------|---------------|--------|--|
| <i>Count</i> | 816 | DINT | Channel 0: Counter value / Frequency value |
| | 820 | DINT | Channel 1: Counter value / Frequency value |
| | 824 | DINT | Channel 2: Counter value / Frequency value |
| | 828 | DINT | Channel 3: Counter value / Frequency value |

| Sub module | Output address | Access | Assignment |
|------------|----------------|--------|------------|
| Count | 816 | DWORD | reserved |
| | 820 | DWORD | reserved |
| | 824 | DWORD | reserved |
| | 828 | DWORD | reserved |

12.6.6.2.2 Basic parameters

Select interrupt

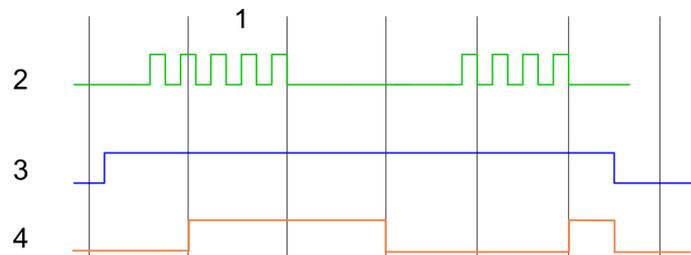
Via 'Basic parameters' you can reach 'Select interrupt'. Here you can define the interrupts the CPU will trigger. The following parameters are supported:

- None: The interrupt function is disabled.
- Process: The following events of the frequency measurement can trigger a hardware interrupt (selectable via 'Frequency counting'):
 - End of measurement
- Diagnostics+process: A diagnostics interrupt is only triggered when a hardware interrupt was lost.

12.6.6.2.3 'Channel x:'

Operating mode

Select via 'Channel' the channel and select for frequency measurement via 'Operating mode' the operating mode 'Frequency counting'. Default values and structure of this dialog box depend on the selected 'Operating mode'. The following parameters are supported:



- 1 Integration time
- 2 Counting pulse
- 3 SW gate
- 4 Evaluated frequency

Parameter overview

| Operating parameters | Description | Assignment | |
|-----------------------------|---|------------|----------------------------------|
| Integration time | Specify the integration time Range of values: 10ms ... 10000ms in steps of 1ms | 100ms | |
| max. counting frequency ... | Specify the max. frequency for the corresponding input | 60kHz | |
| | Frequency | | shortest permissible count pulse |
| | 1kHz | | 400µs |
| | 2kHz | | 200µs |
| | 5kHz | | 80µs |

| Operating parameters | Description | Assignment |
|----------------------|-------------|------------|
| | 10kHz | 40µs |
| | 30kHz | 13µs |
| | 60kHz | 6.7µs |
| | 100kHz | 4µs |

| Hardware interrupt | Description | Assignment |
|--------------------|--|--------------|
| End of measurement | Hardware interrupt at end of measurement | de-activated |

12.6.7 Pulse width modulation - PWM

12.6.7.1 Overview

- 2 channels
- Sub module 'Counter'
- [Chap. 5.8 'Pulse width modulation - PWM' page 153](#)

12.6.7.2 Parametrization in SPEED7 Studio

12.6.7.2.1 'I/O addresses'

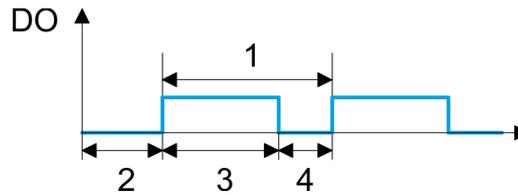
| Sub module | Input address | Access | Assignment |
|------------|---------------|--------|------------|
| Count | 816 | DINT | reserved |
| | 820 | DINT | reserved |
| | 824 | DINT | reserved |
| | 828 | DINT | reserved |

| Sub module | Output address | Access | Assignment |
|------------|----------------|--------|------------|
| Count | 816 | DWORD | reserved |
| | 820 | DWORD | reserved |
| | 824 | DWORD | reserved |
| | 828 | DWORD | reserved |

12.6.7.2.2 'Channel x'

Operating mode

Select via 'Channel' the channel and select for pulse width modulation via 'Operating mode' the operating mode 'Pulse width modulation'. Default values and structure of this dialog box depend on the selected 'Operating mode'. The following parameters are supported:



- 1 Period
- 2 On-delay
- 3 Pulse duration
- 4 Pulse pause

Parameter overview

| Operating parameters | Description | Assignment |
|----------------------|--|------------|
| Output format | <p>Here specify the range of values for the output. The CPU hereby determines the pulse duration:</p> <ul style="list-style-type: none"> ■ Per mil <ul style="list-style-type: none"> – Output value is within 0 ... 1000 – Pulse duration = (Output value / 1000) x Period ■ S7 Analog value: <ul style="list-style-type: none"> – Output value is Siemens S7 analog value 0 ... 27648 – Pulse duration = (Output value / 27648) x Period | Per mil |
| Time base | <p>Here you can set the time base, which will apply for resolution and range of values of the period duration, minimum pulse duration and on-delay.</p> <ul style="list-style-type: none"> ■ 1ms: The time base is 1ms ■ 0.1ms: The time base is 0.1ms ■ 1µs: The time base is 1µs | 0.1ms |
| On-delay | <p>Enter here a value for the time to expire from the start of the output sequence to the output of the pulse. The pulse sequence is output at the output channel, on expiration of the on-delay.</p> <p>Range of values: 0 ... 65535 from this there are the following effective values:</p> <ul style="list-style-type: none"> ■ Time base 1ms: 0 ... 65535ms ■ Time base 0.1ms: 0 ... 6553.5ms ■ Time base 1µs: 0 ... 65535µs | 0 |

| Operating parameters | Description | Assignment |
|------------------------|---|------------|
| Period | <p>With the period you define the length of the output sequence, which consists of pulse duration and pulse pause.</p> <p>Range of values:</p> <ul style="list-style-type: none"> ■ Time base 1ms: 1 ... 87ms ■ Time base 0.1ms: 0.4 ... 87.0ms ■ Time base 1µs: 1 ... 87µs | 20000 |
| Minimum pulse duration | <p>With the minimum pulse duration you can suppress short output pulses and short pulse pauses. All pulses or pauses, which are smaller than the minimum pulse duration, are suppressed. This allows you to filter very short pulses (spikes), which can not be recognized by the periphery.</p> <p>Range of values:</p> <ul style="list-style-type: none"> ■ Time base 1ms: 0 ... $\text{Period} / 2 \cdot 1\text{ms}$ ■ Time base 0.1ms: 2 ... $\text{Period} / 2 \cdot 0.1\text{ms}$ ■ Time base 1µs: 0 ... $\text{Period} / 2 \cdot 1\mu\text{s}$ | 2 |

12.6.8 Pulse Train

12.6.8.1 Overview

- 2 channels
- Sub module 'Count'
- Control by the user program via SFB 49
- [↗ Chap. 5.9 'Pulse train' page 157](#)

12.6.8.2 Parametrization in *SPEED7 Studio*

12.6.8.2.1 'I/O addresses'

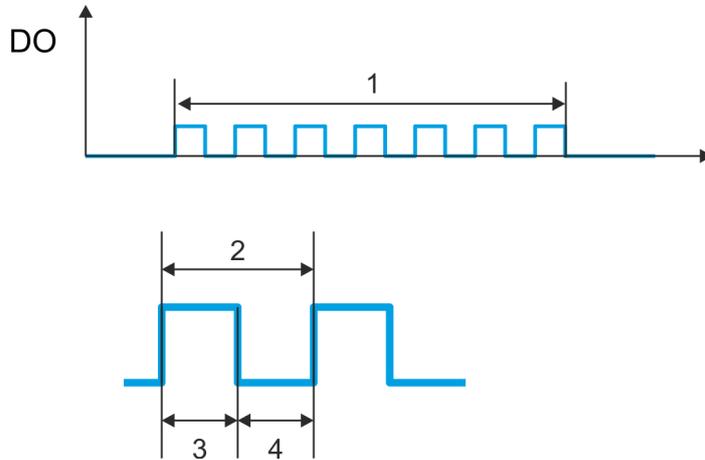
| Sub module | Input address | Access | Assignment |
|------------|---------------|--------|------------|
| Count | 816 | DINT | reserved |
| | 820 | DINT | reserved |
| | 824 | DINT | reserved |
| | 828 | DINT | reserved |

| Sub module | Output address | Access | Assignment |
|------------|----------------|--------|------------|
| Count | 816 | DWORD | reserved |
| | 820 | DWORD | reserved |
| | 824 | DWORD | reserved |
| | 828 | DWORD | reserved |

12.6.8.2.2 'Channel x'

Operating mode

PWM and pulse train output use the same hardware configuration. Switching between these modes is done within the SFB 49. Select via 'Channel x' the channel and select for pulse train via 'Operating mode' the operating mode 'Pulse width modulation'. Default values and structure of this dialog box depend on the selected 'Operating mode'. The following parameters are relevant for pulse train, which must be specified or determined:



- 1 Number of pulses
- 2 Period duration
- 3 Pulse duration
- 4 Pulse pause

Parameter overview

| Operating parameters | Description | Assignment |
|----------------------|---|------------|
| Output format | Here specify the range of values for the output. The CPU hereby determines the pulse duration: <ul style="list-style-type: none"> ■ Per mil <ul style="list-style-type: none"> – Output value is within 0 ... 1000 – Pulse duration = (Output value / 1000) x Period ■ S7 Analog value: <ul style="list-style-type: none"> – Output value is Siemens S7 analog value 0 ... 27648 – Pulse duration = (Output value / 27648) x Period | Per mil |
| Time base | Here you can set the time base, which will apply for resolution and range of values of the period duration, minimum pulse duration and on-delay. <ul style="list-style-type: none"> ■ 1ms: The time base is 1ms ■ 0.1ms: The time base is 0.1ms ■ 1µs: The time base is 1µs | 0.1ms |
| On-delay | This parameter is ignored. | 0 |

| Operating parameters | Description | Assignment |
|------------------------|--|------------|
| Period | <p>With the period you define the length of the output sequence, which consists of pulse duration and pulse pause.</p> <p>Range of values:</p> <ul style="list-style-type: none"> ■ Time base 1ms: 1 ... 87ms ■ Time base 0.1ms: 0.4 ... 87.0ms ■ Time base 1µs: 1 ... 87µs | 50 |
| Minimum pulse duration | <p>With the minimum pulse duration you can suppress short output pulses and short pulse pauses. All pulses or pauses, which are smaller than the minimum pulse duration, are suppressed. This allows you to filter very short pulses (spikes), which can not be recognized by the periphery.</p> <p>Range of values:</p> <ul style="list-style-type: none"> ■ Time base 1ms: 0 ... Period / 2 · 1ms ■ Time base 0.1ms: 2 ... Period / 2 · 0.1ms ■ Time base 1µs: 0 ... Period / 2 · 1µs | 2 |

12.7 Deployment *OPC UA*

↳ *Chap. 7 'Deployment OPC UA' page 183*

12.8 Deployment *WebVisu - Web visualization*

↳ *Chap. 8 'Deployment WebVisu - Web visualization' page 209*

12.9 *SPEED7 Studio - Project transfer*

Overview

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

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

12.9.1 Transfer via MPI

General

For transfer via MPI the CPU has the following interface:

↳ *'X3: MPI(PtP) interface' page 44*

Net structure

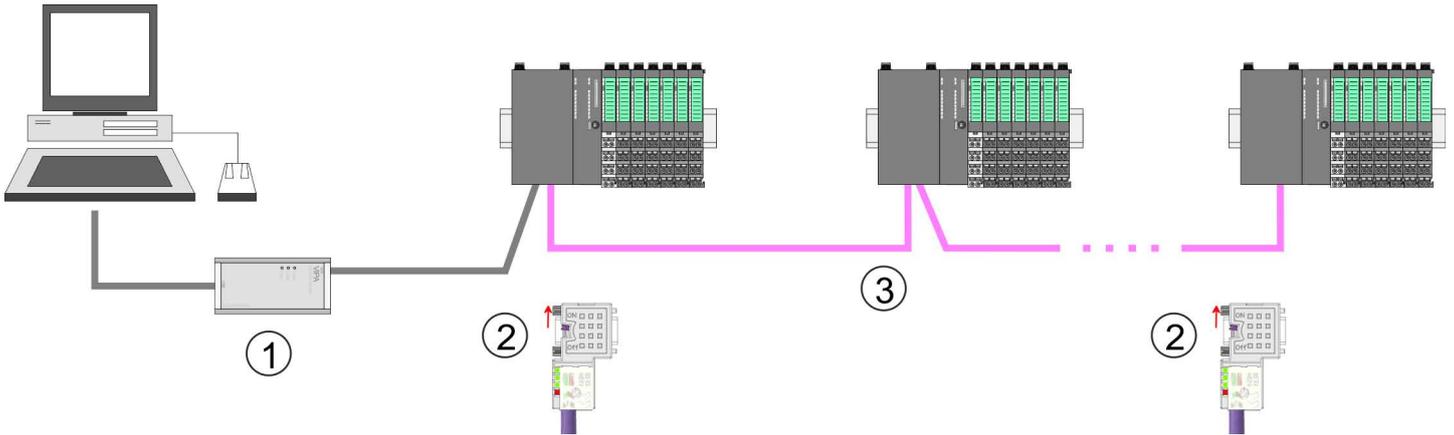
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs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

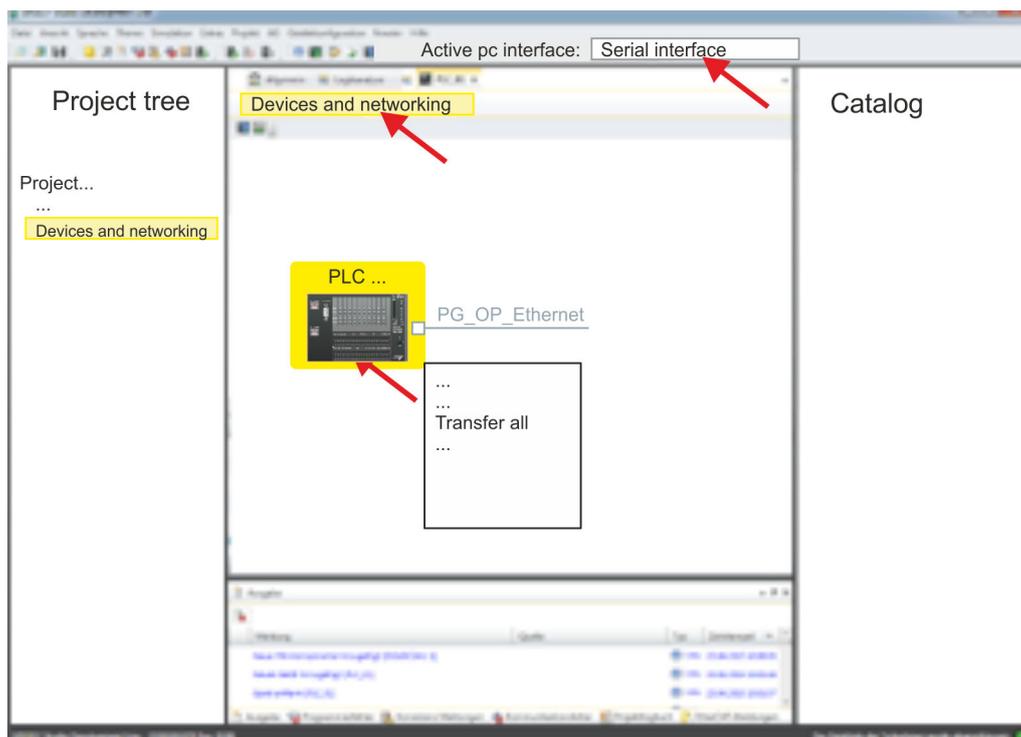
A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment. Please make sure that the participants with the activated terminating resistors are always power supplied. Otherwise it may cause interferences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



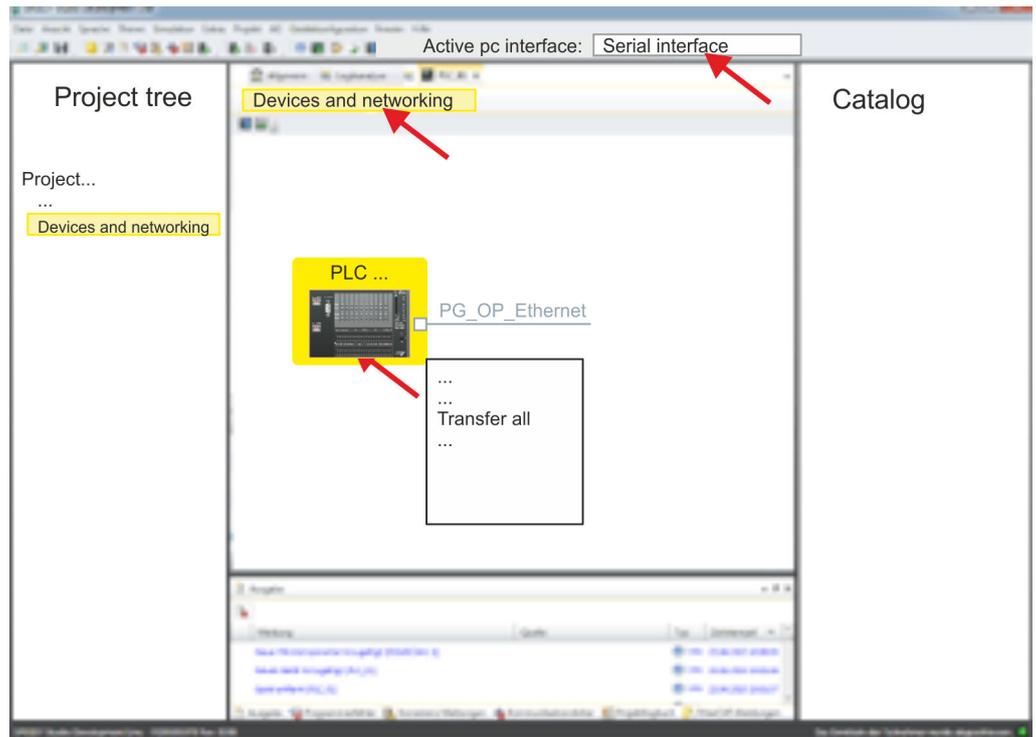
5. To transfer the user program and hardware configuration click in the *Project tree* at your CPU and select *Context menu → Transfer all*.
 - ⇒ A dialog window for project transfer opens
6. Select the *'Port type'* "Serial interface" and start the transfer with *'Transfer'*.
7. Confirm the request 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.

12.9.2 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, X2).

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.

12.9.3 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.

13 Configuration with TIA Portal

13.1 TIA Portal - Work environment

13.1.1 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.

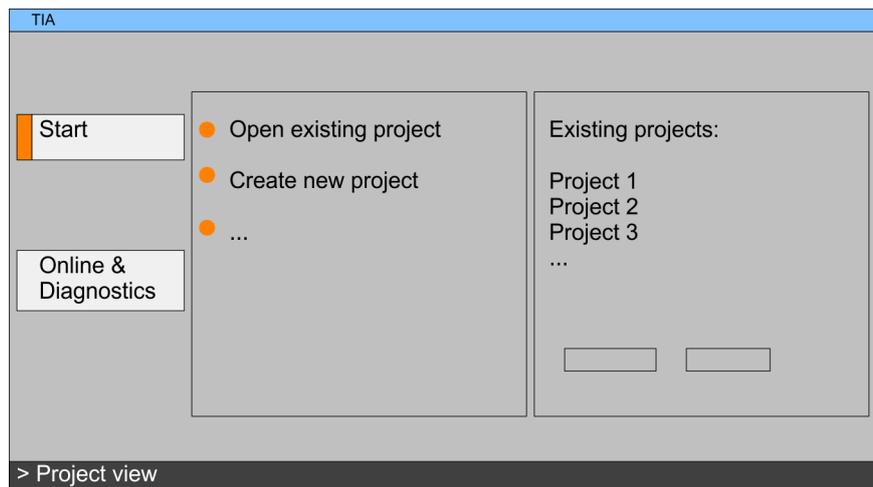


Information about the Siemens TIA Portal can be found in the online help respectively in the according online documentation.

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.

13.1.2 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

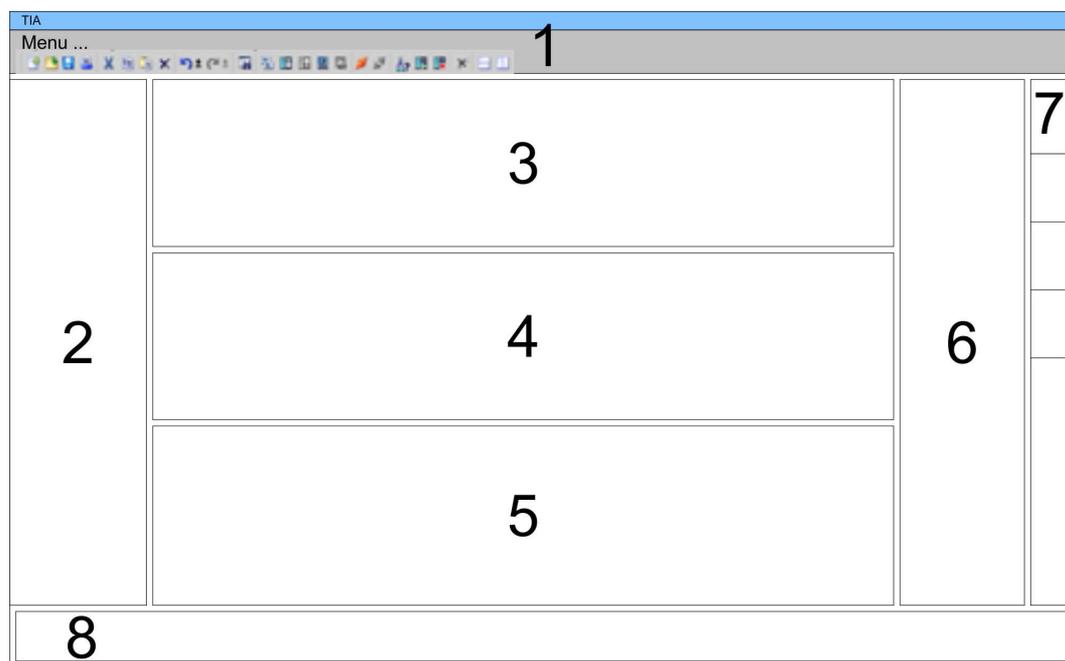
The *'Portal view'* provides a "task oriented" view of the tools for processing your project. Here you have direct access to the tools for a task. If necessary, a change to the Project view takes place automatically for the selected task.

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

13.2 TIA Portal - Functional limitations

| | |
|---|--|
| Limitation of performance data | Please note that the performance data of the CPU from VIPA is limited to the performance data of the Siemens CPU used for the configuration. |
| No 'Upload device as new station...' | <p>Due to the system 'Upload device as new station...' is currently not supported. Instead, use the <i>backup</i> and <i>restore</i> functions in the Siemens TIA Portal:</p> <ol style="list-style-type: none"> 1. ▶ To <i>backup</i> an online connected CPU, select 'Online → Backup from online device'. <ul style="list-style-type: none"> ⇒ A <i>backup</i> object is created and stored in the <i>project navigation</i> at 'Online backups'. The backup contains all blocks of the project and the current device status. 2. ▶ To <i>restore</i> into an online connected CPU it must be overall reset first. Then click in the <i>project navigation</i> under 'Online backups' on the created <i>backup</i> and select 'Context menu → Download to device'. <ul style="list-style-type: none"> ⇒ The backup data is transferred online to the CPU. |
| No online blocks | Due to the system, online blocks of CPUs connected via 'Accessible devices' are currently not listed. |

13.3 TIA Portal - Hardware configuration - CPU

| | |
|---|---|
| Overview | <p>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.</p> <p>The hardware configuration of the CPU is divided into the following parts:</p> <ul style="list-style-type: none"> ■ Installation PROFINET IO device 'VIPA SLIO System' ■ Configuration Siemens CPU ■ Connection SLIO CPU as PROFINET IO device |
| Installation PROFINET IO device 'VIPA SLIO System' | <p>The installation of the PROFINET IO devices 'VIPA SLIO System' happens in the hardware catalog with the following approach:</p> <ol style="list-style-type: none"> 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 <i>Project view</i>. 7. ▶ Select 'Options → Install general station description file (GSD)'. 8. ▶ Navigate to your working directory and install the according GSDML file. <ul style="list-style-type: none"> ⇒ After the installation the hardware catalog is refreshed and the Siemens TIA Portal is finished. <p>After restarting the Siemens TIA Portal the according PROFINET IO device can be found at <i>Other field devices > PROFINET > IO > VIPA GmbH > VIPA SLIO System</i>.</p> |

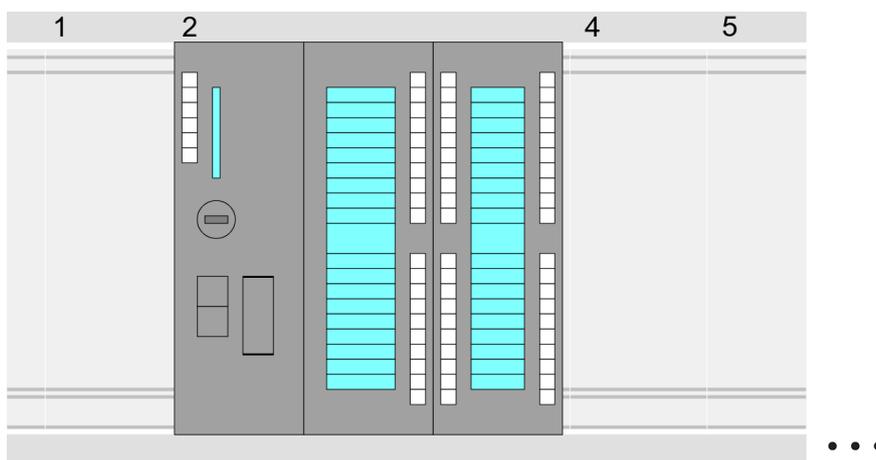


Thus, the VIPA components can be displayed, you have to deactivate the "Filter" of the hardware catalog.

Configuration Siemens CPU

With the Siemens TIA Portal, the CPU from VIPA is to be configured as CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) 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 314C-2 PN/DP (314-6EH04-0AB0 V3.3)
⇒ The CPU is inserted with a profile rail.



Device overview:

| Module | ... | Slot | ... | Type | ... |
|-------------------------|-----|------|-----|--------------------|-----|
| PLC ... | | 2 | | CPU 314C-2 PN/DP | |
| MPI interface... | | 2 X1 | | MPI/DP interface | |
| PROFINET inter- face | | 2 X2 | | PROFINET interface | |
| DI24/DO16 | | 2 5 | | DI24/DO16 | |
| AI5/AO2... | | 2 6 | | AI5/AO2 | |
| Counter... | | 2 7 | | Counter | |
| ... | | | | | |



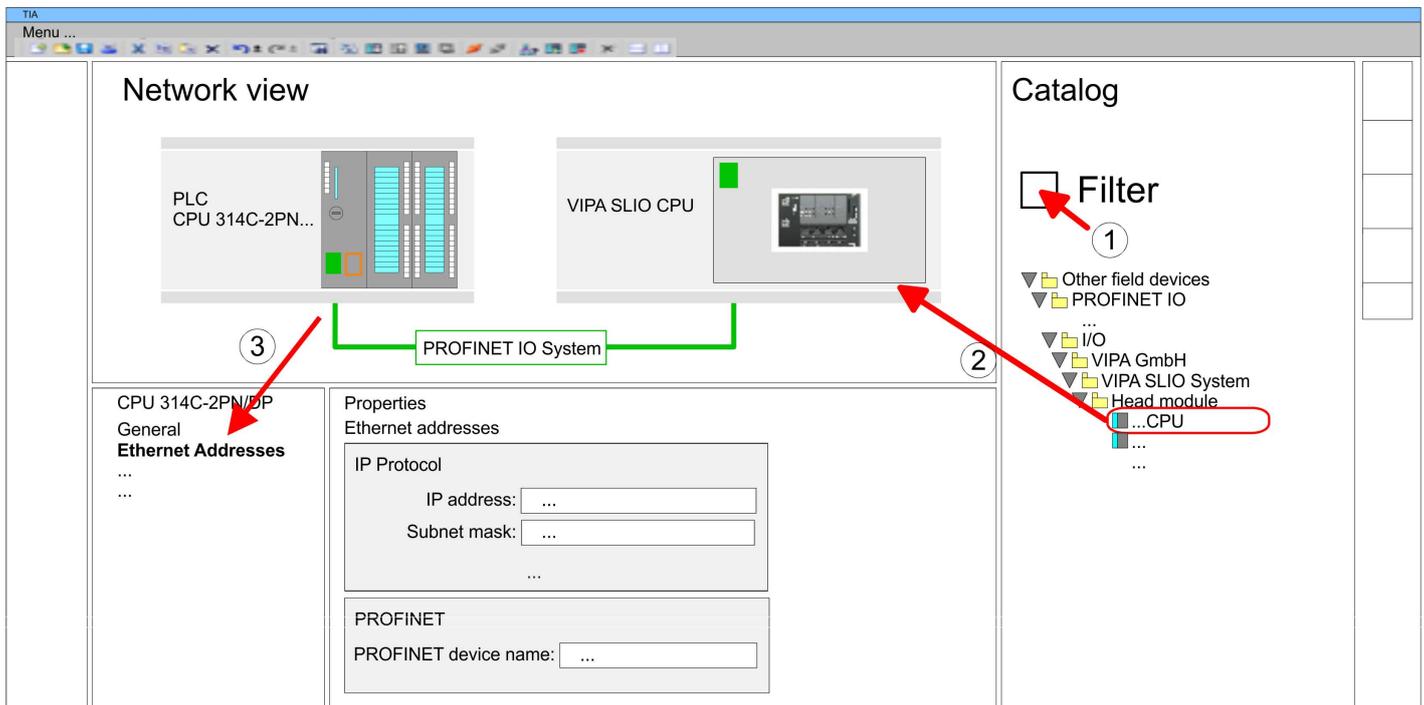
- For parametrization of the digital I/O periphery and the technological functions the corresponding sub modules of the CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) is to be used.
- The controlling of the operating modes of the technological functions happens by means of handling blocks of the user program.

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. ↪ *Chap. 4.8 'Setting standard CPU parameters' page 78*

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 at 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 System*' and switch to the *Device overview*.
 - ⇒ In the *Device overview* of the PROFINET IO device '*VIPA SLIO System*' 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 System'. Then the parameters of the CPU part are shown in the *Properties dialog*. Here you can make your parameter settings. ↪ *Chap. 4.9 'Setting VIPA specific CPU parameters' page 82*

13.4 TIA Portal - Hardware configuration - Ethernet PG/OP channel**Overview****Please note!**

- At the first commissioning respectively after a reset to factory setting the Ethernet interface has no IP address.
- For online access, you have to assign valid IP address data to it by means of "Initialization".
- After initialization, you can transfer the IP address data to your project.

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/X2) is designed as switch. This enables PG/OP communication via the connections X1 and X2.
- Configurable connections are possible.
- DHCP respectively the assignment of the network configuration with a DHCP server is supported.
- Default diagnostics addresses: 2025 ... 2040
- Via the Ethernet PG/OP channel, you have access to:
 - Device website, where you can find information on firmware status, connected peripherals, current cycle times, etc.
 - OPC UA project, which is to be created in the *OPC UA Configurator*.
 - WebVisu project, which is to be created in the *SPEED7 Studio*.
 - PROFINET IO controller or the PROFINET I-Device.

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, X2) 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"

The assignment of IP address data takes place via the MAC address. The IP address of your Ethernet PG/OP channel for the interfaces X1 and X2 can be found on the front of your CPU with the name "MAC PG/OP: ...".

X1 PG/OP



X2 PG/OP

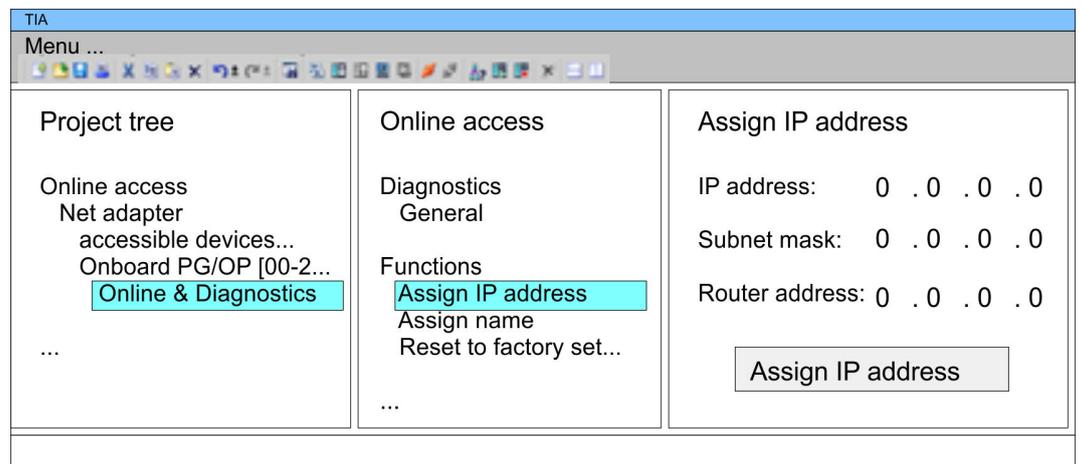


MAC PG/OP: 00-20-D5-77-05-10

"Initialization"

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.

13.4.1 Take IP address parameters in project

2 variants for configuration

From firmware version V2.4.0 and up, you have the following options for configuring the Ethernet PG/OP channel:

- Configuration via integrated CPU interface (firmware version V2.4.0 and up only).
- Configuration via additional CP (all firmware versions).

13.4.1.1 Configuration via integrated CPU interface

Proceeding

From firmware version V2.4.0 this variant for configuration is recommended. The following advantages result:

- The configuration becomes clearer, because the periphery modules and the PROFINET IO devices are configured on the PROFINET line of the CPU and no additional CP is to be configured.
- There are no address collisions, because the S7 addresses for all components are assigned from the address area of the CPU.

Unless during the hardware configuration of the CPU  296 there was no IP address data assigned yet or these are to be changed, the configuration happens to the following proceeding, otherwise the Ethernet PG/OP channel is configured.

1. ➤ Open the Siemens TIA Portal and, if not already done, configure the Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
2. ➤ Click in the *Network view* at the PROFINET part of the Siemens CPU and enter the previous assigned IP address data and subnet in *'Properties'* at *'Ethernet address'* in the area *'IP protocol'*. The IP address data are not accepted without subnet assignment!
3. ➤ Transfer your project.

13.4.1.1.1 Time-of-day synchronization

NTP method

In the NTP mode (**N**etwork **T**ime **P**rotocol) the module sends as client time-of-day queries at regular intervals to all configured NTP servers within the sub net. You can define up to 4 NTP server. Based on the response from the servers, the most reliable and most exact time-of-day is determined. Here the time with the lowest *stratum* is used. *Stratum 0* is the time standard (atomic clock). *Stratum 1* are directly linked to this NTP server. Using the NTP method, clocks can be synchronized over subnet boundaries. The configuration of the NTP servers is carried out in the Siemens SIMATIC Manager via the CP, which is already configured.

1. ➤ In the *'Device configuration'*, click at *'PROFINET interface'* of the Siemens CPU and open the *'Properties'* dialog.
2. ➤ In the *'Properties'*, select *'Time-of-day synchronization'*.
3. ➤ Enable the NTP method.
4. ➤ Add the appropriate NTP servers by specifying their IP addresses.
5. ➤ Set the *'Update interval'* you want. Within this interval, the time of the module is synchronized once.

6.  Save and transfer your project to the CPU.
 - ⇒ After transmission, the NTP time is requested by each configured time server and the best response for the time synchronization is used.



Please note that although the time zone is evaluated, an automatic changeover from winter to summer time is not supported. Industrial systems with time-of-day synchronization should always be set in accordance to the winter time.

With the FC 61 you can determine the local time in the CPU. More information about the usage of this block may be found in the manual "SPEED7 Operation List" from VIPA.

13.4.1.2 Configuration via additional CP

Proceeding

This is the conventional variant for configuration and is supported by all firmware versions. If possible, always use the configuration via the internal interface, otherwise the following disadvantages result:

- Address overlaps are not recognized in the Siemens TIA Portal.
- For PROFINET devices only the address range 0 ... 1023 is available.
- The addresses of the PROFINET devices are not checked with the address space of the CPU from the Siemens TIA portal for address overlaps.

The configuration happens according to the following procedure:

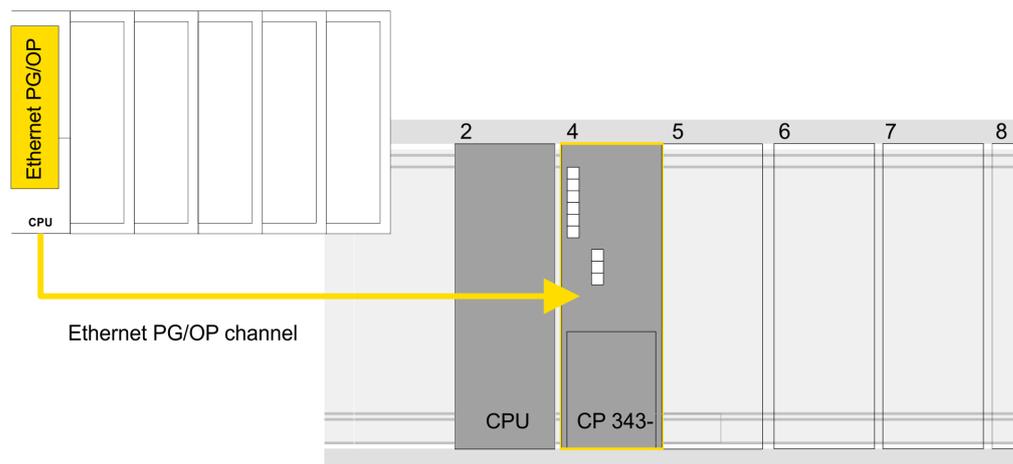
1.  Open your project.
2.  If not already done, configure in the 'Device configuration' a Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
3.  As Ethernet PG/OP channel place at slot 4 the Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).



CAUTION!

Please configure the diagnostic addresses of the CP343-1EX30 for 'PN-IO', 'Port1' and 'Port2' so that no overlaps occur in the periphery input area. Otherwise your CPU can not start-up and you receive the diagnostic entry 0xE904. Address overlaps are not recognized in the Siemens TIA Portal.

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 and subnet, which you have assigned before. The IP address data are not accepted without subnet assignment!
5.  Transfer your project.



1 Ethernet PG/OP channel

Device overview

| Module | ... | Slot | ... | Type | ... |
|--------------------|-----|------|-----|--------------------|-----|
| PLC ... | | 2 | | CPU 314C-2 PN/DP | |
| MPI/DP interface | | 2 X1 | | MPI/DP interface | |
| PROFINET interface | | 2 X2 | | PROFINET interface | |
| ... | | ... | | ... | |
| CP 343-1 | | 4 | | CP 343-1 | |
| ... | | ... | | ... | |

13.4.1.2.1 Time-of-day synchronization

NTP method

In the NTP mode (**N**etwork **T**ime **P**rotocol) the module sends as client time-of-day queries at regular intervals to all configured NTP servers within the sub net. You can define up to 4 NTP server. Based on the response from the servers, the most reliable and most exact time-of-day is determined. Here the time with the lowest *stratum* is used. *Stratum 0* is the time standard (atomic clock). *Stratum 1* are directly linked to this NTP server. Using the NTP method, clocks can be synchronized over subnet boundaries. The configuration of the NTP servers is carried out in the Siemens SIMATIC Manager via the CP, which is already configured.

1. In the *'Device configuration'*, click the CP 343-1EX30.
2. Click on *'PROFINET interface'* in the *'Device overview'*.
3. In the *'Properties'*, select *'Time-of-day synchronization'*.
4. Enable the NTP method by enabling *'Activate time-of-day synchronization'* and selecting *'NTP'* at *'Method'*.
5. Add the appropriate NTP servers by specifying their IP addresses.
6. Select your *'Time zone'*. In the NTP method, UTC (**U**niversal **T**ime **C**oordinated) is generally transmitted; this corresponds to GMT (Greenwich Mean Time). By configuring the local time zone, you can set a time offset to UTC.
7. Set the *'Update interval'* you want. Within this interval, the time of the module is synchronized once.

8. Save and transfer your project to the CPU.
 - ⇒ After transmission, the NTP time is requested by each configured time server and the best response for the time synchronization is used.

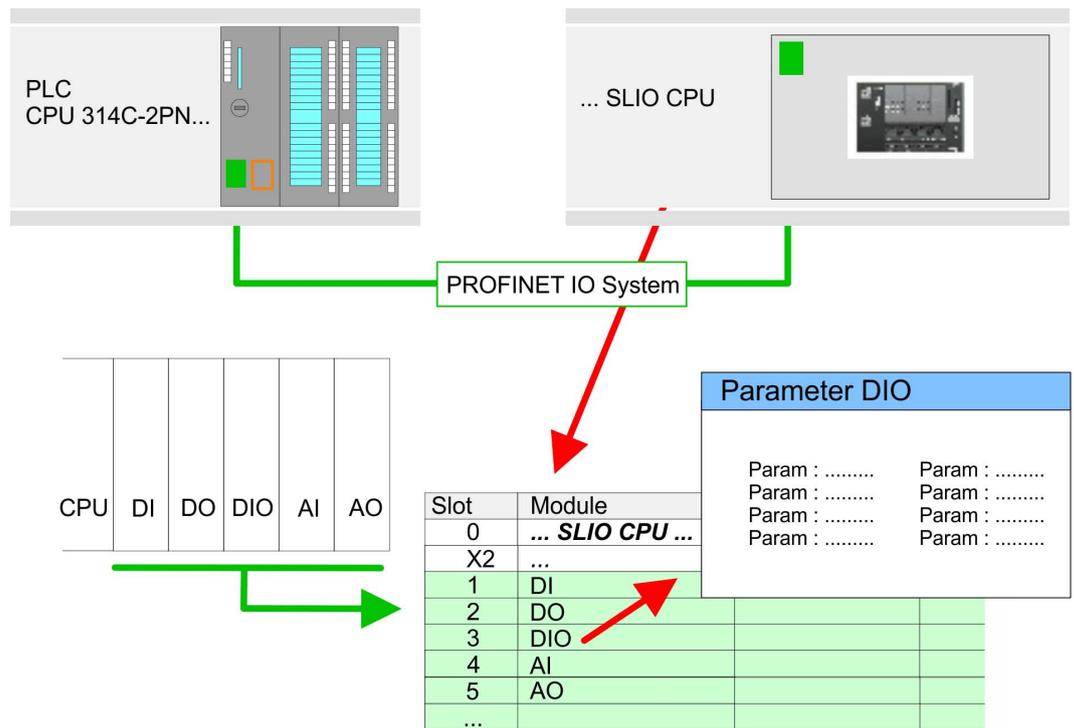
i Please note that although the time zone is evaluated, an automatic changeover from winter to summer time is not supported. Industrial systems with time-of-day synchronization should always be set in accordance to the winter time.

With the FC 61 you can determine the local time in the CPU. More information about the usage of this block may be found in the manual "SPEED7 Operation List" from VIPA.

13.5 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*.



Parameterization

To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU. For parameterization click in the *Device overview* at the module you want to parameterize. Then the parameters of the module are shown in the *Properties* dialog. Here you can make your parameter settings.

13.6 TIA Portal - Deployment PG/OP communication - PROFINET



- With firmware version V2.4.0, there is a PROFINET IO controller available via the Ethernet PG/OP channel.
- As soon as you use the PROFINET functionality via the Ethernet PG/OP channel, this affects the performance and response time of your system and due to the system the cycle time of the OB 1 is extended by 2ms.

13.6.1 Deployment as PROFINET IO controller

13.6.1.1 Steps of configuration



Range of functions

Please regard that the PROFINET IO controller supports only the PROFINET functions, which are described in this manual, even if the Siemens CPU, which is used for configuration, offers further functions! To use some described PROFINET functions, it is necessary to deploy another Siemens CPU for configuration. Here, however, is pointed to explicitly.

The configuration of the PROFINET IO controller for PROFINET communication should be done by the following procedure:

1. Commissioning and Initialization (assignment IP address data)
2. Hardware configuration - CPU
3. Configuration PROFINET IO controller
4. Configuration PROFINET IO device



With the Siemens TIA Portal, the CPU 013-CCF0R00 from VIPA is to be configured as
CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)!

13.6.1.2 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 your PROFINET IO controller with Ethernet.
 4. Switch on the power supply.
- ⇒ After a short boot time, the CP is in idle.

At the first commissioning respectively after an overall reset of the CPU, the PROFINET IO controller and the Ethernet PG/OP channel have no IP address.

Assign IP address parameters

This function is supported only if the PROFINET IO controller is not yet configured. 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 PROFINET interface.
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 ETH: ...".
5. ➤ Choose from the list the module with the known 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 PROFINET IO controller is online reachable using the set IP address data. The hardware configuration allows you to transfer the IP address data to your project. ↪ *Chap. 13.3 'TIA Portal - Hardware configuration - CPU' page 296*

13.6.1.3 Configuration PROFINET IO controller**13.6.1.3.1 Proceeding****Precondition**

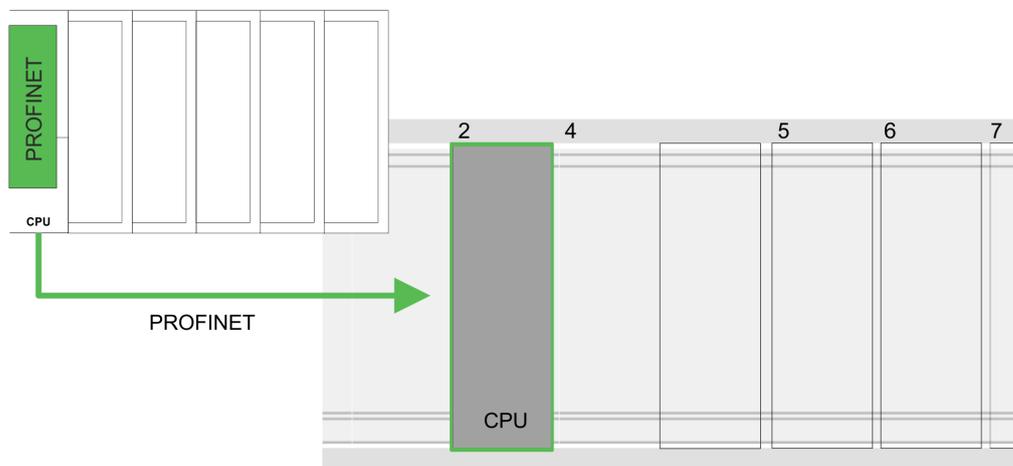
To parametrize the PROFINET IO controller of the CPU, the following conditions must be fulfilled:

- The PROFINET IO controller is online reachable, this means an initialization was established.
- The hardware configuration described before was established and the PROFINET IO controller is networked.

Proceeding

1. ➤ For parametrization click in the *Project area* at the CPU part.
2. ➤ Click on *'PROFINET interface'* in the *Device overview*.
 - ⇒ The parameters of the PROFINET interface are listed in the properties dialog.

3. ➔ Open the properties dialog of the PROFINET IO controller by a double-click at PN-IO.
- The PROFINET interface of the PROFINET IO controller is parametrized with *'PROFINET interface_...'*.
 - With *'Port_1'* the port of the PROFINET interface of the PROFINET IO controller is parametrized.



Device overview

| Module | ... | Slot | ... | Type | ... |
|-------------------------|-----|------|-----|--------------------|-----|
| PLC ... | | 2 | | CPU 314C-2 PN/DP | |
| MPI/DP interface | | 2 X1 | | MPI/DP interface | |
| PROFINET inter- face | | 2 X2 | | PROFINET interface | |
| ... | | ... | | ... | |

13.6.1.3.2 PROFINET interface

General

- Name
 - Here you can change the name of the interface.
- Comment
 - Here you can describe the interface in more detail.

Time-of-day synchronization

- NTP method
 - In this area you can configure time-of-day master for time-of-day synchronization in the network.

Ethernet address

- Interface networked with
 - Here you can connect the PROFINET interface to the desired subnet.
- IP protocol
 - Here you can specify the IP address, subnet mask and gateway for the PROFINET interface.
- PROFINET
 - Here you can specify a '*PROFINET device name*'. On the Ethernet subnet, the '*PROFINET device name*' must be unique. For an integrated PROFINET interface the '*PROFINET device name*' is derived from the short description. You can change this at any time.

Operating mode

- Operating mode
 - You should not change anything here, as the settings for using the PROFINET IO controller as an I-Device are not relevant.

PROFINET interface - Advanced options**Interface Options**

- Support device replacement without exchangeable medium
 - This parameter is not evaluated. With configured topology the *device replacement without exchangeable medium* is supported.

PROFINET interface - Advanced options Port ...1**General**

- Name
 - Here you can change the name of the port, which is also shown in the configuration table.
- Comment
 - Here you can describe the port in more detail. The comment also appears in the configuration table.

Port interconnection

These parameters serve for the port setting of the topology.

Port options

These parameters serve for the port setting. Here the following parameters are supported:

- Connection
 - Here you can make settings for transmission medium and type. Ensure that the setting for the local port and the partner port are identical.
 - PROFINET requires 100Mbps in duplex mode.
- Boundaries
 - *Boundaries* are limitations for the transfer of certain Ethernet frames. The following *Boundaries* are supported:
 - '*End of detection of accessible nodes*': DCP frames for detecting accessible nodes are not forwarded. When enabled, participants which are lying behind this port, are no longer recognized and can not be reached by the controller.
 - '*End of topology discovery*': When activated, this port does not support topology detection, i.e. the LLDP telegrams are not forwarded.

Diagnostic addresses

- Start address
 - The CPU reports errors of the IO controller via the 'Start address', as soon as e.g. an error during synchronization of the IO controller occurs. This address is also used to identify the IO system to which the device belongs, if an IO device fails.

13.6.1.4 Configuration PROFINET IO device**Precondition**

- The modules, which may be configured here are listed in the hardware catalog.
- For the deployment of the PROFINET IO devices from VIPA you have to include the modules into the hardware catalog by means of the GSDML file from VIPA.
- After the installation of the GSDML file the PROFINET IO devices from VIPA may be found in the hardware catalog at *Additional field devices > PROFINET > IO > VIPA GmbH > VIPA SLIO System*

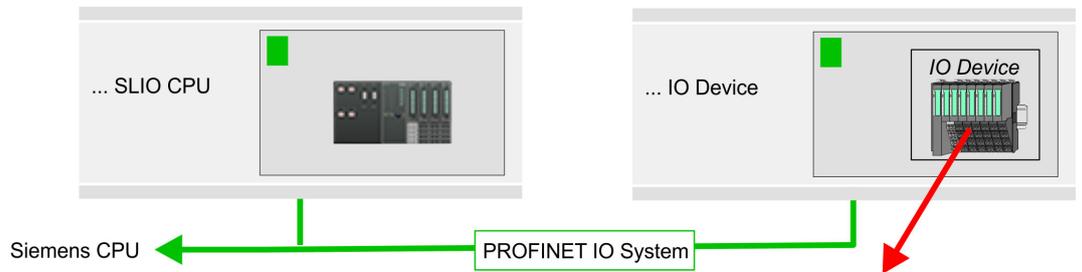


Thus, the VIPA components can be shown, you have to deactivate the "Filter" of the hardware catalog.

Configure IO devices

Now the project engineering of the PROFINET IO controller is finished. Please link up now your IO devices with periphery to your IO controller.

1. ➔ For the project engineering of the PROFINET IO device you search the concerning PROFINET IO device in the hardware catalog at *PROFINET-IO* and drag&drop it in the subnet of your IO controller.
2. ➔ Assign a name to the IO device. The configured name must match the name of the device. Information about setting the device name can be found in the manual of the IO device.
3. ➔ Enter a valid IP address. The IP address is normally assigned automatically by the hardware configurator. If this is not desired, you can assign the IP address manually.
4. ➔ Link up the modules of your IO device in the plugged sequence and add the addresses that should be used by the modules.
5. ➔ If needed, parametrize the modules.
6. ➔ Save, compile and transfer your project. ↪ *Chap. 13.10 'TIA Portal - Project transfer' page 320*



Device overview

| Module | ... | Slot | ... | Type | ... |
|----------------|-----|------|-----|-------------------|-----|
| ...IO device.. | ... | 0 | ... | ...IO device.. | ... |
| ... | ... | 0 | ... | ... | ... |
| Module | ... | 1 | ... | Periphery modules | ... |
| ... | ... | 2 | ... | ... | ... |
| ... | ... | ... | ... | ... | ... |

13.6.2 Deployment as PROFINET I-Device

13.6.2.1 Steps of configuration

Functionality



Please regard that the PROFINET I-Device supports only the PROFINET functions, which are described in this manual, even if the Siemens CPU, which is used for configuration, offers further functionalities!

The *I-Device* (Intelligent IO device) functionality of a CPU allows data to be exchanged with an IO controller, which are preprocessed by the CPU. In this case, the I-Device is connected as an IO device to a higher-level IO controller. The process values, recorded in central or decentralized periphery, can be preprocessed via a user program and made available to the higher-level PROFINET IO controller by means of PROFINET.

- The configuration of the integrated PROFINET IO controller of the CPU as an I-Device is made via a virtual PROFINET devices, which is to be installed by means of a GSDML from VIPA in the hardware catalog.
- The communication takes place via input/output areas, which are defined in the I-Device.
- The size of the areas for input and output data is max. 768byte.
- The I-Device is made available to a deterministic PROFINET IO system via a PROFINET IO interface and thus supports the real-time communication *Real-Time*.
- The I-Device functionality meets the requirements of the RT class I (A) and corresponds to the PROFINET specification version V 2.3.
- In order for the higher-level IO controller to communicate with the VIPA I-Device, the following must be observed:
 - I-Device and higher-level IO controllers must be configured in different networks. Their IP addresses must be in the same IP circuit.
 - The device name of the PROFINET controller of the I-Device must match the device name of the I-Device at the higher-level IO controller.

Configuration

The configuration of the PROFINET IO controller as I-Device should be done by the following procedure:

1. ➤ Installation of the GSDML files
2. ➤ Configuration as I-Device
3. ➤ Configuration in the higher-level IO controller

13.6.2.2 Installation of the GSDML files

The following GSDML files are required for configuring the integrated PROFINET I-Device in the Siemens TIA Portal:

- GSDML for I-Device
- GSDML for I-Device at IO controller

Proceeding

The installation of the PROFINET IO device '*VIPA SLIO CPU*' happens in the hardware catalog with the following approach:

1. ➤ Go to the service area of www.vipa.com.
2. ➤ Download 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)*'.

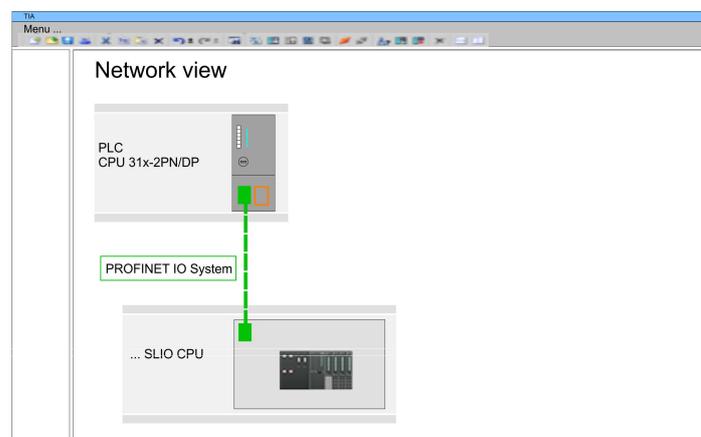
- 8.** → Navigate to your working directory and install the according GSDML file.
- ⇒ After the installation the hardware catalog is refreshed and the Siemens TIA Portal is closed.
- After restarting the Siemens TIA Portal following virtual devices can be found in the hardware catalog at 'PROFINET IO → Other field devices → ... → VIPA SLIO System':
- PN I-Device for VIPA CPU
 - This allows you to configure the Input/output areas in the I-Device of the VIPA CPU.
 - PN I-Device for higher-level CPU
 - This allows you to connect the VIPA I-Device to the higher-level IO controller.



Thus, the VIPA components can be displayed, you have to deactivate the "Filter" of the hardware catalog.

13.6.2.3 Configuration as I-Device

It is assumed that a hardware configuration of the CPU exists. ↪ Chap. 13.3 'TIA Portal - Hardware configuration - CPU' page 296



- 1.** → Place for the Ethernet PG/OP channel at slot 4 the Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
- 2.** → In the *Network view* click on the CP 343-1 and enter a name under 'Properties → PROFINET interface... → General'.



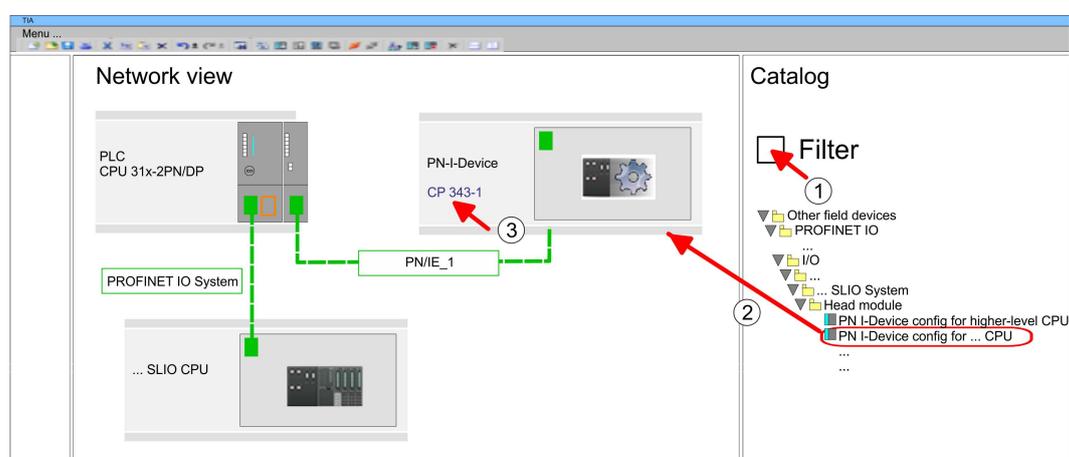
Write down the Name. This name must also be specified as the 'device name' of the I-Device for the higher-level IO controller!

3. ➤ At 'Properties → PROFINET interface... → General' create with 'Add new subnet' a new sub net and assign valid IP address data.

**CAUTION!**

Please configure the diagnostic addresses of the CP343-1EX30 for 'PROFINET interface', 'Port1' and 'Port2' so that no overlaps occur in the periphery input area. Otherwise your CPU can not start-up and you receive the diagnostic entry 0xE904. Address overlaps are not recognized in the Siemens TIA Portal.

4. ➤ After installing the GSDML the 'PN I-Device for VIPA CPU' can be found in the hardware catalog at 'Other field devices → PROFINET IO → I/O → VIPA GmbH → VIPA SLIO System → Head module'. Connect the slave system to the CP 343-1 by dragging&dropping it from the hardware catalog to the Network view and connecting it to the CP 343-1.



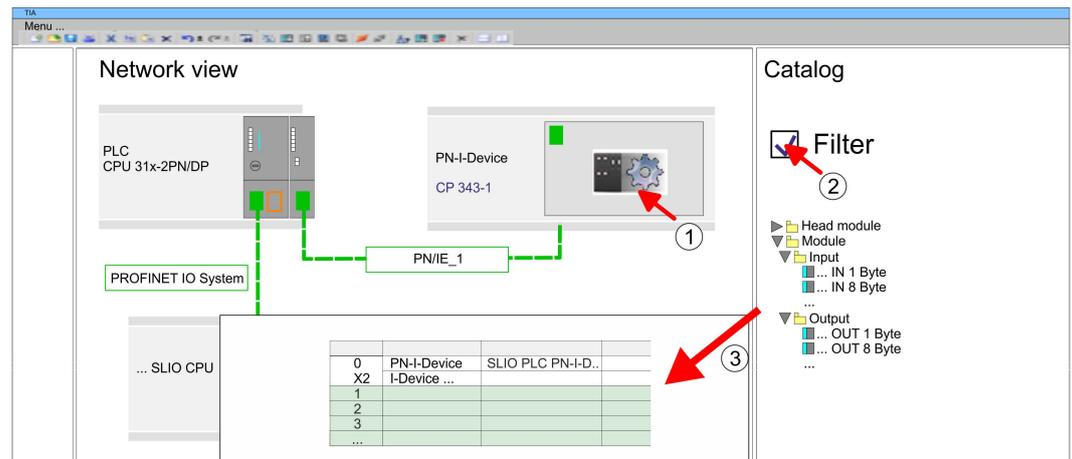
5. ➤ Click in the Network view at the PN I-Device and switch to the Device overview.
6. ➤ Activate the 'Filter' in the hardware catalog.

7. Create the transfer areas by dragging them to the 'Slots' as I/O areas from the hardware catalog to the *Device view*. There must be no gaps in the slots. To create the transfer areas, the following input and output areas are available that can be assigned to the virtual I-Device:

- Input: 1, 8, 16, 32, 64, 128, 256, 512 byte
- Output: 1, 8, 16, 32, 64, 128, 256, 512 byte

The data direction for *Input* or *Output* refers to the view of the I-Device.

- *Input* areas define data that are sent from the higher-level IO controller to the I-Device and which are mapped to the input address area of the CPU.
- *Output* areas define data that are sent to the higher-level IO controller and which are to stored in the output address area of the CPU.

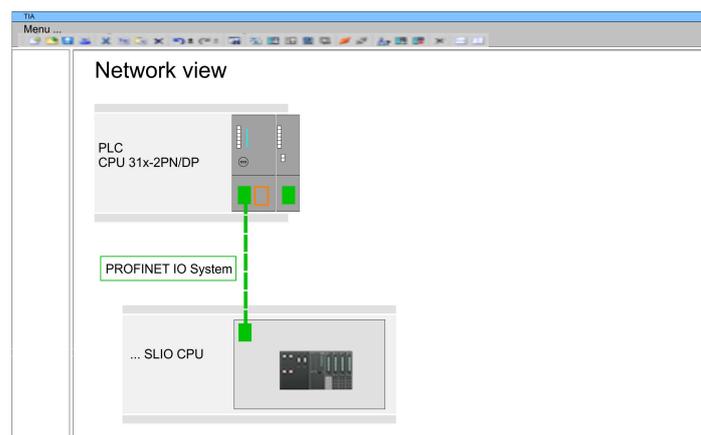


8. Save and transfer your project to the CPU.

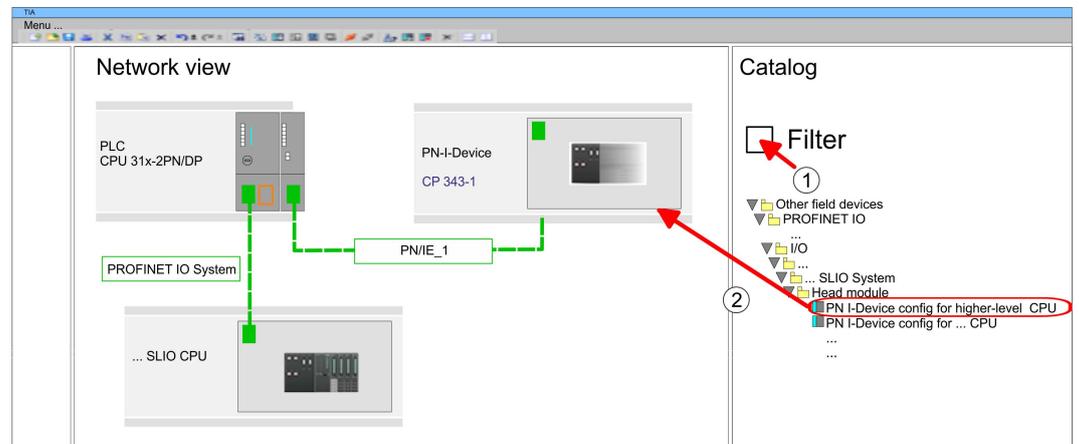
13.6.2.4 Configuration in the higher-level IO controller

It is assumed that a CPU is configured with IP address with the higher-level IO controller. The IP address must be in the same IP circuit as the IP address of the I-Device.

1. Open the project of the CPU with the higher-level IO controller.



2. ➔ For the project engineering of the VIPA I-Device in the high-level IO controller you have to search the device 'PN I-Device config for higher-level CPU' in the hardware catalog at *PROFINET-IO* drag it to *Network view* and connect it via PROFINET to the CPU.

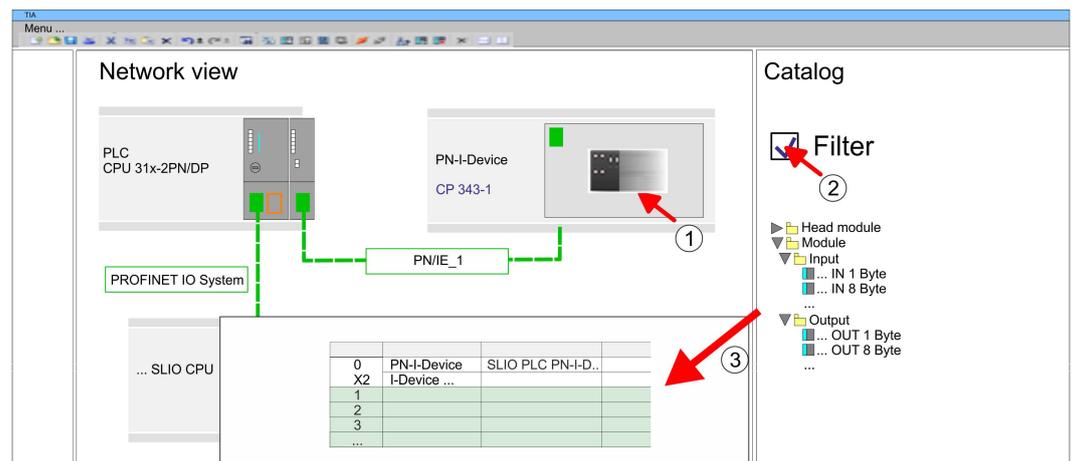


3. ➔ In the 'Network view', click on the 'PN-I-Device' and enter at 'Device name' the previously noted name of the VIPA I-Device.

i The configured name must match the name of the CP 343-1 interface of the I-Device CPU, which you have written down before!
 ↳ Chap. 13.6.2.3 'Configuration as I-Device' page 312

4. ➔ Configure an input area of the same size for each output area of the I-Device in the IO controller and vice versa. Here also no gaps may arise. In particular, make sure that the order of the transfer areas matches that of the I-Device configuration. The following transfer units are available:

- Input: 1, 8, 16, 32, 64, 128, 256, 512 byte per slot
- Output: 1, 8, 16, 32, 64, 128, 256, 512 byte per slot



5. ➔ Save and transfer your project to the CPU.
 - ⇒ Your VIPA PROFINET controller is now connected as an I-Device to the higher-level PROFINET IO controller.

**I-Device with S7 routing**

S7 routing is not possible with the procedure above. S7 routing is only possible if the I-Device and the higher-level I/O controller are configured in the same network. The device names must not be identical. By using identical names and extending the name of the I-Device with "-x", this is detected internally and used appropriately for S7 routing.

13.7 TIA Portal - Option: Deployment PROFIBUS communication

13.7.1 Fast introduction

Overview

The PROFIBUS DP master must be configured in the *Project view*. Here the configuration happens by the sub module X1 (MPI/DP) of the Siemens CPU.



To switch the interface X3 MPI(PtP) 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. ↪ Chap. 4.16 'Deployment storage media - VSD, VSC' page 111

Steps of configuration

The configuration of the PROFIBUS DP master should be done with the following approach:

- Activating bus functionality by means of a VSC
- Hardware configuration - CPU
- Deployment as DP master or deployment as DP slave
- Transfer of the entire project to the CPU

13.7.2 Activating bus functionality by means of a VSC

Proceeding

To use the MPI(PtP) interface X3 as PROFIBUS interface, the PROFIBUS functionality must be activated by means of a VSC memory card from VIPA.

1. ➤ Plug the VSC memory card from VIPA. ↪ 'Overview' page 111
2. ➤ Execute an overall reset. ↪ Chap. 4.13 'Overall reset' page 105
 - ⇒ ■ The PROFIBUS functionality is enabled.
 - With activating the bus functionality "PROFIBUS DP master", the bus functionality "PROFIBUS DP slave" is also enabled.

13.7.3 Hardware configuration - CPU

Perform a hardware configuration for the CPU. ↪ Chap. 13.3 'TIA Portal - Hardware configuration - CPU' page 296

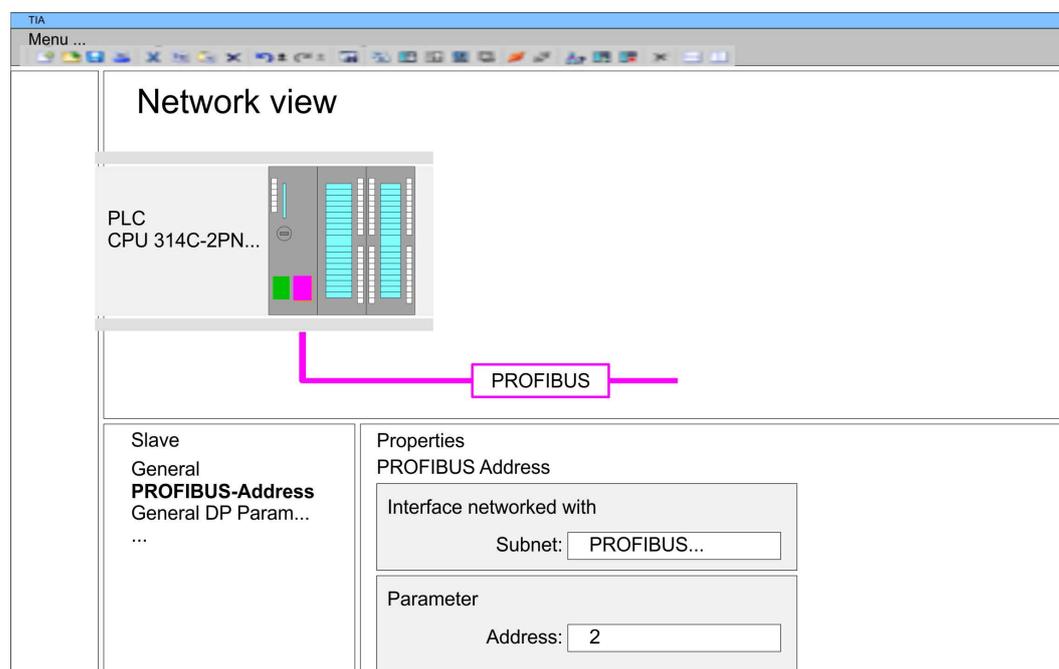
13.7.4 Deployment as PROFIBUS DP master

Precondition

The hardware configuration described before was performed.

Proceeding

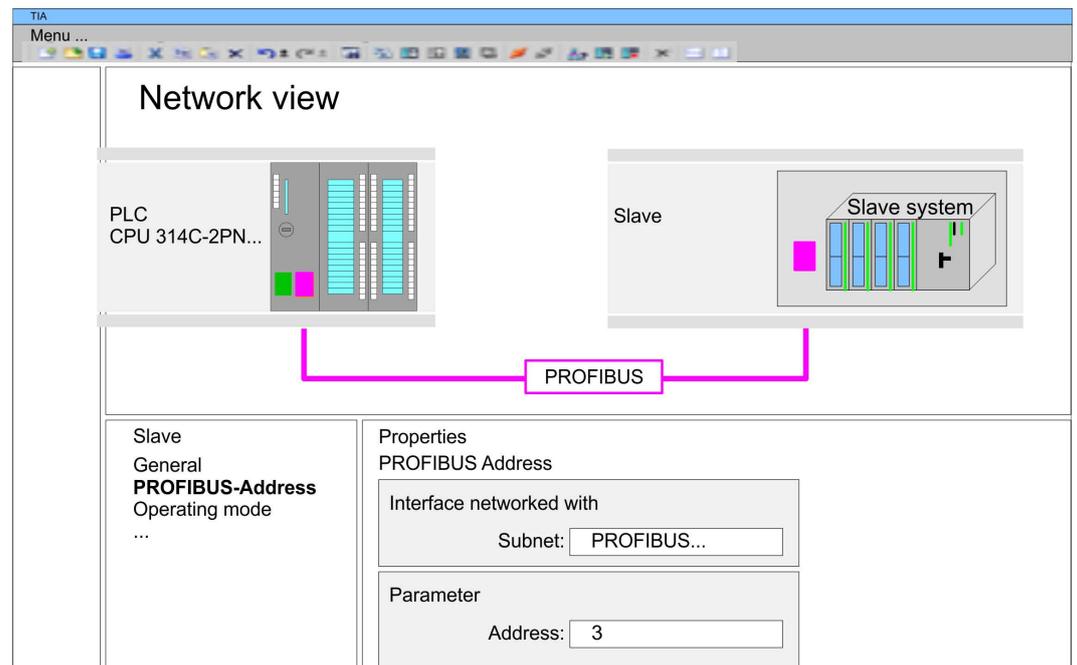
1. ➤ Switch to the 'Device view'.
2. ➤ Select the PROFIBUS interface of your CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) and select 'Context menu → Properties'.
⇒ The 'Properties' dialog is shown.
3. ➤ On the 'General' tab, click 'MPI Address'.
⇒ The dialog for the MPI address appears.
4. ➤ Set the 'Interface type' to "PROFIBUS".
5. ➤ Connect to PROFIBUS and preset an address (preferably 2).
6. ➤ Switch at 'Operating mode' to "DP master".
⇒ A 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.



13.7.5 Deployment as PROFIBUS DP slave

Fast introduction

The following section describes how to use the PROFIBUS part as an "intelligent" DP slave on master systems. Execute the following steps:

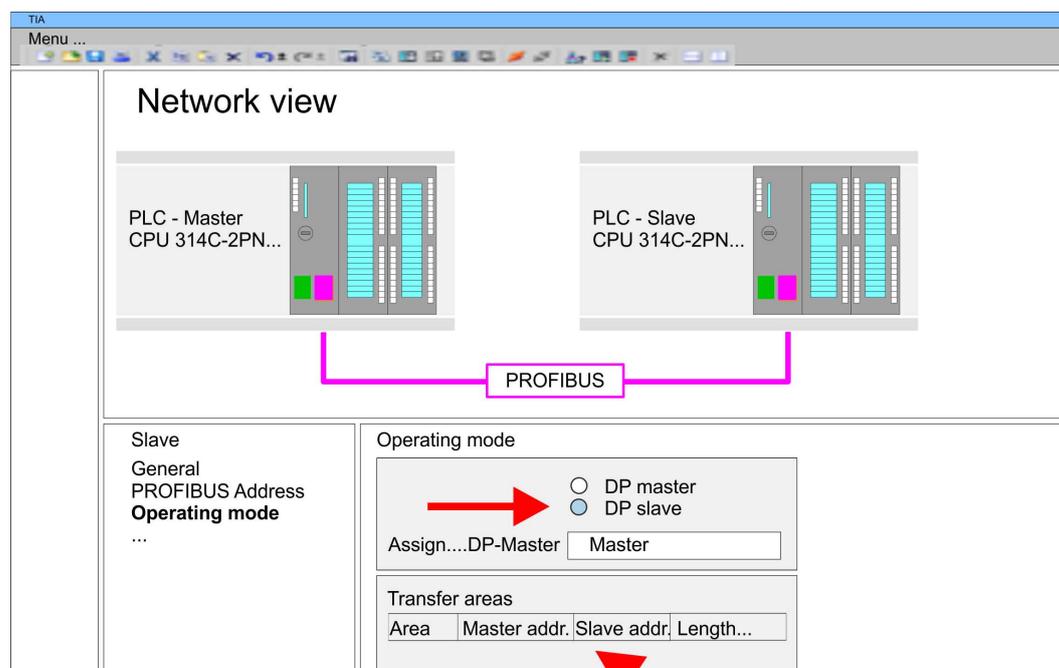
1. Configure a station with a CPU with DP master mode.
2. Network with PROFIBUS
3. Configure another station with a CPU with DP slave operating mode.
4. Network with PROFIBUS and assign the DP slave to the DP master.
5. Configure the I/O areas for the slave part.
 - ⇒ The configuration is automatically adapted to the DP master.
6. Save, translate and transfer the respective project into the corresponding master or slave CPU.

Configuration of the master part

1. Start the Siemens TIA Portal and configure a CPU.
2. Designate the station as "... DP master".
3. Integrate the modules according to the hardware configuration of the master system.
4. Select the PROFIBUS interface of the master CPU and select 'Context menu → Properties'.
 - ⇒ The 'Properties' dialog is shown.
5. Set at *interface*: Type "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 2).
7. Switch at *Operating mode* to "DP master".

Configuration of the slave part

1. ➤ Configure a Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) as another CPU
 ↗ *Chap. 13.3 'TIA Portal - Hardware configuration - CPU' page 296*
2. ➤ Designate the station as "... DP slave".
3. ➤ Integrate your modules according to your hardware configuration.
4. ➤ Select the PROFIBUS interface of your CPU and select 'Context menu
 ➔ Properties'.
 ⇒ The 'Properties' dialog is shown.
5. ➤ Set at interface: Type "PROFIBUS".
6. ➤ Connect to PROFIBUS and preset an address (e.g. 3).
7. ➤ Switch at 'Operating mode' to "DP slave".
8. ➤ Under 'Assigned DP master', select your master system.
9. ➤ Use 'Transfer area' to define the I/O address areas of the slave CPU that are to be assigned to the DP slave.
10. ➤ Save, translate and transfer the respective project into the corresponding master or slave CPU.

**13.8 Deployment OPC UA**

↗ *Chap. 7 'Deployment OPC UA' page 183*

13.9 TIA Portal - VIPA-Include library

Overview

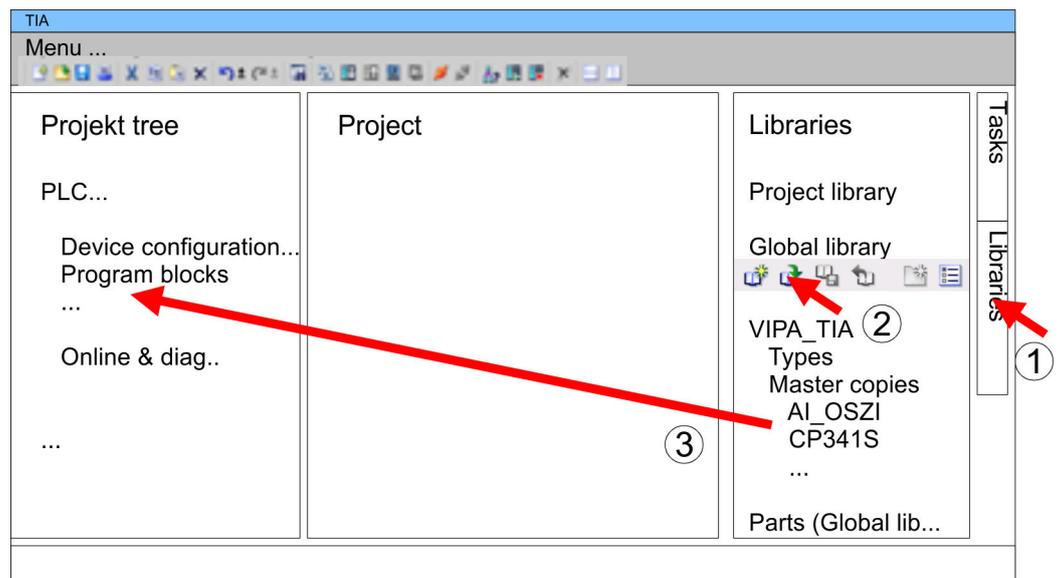
- The VIPA specific blocks can be found in the "Service" area of www.vipa.com as library download file at *Downloads > VIPA LIB*.
- The library is available as packed zip file for the corresponding TIA Portal version.
- As soon as you want to use VIPA specific blocks you have to import them into your project.
Execute the following steps:
 - Load an unzip the file ...TIA_Vxx.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_Vxx.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.alxx.



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.

13.10 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

13.10.1 Transfer via MPI

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.

13.10.2 Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- X1/X2: 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". ↪ *Chap. 13.4 'TIA Portal - Hardware configuration - Ethernet PG/OP channel' page 299*

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.

13.10.3 Transfer via memory card

Proceeding

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 Siemens TIA Portal with your project.
2. ▶ Create 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".
3. ▶ 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".
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.

Appendix

Content

| | | |
|----------|--|------------|
| A | System specific event IDs | 325 |
| B | Integrated blocks | 377 |
| C | SSL partial list | 380 |

A System specific event IDs

Event IDs

[↶ Chap. 4.20 'Diagnostic entries' page 117](#)

| Event ID | Description |
|----------|---|
| 0x115C | Manufacture interrupt (OB 57) for EtherCAT |
| | OB: OB number |
| | ZINFO1: Input / Output address |
| | 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: Recovery 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 |
| | DatID: Input |
| | DatID: Output |
| 0x38D0 | Bus recovery |
| | 0: OB |
| | PK: Priority class |
| | ZINFO1: Logical address of the IO system |
| | ZINFO2: Logical address of the virtual device |

| Event ID | Description |
|----------|--|
| | ZINFO3 - Position 0: Station number |
| | ZINFO3 - Position 11: IO system ID |
| | ZINFO3 - Bit 15: System ID DP/PN |
| 0x38D1 | Bus recovery, but expected configuration does not match actual configuration |
| | 0: OB |
| | PK: Priority class |
| | ZINFO1: Logical address of the IO system |
| | ZINFO2: Logical address of the virtual device |
| | ZINFO3 - Position 0: Station number |
| | ZINFO3 - Position 11: IO system ID |
| | ZINFO3 - Bit 15: System ID DP/PN |
| 0x39D0 | Bus failure |
| | OB: OB number |
| | PK: Priority class |
| | ZINFO1: Logical address of the IO system |
| | ZINFO2: Logical address of the virtual device |
| | ZINFO3 - Position 0: Station number |
| | ZINFO3 - Position 11: IO system ID |
| | ZINFO3 - Bit 15: System ID DP/PN |
| 0x454B | STOP: Maximum number of time-outs of a synchronous OB reached |
| | OB: CPU mode |
| | PK: |
| | ZINFO1: ZInfo1 |
| | ZINFO2: ZInfo2 |
| | ZINFO3: ZInfo3 |
| | DatID: Block type |
| 0x49CA | PROFINET IO system failure (Watchdog) |
| | 0: OB |
| | 1: Cyclic program (OB 1) |
| | 16: Time of day interrupt OB (OB 16) |
| | 17: Time of day interrupt OB (OB 17) |
| | 32: Cyclic interrupt OB (OB 32) |
| | 33: Cyclic interrupt OB (OB 33) |
| | 34: Cyclic interrupt OB (OB 34) |
| | 35: Cyclic interrupt OB (OB 35) |
| | 36: Cyclic interrupt OB (OB 36) |
| | 37: Cyclic interrupt OB (OB 37) |
| | 38: Cyclic interrupt OB (OB 38) |

| Event ID | Description |
|----------|--|
| | 64: Synchronous cycle interrupt (OB 64) |
| | 65: Synchronous technology interrupt (OB 65) |
| | 80: Time error OB (OB 80) |
| | 81: Power supply error OB (OB 81) |
| | 82: Diagnostic interrupt OB (OB 82) |
| | 83: Plug/Pull OB (OB 83) |
| | 85: Program sequence error OB (OB 85) |
| | 86: Component rack failure OB (OB 86) |
| | PK: Priority class |
| | ZINFO1: Logical address of the IO system |
| | ZINFO2: Interrupt reason |
| | 0: Unknown |
| | 1: Alarm overflow |
| | 2: Message box overflow |
| | 3: Cyclic data not in bus cycle |
| | 4: Application bus cycle error |
| | 5: Watchdog |
| | 6: Error handler |
| | 7: Time-out when receiving cyclic data |
| | 8: No not-IO-task available |
| | ZINFO3 - Position 0: Station number |
| | ZINFO3 - Position 11: IO system ID |
| | ZINFO3 - Bit 15: System ID DP/PN |
| 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 |
| 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 standard backplane bus |

| Event ID | Description |
|----------|---|
| | ZINFO2: Slot |
| | ZINFO3: Type identifier |
| 0xE011 | Master project engineering at slave CPU not possible or wrong slave configuration |
| 0xE012 | Error at configuration standard backplane bus |
| 0xE013 | Error at shift register access to standard backplane 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 |
| | ZINFO1: Rack/Slot |
| | ZINFO3: Error type |
| | 4: Rack/Slot (in ZINFO1) undefined |
| | 5: Alarm type (in DatID) undefined |
| | DatID: Interrupt type |
| 0xE030 | Error of the standard backplane bus |
| 0xE033 | Internal error - Please contact the hotline! |
| 0xE0B0 | SPEED7 is not stoppable |
| | 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 |
| | ZINFO1: Error |
| | 4: SSL wrong |
| | 5: Sub-SSL wrong |
| | 6: Index wrong |
| | ZINFO2: SZL-ID |
| | ZINFO3: Index |

| Event ID | Description |
|--------------------|--|
| 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 |
| 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 |

| Event ID | Description |
|----------------------|---|
| | ZINFO1 - Position 0: 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 |
| | 70: SFB |
| | 97: VDB |
| | 98: VSDB |
| 99: VFC | |
| 100: VSFC | |
| 101: VFB | |

| Event ID | Description |
|----------|---|
| | 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 |
| | 23357: 955-C000M00 |
| | 24576: 955-C000050 |
| | 35025: 955-C00MC10 |
| | 36351: FSC-C000S40 |
| | 36794: FSC-C000M40 |
| | 37260: 955-C000S40 |

| Event ID | Description |
|----------|--|
| | 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 |
| | 4940: FSC-C000S30 |
| | 5755: 955-C0ME040 |
| | 6843: FSC-C0NE040 |
| | 8561: FSC-C000S20 |
| | 9012: FSC-C000M20 |
| | 13895: 955-C000060 |

| Event ID | Description |
|----------|--|
| | 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. The configuration is accepted, but the PLC can not go to RUN. |
| | ZINFO1: Required FSC: PROFIBUS |
| | ZINFO1: Required FSC: MOTION |
| | 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 |

| Event ID | Description |
|----------|--------------------|
| | 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 |
| | 47852: 955-C00MC00 |
| | 48709: FSC-C0PE040 |
| | 50574: 955-C000M70 |
| | 52366: 955-C000030 |
| | 53501: FSC-C000030 |
| | 58048: FSC-C000020 |

| Event ID | Description |
|----------|---|
| | 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/memory card 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 |
| | 20046: FSC-C000040 |
| | 21053: 955-C000040 |
| | 22904: 955-C000S00 |
| | 23357: 955-C000M00 |
| | 24576: 955-C000050 |
| | 35025: 955-C00MC10 |

| Event ID | Description |
|----------|--|
| | 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 |
| | 3903: 955-C000S30 |
| | 4361: FSC-C000M30 |
| | 4940: FSC-C000S30 |
| | 5755: 955-C0ME040 |
| | 6843: FSC-C0NE040 |
| | 8561: FSC-C000S20 |

| Event ID | Description |
|----------|---|
| | 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 |
| | 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 |
| 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 (block was not loaded) |
| | 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 |
| 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) |

| Event ID | Description |
|----------|--|
| | 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! |
| | 0: Master system ID |
| | PK: Not user relevant |
| | ZINFO1: Not user relevant |
| | ZINFO2: Slave address |
| | 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 |

| Event ID | Description |
|----------|--|
| | ZINFO3: Number of admissible slaves |
| 0xE70A | PROFIBUS configured, but access way disabled |
| | ZINFO1: Logical base address of the DP master |
| | ZINFO2 - Position 8: DP master system ID |
| 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! |
| | PK: Error code |
| | 1: Error when assigning diagnostic address for slave (no. in ZINFO3) |
| | 2: Error when assigning diagnostic address for master |
| | 3: Error when assigning logical address when de/activating for slave (no. in ZINFO3) |
| | 4: Error when assigning slots for slave (no. in ZINFO3) |
| | 5: Error in DPV1 configuration (inputs) for slave (no. in ZINFO3) |
| | 6: Error in DPV1 configuration (outputs) for slave (no. in ZINFO3) |
| | 7: SubnetID for master (in ZINFO2) invalid |
| | 8: Slave (no. in ZINFO3) could not be configured (CFG length in OB) |
| | 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 |

| Event ID | Description |
|----------|---|
| | 1: Parameter error |
| | 2: Configuration error |
| | ZINFO2: Master system ID |
| | DatID: Not user relevant |
| 0xE780 | Error in configuration of a process image |
| | ZINFO1: Not user relevant |
| | ZINFO2: Logical address |
| | ZINFO3: IO Flag |
| 0xE781 | Address range exceeds process image limit |
| | ZINFO1: Address |
| | ZINFO2: Length of the address range |
| | ZINFO3: Size of the process image |
| | DatID: Address range |
| 0xE801 | CMD - auto command: CMD_START recognized and executed |
| 0xE802 | CMD - auto command: CMD_End recognized and executed |
| 0xE803 | CMD - auto command: WAIT1SECOND recognized and executed |
| 0xE804 | CMD - auto command: WEBPAGE recognized and executed |
| 0xE805 | CMD - auto command: LOAD_PROJECT recognized and executed |
| 0xE806 | CMD - auto command: SAVE_PROJECT recognized and executed |
| | ZINFO3: Status |
| | 0: Error |
| | 1: OK |
| | 32768: Wrong password |
| 0xE807 | CMD - auto command: FACTORY_RESET recognized and executed |
| 0xE808 | Internal message |
| | ZINFO2: Not user relevant |
| | ZINFO3: Not user relevant |
| 0xE809 | Internal message |
| | ZINFO3: Not user relevant |
| 0xE80A | Internal message |
| | ZINFO3: Status |
| | 0: OK |
| | 65153: File create error |
| | 65185: File writing error |
| | 65186: Odd address for reading |
| 0xE80B | CMD - auto command: DIAGBUF recognized and executed |
| | ZINFO3: Status |
| | 0: OK |

| Event ID | Description |
|----------|---|
| | 65153: File create error |
| | 65185: File writing error |
| | 65186: Odd address for reading |
| 0xE80C | Internal message |
| | ZINFO3: Status |
| | 0: OK |
| | 65153: File create error |
| | 65185: File writing error |
| | 65186: Odd address for reading |
| 0xE80D | Internal message |
| 0xE80E | CMD - auto command: SET_NETWORK recognized and executed |
| 0xE80F | Internal message |
| | ZINFO3: Status |
| | 0: OK |
| | 65153: File create error |
| | 65185: File writing error |
| | 65186: Odd address for reading |
| 0xE810 | Internal message |
| 0xE811 | Internal message |
| 0xE812 | Internal message |
| 0xE813 | Internal message |
| 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 message |
| | 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 executed |
| | ZINFO3: Error code |
| 0xE82B | CMD - auto command: CPUTYPE_ORIGINAL recognized and executed |

| Event ID | Description |
|----------|---|
| | ZINFO3: Error code |
| 0xE82C | CMD - auto command: WEBVISU_PGOP_ENABLE recognized and executed |
| 0xE82D | CMD - auto command: WEBVISU_PGOP_DISABLE recognized and executed |
| 0xE82E | CMD - auto command: WEBVISU_CP_ENABLE recognized and executed |
| 0xE82F | CMD - auto command: WEBVISU_CP_DISABLE recognized and executed |
| 0xE830 | CMD - auto command: OPCUA_PGOP_ENABLE recognized and executed |
| 0xE831 | CMD - auto command: OPCUA_PGOP_DISABLE recognized and executed |
| 0xE832 | CMD - auto command: OPCUA_CP_ENABLE recognized and executed |
| 0xE833 | CMD - auto command: OPCUA_CP_DISABLE recognized and executed |
| 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 |
| | ZINFO1: Peripheral address |
| | ZINFO2: Slot |
| | ZINFO3: Data width |
| | DatID: 0x54 Peripheral address is input address |
| | DatID: 0x55 Peripheral address is output address |
| 0xE90A | PROFINET configured, but access way disabled |
| | ZINFO1: Logical address of the IO system |
| | ZINFO3: Station number |
| | ZINFO3: IO system ID |
| | ZINFO3: System ID DP/PN |
| 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 |

| Event ID | Description |
|----------|---|
| | ZINFO2: Slot |
| | ZINFO3: Data width |
| 0xE920 | Configuration error PROFINET |
| | ZINFO1 - Position 0: Error code |
| | 1: Double IP/PROFINET configuration in slot 2 and 4 |
| | 2: PROFINET IO system configured in slot 4 |
| | 3: Too many PROFINET IO controller configured |
| | 4: Virtual device multiple configured |
| | 5: EtherCAT devices for PROFINET CP configured |
| | 6: PROFINET devices for EtherCAT CP configured |
| | 7: PROFINET CP configured at slot 2, although this is not supported |
| | 8: A PROFINET IO system (I-Device) must not be configured on the CP on slot 4 |
| | 9: A PROFINET IO system (controller) must not be configured on the CP on slot 4 |
| 0xE980 | Error when loading the WebVisu project file |
| | ZINFO1: Platform |
| 0xE981 | Error in the configuration of the WebVisu project |
| | ZINFO1: Platform |
| 0xE982 | Internal error of the WebVisu server |
| | ZINFO1: Platform |
| 0xE983 | Hardware configuration of the control is not loaded, WebVisu is not started |
| | ZINFO1: Platform |
| 0xE984 | WebVisu is blocked by the user, start of the WebVisu was prevented |
| | ZINFO1: Platform |
| 0xE985 | WebVisu was started |
| | ZINFO1: Platform |
| 0xE986 | WebVisu was stopped |
| | ZINFO1: Platform |
| 0xE987 | WebVisu was enabled by the user |
| | ZINFO1: Platform |
| 0xE988 | WebVisu was disabled by the user |
| | ZINFO1: Platform |
| 0xE989 | WebVisu and OPC UA project not allowed at the same time |
| | ZINFO1: Platform |
| 0xE9A0 | Error when loading the OPC UA project file |
| | ZINFO1: Platform |
| | ZINFO3 - Bit 0: Error code |
| 0xE9A1 | OPC UA: No FSC enabled |
| | ZINFO1: Platform |

| Event ID | Description |
|----------|--|
| 0xE9A2 | OPC UA: TAR file invalid |
| | ZINFO1: Platform |
| | ZINFO3: Error code |
| 0xE9A3 | OPC UA: Internal error of the OPC UA server |
| | ZINFO1: Platform |
| | ZINFO3: Error code |
| 0xE9A4 | OPC UA: Hardware configuration of the control is not loaded, Server is not started |
| | ZINFO1: Platform |
| 0xE9A5 | OPC UA blocked by user, start of the server was prevented |
| | ZINFO1: Platform |
| 0xE9A6 | OPC UA server was started |
| | ZINFO1: Platform |
| 0xE9A7 | OPC UA server was stopped |
| | ZINFO1: Platform |
| 0xE9A8 | OPC UA was disabled by the user |
| | ZINFO1: Platform |
| 0xE9A9 | OPC UA was enabled by the user |
| | ZINFO1: Platform |
| 0xE9AA | OPC UA: Lock by S7 configuration (access settings) |
| | ZINFO1: Platform |
| 0xE9AB | OPC UA and WebVisu project not allowed at the same time |
| | ZINFO1: Platform |
| 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 |

| Event ID | Description |
|----------|--|
| | 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 |
| | 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 |

| Event ID | Description |
|----------|--|
| | 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 |
| 0xEA0A | SBUS: Internal error (internal plugged sub module wrong) |
| | PK: Not user relevant |
| | ZINFO1: Slot |
| | ZINFO2: Type identifier target |
| | 3: PROFINET-CPU |
| | 4: EtherCAT-CPU |
| | ZINFO3: Type identifier |
| | 3: PROFINET-CPU |
| | 4: EtherCAT-CPU |
| | DatID: Not user relevant |
| 0xEA10 | SBUS: Input periphery address outside the periphery area |
| | ZINFO1: Periphery address |
| | 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 |

| Event ID | Description |
|----------|--|
| | 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: Configuration RS485 interface X2: Value is outside the limits |
| | ZINFO2: Configuration for X2 |
| 0xEA23 | Error: Configuration 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 |
| | 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 |

| Event ID | Description |
|----------|---|
| 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: Too many configured PROFINET IO devices |
| | ZINFO1: Number of configured devices |
| | ZINFO2: Slot |
| | ZINFO3: Maximum possible number of devices |
| 0xEA54 | PROFINET IO controller: Multiple configuration of a periphery address or range too long |
| | 0: Error type |
| | 0: No error |
| | 1: Range too long |
| | 2: Input address already used |
| | 3: Output address already used |
| | 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 IO controller: Too many slots or too high slot number configured |
| | ZINFO1: Rack/slot of the controller |
| | ZINFO2: Device number |
| | ZINFO3: Number of configured slots or too high slot number |
| 0xEA56 | PROFINET IO controller: Too many subslots or too high sub slot number configured |
| | ZINFO1: Rack/slot of the controller |

| Event ID | Description |
|----------|---|
| | ZINFO2: Device number |
| | ZINFO3: Number of configured subslots or too high sub slot number |
| 0xEA57 | PROFINET IO controller: The port configuration in the virtual device has no effect. |
| 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 |
| | PK: Interface |
| | ZINFO1 - Bit 0: Too many devices |
| | ZINFO1 - Bit 1: Too many devices per second |
| | ZINFO1 - Bit 2: Too many input bytes per millisecond |
| | ZINFO1 - Bit 3: Too many output bytes per millisecond |
| | ZINFO1 - Bit 4: Too many input bytes per device |
| | ZINFO1 - Bit 5: Too many output bytes per device |
| | ZINFO1 - Bit 6: Too many productive connections |
| | ZINFO1 - Bit 7: Too many input bytes in the process image |
| | ZINFO1 - Bit 8: Too many output bytes in the process image |
| | ZINFO1 - Bit 9: Configuration not available |
| | ZINFO1 - Bit 10: Configuration invalid |
| | ZINFO1 - Bit 11: Refresh interval too small |
| | ZINFO1 - Bit 12: Refresh interval too large |
| | ZINFO1 - Bit 13: Invalid device number |
| | ZINFO1 - Bit 14: CPU is configured as an I device |

| Event ID | Description |
|----------|---|
| | ZINFO1 - Bit 15: Assume IP address in another way. Is not supported for the IP address of the controller. |
| | ZINFO2 - Bit 0: Incompatible configuration (SDB version not supported) |
| | ZINFO2 - Bit 1: EtherCAT: EoE configured but not supported (Possible cause is a too short cycle time of the EtherCAT master system. When using EoE terminals, at least a cycle time of 4ms must be configured.) |
| | ZINFO2 - Bit 2: DC parameter invalid |
| | ZINFO2 - Bit 3: I device configuration invalid (slot gap) |
| | ZINFO2 - Bit 4: MRP configuration invalid (client) |
| | ZINFO2 - Bit 5: Transfer rate 10 MBit (HD/FD) configured but not supported |
| 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 IO controller: Error in the communication stack |
| | OB: StackError.Service |
| | PK: Rack/slot |
| | ZINFO1: StackError.Error.Code |
| | ZINFO2: StackError.Error.Detail |
| | ZINFO3 - Position 0: StackError.Error.AdditionalDetail |
| | ZINFO3 - Position 8: 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 |

| Event ID | Description |
|--|--|
| | 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 - Position 0: ServiceError.AdditionalDetail |
| ZINFO3 - Position 8: 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) |
| | 4: STOP (internal) |
| | 5: STARTUP (cold start) |
| | 6: STARTUP (restart/warm start) |
| | 7: STARTUP (hot restart) |
| | 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 |
| | 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 |
| | 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 |
| 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) |

| Event ID | Description |
|----------|---|
| | 6: STARTUP (restart/warm start) |
| | 7: STARTUP (hot restart) |
| | 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) |
| | 5: STARTUP (cold start) |
| | 6: STARTUP (restart/warm start) |
| | 7: STARTUP (hot restart) |
| | 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 |

| Event ID | Description |
|----------|--|
| | 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 |
| | 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 |
| 0xEA70 | PROFINET IO controller: 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 |

| Event ID | Description |
|------------------|--|
| | ZINFO2: Not user relevant |
| 0xEA72 | Internal error - Please contact the hotline! |
| | 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 |
| | ZINFO1: Filenamehash[0-3] |
| | 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] |
| | 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] |
| | ZINFO2: Filenamehash[4-7] |

| Event ID | Description |
|----------|--|
| | 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] |
| | 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 |
| | ZINFO1: Filenamehash[0-3] |
| | 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] |
| | ZINFO2: Filenamehash[4-7] |
| | ZINFO3: Line |
| | DatID: Current job number |

| Event ID | Description |
|----------|--|
| 0xEA97 | Internal error - Please contact the hotline! |
| | ZINFO3: Slot |
| 0xEA98 | Error in file reading via SBUS |
| | PK: Not user relevant |
| | ZINFO3: Slot |
| | DatID: Not user relevant |
| 0xEA99 | Parameter assignment job could not be executed |
| | 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) |
| | 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 |

| Event ID | Description |
|----------|--|
| | 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 |
| | 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 |
| | 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 |
| | ZINFO1: Status |
| 0xEB02 | System error: Preset configuration does not match actual configuration |

| Event ID | Description |
|----------|--|
| | 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 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 |
| 0xEB04 | Bus: Multiple configuration of a periphery address |
| | ZINFO1: Periphery address |
| | ZINFO2: Slot |
| | DatID: Input |
| | DatID: Output |
| 0xEB05 | System error: Bus structure for isochronous process image not suitable |
| | PK: Not user relevant |
| | ZINFO2: Slot (0=cannot be determined) |
| | DatID: Not user relevant |
| 0xEB06 | System error: Timeout with the isochronous process image |
| 0xEB10 | System error: Bus error |
| | PK: Not user relevant |
| | ZINFO1: Error type |
| | 96: Bus enumeration error |
| | 128: General error |
| | 129: Queue execution error |
| | 130: Error interrupt |
| | ZINFO2: Error on bus enumeration error (ZINFO1) |
| | DatID: Not user relevant |
| 0xEB11 | System error: Error during bus initialization |
| | PK: Not user relevant |
| | DatID: Not user relevant |
| 0xEB15 | Bus FMM message |
| | ZINFO1: FMM message |
| 0xEB20 | System error: Interrupt information undefined |
| 0xEB21 | System error: Accessing configuration data |

| Event ID | Description |
|----------|---|
| | ZINFO2: Not user relevant |
| | ZINFO3: Not user relevant |
| | DatID: Not user relevant |
| 0xEC02 | EtherCAT: Configuration warning |
| | 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) |
| | 14: Slave station number already occupied by virtual device |
| | ZINFO2: Station number |
| 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) |
| | 14: Slave station number already occupied by virtual device |
| | ZINFO2: Station number |

| Event ID | Description |
|----------|---|
| | ZINFO3: Not user relevant |
| | DatID: Not user relevant |
| 0xEC04 | EtherCAT: Multiple configuration of a periphery address |
| | PK: Not user relevant |
| | ZINFO1: Periphery address |
| | ZINFO2: Slot |
| | DatID: Input |
| | DatID: Output |
| 0xEC05 | EtherCAT: Check the set DC mode of the YASKAWA Sigma 5/7 drive |
| | 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 |
| | 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 |

| Event ID | Description |
|-----------------------|---|
| | 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: Recovery bus with all slaves |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous 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: Station not available |
| | DatID: Station available |
| | DatID: Input address |
| DatID: Output address | |
| 0xEC11 | EtherCAT: Recovery bus with missing slaves |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous status |
| | 0: Undefined/Unkown |
| | 1: Init |

| Event ID | Description |
|----------|--|
| | 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 not available |
| | DatID: Station available |
| | DatID: Input address |
| | DatID: Output address |
| 0xEC12 | EtherCAT: Recovery slave |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous 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: Station not available |
| | DatID: Station available |
| | DatID: Input address |
| | DatID: Output address |
| 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 |

| Event ID | Description |
|----------|--|
| | 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 |
| | 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: DC master out of sync |
| | 1: DC slave stations out of sync |
| 0xEC80 | EtherCAT: Bus error resolved |
| | ZINFO1: Logical address of the IO system |
| | ZINFO3 - Position 0: Station number |
| | ZINFO3 - Position 11: IO system ID |
| | ZINFO3 - Bit 15: System ID DP/PN |
| 0xED10 | EtherCAT: Breakdown bus |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |

| Event ID | Description |
|----------|--|
| | 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 |
| 0xED12 | EtherCAT: Breakdown slave |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO2: Diagnostic address of the station |
| | ZINFO3: AIStatusCode |
| | 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 |
| | 23: Invalid sync manager configuration |
| | 24: No valid inputs available |
| | 25: No valid outputs available |
| | 26: Synchronisation error |
| | 27: Sync manager watchdog |

| Event ID | Description |
|----------|--|
| | 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: Station not available |
| | DatID: Station available |
| | DatID: Input address |
| | DatID: Output address |
| 0xED20 | EtherCAT: Bus state change without calling OB86 |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |

| Event ID | Description |
|----------|--|
| | 8: Op |
| | ZINFO2: Diagnostic address of the master |
| | ZINFO3: Number of stations which are not in the same state as the master |
| | DatID: Station not available |
| | DatID: Station available |
| | DatID: Input address |
| | DatID: Output address |
| 0xED21 | EtherCAT: Incorrect bus status change |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous 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 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |

| Event ID | Description |
|----------|---|
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO2: Diagnostic address of the station |
| | ZINFO3: AIStatusCode |
| | 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 |
| | 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 |

| Event ID | Description |
|----------|--|
| | 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: Station not available |
| | DatID: Station available |
| | DatID: Input address |
| | DatID: Output address |
| 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 |
| | 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 |

| Event ID | Description |
|----------|---|
| | 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 |
| | 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 |
| | 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 |

| Event ID | Description |
|----------|--|
| | 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 |
| | 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 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO2: Slave address |
| | ZINFO3: AIStatusCode |
| | 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 |
| | 23: Invalid sync manager configuration |
| | 24: No valid inputs available |
| | 25: No valid outputs available |
| | 26: Synchronisation error |
| | 27: Sync manager watchdog |

| Event ID | Description |
|----------|--|
| | 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: Cause for slave status change |
| | 0: Regular slave status change |
| | 1: Slave failure |
| | 2: Recovery 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 - Position 0: Error register |
| | ZINFO1 - Position 8: MEF-Byte1 |
| | ZINFO2 - Position 0: MEF-Byte2 |
| | ZINFO2 - Position 8: MEF-Byte3 |
| | ZINFO3 - Position 0: MEF-Byte4 |
| | ZINFO3 - Position 8: MEF-Byte5 |
| | DatID: Error code |
| 0xED62 | EtherCAT: Diagnostic buffer CP: Error on SDO access |
| | OB: EtherCAT station address (high byte) |

| Event ID | Description |
|----------|--|
| | PK: EtherCAT station address (low byte) |
| | ZINFO1: Index |
| | ZINFO2: SDO error code (high word) |
| | 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 |
| | 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 - Position 0: Station number |

| Event ID | Description |
|----------|--|
| | ZINFO3 - Position 11: IO system ID |
| | ZINFO3 - Bit 15: System ID DP/PN |
| 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 |
| 0xF9C1 | Restart of the component |
| | OB: NCM_EVENT |
| | 1: OVS: Component start-up request was denied |
| | 3: Component data basis invalid |
| | 6: IP_CONFIG: New IP address assigned by STEP7 configuration |
| | 10: IP_CONFIG: A non-configured new IP address was assigned |
| | 13: HW reset at P bus (for CPU memory reset) |
| | 19: Switch actuation from STOP to RUN causes the restart of the component |
| | 20: MGT: PG command causes the restart of the component |

| Event ID | Description |
|----------|--|
| | 21: MGT: Take-over of component data basis causes the hot restart of the component |
| | 23: Stopping the sub-system after having loaded the already existing consistency-secured SDBs xxxx by the rack component |
| | 25: The SIMATIC procedure has been selected for the time synchronisation of the component. |
| | 26: Component actively established a connection |
| | 28: The SDB xxxx loaded by the rack component is the consistency securing object (SDB type 0x3118) |
| | 29: The component actively disconnected the system connection to the CPU |
| | 31: Inconsistency of the component data base by loading SDB xxxx by the rack component (SDB type 0x3100) |
| | 32: Periphery enabled by S7-CPU |
| | 33: Periphery disabled by S7-CPU |
| | 34: Component STOP due to switch actuation |
| | 35: Component STOP due to invalid configuration |
| | 36: Component STOP due to PG command |
| | 38: SDB xxxx is not registered in the still valid consistency securing object, or it has an incorrect time stamp (SDB type 0x3107), the error is being corrected |
| | 40: Memory reset executed |
| | 44: Consistency of the data base achieved after loading the SDBs xxxx by the rack component (SDB type xxxx) |
| | 45: Remanent part of the component data base is deleted by the rack component after being loaded |
| | 70: Restore factory defaults (same as memory reset of CPU!) |
| | 83: Network interface: automatic configuration, TP/ITP with 10 Mbit/s semi-duplex |
| | 96: The MAC address was retrieved from the system SDB. This is the configured address. |
| | 97: The MAC address was retrieved from the boot EPROM. This is the factory-provided address. |
| | 100: Restart of the component |
| | 101: Component STOP due to deletion of system SDBs |
| | 104: PG command start was denied due to missing or inconsistent configuration |
| | 105: Component STOP due to double IP address |
| | 107: Start-up request by switch actuation was denied due to missing or inconsistent configuration |
| | PK: NCM_SERVICE |
| | 2: Management |
| | 3: Object management system |
| | 6: Time synchronisation |
| | 10: IP_CONFIG |
| | 38: SEND/RECEIVE |

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 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 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 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 | Name | Description |
|--------|----------|--|
| 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 |

| 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 slave |
| 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 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 | Name | Description |
|---------|----------|---|
| 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 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 |
| xy3Eh | Web server diagnostic information |
| xy3Fh | Configuration of Access settings |
| xy74h | Status of the LEDs |
| xy91h | Status information CPU |
| xy92h | Stations status information (DPM) |
| xy94h | Stations status information (DPM, PROFINET IO and EtherCAT) |
| xy95h | Status information (DPM-, PROFINET IO system) |
| 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 |